REPORT NO. ZS-5881-V-4

VALIDATION OF THE CRASH VICTIM SIMULATOR

VOLUME 4

PROGRAMMER'S MANUAL

CALSPAN CORPORATION
ADVANCED TECHNOLOGY CENTER
4455 GENESEE STREET
BUFFALO, NEW YORK 14225

CONTRACT NO. DOT-HS-6-01300
CONTRACT AMOUNT: $492,175

MARCH 1982
FINAL REPORT

Prepared for:

U.S. DEPARTMENT OF TRANSPORTATION
NATIONAL HIGHWAY TRAFFIC SAFETY ADMINISTRATION
WASHINGTON, D.C. 20590

NOTICE

TECHNICAL REPORT STANDARD TITLE PAGE

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.

4. Title and Subtitle	5. Report Date
Validation of the Crash Victim Simulator Volume 4, Programmer's Manual	March 1982
	6. Performing Organization Code W69
7. Author(s) John T. Fleck and Frank E. Butler	8. Performing Organization Report No. ZS-5881-V-4
9. Performing Organization Name and Address Calspan Corporation Advanced Technology Center 4455 Genesee Street Buffalo, New York 14225	10. Work Unit No.
	11. Contract or Grant No. DOT-HS-6-01300
12. Sponsoring Agency Name and Address U. S. Department of Transportation National Highway Traffic Safety Administration 400 Seventh Street, S.W. Washington, D. C. 20590	13. Type of Report and Period Covered Final Report January 1976 - March 1982
	14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

A combined analytical and experimental research project was carried out to develop and examine the validity of an improved version of the computer program used to simulate the three-dimensional dynamic gross motion responses of motor vehicle crash victims. Among the improvements incorporated in the new (CVS-IV) version of the program are a more efficient integration technique, a routine to automatically position a seated occupant in equilibrium, and modifications of the input and output control routines that make it easier to use the program. Measurements of a Part 572 50th percentile male anthropomorphic dummy were made to define an input data set for a simulation model of the dummy. Dynamic pendulum impact tests of dummy sub-assemblies were performed and modeled with the computer program. Detailed comparisons of predicted system responses with those measured in special impact sled tests of the dummy restrained by a three-point restraint belt and by a pre-inflated air bag are also presented.

Results of the project are documented in four manuals as follows:

Volume 1 - Engineering Manual - Part I: Analytical Formulation
Volume 2 - Engineering Manual - Part II: Validation Effort
Volume 3 - User's Manual
Volume 4 - Programmer's Manual

17. Key Words	18. Distribution Statement
Computer Simulation Three-Dimensional Dynamics Mathematical Models Crash Victim Simulation Rigid Body Dynamics	Document is available to the U. S. public through the National Technical Information Service, Springfield, Virginia 22161

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	595	

Form DOT F 1700.7 (8-69)

FOREWORD

This document is one of four manuals that constitute the final report of the research project conducted under Contract No. DOT-HS-6-01300 for the National Highway Traffic Safety Administration. Dr. John T. Fleck and Mr. Frank E. Butler of J & J Technologies, Inc. served as Principal Investigator and Project Engineer, respectively, during their earlier tenure as members of the Calspan Transportation Research Department. Subsequently, Mr. Norman J. DeLeys coordinated the efforts of Calspan and J & J Technologies, Inc., who was retained as a subcontractor to maintain the continuity necessary in preparation of the report.

The Contract Technical Monitor for this project was Dr. Lee Ovenshire of the National Highway Traffic Safety Administration.

This report has been reviewed and approved by:

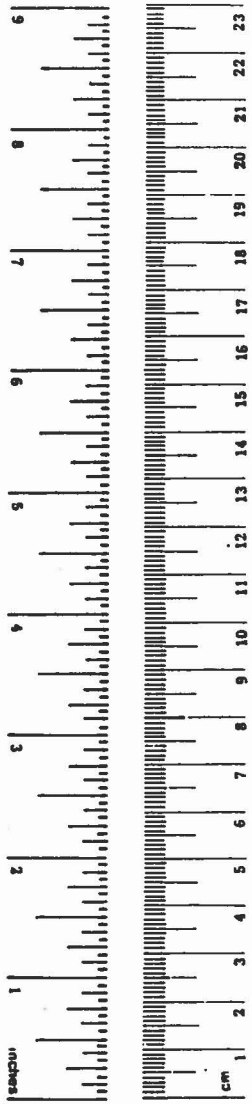
Anthony L. Russo, Head
Transportation Research Department

METRIC CONVERSION FACTORS

Approximate Conversions to Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
		LENGTH		
in	inches	*2.5	centimeters	cm
ft	feet	30	centimeters	cm
yd	yards	0.9	meters	m
mi	miles	1.6	kilometers	km
		AREA		
in^2	square inches	6.5	square centimeters	cm^2
ft^2	square feet	0.09	square meters	m^2
yd^2	square yards	0.8	square meters	m^2
mi^2	square miles	2.6	square kilometers	km^2
	acres	0.4	hectares	ha
		MASS (weight)		
oz	ounces	28	grams	g
lb	pounds	0.45	kilograms	kg
	short tons (2000 lb)	0.9	tonnes	t
		VOLUME		
tsp	teaspoons	5	milliliters	ml
Tbsp	tablespoons	15	milliliters	ml
fl oz	fluid ounces	30	milliliters	ml
c	cups	0.24	liters	l
pt	pints	0.47	liters	l
qt	quarts	0.95	liters	l
gal	gallons	3.8	liters	l
ft^3	cubic feet	0.03	cubic meters	m^3
yd^3	cubic yards	0.76	cubic meters	m^3
		TEMPERATURE (exact)		
°F	Fahrenheit temperature	5/9 (after subtracting 32)	Celsius temperature	°C

*1 in = 2.54 (exactly). For other exact conversions and more detailed tables, see NBS Misc. Publ. 286, Units of Weights and Measures, Price $2.25, SD Catalog No. C13.10:286.



Approximate Conversions from Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
		LENGTH		
mm	millimeters	0.04	inches	in
cm	centimeters	0.4	inches	in
m	meters	3.3	feet	ft
m	meters	1.1	yards	yd
km	kilometers	0.6	miles	mi
		AREA		
cm^2	square centimeters	0.16	square inches	in^2
m^2	square meters	1.2	square yards	yd^2
km^2	square kilometers	0.4	square miles	mi^2
ha	hectares (10,000 m^2)	2.5	acres	
		MASS (weight)		
g	grams	0.035	ounces	oz
kg	kilograms	2.2	pounds	lb
t	tonnes (1000 kg)	1.1	short tons	
		VOLUME		
ml	milliliters	0.03	fluid ounces	fl oz
l	liters	2.1	pints	pt
l	liters	1.06	quarts	qt
l	liters	0.26	gallons	gal
m^3	cubic meters	35	cubic feet	ft^3
m^3	cubic meters	1.3	cubic yards	yd^3
		TEMPERATURE (exact)		
°C	Celsius temperature	9/5 (then add 32)	Fahrenheit temperature	°F

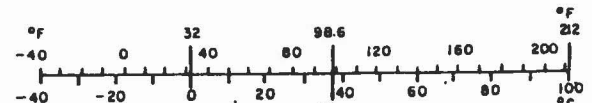


TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1.0 INTRODUCTION

In 1970 Calspan Corporation (formerly Cornell Aeronautical Laboratory, Inc.) began development of a mathematical model for simulating the three-dimensional dynamic responses of a motor vehicle crash victim. Under the joint sponsorship of the Motor Vehicle Manufacturers Association (MVMA) and the National Highway Traffic Safety Administration (NHTSA), the original development and validation of the program was accomplished in two phases (Ref. 1 and 2). Except for a special version of the Phase II crash victim simulation (CVS) program created for the MVMA (Ref. 3), the next major developmental effort was accomplished for the NHTSA and resulted in what was designated as the CVS-III computer program (Ref. 4).

Recognizing the CVS-III as a potentially valuable tool for aiding studies of crew member dynamics during ejection from high-speed aircraft, the Air Force Aerospace Medical Research Laboratory (AFAMRL) sponsored the development of a special version of the program that formed the basis of the AFAMRL "Articulated Total Body" model or ATB (Ref. 5). Later, the ATB model was updated and some new features were added under another contract with the AFAMRL (Ref. 6).

This report documents work performed in the research project entitled "Validation of the Crash Victim Simulator" under Contract No. DOT-HS-6-01300 with the NHTSA which states the general objective as "the development of the CVS to a level that it can be used for a variety of rulemaking activities". A significant goal was "to conduct studies that specifically, quantitatively and validly pertain to the Part 572 dummy in several realistic crash safety compliance test situations". The project consisted of two principal areas of effort: (1) further development, improvement and refinement of the computer program, culminating in a version designated as the CVS-IV, and (2) the performance of detailed measurements and tests to define inputs for modeling the 50th percentile male dummy conforming to government specifications (Ref. 7) and executing computer simulations of experiments performed with the dummy to examine the validity of the model results.

The CVS-IV version of the computer program incorporates many modifications and features developed in this project as well as in conjunction with other closely related research studies (e.g., Ref. 5, 6 and 8). Among the improvements implemented in the CVS-IV are the following:

- a new, more efficient integration technique.
- a routine to automatically position a seated occupant in equilibrium.
- an advanced harness belt formulation that treats interaction of belts connected at a common junction point, belt slippage on deformable segments, and allows use of rate-dependent functions for calculation of belt forces.
- simulation of aerodynamic forces acting on body segments that may be partially shielded.
- improved routines for calculating joint torques.
- use of the main program integrator for computing vehicle and air bag motions.
- the ability to specify the motion of as many as six segments.
- a provision to account for segment principal axes that are not coincident with geometric axes, thereby allowing use of any convenient geometric axis system as the reference for segment input data.
- generality in specifying axes about which segments are rotated, and the sequence of rotations, to achieve a desired initial orientation.
- elimination of the need for multiple output units.
- routines for computing injury criteria values (HIC, HSI, and CSI) and for plotting any output variable(s) against any other variable or time.

During the course of the present study, several interim versions of the computer program were distributed to numerous users throughout the world. However, it should be noted that the modifications of each version were incorporated in such a way that, in most instances, input data decks remained upward compatible and useable with successive versions of the program.

The final report of this project is composed of four volumes:

Volume 1 - Engineering Manual - Part I: Analytical Formulation
Volume 2 - Engineering Manual - Part II: Validation Effort
Volume 3 - User's Manual
Volume 4 - Programmer's Manual

Volume 1 describes the analytical formulations, assumptions and the detailed development of the mathematical equations and relations used in the program.* Volume 2 documents the measurement of the dummy geometric, inertial and joint characteristics and experiments performed to validate computer models of the physical systems tested. The experiments simulated include static tests of an ellipsoidal air bag to check the validity of the idealized bag shape and force algorithms, dynamic pendulum impact tests of dummy component sub-assemblies, and impact sled tests in which the dummy was restrained by an air bag and a three-point belt restraint system (Ref. 9). The third volume provides instruction on how to use the program. Besides giving a detailed description of all data furnished on each input card, it explains the special input and output features and provides examples of program applications along with the Job Control Language needed to execute a simulation run. Volume 4 is intended primarily for use by programmers interested in the detailed structure of the program. Included in Volume 4 are descriptions of each subroutine, cross reference charts showing the subroutines called by other subroutines, labeled common blocks used by each subroutine and usage of each variable in the labeled common blocks in every subprogram, and a complete listing of the computer Fortran source deck.

*See also References 5 and 6 which document the analytical formulation of some algorithms and features not described in detail herein.

2.0 INSTALLING THE CVS PROGRAM

Throughout the development of the CVS program, Calspan has generated and distributed several program tapes as requested by NTHSA. It has been required that the programs contained on these program tapes be current within one month of recent developments to the program, hence the large number of program versions that have been distributed over the past few years. During this time Calspan has strived to insure that any new program versions accept as proper program input those input files that had been developed and were operational for earlier versions of the program. For the most part, this has been achieved. Those few exceptions for which this is not true have been explicitly spelled out in the input description that appears in Section 4.0 of Volume 3. The most recent program tapes distributed by Calspan contain Version 20 of the CVS-IV program as described in this series of reports. These program tapes contain either eight or ten files.

2.1 Format of CVS Program Tapes

The first file contains the 13,138 FORTRAN IV source cards of the 107 subprograms that comprise the CVS program as listed in Section 8.0. This source deck represents the program that is currently operational on the IBM 370/3031 computer system at Calspan and should compile and operate on other IBM 360 and 370 systems. Calspan recommends that the program be compiled with the IBM FORTRAN IV H (level 21.6 or higher) or the H-extended compiler (both with the highest level of optimization) rather than with the G type compilers. The program has compiled without modifications on the VAX 11/780 computer and with modifications on CDC 6000 type and Cyber, Univac 1108 and large scale Burroughs computer systems.

The second file on every program tape has been an up-to-date version of the CVS program input description (Section 4.0 of Volume 3). Although in card image format on the tapes, it is designed to be printed

with program control in column 1 for proper paging. It contains what Calspan considers to be the best description of the program capabilities.

The third and fourth files on the latest program tapes contain the FORTRAN IV source deck and the input description for the three dimensional plotting program VIEW developed by the Air Force Aerospace Medical Research Laboratory (AFAMRL/BBM). This plotting program was written for the CDC computer and has been modified by Calspan to correct those obvious differences between the CDC and IBM FORTRAN IV languages and to accept a new tape 1 output format (Subroutine UNIT1) from the CVS program. Complete documentation for the AFAMRL program View is contained in References 10 and 11.

The succeeding files on the program tapes are paired, the first contains the CVS program input (including IBM job control language) and the second contains the resulting IBM 360 or 370 output of various test cases and simulations that were run at Calspan.

2.2 Required Program Modifications

Although the CVS program FORTRAN source deck contained on the distributed program tapes may compile and execute satisfactorily on other computer systems, slight modifications are necessary to the program due either to differences of other IBM installations from that at Calspan or to those between IBM and other computer manufacturers. They include the following:

a) Function LTIME as contained on the program tapes is a temporary FORTRAN version of an IBM S/370 Assembler Language routine from the Calspan library that measures elapsed CPU time in units of 0.01 seconds. It should be replaced with an equivalent routine to enable Subroutine ELTIME to operate properly on other computer systems. Complete specifications are given in the FORTRAN listing of Function LTIME in Section 8.0. Use of the FORTRAN

version will cause Subroutine ELTIME to give only the count of the number of calls to each routine and not the elapsed CPU time. It will not affect the computational results of the program.

b) The Calcomp plotting subroutines used at Calspan are nonstandard and will not perform properly on other computer plotting systems. If Calcomp plots are desired, it will be necessary to modify the program to accomodate these changes and any others required by the host computer's plotting system. These are discussed on comment cards for Subroutine SLPLOT.

c) If the program is to be compiled on a CDC computer, the double precision computations required by IBM computers are not necessary and the program should be converted to single precision. In order to achieve this the following steps are required:

1) The FORTRAN statement IMPLICIT REAL*8 (A-H,O-Z) that appears near the beginning of most subprograms should be removed.

2) All references to double precision FORTRAN functions should be replaced with their equivalent single precision names, e.g., DSQRT to SQRT, DARSIN to ASIN, DARCOS to ACOS (note IBM require an R here, others do not), DABS to ABS, DMAX1 to AMAX1, etc.

3) All DOUBLE PRECISION FUNCTION statements at the beginning of function subprograms should be replaced with simple FUNCTION statements for EFUNCT, ELONG, EVALFD, FNTERP, RCRT, SPRNGF, VISCOS and XDY.

4) All FORMAT items involving D type conversions should be replaced with E type conversions.

5) Any double precision constants containing a D field for an exponent (e.g., 1.0D0) should be changed to an E field.

Calspan has attempted to remove all constants of this type but it is possible that a few still remain. If one uses a program to convert IBM double precision to CDC single precision, make sure that this program does not convert variable names such as D0 and D1 to E0 and E1.

6) Although not related to double to single precision conversion, all character strings in output FORMAT statements that use the IBM form '...' will have to be changed to the CDC forms *...* or ≠...≠.

d) If the program is to be installed on a Univac 1108 computer, difficulty will be encountered loading the CVS program due to the required allocation of core storage. All of the executable instructions must reside in lower core (the first 64K) followed by the labeled common block storage in extended core storage (beyond 64K). However, there is a further limitation, for variables to reside in labeled common blocks in extended core storage, they must be dimensioned. This requires that each labeled common block be subdivided into two blocks, one containing the dimensioned variables and the other containing the nondimensioned variables. Those subblocks containing the nondimensioned variables must also reside in lower core along with the executable code of the program. The core storage thus required for lower core for the CVS program slightly exceeds the 64K that is available. This will require that an overlay procedure for the CVS program be established (see next section) in order for the program to operate on a Univac 1108 computer.

2.3 Overlay Procedures for the CVS Program

The CVS program (Version 20) requires approximately 600K memory core storage on IBM 360 and 370 computers, 300K on CDC 6000 and Cyber type computers, and 130K on the Univac 1108 computer. These figures are not comparable, they are expressed in the units commonly used for the various

computer systems and can vary depending upon the level of optimization under which the program is compiled, the input and output files assigned and the buffer sizes (which may be installation dependent) allocated. In addition to these large core memory allocations required for the CVS program, the estimated running time may also be large for the host computer. These combinations may cause scheduling and cost problems and result in poor turnaround on some computer systems. Also, it may be impossible to load the CVS program if sufficient core memory is not available.

Some attempts have been made to alleviate these problems by establishing overlay procedures for the CVS program. The structure of the CVS program itself introduces further complications that makes this a very difficult task. In Section 3.0 the overall CVS program flow is described and flow diagrams of the primary subprogram modules (Figure 3.1) are depicted. Also, cross reference charts of the subprogram calls is presented in Section 6.0. It will be observed that some subprograms, even some large program logical groups of subprograms, may be called from more than one calling subprogram. For example, Subroutine DAUX (and all of its subsequent program flow) may be called from Subroutines EQUILB (during the input portion of the program), UPDATE and PDAUX (during the integration portion of the program). This multiplicity of calls greatly complicates any overlay procedure one may wish to employ. One overlay procedure established for the CVS program (Reference 12) solved this problem by setting up multiple load modules for these multiply called subprograms. This procedure was established for a CDC computer and extensive program modifications were required to use the normal CDC overlay procedures.

One major disadvantage of employing overlay procedures for some programs is that the computer running time of the program can be greatly increased if it is necessary to perform multiple roll-ins and roll-outs of overlay segments during the major computational portion of the program. This could easily occur for the CVS program depending upon the structure of the overlay segments especially below the integrator level of the program.

For the overlay procedure that was established by Calspan to run the CVS program on a Univac 1108 computer (see previous section), it was necessary to reduce the core memory of the executable portion of the program by a relatively small amount, and the following overlay segments were used:

1. Main program and utility routines.
2a. The input and initialization routines.
2b. The restart procedure routines.
2c. The program integrator routines.
2d. The post-processing routines.
3. The update and impulse routines.
4. The daux routines.
5. The contact routines.

Referring to Sections 3.0 and 6.0, it will be observed that the overlay segments 2a, 2b, 2c, and 2d are mutually exclusive and thus can share core memory storage. The reduction in overall CVS program storage is small (approximately 5%) but was sufficient to solve the problems encountered on the Univac 1108 computer. This structure has the advantage that there are no extra roll-ins and roll-outs of memory storage because the overlayed portions are no longer required after they have been executed.

In the above structure, it will be observed that each level calls the next level with the following exceptions: level 1 calls all four parts of level 2, level 2a calls level 4, level 2c calls both level 3 and 4, and levels 2b and 2d do not call other levels. Usually, overlay structures are designed to share core storage at the end of a segment sequence and not in the middle of the sequence such as this does. Features that exist in the Univac overlay procedure that do not exist in those for IBM and CDC make this possible. In general, Calspan does not recommend the use of an overlay procedure for the CVS program unless absolutely necessary.

3.0 DESCRIPTION OF CVS PROGRAM FLOW

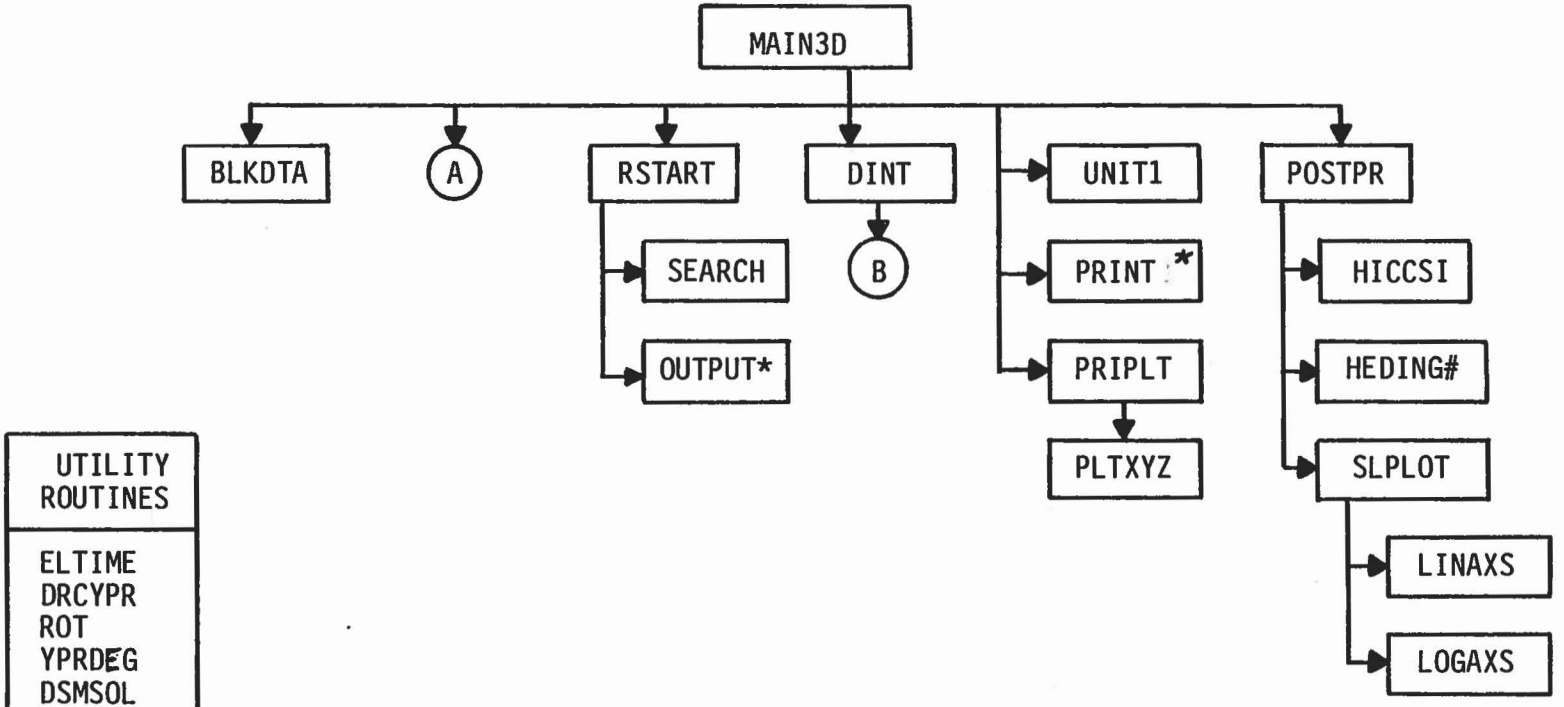
The computer program for the Calspan 3-D Crash Victim Simulation (CVS-IV) Program is comprised of 107 subprogram modules. The program flow of the CVS program is presented in six parts of Figure 3.1 (a-f). Several subprograms that perform certain vector and matrix operations are considered as utility routines, and, since they are called by many of the other subprograms, are not included in the normal program flow. Also, several of the subprograms depicted are called many times, the multiple entries are indicated as either the primary or a secondary entry to the particular subprogram, and subsequent flow proceeds only from the primary entry.

3.1 The CVS Main Program

As shown in Figure 3.1 (a), the main program for the CVS program controls the program initialization, the calling of the required input routines, the restart procedure, the calling of the integrating routine DINT, selected optional output and final post-processing operations. The main program reads input Cards A.1 - A.5. Cards A.1 contain the date, restart control parameters and a description of the run.

If the program is to use the restart procedure, all remaining input is supplied from the restart input file that was generated during a prior computer run, and any program variables to be changed which are supplied on input Cards A.2. No further card input is required. The program then advances the simulation time by reading time point records from the restart input file up to the time it is desired to resume normal operations of the program.

Otherwise, the main program reads input Card A.3 containing the units of distance, force and time to be used for program input and output and the components of the gravity vector, input Card A.4 containing the control parameters for the integrator routine DINT, and input Card A.5



UTILITY ROUTINES
ELTIME
DRCYPR
ROT
YPRDEG
DSMSOL
DOT31
MAT31
CROSS
MAT33
DOT33
DOTT31
DOTT33
XDY
CFACTT

NOTE: PRIMARY FLOW IS DOWN AND TO THE RIGHT.
- PRIMARY ENTRY
* - SECONDARY ENTRY (CHECK PRIMARY ENTRY FOR SUBSEQUENT FLOW)

Figure 3.1 SUBROUTINE FLOW DIAGRAM OF CVS PROGRAM
(A) MAIN PROGRAM

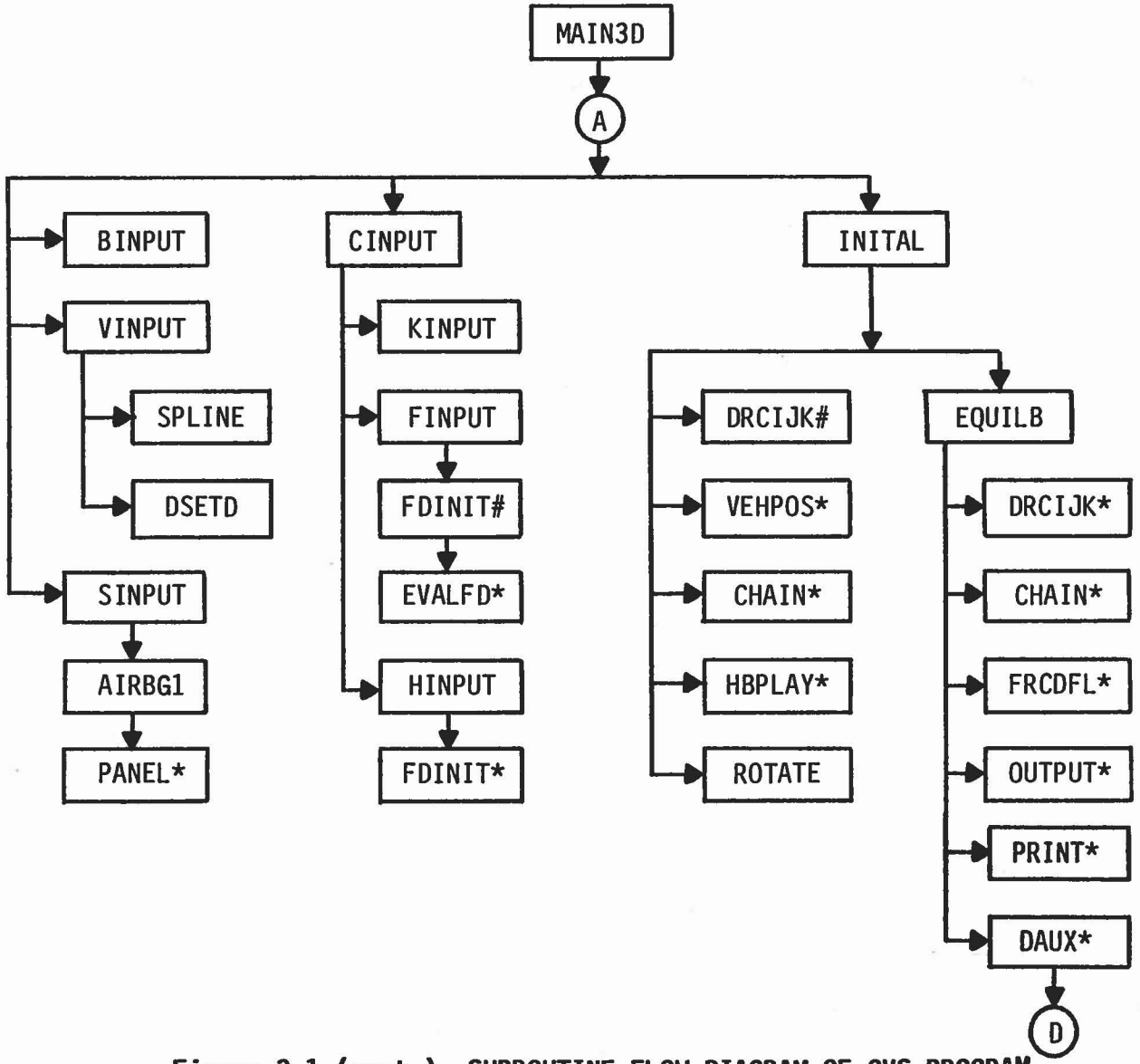


Figure 3.1 (cont.) SUBROUTINE FLOW DIAGRAM OF CVS PROGRAM
(B) INPUT ROUTINES

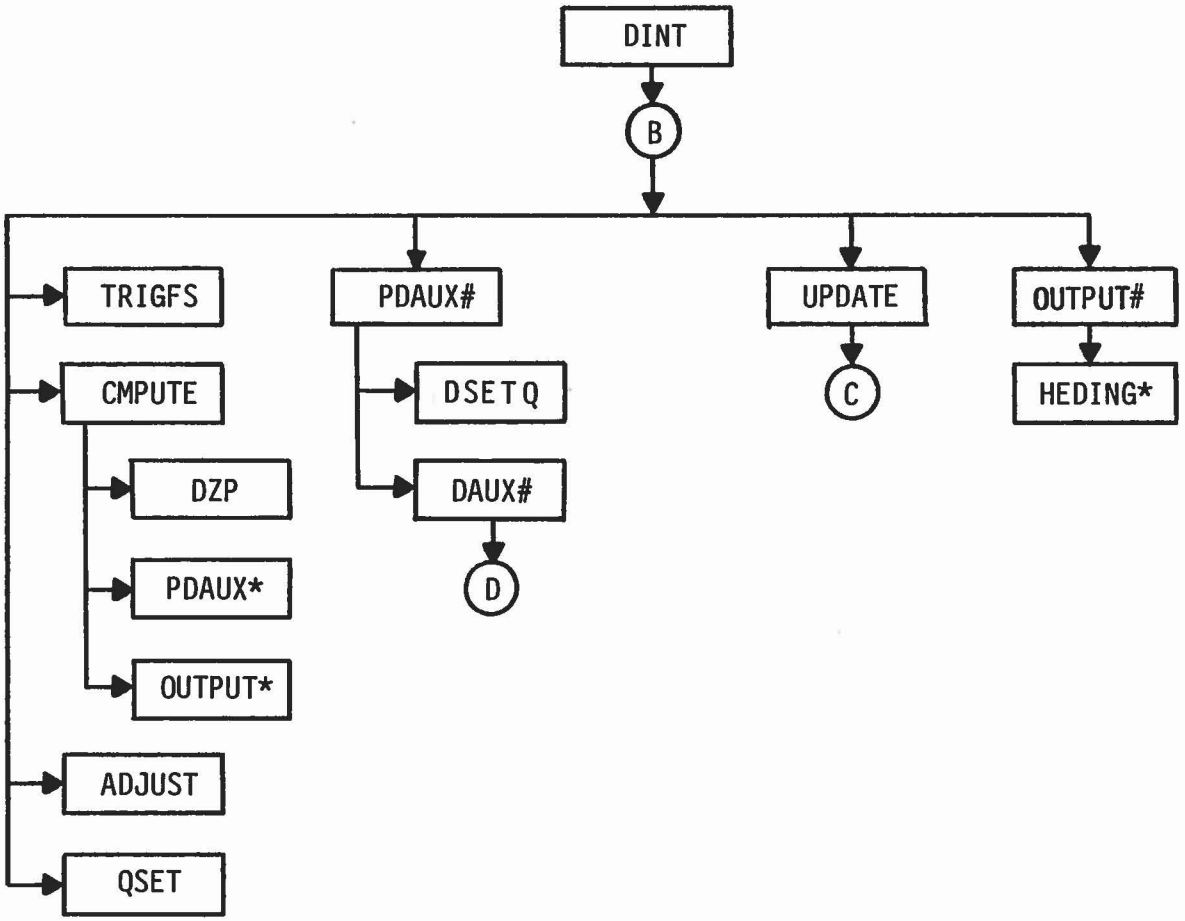


Figure 3.1 (cont.) SUBROUTINE FLOW DIAGRAM OF CVS PROGRAM (C) PROGRAM INTEGRATOR

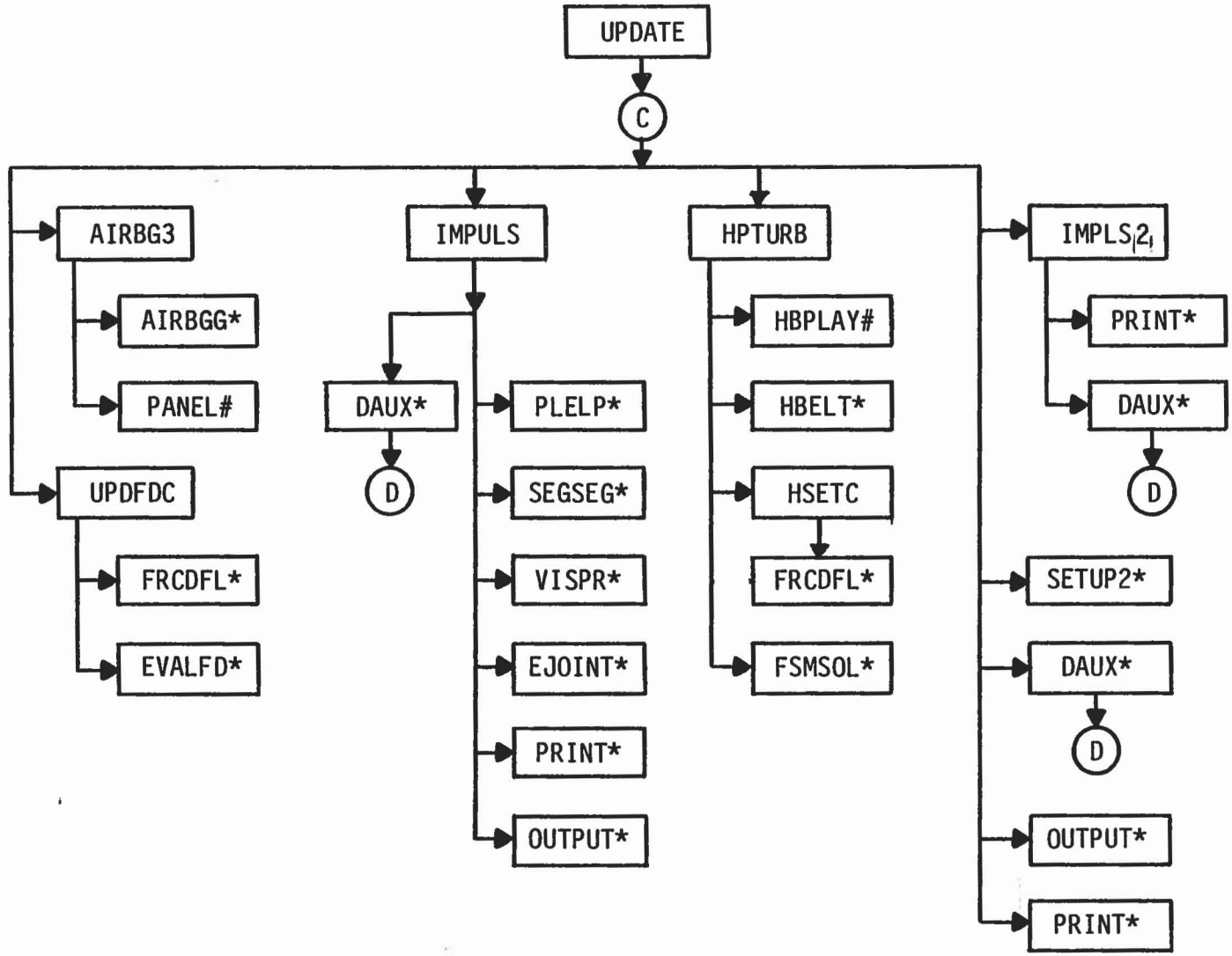


Figure 3.1 (cont.) SUBROUTINE FLOW DIAGRAM OF CVS PROGRAM
(D) SUBROUTINE UPDATE

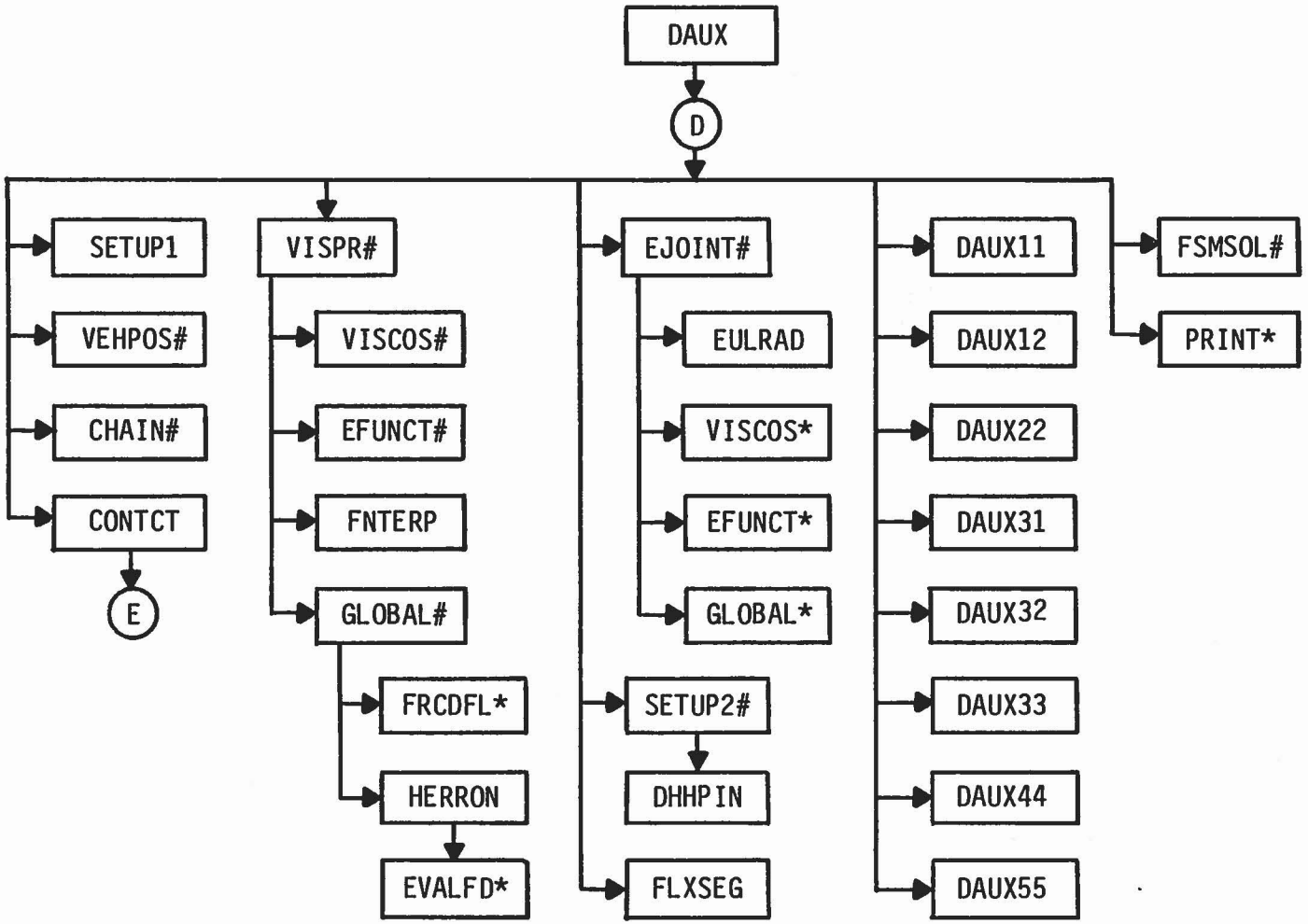


Figure 3.1 (cont.) SUBROUTINE FLOW DIAGRAM OF CVS PROGRAM (E) DAUX ROUTINES

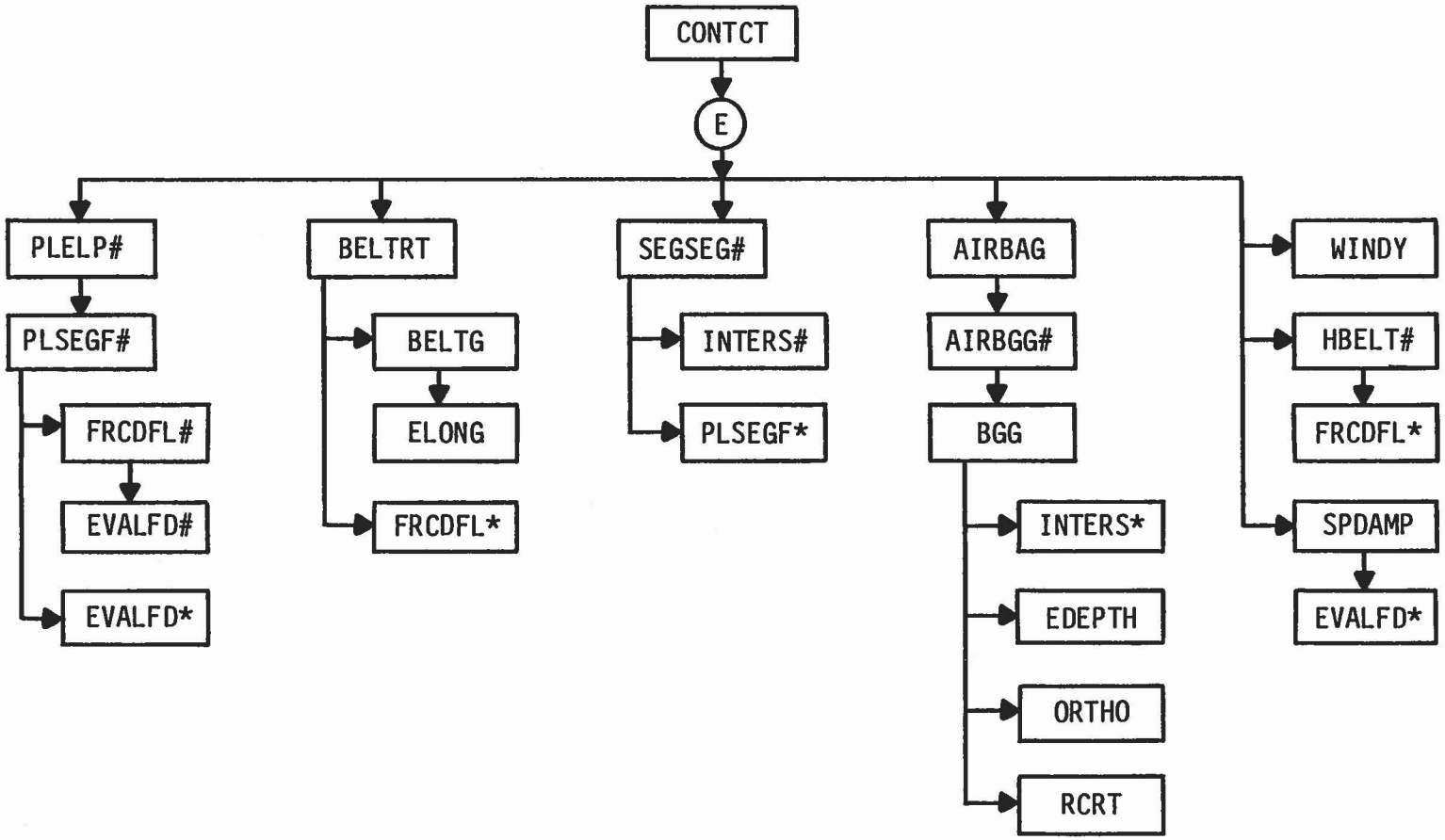


Figure 3.1 (cont.) SUBROUTINE FLOW DIAGRAM OF CVS PROGRAM
(F) CONTACT ROUTINES

containing various output option control parameters. The main program then calls the required input and initial positioning routines which are discussed in Section 3.2.

Each step of the integration loop of the main program advances the simulation time in equal increments of DT seconds (input Card A.4). The step numbers run from zero to NSTEPS and the total simulation run time is therefore NSTEPS*DT seconds. During each step the main program performs these operations:

1) Calls the integrator (Subroutine DINT) or reads a time point record from the restart input file. For step number zero, the integrator is initialized, TIME is set to zero and is not advanced.

2) Optional output is produced at each DT time increment by calls to Subroutines RSTART, PRIPLT, PRINT, UNIT1 and ELTIME as controlled by various optional output control parameters.

After NSTEPS integration steps, the computational portion of the CVS program has been completed. If the time history data has been stored on output unit No. 8, the main program calls Subroutine POSTPR to perform certain post-processing operations including computation of the HIC, HSI and CSI indices, user specified Calcomp plots and writing the tabular time histories on the primary output file. At this point, the current CVS run is terminated.

3.2 The CVS Input Routines

Figure 3.1 (b) depicts the program flow of the input and initial positioning as called by the main program to perform the remaining input required by the CVS program. A short description of this input is given below (a complete detailed description of the variables on each input card is given in Section 4.0 of Volume 3).

Subroutine BINPUT - processes input Cards B.1 - B.7 containing the physical characteristics of the body segments and joints.

Subroutine VINPUT - processes input Cards C.1 - C.4 containing the prescribed motion of specified segments including the vehicle.

Subroutine SINPUT - processes input Cards D.1 - D.8 containing the geometrical description of the contact surfaces (planes, belts, airbags and ellipsoids) and for constraints and symmetry options to be used.

Subroutine CINPUT - processes input Cards E.1 - E.4 containing the function definitions to be used for control of the force deflection functions for the allowed contacts.

Subroutine KINPUT - called by Subroutine CINPUT to process input Cards E.6 and E.7 containing the definitions of the wind and joint restoring force functions.

Subroutine FINPUT - called by Subroutine CINPUT to process input Cards F.1 thru F.7 containing a menu of allowed contacts and associated functions to be used.

Subroutine HINPUT - called by Subroutine CINPUT to process input Cards F.8 containing the input required for the harness-belt systems to be used. Note that Subroutines FINPUT and HINPUT both call Subroutine FDINIT to perform the initialization of the force deflection function tables for each allowed contact that employs them.

Subroutine INITAL - processes input Cards G.1 - G.3 containing the control parameters for the printer plots and the initial position

input data required for the body segments. Subroutine INITAL may be instructed by program input to call Subroutine EQUILB to adjust the initial position parameters to achieve initial equilibrium. If so, input Cards G.4 thru G.6 are processed and Subroutine DAUX (Section 3.5) will be called to solve the system equations.

The remaining input cards are processed by the subroutines that require them when these subroutines are first called. These include input Cards H.1 - H.7 by Subroutine OUTPUT, and input Cards H.8 and I.1 - I.8 by Subroutine POSTPR.

3.3 The CVS Program Integrator

The integration of the state variables is performed by Subroutine DINT as shown in Figure 3.1 (b). Each call to DINT by the main program advances the simulation time by DT seconds in substeps of H seconds where H is allowed to vary between HMIN and HMAX seconds. The integration technique used for each substep resembles the standard 4th order Runge Kutta in that successive estimates are made at both the midpoint and endpoint of the current H interval. The step size is controlled by tests on the predicted and computed values of the state variables after each endpoint computation. If these tests are not satisfied after a specfied number (NDINT) endpoint evaluations, the step size H is halved if it is not less than HMIN. If the tests are satisfied for a predetermined number of consecutive steps, the step size is doubled but never allowed to exceed the input value for HMAX. The complete integrator package is contained in Subroutine DINT and its auxiliary Subroutines TRIGFS, CMPUTE, DZP, ADJUST and QSET. A full description of the CVS variable step exponential integrator is presented in Section 3 of Volume 1.

Subroutine PDAUX acts as an interface between Subroutines DINT and DAUX (Section 3.5) by selecting the derivatives of the integrator state

variables from the CVS program variables and storing the integrated results back into the CVS program variables. It also updates the direction cosine matrices by calling Subroutine DSETQ. Between each integrator time substep, Subroutine UPDATE (Section 3.4) is called to redefine various program parameters for the upcoming time step. At the end of each integrator time substep, Subroutine OUTPUT is called to write the current time point data for the tabular time histories on either secondary output files or output unit No. 8.

3.4 Subroutine UPDATE

The flow for Subroutine UPDATE is presented in Figure 3.1 (d). It is called by Subroutine DINT between integrator time substeps to prepare the integrator for the upcoming time step. This involves updating the parameters for the airbags (Subroutine AIRBG3), the force deflection functions being used by the various contact routines (Subroutine UPDFDC) and the points in play for the harness-belt systems (Subroutine HPTURB). Also, Subroutines IMPULS and IMPLS2 are called to apply impulsive forces produced at initial contacts of certain allowed contacts or by joints entering joint stops if they are specified by program input. If any impulses are applied, it is necessary to reset the program integrator to its initial state.

3.5 The DAUX Routines

Subroutine DAUX is an executive-type routine that controls the flow (Figure 3.1 (e)) of the major computational portion of the CVS program. Its primary call is from Subroutine DINT via Subroutine PDAUX but secondary calls also exist from Subroutines EQUILB and UPDATE. Its primary function is to evaluate the derivatives required by the program integrator. This is accomplished by calling those routines that compute the forces and torques acting on the body segment and joints, set up the system equations and solve these equations. They include the following:

Subroutine SETUP1 - computes the elements of the system equations for the forces and torques acting on the body segments and joints.

Subroutine VEHPOS - computes the linear and angular accelerations for those segments having a prescribed motion including that of the vehicle.

Subroutine CHAIN - computes the linear position and velocity of all body segments from those of the reference segments using the current values of the direction cosine matrices.

Subroutine CONTCT - controls the calling of the subroutines that compute and sum the total forces and torques acting on the body segments produced by the allowed contacts (Section 3.6).

Subroutine VISPR - computes the torques due to the relative position and velocity at the free axes of ball and socket and pinned joints and adds them to the system of equations.

Subroutine EJOINT - computes the torques for Euler joints and adds them to the system of equations.

Subroutine SETUP2 - sets up the equations for constraint forces specified by the program input.

Subroutine FLXSEG - sets up the equations to control the flexible elements, if any, specified by the program input.

Several auxiliary DAUX routines, DAUX11, DAUX12, DAUX22, DAUX31, DAUX32, DAUX33, DAUX44 and DAUX55 (where the numbers represent the indices of the submatrices in equation 4.135 of Volume 1) are then called to set

up and reduce the size of the matrix to represent the system equations. The reduced set of equations is then solved for the constraint forces and torques by calling Subroutine FSMSOL. Subroutine DAUX then performs a "back-up" solution of the system equations to compute the linear and angular accelerations of the body segments.

3.6 The Contact Routines

Subroutine CONTCT processes the menu of allowed contacts and calls the required contact routines as shown in Figure 3.1 (f) to compute and sum the forces and torques acting on the body segments for the following types of allowed contacts.

Subroutine PLELP : plane - ellipsoid contacts.

Subroutine BELTRT: seat belt - ellipsoid contacts.

Subroutine SEGSEG: ellipsoid - ellipsoid contacts.

Subroutine AIRBAG: airbag - ellipsoid contacts.

Subroutine WINDY: ellipsoid - windstream contacts.

Subroutine HBELT: ellipsoid - harness belt contacts.

Subroutine SPDAMP: computes the spring and viscous forces of spring dampers between specified points on selected segments.

It is at this point that any additional contact or force producing routines can be added to the program for special or future applications.

4.0 PROGRAM VARIABLE DICTIONARY BY LABELED COMMON BLOCKS

This section contains a list of all of the variables contained in the labeled common blocks of the CVS program. They are listed in the alphabetical order of the common block names. Following each variable is its dimension, if any, and a short definition. If the variable is supplied as CVS program input, references are indicated to the input card number and a more complete definition may be found in the input description contained in Section 4.0 of Volume 3. Dimensions for the variables will generally determine the indexing rules as given in Table 4.1 unless otherwise indicated.

Table 4.1

INDEXING RULES FOR DIMENSIONS OF CVS VARIABLES

Dimension	Definition of Corresponding Index
3	x, y and z coordinates of a point or components of a vector.
3,3	Elements of a direction cosine matrix.
4	Index for airbag panels (K = 1,NPANEL(J) for J = 1,NBAG).
5	Index for airbags for J = 1,NBAG (sometimes 6 is used).
8	Index for seat belts (J = 1,NBELT). Index for flexible elements (K = 1,NFLX).
12	Index for constraints (K = 1,NQ).
20	Index for forces (J = 1,NBSF, NSSF or NBGSF). Index for belts (J = 1,NBLTPH(I) for I = 1,NHRNSS). Index for spring dampers (J = 1,NSD).
24	Index for pair of values for constraints. The odd (2J-1) and even (2J) refers to segments KQ1 and KQ2 for J = 1,NQ.
30	Index for segments (I = 1,NSEG). Index for joints (J = 1,NJNT). Index for planes (J = 1,NPL). Index for plane forces (J = 1,NPSF).
40	Index for ellipsoids (J = 1,NELP).
60	Index for pairs of values for joints. The odd (2J-1) refers to segment JNT(J), the even (2J) to segment J+1 for J = 1,NJNT.
90	Index for triplets of values for Euler joints. 3J-2, 3J-1 and 3J are used for each J = 1,NJNT.
100	Index for points on harness belt systems (K = 1,N where N is either NPTSPB(J) or NPTPLY(J) for J = 1,NBLTPH(I)).
240	Index for integrator state variables (J = 1,NEQ).

COMMON /ABDATA/

ZDEP	(3,5)	Deployment point of airbag in local reference of 1st reaction panel (Card D.4.c)
DBR	(3,3,5)	Direction cosine matrix of airbag relative to vehicle
DPVCTR	(3,5)	Vector along which airbag c.g. lies during bag inflation
DEPLOY	(3,5)	Location of deployment point
AB	(3,5)	Semiaxes of fully inflated ellipsoid airbag (Cards D.4.b)
B	(9,4,5)	3 x 3 matrix defining ellipsoid $X^TBX=1$ for reaction panel
ZR	(3,4,5)	Location of panel c.g. in vehicle reference (Card D.4.h)
BFB	(3,4,5)	c.g. offset of reaction panel (Card D.4.g)
DRR	(9,4,5)	Direction cosine matrix of reaction panel relative to inertial reference
VBAGG	(5)	Geometric volume of fully inflated airbag
VSCS	(5)	Coefficient of sliding friction of the airbag (Card D.4.f)
SPRK	(5)	Spring constant of a linear spring used to stimulate attachment of the airbag at the deployment point (Card D.4.f)
CK	(5)	Parameter used to stabilize airbag numerical integration (Card D.4.f)
CMASS	(5)	Multiplier to increase or decrease the mass of the airbag to artificially dampen the integrated airbag motion (Card D.4.f)

COMMON /ABDATA/ (cont.)

CYMIN	(5)	Mass flow into the airbag
CYMOUT	(5)	Mass flow out of the airbag
BAGPV	(5)	Undeformed airbag volume
PD	(5)	Airbag pressure differential
VBAG	(5)	Airbag volume
VOLBP	(5)	Total volume of intersection of airbag with contacting segments and panels
PCYV	(5)	Volume of mass flow into airbag at atmospheric pressure at time if initial inflation
PCYMIN	(5)	Mass flow into airbag at time of initial full inflation
PVBAG	(5)	Airbag volume at time if initial inflation
TV1	(3,4,5)	Memory for Subroutines INTERS and EDEPTH for airbag-panel ellipsoid contacts
TV2	(3,10,5)	Memory for Subroutines INTERS and EDEPTH for airbag-segment ellipsoid contacts
SWITCH	(5)	Reciprocal density of airbag at time of initial full inflation
PYMOUT	(5)	Mass flow out of airbag at time of initial full inflation
SCALE	(5)	Ratio (0-1) of linear dimensions of airbag to fully inflated airbag
PREVT		Value of TIME at previous airbag integration step
IFULL	(6)	Indicates that airbag is full inflated

COMMON /CDINT/

UU	(4)	Constants computed by Subroutine TRIGFS
GH	(3,4)	valid for the upcoming integrator time step
E	(3,240)	Intermediate storage of function evaluations in Subroutine DINT
FF	(5,240)	
GG	(5,240)	
Y	(5,240)	
U	(5,240)	
H		Current value of the independent variable step size in Subroutine DINT
HPRINT		Saved value of H while integrating to print point only
HS		Set to zero or H to reset integrator
TPRINT		Value of next print time point
TSTART		Start time of an integration step
ICNT		Count of successive integration steps for which convergence has been successful
IDBL		Maximum value for ICNT before test to double step size is performed
IFLAG		Currently not used by program

COMMON /CEULER/

IEULER	(30)	Current lock-unlock conditions for an Euler joint (see definition under IPIN on Cards B.2)
HIR	(3,3,30)	Direction cosine matrix defining orientation of axes of an Euler joint
ANG	(3,30)	Angles of orientation of an Euler joint

COMMON /CEULER/ (cont.)

ANGD	(3,30)	Time derivative of orientation angles of an Euler joint
FE	(3,30)	Components of torque acting on an Euler joint in joint reference
TQE	(3,30)	Components of torque acting on an Euler joint in inertial reference
CONST	(3,30)	Memory of previous angles of orientation of an Euler Joint

COMMON /CMATRX/

V1	(3,30)	Right hand side of system of equations $B_{11}\ddot{x} + B_{12}\dot{\omega} + B_{13}\ f = V_1$
V2	(3,30)	Right hand side of system of equations $B_{22}\ddot{x} + B_{24}\ t = V_2$
V3	(3,12)	Right hand side of system of equations $B_{31}\ddot{x} + B_{32}\dot{\omega} + B_{35}\ q = V_3$
B12	(3,3,60)	Subarray elements of B_{12}
A22	(3,3,60)	Subarray elements of A_{22}
F	(3,30)	Components of force acting on the joints from the solution of system equations
TQ	(3,30)	Components of torque acting on the joints from the solution of system equations
WJ	(30)	Relative angular velocity of each joint

COMMON /CNSNTS/

PI		FORTRAN Subroutine Library value for Pi, computed by PI = DATAN2(0.0D0,-1.0D0)
RADIAN		Number of radians per degree (PI/180)
G		Resultant of gravity vector (Card A.3)
THIRD		Double precision value for 1/3
EPS	(24)	Values of negative powers of ten, computed by EPS(I) = 10.0D0**(-I)
UNITL		I/O unit of length (Card A.3)
UNITM		I/O unit of force or mass (Card A.3)
UNITT		I/O unit of time (Card A.3)
GRAVITY	(3)	Components of gravity vector (Card A.3)

COMMON /CNTSRF/

PL	(17,30)	Array of parameters that define each plane (See description of Subroutine SINPUT)
BELT	(20,8)	Array of parameters that define each belt (Cards D.3.b - D.3.c)
TPTS	(6,8)	Location of belt tangent points in inertial reference
BD	(24,40)	Array of parameters that define each ellipsoid (See description of Subroutine SINPUT)

COMMON /COMAIN/

VAR	(240)	Integrated function values supplied by Subroutine DINT to Subroutine PDAUX
DER	(240)	Function derivatives supplied by Subroutine PDAUX to Subroutine DINT
DT		Time interval for main program output time points (Card A.4)
HO		Initial integrator step size (Card A.4)
HMAX		Maximum integrator step size (Card A.4)
HMIN		Minimum integrator step size (Card A.4)
RSTIME		Restart time (Card A.1.a)
ISTEP		Current integration step number
NSTEPS		No. of integration steps for duration of simulation (Card A.4)
NDINT		No. of iterations for convergence test for Subroutine DINT (Card A.4)
NEQ		Total number of functions integrated by Subroutine DINT
IRSIN		Restart input unit no. (Card A.1.a)
IRSOUT		Restart output unit no. (Card A.1.a)

COMMON /CONTRL/

TIME	Current simulation time
NSEG	Number of body segments of crach victim, max=30 (Card B.1)
NJNT	Number of joints, max=30 (Card B.1)
NPL	Number of plane definitions supplied on Cards D.2, max=30 (Card D.1)

COMMON /CONTRL/ (cont.)

NBLT		Number of belt definitions supplied on Cards D.3, max=8 (Card D.1)
NBAG		Number of airbag definitions supplied Cards D.4, max=5 (Card D.1)
NVEH		Segment identification number for the vehicle (NVEH=NSEG+1)
NGRND		Segment identification number for the ground (NGRND=NSEG+NBAG+2)
NS		Number of singular segments, i.e., W or at least one component of PHI is zero
NQ		Number of constraints supplied on Cards D.6, final max = 12 (Card D.1)
NSD		Nunber of spring dampers supplied on Cards D.8, max=20 (Card D.1)
NFLX		Total number of interior segments of all flexible elements.
NHRNSS		Number of harness-belt systems supplied on Cards F.8, max=5 (Card D.1)
NWINDF		Number of wind force functions supplied on Cards E.6 (Card D.1)
NJNTF		Number of joint restoring force functions supplied on Cards E.7 (Card D.1)
NPRT	(36)	Indicators that control optional output of the program (Card A.5)

COMMON /CSTRNT/

A13	(3,3,24)	Subarray elements of A_{13} for system of equations $M\ddot{x} + A_{11}f + A_{13}q = U_1$
A23	(3,3,24)	Subarray elements of A_{23} for system of equations $\phi\dot{\omega} + A_{21}f + A_{22}t + A_{23}q = U_2$
B31	(3,3,24)	Subarray elements of B_{31} for system of equations defining constraints
B32	(3,3,24)	Subarray elements of B_{32} for system of equations defining constraints
HHT	(3,3,12)	Array hh^T or $I-hh^T$ for each constraint
RK1	(3,12)	Specified point on segment number KQ1 (Card D.6)
RK2	(3,12)	Specified point on segment number KQ2 (Card D.6)
QQ	(3,12)	Computed force necessary to maintain each constraint
TQQ	(3,12)	Normal vector at the point of contact for each constraint
RQQ	(3,12)	R dot term for constraint equation
HQQ	(3,12)	Reference vector at point of constraint
SQQ	(3,12)	R term for constraint equation
CFQQ	(12)	Coefficient of friction for each constraint
KQ1	(12)	Segment identification number of the 1st specified point (Card D.6)
KQ2	(12)	Segment identification number of the 2nd specified point (Card D.6)
KQTYPE	(12)	Constraint type number (Card D.6)

COMMON /CYDATA/

CYTD	(5)	Gas supply actuator firing time (Card D.4.d)
CYPA	(5)	Atmospheric pressure (Card D.4.d)
CYSP	(5)	Initial gas supply pressure (Card D.4.d)
CYTO	(5)	Initial gas supply temperature (Card D.4.d)
CYVO	(5)	Gas supply reservoir volume (Card D.4.d)
CYCD	(5)	Sonic throat discharge coefficient (Card D.4.e)
CYK	(5)	Ratio of specific heats of supply gas (Card D.4.e)
CYR	(5)	Specific gas constant (Card D.4.e)
CYAT	(5)	Sonic throat area (Card D.4.e)
CYPV	(5)	Vent pressure of the exhaust orifice (Card D.4.e)
CYCDO	(5)	Exhaust orifice discharge coefficient (Card D.4.e.)
CYAO	(5)	Exhaust orifice area (Card D.4.f)
CYPO	(5)	Initial air cylinder gauge supply pressure
CYSS	(5)	Speed of sound
CYLO	(5)	Characteristic length
CYC	(5)	Air cylinder gas constant
CYRHOO	(5)	Initial air cylinder density
CYVMAX	(5)	Air cylinder maximum volume
CYORFC	(5)	Air cylinder exhaust orifice constant
CYRHO	(5)	Density of air cylinder gas supply
CYT	(5)	Temperature of air cylinder gas supply

COMMON /CYDATA/ (cont.)

CYP	(5)	Pressure of air cylinder gas supply
CYV	(5)	Volume of air cylinder gas supply at standard atmospheric pressure

COMMON /DAMPER/

APSDM	(3,20)	Attachment point in local reference of segment M for spring dampers (Card D.8)
APSDN	(3,20)	Attachment point in local reference of segment N for spring dampers (Card D.8)
ASD	(5,20)	Spring and viscous force function coefficients (Card D.8)
MSDM	(20)	Identification number of segment M (Card D.8)
MSDN	(20)	Identification number of segment N (Card D.8)

COMMON /DESCRP/

PHI	(3,30)	Segment principal moments of inertia (Cards B.2)
W	(30)	Segment weight (Cards B.2)
RW	(30)	Reciprocal mass (g/w) for each segment
SR	(3,60)	Joint locations in local reference of adjacent segments (Cards B.3)
HA	(3,60)	Principal line of joint from which flexure angle is measured
HB	(3,60)	Perpendicular to HA (pin axis if joint is pinned)

COMMON /DESCRP/ (cont.)

RPHI	(3,30)	Reciprocal moments of inertia for each segment
HT	(3,3,60)	Principal axes of the joints
SPRING	(5,90)	Flexural and torsional spring characteristics (Cards B.4)
VISC	(7,90)	Flexural and torsional viscous characteristics (Cards B.5)
JNT	(30)	Magnitude indicates the segment identification number that is connected to segment J+1 by joint J (Cards B.3)
IPIN	(30)	Indicator of joint type (Card B.3)
ISING	(30)	Indicator (value=1) that segment is singular
IGLOB	(30)	Input indicator (Card F.4.a) to signify that joint J is to use the globalgraphic option. A nonzero value will be set to index of function to be used.
JOINTF	(30)	The function identification number used to compute the joint restoring force (Card F.5)

COMMON /FLXBLE/

HF	(4,12,8)	Coefficients of quadratic function defining relative orientation of interior segments of flexible elements
B42	(3,3,24)	Subarray elements of matrix B_{42} in the constraint equations for flexible elements
V4	(3,8)	Right hand side of the constraint equations for flexible elements.
NFLEX	(3,8)	The identification numbers of reference, interior and terminating segments for each interior segment

COMMON /FORCES/

PSF	(7,30)	Array of output values for plane-segment contacts
BSF	(4,20)	Array of output values for belt-segment contacts
SSF	(10,20)	Array of output values for segment-segment contacts
BAGSF	(3,20)	Array of output values for airbag-segment contacts
PRJNT	(7,30)	Output arrays for joint parameters
NPANEL	(5)	Number of reaction panels for each airbag (J=1, NBAG)
NPSF		Number of plane-segment contacts (Max=30)
NBSF		Number of belt-segment contacts (Max=20)
NSSF		Number of segment-segment contacts (Max=20)
NBGSF		Number of items to be printed for airbag-segment contacts (Max=20)

COMMON /HRNESS/

BAR	(15,100)	Coordinates of points in local reference (Cards F.8.d)
BB	(100)	Lengths of individual belt segments between reference points
BBDOT	(100)	Time derivative of belt segment lengths
PLOSS	(2,100)	Energy loss of individual belt segments
XLONG	(20)	The initial slack of each belt (Cards F.8.c)
HTIME	(2)	Previous value of TIME for Subroutine HPTURB

COMMON /HRNESS/ (cont.)

IBAR	(5,100)	Array of indicators containing KS, KE, NF index, NPD and NPR (Cards F.8.d) for each point
NL	(2,100)	Pointers to the IBAR and NTHRNS arrays for each point in play
NPTSPB	(20)	Number of points per belt (Cards F.8.b)
NPTPLY	(20)	Number of points in play per belt
NTHRNS	(20)	Index to NTAB array defining the force deflection functions for each belt
NBLTPH	(5)	Number of belts per harness (Card F.8.a)

COMMON /INTEST/

SGTEST	(3,4,30)	Integrator convergence test input numbers (Cards B.6)
XTEST	(3,120)	Integrator convergence test numbers setup by PDAUX for DINT
SEGT	(120)	Segment identification of integrator variable
REGT	(120)	Identification (ANG VEL, ANG ACC, LIN VEL or LIN ACC) of type of integrator variable

COMMON /JBARTZ/

MNPL	(30)	Number of segments to contact each plane (Card F.1.a)
MNBLT	(8)	Number of segments to contact each belt (Card F.2.a)
MNSEG	(30)	Number of segments to contact each segment (Card F.3.a)
MNBAG	(6)	Number of segments to contact each airbag

COMMON /JBARTZ/ (cont.)

MPL	(3,5,30)	Segment and ellipsoid identification numbers for each plane-segment contact
MBLT	(3,5,8)	Segment and ellipsoid identification numbers for each belt-segment contact
MSEG	(3,5,30)	Segment and ellipsoid identification numbers for each segment-segment contact
MBAG	(3,10,6)	Segment and ellipsoid identification numbers for each airbag-segment contact (Cards F.4)
NTPL	(5,30)	Index to NTAB array for each plane-segment contact
NTBLT	(5,8)	Index to NTAB array for each belt-segment contact
NTSEG	(5,30)	Index to NTAB array for each segment-segment contact

COMMON /RSAVE/

XSG	(3,20,3)	Points in local segment reference for first three types of time history output (Cards H.1-H.3)
DPMI	(3,3,30)	Direction cosine matrix of principal moment of inertia to local geometric reference coordinate system for each segment
LPMI	(30)	Indicator that local geometric does not correspond to principal moment of inertia reference coordinate system for each segment (Cards B.2.i1)
NSG	(7)	Number of segments for each type of time history output (Max=20) (Card H.1-H.7)
MSG	(20,7)	The segment identification numbers for each type of time history output (Cards H.1-H.7)

COMMON /SGMNTS/

D	(3,3,30)	Segment direction cosine matrix
WMEG	(3,30)	Segment angular velocity in local reference
WMEGD	(3,30)	Segment angular acceleration in local reference
U1	(3,30)	Total external forces on each segment
U2	(3,30)	Total external torques on each segment
SEGLP	(3,30)	Segment c.g. linear position in inertial reference
SEGLV	(3,30)	Segment c.g. linear velocity in inertial reference
SEGLA	(3,30)	Segment c.g. linear acceleration in inertial reference
NSYM	(30)	Indicators that control the symmetry options for body segments (Cards D.7)

COMMON /TABLES/

MXNTI		Dimension (50) of NTI array
MXNTB		Number of elements in the NTAB array
MXTB1		Number of elements in TAB array used to define functions
MXTB2		Total number of elements in TAB array
NTI	(50)	Index pointers to the TAB array for data defining function no. I.
NTAB	(500)	Index pointers to TAB array for each function used for allowed contacts
TAB	(2600)	Subdivided into arrays containing function definitions and update information for each allowed contact

COMMON /TEMPVI/

CREST		Coefficient of restitution for current impulse
TTI	(3)	Value of U1 array for impulse
R1I	(3)	Value of RK1 for current constraint or impulse
R2I	(3)	Value of RK2 for current constraint or impulse
JSTOP	(4,2,30)	Indicators to signify joint is in joint stop

COMMON /TEMPVS/

Variables is this labeled common block are temporary for each subroutine that refers to it.

COMMON /TITLES/

DATE	(3)	Date of computer run in 12 alphanumeric characters (Card A.1.a)
COMENT	(40)	160 character description of the run (Cards A.1.b - A.1.c)
VPSTTL	(20)	80 character description of the crash vehicle deceleration (Card C.1)
BDYTTL	(5)	20 character description of the crash victim (Card B.1)
BLTTTL	(5,8)	20 character description of each belt (Cards D.3)
PLTTL	(5,20)	20 character description of each plane (Cards D.2)
BAGTTL	(5,6)	20 character description of each airbag (Cards D.4)

COMMON /TITLES/ (cont.)

SEG	(30)	4 character segment nomenclature (Cards B.2)
JOINT	(30)	4 character joint nomenclature (Cards B.3)
CGS	(30)	1 character plot symbol of the segment C.G. (Cards B.2)
JS	(30)	1 character plot symbol of the joint location (Cards B.3)

COMMON /VPOSTN/

ZPLT	(3)	Printer plot coordinates of the vehicle reference origin (Card G.1.a)
SPLT	(3)	Scale factors for the printer plot axes (Card G.1.b)
AXV	(3,6)	Unit vector of deceleration impulse direction
VATAB	(6,101,6)	Tables of computed or supplied (Cards C.3 or C.4) values of linear (1-3) and angular accelerations (4-6) of vehicle motion
VTO	(6)	Beginning time point of the deceleration table input (Card C.2)
VDT	(6)	Fixed time interval for deceleration table input (Card C.2)
TIMEV	(6)	Time duration of the deceleration impulse (Card C.2)
OMEGAV	(6)	Frequency of the half-sine wave deceleration type (Card C.2)
NVTAB	(6)	Number of points in deceleration table. Sign determines type (Card C.2)
INDXV	(6)	Segment identification number for each specified motion definition (MSEG on Card C.2.a or NVEH)

COMMON /WINDFR/

WTIME	(30)	Initial time that segment penetrates wind
QFU	(3,5)	Unit vector for force application
QFV	(3,5)	Vector for torque application
IWIND	(30)	Indicator that wind has been penetrated
MWSEG	(5,30)	Identification numbers for the application of wind forces on each segment (Cards F.7)
NFVSEG	(6)	Segment identification number for each force function (Cards D.9)
NFVNT	(6)	Function identification number for each force function (Cards D.9)

5.0 ANALYSIS VARIABLE DICTIONARY

This section contains a list of analytical symbols and corresponding program variables. The analytical sympols are given in alphabetical order in the table which also includes a brief description of the variables and the page number in Volume 1 (Engineering Manual - Part 1: Analytical Formulation) where the variable is best defined or expressed in equations.

Analysis Variable	Program Variable	Description	Pg of Best Explanation*
A,B	A,BD,B	General Variables defining Matrices	18
A	A,BD,B	Matrix defining the contact ellipsoid	18
A_M	BD (7,L)	Contact ellipsoid matrix for segment M	170
A_{11}			
A_{13}	A13(,,2*k)		
A_{21}	A2	Matrix Coefficients for System Equations	118
A_{22}			
A_{23}	A23(,,2*k)		
a,b,c		Arbitrary vectors used in discussion of and cross product identity	11
a,b,c		Components of a vector in local coordinates	63
a_0	F(3,I)		
a_1	F(4,I)	Coefficients of the polynomial approximation to f(x,t) in the exponential integrator	36
a_2	F(5,I)		
a		Scaled semi-axis (X axis of the air bag)	190
a_1,b_1,c_1	ABC, AB	Semi-axes of the fully inflated air bag	191
a,b,c		Semi-axes of the airbag during inflation	191

* PAGE NUMBERS ARE REFERENCED TO VOLUME I

Analysis Variable	Program Variable	Description	Pg of Best Explanation
B_{11}	B2(, , 2*J)	Matrix Coefficients of the System Equations	118
B_{22}			
B_{12}			
B_{33}			
C		Constant matrix used to define locked joint constraint	72
C_{11}		Matrix coefficients of the reduced system equations used to solve for constraint forces and torques	128
C_{12}			
C_{13}			
C_{22}			
C_{33}			
C_0		Coefficients of the cubic part of the force function for computing force deflection	210
C_1	TAB		
C_2			
C_3			
C_f	CF, CFQQ VSCS	Coefficient of friction between two surfaces	173

Analysis Variable	Program Variable	Description	Pg of Best Explanation
D_n	D(, , N)	Direction cosine matrix of the n^{th} local system	6
d		Scalar corresponding to the distance vector	76
d_{mn}	D(M, N,)	Element in the m^{th} row and n^{th} column of the direction cosine matrix (D)	6
e_1		Exponential forms defined recursively. Used in the exponential integrator	36
e_2			
e_3			
E		Ellipsoid matrix	182

Analysis Variable	Program Variable	Description	Pg of Best Explanation
Fnk	U1	K^{th} external force applied to segment N	65
fnj	TQ	Constraint force on the n^{th} segment applied at joint j	65
f(x, t)		Functional form of the derivative $\dot{x} = f(x, t)$	36
f_{nor}		Normal component of the contact force	173
$f_i(\ )_{\zeta=A_1B}$		Function of penetration parameter used to describe forces resulting from contact	176
$F_C(\ )$		Force function of penetration parameter described by cubic	209
$F_Q(\ )$		Force function of penetration parameter described by quadric	211
G		Deflection factor used in force-deflection routine	207

Analysis Variable	Program Variable	Description	Pg of Best Explanation
h		Arbitrary vector used in discussion of dot and cross product	11
h_n	HB	Unit vector along pin axis in segment n's local system	73
h		Unit vector along pin axis in inertial reference	72
h		Unit vector in the direction of ρ	77
h_1	D1	Defined by $h_1 = {}^{h}/_{2}$ for step 2 of the exponential integrator	37
h		Integration step size in the exp. integrator	37
h_A	HA(, 2*J-1)	Orthogonal unit vectors fixed in segment k where segment k is joined to segment j + 1 thru joint j	133
h_B	HB(, 2*J-1)		
h_A'	HA(, 2*J)	Orthogonal unit vectors fixed in segment j+1, used to define joint flexure and twist torque	133
h_B'	HB(, 2*J)		
h_B		Unit vector defined by rotating h_B through an angle about the $h_A \otimes h_A'$ axis	133
i, j, k		Unit vectors defining orthonormal inertial reference system	4
im, jm, km		Orthonormal vectors defining the m^{th} local system	4
I		Identity matrix	4

Analysis Variable	Program Variable	Description	Pg of Best Explanation
l_n	BD(I+3, N)	Offset of ellipsoid n from the n^{th} local system origin	17
l_n		Vector locating the center of contact ellipsoid from segment n's c.g in n's local system	83
M_n	W(N)	Mass matrix of the n^{th} segment	119
m	M	Used to define m^{th} segment	91
N	NSEG	Total number of segments	118
n		Used to define the n^{th} arbitrary segment most often as a subscript	83

Analysis Variable	Program Variable	Description	Pg of Best Explanation
P_1, P_2, P_3	P1, P2, P3	Location of three points defining a plane	14
P	P	Penetration distance for the specification of contact forces (ellipsoid-ellipsoid)	179
PA		Penetration distance of airbag into contacting ellipsoid	193
PB		Penetration distance of contacting ellipsoid B into airbag	193
P		Defined by $\|PA - PB\|^2$ for airbag	193
q_1, q_2, q_3		Unit vectors representing lines in a plane	15
q	QQ(, k)	Constraint force for position constraint	77
q	QQ	Constraint force for rolling constraint	79
q	QQ	Constraint force for sliding constraint	81
q_0, q_1, q_2		Coefficients used in the quadratic unloading curve in force deflection routine	212

Analysis Variable	Program Variable	Description	Pg of Best Explanation
$\mathbf{r}$		Vector from ellipsoid center to ellipsoid surface	18
$\mathbf{r}_n$		Vector locating a point in segment n in segment n's local system	74
r_{nj}	SR(,2*J)	Location of joint j in n's local system	22
r_n		Vector locating a point in segment n in segment n's local system. Used for position, sliding and rolling constraints	79
r_A		Radius of curvature of the airbag	194
r_B		Radius of curvature of the contacting ellipsoid	194
r_C		Defined by $r_C = (r_A + r_B) / 2$	194
r		Radius of constructed circle for airbag contact computation	194
R		Absorption factor used in force-deflection routine	207
S_1	SPRING(1, JT)	Constants used to define spring torques at the joints	136
S_2	SPRING(2, JT)		
S_3	SPRING(3, JT)		
S_4	SPRING(4, JT)		
S_5	SPRING(5, JT)		

Analysis Variable	Program Variable	Description	Pg of Best Explanation
t	TQQ	Unit Vector defined normal to a plane	82
t	TQ	Constraint torque at the joint	74
t		Unit vector normal to the ellipsoid at r_n in inertial reference	19
t_n		Unit vector normal to the ellipsoid at r_n in n's local system	82
t	T	Independent variable of integration	35
t		Outward normal to the plane	82
T_A	TA	Vector from anchor point A to fixed point (used in belt routine)	182
T_B	TB	Vector from anchor point B to fixed point (used in belt routine)	
T_C	TC	Vector defining the belt plane (used in belt routine)	
T	CV	Viscous torque at the joint	135
T_S	CSA, CSB	Spring torque at the joint	135
U_{2n}	U2 (, N)	Right hand side of the joint constraint equation for segment n	70
U_x		Unit vector in the positive x direction of the panel attached to the airbag	190

Analysis Variable	Program Variable	Description	Pg of Best Explanation
V_{3n}	V3(, N)	Right hand side of distance, sliding or rolling constraint equation for segment n.	126
V_{REL}		Relative velocity between contact ellipsoids during slide	78
V_1^*	V1(, J)	Right side of reduced system equations used to solve for the constraint forces and torques	128
V_2^*	V2(, J)		
V_3^*	V3(, k)		
V_r	VR	Relative velocity between the ellipsoid and plane at the point of contact	173
V_{rt}	TT	Tangential component of V_r	173
V_1	VISC(1, JT)	Constants used to compute viscous joint torques	137
V_2	VISC(2, JT)		
V_3	VISC(3, JT)		
V_g	VBAG	Geometric volume of the airbag	191
V_b		Instantaneous volume of the airbag	205
V_s		Volume of revolution of a sector of a circle	194
V_R		Volume of a ring	196
V		Volume of circle intersection	196

Analysis Variable	Program Variable	Description	Pg of Best Explanation
x_i	SEGLP(, I)	Vector position of segment i in inertial reference	63
x_0, y_0, z_0	XO( )	Components of c.g position in inertial coordinates	63
x	DER(I)	Derivative of the state x(t)	36
x(o)	F(1, I)	Value of the state at the start of the integration step	36
x(t)		Value of the state for t } in the exponential integration	36
x(h/2)		Value of the state for h/2 } in the exponential integration	38
x(h)		Value of the state for h } in the exponential integration	39
x(t+h)		Value of the state for t + h } in the exponential integration	37
x_E	XE	Vector to the center of the belt plane ellipse from the center of the contacting ellipsoid	182
x_A	XA	Location of airbag c.g. in inertial coordinated	193
x_B	BD	Location of the segment c.g to which the contacting ellipsoid is attached	193

Analysis Variable	Program Variable	Description	Pg of Best Explanation
Y		Arbitrary vector locating a point in segment m in inertial reference	4
y_n	PSF,GSF,SSF	Vector from the c.g of segment n to the point of application of the contact force	172
y_2		Parameter used to define the inertial spike in the force deflection routine	210
z_A	ZA	Location of anchor point A relative to the contacting ellipsoid center	182
z_B	ZB	Location of anchor point B relative to the contacting ellipsoid center	182
z	ZR	Location of the center of the primary reaction panel with respect to the vehicle reference for airbag	190
z_d	DEPLOY	Location of the airbag deployment point with respect to the center of the primary panel	190
z_b	ZA	Location of the airbag center with respect to the vehicle origin	190

Analysis Variable	Program Variable	Description	Pg of Best Explanation
α	F(2, I)	Constant used in the approximation of the function f(x, t) for the exponential integrator	36
α_2	PL(11, J)	Parameters used to define the extent of a finite plane	171
α_3	PL(12, J)		
α		Defined by the equation $\alpha = r_c \sin\phi$ used for airbag volume of contact computations	196
β	BTE	Distance represented by the vector $\vec{P}$, ($\beta = \lvert\vec{P}\rvert$)	14
β		Vector connecting the centers of two contacting ellipsoids used for sliding and rolling constraints	83
β	BTE	Distance from segment c. g to contacting plane	172
β_1	PL(8, J)	Parameters used to define the size of a finite plane	171
β_2	PL(9, J)		
β_3	PL(10, J)		
$\tilde{\beta}$	BET	Distance from ellipsoid center to belt plane	182
β		Parameter defined by $\beta^2 = r_A^2 - \alpha$ in the computation of the airbag contact volume	197

Analysis Variable	Program Variable	Description	Pg of Best Explanation
γ		Parameter used to define the tangent point in the belt plane	183
δ		Distance between point of maximum penetration and the ellipse resulting from the plane-ellipsoid contact	172
δ		Force deflection parameter, a geometric property of the surface	207
δ_i $i=1,2$		Values of δ used to describe different regions of the force deflection curve	208
δ_{cubic}		Values of δ used to define the specific force deflection curve	209
δ_{quad}			
δ_{ref}			
δ_{co}			
$\triangle$		Parameter defined by the equation $\triangle = \delta_{ref} - \delta_{cubic}$	209

Analysis Variable	Program Variable	Description	Pg of Best Explanation
θ		The flexure angle used in computing joint torque	133
λ		Arbitrary scalar used in the constraint equations	77
μ		Parameter used to compute r_n' for the sliding and rolling constraints	85
μ		Scalar proportionality constant used to compute contact forces	170
f		Shortest vector from origin to plane	13
f		Vector defining the fixed distance in the fixed distance constraint	75
f	CFQQ(F)	Friction coefficient for the sliding constraint	80

Analysis Variable	Program Variable	Description	Pg of Best Explanation
τ_{ex}	TQ(, N)	Torques from external sources on the n^{th} segment	67
τ_{cons}	TQQ(, N)	Torques (couples) due to constraints on the n^{th} segment	67
ϕ_n	PHI(, N)	Inertia tensor for the n^{th} segment	118
ϕ_{xx} ϕ_{yy} ϕ_{zz}	BPHI	Principal inertial components of the airbag	192
ϕ_A, ϕ_B		Angle of a line drawn from the center of the circle to points of intersection. Used to compute contact volume for the airbag	197
ω_n	WMEG(, N)	Angular velocity vector for the n^{th} segment in n's local coordinate system	67

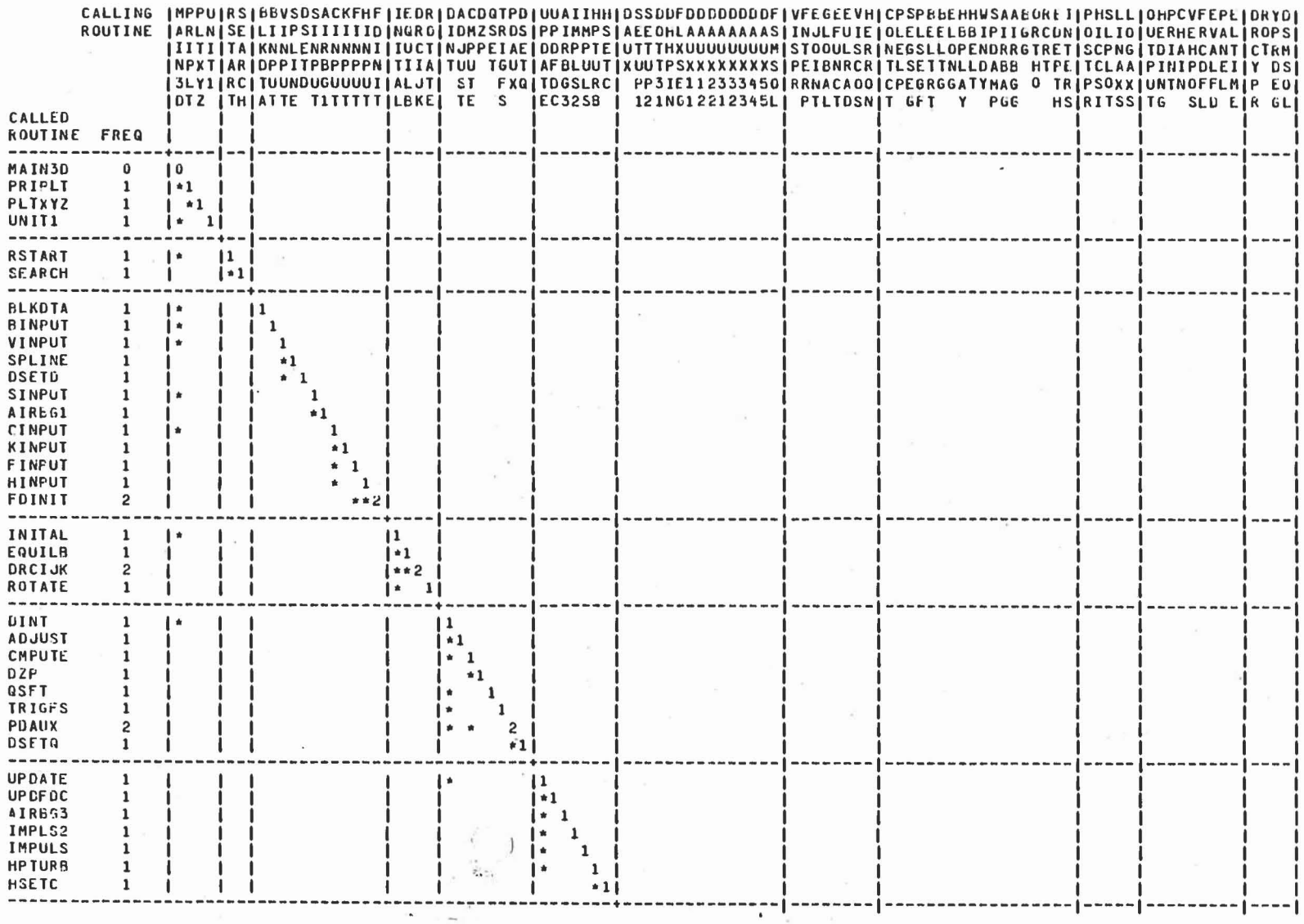
6.0 SUBROUTINE, COMMON BLOCK AND VARIABLE CROSS REFERENCE CHARTS

This section presents the output of a computer program written at Calspan to generate a cross reference chart showing the relationships and usage of the CVS IV program subprograms, FORTRAN library routines, CALCOMP plotting routines, labeled common blocks and all of the variables contained in each labeled common block. The input to the program was obtained from the information produced by the MAP procedure and from a data set produced by the INDEX program on the Univac 1108 Computer System at Edgewood, Maryland.

Pages 61 to 63 of the cross reference chart present a lower diagonal matrix that is symmetric in that the row identifications are identical to the columns. The subprograms are divided into logical blocks of the program flow. Those subprograms called by other subprograms of the CVS program are indicated by asterisks in the matrix.

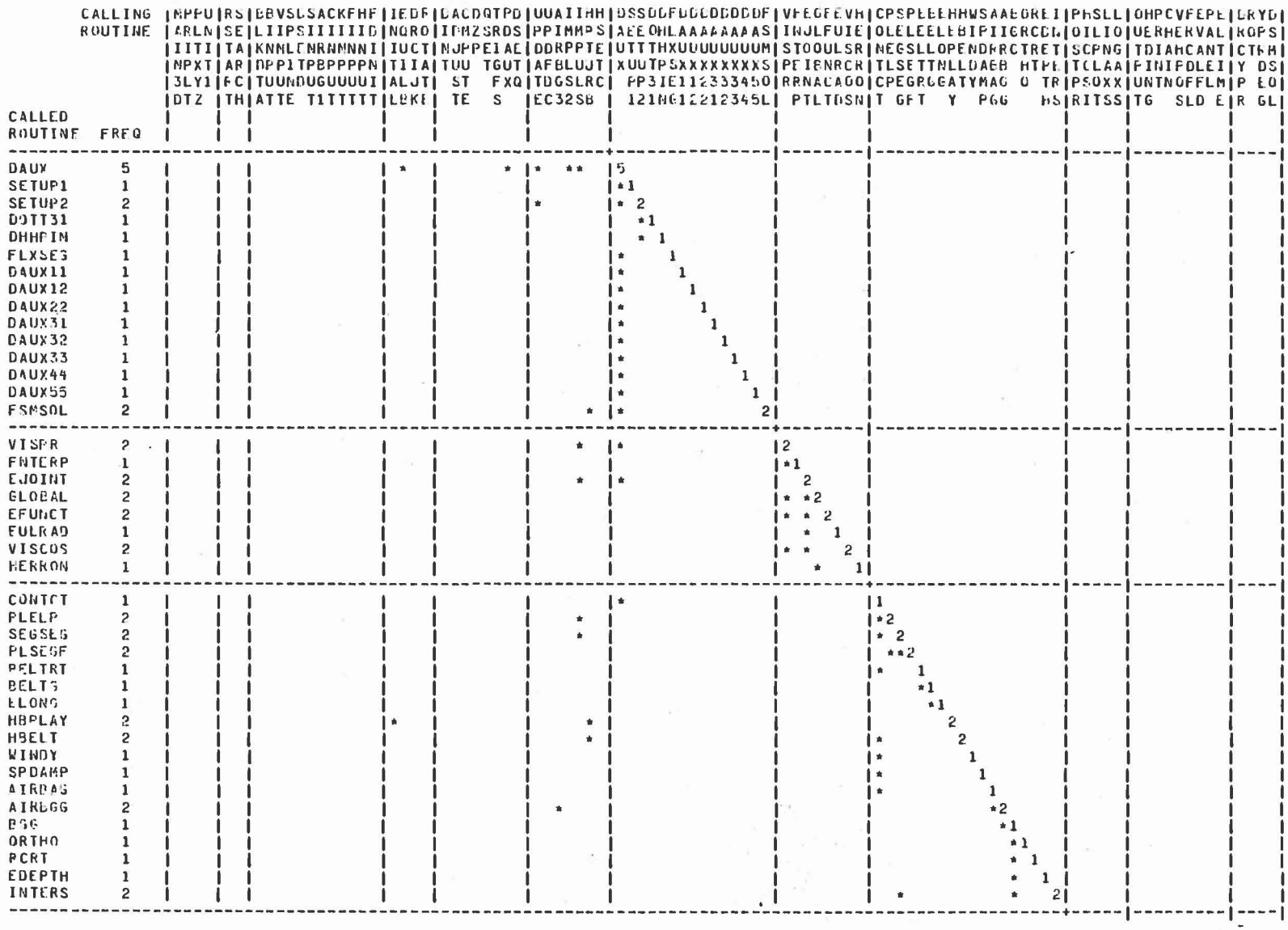
Page 64, using the same column identifications of pages 61 to 63, presents those labeled common blocks that are used by each subprogram.

In like manner, pages 65 to 73 show the usage of each variable in the labeled common blocks in every subprogram. The appearance of an asterisk indicates that the variable in the labeled common block is used by that subprogram, an X indicates that the variable is used as an internal variable in the subprogram not transmittable to the rest of the program through the labeled common block.



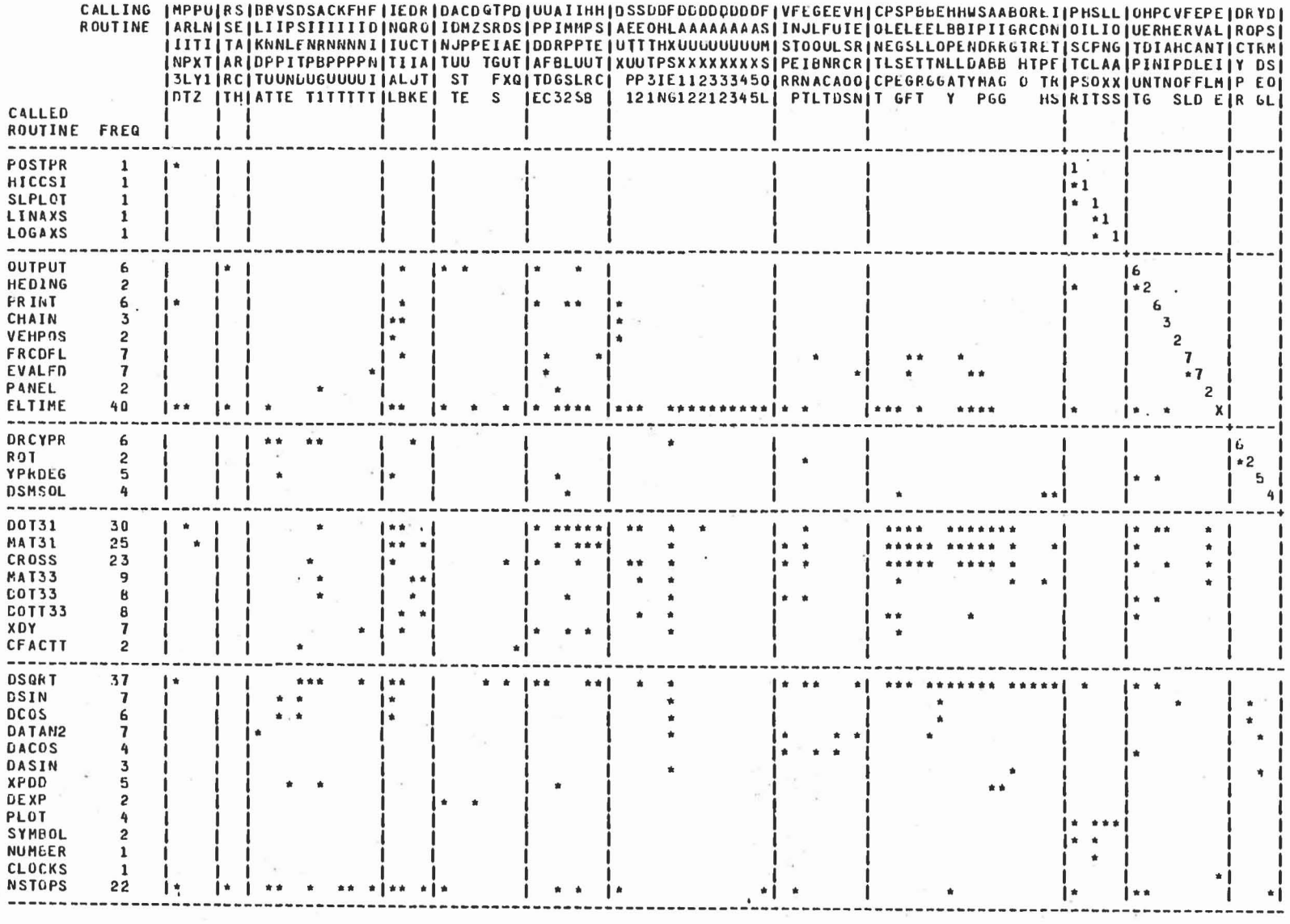
Subroutine Cross Reference Chart

CVS IV (Version 20)



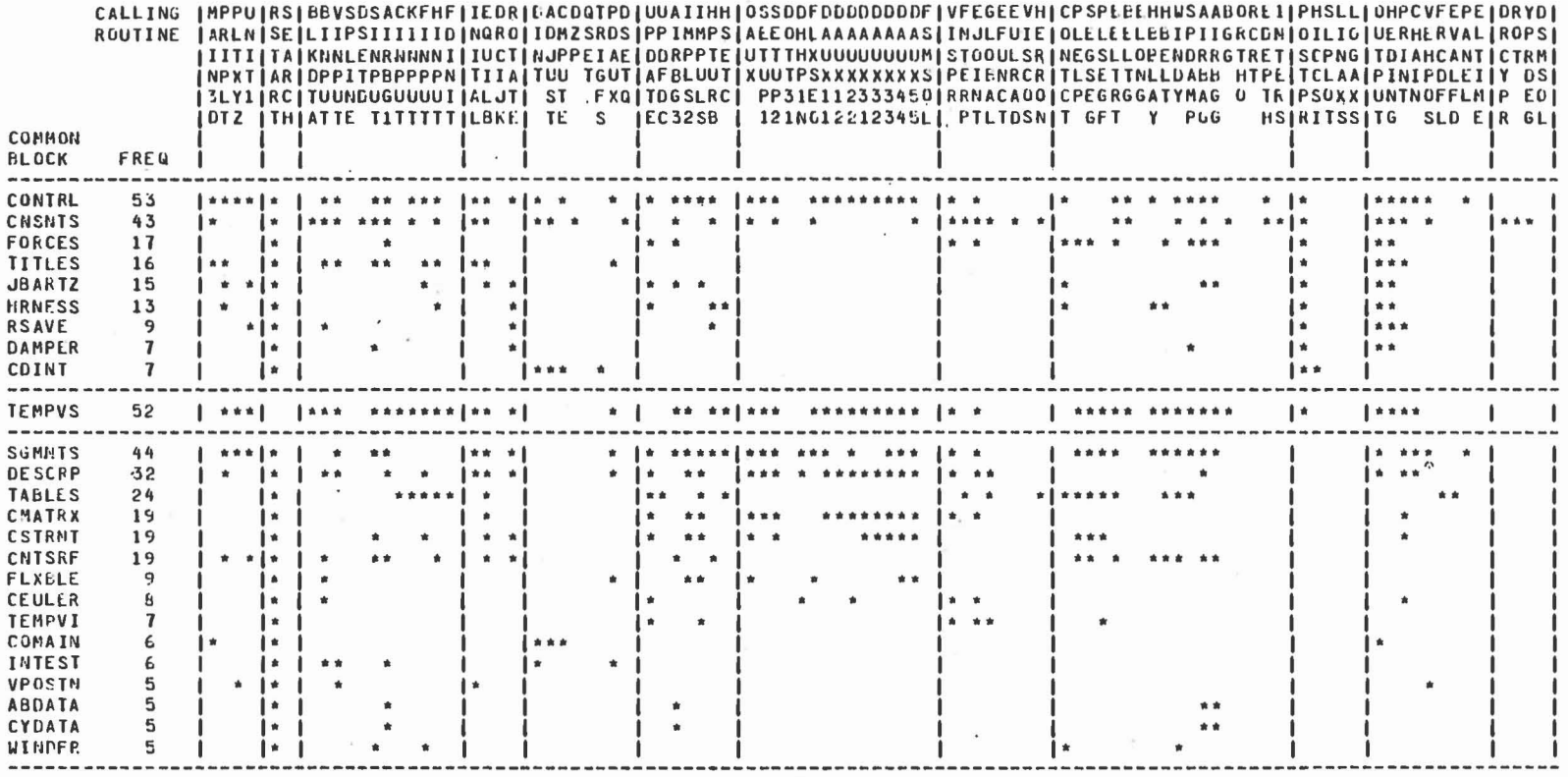
Subroutine Cross Reference Chart

CVS IV (Version 20)



Subroutine Cross Reference Chart

CVS IV (Version 20)

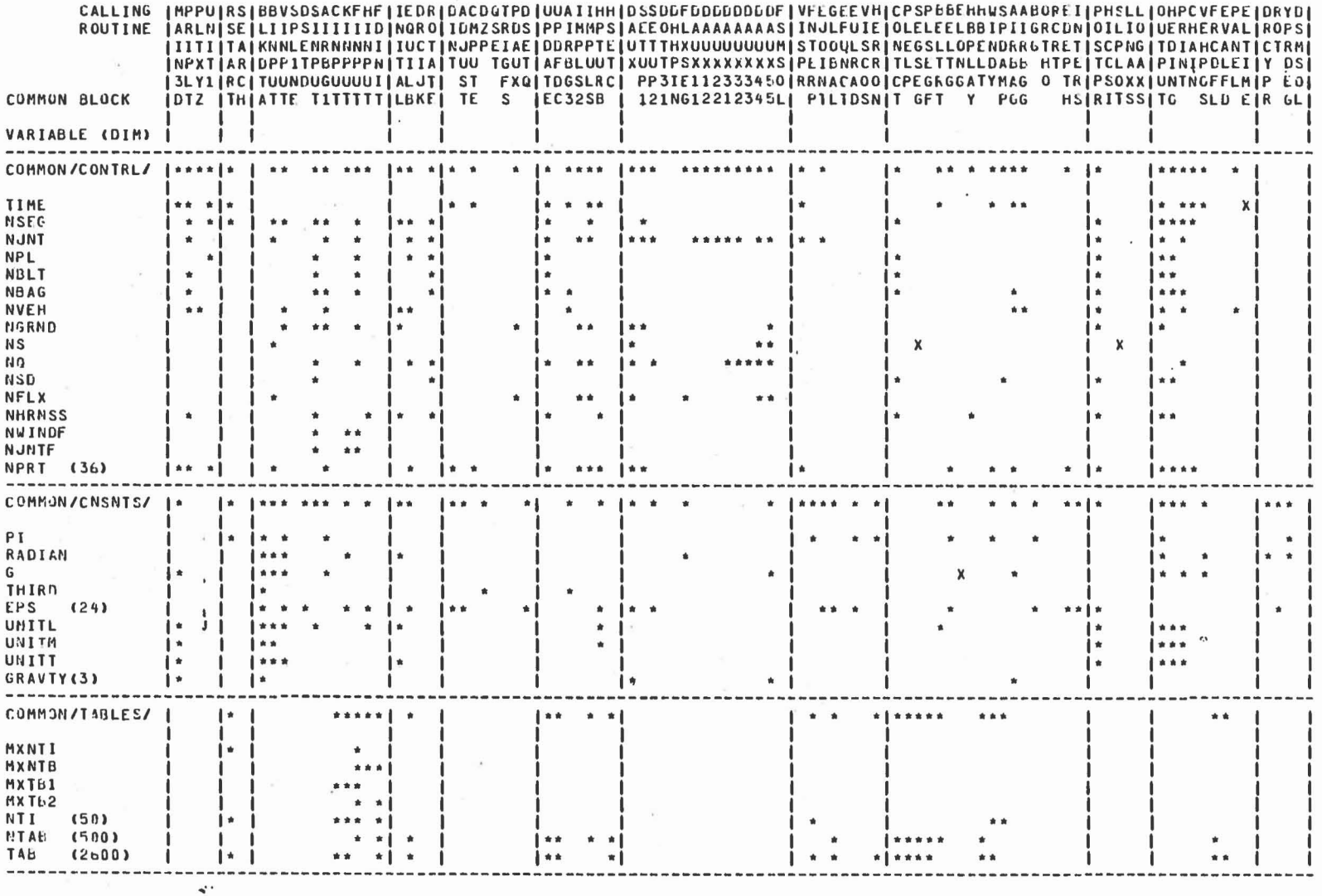


COMMON BLOCK	FREQ	CALLING ROUTINE	MPPU ARLN IITI NPXT 3LY1 DTZ	RS SE TA AR RC TH	BBVSDSACKFHF LIIPSIIIIIID KNNLENRNNNNI DPPITPBPPPPN TUUNDUGUUUUI ATTE T1TTTTT	IEDR NQRO IUCT TIIA ALJT LBKE	GACDQTPD IDMZSRDS NJPPEIAE TUU TGUT ST FXQ TE S	UUAIIHH PPIMMPS DDRPPTE AFBLUUT TDGSLRC EC32SB	OSSDDFDDDDDDDDF AEEOHLAAAAAAAAS UTTTHXUUUUUUUUM XUUTPSXXXXXXXXS PP31E112333450 121NG122123451	VFEGEEVH INJLFUIE STOOULSR PEIENRCR RRNACAOO PTLTDSN	CPSPLBEHHWSAABORE1 OLELEELEBIPIIGRCDN NEGSLLOPENDRRGTRET TLSETTNLLDABB HTPE CPEGRGGATYMAG O TR T GFT Y PUG HS	PHSLL OILIG SCPNG TCLAA PSOXX RITSS	OHPCVFEPE UERHERVAL TDIAHCANT PINIPDLEI UNTNOFFLM TG SLD E	DRYD ROPS CTRM Y DS P EO R GL
CONTRL	53	****	*	** ** ***	** *	* * *	* ****	*** *********	* *	* ** * **** *	*	***** *	
CNSNTS	43	*	*	*** *** * *	**	** * *	* * * *	* * *	**** * *	** * * * **	*	*** *	***
FORCES	17		*	*		* *			* *	*** * * ***	*	**	
TITLES	16	**	*	** ** **	**	*				**	*	***	
JBARTZ	15	* *	*	*	* *	* * *				* **	*	**	
HRNESS	13	*	*	*	*	**				* **	*	**	
RSAVE	9	*	*	*	*	*					*	***	
DAMPER	7		*	*	*					*	*	**	
CDINT	7		*			*** *					**		
TEMPVS	52	***		*** *******	** *	*	** **	*** *********	* *	***** *******	*	****	
SGMNTS	44	***	*	* **	** *	*	* *****	*** *** * ***	* *	**** ******		* *** *	
DESCRP	32	*	*	** * *	** *	*	* **	*** * ********	* **	*		* **	
TABLES	24		*	*****	*		** * *		* * *	***** ***		**	
CMATRX	19		*		*		* **	*** ********	* *			*	
CSTRNT	19		*	* *	* *	* *	* **	* * *****		***		*	
CNTSRF	19	* *	*	* ** *	* *		* *			** * *** **			
FLXBLE	9		*	*		*	**	* * **					
CEULER	8		*	*			*	* * *	* *			*	
TEMPVI	7		*				* *		* **	*			
COMAIN	6	*	*			***						*	
INTEST	6		*	** *		* *							
VPOSTN	5	*	*	*	*							*	
ABDATA	5		*	*			*			**			
CYDATA	5		*	*			*			**			
WINDFR	5		*	* *						* *			

64

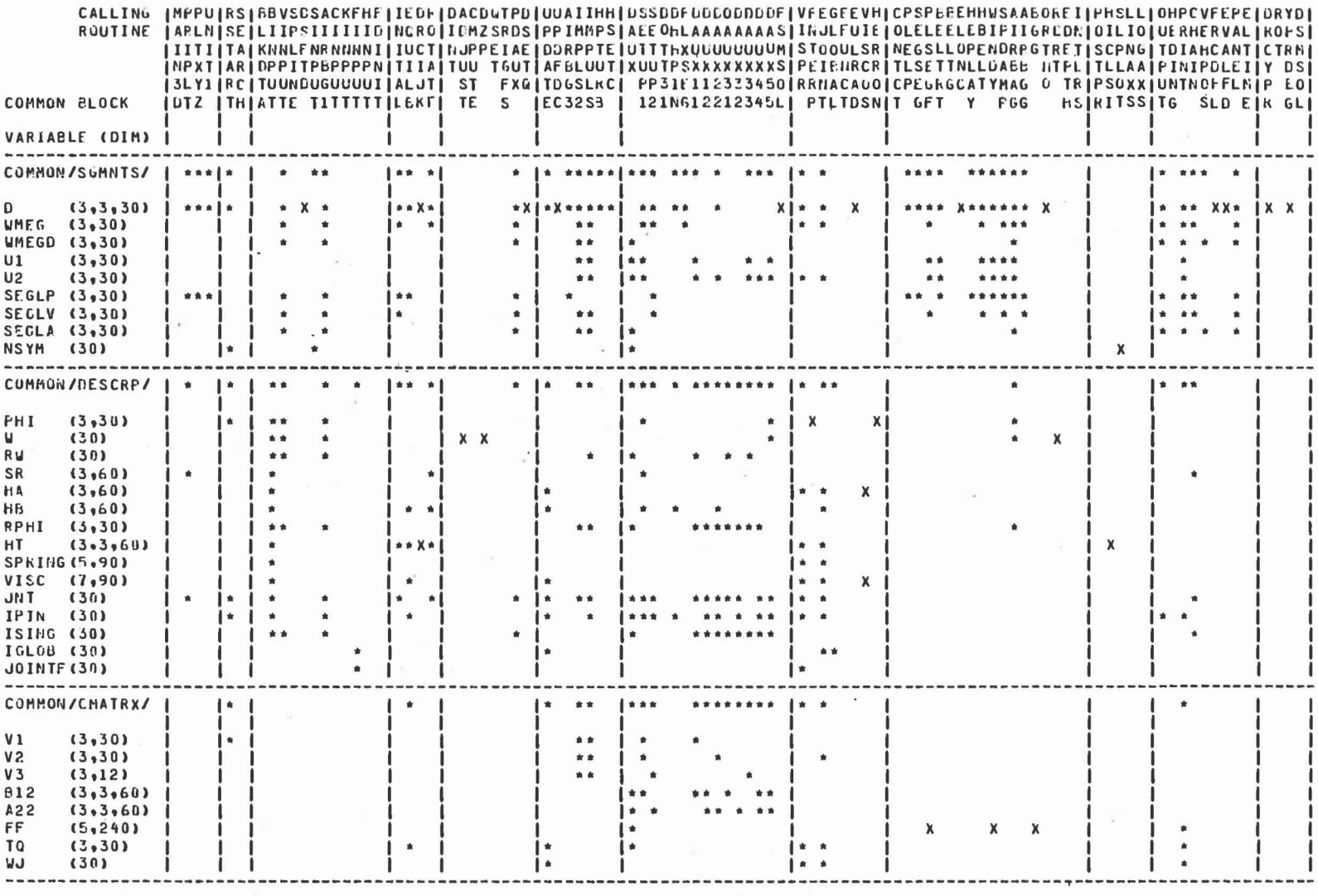
Common Block Cross Reference Chart

CVS IV (Version 20)



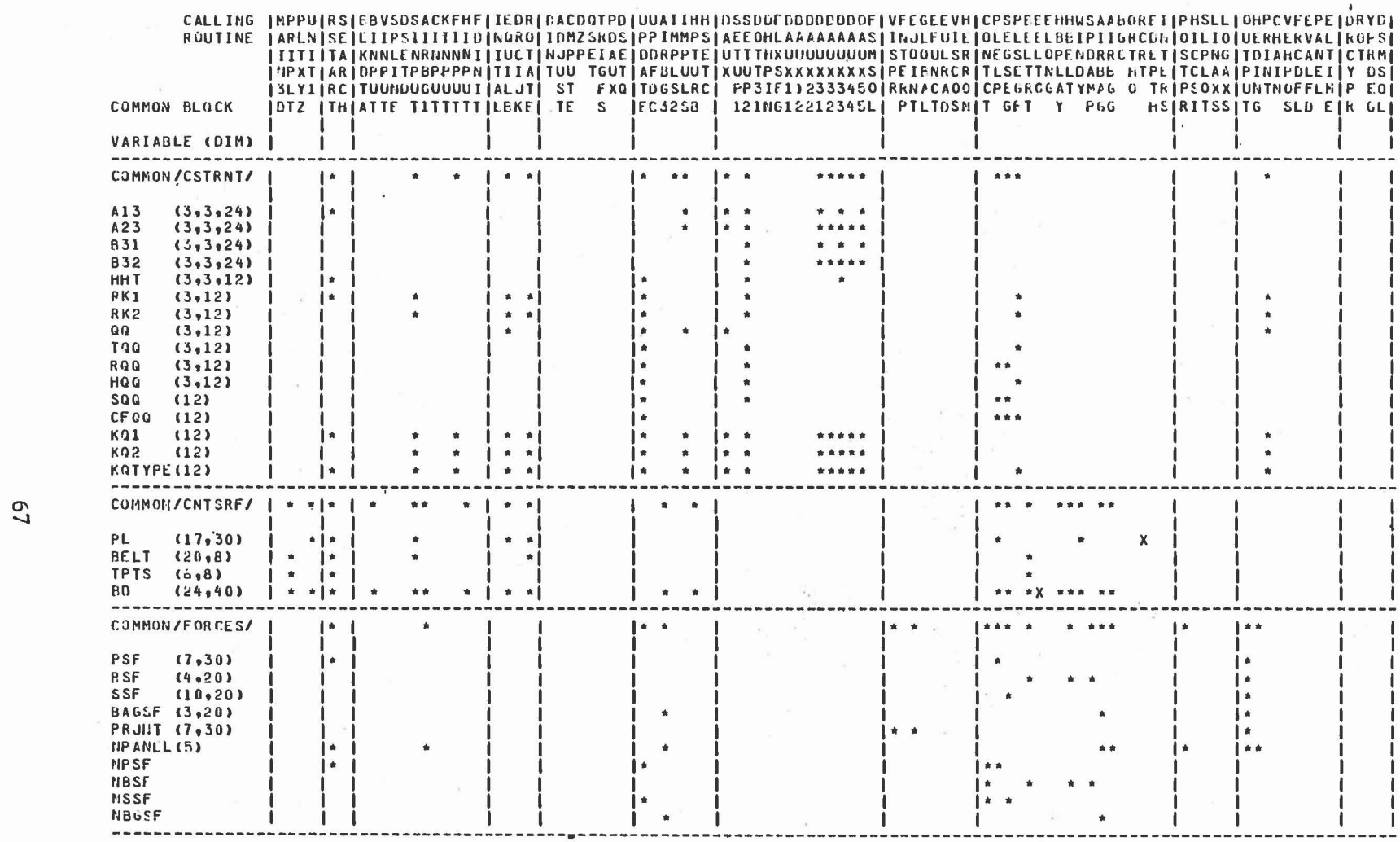
Variable Cross Reference Chart

CVS IV (Version 20)



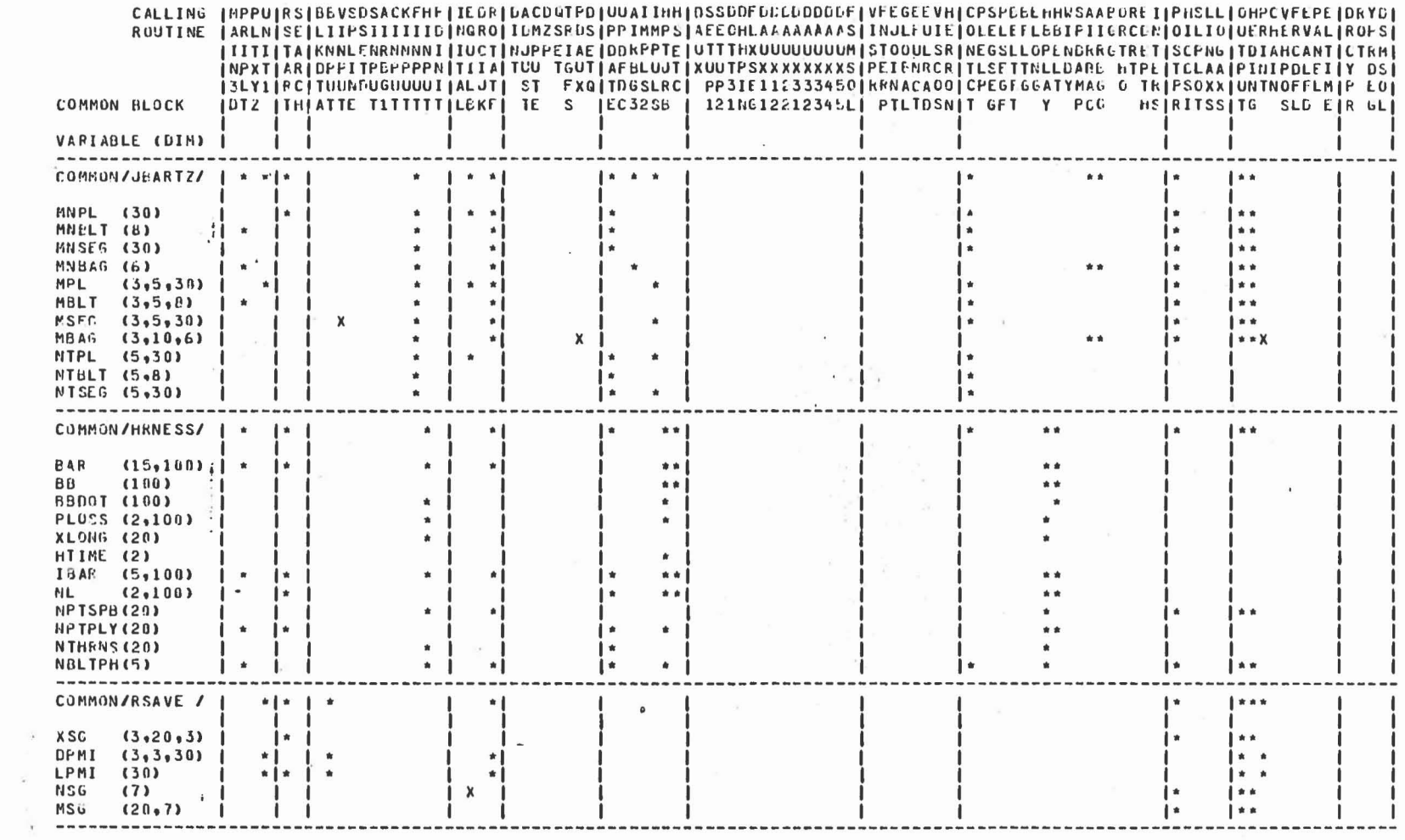
Variable Cross Reference Chart

CVS IV (Version 20)



Variable Cross Reference Chart

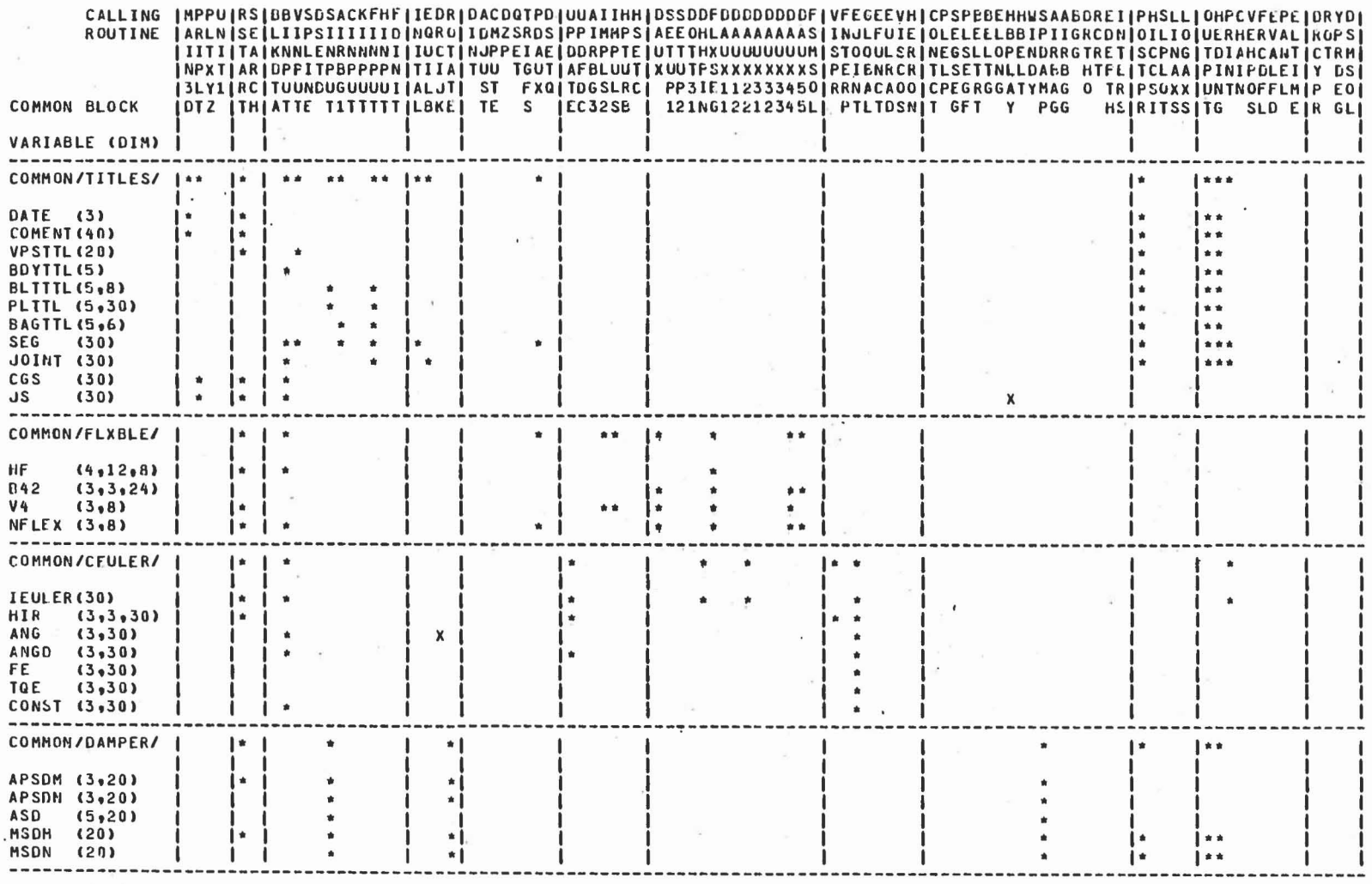
CVS IV (Version 20)



CALLING ROUTINE / COMMON BLOCK / VARIABLE (DIM)	MPPU ARLN IITI NPXT 3LY1 DTZ	RS SE TA AR PC TH	BBVSDSACKFHF LIIPSIIIIIIG KNNLENRNNNNI DPPITPBPPPPN TUUNDUGUUUUI ATTE T1TTTTT	IEGR NGRO IUCT TIIA ALJT LBKF	DACDGTPD IDMZSRDS NJPPEIAE TCU TGUT ST FXQ IE S	UUAIIHH PPIMMPS DDKPPTE AFBLUJT TDGSLRC EC32SB	DSSDDFDECDDDDDF AEECHLAAAAAAAAS UTTTHXUUUUUUUUM XUUTPSXXXXXXXXS PP3IE112333450 121NG12212345L	VFEGEEVH INJLFUIE STOOULSR PEIFNRCR KRNACAOO PTLTDSN	CPSPDBLHHWSAAPURE I OLELEFLBBIPIIGRCLN NEGSLLOPLNDRRGTRET I TLSFTTNLLDARB HTPL CPEGFGGATYMAG G TR T GFT Y PCG HS	PHSLL OILIO TCLAA PSOXX RITSS	OHPCVFLPE UERHERVAL TDIAHCANT PINIPDLEI UNTNOFFLM TG SLG E	DRYD ROFS CTRM Y DS P EO R GL
COMMON/JBARTZ/	* *	*	*	* *		* * *			* **	*	**	
MNPL (30)		*	*	* *		*			*	*	**	
MNBLT (8)	*		*	*		*			*	*	**	
MNSEG (30)			*	*		*			*	*	**	
MNBAG (6)	*		*	*		*			**	*	**	
MPL (3,5,30)	*		*	* *		*			*	*	**	
MBLT (3,5,8)	*		*	*		*			*	*	**	
MSEG (3,5,30)			X *	*		*			*	*	**	
MBAG (3,10,6)			*	*	X				**	*	**X	
NTPL (5,30)			*	*		* *			*			
NTBLT (5,8)			*			*			*			
NTSEG (5,30)			*			* *			*			
COMMON/HKNESS/	*	*	*	*		* **			* **	*	**	
BAR (15,100)	*	*	*	*		**			**			
BB (100)						**			**			
BBDOT (100)			*			*			*			
PLOSS (2,100)			*			*			*			
XLONG (20)			*						*			
HTIME (2)						*						
IBAR (5,100)	*	*	*	*		* **			**			
NL (2,100)	-	*		*		* **			**			
NPTSPB(20)			*	*					*	*	**	
NPTPLY(20)	*	*				* *			**			
NTHRNS(20)			*			*			*			
NBLTPH(5)	*		*	*		* *			* *	*	**	
COMMON/RSAVE /	*	*	*	*						*	***	
XSC (3,20,3)		*								*	**	
DPMI (3,3,30)	*		*	*							* *	
LPMI (30)	*	*	*	*							* *	
NSG (7)				X						*	**	
MSG (20,7)										*	**	

Variable Cross Reference Chart

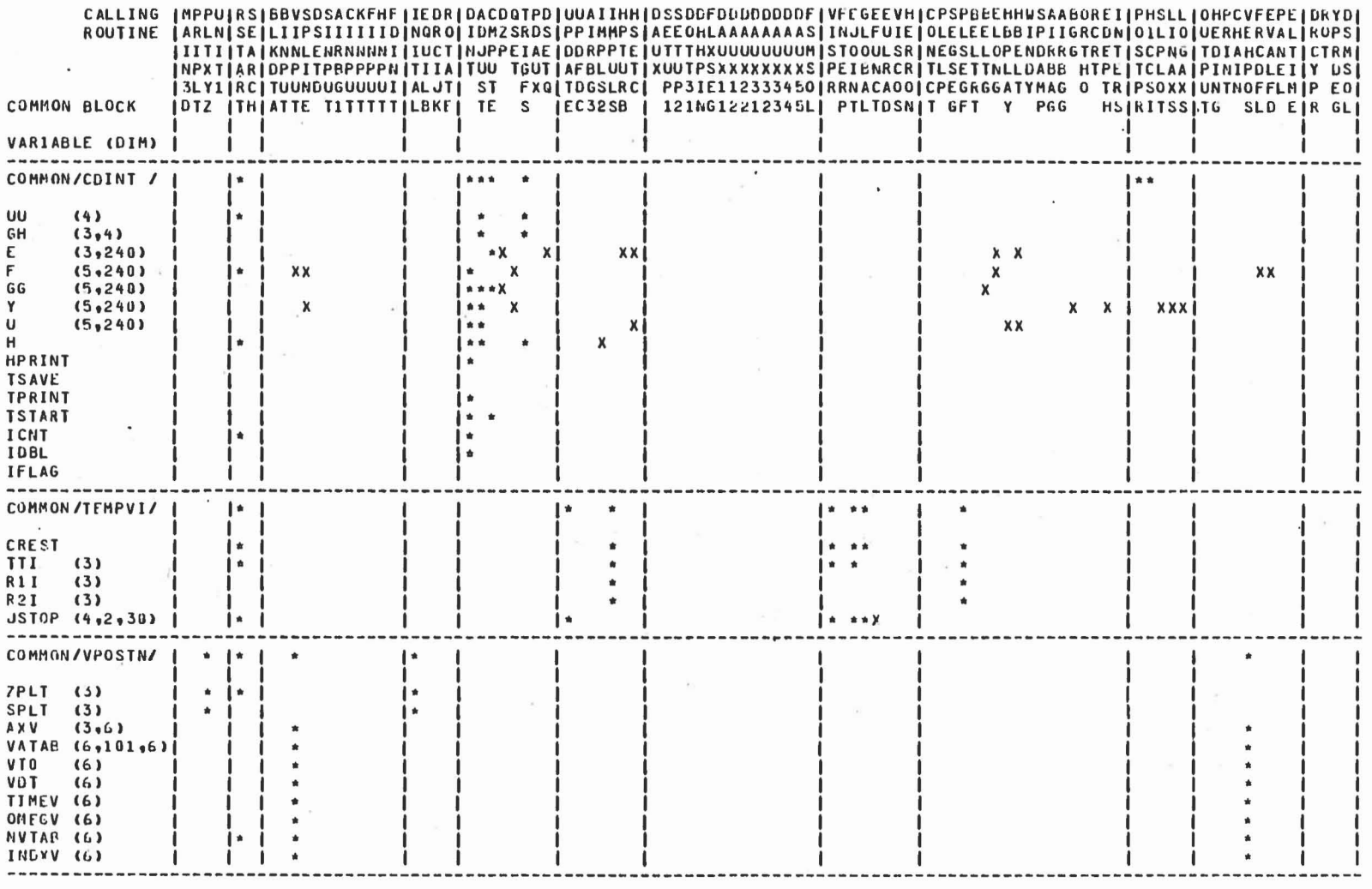
CVS IV (Version 20)



COMMON BLOCK / VARIABLE (DIM)	CALLING ROUTINE	MPPU ARLN IITI NPXT 3LY1 DTZ	RS SE TA AR RC TH	BBVSDSACKFHF LIIPSIIIIIID KNNLENRNNNNI DPFITPBPPPPN TUUNDUGUUUUI ATTE T1TTTTT	IEDR NQRO IUCT TIIA ALJT LBKE	DACDQTPD IDMZSRDS NJPPEIAE DDRPPTE TUU TGUT ST FXQ TE S	UUAIIHH PPIMHPS AEEOHLAAAAAAAAS UTTTHXUUUUUUUUM AFBLUUT TDGSLRC EC32SB	DSSDDFDDDDDDDDF AEEOHLAAAAAAAAS UTTTHXUUUUUUUUM XUUTFSXXXXXXXXS PP3IE112333345O 121NG12212345L	VFEGEEVH INJLFUIE STOOULSR PEIENRCR RRNACAOO PTLTDSN	CPSPEBEHHWSAABOREI OLELEELBBIPIIGRCDN NEGSLLOPENDRRGTRET TLSETTNLLDABB HTFL CPEGRGGATYMAG O TR T GFT Y PGG HS	PHSLL OILIO SCPNG TCLAA PSOXX RITSS	OHPCVFEPE UERHERVAL TDIAHCANT PINIPDLEI UNTNOFFLM TG SLD E	DRYD KOPS CTRM Y DS P EO R GL
COMMON/TITLES/		**	*	** ** **	**	*					*	***
DATE (3)		*	*								*	**
COMENT(40)		*	*								*	**
VPSTTL(20)			*	*							*	**
BDYTTL(5)				*							*	**
BLTTTL(5,8)				* *							*	**
PLTTL (5,30)				* *							*	**
BAGTTL(5,6)				* *							*	**
SEG (30)				** * *	*	*					*	***
JOINT (30)				* *	*						*	***
CGS (30)		*	*	*								
JS (30)		*	*	*						X		
COMMON/FLXBLE/			*	*		*	**	* * **				
HF (4,12,8)			*	*				*				
D42 (3,3,24)								* * **				
V4 (3,8)			*				**	* * *				
NFLEX (3,8)			*	*		*		* * **				
COMMON/CFULER/			*	*			*	* *	* *			*
IEULER(30)			*	*			*	* *	*			*
HIR (3,3,30)			*				*		* *			
ANG (3,30)				*	X				*			
ANGD (3,30)				*			*		*			
FE (3,30)									*			
TQE (3,30)									*			
CONST (3,30)				*					*			
COMMON/DAMPER/			*	*	*			*	*			
APSDM (3,20)			*	*	*			*				
APSDN (3,20)				*	*			*				
ASD (5,20)				*				*				
MSDM (20)			*	*	*			*	*			
MSDN (20)				*	*			*	*			

Variable Cross Reference Chart

CVS IV (Version 20)



CALLING ROUTINE / COMMON BLOCK / VARIABLE (DIM)	MPPU ARLN IITI NPXT 3LY1 DTZ	RS SE TA AR RC TH	BBVSDSACKFHF LIIPSIIIIIID TUUNDUGUUUUI ATTE T1TTTTT	IEDR NQRO IUCT TIIA ALJT LBKF	DACDQTPD IDMZSRDS NJPPEIAE TUU TGUT ST FXQ TE S	UUAIIHH PPIMMPS DDRPPTE AFBLUUT TDGSLRC EC32SB	DSSDDFDDDDDDDDF AEEOHLAAAAAAAAS UTTTHXUUUUUUUUM XUUTPSXXXXXXXXS 121NG12212345L	VFEGEEVH INJLFUIE STOOULSR PEIENRCR RRNACAOO PTLTDSN	CPSPBEEHHWSAABOREI OLELEELBBIPIIGRCDN NEGSLLOPENDRRGTRET TLSETTNLLDABB HTPE CPEGRGGATYMAG O TR T GFT Y PGG HS	PHSLL O1LIO SCPNG TCLAA PSOXX RITSS	OHPCVFEPE UERHERVAL TDIAHCANT PINIPDLEI UNTNOFFLN TG SLD E	DKYD KUPS CTRM Y US P EO R GL
COMMON/CDINT /		*			*** *					**		
UU (4)		*			* *							
GH (3,4)					* *							
E (3,240)					*X X	XX			X X			
F (5,240)		*	XX		* X				X		XX	
GG (5,240)					***X				X			
Y (5,240)			X		** X				X X	XXX		
U (5,240)					**	X			XX			
H		*			** *	X						
HPRINT					*							
TSAVE												
TPRINT					*							
TSTART					* *							
ICNT		*			*							
IDBL					*							
IFLAG												
COMMON/TEMPVI/		*				* *		* **	*			
CREST		*				*		* **	*			
TTI (3)		*				*		* *	*			
R1I (3)						*			*			
R2I (3)						*			*			
JSTOP (4,2,30)		*				*		* **X				
COMMON/VPOSTN/	*	*	*	*							*	
ZPLT (3)	*	*			*							
SPLT (3)	*				*							
AXV (3,6)			*								*	
VATAB (6,101,6)			*								*	
VTO (6)			*								*	
VDT (6)			*								*	
TIMEV (6)			*								*	
OMEGV (6)			*								*	
NVTAB (6)		*	*								*	
INDXV (6)			*								*	

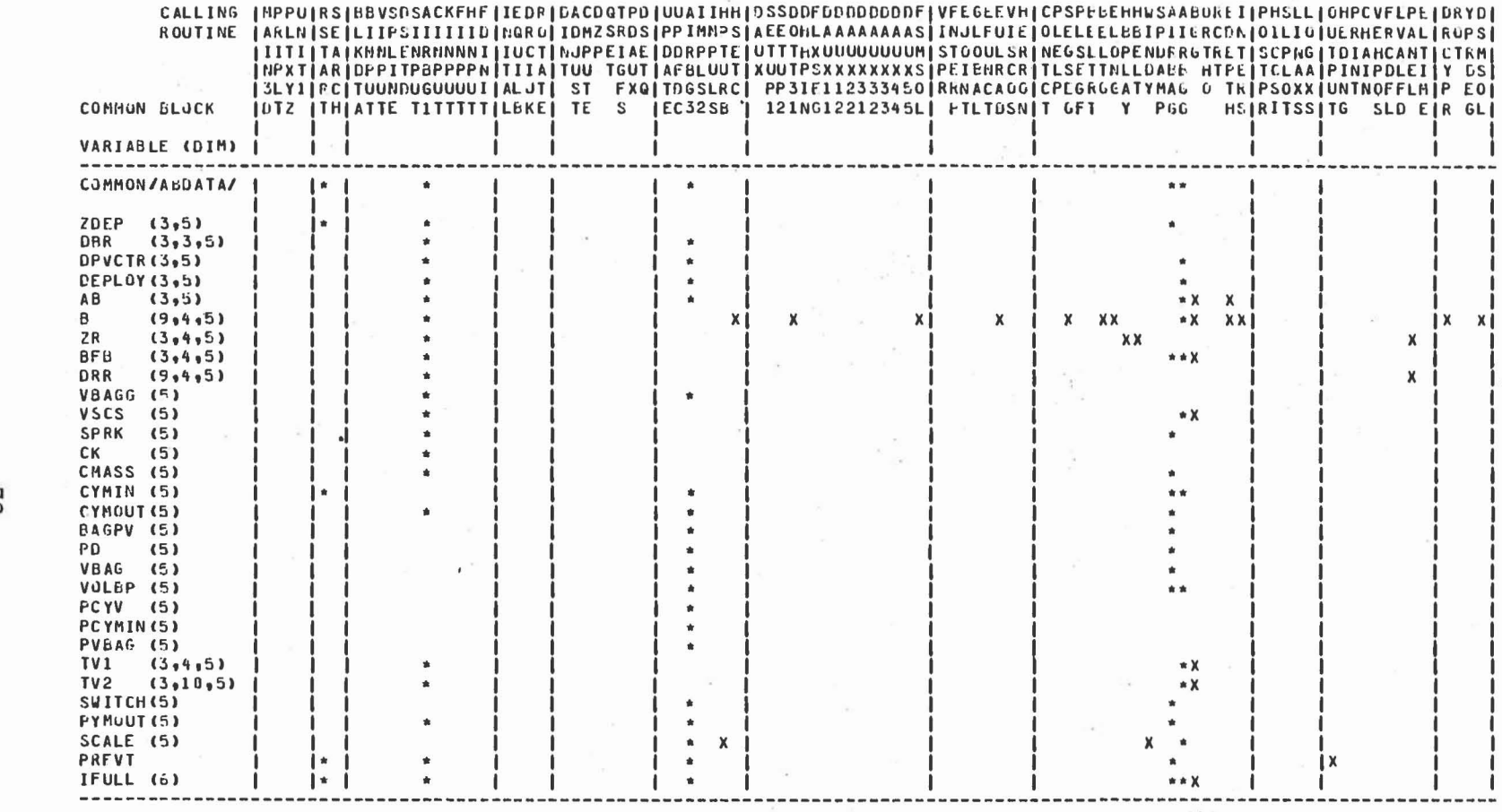
Variable Cross Reference Chart

CVS IV (Version 20)

CALLING ROUTINE / COMMON BLOCK / VARIABLE (DIM)	MPPU ARLN IITI NPXT 3LY1 DTZ	RS SE TA AR RC TH	BEVSDSACKFHF LIIPSIIIIIID KNNLENRNNNNI DPPITPEPPPPN TUUNDUGUUUUI ATTE T1TTTTT	ILDR NQRO IUCT TIIA ALJT LBKE	DACDQTPD IDMZSRDS MJPPEIAE TUU TGUT ST FXQ TE S	UUAIIHH PPIMMPS DDRPPTE AFBLUUT TDGSLRC EC32SB	DSSDDFDDDDDDDDF ALEOHLAAAAAAAAS UTTTHXUUUUUUUUM XUUTPSXXXXXXXXS 121NG12212345L	VFEGLEVH INJLFUIE STOOULSR PEIBNRCR RRNACAQQ PTLTDSN	CPSPEBEHHWSAAEOREI OLELLELBBIPIIGRCDN NEGSLLOPENDRRGTRET TLSETTNLLDABB HTPE CPEGRGGATYMAG O TR T GFT Y PGG HS	PHSLL OILIO SCPNG TCLAA PSOXX RITSS	OHPCVFLPE UERHERVAL TDIAHCANT PINIPDLEI UNTNOFFLM TG SLD E	DRYD ROPS CTRM Y DS P ED R GL
COMMON/COMAIN/	*	*			***						*	
VAR (240)		*			*** X							
DER (240)		*			*** X X							
DT	*	*			*					X		
H0	*	*			*							
HMAX	*	*			*							
HMIN	*	*			*							
RSTIME	*											
ISTEP	*X	*			*							
NSTEPS	*	*										
NDINT	*	*			*							
NEQ		*			*** X							
IRSIN	*										*	
IRSOUT	*											
COMMON/INTEST/		*	** *		* *							
SGTEST(3,4,30)		*	** *		*							
XTEST (3,120)		*			* *							
SEGT (120)		*			* *							
REGT (120)		*			* *							
COMMON/WINDFR/		*	* *						* *			
WTIME (30)		*	*						*			
QFU (3,5)			*						*			
QFV (3,5)			*						*			
IWIND (30)		*	*						*			
MWSEG (5,30)			*						*			
NFVSEG(6)			*						* *			
NFVNT (5)			*						*			

Variable Cross Reference Chart

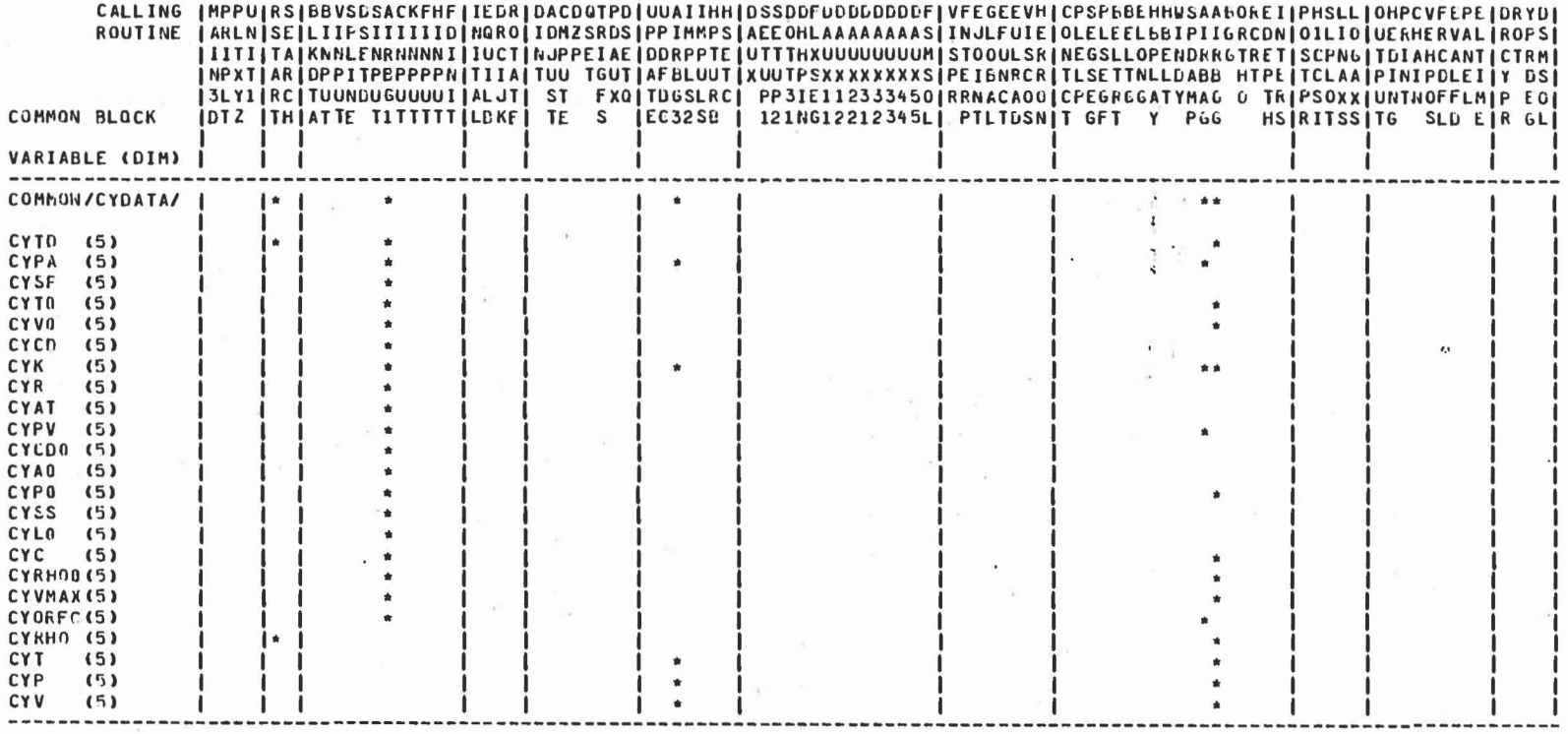
CVS IV (Version 20)



COMMON BLOCK / VARIABLE (DIM)	CALLING ROUTINE	MPPU ARLN IIII NPXT 3LY1 DTZ	RS SE TA AR FC TH	BBVSDSACKFHF LIIPSIIIIIID KMNLENRNNNNI DPPITPBPPPPN TUUNDUGUUUUI ATTE T1TTTTT	IEDR NQRU IUCT TIIA ALJT LBKE	DACDQTPD IDMZSRDS MJPPEIAE TUU TGUT ST FXQ TE S	UUAIIHH PPIMNPS DDRPPTE AFBLUUT TDGSLRC EC32SB	DSSDDFDDDDDDDDF AEEOHLAAAAAAAAS UTTTHXUUUUUUUUM XUUTPSXXXXXXXXS PP3IF112333450 121NG12212345L	VFEGEEVH INJLFUIE STOOULSR PEIENRCR RKNACAOG PTLTDSN	CPSPEEEHHWSAABUKEI OLELEELBBIPIIERCDN TLSETTNLLDABE HTPE CPEGRGEATYMAG O TK T GFT Y PGG HS	PHSLL OILIO SCPWG TCLAA PSOXX RITSS	OHPCVFLPE UERHERVAL TDIAHCANT PINIPDLEI UNTNOFFLH TG SLD E	DRYD RUPS CTRM Y DS P EO R GL
COMMON/ABDATA/			*	*			*			**			
ZDEP (3,5)			*	*						*			
DBR (3,3,5)				*			*						
DPVCTR(3,5)				*			*			*			
DEPLOY(3,5)				*			*			*			
AB (3,5)				*			*			*X X			
B (9,4,5)				*			X	X X	X	X	X XX		X X
ZR (3,4,5)				*						XX	*X XX	X	
BFB (3,4,5)				*							**X		
DRR (9,4,5)				*								X	
VBAGG (5)				*			*						
VSCS (5)				*							*X		
SPRK (5)			*	*							*		
CK (5)				*									
CMASS (5)				*							*		
CYMIN (5)			*				*				**		
CYMOUT(5)				*			*				*		
BAGPV (5)							*				*		
PD (5)							*				*		
VBAG (5)							*				*		
VOLBP (5)							*				**		
PCYV (5)							*						
PCYMIN(5)							*						
PVBAG (5)							*						
TV1 (3,4,5)				*							*X		
TV2 (3,10,5)				*							*X		
SWITCH(5)							*				*		
PYMOUT(5)				*			*				*		
SCALE (5)							* X			X	*		
PRFVT			*	*			*				*	X	
IFULL (6)			*	*			*				**X		

Variable Cross Reference Chart

CVS IV (Version 20)



CALLING ROUTINE / COMMON BLOCK / VARIABLE (DIM)	MPPU ARLN IITI NPXT 3LY1 DTZ	RS SE TA AR RC TH	BBVSDSACKFHF LIIPSIIIIIID KNNLENRNNNNI TUUNDUGUUUUI ATTE T1TTTTT	IEDR NQRO IUCT TIIA ALJT LDKF	DACDQTPD IDMZSRDS NJPPEIAE TUU TGUT ST FXQ TE S	UUAIIHH PPIMMPS DDRPPTE AFBLUUT TDGSLRC EC32SB	DSSDDFUDDDDDDDF AEEOHLAAAAAAAAS UTTTHXUUUUUUUUM XUUTPSXXXXXXXXS PP3IE112333450 121NG12212345L	VFEGEEVH INJLFUIE STOOULSR PEIBNRCR RRNACAOO PTLTDSN	CPSPBBEHHWSAABORE I OLELEELBBIPIIGRCDN NEGSLLOPENDRRGTRET TLSETTNLLDABB HTPE CPEGRGGATYMAG O TR T GFT Y PGG HS	PHSLL O1LIO SCPNG TCLAA PSOXX RITSS	OHPCVFEPE UERHERVAL TDIAHCANT PINIPDLEI UNTNOFFLM TG SLD E	DRYD ROPS CTRM Y DS P EQ R GL
COMMON/CYDATA/		*	*			*			**			
CYTD (5)		*	*						*			
CYPA (5)			*			*			*			
CYSF (5)			*									
CYTO (5)			*						*			
CYVO (5)			*						*			
CYCD (5)			*									
CYK (5)			*			*			**			
CYR (5)			*									
CYAT (5)			*									
CYPV (5)			*						*			
CYCDO (5)			*									
CYAO (5)			*									
CYPO (5)			*						*			
CYSS (5)			*									
CYLO (5)			*									
CYC (5)			*						*			
CYRHOD (5)			*						*			
CYVMAX (5)			*						*			
CYORFC (5)			*						*			
CYRHO (5)		*							*			
CYT (5)						*			*			
CYP (5)						*			*			
CYV (5)						*			*			

Variable Cross Reference Chart

CVS IV (Version 20)

7.0 CVS PROGRAM SUBROUTINE DESCRIPTIONS

LIST OF 107 SUBPROGRAMS OF CVS COMPUTER PROGRAM

No.	Name	Rev.	Date	No.	Name	Rev.	Date	No.	Name	Rev.	Date
1.	COMMON	20	05/18/80	37.	DSETQ	19	08/05/78	73.	ORTHO	03	05/31/73
2.	MAIN3D	20	04/11/80	38.	DSMSOL	03	07/08/74	74.	OUTPUT	20	05/18/80
3.	ADJUST	19	09/18/79	39.	DZP	19	08/05/78	75.	PANEL	19	08/05/78
4.	AIRBAG	20	04/11/80	40.	EDEPTH	19	08/05/78	76.	PDAUX	20	12/18/79
5.	AIRBGG	20	04/11/80	41.	EFUNCT	20	04/29/80	77.	PLELP	20	04/11/80
6.	AIRBG1	20	04/11/80	42.	EJOINT	20	04/11/80	78.	PLSEGF	19	10/19/79
7.	AIRBG3	20	04/11/80	43.	ELONG	01	10/05/72	79.	PLTXYZ	20	05/06/80
8.	BELTG	19	08/05/78	44.	ELTIME	19	09/18/79	80.	POSTPR	20	05/18/80
9.	BELTRT	20	05/23/80	45.	EQUILB	19	10/19/79	81.	PRINT	20	04/11/80
10.	BGG	19	08/05/78	46.	EULRAD	20	05/02/80	82.	PRIPLT	20	05/05/80
11.	BINPUT	20	05/06/80	47.	EVALFD	20	04/30/80	83.	QSET	16	03/24/76
12.	BLKDTA	19	08/05/78	48.	FDINIT	20	05/14/80	84.	RCRT	03	07/19/73
13.	CFACTT	03	05/31/73	49.	FINPUT	20	05/27/80	85.	ROTATE	20	04/23/80
14.	CHAIN	19	09/05/78	50.	FLXSEG	19	08/05/78	86.	ROT	19	08/05/78
15.	CINPUT	19	08/05/78	51.	FNTERP	19	08/05/78	87.	RSTART	20	05/23/80
16.	CMPUTE	19	09/18/79	52.	FRCDFL	19	10/19/79	88.	SEARCH	20	05/23/80
17.	CONTCT	20	05/18/80	53.	FSMSOL	20	04/11/80	89.	SEGSEG	20	04/11/80
18.	CROSS	03	05/31/73	54.	GLOBAL	19	10/19/79	90.	SETUP1	20	04/26/80
19.	DAUX	20	05/18/80	55.	HBELT	20	05/18/80	91.	SETUP2	19	08/05/78
20.	DAUX11	19	09/05/78	56.	HBPLAY	20	06/11/80	92.	SINPUT	20	05/09/80
21.	DAUX12	19	09/05/78	57.	HEDING	20	05/18/80	93.	SLPLOT	20	11/30/79
22.	DAUX22	19	09/05/78	58.	HERRON	19	08/05/78	94.	SPDAMP	20	05/19/80
23.	DAUX31	19	09/05/78	59.	HICCSI	18	07/26/78	95.	SPLINE	19	05/14/79
24.	DAUX32	19	09/05/78	60.	HINPUT	19	10/23/79	96.	SPRNGF	19	08/05/78
25.	DAUX33	19	09/05/78	61.	HPTURB	20	04/11/80	97.	TRIGFS	19	08/05/78
26.	DAUX44	19	09/05/78	62.	HSETC	20	06/18/80	98.	UNIT1	20	01/22/80
27.	DAUX55	19	09/05/78	63.	IMPLS2	19	09/05/78	99.	UPDATE	20	04/11/80
28.	DHHPIN	19	08/05/78	64.	IMPULS	19	09/05/78	100.	UPDFDC	19	10/19/79
29.	DINT	19	09/18/79	65.	INITAL	20	01/22/80	101.	VEHPOS	19	09/15/78
30.	DOTT31	17	12/20/76	66.	INTERS	19	08/05/78	102.	VINPUT	20	05/07/80
31.	DOTT33	17	01/03/77	67.	KINPUT	19	09/18/79	103.	VISCOS	19	10/23/78
32.	DOT31	17	01/03/77	68.	LINAXS	18	02/28/78	104.	VISPR	20	04/11/80
33.	DOT33	17	01/03/77	69.	LOGAXS	19	09/18/79	105.	WINDY	20	05/09/80
34.	DRCIJK	18	02/24/78	70.	LTIME	01	11/29/72	106.	XDY	07	01/31/74
35.	DRCYPR	19	08/05/78	71.	MAT31	17	01/03/77	107.	YPRDEG	19	08/05/78
36.	DSETD	19	08/05/78	72.	MAT33	17	01/03/77				

1. BLOCK DATA COMMON

a. Purpose:

Block data subprogram required as first unit of CVS program computer load module to allocate the maximum storage for each of the labelled common blocks and to establish their order as required by Subroutine POSTPR.

b. Subroutines required:

None.

c. Labelled common blocks used:

(1) Required by Subroutine POSTPR and subordinate subroutines:
CONTRL, CNSNTS, JBARTZ, TITLES, FORCES, RSAVE, DAMPER, HRNESS.

(2) Overwritten by Subroutine POSTPR for Subroutine HICCSI:
CDINT.

(3) Overwritten by Subroutine POSTPR for tabular time history data:
TEMPVS, SGMNTS, DESCRP, CNTSRF, TABLES, VPOSTN, CMARTX, CEULER, FLXBLE, CSTRNT, TEMPVI, INTEST, COMAIN, ABDATA, CYDATA, WINDFR.

d. Input or argument parameters:

None.

e. Optional output:

None.

2. MAIN PROGRAM - CVS

a. Purpose:

Main program for the Calspan Three Dimensional Crash Victim Simulation (CVS) program. Controls the program input and initialization, calls the program integrator returning at specified time intervals for optional output, and calls the post-processing routine at the end.

b. Subroutines required:

BINPUT, BLKDTA, CINPUT, DINT, ELTIME, INITAL, POSTPR, PRINT, PRIPLT, RSTART, SINPUT, UNIT1, VINPUT.

c. Labelled common blocks used:

CONTRL, TITLES, CNSNTS, COMAIN.

d. Input or argument parameters:

Input Cards A.1, A.3, A.4 and A.5.

e. Optional output:

(1) NPRT(5), (6) or (7) $\neq$ 0: Subroutine PRIPLT
(2) IRSOUT $\neq$ 0: Subroutine RSTART
(3) NPRT(3) $\neq$ 0: Subroutine PRINT
(4) NPRT(1) $\neq$ 0: Subroutine UNIT1
(5) NPRT(2) $\neq$ 0: Subroutine ELTIME

f. Procedure:

1. Call Subroutines ELTIME as initial call and BLKDTA to initialize variables in COMMON /CNSNTS/.

2. Read input Cards A.1, and, if IRSIN ≠ 0, then read initial record from the restart input unit and go to step 5.

3. Read input Cards A.3, A.4 and A.5, and, if NPRT(4) < 0, then go to step 7.

4. Call the input and initialization Subroutines BINPUT, VINPUT, SINPUT, CINPUT and INITIAL.

5. If IRSOUT ≠ 0, then call Subroutine RSTSRT to write initial record on the restart output unit.

6. Integration loop - advance time by either integration via Subroutine DINT or by reading a time point record from the restart input unit. At the end of each time step, test for each of the optional output listed above.

7. If NPRT(4) ≠ 0 or 4, then call the post-processing Subroutine POSTPR.

3. SUBROUTINE ADJUST (M,D1)

a. Purpose:

Called by Subroutine DINT to update the values of the GG, U and Y arrays for the step number (M) from DINT.

b. Subroutines required:

None.

c. Labelled common blocks used:

CNSNTS, CDINT, COMAIN.

d. Input or argument parameters:

M: Subroutine DINT step number (1 to 5).
D1: Half step size (not currently used by routine)

e. Optional output:

None.

4. SUBROUTINE AIRBAG

a. Purpose:

This routine is called by Subroutine CONTCT to control the calls to Subroutine AIRBGG which computes the interaction of the bag with the reaction panels and the body segments. It then computes the differential pressure of the bag and adds the forces and torques on the segments to the U1 and U2 vectors. It also evaluates the linear and angular accelerations of the bag, whose motion is integrated by the program integrator.

b. Subroutines required:

AIRBGG, ELTIME.

c. Labelled common blocks used:

CONTRL, JBARTZ, DESCRP, FORCES, ABDATA, CNSNTS, CNTSRF, SGMNTS, TEMPVS.

d. Input or argument parameters:

None.

e. Optional output:

NPRT(21) $\neq$ 0 will cause the routine to print the forces, torques, volumes and pressure which have been computed.

Airbag contact forces are stored in the BAGSF array for output to the tabular time histories.

f. Procedure:

Subroutine AIRBGG is called for each bag. If a bag is not fully inflated the next bag is considered. If a bag is fully inflated the differential pressure is evaluated and the forces and torques on the segments and bags are added to the U1 and U2 vectors, respectively. If the airbag is not intersecting the primary reaction panel, the forces and torques of a linear spring function are computed to tie the endpoint of the +x semi axis of the airbag with the deployment point on the reaction panel. Finally, the linear and angular accelerations of each airbag are computed from the total forces and torques acting on the airbag whose motion is then computed by the program integrator.

5. SUBROUTINE AIRBGG (J)

a. Purpose:

Called by Subroutines AIRBAG and AIRBG3 to compute the volume of intersection between an airbag ellipsoid and other ellipsoids representing vehicle panels or body segments.

b. Subroutines required:

BGG, DOT31.

c. Labelled common blocks used:

CONTRL, SGMNTS, JBARTZ, FORCES, CNTSRF, ABDATA, CYDATA, TEMPVS.

d. Input or argument parameters:

J: The airbag identification number.

e. Optional output:

None.

f. Procedure:

1. Compute thermodynamic properties of the gas supply which inflates the air bag

$$Q = 1 + C(t-t_d)$$

$$\rho = \rho_0 Q^{\frac{k-1}{2}} \qquad \text{density}$$

$$T = T_0/Q^2 \qquad \text{temperature}$$

$$P = P_0 Q^{\frac{k(k-1)}{2}} \qquad \text{pressure}$$

$$M_{in} = V_0(\rho_0 - \rho) \qquad \text{mass flow into the bag}$$

$$V_{B_{calc}} = V_{max}\left(1 - Q^{\frac{k(k-1)}{2}}\right) \qquad \text{calculated volume}$$

2. Computation of air bag geometry up to the time at which it is fully inflated.

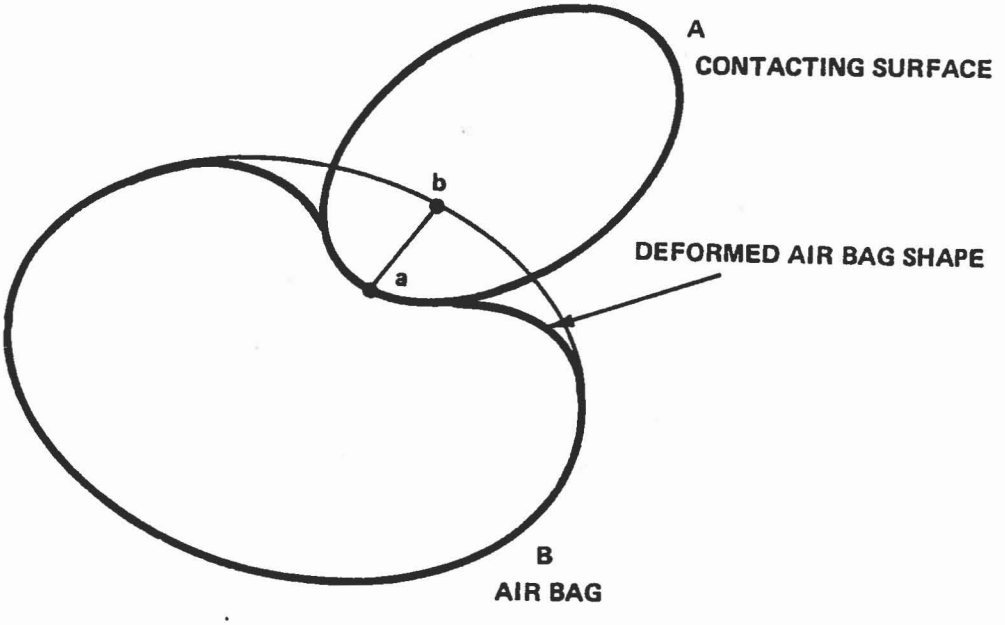
$A_k = a_k * S$ air bag semiaxes

$Z_a = D_V^T (Z_{deploy} + A_X * D_{PVCTR}) + X_{COMP}$ the position of the air bag center of gravity in inertial reference, determined by displacing it a distance A_X along the deployment vector from the deployment point.

3. Compute the volume of intersection with vehicle panels and body segments by calls to Subroutine BGG and the resulting forces and torques on the airbag.

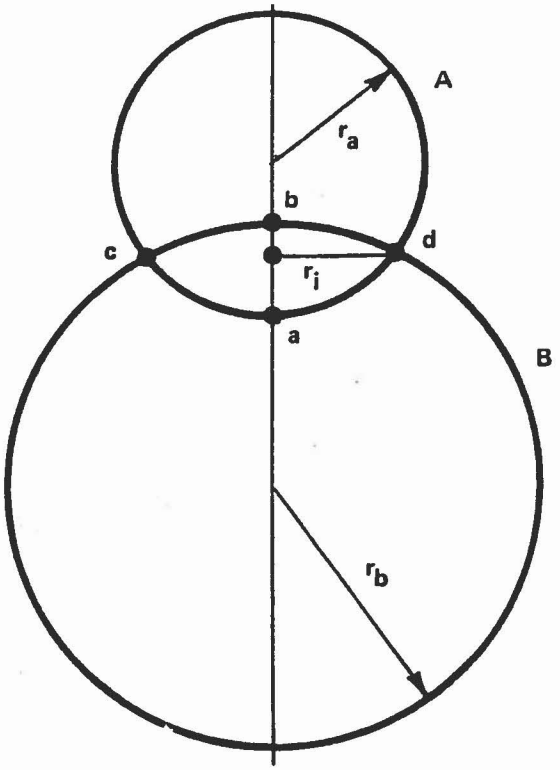
Geometry of the Air Bag

The undeformed air bag shape is represented by an ellipsoid, and all contacting surfaces, both body segments and vehicle panels, are also represented by ellipsoids in the air bag model. During air bag inflation, the air bag ellipsoid is similar (in the geometrical sense) to the fully inflated, undeformed air bag, with a volume at any given instant equal to the volume of gas (at 1 atmosphere pressure) inputted from the gas supply. Each surface contacting the air bag is considered independent of the others in calculating local deformation of the air bag. The contacting surfaces are assumed to be nondeformable, and the air bag is assumed to be deformable, but stretchless.



If a contact occurs, the points a and b of maximum penetration are computed. Point a is on A and point b is on B. These points are the two points that are a maximum distance apart in the region of contact. Two perpendicular planes that contain the line segment ab are then generated. In each plane, the radii of curvature of the respective ellipsoids are computed at the points a and b. Ellipses, defined by the intersections of the planes and ellipsoids, are then locally approximated by circles with the radii that have been computed.

Consider the resulting geometry in one of the planes:



There are two cases to consider:

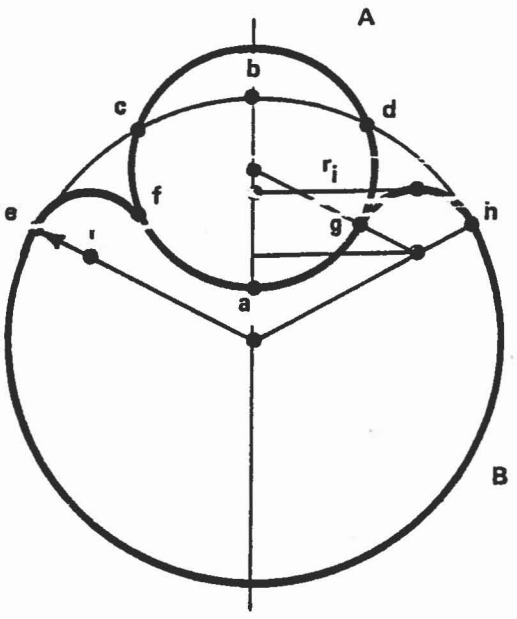
1. $r_a \geq r_b$

Because the arc length cbd is greater than the arc length cad, the air bag, considered stretchless, is assumed to collapse along the arc cad.

The estimate of decrease of volume V_i of the air bag in plane i, corresponding to the plane figure cbdac revolved about the line segment ab, is then computed.

In addition, the distance r_i of the point c (or d) from the line ab is computed. Here, r_i is the estimate of the contact area halfwidth in plane i.

2. $r_a < r_b$



In this case, the arc cbd is less than the arc cad. The air bag is assumed to deform to the shape efagh, such that the path length efagh is equal to the arc length ecbdh, corresponding to a stretchless membrane.

The arcs ef and gh are approximated by arcs of circles which are tangent to A at the points of contact f and g and to B at the points e and h. From symmetry, these circles are the same size, each of radius r.

For path length efagh to be equal to arc length ecbdh, it is necessary that

$$r = \frac{r_b - r_a}{2}$$

The center of the circle of radius r is a distance $\frac{r_b + r_a}{2}$ from the center of circle B, and is equidistant from the centers of circles A and B.

The estimate of decrease of volume V_i of the air bag in plane i, corresponding to the plane figure ecbdhgafe revolved about the line segment ab, is then computed.

In addition, the distance r_i from the center of the circle r to the line segment ab is computed. Here, r_i is the estimate of the effective contact area half-width in plane i, which includes the contributions of both membrane tension and air bag static pressure in exerting a reaction force on the contact surface.

Following the computations for the two perpendicular planes, the volume decrease of the air bag is estimated as $\frac{V_1 + V_2}{2}$, and the projected area for the computation of normal contact force is estimated as $r_1 r_2$. This normal force is applied at point a in the direction of the line ab. In addition, a coulomb friction force opposes relative motions between the contacting surface and the air bag.

6. SUBROUTINE AIRBG1

a. Purpose:

Called by Subroutine SINPUT if NBAG is not zero to process the input cards that describe the physical dimensions and gas dynamics of the air bag restraints and performs the initialization required by Subroutine AIRBAG.

b. Subroutines required:

DOT31, DOT33, DRCYPR, MAT33, PANEL.

c. Labelled common blocks used:

ABDATA, CNSNTS, CONTRL, FORCES, TITLES, SGMNTS, DESCRP, CNTSRF, INTEST, CYDATA, TEMPVS.

d. Input or argument parameters:

Cards D.4.a - D.4.h for each air bag.

e. Optional Output:

1. Cards D.4.a - D.4.h for each air bag.

2. NPRT(22) $\neq$ 0, prints the following results of the initialization of each air bag: SEGLP, SEGLV, WMEG, VBAGG, CYP0, CYSS, CYC, CYRHO0, CYVMAX, and CYORFC.

f. Procedure:

Since the motion of the airbags are now computed by the program integrator, the airbag data is stored into the segment arrays between the segment numbers for the vehicle (NVEH) and the ground (NGRND). For each air bag J = 1, NBAG (segment No. JB=NVEH+J)

1. Read and print input Cards D.4.a - D.4.h.

2. Perform air bag geometry initialization:

   (a) Compute direction cosine matrix for the Jth air bag (DBR) and for the Kth reaction panel (DRR) by calling the DRCYPR routine.

   (b) Compute fully inflated geometric air bag volume (VBAGG)

$$V_{bag} = \frac{4}{3} \pi abc$$

   and compute components of moment of inertia (BPHI)

$$\phi_{Bx} = \frac{1}{5}(b^2 + c^2)$$

$$\phi_{By} = \frac{1}{5}(a^2 + c^2)$$

$$\phi_{Bz} = \frac{1}{5}(a^2 + b^2)$$

   where a , b , c are air bag semiaxes.

(c) Set IFULL = 0 which indicates air bag is not fully inflated, set the initial air bag semiaxes (ABC) = 0 and compute the deployment vector (DPVCTR) by

$$\text{DPVCTR} = -(\text{DBR})\ (X) \text{ where } X = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

(d) Compute the coordinates of the deployment point in vehicle reference (DEPLOY) by

$$Z_{deploy} = Z_{R_1} + D^T_{RR_1} (O_{R_1} + Z_{DEP})$$

(e) Compute the initial air bag CG position parameters by calling Subroutine PANEL.

3. Perform air cylinder initialization:

(a) gauge supply pressure (CYPO): P_o = CYSP + CYPA

(b) speed of sound (CYSS): $a_0 = (kRT_0 g)^{1/2}$

(c) characteristic length (CYLO): L_0 = CYVO/CYAT

(d) gas constant (CYC): $C = \left(\frac{k-1}{2}\right) \frac{a_0 C_D}{L_0} \left(\frac{k+1}{2}\right)^{-\frac{k+1}{2(k-1)}}$

(e) density (CYRHOO): $\rho_0 = p_0/RT_0$

(f) maximum volume (CYVMAX): $V_{max} = \frac{V_0 \rho_0 RT_0}{kp_a} = \frac{V_0 p_0}{kp_a}$

(g) exhaust orifice constant (CYORFC):

$$C_0 = C_{D_0} A_0 gk\left(\frac{k+1}{2}\right)^{-\frac{k+1}{2(k-1)}}$$

7. SUBROUTINE AIRBG3 (IRESET)

a. Purpose:

This routine is called by Subroutine UPDATE to calculate the thermodynamic properties of the air bag during its inflation cycle and determines if each airbag has become fully inflated.

b. Subroutines required:

AIRBGG, PANEL, DOT31, MAT31, YPRDEG, ELTIME.

c. Labelled common blocks used:

CONTRL, JBARTZ, SGMNTS, FORCES, ABDATA, CYDATA,
CNSNTS, CNTSRF, TEMPVS.

d. Input or argument parameters:

IRSET = 1: Subroutine is called at start of integration step.
= 2: Subroutine is called at end of integration step.

e. Optional output:

None.

8. SUBROUTINE BELTG (ZA,ZB,ZC,BD)

a. Purpose:

Called by Subroutine BELTRT to compute the tangency points, vectors from the tangency points to the anchor points and the length of the restraint belt segments.

b. Subroutines required:

MAT31, CROSS, ELONG.

c. Labelled common blocks used:

CONTRL, TEMPVS, CNSNTS.

d. Input or argument parameters:

1. Arguments

ZA, ZB: location of anchor points relative to ellipsoid center.
ZC: Fixed point on the body segment associated with the belt with respect to ellipsoid center.
BD: Array of 6 elements as defined on input Cards B.2.i that contain the semiaxes and offset of the segment ellipsoid.

2. Following variables are returned to calling program through labeled common TEMPVS

APA: Tangent point A in segment reference.

UVA: Unit vector from tangent point A to the anchor point A in segment reference.

DLGA: Length of belt contacting body segment from ZC to tangent point A.

UAA: Length of belt from tangent point A to its anchor point A.

APB: Tangent point B in segment reference.

UVB: Unit vector from tangent point B to its anchor point B in segment reference.

DLGB: Length of belt contacting body segment from ZC to tangent point B.

UBB: Length of belt from tangent point B to its anchor point B.

e. Optional output:

NPRT(15) ≠ 0 APA, UVA, DLGA, UAA,
ABB, UVB, DLGB, UBB,
[BELT(I), I=12, 17]

The belt strain and force at each anchor point are stored in the BSF array for output to the tabular time histories.

f. Procedure:

1. Determine the plane of the belt and the normalized belt plane vector T_C and the distance from the belt plane to the center of the ellipsoid.

2. Calculate X_E, the center of the ellipse that is determined by the intersection of the plane defining the belt with the segment ellipsoid.

3. Calculate possible tangent points from the vectors from ellipse center to midpoint of line connecting the tangent points and the vectors from these points to the tangent points.

4. Obtain the equation of the ellipse

$$b_1x^2 + 2b_2xy + b_3y^2 = 1$$

in UC, UP coordinates where UC points to the fixed point.

5. Compute the angles of the four possible tangent points from the fixed point.

6. Choose the two tangent points that prevent the belts from criss-crossing and such that the fixed point lies on the arc of contact.

7. Determine the lengths of the belt segments contacting the ellipsoid, DLGA and DLGB, by FUNCTION ELONG.

8. Calculate belt lengths UAA and UBB and the unit vectors UVA and UVB from the tangent points to the respective anchor points.

9. SUBROUTINE BELTRT (I,II,MM,M,NT)

a. Purpose:

The routine is called by Subroutine CONTCT to compute the tangent points and belt lengths and applies the restraint belt forces to the U_1 array and belt torques to the U_2 array.

b. Subroutines required:

BELTG, DOT31, CROSS, MAT31, FRCDFL, ELTIME.

c. Labeled common blocks used:

FORCES, TABLES, TEMPVS, CNTSRF, SGMNTS, CONTRL, CNSNTS.

d. Input or argument parameters:

I: Segment identification number.
II: Ellipsoid identification number attached to segment I.
M: Belt identification number.
MM: Segment identification number to which belt M is attached.
NT: Index to NTAB array containing pointers to the function definitions for this belt-segment contact.

e. Optional output:

The initial length of each belt and its segments, and initial location of the tangent points are printed on the primary output unit.

f. Procedure:

1. Convert the segment position to segment reference and obtain the belt lengths from Subroutine BELTG.

2. If TIME = 0, divide total belt length including slack into portions A and B and store results into BELT array.

3. If zero belt friction, compute strain of entire belt and force as a function of strain by Function FRCDFL.

4. If full belt friction, compute strain and force as a function of strain for each part A and B separately by Function FRCDFL.

5. Convert forces to inertial reference and add to elements of the U_1 array for the Ith segment by:

$$U_{1_i} = U_{1_i} + W_i D^{-1}_i (F_A U_A + F_B U_A)$$

where U_A and U_B are unit vectors along portions A and B of belt.

6. Convert torques to local reference and add to the elements of the U_2 array for the Ith segment by:

$$U_{2_i} = U_{2_i} + \phi_i^{-1} [(P_A \times F_A U_A) + (P_B \times F_B U_B)]$$

where P_A and P_B are the tangent points for portions A and B of belt.

10. SUBROUTINE BGG

a. This routine is called by Subroutine AIRBGG to compute the volume of the intersection of the air bag with a body segment or ellipsoid reaction panel. The force per unit pressure and the torque per unit pressure acting on both the bag and the intersecting object are computed.

b. Subroutines required:

MAT31, MAT33, INTERS, EDEPTH, ORTHO, RCRT, DOT31, CROSS.

c. Labelled common blocks used:

TEMPVS, CNSNTS.

d. Arguments:

Air Bag Inputs:	A(3,3)	- Ellipsoid Matrix
	ZA(3)	- C.G.
	DA(3,3)	- Direction Cosine Matrix
	BFA(3)	- Offset
	VA(3)	- CG Velocity (Inertial Ref.)
	WA(3)	- Angular Velocity (Local Ref.)
Contact Surface:	B(3)	- Ellipsoid Semiaxes
	ZB(3)	- C.G.
	DB(3,3)	- Direction Cosine Matrix
	BFB(3)	- Offset
	VB(3)	- CG Velocity (Inertial Ref.)
	WB(3)	- Angular Velocity (Local Ref.)
	VSCS	- Coefficient of Sliding Friction
	IFULL	- If zero, compute volume only
	TV(3)	- Memory for Subroutines INTERS and EDEPTH

Output:

FRA(3)	-	Force on Bag
TORQ(3)	-	Torque on Bag
TOB(3)	-	Torque on Contact Surface
VOL	-	Volume of Intersection

e. Optional Output:

None.

f. Procedure:

The routine requires that the bag and reaction segment be ellipsoids. It converts appropriate quantities to the bag reference system and then checks for an intersection. If the ellipsoids intersect, the points defining the line of maximum penetration are computed. When the segments deform the bag, the point of maximum penetration on the segment is used as a reference point for application of the force and for the computation of the bag shape.

At this point two perpendicular planes containing the line of intersection are determined. In each plane the radii of curvature of the ellipsoids are computed at the points of maximum penetration. The estimates of volume and area are made by replacing the ellipses by circles with radii equal to the radii of curvature. The location and radius of a circle tangent to both the bag and segment circles is determined such that the perimeter of the resulting figure is equal to the perimeter of the bag circle. This approximates the assumption that the bag does not stretch. This figure is revolved about the line of intersection and the differential volumes and projected area are computed. The results obtained in the two planes

are averaged for the estimate of differential volume and projected area.

The force per unit area is computed and the torques per unit acting on both the bag and the segment are computed.

No attempt has been made to approximate the effect of interaction of two or more segments intersecting the bag. Each segment is considered independent of the others.

The routine has been written in such a fashion that it may be easily modified to consider the effects in more than two planes.

11. SUBROUTINE BINPUT

a. Purpose:

Reads the input cards that contains the physical dimensions and characteristics of the crash victim's body segments and joints. Also performs some program initialization.

b. Subroutines required:

DRCYPR, ELTIME.

c. Labelled common blocks used:

CNSNTS, CONTRL, DESCRP, TEMPVS, TITLES, CNTSRF, INTEST.

d. Input of argument parameters:

Input cards B.1, B.2, B.3, B.4, B.5, B.6, B.7.

e. Optional output:

NPRT(23) $\neq$ 0: prints the contents of the computed even numbered HA and HB joint arrays.

f. Procedure:

1. Reads and prints contents of:

   (a) Card B.1.
   (b) Cards B.2.i for each segment.
   (c) Cards B.3.j for each joint.
   (d) Cards B.4.j for each joint.
   (e) Cards B.5.j for each joint.
   (f) Cards B.6.i for each segment.

2. Performs the initialization:

   (a) Convert input values in the SPRING and VISC arrays from degrees to radians.

   (b) Convert input values of W array from lbs. to reciprocal mass by:

$$W_i \leftarrow G/W_i$$

   (c) For each joint j:

   (1) set vector $HA_{2j-1} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$

   and vector $HB_{2j-1} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$

(2) Compute direction cosine matrix D_j from values of yaw, pitch, and roll on Card B.3.j by calling Subroutine DRCYPR.

(3) Compute vectors $HA_{2j} = D_j\, HA_{2j-1}$

and $HB_{2j} = D_j\, HB_{2j-1}$.

(d) Set up ellipsoid matrix and inverse for 1st NSEG ellipsoids in BD_7-BD_{15} and BD_{16}-BD_{24} using the body segment semiaxes from Cards B.2.I.

12. SUBROUTINE BLKDTA

a. Purpose:

Initializes constants used by program in labeled common block CNSNTS in a manner that is independent of the computer system being utilized.

b. Subroutines required:

None.

c. Labelled common blocks used:

CNSNTS, TEMPVS.

d. Input or argument parameters:

None.

e. Optional output.

None.

f. Procedure:

Set up double precision values (for IBM 360 or 370 computers) for the following constants:

$EPS_i = 10^{-i}$ for i = 1 to 24

$PI = \text{Tan}^{-1} \frac{0}{-1}$ (using the 4 quadrant DATAN2 routine)

RADIAN = PI/180.0

13. SUBROUTINE CFACTT (A,B,D)

a. Purpose:

Given a 3x3 matrix A, computes matrix B, the transpose of cofactors of A (signed minors) and D, the determinant of A.

b. Subroutines required:

None.

c. Labelled common blocks used:

None.

d. Input or argument parameters:

A: Input matrix of size 3x3.
B: Output matrix of size 3x3, the transpose of cofactors of A.
D: Value of the determinant of A.

e. Optional output:

None.

f. Procedure:

Computes matrix B.

$$B = \begin{pmatrix} a_{22}\,a_{33} - a_{23}\,a_{32} & a_{13}\,a_{32} - a_{12}\,a_{33} & a_{12}\,a_{23} - a_{13}\,a_{22} \\ a_{23}\,a_{31} - a_{21}\,a_{33} & a_{33}\,a_{11} - a_{31}\,a_{13} & a_{21}\,a_{13} - a_{23}\,a_{11} \\ a_{32}\,a_{21} - a_{31}\,a_{22} & a_{31}\,a_{12} - a_{32}\,a_{11} & a_{11}\,a_{22} - a_{12}\,a_{21} \end{pmatrix}$$

and D by

$$D = \sum_{j=1}^{3} a_{1j}\,b_{j1}$$

The calling program computes the inverse of A by

$$A^{-1} = \frac{1}{D} B$$

14. SUBROUTINE CHAIN

a. Purpose:

Computes the linear positions and velocities of the center of gravity of all other segments from those of the reference segments.

b. Subroutines required:

DOT31, CROSS, ELTIME.

c. Labelled common blocks used:

DESCRP, CONTRL, SGMNTS, TEMPVS.

d. Input or argument parameters:

None.

e. Optional output:

NPRT(20) ≠ 0 prints TIME and the arrays SEGLP and SEGLV.

f. Procedure:

For each joint j = 1, NJNT

set m = J + 1

n = |JNT(J)|

m_j = 2*J

n_j = 2*J-1

and compute next segment position and velocity by

$$\rho_m = \rho_n + D_n^{-1} r_{nj} - D_m^{-1} r_{mj}$$

$$\dot{\rho}_m = \dot{\rho}_n + D_n^{-1}\omega_n \times r_{nj} - D_m^{-1}\omega_m \times r_{mj}$$

where

D_i is the direction cosine matrix for segment No. i.

r_{ij} is the location of joint j with respect to the c.g. of segment No. i in the local segment reference.

ω_i is the angular velocity of segment No. i in local segment reference.

(Note: by definition $|JNT(J)| \leq J$ and $n < m$ and therefore ρ_n and $\dot{\rho}_n$ have been computed previous to ρ_m and $\dot{\rho}_m$.)

If $n \leq 0$, segment m is base or reference segment of another body in which case joint j does not exist and ρ_m , $\dot{\rho}_m$ have been computed by program.

15. SUBROUTINE CINPUT

a. Purpose:

Controls the input of Cards E.1 to E.4 for the force-deflection, inertial spike, R factor, G factor, and friction coefficient function definitions, and stores pointers and function input data in the NTI and TAB arrays.

b. Subroutines required:

KINPUT, FINPUT, HINPUT.

c. Labelled common blocks used:

TABLES, TEMPVS.
(Note: this TEMPVS is shared by CINPUT, FINPUT, HINPUT and FDINIT.)

d. Input or argument parameters:

Cards E.1 - E.4 for each function.

e. Output:

Complete description of each function on primary output unit.

f. Procedure:

1. Read Card E.1 containing the function number I and title. If I > 50, go to step number 4.

2. If function number I has already been used, print diagnostic message, but proceed to replace it with new function. Place J1, the index of next element in TAB array, into NT(I).

3. Read Card E.2 which contains information that determines existence, type and range of definition for both 1st and 2nd parts of function. Read further input Cards E.3 and E.4 as required, placing data into the TAB array. Go back to step number 1 for next function.

4. Call Subroutines KINPUT, FINPUT and HINPUT.

16. SUBROUTINE CMPUTE (K,M,FT)

a. Purpose:

Called by Subroutine DINT to update the current value of TIME and to call Subroutines DZP and PDAUX at each intermediate step of the program integrator.

b. Subroutines required:

DZP, OUTPUT, PDAUX.

c. Labelled common blocks used:

CNSNTS, CDINT, COMAIN.

d. Input or argument parameters:

K: Subroutine DINT step number to be used as 4th argument of PDAUX.
M: Subroutine DINT step number to be used as 6th argument of DZP.
FT: Step size from start of current integration step.

e. Optional output:

If NPRT(26) = 2, then Subroutine OUTPUT is called to produce a line of data on each tabular time history output for each intermediate time step of the program integrator.

17. SUBROUTINE CONTCT

a. Purpose:

Called by Subroutine DAUX to control the calling of subroutines required to compute external forces and torques acting on the body segments.

b. Subroutines required:

PLELP, BELTRT, SEGSEG, AIRBAG, WINDY, HBELT, SPDAMP.

c. Labelled common blocks used:

CONTRL, FORCES, JBARTZ, TABLES, HRNESS, WINDFR.

d. Input or argument parameters:

None.

e. Optional output:

None.

f. Procedure:

1. If NPL $\neq$ 0,
   (a) Initialize NPSF = 0, which will be a counter for PSF output array.
   (b) For each plane J = 1 to NPL, fetch number of segments allowed

to contact Jth plane from MNPL(J) and place into KPL. If KPL ≠ 0, for each segment I = 1 to KPL, do:

(1) Increment NPSF by 1.
(2) Fetch segment and ellipsoid identification numbers M1, M2 and M3 from MPL(I,J).
(3) Fetch index NT to NTAB array for this contact from NTPL(I,J).
(4) Call PLELP routine with M1, M2, M3, J and NT as arguments.

2. If NBLT ≠ 0, repeat step 1 above but for allowed belt-segment contacts by calls to Subroutine BELTRT.

3. Repeat step 1 above for allowed segment-segment contacts, if any, by calls to Subroutine SEGSEG.

4. If NBAG ≠ 0, call the AIRBAG routine.

5. Repeat step 1 above for wind force calculations, if any, by calls to Subroutine WINDY.

6. If NFVSEG(6) is greater than zero, call Subroutine WINDY for force function calculations.

7. If NHRNSS is greater than zero, call Subroutine HBELT for each harness-belt system i = 1 to NHRNSS.

8. If NSD is greater than zero, call Subroutine SPDAMP for spring forces calculations.

18. SUBROUTINE CROSS (A, B, C)

a. Purpose:

Computes vector cross product C = AxB.

b. Subroutines required:

None.

c. Labelled common blocks used:

None.

d. Input or argument parameters:

A, B, C: Arrays, each consisting of 3 elements, that represent vectors where the cross product is defined by C = AxB.

e. Optional output:

None.

f. Procedure:

Computes each element of C by

$$c_1 = a_2b_3 - a_3b_2$$
$$c_2 = a_3b_1 - a_1b_3$$
$$c_3 = a_1b_2 - a_2b_1$$

19. SUBROUTINE DAUX(I1)

a. Purpose:

Computes derivatives for integrator routine:
(1) Set up initial value of arrays for system equations
(2) Modify arrays by external forces and constraints
(3) Reduce size of system of equations and solve for F, TQ and QQ.
(4) Evaluate the derivatives SEGLA ($\ddot{x}$) and WMEGD ($\dot{\omega}$).

b. Subroutines required:

ELTIME, SETUP1, VEHPOS, CHAIN, CONTCT, VISPR, EJOINT, SETUP2, FLXSEG, DAUX11, DAUX12, DAUX22, DAUX31, DAUX32, DAUX33, DAUX44, DAUX55, FSMSOL, PRINT.

c. Labelled common blocks used:

CONTRL, DESCRP, FLXBLE, SGMNTS, CMATRX, CSTRNT, CNSNTS, TEMPVS.
(Note: this TEMPVS is shared by all of the DAUX routines)

d. Input or argument parameters:

I1 = 0 : normal call-set up all arrays
≠ 0 : special call where B2, U1, U2, V1, V2 and V3 arrays have been set up by calling routine.

e. Optional output:

NPRT(8) = 1: complete print out of IJK and C matrix every time in routine.

NPRT(8) = 2: complete print out of IJK and C matrix first time in routine only.

NPRT(9) ≠ 0: output from Subroutine PRINT every time in routine.

f. Procedure:

1. If I1 ≠ 0, assume arrays B2, U1, U2, V1, V2 and V3 have either been computed by calling routine or that last computed values are still valid and go to step no. 10.

2. Call Subroutine SETUP1 to compute initial values for U1, U2, B2, V1 and V2 arrays.

3. Call Subroutine VEHPOS to compute the vehicle position and velocity.

4. Call Subroutine CHAIN to compute segment positions and velocities.

5. Call Subroutine CONTCT to compute and add to the U1 and U2 arrays all external forces and torques.

6. Call Subroutine VISPR to compute all torques acting on the joints and add them to the U2 array.

7. Call Subroutine SETUP2 to compute the V3 array.

8. Modify the U_1 and U_2 arrays by

$$U_{1i} = M_i^{-1} U_{1i} + g$$

$$U_{2i} = \varphi_i^{-1} U_{2i}$$

9. Modify U_1 and U_2 arrays by symmetry options, if any, by for each segment j

   a. If $NSYM_j = 0$, normal 3D motion
   b. If $NSYM_j = j$, allow no lateral motion by setting

      $U_{1y} = U_{2x} = U_{2y} = 0$ for segment j.

   c. If $NSYM_j = k$, segment j is symmetric to segment k with all motion in the X-Z plane and no lateral motion by

      $$f_j = f_k = (f_j + f_k)/2$$

      for $f = U_{1x}, U_{1z}$ and U_{2y}

      $$U_{1y} = U_{2x} = U_{2y} = 0$$

d. If $NSYM_j = -k$, segment j is mirror symmetric to segment k, equal but opposite lateral motions is permitted by

$$f_j = f_k = (f_j + f_k)/2$$

for $f = U_{1x}, U_{1z}$ and U_{2y}

$$f_j = -f_k = (f_j - f_k)/2$$

for $f = U_{1y}, U_{2x}$ and U_{2z}

10. Eliminate SEGLA (x) and WMEGD ( ) from the system of equations

$$\begin{pmatrix} M & 0 & A_{11} & 0 & A_{13} \\ 0 & \Phi & A_{21} & A_{22} & A_{23} \\ A_{11}^T & A_{21}^T & C_{11} & 0 & 0 \\ 0 & A_{22}^T & 0 & C_{22} & 0 \\ A_{13}^T & A_{23}^T & 0 & 0 & C_{33} \end{pmatrix} \begin{pmatrix} \ddot{x} \\ \dot{\omega} \\ f \\ t \\ q \end{pmatrix} = \begin{pmatrix} u_1 \\ u_2 \\ V_1 \\ V_2 \\ V_3 \end{pmatrix}$$

to obtain

$$\begin{pmatrix} M & 0 & A_{11} & 0 & A_{13} \\ 0 & \Phi & A_{21} & A_{22} & A_{23} \\ 0 & 0 & C_{11} & C_{12} & C_{13} \\ 0 & 0 & C_{21} & C_{22} & C_{23} \\ 0 & 0 & C_{31} & C_{32} & C_{33} \end{pmatrix} \begin{pmatrix} \ddot{x} \\ \dot{\omega} \\ f \\ t \\ q \end{pmatrix} = \begin{pmatrix} u_1 \\ u_2 \\ R_1 \\ R_2 \\ R_3 \end{pmatrix}$$

by calling Subroutines DAUX11, DAUX12, DAUX22, DAUX31, DAUX32 and DAUX33.

11. Solve the reduced set of equations

$$\begin{pmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{pmatrix} \begin{pmatrix} f \\ t \\ q \end{pmatrix} = \begin{pmatrix} R_1 \\ R_2 \\ R_3 \end{pmatrix}$$

for f, t and q by calling Subroutine FSMSOL.

12. Solve the original set of equations for $\ddot{x}$ and $\dot{\omega}$ by

$$\ddot{x} = U_1 - M^{-1}A_{11}f - M^{-1}A_{13}q$$

$$\dot{\omega} = U_2 - \Phi^{-1}A_{21} - \Phi^{-1}A_{22}t - \Phi^{-1}A_{23}q$$

and return to the calling routine.

20. SUBROUTINE DAUX11

a. Purpose:

Called by Subroutine DAUX if the number of joints (NJNT) is nonzero to compute

$$C_{11} = B_{11}M^{-1}A_{11} + B_{12}\Phi^{-1}A_{21}$$

$$R_1 = B_{11}M^{-1}U_1 + B_{12}\Phi^{-1}U_2 - V_1$$

b. Subroutines required:

ELTIME.

c. Labelled common blocks used:

CONTRL, DESCRP, SGMNTS, CMATRX, TEMPVS.
(Note: this TEMPVS is shared by all of the DAUX routines.)

d. Input or argument parameters:

None.

e. Optional output:

None.

21. SUBROUTINE DAUX12

a. Purpose:

Called by Subroutine DAUX if the number of joints (NJNT) is nonzero to compute

$$C_{12} = B_{12} \Phi^{-1} A_{22}$$

$$C_{21} = C_{12}^{T}$$

b. Subroutines required:

ELTIME.

c. Labelled common blocks used:

CONTRL, DESCRT, CMATRX, TEMPVS.
(Note: this TEMPVS is shared by all of the DAUX routines.)

d. Input or argument parameters:

None.

e. Optional output:

None.

22. SUBROUTINE DAUX22

a. Purpose:

Called by Subroutine DAUX if the number of joints (NJNT) is nonzero to compute

$$C_{22}^{\cdot} = B_{22}\,\Phi^{-1}A_{22} - B_{24}$$

$$R_Z = B_{22}\,\Phi^{-1}U_Z - V_2$$

b. Subroutines required:

ELTIME.

c. Labelled common blocks used:

CONTRL, DESCRP, SGMNTS, CMATRX, CEULER, TEMPVS.
(Note: this TEMPVS is shared by all of the DAUX routines.)

d. Input or argument parameters:

None.

e. Optional output:

None.

23. SUBROUTINE DAUX31

a. Purpose:

Called by Subroutine DAUX if the number of joints (NJNT) and the number of constraints (NQ) are both nonzero to compute

$$C_{13} = B_{11}M^{-1}A_{13} + B_{12}\Phi^{-1}A_{23}$$

$$C_{31} = B_{31}M^{-1}A_{11} + B_{32}\Phi^{-1}A_{21}$$

b. Subroutines required:

ELTIME.

c. Labelled common blocks used:

CONTRL, DESCRP, CMATRX, CSTRNT, TEMPVS.
(Note: this TEMPVS is shared by all of the DAUX routines.)

d. Input or argument parameters:

None.

e. Optional output:

None.

24. SUBROUTINE DAUX32

a. Purpose:

Called by Subroutine DAUX if the number of joints (NJNT) and the number of constraints (NQ) are both nonzero to compute

$$C_{23} = B_{22}\Phi^{-1}A_{23}$$

$$C_{32} = B_{32}\Phi^{-1}A_{22}$$

b. Subroutines required:

ELTIME.

c. Labelled common blocks used:

CONTRL, DESCRP, CMATRX, CSTRNT, TEMPVS.
(Note: this TEMPVS is shared by all of the DAUX routines.)

d. Input or argument parameters:

None.

e. Optional output:

None.

25. SUBROUTINE DAUX33

a. Purpose:

Called by Subroutine DAUX if the number of constraints (NQ) is nonzero to compute

$$C_{33} = B_{31}M^{-1}A_{13} + B_{32}\Phi^{-1}A_{23} - B_{35}$$

$$R_3 = B_{31}M^{-1}U_1 + B_{32}\Phi^{-1}U_2 - V_3$$

b. Subroutines required:

ELTIME.

c. Labelled common blocks used:

CONTRL, DESCRP, SGMNTS, CMATRX, CSTRNT.
(Note: this TEMPVS is shared by all of the DAUX routines.)

d. Input or argument parameters:

None.

e. Optional Output:

None.

26. SUBROUTINE DAUX44

a. Purpose:

Called by Subroutine DAUX if the number of joints (NJNT) and the number of flexible elements (NFLX) are both nonzero to compute

$$C_{44} = B_{42}\,\Phi^{-1}A_{24}$$

$$R_4 = B_{42}\,\Phi^{-1}U_2 - V_4$$

b. Subroutines required:

ELTIME.

c. Labelled common blocks used:

CONTRL, SGMNTS, DESCRP, CMATRX, CSTRNT, FLXBLE, TEMPVS.
(Note: this TEMPVS is shared by all of the DAUX routines.)

d. Input or argument parameters:

None.

e. Optional output:

None.

27. SUBROUTINE DAUX55

a. Purpose:

Called by Subroutine DAUX if the number of singular elements (NS) is nonzero to compute

$$C_{1,5,5} = M \quad \text{and} \quad R_{1,5} = U_1 + Mg$$
$$C_{2,5,5} = \Phi \qquad\quad R_{2,5} = U_2$$

b. Subroutines required:

ELTIME.

c. Labelled common blocks used:

CONTRL, SGMNTS, DESCRP, CMATRX, CSTRNT, FLXBLE, CNSNTS, TEMPVS. (Note: this TEMPVS is shared by all of the DAUX routines.)

d. Input or argument parameters:

None:

e. Optional output:

None.

28. SUBROUTINE DHHPIN (DD,BN,L,M,N)

a. Purpose:

Called by Subroutine SETUP2 to return DD = D_1 if joint m is not pinned or DD = D_1 (I - $hb_n hb_n$.) if joint m is pinned.

b. Subroutines required:

None.

c. Labelled common blocks used:

DESCRP, SGMNTS, CEULER.

d. Input or argument parameters:

DD: Array of size (3,3) that contains result to be returned to calling routine.

BN: Array of size (3) that contains zero if joint m is not pinned or $D_1 hb_n hb_n$. if joint m is pinned to be returned to calling routine.

L: Index of direction cosine matrix to be used.

M: Joint identification number.

N: Index of hb array to be used.

e. Optional output:

None.

29. SUBROUTINE DINT

a. Purpose:

Called by the main program to control the integration of the state variables between program time points.

b. Subroutines required:

ADJUST, CMPUTE, ELTIME, OUTPUT, PDAUX, QSET, TRIGFS, UPDATE.

c. Labelled common blocks used:

CONTRL, INTEST, CNSNTS, CDINT, COMAIN.

d. Input or argument parameters:
(Transmitted via COMMON/COMAIN/)

ISTEP : Integration step number.
NEQ : Number of variables.
DT : Print time interval desired.
HO : Initial integration step size.
HMAX : Maximum integration step size.
HMIN : Minimum integration step size.
TIME : Time to be initialized by calling routine and updated by this routine.
VAR : Array of state variables.
DER : Array of derivatives of state variables.
NDINT : Number of estimates of integration parameters to be made at the end of any intermediate time step.

e. Optional output:

If NPRT(25) ≠ 0, the results of the convergence tests are printed at each step.
If NPRT(26) = 2, Subroutine OUTPUT is called at the end of each intermediate integration step.

f. Procedure:

This routine is a variable step integrator which returns to the calling program whenever an estimate of the variables at a specified time interval is achieved. The integration procedure is based on a fourth order modified Runge-Kutta technique which also includes an exponential term for each variable. The basic form of the derivatives is

$$\dot{x} = \alpha(x-x_0) + a_0 + a_1 t + a_2 t^2$$

where α, x_0, a_0, a_1 and a_2 are parameters that are being updated by the routine.

1. If ISTEP = 1 , it is the first time in the routine and initialization is performed, variables are computed for the initial time point and control is returned to calling program.

2. At the start of a new print point interval, the next print time is updated, if the integration time is greater than this new print time, variables are interpolated for the new print time and control is returned to calling program.

3. At the start of a new integration step interval, Subroutine

UPDATE(K) is called with K=1, to update time history information to be used for the interval. If K is returned as -1, it means that UPDATE has exercised the impulse option, the derivatives are re-evaluated for the start of the time interval and the integration parameters are reinitialized.

4. Two evaluations of the derivatives and variables are performed at $t = t_0 + h/2$ and the integration parameters are updated.

5. NDINT evaluations of the derivatives and variables are performed at $t = t_0 + h$ and a least square fit of the integration parameters is performed.

6. A final evaluation of the derivatives and variables is performed at $t = t_0 + h$, these new derivatives are compared with the values of the functional form to test if the integration step has converged. These tests are controlled by the input data that were supplied on Cards B.6.

7. If the convergence test has failed and if h > HMIN, the step size is halved and control is returned to step No. 4.

8. If the convergence test has been successful for three consecutive steps, and if h < HMAX, then the integration step size, h, is doubled.

9. If the integration is at the next print time point, control is returned to the calling program.

10. If the upcoming integration step size would take us beyond the print time point, h is reduced so as to integrate to the print point. Control is then returned to step No. 3.

30. SUBROUTINE DOTT31 (A,B,C)

a. Purpose:

Performs the matrix multiplication C = AB', where C is a 3x3 matrix, A and B are vectors with 3 components, and B' is the transpose of B.

b. Subroutines required:

None.

c. Labelled common blocks used:

None.

d. Input or argument parameters:

A,B: Vectors with 3 components.
C: Product matrix of size 3x3.

e. Optional output:

None.

f. Procedure:

Each element c_{ij} of the product matrix C(3,3) is computed by

$c_{ij} = a_i b_j$ for both i and j = 1 to 3.

31. SUBROUTINE DOTT33 (A,B,C)

a. Purpose:

Performs the matrix multiplication C = AB', where A, B and C are all 3x3 matrices, and B' is the transpose of B.

b. Subroutines required:

None.

c. Labelled common blocks used:

None.

d. Input or argument parameters:

A,B: Matrices of size 3x3.
C: Product matrix of size 3x3.

e. Optional output:

None.

f. Procedure:

Each element c_{ij} of the product matrix C(3,3) is computed by

$$c_{ij} = \sum_{k=1}^{3} a_{ik}\, b_{jk} \quad \text{for both i and j = 1 to 3.}$$

32. SUBROUTINE DOT31 (A,B,C)

a. Purpose:

Performs the matrix multiplication C = A'B, where A is a 3x3 matrix, B and C are vectors with 3 components, and A' is the transpose of A.

b. Subroutines required:

None.

c. Labelled common blocks used:

None.

d. Input or argument parameters:

A: Matrix of size 3x3.
B: Vector with 3 components.
C: Product vector with 3 components.

e. Optional output:

None.

f. Procedure:

Each element c_i of the product vector C(3) is computed by

$$c_i = \sum_{k=1}^{3} a_{ki}\, b_k \quad \text{for } i = 1 \text{ to } 3.$$

33. SUBROUTINE DOT33 (A,B,C)

a. Purpose:

Performs the matrix multiplication C = A'B, where A, B and C are all 3x3 matrices, and A' is the transpose of A.

b. Subroutines required:

None.

c. Labelled common blocks used:

None.

d. Input or argument parameters:

A,B: Matrices of size 3x3.
C: Product matrix of size 3x3.

e. Optional output:

None.

f. Procedure:

Each element c_{ij} of the product matrix C(3,3) is computed by

$$c_{ij} = \sum_{k=1}^{3} a_{ki} \, b_{kj} \quad \text{for both i and j = 1 to 3.}$$

34. SUBROUTINE DRCIJK (D,ANG,ID,HT,J)

a. Purpose:

Called by Subroutine INITAL to compute the direction cosine matrix (D) of segment No. J from the rotation angles (ANG) that can be relative to either inertial reference, another segment reference or a joint reference depending on the value of the 4th element of ID.

b. Subroutines required:

DOT33, DRCYPR, MAT33.

c. Labelled common blocks used:

None.

d. Input or argument parameters:

D: 3x3 direction cosine matrix to be computed.
ANG: 3 rotation angles given in degrees.
ID: 4 integers as defined on input Card G.3.j1, first three are used as the 3rd argument to Subroutine DRCYPR and the fourth defines the relative orientation of the rotation angles.
HT: The transpose of direction cosine matricees that defines the orientation of a joint with respect to its adjacent segments.
J: The segment identification number.

e. Optional output:

None.

f. Procedure:

The values of ANG and ID are used to compute a direction cosine matrix R by calling Subroutine DRCYPR. The direction cosine matrix D(J) of segment No. J is then determined by the value of M = ID(4) as follows:

M = 0: D(J) = R(J)

M > 0: D(J) = R(J)D(M)

M < 0: D(J) = HT(J)R(J)HT'(K)D(K) where K = -M

35. SUBROUTINE DRCYPR (D,A,ID)

a. Purpose:

Sets up direction cosine matrix D for rotation angles A given in degrees about the x, y, or z axis as indicated by I1, I2, or I3.

b. Subroutines required:

ROT.

c. Labelled common blocks used:

CNSNTS.

d. Input or argument parameters:

D: 3x3 direction cosine matrix to be computed.
A: 3 rotation angles given in degrees.
ID: 3 integers (I1,I2 and I3) that indicates axis of rotation for each of the 3 angles in A (1, 2, or 3 indicates x, y, or z axis, respectively).

e. Optional output:

None.

f. Procedure:

Computes as a matrix product

$$D = D_{I3}\, D_{I2}\, D_{I1}$$

where each D_i is one of the following

$$D_1 = \begin{vmatrix} 1 & 0 & 0 \\ 0 & \text{Cos}\alpha & \text{Sin}\alpha \\ 0 & -\text{Sin}\alpha & \text{Cos}\alpha \end{vmatrix}$$

$$D_2 = \begin{vmatrix} \text{Cos}\alpha & 0 & -\text{Sin}\alpha \\ 0 & 1 & 0 \\ \text{Sin}\alpha & 0 & \text{Cos}\alpha \end{vmatrix}$$

$$D_3 = \begin{vmatrix} \text{Cos}\alpha & \text{Sin}\alpha & 0 \\ -\text{Sin}\alpha & \text{Cos}\alpha & 0 \\ 0 & 0 & 1 \end{vmatrix}$$

depending as I1, I2, and I3 have values of 1, 2, or 3.

Note: (1) for the normal sequence of yaw (ψ), pitch (θ), and roll of (ϕ), let ID = 3, 2, 1 to obtain

$$D = D_\phi\, D_\theta\, D_\psi = D_1 D_2 D_3$$

(2) for the reverse sequence as required by Subroutine INITAL prior to Version 18 of the program, let ID = 1, 2, 3 to obtain

$$D = D_\psi\, D_\theta\, D_\phi = D_3 D_2 D_1$$

(3) for Euler angles, precession (ϕ), nutations (θ), and spin (ψ), let ID = 3, 1, -3 to obtain

$$D = D_\psi\, D_\theta\, D_\phi = D_3 D_1 D_3$$

36. SUBROUTINE DSETD (D,TH,T)

a. Purpose:

Updates a direction cosine using an incremental angular motion and renormalizes resulting matrix.

b. Subroutines required:

CFACTT.

c. Labelled common blocks used:

CNSNTS.

d. Input or argument parameters:

D: Direction cosine matrix.
TH: Components of incremented angular motion about local x, y, z axis, respectively.
T: Magnitude of vector TH(K), computed by subroutine.

e. Optional output:

None (however an error message is printed if the renormalization iteration does not converge).

f. Procedure:

Since $\dot{D} = -\omega \times D$, for small angular motions we may write (as an approximation)

$$D(t) = e^{-\int_{t_o}^{\tau} (\omega \times) d\tau} \qquad D(t_o) \equiv e^{-(\Theta \times)} D$$

We have $\Theta x \equiv \int_{t_0}^{t} (\omega \times) dT$ where Θ is the integral of ω.

The routine updates D by

$$D = e^{-\Theta x} D \qquad \text{where}$$

$$e^{-\Theta x} = I \mathrm{Cos}|\Theta| - \frac{\mathrm{Sin}|\Theta|}{|\Theta|} \Theta x + \left(\frac{\mathrm{Sin}|\Theta|}{|\Theta|}\right)^2 \frac{\Theta\Theta \cdot}{1 + \mathrm{Cos}|\Theta|}$$

$$|\Theta| = \left(\Theta_x^2 + \Theta_y^2 + \Theta_z^2\right)^{1/2}$$

The routine then renormalizes the direction cosine matrix by averaging the matrix and transpose of its inverse.

37. SUBROUTINE DSETQ (E,TH,ES,EC,D)

a. Purpose:

Compute new direction cosine matrix (D) given original matrix (E) and incremental motion expressed in quaternion form.

b. Subroutines required:

CFACTT.

c. Labelled common blocks used:

CNSNTS.

d. Input or argument parameters:

E : Original or saved direction cosine matrix.

TH : Components of Q, i.e. $\mu_x \text{Sin } \theta/2, \mu_y \text{Sin } \theta/2, \mu_z \text{Sin } \theta/2$

ES : $\text{Sin}^2 \ \theta/2$

EC : $\text{Cos } \theta/2$

D : New direction cosine matrix to be computed.

e. Optional output:

None.

f. Procedure:

Computes

(1) $D = RE$

where

$$R = 2vv. + I(1-2\sin^2 \theta/2) - 2\cos \theta/2\ vx$$

$$v = \sin \theta/2\ y$$

(2) D is then renormalized by iterating

$$D_{n+1} = (D_n + D_n^{-1^T})/2$$

until $|\det D_{n+1} - 1| < 10^{-6}$

38. SUBROUTINE DSMSOL (A,N,LL)

a. Purpose:

To solve a set of simultaneous linear algebraic equations, AX = B.

b. Subroutines required:

None.

c. Labelled common blocks used:

None.

d. Input or argument parameters:

A: A two-dimensional matrix Nx(N+1) of coefficients.
N: The number of equations and unknowns.
LL: First dimension of A in calling program (LL $\geq$ N); second dimension must be $\geq$ N + 1.

(Note: Calling program must set up A(I,J) for I and J = 1 to N and A(I,N+1) = B(I) for I = 1 to N. The solution is returned in column N+1 of matrix A and matrix A is destroyed by the subroutine.)

e. Optional output:

None (however, an error message is printed, and the job is terminated if A is singular).

f. Procedure:

Solution is obtained by elimination, using the largest pivotal divisor of each column. Each stage of elimination consists of interchanging rows when necessary to avoid division by zero or small numbers.

The forward solution to obtain variable N is done in N stages. The back solution for the other variables is calculated by successive substitutions. Final solution values are developed in column N+1 of matrix A, with variable 1 in A(1,N+1), variable 2 in A(2,N+1),..., variable N in A(N,N+1).

39. SUBROUTINE DZP (N,X,G,E,R,M)

a. Purpose:

Called by Subroutine CMPUTE to evaluate the state variables from the parametric form assumed in the integration routine. It also evaluates the exponential weights if required.

b. Subroutines required:

ELTIME.

c. Labelled common blocks used:

CNSNTS.

d. Input or argument parameters:

N : number of state variables
X(I) : state variables
G(5,N) : array of parameters
E(3,N) : exponential weights
R : time step size
M = 0 : evaluate state variable only
≠ 0 : also evaluate exponential weights.

e. Optional output:

None.

f. Procedure:

The integrator assumes that the derivative $\dot{x}$ of a state variable x can be approximated by:

$$\dot{x} = G_5(x-G_1) + G_2 + G_3t + G_4t^2,$$

for $0 \leq t \leq h$ where h is the current integrator step size, and x is given by:

$$x = G_1 + G_2tE_1 + G_3t^2E_2 + G_4t^3E_3$$

where

$$E_1 = (e^a-1)/a$$

$$E_2 = (E_1-1)/a$$

$$E_3 = (2\ E_2-1)/a \qquad a = G_5t$$

The subroutine evaluates x from this formula, and evaluates the E's if G_5 or t has been changed as indicated by M.

40. SUBROUTINE EDEPTH (A,B,M,T,X,XA,XB)

a. Purpose:

Determines points X_A on A and X_B on B, the points of deepest penetration of the two ellipsoids, X'AX = 1 and (X'-M')B(X-M) = 1.

b. Subroutines required:

DSMSOL, MAT33.

c. Labelled common blocks used:

CONTRL, CNSNTS.

d. Input or argument parameters:

A, B, M, T, X: Same as for Subroutine INTERS.
XA, XB: Points of deepest penetration of intersecting ellipsoids A and B returned to calling program.

e. Optional output:

NPRT(17) ≠ 0: Points λ , μ (defined herein), X_A and X_B for each iteration step.

f. Procedure:

1. Initial guesses:

$$X_A = \frac{1}{t} X$$

$$X_B = M + \frac{1}{t}(X-M)$$

$$= - \frac{|X_B - X_A|}{|AX_A|}$$

$$= - \frac{|X_B - X_A|}{|B(X_B - M)|}$$

2. Start of iteration, form matrices:

$$C_1 = \lambda\mu AB + \lambda A + \mu B$$

$$C_2 = C_1$$

$$C_3 = C_1'$$

3. Solve $C_1(X_B - M) = -\lambda AM$ for $(X_B - M)$.

4. Evaluate:

$$X_B = (X_B - M) + M$$

$$B\ (X_B - M)$$

$$AX_A$$

$$C_{13} = \frac{1}{2}\ (1 - X_A' AX_A)$$

$$C_{23} = \frac{1}{2}\ [1 - (X_B - M)'B(X_B - M)]$$

5. Solve $C_2 \dfrac{\partial X_B}{\partial \lambda} = -AX_A$ for $\dfrac{\partial X_B}{\partial \lambda}$.

6. Evaluate:

$$\frac{\delta X_A}{\delta \lambda} = \frac{\delta X_B}{\delta \lambda} + \mu B \frac{\delta X_B}{\delta \lambda}$$

$$C_{11} = X_A' A \frac{\delta X_B}{\delta \lambda}$$

$$C_{21} = (X_B - M)' B \frac{\delta X_B}{\delta \lambda}$$

7. Solve $C_3 \frac{\delta X_A}{\delta \mu} = -B(X_B - M)$ for $\frac{\delta X_A}{\delta \mu}$

8. Evaluate:

$$\frac{\delta X_B}{\delta \mu} = \frac{\delta X_A}{\delta \mu} + \lambda A \frac{\delta X_A}{\delta \lambda}$$

$$C_{12} = X_A' A \frac{\delta X_A}{\delta \mu}$$

$$C_{22} = (X_B - M) B \frac{\delta X_B}{\delta \mu}$$

9. Solve for $\Delta\lambda$ and $\Delta\mu$,

$$C_{11}\Delta\lambda + C_{12}\Delta\mu = C_{13}$$

$$C_{21}\Delta\lambda + C_{22}\Delta\mu = C_{23}$$

10. Increment λ and μ , $\lambda + \Delta\lambda \rightarrow \lambda$ and $\mu + \Delta\mu \rightarrow \mu$.

11. Test for convergence

if $\left|\frac{\Delta\lambda}{\lambda}\right| > \epsilon$ or $\left|\frac{\Delta\mu}{\mu}\right| > \epsilon$, go back to step 2.

12. Return X_A and X_B to calling program.

41. FUNCTION EFUNCT (TH,THD,SRO,JSTOP)

a. Purpose:

Called by Subroutines VISPR and EJOINT to compute the nonlinear spring torques for joints. (Note: in earlier versions of the CVS program, Subroutine VISPR called Function SPRNGF to perform this function.)

b. Subroutines required:

None.

c. Labelled common blocks used:

None.

d. Input or argument parameters:

TH: Angle of the joint axis.
THD: Time derivative of TH.
SPR: Array of 5 values describing the function evaluation.
JSTOP: Indicator to be set to one if in the joint stop.

e. Optional output:

None.

42. SUBROUTINE EJOINT (IJ,NJ)

a. Purpose:

Called by Subroutines DAUX and IMPULS to compute the torques acting on an Euler joint and adds them to the U2 array for the adjacent segments.

b. Subroutines required:

CROSS, DOT31, DOT33, EFUNCT, ELTIME, EULRAD, GLOBAL, MAT31, ROT, VISCOS.

c. Labelled common blocks used:

CONTRL, SGMNTS, DESCRP, CMATRX, CEULER, FORCES, TEMPVI, CNSNTS, TEMPVS.

d. Input or argument parameters:

NJ = 0: Normal computations for all Euler joints.
≠ 0: IMPULS computations for joint no. NJ impulse.
IJ = 1: Impulse on precession axis only.
2: Impulse on nutation axis only.
3: Impulse on spin axis only.
4: Impulse on globalgraphic axis.

e. Optional output:

Data is stored into the PRJNT array for Subroutine OUTPUT.

43. SUBROUTINE ELONG (A,B,C,D,E)

a. Purpose:

Called by Subroutine BELTG to compute arc length of the ellipse

$Ax^2 + 2Bxy + Cy^2 = 1$ from $\theta = 0$ (positive x axis) to $\theta_1 = E$.

b. Subroutines required:

None.

c. Labelled common blocks used:

None.

d. Input or argument parameters:

A, B, C: Coefficients of ellipse $Ax^2 + 2Bxy + Cy^2 = 1$.
D: Maximum $\Delta\theta$ for integration
E: Integration is performed from $\theta = 0$ to $\theta_1 = E$ (θ_1 in radians).

e. Optional output:

None.

f. Procedure:

Given the equation of an ellipse $ax^2 + 2bxy + cy^2 = 1$,

let

$$x = \rho \cos\theta$$

$$y = \rho \sin\theta$$

$$\text{arc length} = \int_0^{\theta_1} \sqrt{\rho^2 + \left(\frac{d\rho}{d\theta}\right)^2}\, d\theta \quad > 0 \text{ if } \theta_1 > 0, \quad < 0 \text{ if } \theta_1 < 0,$$

where $\rho = (a\cos^2\theta + 2b\cos\theta\sin\theta + c\sin^2\theta)^{-1/2}$.

Integration is done by Simpson's rule with maximum step size $\Delta\theta$. Initial value of ρ for $\theta = 0$ is $1/\sqrt{a}$, further values are updated by a one step Newton-Raphson iteration of square root from previous value. Sine and cosine are updated by:

$$\cos(\theta + \Delta\theta) = \cos\theta\cos\Delta\theta - \sin\theta\sin\Delta\theta$$

$$\sin(\theta + \Delta\theta) = \sin\theta\cos\Delta\theta + \cos\theta\sin\Delta\theta$$

44. SUBROUTINE ELTIME (L,N)

a. Purpose:

Counts the number of times certain subroutine (identified by N) are called and accounts for all computer time used by these subroutines.

b. Subroutines required:

LTIME.

c. Labelled common blocks used:

None.

d. Input or argument parameters:

L: A value of 1 or 2 indicating that this routine is being called at the beginning or the end of the Nth subroutine.

N: Subroutine identification number as follows:

1. MAIN3D	11. CHAIN	21. PLELP	31. EJOINT
2. INPUT	12. CONTCT	22. BELTRT	32. SPDAMP
3. DINT	13. VISPR	23. SEGSEG	33. DAUX44
4. PRIPLT	14. DAUX11	24. AIRBAG	34. FLXSEG
5. DZP	15. DAUX12	25. RSTART	35. EQUILB
6. PDAUX	16. DAUX22	26. SETUP2	36. POSTPR
7. UPDATE	17. DAUX31	27. IMPULS	37. WINDY
8. OUTPUT	18. DUAX32	28. IMPLS2	38. HBELT
9. DAUX	19. DAUX33	29. AIRBG3	39. HPTURB
10. SETUP1	20. FSMSOL	30. DAUX55	

e. Optional output:

When ELTIME(2,1) is called, a table is printed listing the number of calls, the CPU time and percent of total CPU time used by each of the subroutines listed above.

f. Procedure:

1. If L = 1 and N = 1 (initial call at beginning of main program), the LTIME counter is initialized by NT1(1) = LTIME(0), all number of calls (NC) and number of time units counters (NT) are set to zero and NC(1) is set to 1.

2. If L = 1 and N ≠ 1 (call at beginning of Nth subroutine), NT1(N) = LTIME(1) which measures the total CPU time expended since start of program and NC(N) is incremented by 1.

3. If L = 2 and N ≠ 1 (call at end of Nth subroutine), total time from beginning to end of the Nth Subroutine (including all subroutines it calls) is computed and summed by NT(N) = LTIME(1) - NT1(N) + NT(N).

4. If L = 2 and N = 1 (call at end of main program):
   (a) Total CPU time is computed by NT(1) = LTIME(1).
   (b) Each subroutine total CPU time is decreased by the CPU time of the subroutines it calls.
   (c) The % of CPU time expended by each subroutine is computed.
   (d) The table described above is printed.

45. SUBROUTINE EQUILB (YPR,ID)

a. Purpose:

Called by Subroutine INITAL if I1 = 15 on input Card G.1.a to adjust the initial input position parameters supplied on Cards G.2 and G.3 such that the normal contact forces are equal to either supplied values or those computed by constraint forces.

b. Subroutines required:

CHAIN, DAUX, DOTT33, DOT31, DRCIJK, ELTIME, FRCDFL, MAT31, OUTPUT, PRINT, XDY.

c. Labelled common blocks used:

CONTRL, SGMNTS, DESCRP, CMATRX, CSTRNT, TABLES, JBARTZ, CNTSRF, CNSNTS, TITLES, TEMPVS.

d. Input or argument parameters:

Input Cards G.4, G.5 and G.6.

YPR,ID: The YPR and ID arrays for the NSEG segments that were read from input Cards G.3.j1 by Subroutine INITAL.

e. Optional output:

An annotated listing of input Cards G.4, G.5 and G.6 is produced on the primary output unit.

If NPRT(27) is nonzero, then a comprehensive dignostic output of every iteration step is produced on the primary output unit including that produced by calling Subroutines PRINT and OUTPUT.

After convergence has been achieved, a list of all of the initial position parameters that have been changed is produced on the primary output unit.

f. Procedure:

1. Subroutine EQUILB is called by Subroutine INITAL if I1 = 15 on input Card G.1.a.

2. Input Cards G.4, G.5.a-G.5.n and G.6.a-G.6.m are read and printed. In addition the variables are checked to ascertain they meet the following conditions:

$1 \leq NVAR \leq 10$	
$1 \leq NCON \leq 5$	
$1 \leq NTV(J) \leq 2$	for each J from 1 to NVAR
$1 \leq NI1(J) \leq 3$	
$1 \leq NSG(J) \leq NSEG$	
$0 \leq NAV(J) \leq 5$	
$1 \leq JPL(J) \leq NPL$	
$1 \leq JSG(J) \leq NSEG$	
$1 \leq MNPL(K) \leq 5$	for K = JPL(J)
JSG(J) = MPL(2,I,K)	for some I from 1 to MNPL(K)
$1 \leq KSG(I,J) \leq NSEG$	for each I from 1 to NAV(J) $\neq$ 0
$1 \leq IPL(I) \leq NPL$	for each I from 1 to NCON $\neq$ 0

$1 \leq ISG(I) \leq NSEG$

$3 \leq LTYPE(I) \leq 4$

$0 \leq INDGX(I) \leq NVAR$

$1 \leq MNPL(J) \leq 5$ for $J = IPL(I)$

$ISG(I) = MPL(2,K,J)$ for some K from 1 to MNPL(J)

$IPL(I) = JPL(K)$ for $K = INDGX(I) \neq 0$

$ISG(I) = JSG(K)$

If any of the above conditions are not satisfied, an error message is printed and the program is terminated after the card in error is printed.

3. Required data initialization for the double iteration is performed.

4. If NPRT(27) ≠ 0, optional output for the solution of the system equations using the initial position parameters is performed.

5. Start iteration (maximum of 10 steps) of force deflection normal forces to equal constraint normal forces.

6. Set up sequence for index j (where j will be one of the NVAR input variables to be adjusted) so that the adjustment to each variable will reflect adjustments made to other variables. Sequence will be j = 1,2; 1,2,3; 1, ----, NVAR and then entire sequence is repeated once more.

7. Start iteration (maximum of 10 steps normally but 50 steps for first time to compute derivative DXP_j) to adjust initial position variable X_j such that $F(X_j)$ will equal $G(X_j)$, where $F(X_j)$ is the

normal component of the force deflection function to be controlled by variable X_j, and $G(X_j)$ is the desired value of $F(X_j)$.

8. Call Subroutine CHAIN to compute position of c.g. of all segments for the current values of all input position parameters.

9. Compute (by shortened version of Subroutine PLELP) the penetration $P(X_j)$ of segment JSG_j into plane JPL_j and the resulting normal component $F(X_j)$ of the specified force deflection function. Set $F1(X_j)$ equal to previous value of $F(X_j)$.

10. If $JX_j = 1$ (penetration has not been achieved for two steps) set $X_j = X_j + DX_j$, and if $P(X_j) > 0$ set $JX_j = 2$, and go to step 13.

11. If $JX_j = 2$ (penetration has been achieved for two consecutive steps)

    a) If $F(X_j)$ and $F1(X_j)$ are on opposite sides of $G(X_j)$, compute $DXP_j = DX_j/(F(X_j) - F1(X_j))$, set $JX_j = 3$, and go to step 12.

    b) If $F(X_j)$ is further than $F1(X_j)$ from $G(X_j)$, change sign of DX_j, set $F(X_j) = F1(X_j)$, $X_j = X_j + 2*DX_j$, and go to step 13.

    c) If either $F(X_j)$ or $F1(X_j)$ is zero, set $JX_j = 1$ if $F1(X_j) = 0$, set $F(X_j) = F1(X_j)$, halve DX_j, set $X_j = X_j - DX_j$, and go to step 13.

12. If $JX_j = 3$ (derivative DXP_j has been computed) and if $|F(X_j) - G(X_j)| < 10^{-6}$ (iteration has converged) go to step 15; otherwise, set $X_j = X_j - DXP_j*[F(X_j) - G(X_j)]$.

13. Place new value of X_j into $SEGLP_{i1,i2}$ if $NTV_j = 1$ or into $YPR_{i1,i2}$ if $NTV_j = 2$ for $i1 = NI1_j$ and $i2 = NSG_j$. Make any corresponding changes to the NAV_j associated variables. If $NTV_j = 2$, compute new direction cosine matrices (Subroutine DRCYPR) for all segments whose yaw, pitch or roll angles have been changed.

14. If the number of iterations required by step 7 have not been performed, go back to step 8.

15. If NPRT(27) $\neq$ 0, print new values of X_j and $F(X_j)$.

16. If sequence established in step 6 has not been completed, determine next value for j and go back to step 7.

17. Compute new values of RK2 originally supplied on Cards D.6 for any fixed point constraint (KQTYPE = 1) where the second segment No. (KQ2) is NVEH.

18. If NPRT(27) $\neq$ 0, print the solution of the system equations with the constraints (Cards G.6) off.

19. If NCON = 0, go to step 24. In this case, the input position parameters will be adjusted such that the normal component of the force deflection functions will be equal to the values of GX supplied on Cards G.5.

20. Set up the constraints specified on Cards G.6. Solve the system equations with the constraints on. If NPRT(27) $\neq$ 0 and, if this is the first step of the outer iteration, call Subroutine PRINT so that the user can ascertain his imposed constraints do indeed produce zero accelerations.

21. Fetch the normal component of the constraint forces and store them into GX(K) where K = INDGX(I), and store the tangential component of the constraint force into the TAB array used for the force deflection function computation for each of the user imposed constraints (I = 1 to NCON).

22. If all of the normal constraint forces differ by less than $\epsilon = 10^{-2}$ from the previous values of GX(K) for K = INDGX(I) for I = 1 to NCON, this iteration has converged, go to step 24.

23. If the number of iterations set up in step 5 have not been performed, go back to step 6.

24. Print out changes that have been made to the VISC (T1 on Cards B.5), RK2 (Cards D.6), SEGLP (Cards G.2), YPR (Cards G.3) and GX (Cards G.5) arrays and return to the calling program.

46. SUBROUTINE EULRAD (D,A,IC)

a. Purpose:

Called by Subroutine EJOINT to compute the Euler angles (precession, nutation and spin) in radians for a given direction cosine matrix D, resolving ambiguities such that the new angles differ the least from the given angles. (To use in a non-memory mode, supply all A's = 0 and IC = 8.)

b. Subroutines required:

None.

c. Labelled common blocks used:

CNSNTS.

d. Input or argument parameters:

D: Given 3x3 direction cosine matrix.
A: Array that contains the 3 Euler angles (precession, nutation and spin) in radians to be returned to calling routine.
IC: Interger (IEULER(J)) that indicates the lock-unlock condition of the Euler joint.

e. Optional output:

None.

f. Procedure:

Assumes $D = D(s)D(n)D(p)$, where Cx = Cos x, Sx = Sin x and

$$D(s) = \begin{pmatrix} Cs & Ss & 0 \\ -Ss & Cs & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad D(n) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & Cn & Sn \\ 0 & -Sn & Cn \end{pmatrix}, \quad D(p) = \begin{pmatrix} Cp & Sp & 0 \\ -Sp & Cp & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

47. SUBROUTINE EVALFD (D,N,L)

a. Purpose:

Evaluate derivative, function, or integral (as L = 0, 1, or 2) of the function that is defined at location N of TAB array for abscissa value D.

b. Subroutines required:

None.

c. Labelled common blocks used:

TABLES.

d. Input or argument parameters:

D: Abscissa value at which function is to be evaluated.
N: Location in the TAB array of function definition that was supplied on Cards E.1 - E.4 for the desired function.
L: Routine will evaluate value of derivative, function or integral as L = 0, 1 or 2.

e. Optional output:

None.

f. Procedure:

Routine tests magnitude and sign of D_0, D_1, and D_2 to determine type of function that is valid for argument D and returns for:

N = 0: Derivative of function at D.
N = 1: Value of function at D.
N = 2: Integral of function from 0 to D.

48. SUBROUTINE FDINIT

a. Purpose:

Called by Subroutines FINPUT and HINPUT to set up pointers in the NTAB array and initialize the storage in the TAB array for the five functions specified in the NF array from input Cards F.1.j, F.2.j, F.3.j, F.4.j, F.8.c and F.8.d1.

b. Subroutines required:

EVALFD.

c. Labelled common blocks used:

TABLES, TEMPVS.
(Note: this TEMPVS is shared by CINPUT, FINPUT, HINPUT and FDINIT.)

d. Input or argument parameters:

None (input and output data transmitted via COMMON /TEMPVS/).

e. Optional output:

None.

49. SUBROUTINE FINPUT

a. Purpose:

Called by Subroutine CINPUT to read and process input Cards F.1 to F.7 that specify the allowed contacts of body segments with vehicle panels, belts, airbags and other segments along with the associated funtions to be used for each contact, special joint forces and wind forces.

b. Subroutines required:

EVALFD.

c. Labelled common blocks used:

CONTRL, DESCRP, JBARTZ, TABLES, TITLES, CSTRNT, WINDFR, TEMPVS.
(Note: this TEMPVS is shared by CINPUT, FINPUT, HINPUT and FDINIT.)

d. Input or argument parameters:

Input Cards F.1 to F.7.

e. Optional output:

An annotated listing (menu) of all allowed contacts and functions to be used as derived from input Cards F.1 to F.7 is produced on the primary output unit.

f. Procedure:

1. Do the following for I to 4, in order, where

I = 1 represents plane vs. segment contacts,

I = 2 represents belt vs. segment contacts,

I = 3 represents segment vs. segment contacts, and

I = 4 represents global graphic joints.

(a) I = 1: IF NPL ≠ 0, set NJJ = NPL and read MNPL array on Card F.1.a.

I = 2: If NBLT ≠ 0, set NJJ = NBLT and read MNBLT array on Card F.2.a.

I = 3: Set NJJ = NSEG and read MNSEG array on Card F.3.a.

I = 4: If NJNT ≠ 0, set NJJ = NJNT and read IGLOB array on Card F.4.a.

(b) Do for every item J = 1 to NJJ,
set NK = MNPL(J), MNBLT(J), MNSEG(J) or IGLOB(J) as I = 1, 2, 3 or 4 which is the number of segments to contact the Jth element of the Ith type. If NK = 0, go to end of J loop.
Then for every item K = 1 to NK do the following.
(1) Read next card F.(I).(k) which contains

NJ - the contacted surface number

MS - an array of 3 integers containing the segment No. to which the contacted surface is attached, the segment number to be contacted and its associated ellipsoid No.

NF - an array of 5 integers specifying the 5 function numbers to be used.

(2) Print the contents of the input card and, on the following line, the corresponding surface, segment and function titles.

(3) Place MS into MPL(K,J), MBLT(K,J) or MSEG(K,J) as I = 1, 2, or 3 and place NT, the index of next element in the NTAB array, into NTPL(K,J), NTBLT(K,J), NTSEG(K,J) or IGLOB(J) as I = 1, 2, 3 or 4.

(4) Set up the pointers to the NTAB array and initialize storage in the TAB array for the functions specified by NF by calling Subroutine FDINIT.

2. If NJNTF ≠ 0, read and print input Cards F.5 defining the joint functions to be used.

3. If NBAG ≠ 0, read and print input Cards F.6.a.- F.6.n.

4. If NWINDF ≠ 0, read and print input Cards F.7 defining the input for Subroutine WINDY.

50. SUBROUTINE FLXSEG

a. Purpose:

Called by Subroutine DAUX if NFLX ≠ 0 to compute the submatrices B42 and V4 of the system equations for the flexible elements defined on input Cards B.7.

b. Subroutines required:

CROSS, DOTT33, DOT31, DOT33, DRCYPR, ELTIME, MAT31, MAT33, XDY.

c. Labelled common blocks used:

CONTRL, SGMNTS, FLXBLE, CNSNTS, TEMPVS.

d. Input or argument parameters:

None.

e. Optional output:

None.

51. FUNCTION FNTERP (THETA,PHI,NT)

a. Purpose:

Called by Subroutines VISPR if JOINTF(J) ≠ 0 to compute the restoring torque for joint No. J as a function of the flexure angle (THETA) and the azimuth angle (PHI) using the two dimensional table supplied on input Cards E.7 designated by NT = JOINTF(J).

b. Subroutines required:

None.

c. Labelled common blocks used:

CNSNTS, TABLES.

d. Input or argument parameters:

THETA: Flexure angle (radians) of the joint.
PHI: Azimuth angle (radians) of the joint.
NT: Index to the NTI array element that contains a pointer to the TAB array where the tabular data of the function is stored.

e. Optional Output:

None.

f. Procedure:

Subroutine evaluates

G1(THETA) = F(THETA,PHI(I))

and G2(THETA) = F(THETA,PHI(I+1))

for PHI(I) < PHI < PHI(I+1)

by linear interpolation or polynomial evaluation and then linear interploates between G1 and G2 to obtain F(THETA,PHI). If F < 0, then F is set to zero to establish a "dead band" defined by negative values in the table.

52. SUBROUTINE FRCDFL (D,RATE,M,N,FRCDF,ELOSS)

a. Purpose:

Evaluates the force deflection function established by Subroutine UPDFDC at point D where definition of function is controlled by M index of NTAB array. N = 0, 1 or 2 indicates value of derivative, function or integral is returned.

b. Subroutines required:

EVALFD.

c. Labelled common blocks used:

TABLES.

d. Input or argument parameters:

D: Abscissa value (d) at which the force deflection function is to be evaluated.

RATE: Time derivative of D (d').

M: Index of NTAB array which contains pointers to TAB array of function definition.

N: 0, 1 or 2 indicates value of derivative, function or integral is to be returned.
[Note: value of integral not currently required by program and is not operative in routine.]

FRCDF: Final evaluation of the force deflection function returned to the calling subroutine.

ELOSS: Value of the energy loss encountered and returned to the calling subroutine if rate dependent functions are used.

e. Optional output:

None.

f. Procedure:

Assumes: $0 \leq d_{quad} \leq d_{cubic} \leq d_{ref} \leq d_{max}$

1. If $d \geq d_{max}$, returns
   (a) N = 0: F = 0
   (b) N = 1: $F = F(d_{max})$

2. If $d_{ref} \leq d \leq d_{max}$, use base function
   (a) N = 0: $F = F'_{base}(d)$
   If inertial spike exists, $F = F + F'_{iner}(d)$
   (b) N = 1: $F = F_{base}(d)$
   If inertial spike exists, $F = F + F_{iner}(d)$

3. If $d_{cubic} \leq d \leq d_{ref}$, use cubic reloading function
   (a) N = 0: $F = F'_{cubic}(d)$
   (b) N = 1: $F = F_{cubic}(d)$

4. If $d_{quad} \leq d \leq d_{cubic}$, use quadratic unloading function
   (a) N = 0: $F = F'_{quad}(d)$
   (b) N = 1: $F = F_{quad}(d)$

5. If $d \leq d_{quad}$, then F = 0 is returned.

6. If either NTAB(M+2, 3 or 4) is negative, they define rate dependent functions F_2, F_3 and F_4 and
   $$F = F(d,d') = F(d) + F_2(d)F_3(d') + F_4(d')$$
   is evaluated and returned as FRCDF.

53. SUBROUTINE FSMSOL (C,R,NN,MM,MAXN,JN,MAXDIM)

a. Purpose:

Solves a set of 3*MM simultaneous equations (CX = R) where the matrix of coefficients consists of 3x3 submatrices stored in C(3,3,IJ). The location of the IJ submatrix is stored in NN(I,J), i.e., IJ = NN(I,J). The C matrix is destroyed and its size may be increased during the solution. The solution X is returned in R.

b. Subroutines required:

ELTIME.

c. Labelled common blocks used:

None.

d. Input or argument parameters:

C(3,3,MAXN)	: Given array of coefficients.
R(3,MM)	: Given right hand side.
NN(MM,MM)	: Array containing location of I,J elements.
MM	: Size of system of submatrices.
MAXN	: Largest value in NN array, number of submatrices in C array.
JN	: 1st dimension of NN in calling program.
MAXDIM	: The maximum number of elements for the C array allocated by the calling routine.

e. Optional output:

None. However if MAXN exceeds MAXDIM a complete print out of the C, R and NN arrays is produced and the program is terminated.

f. Procedure:

The C array is considered to be composed of 3x3 submatrices whose location of nonzero elements are noted in the NN array. A Guassian elimination procedure is used starting at the lower right corner of the C array. Information in the NN array allows the routine to process only the nonzero 3x3 submatrices and thus takes advantage of the sparseness of the C array. A negative integer in the NN array implies that the corresponding element in the C array is an identity submatrix and that the right side is zero, this will occur only for a diagonal element of NN. During the solution, the C array is destroyed, and it may be necessary to add elements to C and the corresponding elements in NN while MAXN is incremented. The final answer is returned to the calling routine in the R array.

54. SUBROUTINE GLOBAL (J,HD3,DH1,TQC,T9,ANGL)

a. Purpose:

Called by Subroutines VISPR (if IPIN(J) = 3) and EJOINT (if IGLOB(J) is not zero) to compute the torques applied if the joint is into the globalgraphic joint stop.

b. Subroutines required:

FRCDFL, HERRON.

c. Labelled common blocks used:

DESCRP, TABLES, TEMPVI, CNSNTS.

d. Input or argument parameters:

J: The joint identification number.
HD3: Reference vector in rotated segment.
DH1: Reference vectors defining coordinate system.
TQC: Torque magnitude.
T9: Torque vector.
ANGL: Flexure angle.

e. Optional output:

None.

55. SUBROUTINE HBELT (J1,J2,KNL0,IND)

a. Purpose:

Called by Subroutines CONTCT and HPTURB to compute the interactions of the harness-belt systems with their contacting body segment points and the resulting forces and torques that are added to the U1 and U2 arrays.

b. Subroutines required:

CROSS, DOT31, ELTIME, FRCDFL, MAT31.

c. Labelled common blocks used:

CNTSRF, SGMNTS, TABLES, FORCES, HRNESS, TEMPVS.
(Note: this TEMPVS is shared by HBELT, HBPLAY, HPTURB and HSETC.)

d. Input or argument parameters:

J1,J2: First and last index for belts.
KNL0: Zero value for KNL index.
IND: 0 - Call is from Subroutine CONTCT.
1 - Call is from Subroutine HPTURB.

e. Optional output:

The strain and forces at each belt endpoint are stored in the BSF array for output to the tabular time histories.

56. SUBROUTINE HBPLAY

a. Purpose:

Called by Subroutines INITAL and HPTURB to compute the current points in play for the harness-belt systems and the reference lengths between these points for the upcoming integration interval.

b. Subroutines required:

DOT31, MAT31.

c. Labelled common blocks used:

CONTRL, CNTSRF, SGMNTS, HRNESS, TEMPVS.
(Note: this TEMPVS is shared by HBELT, HBPLAY, HPTURB and HSETC.)

d. Input or argument parameters:

None.

e. Optional output:

If the new set of points differ from the previous set, then a list of the new set of points and the distances between them is printed on the primary output unit.

57. SUBROUTINE HEDING (LINES,LPP)

a. Purpose:

Called by either Subroutine OUTPUT or POSTPR to print the heading information at the start of each new page of tabular time history output. If the call is from Subroutine OUTPUT (NPRT(4) = 0, 1 or 4 on input Card A.5), then the headings are written on FORTRAN output unit Nos. 21 and up. If the call is from Subroutine POSTPR (NPRT(4) = $\pm$2 or $\pm$3), then the heading and a full page of tabular data is written on the primary output unit for each page number.

b. Subroutines required:

None.

c. Labelled common blocks used:

CONTRL, JBARTZ, TITLES, FORCES, CNSNTS, RSAVE, DAMPER, HRNESS, TEMPVS.

d. Input or argument parameters:

LINES: Current line count for all pages.
LPP: Number of lines of tabular data per page.

e. Optional output:

If NPRT(4) = 0, 1 or 4, then headings are printed on FORTRAN output unit Nos. 21 and up.

If NPRT(4) = ±2 or ±3, then a heading and full page of tabular data are printed for each tabular time history on the primary output unit.

f. Procedure:

Each heading at the start of each page consists of:

1. DATE from input Card A.1.

2. The page number of the form NT.xx, where NT corresponds to the FORTRAN output unit No. if the call is from Subroutine OUTPUT. NT is initially set to 21 and is incremented by one for each page type. .xx is initially set to .01 and is incremented by .01 for each new page of NT. Therefore, all pages with the same NT will contain data for the same time history, and all pages with the same sub-page .xx will contain the same time points in the different NT time histories.

Note: a different order of the pages will be produced depending upon if the call if from Subroutine OUTPUT or from Subroutine POSTPR. Subroutine OUTPUT will produce the order pages 21.01, 21.02, ... , 21.nn; 22.01, 22.02, ... , 22.nn; ... ; NN.01, NN.02, ... , NN.nn. Subroutine POSTPR will produce the order pages 21.01, 22.01, ... , NN.01; 21.02, 22.02, ... , NN.02; ... ; 21.nn, 22.nn, ... , NN.nn.

3. COMENT (two lines) from input Cards A.1.b and A.1.c.

4. VPSTTL from input Card C.1.

5. BDYTTL from input Card B.1.

6. A complete description of the type of tables contained on the current page along with an annotated identification of each column for the tabular data to follow. The following tabular time histories are available:

(a) Selected output controlled by input Cards H.1 to H.7.
(b) Plane-segment contact forces specified by input Cards F.1.
(c) Belt-segment contact forces specified by input Cards F.2.
(d) Harness-belt forces for endpoints on input Cards F.8.d.
(e) Spring damper forces as specified by input Cards D.8.
(f) Segment-segment contact forces specified by input Cards F.3.
(g) Airbag contact forces as specified by input Cards F.6.

58. SUBROUTINE HERRON (HD3,NT1,THETO,THETOP)

a. Purpose:

Called by Subroutine GLOBAL to compute the angle (THETO) and it derivative (THETOP) of the joint stop of a globalgraphic joint.

b. Subroutines required:

EVALFD.

c. Labelled common blocks used:

TABLES, CNSNTS.

d. Input or argument parameters:

HD3: The 3 components of a vector defining PHI.
NT1: Index to the TAB array defining the Herron function.
THETO: Angle (radians) of the joint stop as a function of PHI returned to GLOBAL.
THETOP: Derivative of THETO with respect to PHI returned to GLOBAL.

e. Optional output:

None.

59. SUBROUTINE HICCSI (NPTS)

a. Purpose:

Called by Subroutine POSTPR (if either JDTPTS(1) or (2) is not zero on input Card H.8) to compute the head injury criteria (HIC), head severity index (HSI) and chest severity index (CSI) from the resultant head and chest accelerations data that was read from output unit no. 8.

b. Subroutines required:

None.

c. Labelled common blocks used:

CDINT (Note: the normal CDINT has been overwritten by POSTPR).

d. Input or argument parameters:

NPTS: The number of resultant accelerations that have been read from output unit no. 8 by Subroutine POSTPR.

e. Optional output:

The computed values for HIC, HSI and HSI and related data are printed on the primary output unit.

f. Procedure:

The standard formulae for HIC and SI are computed by using an unequal step size trapozoidal integration on the unequally spaced time points available.

60. SUBROUTINE HINPUT

a. Purpose:

Called by Subroutine CINPUT to perform the input of Cards F.8.a to F.8.d that contain the setup and control information and performs the initialization of pointers to program tables used for the harness-belt systems.

b. Subroutines required:

FDINIT, XDY.

c. Labelled common blocks used:

CONTRL, CNTSRF, CNSNTS, TABLES, TITLES, HRNESS, TEMPVS.
(Note: this TEMPVS is shared by CINPUT, FINPUT, HINPUT and FDINIT.)

d. Input or argument parameters:

Input Cards F.8.a to F.8.d.

e. Optional output:

An annotated listing of the contents of input Cards F.8.a to F.8.d is produced on the primary output unit.

61. SUBROUTINE HPTURB

a. Purpose:

Called by Subroutine UPDATE to compute the perturbations of the points in play and belt reference points of the harness-belt systems for the upcoming integration interval.

b. Subroutines required:

DOT31, ELTIME, FSMSOL, HBELT, HBPLAY, HSETC, MAT31, XDY.

c. Labelled common blocks used:

CONTRL, CNSNTS, CNTSRF, SGMNTS, RSAVE, HRNESS, TEMPVS.
(Note: this TEMPVS is shared by HBELT, HBPLAY, HPTURB and HSETC.)

d. Input or argument parameters:

None.

e. Optional output:

NPRT(28) controls the frequency and level of diagnostic harness-belt forces output produced. Values of 0, 1, 2 and 3 are allowed as follows (each value includes output of all lower values).

0: Produces a table of the final harness-belt forces at each point in play at the same time points as output is produced by Subroutine PRINT as specified by NPRT(3).

1: Prints a table of the final harness-belt forces at each point in play at each time point of Subroutine HPTURB.

2: Prints a table of the harness-belt forces at each point in play for every iteration step of Subroutine HPTURB.

3: Prints the RHS, IJK and C arrays before the call to FSMSOL at each iteration step of each time point of HPTURB.

62. SUBROUTINE HSETC (NPTS,KHO,KNLO,NTP,IJ)

a. Purpose:

Called by Subroutine HPTURB for each individual belt segment of a harness-belt system to compute the coefficients of the constraint equation for each point in play and store them in the C and RHS arrays to be later solved by Subroutine FSMSOL.

b. Subroutines required:

DOT31, FRCDFL, MAT31.

c. Labelled common blocks used:

SGMNTS, TABLES, HRNESS, TEMPVS.
(Note: this TEMPVS is shared by HBELT, HBPLAY, HPTURB and HSETC.)

d. Input or argument parameters:

NPTS: Number of point in play for given belt segment.
KHO: Zero value for KH index (points in play for given harness).
KNLO: Zero value for KNL index (all points in play).
NTP: Number of tie points.
IJ: Running count or index of number of 3x3 submatrices in the C array.

e. Optional output:

None.

63. SUBROUTINE IMPLS2 (MODE,J,H)

a. Purpose:

Called by Subroutine UPDATE whenever joint No. J locks to apply an impulse by setting $P \cdot (D'(m)\omega(m) - D'(n)\omega(n)) = 0$.

b. Subroutines required:

DAUX, DOT31, DOT33, DSMSOL, ELTIME, PRINT, XDY.

c. Labelled common blocks used:

CONTRL, SGMNTS, DESCRP, CMATRX, CSTRNT, FLXBLE, TEMPVS.

d. Input or argument parameters:

MODE = 0: Full lock, P = I
= 1: Axis (H) free, P = I-HH'
=-1: Axis (H) locked, P = HH'
J: Joint identification number.
H: 3 component axis vector.

e. Optional output:

If NPRT(3) ≠ 0, then Subroutine PRINT is called.

64. SUBROUTINE IMPULS (I1,I2,I3)

a. Purpose:

Solves again the system of equations, but only with those external forces and torques caused by an impulsive force of the first type, to compute linear and angular accelerations produced by the impulse and modify the linear and angular velocities for the system variables.

b. Subroutines required:

CROSS, DAUX, DOT31, EJOINT, ELTIME, MAT31, OUTPUT, PLELP, PRINT, SEGSEG, VISPR.

c. Labelled common blocks used:

CONTRL, SGMNTS, CMATRX, DESCRP, FLXBLE, JBARTZ, CSTRNT, TABLES, TEMPVI.

d. Input or argument parameters:

I1 = 1: Impulse for PLELP
3: Impulse for SEGSEG
4: Impulse for VISPR or EJOINT
I2: Index of contacting segment.
I3: Plane, segment or joint number.

e. Optional output:

If NPRT(10) ≠ 0: Diagnostic output is produced and Subroutine OUTPUT is called.

If NPRT(3) ≠ 0: Subroutine PRINT is called after the impulse has been applied.

f. Procedure:

1. Set U_1, U_2, V_1 and V_2 arrays to zero.

2. Calls Subroutine PLELP, SEGSEG, VISPR or EJOINT for the single contact for the impulse. These routines have been modified to store data in COMMON/TEMPVI/ to be available for Subroutine IMPULS.

3. Subroutine DAUX is then called with a non-zero argument to set up and solve the system of equations. The non-zero argument to DAUX indicates that it should not call Subroutines SETUP, CHAIN, CONTCT, VISPR and EJOINT as in a normal call from PDAUX. It returns the values of $\ddot{x}$ (SEGLA) and $\dot{\omega}$ (WMEGD) that are the result of the impulse.

4. The normal component of relative velocity (V_{rel}) and incremental velocity due to unit impulse (δV) are computed by:

$$V_{rel} = V_1 + D_1^{-1}\omega_1 x r_1 - V_2 - D_2^{-1}\omega_2 x r_2$$

$$\delta V = V_1 + D_1^{-1}\dot{\omega}_1 x r_1 - V_2 - D_2^{-1}\omega_2 x r_2$$

or in the case of angular velocity for joints,

$$V_{rel} = D_1^{-1}\omega_1 - D_2^{-1}\omega_2$$

$$\delta V = D_1^{-1}\dot{\omega}_1 - D_2^{-1}\dot{\omega}_2$$

and then

$$\alpha = -(\mu + 1)\frac{t \cdot V_{rel}}{t \cdot \delta V}$$

where t is a vector representing the direction of the force applied by Subroutine PLELP, SEGSEG, VISPR or EJOINT

and $\mu = 2e - 1$.

5. Final corrections to all segment linear and angular velocities by:

$$\dot{x}_i = \dot{x}_i + \alpha \ddot{x}_i$$

$$\omega_i = \omega_i + \alpha\dot{\omega}_i$$

6. Subroutine PRINT is then called to give an indication that the impulse has occurred and an output of the result of the impulse.

65. SUBROUTINE INITAL

a. Purpose:

Called by the main program to perform the input and computation for initial positioning of the crash victim's body segments.

b. Subroutines required:

CHAIN, CROSS, DOT31, DRCIJK, ELTIME, EQUILB, HBPLAY, MAT31, ROTATE, VEHPOS, YPRDEG.

c. Labelled common blocks used:

CONTRL, DESCRP, VPOSTN, TITLES, SGMNTS, CNSNTS, TEMPVS.

d. Input or argument parameters:

Input Cards G.1, G.2 and G.3.

e. Output:

A table of the initial linear and angular positions and velocities is printed on the primary output unit.

f. Procedure:

1. Input Card G.1.a, the plot coordinates for Subroutine PRIPLT of the vehicle reference system origin. Also supply I3, an indicator of whether or not the segment velocities are to be supplied on Cards G.2. If J1 $\neq$ 0 on Card G.1.a, then read input Card G.1.b and recompute the scaling parameters for Subroutine PRIPLT for printers that differ from the default

values of 10 characters per inch and 6 lines per inch.

2. Read input Cards G.2 that contain the linear position and, if I3 = 1, the linear velocity of each reference body segment. A segment is a reference segment if it is the first segment or if JNT(I-1) $\leq$ 0. If I3 = 0 on Card G.1.A, then the linear velocity of each reference segment is set equal to that of the vehicle.

3. Read input Cards G.3 that contain the initial angular orientation and, if I3 = 1, angular velocity in local reference for all segments J for J = 1 to NSEG. Also on these cards are the indicators IYPR that identify the manner in which the angular orientations are supplied; if they are zero, they are set to default values of 1, 2 and 3 to indicate that the normal yaw, pitch, roll sequence is to be done in the reverse order (to be compatible with older versions of the CVS program). If IYPR(1,J) is negative, the angular rotations are to be computed by projections of the segment axes and an input Card G.3.j2 is read for segment No. J and corresponding yaw, pitch, roll angles are computed. The initial direction cosine matrix for segment No. J is then computed by calling Subroutine DRCIJK and, if I3 = 0, the initial angular velocity of the vehicle is converted to the local reference system and used as the initial angular velocity of segment No. J.

4. Subroutine VEHPOS and CHAIN are called to complete the computations of the initial positions of all segments and a complete table is printed on the primary output unit.

5. The final program initialization is performed by
   (a) calling Subroutine HBPLAY if NHRNSS $\neq$ 0 on input Card D.1
   (b) calling Subroutine EQUILB if I1 = 15 on input Card G.1.a
   (c) calling Subroutine ROTATE.

66. SUBROUTINE INTERS (A,B,M,T,X,V,AX)

a. Purpose:

Determines the intersection of ellipsoids A and B, where

$$X'AX = 1$$
$$(X-M)'B(X-M) = 1$$

Subroutine returns: $T > 1$ for no intersection

$T \leq 1$ ellipsoids intersect, in which case X will be the point of contact of contracted ellipsoids.

b. Subroutines required:

DSMSOL, MAT31.

c. Labelled common blocks used:

CNSNTS.

d. Input or argument parameters:

A: 3x3 matrix describing ellipsoid A.

B: 3x3 matrix describing ellipsoid B.

M: Vector from the center of A to the center of B.

T: The expansion or contraction factor applied equally to both ellipsoids A and B so that they intersect at a single point ($T < 1$ for contraction, $T > 1$ for expansion).

X: The single contact point returned to the calling routine if $T \leq 1$.

V: Memory (previous final value) for initial value of V if $V \neq 0$.

AX: Vector Ax returned to calling routine.

e. Optional output:

None.

f. Procedure:

1. Initialization

evaluate Bm,m'Am,m'Bm

set initial v = m'Bm/m'Am if argument V = 0.

2. Start of Newton-Raphson iteration for

$$G(v) = F_a(v) - F_b(v)$$

3. Solve $(vA + B)x = Bm$ for x

4. Evaluate Ax

$F_a(v) = x'Ax$

$F_b(v) = -v(x-m)'Ax$

5. If $F_a > F_b > 1$ (no intersection), proceed to step 9.

   If $F_a \leq F_b \leq 1$, then intersection exists.

6. Solve $(vA + B)z = Ax$ for z.

7. Evaluate

$$F'_a(v) = -2x'Az$$

$$dv = \frac{F_b(v) - F_a(v)}{(v+1)F_a(v)}$$

8. Test for convergence

$$v = v + dv$$

   if $\left|\frac{dv}{v}\right| > 10^{-12}$, go back to step 3.

9. Evaluate $T = [F_a(v)]^{1/2}$ and return to calling routine.

67. SUBROUTINE KINPUT

a. Purpose:

Called by Subroutine CINPUT to read the input Cards E.6 and E.7 that define the functions to be used for wind force and joint restoring force functions.

b. Subroutines required:

None.

c. Labelled common blocks used:

CONTRL, CNSNTS, TABLES, TEMPVS.

d. Input or argument parameters:

Input Cards E.6 and E.7.

e. Optional output:

An annotated listing of input Cards E.6 and E.7 is produced on the primary output unit.

68. SUBROUTINE LINAXS (XO,YO,THETA,NINTVS,TOTLGT)

a. Purpose:

Called by Subroutine SLPLOT to prepare a linear axis on a plot.

b. Subroutines required:

SIN, COS (single precision), PLOT.

c. Labelled common blocks used:

None.

d. Input or argument parameters:

XO,YO: Starting point (inches) coordinates of axis relative to current plotter origin.

THETA: Angle (degrees) measured counter-clockwise from a horizontal line extending to the right from XO,YO.

NINTVS: Integer whose magnitude indicates the number of intervals that are delineated by tic marks and whose sign determines whether the tic marks are placed on the positive or negative side of axis (positive side is to the left of direction of travel).

TOTLGT: Total length (inches) of linear axis.

e. Optional output:

None.

69. SUBROUTINE LOGAXS (XO,YO,THETA,NDEC,EXTENT)

a. Purpose:

Called by Subroutine SLPLOT to prepare a logarithmic axis on a plot.

b. Subroutines required:

SIN, COS (single precision), PLOT.

c. Labelled common blocks used:

None.

d. Input or argument parameters:

XO,YO: Starting point (inches) coordinates of axis relative to current plotter origin.

THETA: Angle (degrees) measured counter-clockwise from a horizontal line extending to the right from XO,YO.

NDEC: Integer whose magnitude indicates the number of logarithmic decades to be plotted on the axis and whose sign determines whether the tic marks are placed on the positive or negative side of axis (positive side is to the left of direction of travel).

EXTENT: Magnitude designates the total length (inches) of logarithmic axis and sign determines whether the tic marks are spaced in the normal order (positive - large intervals first) or in reverse order (negative - small intervals first).

e. Optional output:

None.

f. Procedure:

Major tic marks are prepared to delineate each decade and minor tic marks are prepared at every 0.5 times current power of ten intervals.

70. FUNCTION LTIME (N)

a. Purpose:

This FORTRAN routine replaces a S/370 Assembler Language routine that measures CPU elapsed time in units of 0.01 seconds. It should be replaced by the user to enable Subroutine ELTIME to perform properly on his computer.

b. Subroutines required:

System timing macros.

c. Labelled common blocks used:

None.

d. Input or argument parameters:

1. IT = LTIME(0) gives elapsed CPU time (integer number of 0.01 second units) since subroutine reference was reset, and resets this reference.

2. IT = LTIME(1) same, except that the reference is not reset.

e. Optional output:

None.

f. Procedure:

FORTRAN version merely counts the number of calls to LTIME as follows:

```
FUNCTION LTIME(N)
DATA KTIME /0/
KTIME = KTIME + 1
LTIME = KTIME
IF (N.EQ.0) KTIME = 0
RETURN
END
```

71. SUBROUTINE MAT31 (A,B,C)

a. Purpose:

Performs the matrix multiplication C = AB, where A is a 3x3 matrix, and B and C are vectors with 3 components.

b. Subroutines required:

None.

c. Labelled common blocks used:

None.

d. Input or argument parameters:

A: Matrix of size 3x3.
B: Vector with 3 components.
C: Product vector with 3 components.

e. Optional output:

None.

f. Procedure:

Each element c_i of the product vector C(3) is computed by

$$c_i = \sum_{j=1}^{3} a_{ij} b_j \quad \text{for } i = 1 \text{ to } 3.$$

72. SUBROUTINE MAT33 (A,B,C)

a. Purpose:

Performs the matrix multiplication C = AB, where A, B and C are all 3x3 matrices.

b. Subroutines required:

None.

c. Labelled common blocks used:

None.

d. Input or argument parameters:

A,B: Matrices of size 3x3.
C: Product matrix of size 3x3.

e. Optional output:

None.

f. Procedure:

Each element c_{ij} of the product matrix C(3,3) is computed by

$$c_{ij} = \sum_{k=1}^{3} a_{ik} b_{kj} \quad \text{for both i and j = 1 to 3.}$$

73. SUBROUTINE ORTHO (P,X,L)

a. Purpose:

Generates a set of right-handed orthonormal vectors, given one of the vectors.

b. Subroutines required:

None.

c. Labelled common blocks used:

None.

d. Input or argument parameters:

P: Lx3 matrix of 3 right handed orthonormal vectors to be generated.
X: Given vector.
L: 1st subscript of P in calling routine.

e. Optional output:

None.

f. Procedure:

The given vector X is placed in P(I,3) and the two orthogonal vectors are computed and placed into P(I,1) and P(I,2) for I = 1 to 3. It is assumed that any nonzero column of the matrix $I-XX^T$ is orthogonal to X. The column of the largest diagonal element is selected as the first vector, the second vector is computed by $p_2 = p_1 x p_3$ and X is used as the third vector. If X has only one nonzero element, the other two vectors will have the same property.

74. SUBROUTINE OUTPUT (IJK)

a. Purpose:

Controls output of the selected optional segment linear and angular accelerations, velocities and displacements, joint parameters and selected data from all allowed contacts between body segments and vehicle components. This output is either printed on FORTRAN o RSAVE, COMAIN, DAMPER, HRNESS.

b. Subroutines required:

CROSS, DOT33, DOT31, DOT33, ELTIME, HEDING, MAT31, YPRDEG.

c. Labelled common blocks used:

CNSNTS, CONTRL, FORCES, SGMNTS, JBARTZ, TEMPVS, SGMNTS, TITLES, RSAVE, COMAIN, DAMPER, HRNESS.

d. Input or argument parameters:

1. Input Cards H.1 to H.7.
2. Arguments

IJK = 0: zeros print arrays at start of a time step computation.
= 1: performs all desired output at the termination of a successful integration step.

e. Optional output:

If NPRT(4) = 0, 1 or 4, all output is written on FORTRAN output unit Nos. 21 and up.

If NPRT(4) = 1, 2, 3 or 4, all output is written on FORTRAN output unit No. 8 (unformatted) to be processed by Subroutine POSTPR.

f. Procedure:

1. If IJK = 0, zero out all print arrays and return to calling program.

2. If this is the first time in the routine, read input cards that selects optional output as follows:

k = 1: Input Cards H.1.a - H.1.n, one card for each point on the crash victim body containing the segment number and the location of the point on the segment for which tables of segment linear accelerations are desired.

k = 2: Input Cards H.2.a - H.2.n, same as above, but for segment linear accelerations.

k = 3: Imput Cards H.3.a - H.3.n, same as above, but for segment linear displacements.

k = 4: Input Card H.4 containing segment numbers for which tables of segment angular accelerations are desired.

k = 5: Input Card H.5, same as above, but for segment angular velocities.

k = 6: Input Card H.6, same as above, but for segment angular displacements.

k = 7: Input Card H.7 containing the joint numbers for which tables of joint computations are desired.

3. Increment line count by 1 and, if this line represents the start of a new page and if NPRT(4) = 0, 1 or 4, then call Subroutine HEDING.

4. Process one line of output on each unit (NT) as follows:

(a) Selected optional output as controlled by input Cards H.1 to H.7, by processing the components and resultant of:

(1) Segment linear acceleration for each desired point r_j in local segment reference by:

$$D_j z_{r_j} = D_j z_j + \dot{w}_j x r_j + w_j x (w_j x r_j)$$

(2) Segment linear velocity for each desired point r_j in vehicle reference by:

$$D_v(\dot{z}_{r_j} + \dot{z}_v) = D_v(\dot{z}_j + D_j' w_j x r_j - \dot{z}_v)$$

(3) Segment linear displacement for each desired point r_j in vehicle reference by:

$$D_v(z_{r_j} + z_v) = D_v(z_j + D_j' r_j - z_v)$$

(4) Segment angular decelerations for each desired segment j in local segment regerence, w_j.

(5) Segment angular velocities for each desired segment j in vehicle reference by:

$D_v D'_j w_j - w_v$

(6) Segment angular displacement for each desired segment j in vehicle reference by obtaining yaw, pitch and roll from:

$D_j D_v$

(7) Joint parameters for each desired joint j by printing values from PRJNT array.

(b) For each allowed plane-segment contact:

(1) Deflection (inches)
(2) Normal force (lbs)
(3) Friction force (lbs)
(4) Resultant force (lbs)
(5) Contact location in vehicle reference.

(c) For each allowed belt-segment contact, the strain and force at each anchor point.

(d) For each harness-belt system, the strain and force at each belt end point.

(e) The results of all spring damper forces.

(f) For each allowed segment-segment contact:

(1) Deflection (inches)
(2) Normal force (lbs)
(3) Friction force (lbs)
(4) Resultant force (lbs)
(5) Contact location in each segment reference.

(g) For each air bag:

(1) Supply pressure (PSIG).
(2) Cylinder temperature (^{o}R).
(3) Static pressure (PSIG).
(4) Air bag center location in vehicle reference (inches).
(5) Length of air bag semiaxes (inches).
(6) Air bag angular displacement, yaw, pitch and roll (degrees).
(7) Components of and resultant force of the simulated spring at the deployment point, each reaction panel and each segment contacting the air bag.

5. If NPRT(4) equals 1, 2, 3 or 4, then store all of the above data for this time point on FORTRAN output unit No. 8.

75. SUBROUTINE PANEL (DRR,ZR,JB)

a. Purpose:

Called by Subroutines AIRBG1 and AIRBG3 to compute the position parameters of the airbags during inflation.

b. Subroutines required:

CROSS, DOT31, MAT31.

c. Labelled common blocks used:

CONTRL, SGMNTS.

d. Input or argument parameters:

DRR: Direction cosine matrix of airbag relative to vehicle.
ZR: Location of airbag ellipsoid center in vehicle reference.
JB: Segment identification number for the airbag.

e. Optional output:

None.

f. Procedure:

$$D_{jb} = D_{rr} D_v$$

$$w_{jb} = D_{rr} w_v$$

$$\dot{w}_{jb} = \dot{w}_v$$

$$z_{jb} = z_v + D_v' z_r$$

$$\dot{z}_{jb} = \dot{z}_v + D_v' w_v x z_r$$

$$\ddot{z}_{jb} = \ddot{z}_v + D_v' w_v x w_v x z_r$$

76. SUBROUTINE PDAUX (VAR,DER,NEQ,KDINT)

a. Purpose:

This subroutine is to act as an interface between the integrator Subroutine DINT and the derivative evaluater Subroutine DAUX to accommodate a variable number of functions to be integrated.

b. Subroutines required:

ELTIME, DSETQ, DAUX, CROSS.

c. Labelled common blocks used:

CONTRL, DESCRP, SGMNTS, TITLES, INTEST, FLXBLE, TEMPVS.

d. Input or argument parameters:

VAR : Array of NEQ state variables to be updated by DINT.
DER : Array of NEQ derivatives to be supplied by DAUX.
NEQ : Number of state variables and derivatives.
KDINT : Integrator step number in DINT.

e. Optional output:

None.

f. Procedure:

1. If KDINT $\leq$ 0, implies that this is an initialization call from DINT.

PDAUX is to supply values to the state variables and compute value of NEQ. In successive cells of XTEST (control of integrator convergence) and VAR are stored the following:

(a) 3*NSEG values for quaternions

XTEST(I,M) = SGTEST(I,1,M)**2

VAR(I,M) = 0.0

for I = 1 to 3 and M = 1 to NSEG.

(b) 3*K values of segment linear positions (SEGLP) for each of K reference segments

XTEST(I,N) = SGTEST(I,2,M)**2

VAR(I,N) = SEGLP(I,M)

for I = 1 to 3, M = 1 and any M for which JNT(M-1) $\leq$ 0, and N is a running index of items to be stored.

(c) 3*NSEG values of angular velocities (WMEG)

XTEST(I,N) = SGTEST(I,3,M)**2

VAR(I,N) = WMEG(I,M)

for I = 1 to 3 and M = 1 to NSEG.

(d) 3*K values of segment linear velocities (SEGLV) for each of K reference segments

XTEST(I,N) = SGTEST(I,4,M)**2

VAR(I,N) = SEGLV(I,M)

for I = 1 to 3 and M = 1 and any M for which JNT(M-1) $\leq$ 0.

NEQ is then set equal to 3*N which therefore equals 6*(NSEG+K) and control is passed to step No. 5.

2. If KDINT=1, DINT is executing the first step in advancing the integration interval, save the direction cosing matrices for the start of an integration step in the array SD, and proceed to step 5.

3. If KDINT ≠ 0 or 1, fetch saved direction cosine matrices and update by current values of (in the first 3*NSEG elements of VAR) by calling Subroutine DSETQ.

4. Store the remaining state variables from DINT into the program arrays as follows:

   (b) VAR(I,N) —> SEGLP(I,M) for reference segments.
   (c) VAR(I,N) —> WMEG(I,M) for all segments.
   (d) VAR(I,N) —> SEGLV(I,M) for reference segments.

5. Call Subroutine DAUX with an argument of zero to compute the derivatives SEGLA ($\ddot{z}$) and WMEGD ($\dot{w}$).

6. Store the derivatives of the state variables into successive elements of the DER array for the integrating routine DINT as follows:

   (a) The derivative of Q for all segments.
   (b) Segment linear velocity (SEGLV) of reference segments.
   (c) Segment angular accelerations (WMEGD) of all segments.
   (d) Segment linear accelerations (SEGLA) of reference segments.

7. If KDINT=4, DINT is at the final step of an integration interval, set the quaternions equal to the identity.

8. Return to the calling program.

77. SUBROUTINE PLELP (M,L,N,J,NT)

a. Purpose:

Computes the forces and torques (which are added to the U1 and U2 arrays) produced by ellipsoid (L) attached to body segment (M) intersecting plane (J) attached to segment (N).

b. Subroutines required:

CROSS, DOTT31, DOT31, ELTIME, MAT31, PLSEGF.

c. Labelled common blocks used:

CNTSRF, CSTRNT, FORCES, SGMNTS, TABLES, TEMPVS.

d. Input or argument parameters:

M: Segment identification number.
L: Contact ellipsoid number that is attached to segment M.
J: Plane identification number.
N: Segment identification number to which plane J is attached.
NT: Index to NTAB array containing pointers to the function definitions for this plane-segment contact.

e. Optional output:

Plane-segment contact force data is stored in the PSF array for output to the tabular time histories.

f. Procedure:

1. Compute the penetration distance P and, if negative or if penetration is complete, return to the calling program.

2. Compute the point Tg in segment reference at which the contact forces are to be applied. The point lies on the scaled line between the point of maximum penetration ($\rho = 0$) and the center of the intersection ellipse ($\rho = 1$) where ρ is defined with the force deflection function.

3. Compute the distance from the ellipse center to the plane; if negative or greater than the extent of the plane, return to the calling program.

4. Call Subroutine PLSEGF to calculate the contact forces and torques.

5. If a force deflection function does not exist for this contact (as indicated by NTAB(NT+1) < 0), then a constraint exists for this contact and the following information is stored in COMMON/CSTRNT/ where NCF = -NTAB(NT+1) is the constraint number.

```
KQTYPE(NCF) = 3
RK1(NCF)    = Tg
RK2(NCF)    = Tg (in vehicle reference)
CFQQ(NCF)   = the coefficient of friction
TQQ(NCF)    = t (the normalized direction of the constraint force)
```

and control is returned to the calling routine.

78. SUBROUTINE PLSEGF (M,N,NT)

a. Purpose:

Performs the identical calculations required by Subroutine PLELP and SEGSEG, i.e., the calculation of forces and torques which are added to the U1 and U2 arrays for both segments M and N.

b. Subroutines required:

CROSS, DOT31, EVALFD, FRCDFL, MAT31.

c. Labelled common blocks used:

SGMNTS, CSTRNT, TEMPVI, TABLES, TEMPVS.

d. Input or argument parameters:
(Same as for calling Subroutine PLELP or SEGSEG)

M,N: Segment identification numbers to which planes on ellipsoids are attached.

NT: Index to NTAB array containing pointers to the function definitions for this plane-segment or segment-segment contact.

(Other input and output values are transmitted via COMMOM/TEMPVS/.)

e. Optional output:

None.

f. Procedure:

1. Compute the relative velocity of the point y of segment N with respect to segment M in M's reference by:

$$v_y = (D_m D_n^T w_n - w_m) x t_2 + D_m(\dot{\rho}_n - \dot{\rho}_m) - D_m D_n^T w_n x D_m(\rho_n - \rho_m)$$

and the tangential velocity in the plane normal to y by:

$$v_p = v_y - v_y \cdot \frac{A_y}{|A_y|}$$

2. Compute the normal force F_m by evaluating the force-deflection function for this contact by FUNCTION FRCDFL and the coefficient of friction C_f for this contact by FUNCTION EVALFD both as a function of S_1. Then evalutate the total force vector in the M's reference by:

$$F_t = F_m \frac{A_y}{|A_y|} - C_f F_m \frac{V_p}{|V_p|}$$

a linear ramp function is utilized for the friction force if $|V_p| < 1$.

3. Convert the total force to inertial references and add to elements of the U1 array for both the M and Nth segments by:

$$U1_m = U1_m - w_m D'_m F_t$$

$$U1_n = U1_n + w_n D'_n F_t$$

4. Convert the torque to local reference and add to the elements of the U2 array for both the M and Nth segments by:

$$U2_m = U2_m - \phi_m^{-1}(t_2 x F_t)$$

$$U2_n = U2_n + \phi_n^{-1} D_n D'_m (t_q x F_t)$$

79. SUBROUTINE PLTXYZ (P,C)

a. Purpose:

Stores plot character (C) in vehicle reference into the PLOTYZ, PLOTXZ the PLOTXY print arrays for point P given in inertial reference.

b. Subroutines required:

MAT31.

c. Labelled common blocks used:

CONTRL, SGMNTS, VPOSTN, TEMPVS.

d. Input or argument parameters:

P: Coordinates of point in inertial reference to be plotted.
C: Character or symbol to be used to depict point in vehicle reference.

e. Optional output:

None.

f. Procedure:

1. Convert point P from inertial reference to vehicle reference by:

$$Z = D_v(P - X_v)$$

2. Convert coordinates of point z to plot coordinates by:

$$I_j = S_j Z_j + O_j + 0.5 \quad \text{for } j = x, y \text{ and } z \text{ coordinates}$$

where S_j and O_j are the SPLT and ZPLT arrays defined on input Cards G.1.a and G.1.b.

3. If $1 \leq I_z \leq N_z$,

a) and, if $1 \leq I_y \leq N_x$, then store C into PLOTYZ(I_z,I_y)

b) and, if $1 \leq I_x \leq N_x$, then store C into PLOTXZ(I_z,I_x)

where N_z and N_x are the dimensions of the plot arrays.

4. Recompute $I_y = -S_z Z_y + O_y + 0.5$

and, if $1 \leq I_y \leq N_z$ and $1 \leq I_x \leq N_x$,

then store C into PLOTXY(I_y,I_x).

80. SUBROUTINE POSTPR (PRDT)

a. Purpose:

Called at the end of the main program (if NPRT(4) ≠ 0 or 4) to process the output unit No. 8 generated by the current or a previous run of the CVS program. Depending on the value of NPRT(4), HIC and CSI values are computed, tabular time histories are printed, and/or plots of any variables in the time histories are produced.

b. Subroutines required:

ELTIME, HEDING, HICCSI, SLPLOT.

c. Labelled common blocks used:

CONTRL, FORCES, TITLES, CNSNTS, JBARTZ, DAMPER, HRNESS, RSAVE, CDINT, TEMPVS. (Note: the normal CDINT is overwritten to store data for plotting and for the HIC and CSI computations. Also, TEMPVS intentionally overflows to overwrite all of the other labelled common blocks that are no longer needed by the CVS program to store a complete page of every time history.)

d. Input or argument parameters:

Input Card H.8 and (if NPRT(4) is odd) Cards I.1 to I.8.
PRDT: The value of DT (input Card A.4) converted to msec used here to control the time point intervals on the tabular time histories.

e. Optional output:

If either JDTPTS(1) or (2) on input Card H.8 is not zero, then Subroutine HICCSI is called.

If NPRT(4) is odd, then Calcomp plots of data from the tabular time histories are generated by calling Subroutine SLPLOT. (Note: all of the time points from output unit no. 8, not PRDT intervals, are plotted.)

If NPRT(4) is plus or minus 2 or 3, then tabular time histories are printed on the primary output unit at time intervals controlled by the value of NPRT(26) on input Card A.5.

f. Procedure:

The tabular time histories and plots will be generated depending on the value of NPRT(4) from input Card A.5 as follows:

Value of NPRT(4)	Time Histories	Plots
+4	**	No
+3	Yes	Yes
+2	Yes	No
+1	**	Yes
0	**	No
-1	No	Yes
-2	Yes	No
-3	Yes	Yes

** - Time histories were generated by Subroutine OUTPUT.

Note: A positive value for NPRT(4) is designed to process the output unit no. 8 from the current run, while a negative value will process one generated by a previous (or uncompleted) run.

81. SUBROUTINE PRINT (SUB)

a. Purpose:

Output routine to print in tabular form at a given time point the linear and angular positions, velocities and accelerations of the vehicle and all body segments; the final total values of the U_1 and U_2 arrays representing external linear and angular accelerations for all body segments; the forces, torques and relative angular velocity of all the joints; and constraint force information, if any.

b. Subroutines required:

YPRDEG, DOT31, DOT33.

c. Labelled common blocks used:

CONTRL, CEULER, SGMNTS, CMATRX, DESCRP, CSTRNT, TITLES, CNSNTS, RSAVE, TEMPVS.

d. Input or argument parameters:

SUB: Calling subroutine name (alphanumeric).

e. Optional output:

The output from this routine is printed on the primary output unit. Calls to this routine are controlled by nonzero values of the following print indicators:

NPRT(3) - Main Program, IMPLS2 and UPDATE
NPRT(9) - Subroutine DAUX
NPRT(10) - Subroutine IMPULS
NPRT(27) - Subroutine EQUILB

f. Procedure:

At every call to the routine, the following information is printed in tabular form on the primary output unit:

1. Name of calling routine and time (msec).

2. Components of all segment (including the vehicle and airbags, if any) angular rotation (deg), velocity (rad/sec) and acceleration (rad/sec^2) in local segment reference.

3. Components of all segment (including the vehicle and airbags, if any) linear position (in), velocity (in/sec) and acceleration (in/sec^2) in inertial reference.

4. Components of all segment external linear accelerations (U_1 array - in/sec^2) and angular accelerations (U_2 array - rad/sec^2).

5. Components of all joint forces (lb) and torques (in. lb) and the relative angular velocity (rad/sec).

6. For all constraints, if any,
   (a) Constraint number
   (b) Constraint type
   (c) Constraint force (lbs.)
   (d) Constraint distance (in.)

82. SUBROUTINE PRIPLT

a. Purpose:

A printer plot program to produce Y-Z, X-Z amd X-Z plane views of locations of the centers of gravity of the body segments, joints, restraint belt points and the air bag center and semiaxes end points.

b. Subroutines required:

DOT31, PLTXYZ, ELTIME.

c. Labelled common blocks used:

DESCRP, HRNESS, CONTRL, TEMPVS, JBARTZ, TITLES, CNTSRF, SGMNTS.

d. Input or argument parameters:

None.

e. Optional output:

1. NPRT(13) $\neq$ 0: Prints table of joint locations.

2. NPRT(5) $\neq$ 0: At a DT time step frequency determined by the value of NPRT(5), writes on output unit 2, TIME, the coordinates of the origin of the vehicle in inertial reference, and the array of alphanumeric plot characters depicting the y and z coordinates of the points computed below. Note that the plot coordinates are converted from inertial to vehicle reference by Subroutine PLTXYZ.

3. NPRT(6) ≠ 0: Same as above, except that x and z coordinates are plotted.

4. NPRT(7) ≠ 0: Same as above, except that x and y coordinates are plotted.

f. Procedure:

1. Determine if plotting is to be done for this time step by values of NPRT(5), NPRT(6), and NPRT(7).

2. The 60x120 arrays PLOTYZ, PLOTXZ and PLOTXY of alphanumeric characters are set to blanks.

3. Plot the vehicle reference origin using the symbol "*" and the center of gravity of body segments using symbols in the CGS array by calls to Subroutine PLTXYZ.

4. The locations of the joints are computed by

$$C_m = z_n + D_n^{-1} r_{n_j}$$

and plotted using symbols from the JS array by calls to Subroutine PLTXYZ.

5. If NBLT ≠ 0, the locations of the anchor points, fixed point and tangency points of the belts are transformed from local to inertial coordinates by:

$$T^{*} = D_{n}^{-1} T + z_{n}$$

and plotted using the symbol "." by calls to Subroutine PLTXYZ.

6. If NHRNSS ≠ 0, the locations for all points in play for the harness-belt systems are converted from local to inertial reference and plotted using the symbol "." by calls to Subroutine PLTXYZ.

7. IF NBAG ≠ 0, the center and semiaxes end points of the air bags are plotted using the symbol "@" for the center, the symbol "-" for the vertical semiaxes and the symbol "¦" for the horizontal semiaxes by calls to Subroutine PLTXYZ.

8. The contents of the PLOTYZ, PLOTXZ and PLOTXY arrays (as controlled by nonzero values of NPRT(4), NPRT(5) and NPRT(7), respectively) are transmitted to FORTRAN output unit No.2 to be printed as a secondary output.

83. SUBROUTINE QSET (F,Y,X,DER,N)

a. Purpose:

Called by Subroutine DINT at the end of each integration interval after the convergence tests have been successful to complete the computations for the quaternions stored as the first elements in the F and Y arrays.

b. Subroutines required:

None.

c. Labelled common blocks used:

None.

d. Input or argument parameters:

F,Y: The F and Y arrays from COMMON /CDINT/ but with the 240 dimension changed to 3,80 for indexing here.

X,DER: The VAR and DER arrays from COMMON /COMAIN/ but with the 240 dimension changed to 3,80 for indexing here.

N: The number of three component vectors at the beginning of the variable arrays that represent quaternions as computed by Subroutine PDAUX for KDINT = 2.

e. Optional output:

None.

84. FUNCTION RCRT (A,PL,Z,IP)

a. Purpose:

Computes the radius of curvature on ellipsoid A at the point z in the plane specified by PL(I,IP) for I = 1 to 3.

b. Subroutines required:

None.

c. Labelled common blocks used:

None.

d. Input or argument parameters:

A(3,3): Ellipsoid matrix.
PL(3,3): Array containing three orthonormal vectors.
Z(3): Point on ellipsoid as measured from the center of the ellipsoid.
IP: Identifies the normal vector of the plane in which the radius of curvature is desired.

e. Optional output:

None.

f. Procedure:

Let $z = xu_1 + yu_2 + au_3$, where u_1, u_2 and u_3 are the three orthogonal vectors and u_3 is the vector normal to the desired plane.

The equation of the ellipse in the plane is then

$$z^T Az = (xu_1 + yu_2 + au_3)^T A(xu_1 + yu_2 + au_3) = 1$$

Differentiation gives

$$\frac{dy}{dx} = \frac{u_1^T Az}{u_2^T Az} \quad \text{and} \quad \frac{d^2y}{dx^2} = \frac{(u_1 + \frac{dy}{dx}u_2)^T A(u_1 + \frac{dy}{dx}u_2)}{u_2^T Az}$$

The radius of curvature ρ is the given by

$$\rho = \frac{\left[1 + \left(\frac{dy}{dx}\right)^2\right]^{3/2}}{\frac{d^2y}{dx^2}} = \frac{-(t_1^2 + t_2^2)^{3/2}}{(t_2^2 t_3 - 2t_1 t_2 t_5 + t_1^2 t_4)}$$

where $t_1 = u_1^T Az$, $t_2 = u_2^T Az$, $t_3 = u_1^T Au_1$, $t_4 = u_2^T Au_2$,

and $t_5 = u_1^T Au_2 = u_2^T Au_1$.

85. SUBROUTINE ROTATE

a. Purpose:

Called at the end of Subroutine INITAL to transform those input data variables that have been supplied in local segment geometric coordinates to principal axes coordinates as required by the CVS program for segments (I = 1 to NSEG) whose LPMI(I) is not zero.

b. Subroutines required:

DOTT33, MAT31, MAT33.

c. Labelled common blocks used:

CONTRL, RSAVE, DESCRP, CNTSRF, SGMNTS, JBARTZ, CSTRNT, DAMPER, HRNESS, TEMPVS.

d. Input or argument parameters:

None.

e. Optional output:

None.

f. Procedure:

The following input parameters are transformed for segments whose LPMI(I) are not zero for I = 1 to NSEG.

(1) D and WMEG from input Cards G.3.

(2) SR, HT and HB from input Cards B.3.

(3) RK1 and RK2 from input Cards D.6.

(4) APSDM and APSDN from input Cards D.8.

(5) PL from input Cards D.1 that have been assigned to affected segments on input Cards F.1.

(6) BELT(L,K), for L = 1 to 9, from input Cards D.3 that have been assigned to affected segments on input Cards F.2.

(7) BAR(L,J), for L = 4 to 12, from input Cards F.8.d that have been assigned to affected segments on input Cards F.8.

(8) BD(L,K), for L = 4 to 9 and 16 to 18, for ellipsoid data defined on input Cards B.2 and D.5 for those ellipsoids that have been assigned to affected segments on input Cards F.1, F.2, F.3, F.6 and F.8.

86. SUBROUTINE ROT (A,L,TH)

a. Purpose:

Computes rotation matrix A for angle TH about x, y or z axes as L = 1, 2 or 3.

b. Subroutines required:

None.

c. Labelled common blocks used:

CNSNTS.

d. Input or argument parameters:

A: 3x3 rotation matrix to be computed.
L: 1, 2 or 3 indicating rotation about x, y or z axes, respectively.
TH: Angle of rotation Θ, in radians.

e. Optional output:

None.

f. Procedure:

1. For L = 1, computes

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \text{Cos}\,\theta & \text{Sin}\,\theta \\ 0 & -\text{Sin}\,\theta & \text{Cos}\,\theta \end{pmatrix}$$

2. For L = 2, computes

$$A = \begin{pmatrix} \text{Cos}\,\theta & 0 & -\text{Sin}\,\theta \\ 0 & 1 & 0 \\ \text{Sin}\,\theta & 0 & \text{Cos}\,\theta \end{pmatrix}$$

3. For L = 3, computes

$$A = \begin{pmatrix} \text{Cos}\,\theta & \text{Sin}\,\theta & 0 \\ -\text{Sin}\,\theta & \text{Cos}\,\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Note: Special tests are performed to insure that $\text{Cos}\,\theta$ and $\text{Sin}\,\theta$ are exactly 0 or $\pm$ 1 for values of θ that are multiples of $\pi/2$ to correct for small errors introduced by the SIN and COS routines.

87. SUBROUTINE RSTART (IF,IT)

a. Purpose:

Called by the main program to control the reading and/or writing of the restart input or output units that enables the CVS program to be restarted at a given time point with new values of any of the variables in the labelled common blocks.

b. Subroutines required:

ELTIME, OUTPUT, SEARCH.

c. Labelled common blocks used:

CONTRL, CNTSRF, VPOSTN, SGMNTS, CMATRX, ABDATA, TITLES, CNSNTS, DESCRP, JBARTZ, FORCES, INTEST, CSTRNT, TABLES, COMAIN, CDINT , DAMPER, CEULER, TEMPVI, CYDATA, RSAVE , FLXBLE, HRNESS, WINDFR.

d. Input or argument parameters:

Input Cards A.2.
IF: Integer from 1 to 5 that defines function to be performed.
IT: Defines the FORTRAN unit No. for the restart input unit (IRSIN) if IF = 1 or 2, or for the restart output unit (IRSOUT) if IF = 3 or 5.

e. Optional output:

Contents of input Cards A.2 and values of data replaced.

f. Procedure:

Five separate functions are performed depending on the value of IF.

(1) Read input and initialization record from the restart input unit.

(2) Write input and initialization record onto the restart output unit.

(3) Read old time point record from the restart input unit.

(4) Read new input data from input Cards A.2 for restart.

(5) Write new time point record onto the restart output unit.

88. SUBROUTINE SEARCH (AVAR,INDEX,NCOM,ITEM)

a. Purpose:

Called by Subroutine RSTART to compute the labelled common block number (NCOM) and item number (ITEM) within the block for a given variable name (AVAR) and its index (INDEX). Returns NCOM = 0 for error and NCOM = 50 if AVAR is blank.

b. Subroutines required:

None.

c. Labelled common blocks used:

None.

d. Input or argument parameters:

AVAR: Alphanumer name of variable from input Card A.2.
INDEX: 3 integers representing the index of AVAR.
NCOM: Integer (1 to 24) of labelled common block containing AVAR returned to RSTART. Zero for error, 50 for blanks.
ITEM: Relative location within the labelled common block No. NCOM of AVAR(INDEX).

e. Optional output:

None.

89. SUBROUTINE SEGSEG (M,MM,N,NN,NT)

a. Purpose:

Determines if the ellipsoid MM attached to segment M intersects the ellipsoid NN attached to segment N and adds the resulting forces and torques to the elements of the U1 and U2 arrays for both segments.

b. Subroutines required:

CROSS, DOT33, DOT31, DSMSOL, ELTIME, INTERS, MAT31, MAT33, PLSEGF, XDY.

c. Labelled common blocks used:

FORCES, TABLES, TEMPVS, CNTSRF, SGMNTS, CSTRNT.

d. Input or argument parameters:

MM,NN: Ellipsoid identification numbers (If NN < 0, ellipsoid MM is assumed to be interior to ellipsoid NN).

M,N: Segment identification numbers to which ellipsoids are attached.

NT: Index to NTAB array containing pointers to the function definitions for this segment-segment contact.

e. Optional output:

Segment-segment contact force data are stored in the SSF array for output to the tabular time histories.

f. Procedure:

1. Compute the ellipsoid matrices A and B for segments M and N, and R, the vector from the center of A to the center of B, all in the local reference of segment M.

2. Call Subroutine INTERS to test intersection of ellipsoids A and B, which also computes the contraction factor T_B, and the point y , the single point of intersection of equally contracted ellipsoids, at which the forces will be applied. If no intersection ($T_B \geq 1$) , return to the calling routine.

3. Compute the penetration distance

$$S_1 = (1/T_B - 1) * |R|$$

and express the point y with respect to the centers of both ellipsoids by

$$T_2 = y + B_m$$

$$T_q = y - R + D_m D_n^T B_n$$

where D_i and B_i are the direction cosine matrix and the C.G. offset for the ith segment.

4. Call Subroutine PLSEGF to calculate the contact forces and torques.

5. If a force deflection function does not exist for this function (as indicated by NTAB(NT+1) < 0), data for a rolling constraint are stored in COMMON/CSTRNT/ where MCF = -NTAB(NT+1) is the constraint number

KQTYPE(MCF) = 3

RK1(MCF) = T_2

RK2(MCF) = T_2 expressed in segment N's reference

CFQQ(MCF) = the coefficient of friction

TQQ(MCF) = the normalized direction of the constraint force.

90. SUBROUTINE SETUP1

a. Purpose:

Called by Subroutine DAUX prior to other contact routines to set up initial values for the B12, U1, U2, V1 and V2 arrays.

b. Subroutines required:

ELTIME, CROSS, DOT31.

c. Labelled common blocks used:

CONTRL, DESCRP, SGMNTS, CMATRX, TEMPVS.

d. Input or argument parameters:

None.

e. Optional output:

NPRT(11) $\neq$ 0 will print out the values of the U2 and V1 arrays as they are computed.

f. Procedure:

1. Each 3x1 subarray $u1_n$ of the 3xNSEG array U1 is initially set to zero for n = 1 to NSEG.

2. Each 3x1 subarray $u2_n$ of the 3xNSEG array U2 is initially computed by

$$u2_n = \omega_n \times (\phi_n \omega_n) \qquad \text{for } n = 1 \text{ to NSEG}$$

3. Assume that the (3*NJNT)x(3*NSEG) array B12 is subdivided into NJNTxNSEG subarrays $b12_{m,n}$ of size 3x3. Also, B12 remains constant for a given time and each row j of B12 contains only two elements, $b12_{j,j+1}$ and $b12_{j,i}$ where $i = JNT_j$ for j = 1 to NJNT. Therefore, it is necessary only to allocate 2*NJNT subarrays b12 of size 3x3 to the B12 array by:

for each joint j = 1 to NJNT, where $JNT_j > 0$, compute

$$b12_{2j-1} = b12_{j,i} = -D_i^{-1} r_{2j-1}^x$$

$$b12_{2j} = b12_{j,j+1} = D_{j+1}^{-1} r_{2j}^x$$

The A21 array (previously denoted by A2) is not computed since the program takes advantage of the relation

$$a21_{m,n} = \phi_n^{-1} b12_{m,n}^T$$

4. Set each 3x1 subarray $v1_j$ of the 3xNJNT array V1 by

$$v1_j = -D_i^{-1} \omega_i \times (\omega_i \times r_{2j-1}) + D_{j+1}^{-1} \omega_{j+1} \times (\omega_{j+1} \times r_{2j})$$

for each j = 1 to NJNT where $i = JNT_j > 0$.

If $i \leq 0$, set $v1_j = 0$.

5. Set each 3x1 subarray $v2_j$ of the 3xNJNT array V2 by

if $IPIN_j \neq 1$, $v2_j = 0$, otherwise

$$v2_j = (\omega_i \cdot hb_{2j-1} - \omega_j \cdot hb_{2j}) D_i^{-1} \omega_i \times hb_{2j-1}$$

for each j = 1 to NJNT where $i = JNT_j$.

91. SUBROUTINE SETUP2

a. Purpose:

Called by Subroutine DAUX after all of the other contact routines and by Subroutine UPDATE during state changes for roll-slide constraints to set up the A22, A13, A23, B31, B32 and V3 arrays.

b. Subroutines required:

DHHPIN, DOTT31, DOTT33, DOT31, ELTIME, MAT33.

c. Labelled common blocks used:

CONTRL, SGMNTS, DESCRP, CMATRX, CSTRNT, CNSNTS, TEMPVS.

d. Input or argument parameters:

None.

e. Optional output:

None.

f. Procedure:

1. Compute A22 array for pinned joints as required by the DAUX2 routines by calls to Subroutine DHHPIN.

2. For each constraint k = 1 to NQ sets up the coefficients of the constraint equation

$$a13_{2k-1}\ddot{x}_1 + a13_{2k}\ddot{x}_2 + a23_{2k-1}\omega_1 + a23_{2k}\omega_2 + a33_{k,k}q_k = v3_k$$

where the subscripts 1 and 2 refer to the two segments specified by $KQ1_k$ and $KQ2_k$.

a. For $KQTYPE_k$ = 1 (fixed point constraint)

$$a13_{2k-1} = a13_{2k} = I$$

$$a23_{2k-1} = -D_1^{-1} r_1 x$$

$$a23_{2k} = D_2^{-1} r_2 x$$

$$a33_{k,k} = 0$$

$$v3_k = V(r_2, r_1) = D_2^{-1}\omega_2 \times (\omega_2 \times r_2) - D_1^{-1}\omega_1 \times (\omega_1 \times r_1)$$

b. For $KQTYPE_k$ = 2 (fixed distance constraint)

$$a13_{2k-1} = hh^T$$

$$a13_{2k} = -hh^T$$

$$a23_{2k-1} = -hh^T D_1^{-1} r_1 x$$

$$a23_{2k} = hh^T D_2^{-1} r_2 x$$

$$a33_{k,k} = \lambda (I - hh^T)$$

$$v3_k = hh^T [\, V(r_2, r_1) - \frac{|\dot{\rho}_d|^2}{d} h \,]$$

c. For $KQTYPE_k = 3$ (rolling constraint)

$$a13_{2k-1} = I$$

$$a13_{2k} = -I$$

$$a23_{2k-1} = -D_1^{-1}(s_1 + r_1)x$$

$$a23_{2k} = D_2^{-1}(s_2 + r_2)x$$

$$a33_k = 0$$

$$v3_k = V(s_2 + r_2,\ s_1 + r_1) + D_2^{-1}\omega_2 \times \dot{r}_2 - D_1^{-1}\omega_1 \times \dot{r}_1$$

92. SUBROUTINE SINPUT

a. Purpose:

Reads and prints those input data cards that describe the physical dimensions of planes representing the panels of the vehicle, and of the restraint belts. Also processes those data cards that describe additional contact ellipsoids, constraints and body segment symmetry options.

b. Subroutines required:

AIRBG1, CROSS, DRCYPR.

c. Labelled common blocks used:

CNTSRF, CONTRL, TITLES, CSTRNT, CNSNTS, TEMPVS, SEGMENTS, DAMPER, WINDFR.

d. Input or argument parameters:

Cards D.1, D.2, D.3, D.5, D.6, D.7, D.8 and D.9.

e. Output:

Cards D.1, D.2, D.3, D.5, D.6, D.7, D.8 and D.9.

f. Procedure:

1. Read input Card D.1 containing the number of planes (NPL), belts (NBLT), air bags (NBAG), ellipsoids (NELP), constraints (NQ), spring

dampers (NSD), harness-belt systems (NHRNSS), wind force functions (NWINDF), joint restoring functions (NJNTF) and force functions (NFORCE).

2. If NPL $\neq$ 0, for each plane J = 1 to NPL

   (a) Read Card D.2.a containing the plane No. J and a 20 character description of the plane.

   (b) Read Cards D.3.a, b and c containing the x, y, z coordinates in vehicle reference of three of the corners (p_1, p_2 and p_3) of a bounded rectangular plane defined such that the edge p_1 p_2 is 90 degrees clockwise as viewed by the crash victim from the edge p_1 p_3.

   (c) Set up PL array for the Jth plane consisting of 17 elements as follows:

   PL(I,J) for I = 1 to 4 contain the coefficients a_0, b_0 ,c_0 and d_0 of the normal form of the equation for the Jth plane

   $$a_0X + b_0Y + c_0Z = d_0$$

   where d_0 is a positive number (inches) that is numerically equal to the length of the normal drawn from the origin of the coordinate system to the plane, and a_0 , b_0 and c_0 are the direction cosines of the normal directed from the origin to the plane.

   PL(I,J) for I = 5 to 7 are not currently used by the program.

   PL(I,J) for I = 8 to 12 contain the coefficients a_1, b_1 ,c_1 and d_1

of the normal form of the equation for the boundary plane

$$a_1X + b_1Y + c_1Z = d_1$$

(defined as above) that is perpendicular to the Jth plane at the edge p_1p_3, and e_1 is the length p_1p_2.

PL(I,J) for I = 13 to 17 contain the coefficients a_2, b_2, c_2 and d_2 of the normal form of the equation for the boundary plane

$$a_2X + b_2Y + c_2Z = d_2$$

(defined as above) that is perpendicular to the Jth plane at the edge p_1p_2, and e_2 is the length p_1p_3.

3. If NBLT ≠ 0, read and print the contents of cards D.3.a, b and c.

4. If NBAG ≠ 0, call Subroutine AIRBG1 to process input Cards D.4.

5. If NELP ≠ 0, read cards D.5 for new ellipsoid definitions.
Note: NELP is the number of contact ellipsoids to be supplied on input cards here, not the number of contact ellipsoids in the program. The first NSEG ellipsoids were supplied on Cards B.2.a-B.2.i with no angular rotations. They may be replaced here if desired.

For each J where J = 1 to NELP perform the following:

(a) Read input card D.5 containing

M: Contact ellipsoid number. Maximum value = 24.
If M < NSEG, data will replace input data supplied on card B.2.M.

P1(I), I = 1,3: The X, Y and Z semiaxes of the contact ellipsoid.

P2(I), I = 1,3: The X. Y and Z coordinates of the ellipsoid offset, i.e., the vector from the segment C.G. to the ellipsoid center.

P3(I), I = 1,3: The yaw, pitch and roll of the contact ellipsoid from the principal axis of the segment.

(b) Store data into BD(J,M) for I = 1 to 24 as follows:

I = 1,3: P1 as defined above

I = 4,6: P2 as defined above

I = 7,15: The ellipsoid matrix $A = D_e^{-1} B D_e$

where $B = \begin{pmatrix} 1/P1_x^2 & 0 & 0 \\ 0 & 1/P1_y^2 & 0 \\ 0 & 0 & 1/P1_z^2 \end{pmatrix}$

and D_e is the direction cosine matrix of the ellipsoid as computed by Subroutine DRCYPR with P3 as arguments.

I = 16,24: The inverse of matrix A

6. If NQ ≠ 0, read NQ input Cards D.6 for constraint input.

7. Read Card D.7 which controls the symmetry option of the body segments.

8. IF NSD ≠ 0, read NSD input Cards D.8 for spring damper input.

9. If NFORCE ≠ 0, read NFORCE input Cards D.9 for force functions input.

93. SUBROUTINE SLPLOT (X,NX,XO,XN,XL,XSIZE,XLAB,NXLB,
Y,NY,YO,YN,YL,YSIZE,YLAB,NYLB,
NPTS,NYY,NDY,PLAB1,NPLB1,PLAB2,NPLB2)

a. Purpose:

Called by Subroutine POSTPR to produce a labelled X-Y plot on the host computer system plotter output unit. Each axis may be either linear or logarithmic and their characteristics are completely determined by the argument input parameters.

b. Subroutines required:

(CVS Program subroutines) LINAXS, LOGAXS.
(CALCOMP subroutines) NEWPEN, NUMBER, PLOT, PLOTS, SYMBOL.

c. Labelled common blocks used:

None.

d. Input or argument parameters:
(Note: all real data is single precision.)

X(NPTS):	Array of NPTS abscissas to be plotted.
Y(NDY,NYY):	Array of NPTS*NYY ordinates to be plotted.
NX,NY:	Positive - number of linear subdivisions on axis.
	Negative - number of logarithmic decades on axis.
XO,YO:	Axes origins (power of ten if NX,NY is negative).
XN,YN:	Axes end values (not used if NX,NY is negative).
XL,YL:	Length (inches) of X,Y axes.

XSIZE,YSIZE: Paper size (inches) in X,Y directions.

XLAB,YLAB: X,Y axes labels (alphanumeric arrays).

NXLB,NYLB: Number of characters in the X,Y labels.

NPTS: Number of points of X abscissas and each Y ordinates.

NYY: Number of Y ordinates (lines) to be plotted vs. X.

NDY: First dimension of Y array in calling routine (NDY must be greater than or equal to NPTS).

PLAB1,PLAB2: Two lines of plot ID labels (alphanumeric arrays).

NPLB1,NPLB2: Number of characters in each plot ID label.

e. Optional output:

None.

94. SUBROUTINE SPDAMP

a. Purpose:

Called by Subroutine CONTCT if NSD ≠ 0 to compute the spring and viscous forces of spring dampers between points on selected segments as specified on input Cards D.8. The resulting forces and torques are then added to the U1 and U2 arrays for these segments.

b. Subroutines required:

CROSS, DOT31, ELTIME, EVALFD, MAT31.

c. Labelled common blocks used:

SGMNTS, DAMPER, TABLES, FORCES, TEMPVS.

d. Input or argument parameters:

None.

e. Optional output:

The values of DEL and FD+FS are stored in the BSF array for output on the tabular time histories.

f. Procedure:

For each spring damper i = 1 to NSD, perform the following calculations:

1. Fetch the segment numbers m and n and the points on these segments r_m and r_n specified in local reference.

2. Convert r_m and r_n to inertial reference by

$$z_m = x_m + D_m^T r_m \quad \text{and} \quad z_n = x_n + D_n^T r_n$$

where x_j and D_j are the location of the C.G. and direct cosine matrix for segment No. j.

3. Compute the vector and distance between the points by

$$R = z_m - z_n \quad \text{and} \quad d = |R|$$

4. Compute the relative velocity by

$$d_v = \dot{x}_m + D_m^T w_m x r_m - \dot{x}_n - D_n^T w_n x r_n$$

and the component of d_v along the line connecting the points by

$$d_R = d_v \cdot \frac{R}{|R|}$$

5. Compute the spring and viscous force by

$$F_S = a_1(d - d_0) + a_2(d - d_0)^2$$

$$F_D = b_1 d_R + b_2 d_R^2$$

and their components along the unit vector by

$$F = (F_S + F_D)\frac{R}{|R|}$$

6. Add the components of the force and torque to the U1 and U2 arrays by

$$u1_m = u1_m - F$$

$$u1_n = u1_n + F$$

$$u2_m = u2_m - r_m x D_m F$$

$$u2_n = u2_n + r_n x D_n F$$

95. SUBROUTINE SPLINE (X,Y,F,N,L)

a. Purpose:

Called by Subroutine VINPUT to fit a set of polynomials of degree L to a set of given data points (X(i), Y(i) for i = 1 to N).

b. Subroutines required:

None.

c. Labelled common blocks used:

None.

d. Input or argument parameters:

X,Y: Given set of N data points.
F: Array of size (5,N) allocated by calling routine to contain the coefficients of the N polynomials to be computed.
N: Number of given data points and polynomials to be computed.
L: Degree of the N polynomials to be computed.

e. Optional output:

None.

f. Procedure:

Assumes function to be of the form

```
Y(X) = F(2,K) + F(3,K)*DX + F(4,K)*DX**2 + F(5,K)*DX**3
```

where DX = X - F(1,K)
for K such that F(1,K) < X < F(1,K+1)

		CONTINUITY
Returns: F(1,I) = X(I) and F(2,I) = Y(I)	for all L	
F(3,I) = F(4,I) = F(5,I) = 0	for L = 0	None
F(4,I) = F(5,I) = 0	for L = 1	Y
F(5,I) = 0	for L = 2	Y,Y'
cubic spline	for L = 3	Y,Y',Y''
F(K,N) = 0 for K = 3 to 5 in all cases		

For L = 2 or 3, the changes in the Lth derivative are minimized.

Usage:

```
C   Given L, N, (X(I),Y(I) for I = 1 to N)
        CALL SPLINE (X,Y,F,N,L)
C   To evaluate function and derivatives at point XX
        DO  10  K=1,N
        IF  (K.EQ.N)  GO TO 11
        IF  (XX.LT.F(1,K))  GO TO 11
  10    CONTINUE
  11    DX = XX - F(1,K)
        YY = F(2,K) + DX*(F(3,K)+DX*(F(4,K)+DX*F(5,K)))
        YD = F(3,K) + DX*(2.0*F(4,K)+3.0*DX*F(5,K))
        YDD = 2.0*F(4,K) + 6.0*DX*F(5,K)
        YDDD = 6.0*F(5,K)
        YDDDD = 0.0
C   Where functional value is YY, derivatives are YD's.

C   Repeat for next value of XX.
```

96. SUBROUTINE SPRNGF (Z,D,ZD,S,JSTOP)

a. Purpose:

Computes the nonlinear spring torque for joints as a function of flexural and torsional angles. (Note: this routine is no longer used in the CVS program. Its functions are now performed by Subroutine EFUNCT.)

b. Subroutines required:

None.

c. Labelled common blocks used:

CNSNTS.

d. Input or argument parameters:

1. Z: $\cos\theta$, where θ is the flexural or torsional angle of the joint.
2. D: $|\sin\theta|$.
3. ZD: $-\dot{\theta}$.
4. S: Array of 5 values from SPRING(5,28) as follows.
   (a) S_1, S_2, S_3: coefficients of a polynominal define nonlinear spring torque for the loading curve as a function of angle
   $$T = S_1\theta + S_2\theta^2 + S_3\theta^3$$
   (b) S_4: energy dissipation coefficient for the unloading curve.
   (c) S_5: joint stop angle, must be in radians.

5. JSTOP: an indicator to be returned to calling program to signify that the joint has reached the joint stop. Returns 0 for not in the stop, 1 if in the stop.

e. Optional output:

None.

f. Procedure:

Evaluates and returns to the calling program the spring torque for joints as a function of flexural or torsional angle as follows.

1. If $\theta \leq S_5$, $T = S_1\theta$ (note $0 \leq \theta \leq \pi$).

2. If $\theta > S_5$ and $\dot{\theta} \geq 0$, $T = S_1\theta + S_2\theta^2 + S_3\theta^3$.

3. If $\theta > S_5$ and $\dot{\theta} < 0$, $T = S_1\theta + (S_2\theta^2 + S_3\theta^3)S_4$.

Actually routine returns $T/|\sin\theta|$ to calling program.
Provision is made for small angles and angular rates.

97. SUBROUTINE TRIGFS

a. Purpose:

Called by Subroutine DINT at the beginning of each integration step to compute the values for the GH and UU arrays to be used at each intermediate step of the upcoming integration interval.

b. Subroutines required:

None.

c. Labelled common blocks used:

CDINT.

d. Input or argument parameters:

None.

e. Optional output:

None.

98. SUBROUTINE UNIT1 (IND)

a. Purpose:

Called by the main program if NPRT(1) $\neq$ 0 to write the segment and plane data on output unit No. 1 (unformatted) at NPRT(1)*DT seconds time intervals. This unit is designed to serve as the input file for the AFAMRL program VIEW.

b. Subroutines required:

None.

c. Labelled common blocks used:

CONTRL, SGMNTS, CNTSRF, JBARTZ, RSAVE, TEMPVS.

d. Input or argument parameters:

IND: (Currently not used by subroutine).

e. Optional output:

The output data records are written (unformatted) on output unit No. 1.

f. Procedure:

For the first time in the routine only, write the initial record onto output unit No. 1. This record consists of NSEG, NPL and the PL(*), BD(*) and MPL arrays (* indicates the data are converted to single

precision prior to transmission).

At the fixed time point intervals (multiples of TIME = NPRT(1)*DT seconds including zero), write the time point data records onto output unit No. 1. These records consist of TIME(*) and the SEGLP(*) and D(*) arrays.

99. SUBROUTINE UPDATE (I)

a. Purpose:

Called by Subroutine DINT at the end of a successful integration step (I = 2) to complete calculations for that step and at the start of a new integration step (I = 1) to set up any new conditions to be valid for the entire step.

b. Subroutines required:

AIRBAG3, CROSS, DAUX, DOT31, ELTIME, HPTURB, IMPULS2, OUTPUT, PRINT, SETUP2, UPDFDC, XDY.

c. Labelled common blocks used:

CONTRL, JBARTZ, TABLES, DESCRP, SGMNTS, CMATRX, TEMPVI, FORCES, CSTRNT, CEULER, HRNESS.

d. Input or argument parameters:

I = 1: indicates that subroutine is being called at the start of a new integration step by Subroutine DINT. Value of -1 will be returned is either Subroutine IMPULS or IMPLS2 has been called during current update to indicate that integrator should be reset.

I = 2: indicates that subroutine is being called at the end of a successful integration step.

e. Optional output:

None.

f. Procedure:

1. Subroutine AIRBG3 is called if NBAG ≠ 0 and, if J = 2, control is returned to calling routine.

2. Otherwise, for each plane-segment contact, segment-segment contact, and belt-segment contact and harness-belt systems:
   (a) call Subroutine UPDFDC with argument NT, the index to the elements in the NTAB array controlling this contact; and
   (b) if initial contact has occured (indicated by NT being set negative in UPDFDC), and, if the coefficient of restitution for this contact is nonzero, then call Subroutine IMPULS for plane-segment and segment-segment contacts only.

3. If an initial entry into a joint stop has been detected by Subroutine VISPR as indicated by the value of JSTOP, then call Subroutine IMPULS2.

4. Test to lock or unlock joints by changing the sign of IPIN(J) according to the following table. If an unlocked joint is changed to a locked joint, then Subroutine IMPLS2 is called.

Conditions to change sign of IPIN(J)

	Pinned (1)	Unpinned (2)
Locked (-)	H $t_q > T_{1j}$	$\|t_q\| > T_{1j}$
Unlocked (+)	H $t_q < T_{2j}$ or $j < T_{3j}$	$\|t_q\| < T_{2j}$ or $j > T_{3j}$

Note: If the values of T_1, T_2 or T_3 is zero the corresponding test is not performed.

5. Step No. 4 is repeated for Euler joints.

6. Perform test to determine if state conditions should be changed for rolling and sliding constraints (KQTYPE = 3 or 4).

100. SUBROUTINE UPDFDC(M)

a. Purpose:

Updates the force-deflection curve definition that is defined in location M of the NTAB array. Subroutine assumes that a successful integration step has just been completed and will compute the entire curve definition to be valid for the next time step.

b. Subroutines required:

EVALFD, FRCDFL.

c. Labelled common blocks used:

TABLES.

d. Input or argument parameters:

M is the index to the NTAB array element L which in turn is the index to a subarray of 20 elements in the TAB array which defines the force-deflection function.

e. Optional output:

None.

f. Procedure:

Each allowed contact will have a pointer M to six elements in the NTAB

array which are in turn pointers to data in the TAB array as follows:

NTAB(M) - TAB pointer (L) to real data for this contact.

NTAB(M+1) - TAB pointer to force-deflection function input data.

NTAB(M+2) - TAB pointer to inertial spike function unput data (0 indicates no function).

NTAB(M+3) - TAB pointer to R (energy absorption) function input input data (0 indicates default R = 1.0).

NTAB(M+4) - TAB pointer to G (deflection) function input data (0 indicates default G = 0).

NTAB(M+5) - TAB pointer to coefficient of friction function input data.

The first pointer L located in NTAB(M) points to an array of 20 variables in TAB which contains the following data for this contact:

TAB(L): δ - Current abscissa of force-deflection function.

TAB(L+1): δ_{last} - Previous value of deflection δ .

TAB(L+2): AREA - Area under force-deflection curve.

TAB(L+3): R_{last} - Last value of R while on base function.

TAB(L+4): G_{last} - Last value of G while on base function.

TAB(L+5): δ_{quad} - Lower abscissa value for $F_q(\delta)$.

TAB(L+6): δ_{cubic} - Upper abscissa value for $F_q(\delta)$ and lower abscissa value for $F_c(\delta)$.

TAB(L+7): δ_{ref} - Upper abscissa value for $F_c(\delta)$ and point on base function from which unloading originated.

TAB(L+8): δ_{max} - Maximum abscissa value of base function definition.

TAB(L+9): δ_{iner} - Maximum abscissa value of inertial spike function definition.

TAB(L+10): $F(\delta_{max})$ - Value of force-deflection curve for $\delta \geq \delta_{max}$.

TAB(L+11): q_0
TAB(L+12): q_1
TAB(L+13): q_2

- Coefficients of unloading quadratic function

$$F_q(\delta) = q_0 + q_1(\delta - \delta_{quad}) + q_2(\delta - \delta_{quad})^2$$

defined for $\delta_{quad} \leq \delta \leq \delta_{cubic}$.

TAB(L+14): c_0
TAB(L+15): c_1
TAB(L+16): c_2
TAB(L+17): c_3

- Coefficients of reloading cubic function

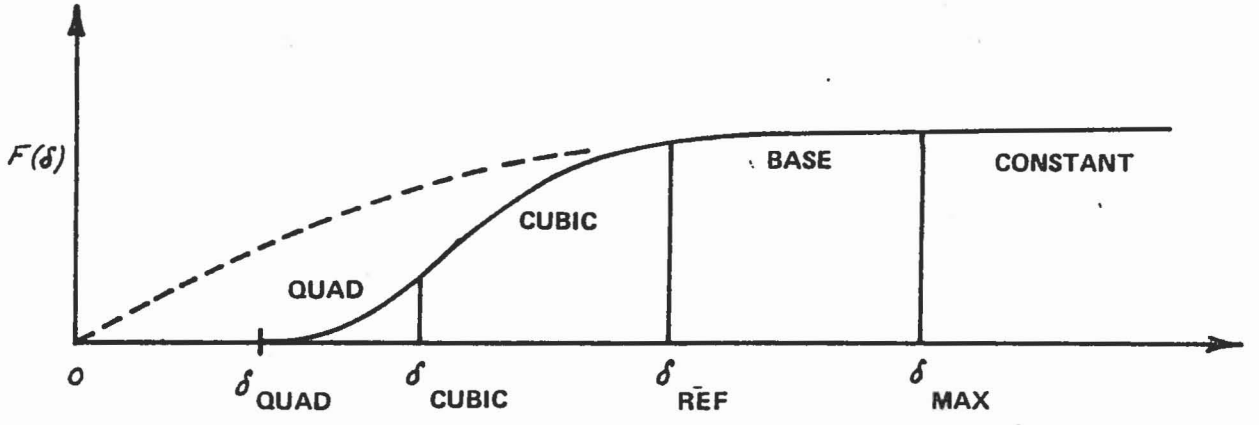
$$F_c(\delta) = c_0 + c_1(\delta - \delta_{cubic}) + c_2(\delta - \delta_{cubic})^2 + c_3(\delta - \delta_{cubic})^3$$

defined for $\delta_{cubic} \leq \delta \leq \delta_{ref}$.

TAB(L+18): c_0 - δ_{cubic} for current cubic definition.

TAB(L+19): dummy - Not currently used.

At any time the entire force-deflection function is as follows:



where $0 \leq \delta_{quad} \leq \delta_{cubic} \leq \delta_{ref} \leq \delta_{max}$

(initially $\delta_{quad} = \delta_{cubic} = \delta_{ref} = 0$).

The subroutine then redefines a new force-deflection function depending on current value of δ as follows:

1. If L = 0, or if $\delta \leq 0$, or if $\delta = \delta_{last}$, return to the calling program.

2. If $\delta < \delta_{cubic}$, unloading is occurring, define reloading cubic function from δ to δ_{ref}

   (a) If inertial spike exists and if $\delta_{ref} > D_{imax}$, remove inertial spike from further consideration by setting NTAB(M+2) = 0.

   (b) Set $\delta_{cubic} = \text{Max}(\delta, \delta_{quad})$

   $$\delta_{c_0} = \delta_{cubic}$$

   $$\Delta = \delta_{ref} - \delta_{cubic}$$

   (c) Define new cubic reloading function

   $$F_c(\delta) = c_0 + c_1(\delta - \delta_{cubic}) + c_2(\delta - \delta_{cubic})^2 + c_3(\delta - \delta_{cubic})^3$$

   for $\delta_{cubic} \leq \delta \leq \delta_{ref}$ that satisfies the following conditions:

   $$F_c(\delta_{ref}) = F_{base}(\delta_{ref})$$

   $$F'_c(\delta_{ref}) = F'_{base}(\delta_{ref})$$

   $$F_c(\delta_{cubic}) = F_q(\delta_{cubic})$$

   by setting $c_0 = F_q(\delta_{cubic})$ and $c_1 = F'_q(\delta_{cubic})$

and by solving simultaneously for c_2 and c_3

$$c_2 \Delta^2 + c_3 \Delta^3 = F_{base}(\delta_{ref}) - c_0 - c_1 \Delta$$

$$2c_2 \Delta + 3c_3 \Delta^2 = F'_{base}(\delta_{ref}) - c_1$$

(d) If local minimum of new cubic definition lies between δ_{cubic} and δ_{ref} and is negative, then replace cubic definition with a straight line between the points $[\delta_{cubic}, F_q(\delta_{cubic})]$ and $[\delta_{ref}, F_{base}(\delta_{ref})]$ by

$$c_0 = F_q(\delta_{cubic})$$

$$c_1 = \frac{1}{\Delta}[F_{base}(\delta_{ref}) - F_q(\delta_{cubic})]$$

$$c_2 = c_3 = 0$$

and return to the calling program.

3. If $\delta_{cubic} \leq \delta \leq \delta_{ref}$, reloading is occuring; define new quadratic unloading curve from cubic reloading curve by:

let $y_2 = F_c(\delta_{c_0})$

and AREA $= \int_{\delta_{quad}}^{\delta_{cubic}} F_q(\delta)d\delta + \int_{\delta_{c_0}}^{\delta} F_c(\delta)d\delta - \int_{\delta_{c_0}}^{\delta_{cubic}} F_c(\delta)d\delta$

(Note: δ_{c_0} was the value of δ_{cubic} when $F_c(\delta)$ was defined.)

and go to step number 5.

4. Otherwise, $\delta_{ref} < \delta$; define new quadratic unloading curve from base curve.

(a) If $\delta \geq \delta_{iner}$, remove inertial spike from further consideration by setting NTAB(M+2) = 0.

(b) Determine R factor and place into R_{last}.

If R = 1, use base curve for unloading by setting

$$\sigma_{quad} = \sigma_{cubic} = \sigma_{ref} = q_0 = q_1 = q_2 = 0$$

and return to the calling program.

(c) Determine G factor and place into G_{last}. Fetch D_0 from input data for base function and compute

$$\sigma_{quad} = D_0 + G_{last}(\sigma - D_0)$$

$$y_2 = F_{base}(\sigma)$$

$$y_2 = y_2 + F_{inertial}(\sigma) \text{ if inertial spike exists.}$$

$$AREA = \int_0^{\sigma} F_{base}(\sigma)d\sigma$$

5. Using values of y_2 and AREA defined in either step 3 or 4, determine new quadratic unloading function

$$F_q(\sigma) = q_0 + q_1(\sigma - \sigma_{quad}) + q_2(\sigma - \sigma_{quad})^2$$

for $\sigma_{quad} \leq \sigma \leq \sigma_{cubic}$ that satisfies the following conditions where $\sigma_{cubic} = \sigma$

$$F_q(\sigma_{quad}) = 0$$

$$F_q(\sigma_{cubic}) = y_2$$

and $$\int_{\sigma_{quad}}^{\sigma_{cubic}} F_q(\sigma)d\sigma = R_{last} + AREA$$

by setting

$$q_0 = 0$$

$$q_1 = \frac{2}{\sigma_{cubic} - \sigma_{quad}} \left[\frac{3R_{last}\ AREA}{\sigma_{cubic} - \sigma_{quad}} - y_2 \right]$$

(if $q_1 < 0$, q_1 is set to 0 to guarantee non-negative derivative at $\delta = \delta_{quad}$)

and $$q_2 = \frac{1}{\delta_{cubic} - \delta_{quad}} \left[\frac{y_2}{\delta_{cubic} - \delta_{quad}} - q_1 \right]$$

(If $q_2 < 0$, $q_2 = \frac{y_2}{\delta_{cubic} - \delta_{quad}}$ to guarantee non-negative derivative at $\delta = \delta_{cubic}$.)

6. Array of 20 elements in TAB array are restored reflecting any changes that have been made by the routine.

7. If initial contact or recontact has occurred during previous integration time step as determined by

$$\delta > \delta_{quad_0} \quad \text{and} \quad \delta_{last} \leq \delta_{quad_0}$$

where δ_{quad_0} was the value of δ_{quad} at entry to routine, then M is set negative as an indicator to the calling routine.

101. SUBROUTINE VEHPOS

a. Purpose:

Computes the components of vehicle linear and angular acceleration as a function of time using data and tables produced by Subroutine VINPUT.

b. Subroutines required:

None.

c. Labelled common blocks used:

CNSNTS, VPOSTN, CONTRL, SGMNTS.

d. Input or argument parameters:

None.

e. Optional output:

None.

f. Procedure:

Since the computation of the vehicle motion is now performed by the program integrator, it is only necessary for this routine to return the components of linear acceleration, x(t) stored in the SEGLA array, and of angular acceleration, (t) stored in the WMEGD array for the current value of t (TIME). Three options are possible depending on

the value of NATAB.

1. NATAB = 0 (half-sine wave deceleration)

$$\ddot{x}(t) = -a_x \omega^2 \text{Sin } \omega t \qquad \text{for } t = \text{Min(TIME,VTIME)}$$

$$\dot{\omega}(t) = 0$$

2. NATAB > 0 (Unidirectional deceleration table)
   a) If $t \geq$ VTIME, then set c_0 equal to last entry in table, else compute c_0 by quadratic interpolation from table.
   b) $\ddot{x}(t) = -a_x c_0 g$

   $\dot{\omega}(t) = 0$

3. NATAB < 0 (Omnidirectional deceleration table)
   a) If $t \geq$ VTIME, then set c_0 and d_0 equal to last entries in tables, else compute c_0 and d_0 by linear interpolation from tables.
   b) $\ddot{x}(t) = -gc_0$

   $\dot{\omega}(t) = d_0$ (converted from degrees to radians)

102. SUBROUTINE VINPUT

a. Purpose:

Called by the main program to perform the input and compute data and tables required by Subroutine VEHPOS to integrate the motion of specified segments including the vehicle per one of four permissible options:

1. Half sine-wave linear deceleration impulse
2. Unidirectional linear deceleration tabular input
3. Omnidirectional linear and angular acceleration tabular input (6 degrees of freedom vehicle motion)
4. Spline fit of position, velocity or acceleration data.

b. Subroutines required:

DSETD, YPRDEG, DRCYPR, SPLINE.

c. Labelled common blocks used:

CNSNTS, TITLES, VPOSTN, CONTRL, SGMNTS, DESCRP, INTEST, TEMPVS.

d. Input or argument parameters:

Cards C.1, C.2, and C.3, C.4 or C.5 as required.

e. Optional output:

In addition to the contents of cards C.1 and C.2, the following output is produced depending on the choice of vehicle motion.

1. Half sine-wave linear deceleration impulse (NATAB = 0) simply lists the initial velocity, the angles describing the direction of the vehicle and the time duration of the impulse.

2. Unidirectional linear deceleration tabular input (NATAB > 0) produces a table of the inputted linear deceleration and the integrated values of linear velocity and linear displacement for equally spaced time points from zero to (NATAB - 1)*ADT seconds.

3. Omnidirectional linear and angular acceleration tabular input or the 6 degrees of freedom vehicle motion (NATAB < 0 and LTYPE = 0) produces tables of the three components of inputted linear deceleration and angular acceleration and the integrated values of linear and angular velocity and displacement for equally spaced time points from zero to (-1 - NATAB)*ADT seconds.

4. Spline fit data input (NATAB < 0 and LTYPE > 0) produces tables that are identical to those described under step 3 above from the polynomials returned from Subroutine SPLINE evaluated at the specified time points.

f. Procedure:

1. Input and print contents of cards C.1 and C.2.

2. Compute the vector

$$a_x = \begin{pmatrix} \text{Cos } a_1 \text{Cos } a_2 \\ \text{Sin } a_1 \text{Cos } a_2 \\ \text{Sin } a_2 \end{pmatrix} \qquad \text{where } a_1 = \text{yaw}, \quad a_2 = \text{pitch}$$

3. If NATAB $\neq$ 0, go to step No. 5.

4. For half wave deceleration impulse compute:

$$\omega = \pi/\text{VTIME}$$

$$a_t = \frac{1}{2} \frac{V_{ips}}{\omega}$$

$$a_x = a_t\, a_x$$

$$\dot{x}_0 = V_{ips} a_x$$

and go to step No. 9.

5. If NATAB < 0, go to step No. 10.

6. Read unidirectional linear deceleration table from cards C.3

$\ddot{x}_i$ at $t_i = (i - 1)\Delta T$ for i = 1 to NATAB

and extend table if necessary such that NATAB is odd and $\ddot{x}_{NATAB} = 0$.

7. Initialize:

$$\dot{x}_1 = V_{ips}$$

$$x_1 = 0$$

8. Using Simpson's integration, compute velocity table by:

(a) $$\dot{x}_i = \dot{x}_{i-1} - \frac{\Delta T}{12}(5x_{i-1} + 8x_i - x_{i+1})g$$

(b) $$\dot{x}_{i+1} = \dot{x}_{i-1} - \frac{\Delta T}{3}(x_{i-1} + 4x_i + x_{i+1})g$$

for i = 2, 4, 6, ----, NATAB-1

and compute displacement table by:

(c) $x_i = x_{i-1} + \frac{\Delta T}{12}(5\dot{x}_{i-1} + 8\dot{x}_i - \dot{x}_{i+1})$

(d) $x_{i+1} = x_{i-1} + \frac{\Delta T}{3}(\dot{x}_{i-1} + 4\dot{x}_i + \dot{x}_{i+1})$

for i = 2, 4, 6, ----, NATAB-1

and print $\ddot{x}_i$, $\dot{x}_i$ and x_i in tabular output form.

9. Perform initialization:

$D_V = I$ (direction cosine matrix of vehicle)

$\omega_V = 0$ (angular rotation vector of vehicle)

$\dot{\omega}_V = 0$ (time derivative of ω_V)

and go to step no. 13.

10. Read input Card C.2.B and, if LTYPE = 0 which designates omni-directional (6 degrees of freedom) vehicle motion, read three components of linear deceleration ($\ddot{x}_j$) and angular acceleration ($\dot{\omega}_j$) from Cards C.4. These tables are defined at time points $t_j = (j - 1)\Delta T$ for j = 1 to MATAB (MATAB = -NATAB).

11. If LTYPE ≠ 0 which designates the spline fit data option, then read input Cards C.5 and call Subroutine SPLINE to compute the spline polynomials to pit the input data. Use these polynomials to compute the three components of linear deceleration and angular acceleration for the time points defined in step No. 10 above.

12. Compute and print complete vehicle motion output by:

perform the initialization

D_V by calling the DRCYPR routine using the values of yaw, pitch, and roll form card C.2.

$\dot{x}_0 = V_{ips} D_V X$ where $X = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$

$\dot{x}_1 = \dot{x}_0$

$x_1 = x_0$ from card C.2

$\omega_1 = 0.$

For each time step j = 2 to MATAB compute vectors:

(a) $\dot{x}_j = \dot{x}_{j-1} - \frac{\Delta T}{2}(\ddot{x}_{j-1} + \ddot{x}_j)g$

(b) $x_j = x_{j-1} + \Delta T[\dot{x}_{j-1} - \frac{\Delta T}{6}(2\ddot{x}_{j-1} + \ddot{x}_j)g]$

(c) $\omega_j = \omega_{j-1} + \frac{\Delta T}{2}(\dot{\omega}_{j-1} + \dot{\omega}_j)$

(d) $\Delta\theta_j = \Delta T[\omega_{j-1} + \frac{\Delta T}{6}(2\dot{\omega}_{j-1} + \dot{\omega}_j)$

(e) Integrate D_V from t_{j-1} to t_j by calling Subroutine DSETD.

(f) Compute θ_j (yaw, pitch, and roll) by calling Subroutine YPRDEG.

(g) Print $\ddot{x}_j$, $\dot{x}_j$, x_j , $\dot{\omega}_j$, ω_j and θ_j in tabular output form.

13. Perform the required program initialization for prescribed segment motion. If MSEG (Card C.2) is not zero then repeat by going back to step (1); otherwise (last set of data was for the primary vehicle) return to the calling routine.

103. FUNCTION VISCOS (ZD,V,HA)

a. Purpose:

Called by Subroutine VISPR and EJOINT to compute the sum of coulomb friction and viscous torques at the joints as a function of , the relative angular velocity at the joint. Routine returns sum/ .

b. Subroutines required:

None.

c. Labelled common blocks used:

None.

d. Input or argument parameters:

1. ZD: ω.

2. V: Array of 5 values from VISC(5,28) as defined on input Cards B.5.j.
   (a) V_1: linear coefficient for viscous torque = $V_1\omega$.
   (b) V_2: constant value of coulomb friction torque for $\omega \geq V_3$.
   (c) V_3: limit of linear form of coulomb friction torque, used to reduce torques to zero at ω = 0 rad/sec to eliminate discontinuity in torque at ω = 0.
   (d) V_4, V_5: not currently used.
3. HA: Ramp function constant returned to calling routine.

e. Optional output:

None.

f. Procedure:

1. Evaluates coulomb friction torque T_c by:

   (a) if $|\omega| \geq V_3$, $T_c = V_2$.

   (b) if $|\omega| < V_3$, $T_c = V_2(\omega/V_3)$.

2. Evaluates viscous torque T_V by $T_V = V_1\omega$.

3. Returns sum $T = (T_c + T_V)/\omega$ to the calling program.

104. SUBROUTINE VISPR (IJ,NJ)

a. Purpose:

Computes viscous and spring torques at the joints and adds them to the U2 array.

b. Subroutines required:

CROSS, DOT33, EFUNCT, ELTIME, FNTERP, GLOBAL, MAT31, VISCOS.

c. Labelled common blocks used:

DESCRP, CONTRL, CMATRX, TEMPVS, FORCES, CEULER, SGMNTS, CNSNTS, TEMPVI.

d. Input or argument parameters:

NJ = 0: regular computation for all joints
≠ 0: compute only for joint No. NJ impulse
IJ = 1: impulse for flexure only
= 2: impulse for torsion only.
= 4: impulse for global graphic only.

e. Optional output:

NPRT(12) ≠ 0 will print for diagnostic of check-out purposes J, CV, CSA, CSB, HAC, RA, RB; the vectors TQ, WIJ, T7 and ANGL; and the arrays DH1, HD3 and HIR.

f. Procedure:

1. For each joint j = 1 to NJNT, determine the segment number i = JNT_j which is connected to segment number j+1 by joint number j.

2. Convert to inertial reference system

$T_1 = A_i = D_i^{-1} Ha_{2j-1}$ $\qquad$ $T_4 = A_j = D_{j+1}^{-1} Ha_{2j}$

$T_2 = B_i = D_i^{-1} Hb_{2j-1}$ $\qquad$ $T_5 = B_j = D_{j+1}^{-1} Hb_{2j}$

$T_3 = W_i = D_i^{-1} \omega_i$ $\qquad$ $T_6 = W_j = D_{j+1}^{-1} \omega_{j+1}$

3. Compute

$HAD = \cos\theta_a = A_i \cdot A_j$

$HBD = \cos\theta_b = B_i \cdot B_j$

$WIJ = W_{ij} = \omega_i - \omega_j$

$T_7 = A_i \times A_j$

$T_8 = B_i \times B_j$

$HAC = |A_i \times A_j| = |T_7|$

4. If HAC $\neq$ 0, recompute HBD by

$$HBD = \cos\theta_b = \frac{(B_i \cdot T_7)(B_j \cdot T_7) - (B_i \cdot A_j)(A_i \cdot B_j)}{|T_7|^2}$$

and T_8 is projected into the plane perpendicular to $A_i \times A_j$

$$T_{8_p} = T_8 - T_7(T_7\ T_8)/|T_7|^2$$

5. Compute the magnitude of the viscous and coulomb friction torque (C_v) divided by $|W_{ij}|$ from Function VISCOS.

6. Compute RA($-\text{sgn}\ \dot{\theta}_a$) and RB($-\text{sgn}\ \dot{\theta}_b$) by

$$RA = W_{ij} \cdot T_7$$

$$RB = W_{ij} \cdot T_8$$

7. Compute the magnitude of the flexure torque/HAC(CSA) and torsional torque/HBC(CSB) by FUNCTION EFUNCT.

8. Compute total torque in the inertial reference system by

$$T_Q = -C_v W_{ij} + C_{sa} T_7 + C_{sb} T_{8_p}$$

9. Add torque converted to segment reference to U2 by

$$u2_i = u2_i + D_i T_Q$$

$$u2_{j+1} = u2_{j+1} - D_{j+1} T_Q$$

105. SUBROUTINE WINDY (M,MM,N,NN,NT)

a. Purpose:

Called by Subroutine CONTCT, if either NWINDF or NFORCE are not zero, to compute the wind blast forces specified on input Cards E.6 and F.7, or the directed force functions specified on input Cards D.9. The resulting forces and torques are then added to the U1 and U2 arrays for the affected segments.

b. Subroutines required:

CROSS, DOTT33, DOT31, ELTIME, EVALFD, MAT31.

c. Labelled common blocks used:

CONTRL, SGMNTS, TABLES, WINDFR CNTSRF, CNSNTS, TEMPVS.

d. Input or argument parameters:

M: Positive - segment identification number for wind blast forces.
Negative - indicates directed force calculations are performed.
Other argument parameters will not be used here.
MM: The ellipsoid number assigned to segment no. M.
N: The intersecting boundary plane number.
NN: The segment identification number assigned to plane no. N.
NT: The wind blast function number from input Card E.6.a.

e. Optional output:

If NPRT(14) is not zero, then diagnostic output of the wind blast forces is produced on the primary output unit.

106. FUNCTION XDY (X,D,Y)

a. Purpose:

Function routine to compute X•DY or Y•D'X, where D is a 3x3 matrix, D' is the transpose of D, X and Y are vectors with 3 components, and • represents the standard dot product.

b. Subroutines required:

None.

c. Labelled common blocks used:

None.

d. Input or argument parameters:

X,Y: Vectors with 3 components.
D: Matrix of size 3x3.

e. Optional output:

None.

f. Procedure:

The returned answer z is a scalar computed by

$$z = \sum_{i=1}^{3} \sum_{j=1}^{3} x_i \, d_{ij} \, y_j$$

107. SUBROUTINE YPRDEG (D,A)

a. Purpose:

Computes yaw, pitch, and roll in degrees and places them into the A array for a given cosine matrix D.

b. Subroutines required:

None.

c. Labelled common blocks used:

CNSNTS.

d. Input or argument parameters:

D: 3x3 direction cosine matrix.

A: Array of length 3 into which yaw, pitch, and roll in degrees will be stored to be returned to the calling routine.

e. Optional output:

None.

f. Procedure:

Assumes $D = D_r D_p D_y$, where

$$D_r = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \text{Cos } r & \text{Sin } r \\ 0 & -\text{Sin } r & \text{Cos } r \end{pmatrix} \quad , r = \text{roll}$$

$$D_p = \begin{pmatrix} \text{Cos } p & 0 & -\text{Sin } p \\ 0 & 1 & 0 \\ \text{Sin } p & 0 & \text{Cos } p \end{pmatrix} \quad , p = \text{pitch}$$

$$D_y = \begin{pmatrix} \text{Cos } y & \text{Sin } y & 0 \\ -\text{Sin } y & \text{Cos } y & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad , y = \text{yaw}$$

1. If $d_{11} = d_{12} = 0$ or $d_{23} = d_{33} = 0$

$$y = \text{Tan}^{-1} \frac{-d_{21}}{d_{22}} \quad \text{and} \quad r = 0.0.$$

2. Otherwise

$$y = \text{Tan}^{-1} \frac{-d_{12}}{d_{11}} \quad \text{and} \quad r = \text{Tan}^{-1} \frac{d_{23}}{d_{33}} \ .$$

3. Compute $p = -\text{Sin}^{-1} d_{13}$.

4. y, p and r are converted to degrees, stored into A and returned to the calling routine.

8.0 CVS IV PROGRAM SOURCE LISTING

```
      BLOCK DATA                                                        COMMON  0010
C                                                          REV 20 05/18/80  COMMON  0020
      IMPLICIT REAL*8 (A-H,O-Z)                                         COMMON  0030
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           COMMON  0040
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)        COMMON  0050
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         COMMON  0060
     *               UNITL,UNITM,UNITT,GRAVTY(3)                        COMMON  0070
      COMMON/JBARTZ/ MNPL(   30),MNBLT(    8),MNSEG(   30),MNBAG(    6),COMMON  0080
     *               MPL(3,5,30),MBLT(3,5,8),MSEG(3,5,30),MBAG(3,10,6), COMMON  0090
     *               NTPL( 5,30),NTBLT( 5,8),NTSEG( 5,30)               COMMON  0100
      COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),           COMMON  0110
     *               BLTTTL(5,8),PLTTL(5,30),BAGTTL(5,6),SEG(30),       COMMON  0120
     *               JOINT(30),CGS(30),JS(30)                           COMMON  0130
      REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT      COMMON  0140
      LOGICAL*1 CGS,JS                                                  COMMON  0150
      COMMON/FORCES/ PSF(7,30),BSF(4,20),SSF(10,20),BAGSF(3,20),        COMMON  0160
     *               PRJNT(7,30),NPANEL(5),NPSF,NBSF,NSSF,NBGSF         COMMON  0170
      COMMON/RSAVE/  XSG(3,20,3),DPMI(3,3,30),LPMI(30),NSG(7),MSG(20,7) COMMON  0180
      COMMON/CDINT/  UU(4),GH(3,4),                                     COMMON  0190
     *               E(3,240),FF(5,240),GG(5,240),Y(5,240),U(5,240),    COMMON  0200
     *               H,HPRINT,TSAVE,TPRINT,TSTART,ICNT,IDBL,IFLAG       COMMON  0210
      COMMON/DAMPER/ APSDM(3,20),APSDN(3,20),ASD(5,20),MSDM(20),MSDN(20)COMMON  0220
      COMMON/HRNESS/ BAR(15,100),BB(100),BBDOT(100),PLOSS(2,100),       COMMON  0230
     *               XLONG(20),HTIME(2),IBAR(5,100),NL(2,100),          COMMON  0240
     *               NPTSPB(20),NPTPLY(20),NTHRNS(20),NBLTPH(5)         COMMON  0250
C     NOTE:  FF REPLACES F.                                             COMMON  0260
      COMMON/TEMPVS/ JTMPVS(10538)                                      COMMON  0270
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),COMMON  0280
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       COMMON  0290
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(3,60),HA(3,60),HB(3,60), COMMON  0300
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),     COMMON  0310
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)    COMMON  0320
      COMMON/CNTSRF/ PL(17,30),BELT(20,8),TPTS(6,8),BD(24,40)           COMMON  0330
      COMMON/TABLES/ MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(500),TAB(2600)COMMON  0340
      COMMON/VPOSTN/ ZPLT(3),SPLT(3),AXV(3,6),VATAB(6,101,6),           COMMON  0350
     *               VT0(6),VDT(6),TIMEV(6),OMEGV(6),NVTAB(6),INDXV(6)  COMMON  0360
      COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60),COMMON  0370
     *               F(3,30),TQ(3,30),WJ(30)                            COMMON  0380
      COMMON/CEULER/ IEULER(30),HIR(3,3,30),ANG(3,30),ANGD(3,30),       COMMON  0390
     *               FE(3,30),TQE(3,30),CONST(3,30)                     COMMON  0400
      COMMON/FLXBLE/ HF(4,12,8),B42(3,3,24),V4(3,8),NFLEX(3,8)          COMMON  0410
      COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),   COMMON  0420
     *               HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12),COMMON  0430
     *               RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),              COMMON  0440
     *               KQ1(12),KQ2(12),KQTYPE(12)                         COMMON  0450
      COMMON/TEMPVI/ CREST,TTI(3),R1I(3),R2I(3),JSTOP(4,2,30)           COMMON  0460
      COMMON/INTEST/ SGTEST(3,4,30),XTEST(3,120),SEGT(120),REGT(120)    COMMON  0470
      REAL    SEGT                                                      COMMON  0480
      COMMON/COMAIN/ VAR(240),DER(240),DT,H0,HMAX,HMIN,RSTIME,          COMMON  0490
     *               ISTEP,NSTEPS,NDINT,NEQ,IRSIN,IRSOUT                COMMON  0500
      COMMON/ABDATA/ ZDEP(3,5),DBR(3,3,5),DPVCTR(3,5),DEPLOY(3,5),      COMMON  0510
     *               AB(3,5),B(9,4,5),ZR(3,4,5),BFB(3,4,5),DRR(9,4,5),  COMMON  0520
     *               VBAGG(5),VSCS(5),SPRK(5),CK(5),CMASS(5),CYMIN(5),  COMMON  0530
     *               CYMOUT(5),BAGPV(5),PD(5),VBAG(5),VOLBP(5),         COMMON  0540
     *               PCYV(5),PCYMIN(5),PVBAG(5),TV1(3,4,5),TV2(3,10,5), COMMON  0550
     *               SWITCH(5),PYMOUT(5),SCALE(5),PREVT,IFULL(6)        COMMON  0560
      COMMON/CYDATA/ CYTD(5),CYPA(5),CYSP(5),CYT0(5),CYV0(5),CYCD(5),   COMMON  0570
     *               CYK(5),CYR(5),CYAT(5),CYPV(5),CYCD0(5),CYA0(5),    COMMON  0580
     *               CYP0(5),CYSS(5),CYL0(5),CYC(5),CYRHO0(5),CYVMAX(5),COMMON  0590
     *               CYORFC(5),CYRHO(5),CYT(5),CYP(5),CYV(5)            COMMON  0600
      COMMON/WINDFR/ WTIME(30),QFU(3,5),QFV(3,5),                       COMMON  0610
     *               IWIND(30),MWSEG(5,30),NFVSEG(6),NFVNT(5)           COMMON  0620
      END                                                               COMMON  0630
```

```
C     CALSPAN THREE DIMENSIONAL CRASH VICTIM SIMULATION PROGRAM          MAIN3D  0010
C                                                        REV 20 04/11/80  MAIN3D  0020
C     MAIN PROGRAM                                                       MAIN3D  0030
C                                                                        MAIN3D  0040
C     PERFORMS CARD INPUT, PROGRAM INITIALIZATION,                       MAIN3D  0050
C     CONTROL OF INTEGRATION LOOP AND OPTIONAL OUTPUT.                   MAIN3D  0060
C                                                                        MAIN3D  0070
      IMPLICIT REAL*8(A-H,O-Z)                                           MAIN3D  0080
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,            MAIN3D  0090
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)         MAIN3D  0100
      COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),            MAIN3D  0110
     *               BLTTTL(5,8),PLTTL(5,30),BAGTTL(5,6),SEG(30),        MAIN3D  0120
     *               JOINT(30),CGS(30),JS(30)                            MAIN3D  0130
      REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT       MAIN3D  0140
      LOGICAL*1 CGS,JS                                                   MAIN3D  0150
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                          MAIN3D  0160
     *               UNITL,UNITM,UNITT,GRAVTY(3)                         MAIN3D  0170
      COMMON/COMAIN/ VAR(240),DER(240),DT,HO,HMAX,HMIN,RSTIME,           MAIN3D  0180
     *               ISTEP,NSTEPS,NDINT,NEQ,IRSIN,IRSOUT                 MAIN3D  0190
      LOGICAL NPRT1,NPRT2,NPRT3                                          MAIN3D  0200
      CALL ELTIME(1, 1)                                                  MAIN3D  0210
C                                                                        MAIN3D  0220
C     WRITE PROLOGUE ON PRIMARY OUTPUT UNIT.                             MAIN3D  0230
C                                                                        MAIN3D  0240
      WRITE  (6,11)                                                      MAIN3D  0250
   11 FORMAT('1',30X,'CALSPAN 3-D CRASH VICTIM SIMULATION PROGRAM'  ////  MAIN3D  0260
     * 31X,'DEVELOPED BY CALSPAN CORP., P.O. BOX 400, BUFFALO NY 14225'/  MAIN3D  0270
     */31X,'FOR THE NATIONAL HIGHWAY TRAFFIC SAFETY ADMINISTRATION,'   /  MAIN3D  0280
     * 31X,'U.S. DEPARTMENT OF TRANSPORTATION, UNDER CONTRACTS'        /  MAIN3D  0290
     * 31X,'FH-11-7592, HS-053-2-485, HS-6-01300 AND HS-6-01410;'     //  MAIN3D  0300
     * 31X,'AND FOR THE AEROSPACE MEDICAL RESEARCH LABORATORY,'       /  MAIN3D  0310
     * 31X,'AEROSPACE MEDICAL DIVISION, WRIGHT-PATTERSON AIR FORCE BASE'  MAIN3D  0320
     */31X,'UNDER CONTRACTS F33615-75C-5002 AND F33615-78C-0516.'   ////  MAIN3D  0330
     * 31X,'PROGRAM DOCUMENTATION - NHTSA REPORT NOS. DOT-HS-801-507' /  MAIN3D  0340
     * 31X,'THROUGH 510 (FORMERLY CALSPAN REPORT NO. ZQ-5180-L-1)'    /  MAIN3D  0350
     * 31X,'AVAILABLE FROM NTIS (ACCESSION NOS. PB-241692,3,4 AND 5)' /  MAIN3D  0360
     * 31X,'AND APPENDIXES A-J TO THE ABOVE, AVAILABLE FROM CALSPAN;' /  MAIN3D  0370
     * 31X,'AND REPORT NO. AMRL-TR-75-14, AVAILABLE FROM NTIS.'     ////  MAIN3D  0380
     * 31X,'PROGRAM VERSION 20A, EXECUTED ON THE IBM 370/3031 COMPUTER'/  MAIN3D  0390
     * 31X,'CALSPAN COMPUTER CENTER, CALSPAN CORP, BUFFALO NY 14225.'  )  MAIN3D  0400
C                                                                        MAIN3D  0410
C     INPUT CARDS A.1 AND A.2, TEST FOR RESTART.                         MAIN3D  0420
C                                                                        MAIN3D  0430
      CALL BLKDTA                                                        MAIN3D  0440
      READ  (5,12)  DATE,IRSIN,IRSOUT,RSTIME,COMENT                      MAIN3D  0450
   12 FORMAT(3A4,2I4,F8.0/20A4/20A4)                                     MAIN3D  0460
      WRITE (6,13)  DATE,IRSIN,IRSOUT,RSTIME,COMENT                      MAIN3D  0470
   13 FORMAT(/////4X,3A4,'  IRSIN=',I4,'  IRSOUT=',I4,'  RSTIME =',F8.4, MAIN3D  0480
     *       61X,'CARDS A'//1X,20A4/1X,20A4//)                           MAIN3D  0490
      IF (IRSIN.NE.0) GO TO 18                                           MAIN3D  0500
```

```
C                                                                       MAIN3D  0510
C     INPUT CARDS A.3,A.4 AND A.5.                                      MAIN3D  0520
C                                                                       MAIN3D  0530
      READ  (5,14)  UNITL,UNITM,UNITT,GRAVTY,G                          MAIN3D  0540
   14 FORMAT(3A4,4F12.0)                                                MAIN3D  0550
      IF  (G.EQ.0.0)  G = DSQRT(GRAVTY(1)**2+GRAVTY(2)**2+GRAVTY(3)**2) MAIN3D  0560
      READ  (5,15)  NDINT,NSTEPS,DT,H0,HMAX,HMIN,NPRT                   MAIN3D  0570
   15 FORMAT(2I4,4F8.0/36I2)                                            MAIN3D  0580
      WRITE (6,16)  UNITL,UNITM,UNITT,GRAVTY,                           MAIN3D  0590
     *              NDINT,NSTEPS,DT,H0,HMAX,HMIN                       MAIN3D  0600
   16 FORMAT(5X,'UNITL = ',A4,5X,'UNITM = ',A4,5X,'UNITT = ',A4,        MAIN3D  0610
     *       5X,'GRAVITY VECTOR = (',F9.4,',',F9.4,',',F9.4,')'//       MAIN3D  0620
     *       5X,'NDINT =',I4,5X,'NSTEPS =',I5,5X,'DT =',F8.6,           MAIN3D  0630
     *       5X,'H0 =',F8.6,5X,'HMAX =',F8.6,5X,'HMIN =',F8.6)          MAIN3D  0640
      WRITE (6,17)  (I,I=1,36),NPRT                                     MAIN3D  0650
   17 FORMAT('0 NPRT ARRAY'/3X,36I3/3X,36I3)                            MAIN3D  0660
      NPRT4 = NPRT(4)                                                   MAIN3D  0670
      IF  (NPRT(4).LT.0)  GO TO 50                                      MAIN3D  0680
C                                                                       MAIN3D  0690
C     CALL INPUT ROUTINES                                               MAIN3D  0700
C                                                                       MAIN3D  0710
      CALL BINPUT                                                       MAIN3D  0720
      CALL VINPUT                                                       MAIN3D  0730
      CALL SINPUT                                                       MAIN3D  0740
      CALL CINPUT                                                       MAIN3D  0750
C                                                                       MAIN3D  0760
C     PROGRAM INITIALIZATION                                            MAIN3D  0770
C                                                                       MAIN3D  0780
      TIME = 0.0                                                        MAIN3D  0790
      CALL INITAL                                                       MAIN3D  0800
      GO TO 19                                                          MAIN3D  0810
C                                                                       MAIN3D  0820
C     READ INPUT DATA FROM RESTART TAPE AND WRITE NEW TAPE.             MAIN3D  0830
C     THE FIVE FUNCTIONS OF SUBROUTINE RSTART ARE:                      MAIN3D  0840
C        1. READ INPUT & INITIALIZATION RECORD FROM OLD RESTART TAPE.   MAIN3D  0850
C        2. WRITE INPUT & INITIALIZATION RECORD ONTO NEW RESTART TAPE.  MAIN3D  0860
C        3. READ TIME POINT RECORD FROM OLD RESTART TAPE.               MAIN3D  0870
C        4. READ NEW INPUT DATA FROM INPUT STREAM FOR RESTART.          MAIN3D  0880
C        5. WRITE TIME POINT RECORD ONTO NEW RESTART TAPE.              MAIN3D  0890
C                                                                       MAIN3D  0900
   18 CALL RSTART(1,IRSIN)                                              MAIN3D  0910
      CALL RSTART(4,5)                                                  MAIN3D  0920
      NPRT4 = NPRT(4)                                                   MAIN3D  0930
   19 IF (IRSOUT.NE.0) CALL RSTART(2,IRSOUT)                            MAIN3D  0940
C                                                                       MAIN3D  0950
C     INTEGRATION LOOP - ADVANCE TIME BY EITHER INTEGRATING THROUGH     MAIN3D  0960
C     SUBROUTINE DINT OR BY FETCHING TIME POINT RECORD FROM RESTART TAPE MAIN3D  0970
C                                                                       MAIN3D  0980
      TIME = 0.0                                                        MAIN3D  0990
      ISTEP = 0                                                         MAIN3D  1000
```

```
   20 IF (IRSIN.EQ.0) GO TO 23                                                  MAIN3D  1010
      IF (TIME.GT.RSTIME+0.5*DT) GO TO 23                                       MAIN3D  1020
      IF (DABS(TIME-RSTIME).LT.0.5*DT) GO TO 21                                 MAIN3D  1030
      CALL RSTART(3,IRSIN)                                                      MAIN3D  1040
      GO TO 24                                                                  MAIN3D  1050
   21 CALL RSTART(4,5)                                                          MAIN3D  1060
      IF  (NPRT(4).LT.0)  GO TO 50                                              MAIN3D  1070
   23 CALL DINT                                                                 MAIN3D  1080
C                                                                               MAIN3D  1090
C     OPTIONAL OUTPUT                                                           MAIN3D  1100
C     1. PRINTER PLOT ON OUTPUT UNIT 2 CONTROLLED BY NPRT(5) & (6).             MAIN3D  1110
C                                                                               MAIN3D  1120
   24 CALL PRIPLT                                                               MAIN3D  1130
C                                                                               MAIN3D  1140
C     2. RESTART DATA ON UNIT IRSOUT CONTROLLED BY IRSOUT # 0.                  MAIN3D  1150
C                                                                               MAIN3D  1160
      IF (IRSOUT.NE.0) CALL RSTART(5,IRSOUT)                                    MAIN3D  1170
C                                                                               MAIN3D  1180
C     3. SUBROUTINE PRINT ON PRIMARY OUTPUT UNIT CONTROLLED BE NPRT(3).         MAIN3D  1190
C                                                                               MAIN3D  1200
      NPRT3 = (NPRT(3).EQ.1)                                                    MAIN3D  1210
      IF (NPRT(3).GT.1) NPRT3 = (MOD(ISTEP,NPRT(3)).EQ.0)                       MAIN3D  1220
      IF (NPRT3) CALL PRINT(6HMAIN3D)                                           MAIN3D  1230
C                                                                               MAIN3D  1240
C     4. PROGRAM VIEW PLOT DATA ON UNIT 1 CONTROLLED BY NPRT(1).                MAIN3D  1250
C                                                                               MAIN3D  1260
      NPRT1 = (NPRT(1).EQ.1)                                                    MAIN3D  1270
      IF (NPRT(1).GT.1) NPRT1 = (MOD(ISTEP,NPRT(1)).EQ.0)                       MAIN3D  1280
      IF  (NPRT1)  CALL UNIT1(0)                                                MAIN3D  1290
C                                                                               MAIN3D  1300
C     5. SUBROUTINE ELTIME ON PRIMARY OUTPUT UNIT CONTROLLED BY NPRT(2).        MAIN3D  1310
C                                                                               MAIN3D  1320
      NPRT2 = (NPRT(2).EQ.1)                                                    MAIN3D  1330
      IF (NPRT(2).GT.1) NPRT2 = (MOD(ISTEP,NPRT(2)).EQ.0)                       MAIN3D  1340
      IF (NPRT2) CALL ELTIME(2,1)                                               MAIN3D  1350
C                                                                               MAIN3D  1360
C     END OF INTEGRATION LOOP.                                                  MAIN3D  1370
C                                                                               MAIN3D  1380
      ISTEP = ISTEP+1                                                           MAIN3D  1390
      IF (ISTEP.LE.NSTEPS) GO TO 20                                             MAIN3D  1400
C                                                                               MAIN3D  1410
C     6. SUBROUTINE POSTPR ON PRIMARY OUTPUT UNIT CONTROLLED BY NPRT(4).        MAIN3D  1420
C                                                                               MAIN3D  1430
   50 IF  (NPRT4.GT.0)  END FILE 8                                              MAIN3D  1440
      IF  (NPRT(4).EQ.0 .OR. NPRT(4).EQ.4)  GO TO 60                            MAIN3D  1450
      PRDT = 1000.0*DT                                                          MAIN3D  1460
      CALL POSTPR (PRDT)                                                        MAIN3D  1470
      IF  (NPRT2)  CALL ELTIME (2,1)                                            MAIN3D  1480
C                                                                               MAIN3D  1490
C     7. END OF RUN - CALL ELTIME IF NOT CALLED ABOVE.                          MAIN3D  1500
C                                                                               MAIN3D  1510
   60 IF  (.NOT.NPRT2)  CALL ELTIME (2,1)                                       MAIN3D  1520
      STOP  1                                                                   MAIN3D  1530
      END                                                                       MAIN3D  1540
```

```
      SUBROUTINE ADJUST (M,D1)                                          ADJUST  0010
C                                                        REV 19 09/18/79  ADJUST  0020
      IMPLICIT REAL*8 (A-H,O-Z)                                         ADJUST  0030
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         ADJUST  0040
     *               UNITL,UNITM,UNITT,GRAVTY(3)                        ADJUST  0050
      COMMON/CDINT/ UU(4),GH(3,4),                                      ADJUST  0060
     *              E(3,240), F(5,240),GG(5,240),Y(5,240),U(5,240),     ADJUST  0070
     *              H,HPRINT,HS,TPRINT,TSTART,ICNT,IDBL,IFLAG           ADJUST  0080
      COMMON/COMAIN/ VAR(240),DER(240),DT,HO,HMAX,HMIN,RSTIME,          ADJUST  0090
     *               ISTEP,NSTEPS,NDINT,NEQ,IRSIN,IRSOUT                ADJUST  0100
      IF  (M.NE.1)  GO TO 12                                            ADJUST  0110
C                                                                       ADJUST  0120
C     M = 1:                                                            ADJUST  0130
C                                                                       ADJUST  0140
      DO  11  I=1,NEQ                                                   ADJUST  0150
      W = VAR(I) - GG(1,I)                                              ADJUST  0160
      Z = DER(I) - GG(2,I)                                              ADJUST  0170
      ZZ = Z - GG(5,I)*W - GG(3,I)*UU(3) - GG(4,I)*UU(4)                ADJUST  0180
      GG(3,I) = GG(3,I) + ZZ*UU(1)                                      ADJUST  0190
      GG(4,I) = GG(4,I) + ZZ*UU(2)                                      ADJUST  0200
      Y(1,I) = VAR(I)                                                   ADJUST  0210
  11  Y(2,I) = DER(I)                                                   ADJUST  0220
      GO TO 99                                                          ADJUST  0230
  12  IF  (M.EQ.3)  GO TO 23                                            ADJUST  0240
C                                                                       ADJUST  0250
C     M = 2,4,5:                                                        ADJUST  0260
C                                                                       ADJUST  0270
      H1 = EPS(1)/H                                                     ADJUST  0280
      N2 = NEQ/2                                                        ADJUST  0290
      DO  20  I=1,NEQ,3                                                 ADJUST  0300
      ZA = 0.0                                                          ADJUST  0310
      IF  (I.LE.N2)  GO TO 20                                           ADJUST  0320
      IF  (M.EQ.4)   GO TO 16                                           ADJUST  0330
      VARX = VAR(I  ) - Y(1,I  )                                        ADJUST  0340
      VARY = VAR(I+1) - Y(1,I+1)                                        ADJUST  0350
      VARZ = VAR(I+2) - Y(1,I+2)                                        ADJUST  0360
      DERX = DER(I  ) - Y(2,I  )                                        ADJUST  0370
      DERY = DER(I+1) - Y(2,I+1)                                        ADJUST  0380
      DERZ = DER(I+2) - Y(2,I+2)                                        ADJUST  0390
      GO TO 17                                                          ADJUST  0400
  16  VARX = VAR(I  ) - U(1,I  )                                        ADJUST  0410
      VARY = VAR(I+1) - U(1,I+1)                                        ADJUST  0420
      VARZ = VAR(I+2) - U(1,I+2)                                        ADJUST  0430
      DERX = DER(I  ) - U(2,I  )                                        ADJUST  0440
      DERY = DER(I+1) - U(2,I+1)                                        ADJUST  0450
      DERZ = DER(I+2) - U(2,I+2)                                        ADJUST  0460
  17  U(3,I) = U(3,I) + VARX*DERX + VARY*DERY + VARZ*DERZ               ADJUST  0470
      U(4,I) = U(4,I) + VARX**2 + VARY**2 + VARZ**2                     ADJUST  0480
      IF  (U(4,I).NE.0.0)  ZA = U(3,I)/U(4,I)                           ADJUST  0490
      IF  (ZA.GT.H1)  ZA = H1                                           ADJUST  0500
```

```
      SUBROUTINE EQUILB (YPR,IYPR)                                          EQUILB  0010
C                                                          REV 19 10/19/79  EQUILB  0020
C                                                                           EQUILB  0030
C     ADJUSTS INITIAL INPUT POSITION PARAMETERS SUPPLIED ON CARDS G.2       EQUILB  0040
C     AND G.3 SUCH THAT INITIAL NORMAL CONTACT FORCES ARE EQUAL TO          EQUILB  0050
C     EITHER SUPPLIED VALUES OR THOSE COMPUTED BY CONSTRAINT FORCES.        EQUILB  0060
C                                                                           EQUILB  0070
      IMPLICIT REAL*8(A-H,O-Z)                                              EQUILB  0080
      DIMENSION  YPR(3,30) , IYPR(4,30)                                     EQUILB  0090
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,               EQUILB  0100
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)            EQUILB  0110
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),    EQUILB  0120
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)           EQUILB  0130
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(3,60),HA(3,60),HB(3,60),     EQUILB  0140
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),         EQUILB  0150
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)        EQUILB  0160
      COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60),    EQUILB  0170
     *               F(3,30),TQ(3,30),WJ(30)                                EQUILB  0180
      COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),       EQUILB  0190
     *               HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12),    EQUILB  0200
     *               RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),                  EQUILB  0210
     *               KQ1(12),KQ2(12),KQTYPE(12)                             EQUILB  0220
      COMMON/TABLES/ MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(500),TAB(2600)    EQUILB  0230
      COMMON/JBARTZ/ MNPL(   30),MNBLT(    8),MNSEG(   30),MNBAG(    6),    EQUILB  0240
     *               MPL(3,5,30),MBLT(3,5,8),MSEG(3,5,30),MBAG(3,10,6),     EQUILB  0250
     *               NTPL( 5,30),NTBLT( 5,8),NTSEG( 5,30)                   EQUILB  0260
      COMMON/CNTSRF/ PL(17,30),BELT(20,8),TPTS(6,8),BD(24,40)               EQUILB  0270
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                             EQUILB  0280
     *               UNITL,UNITM,UNITT,GRAVTY(3)                            EQUILB  0290
      COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),               EQUILB  0300
     *               BLTTTL(5,8),PLTTL(5,30),BAGTTL(5,6),SEG(30),           EQUILB  0310
     *               JOINT(30),CGS(30),JS(30)                               EQUILB  0320
      REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT          EQUILB  0330
      LOGICAL*1 CGS,JS                                                      EQUILB  0340
      COMMON/TEMPVS/ DMNT(3,3),XMN(3),XMM(3),TM(3),RM(3),                   EQUILB  0350
     *               TEMP(3),T(5),FX(10),FX1(10)                            EQUILB  0360
      DIMENSION  X(10),GX(10),DX(10),DXP(10),DPN(5,10)                      EQUILB  0370
      DIMENSION  JPL(10),JSG(10),JX(10),M1(10),M2(10),M3(10),MT(10)         EQUILB  0380
      DIMENSION  NTV(10),NI1(10),NSG(10),NAV(10),KSG(5,10)                  EQUILB  0390
      DIMENSION  ISG(5),IPL(5),LTYPE(5),INDGX(5),NTNQ(5)                    EQUILB  0400
      DIMENSION  SX(10),SGX(10),XDEV(10),WORD(2)                            EQUILB  0410
      DATA       BLANK/'      '/ , WORD/' SEGLP' , '   YPR'/               EQUILB  0420
      CALL ELTIME (1,35)                                                    EQUILB  0430
C                                                                           EQUILB  0440
C     INPUT CARDS G.4, G.5.A-G.5.N, AND G.6.A-G.6.M                         EQUILB  0450
C                                                                           EQUILB  0460
      READ   (5,60)  NVAR,NCON                                              EQUILB  0470
      WRITE  (6,51)  NVAR,NCON                                              EQUILB  0480
   51 FORMAT('1',5X,'NVAR =',I3,3X,'NCON =',I3,93X,'CARD  G.4'/ )           EQUILB  0490
      ICARD = 4                                                             EQUILB  0500
```

```
C     CALL AT END OF NTH SUBROUTINE.                                    ELTIME  0510
C                                                                       ELTIME  0520
   30 MTOUT = LTIME(1)                                                  ELTIME  0530
      NDIFF = MTOUT-MTIN(N)                                             ELTIME  0540
      MTIN(N) = -1                                                      ELTIME  0550
      IF (NDIFF.EQ.0) GO TO 32                                          ELTIME  0560
      NT(N) = NT(N) + NDIFF                                             ELTIME  0570
      DO 31 I=1,40                                                      ELTIME  0580
      IF (MTIN(I).NE.-1) MTIN(I) = MTIN(I) + NDIFF                      ELTIME  0590
   31 CONTINUE                                                          ELTIME  0600
   32 GO TO 99                                                          ELTIME  0610
C                                                                       ELTIME  0620
C     SUBSEQUENT CALLS FROM MAIN PROGRAM, PRINT SUMMARY TABLE.          ELTIME  0630
C                                                                       ELTIME  0640
   40 NTSUM = LTIME(1)                                                  ELTIME  0650
      NT(1) = NTSUM - MTIN(1)                                           ELTIME  0660
      TIME = FLOAT(NTSUM)/100.0                                         ELTIME  0670
      WRITE (6,41) TIME                                                 ELTIME  0680
   41 FORMAT('1 ELAPSED CPU TIME =',F10.2,' SECONDS'//                  ELTIME  0690
     *    '     SUB         CALLS       TIME        %     '//)         ELTIME  0700
      PCSUM = 0.0                                                       ELTIME  0710
      NTSUM = 0                                                         ELTIME  0720
      DO 42 I=1,NSUB                                                    ELTIME  0730
      J = IND(I)                                                        ELTIME  0740
      PC = FLOAT(NT(J))/TIME                                            ELTIME  0750
      PCSUM = PCSUM + PC                                                ELTIME  0760
      NTSUM = NTSUM + NT(J)                                             ELTIME  0770
   42 WRITE (6,43) SUB(J),NC(J),NT(J),PC                                ELTIME  0780
   43 FORMAT(A10,2I10,F10.2)                                            ELTIME  0790
      WRITE (6,44) NTSUM,PCSUM                                          ELTIME  0800
   44 FORMAT('0TOTAL',14X,I10,F10.2)                                    ELTIME  0810
   99 RETURN                                                            ELTIME  0820
      END                                                               ELTIME  0830
```

```
      SUBROUTINE ELTIME(L,N)                                                ELTIME  0010
C                                                          REV 19 09/18/79  ELTIME  0020
C     COUNTS THE NUMBER OF TIMES CERTAIN BASIC SUBROUTINES ARE CALLED       ELTIME  0030
C     AND ACCOUNTS FOR ALL COMPUTER CPU TIME USED BY THESE ROUTINES.        ELTIME  0040
C                                                                           ELTIME  0050
C       ARGUMENTS L: 1 INDICATES CALL IS AT START OF ROUTINE                ELTIME  0060
C                    2 INDICATES CALL IS AT END OF ROUTINE.                 ELTIME  0070
C                  N: THE SUBROUTINE IDENTIFICATION NUMBER.                 ELTIME  0080
C                                                                           ELTIME  0090
C       ASSUMES FUNCTION LTIME(1) IS GIVING ELAPSED CPU TIME IN INTEGER     ELTIME  0100
C       UNITS OF 0.01 SECONDS SINCE FUNCTION LTIME(0) WAS CALLED.           ELTIME  0110
C                                                                           ELTIME  0120
      DIMENSION NT(40),MTIN(40),NC(40),IND(40)                              ELTIME  0130
      REAL*8 SUB(40)                                                        ELTIME  0140
      DATA   SUB/                                                           ELTIME  0150
     *   8H MAIN3D ,8H INPUT  ,8H DINT   ,8H PRIPLT ,8H DZP    ,           ELTIME  0160
     *   8H PDAUX  ,8H UPDATE ,8H OUTPUT ,8H DAUX   ,8H SETUP1 ,           ELTIME  0170
     *   8H CHAIN  ,8H CONTCT ,8H VISPR  ,8H DAUX11 ,8H DAUX12 ,           ELTIME  0180
     *   8H DAUX22 ,8H DAUX31 ,8H DAUX32 ,8H DAUX33 ,8H FSMSOL ,           ELTIME  0190
     *   8H PLELP  ,8H BELTRT ,8H SEGSEG ,8H AIRBAG ,8H RSTART ,           ELTIME  0200
     *   8H SETUP2 ,8H IMPULS ,8H IMPLS2 ,8H AIRBG3 ,8H DAUX55 ,           ELTIME  0210
     *   8H EJOINT ,8H SPDAMP ,8H DAUX44 ,8H FLXSEG ,8H EQUILB ,           ELTIME  0220
     *   8H POSTPR ,8H WINDY  ,8H HBELT  ,8H HPTURB ,8H        /           ELTIME  0230
      IF (N.GT.1) GO TO 20                                                  ELTIME  0240
      IF (L.GT.1) GO TO 40                                                  ELTIME  0250
C                                                                           ELTIME  0260
C     INITIAL CALL AT BEGINNING OF MAIN PROGRAM.                            ELTIME  0270
C                                                                           ELTIME  0280
      MTIN(1) = LTIME(0)                                                    ELTIME  0290
      DO 11 I=1,40                                                          ELTIME  0300
      IND(I)  =  0                                                          ELTIME  0310
      NC(I)   =  0                                                          ELTIME  0320
      MTIN(I) = -1                                                          ELTIME  0330
   11 NT(I)   =  0                                                          ELTIME  0340
      NSUB    =  1                                                          ELTIME  0350
      IND(1)  =  1                                                          ELTIME  0360
      NC(1)   =  1                                                          ELTIME  0370
      MTIN(1) =  0                                                          ELTIME  0380
      GO TO 99                                                              ELTIME  0390
C                                                                           ELTIME  0400
C      CALL AT BEGINNING OF NTH SUBROUTINE.                                 ELTIME  0410
C                                                                           ELTIME  0420
   20 IF (L.GT.1) GO TO 30                                                  ELTIME  0430
      MTIN(N) = LTIME(1)                                                    ELTIME  0440
      IF (NC(N).NE.0) GO TO 21                                              ELTIME  0450
      NSUB = NSUB+1                                                         ELTIME  0460
      IND(NSUB) = N                                                         ELTIME  0470
   21 NC(N) = NC(N)+1                                                       ELTIME  0480
      GO TO 99                                                              ELTIME  0490
C                                                                           ELTIME  0500
```

```
      DOUBLE PRECISION FUNCTION ELONG(A,B,C,D,E)                        ELONG   0010
C                                                     REV 01 10/05/72   ELONG   0020
C     COMPUTES ARC LENGTH OF ELLIPSE       AX**2 + 2BXY + CY**2 = 1     ELONG   0030
C     FROM THETA=0 (POSITIVE X AXIS) TO THETA=E (RADIANS)               ELONG   0040
C     WHERE D IS NOMINAL INCREMENT OF INTEGRATION.                      ELONG   0050
C                                                                       ELONG   0060
      IMPLICIT REAL*8(A-H,O-Z)                                          ELONG   0070
       N=DABS(E/D)                                                      ELONG   0080
       N=N+N                                                            ELONG   0090
       IF(N.EQ.0)N=2                                                    ELONG   0100
       Z=N                                                              ELONG   0110
       T=E/Z                                                            ELONG   0120
       F = DSQRT ((1.+(B/A)**2)/A)                                      ELONG   0130
       CS=1.                                                            ELONG   0140
       SN=0.                                                            ELONG   0150
       DCS=DCOS(T)                                                      ELONG   0160
       DSN=DSIN(T)                                                      ELONG   0170
       S=F/2.                                                           ELONG   0180
      AC = A+C                                                          ELONG   0190
      BAC = B*B-A*C                                                     ELONG   0200
       DO 10 I=1,N,2                                                    ELONG   0210
       CSS=CS*DCS-SN*DSN                                                ELONG   0220
       SN=SN*DCS+CS*DSN                                                 ELONG   0230
       CS=CSS                                                           ELONG   0240
       G=(A*CS+B*SN)*CS+(B*CS+C*SN)*SN                                  ELONG   0250
      G = G**2/(AC + BAC/G)                                             ELONG   0260
       F=(F+1./(F*G))/2.                                                ELONG   0270
       S=S+F+F                                                          ELONG   0280
       CSS=CS*DCS-SN*DSN                                                ELONG   0290
       SN=SN*DCS+CS*DSN                                                 ELONG   0300
       CS=CSS                                                           ELONG   0310
       G=(A*CS+B*SN)*CS+(B*CS+C*SN)*SN                                  ELONG   0320
      G = G**2/(AC + BAC/G)                                             ELONG   0330
       F=(F+1./(F*G))/2.                                                ELONG   0340
       S=S+F                                                            ELONG   0350
   10  CONTINUE                                                         ELONG   0360
       ELONG=(S+S-F)*T/3.                                               ELONG   0370
       RETURN                                                           ELONG   0380
      END                                                               ELONG   0390
```

```
      HD3(3) = TH(3,3)                                                  EJOINT  1510
      CALL GLOBAL (J,HD3,DH1,TQC,T9,ANGL)                               EJOINT  1520
   34 CONTINUE                                                          EJOINT  1530
C                                                                       EJOINT  1540
C     ADD TORQUE CONVERTED TO LOCAL REFERENCE TO U2 ARRAY BY            EJOINT  1550
C      U2(M   ) = U2(M   ) + D(M   )*TQ                                 EJOINT  1560
C      U2(J+1) = U2(J+1) - D(J+1)*TQ                                    EJOINT  1570
C                                                                       EJOINT  1580
   35 DO  51  I=1,3                                                     EJOINT  1590
      TQ(I,J) = TQE(I,J)+TQC*T9(I)                                      EJOINT  1600
      TTI(I) = TQ(I,J)                                                  EJOINT  1610
      DO 51 K=1,3                                                       EJOINT  1620
      U2(K,M   ) = U2(K,M   ) + D(K,I,M   )*TQ(I,J)                     EJOINT  1630
   51 U2(K,J+1) = U2(K,J+1) - D(K,I,J+1)*TQ(I,J)                        EJOINT  1640
                                                                        EJOINT  1650
C                                                                       EJOINT  1660
C     STORE DATA INTO PRJNT ARRAY FOR OUTPUT ROUTINE                    EJOINT  1670
C                                                                       EJOINT  1680
   97 PRJNT(1,J) = IEULER(J)                                            EJOINT  1690
      PRJNT(2,J) = ANG(1,J)                                             EJOINT  1700
      PRJNT(3,J) = ANG(2,J)                                             EJOINT  1710
      PRJNT(4,J) = ANG(3,J)                                             EJOINT  1720
      PRJNT(5,J) = CS(1)**2 + CS(3)**2 + 2.0*CS(1)*CS(3)*TH(3,3)        EJOINT  1730
      PRJNT(6,J) = CV(1)**2 + CV(3)**2 + 2.0*CV(1)*CV(3)*TH(3,3)        EJOINT  1740
      PRJNT(7,J) = TQ(1,J)**2 + TQ(2,J)**2 + TQ(3,J)**2                 EJOINT  1750
   98 CONTINUE                                                          EJOINT  1760
      CALL ELTIME(2,31)                                                 EJOINT  1770
   99 RETURN                                                            EJOINT  1780
      END
```

```
      IF (DABS(HX).GE.EPS(6)) GO TO 20                                                                  EJOINT  1010
      SH(1) = ANGD(1,J)                                                                                 EJOINT  1020
      SH(3) = ANGD(3,J)                                                                                 EJOINT  1030
      GO TO 21                                                                                          EJOINT  1040
   20 CALL DOT31 (H2,WMJ,SH)                                                                            EJOINT  1050
      SH(1) = SH(1)/HX                                                                                  EJOINT  1060
      IF  (N.EQ.2)  SH(2) = 0.0                                                                         EJOINT  1070
      SH(3) = SH(3)/HX                                                                                  EJOINT  1080
   21 DO 22 I=1,3                                                                                       EJOINT  1090
      ANGD(I,J) = SH(I)                                                                                 EJOINT  1100
   22 HDT(I,2) = HDT(I,2) + SH(1)*H2(I,3)                                                               EJOINT  1110
      IF (NJ.NE.0) N = IJ+3                                                                             EJOINT  1120
      IF (N.GT.3) GO TO 30                                                                              EJOINT  1130
      N4 = 4-N                                                                                          EJOINT  1140
      IF (N.EQ.2) AHDT = HDT(1,2)*WMJ(1)+HDT(2,2)*WMJ(2)+HDT(3,2)*WMJ(3)                                EJOINT  1150
      IF (N.NE.2) AHDT = -(SH(2)*HDT(1,2)+SH(N4)*HDT(1,N4))*H2(1,N)                                     EJOINT  1160
     *                   -(SH(2)*HDT(2,2)+SH(N4)*HDT(2,N4))*H2(2,N)                                     EJOINT  1170
     *                   -(SH(2)*HDT(3,2)+SH(N4)*HDT(3,N4))*H2(3,N)                                     EJOINT  1180
      CALL MAT31 (D(1,1,M  ),H2(1,N),HB(1,2*J-1))                                                       EJOINT  1190
      CALL MAT31 (D(1,1,J+1),H2(1,N),HB(1,2*J  ))                                                       EJOINT  1200
      DO 25 I=1,3                                                                                       EJOINT  1210
      V2(I,J) = AHDT*H2(I,N)                                                                            EJOINT  1220
   25 IF (N.EQ.I) LSKIP(I) = .TRUE.                                                                     EJOINT  1230
      GO TO 42                                                                                          EJOINT  1240
   30 IF (N.GT.6) GO TO 40                                                                              EJOINT  1250
      K3J = 3*J-2                                                                                       EJOINT  1260
      DO 32 I=1,3                                                                                       EJOINT  1270
      IF (NJ.EQ.0) GO TO 31                                                                             EJOINT  1280
      IF  (I.EQ.N-3)  CREST = VISC(7,K3J)                                                               EJOINT  1290
      TQE(I,J) = H2(I,N-3)                                                                              EJOINT  1300
      GO TO 32                                                                                          EJOINT  1310
   31 V2(I,J) = -HDT(I,N-3)*AD(N-3)                                                                     EJOINT  1320
      HB(I,2*J-1) = HIM(I,N-3)                                                                          EJOINT  1330
      HB(I,2*J  ) = HIJ(I,N-3)                                                                          EJOINT  1340
      IF (I.NE.N-3) LSKIP(I) = .TRUE.                                                                   EJOINT  1350
   32 K3J = K3J + 1                                                                                     EJOINT  1360
      IF  (NJ)  35,42,35                                                                                EJOINT  1370
   40 IF (N.EQ.7) GO TO 97                                                                              EJOINT  1380
   42 DO 41 I=1,3                                                                                       EJOINT  1390
      IF  (LSKIP(I))  GO TO 41                                                                          EJOINT  1400
      K3J = 3*J-3+I                                                                                     EJOINT  1410
      CV(I) = ANGD(I,J)*VISCOS(DABS(ANGD(I,J)),VISC(1,K3J),HA(I,2*J))                                   EJOINT  1420
      CS(I) = EFUNCT(ANG(I,J),ANGD(I,J),SPRING(1,K3J),JSTOP(I,1,J))                                     EJOINT  1430
      FE(I,J) = CS(I) + CV(I) + HA(I,2*J)*HA(I,2*J-1)                                                   EJOINT  1440
   41 CONTINUE                                                                                          EJOINT  1450
   50 CALL MAT31(HIR(1,1,J),FE(1,J),TQE(1,J))                                                           EJOINT  1460
      IF(NJ.GT.0) GO TO 34                                                                              EJOINT  1470
   55 IF (IGLOB(J).EQ.0) GO TO 34                                                                       EJOINT  1480
      HD3(1) = TH(3,1)                                                                                  EJOINT  1490
      HD3(2) = TH(3,2)                                                                                  EJOINT  1500
```

```
      IC = IEULER(J)                                                    EJOINT  0510
      CALL EULRAD (TH,ANG(1,J),IC)                                      EJOINT  0520
      CALL ROT(H2,3,-ANG(1,J))                                          EJOINT  0530
      DO 13 I=1,3                                                       EJOINT  0540
      ANG(I,J) = ANG(I,J) - CONST(I,J)                                  EJOINT  0550
      HIR(I,1,J) = DH1(I,3)                                             EJOINT  0560
      HIR(I,3,J) = DH4(I,3)                                             EJOINT  0570
      HIM(I,1) = HT(I,3,2*J-1)                                          EJOINT  0580
      HIJ(I,3) = HT(I,3,2*J)                                            EJOINT  0590
      LSKIP(I) = .FALSE.                                                EJOINT  0600
      FE(I,J) = 0.0                                                     EJOINT  0610
      CV(I)   = 0.0                                                     EJOINT  0620
      CS(I)   = 0.0                                                     EJOINT  0630
      V2(I,J) = 0.0                                                     EJOINT  0640
      TQE(I,J) = 0.0                                                    EJOINT  0650
   13 TQ(I,J) = 0.0                                                     EJOINT  0660
      WJ(J)   = 0.0                                                     EJOINT  0670
      TQC = 0.0                                                         EJOINT  0680
      IF (IJ.EQ.4) GO TO 55                                             EJOINT  0690
      CALL MAT31 (HT(1,1,2*J-1),H2(1,1),HIM(1,2))                       EJOINT  0700
      CALL MAT31 (HT(1,1,2*J-1),H2(1,2),HIM(1,3))                       EJOINT  0710
      CALL DOT31 (D(1,1,M),HIM(1,2),H2(1,2))                            EJOINT  0720
      CALL DOT31 (D(1,1,M),HIM(1,3),H2(1,3))                            EJOINT  0730
      CALL CROSS (H2(1,2),HIR(1,3,J),H2(1,1))                           EJOINT  0740
      CALL DOT31 (D(1,1,M  ),WMEG(1,M  ),TM)                            EJOINT  0750
      CALL DOT31 (D(1,1,J+1),WMEG(1,J+1),TJ)                            EJOINT  0760
      SWJ = 0.0                                                         EJOINT  0770
      DO 14 I=1,3                                                       EJOINT  0780
      HIR(I,2,J) = H2(I,2)                                              EJOINT  0790
      WMJ(I) = TJ(I) - TM(I)                                            EJOINT  0800
   14 SWJ = SWJ + WMJ(I)**2                                             EJOINT  0810
      WJ(J) = DSQRT(SWJ)                                                EJOINT  0820
      CALL DOT31 (HIR(1,1,J),WMJ,AD)                                    EJOINT  0830
      CALL CROSS (TM,HIR(1,1,J),HDT(1,1))                               EJOINT  0840
      CALL CROSS (TM,HIR(1,2,J),HDT(1,2))                               EJOINT  0850
      CALL CROSS (TJ,HIR(1,3,J),HDT(1,3))                               EJOINT  0860
      CALL MAT31 (D(1,1,J+1),HIR(1,1,J),HIJ(1,1))                       EJOINT  0870
      CALL MAT31 (D(1,1,J+1),HIR(1,2,J),HIJ(1,2))                       EJOINT  0880
      CALL MAT31 (D(1,1,M  ),HIR(1,3,J),HIM(1,3))                       EJOINT  0890
      N = IEULER(J)                                                     EJOINT  0900
      DO 15 I=1,3                                                       EJOINT  0910
   15 SH(I) = AD(I)                                                     EJOINT  0920
      IF (N.EQ.8) GO TO 19                                              EJOINT  0930
      IF (N.GT.3) GO TO 16                                              EJOINT  0940
      SH(N) = 0.0                                                       EJOINT  0950
      GO TO 18                                                          EJOINT  0960
   16 DO 17 I=1,3                                                       EJOINT  0970
   17 IF (I.NE.N-3) SH(I) = 0.0                                         EJOINT  0980
   18 IF (N.NE.2) GO TO 21                                              EJOINT  0990
   19 HX = H2(1,1)*HIR(1,1,J) + H2(2,1)*HIR(2,1,J) + H2(3,1)*HIR(3,1,J) EJOINT  1000
```

```
      SUBROUTINE EJOINT(IJ,NJ)                                          EJOINT  0010
C                                                          REV 20 04/11/80  EJOINT  0020
CC    COMPUTES THE TORQUES ACTING ON AN EULER JOINT                     EJOINT  0030
CC    AND ADDS THEM TO THE U2 ARRAY.                                    EJOINT  0040
CC                                                                      EJOINT  0050
CC    ARGUMENTS:                                                        EJOINT  0060
CC       NJ = 0 - REGULAR COMPUTATION FOR ALL EULER JOINTS              EJOINT  0070
CC          # 0 - COMPUTE ONLY FOR JOINT NJ IMPULSE                     EJOINT  0080
CC                                                                      EJOINT  0090
CC       IJ = 1 IMPULSE ON PRECESSION AXIS ONLY                         EJOINT  0100
CC          = 2 IMPULSE ON NUTATION AXIS ONLY                           EJOINT  0110
CC          = 3 IMPULSE ON SPIN AXIS ONLY                               EJOINT  0120
CC          = 4 IMPULSE ON GLOBALGRAPHIC AXIS                           EJOINT  0130
C                                                                       EJOINT  0140
      IMPLICIT REAL*8(A-H,O-Z)                                          EJOINT  0150
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           EJOINT  0160
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)        EJOINT  0170
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30), EJOINT  0180
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       EJOINT  0190
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(3,60),HA(3,60),HB(3,60), EJOINT  0200
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),     EJOINT  0210
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)    EJOINT  0220
      COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60), EJOINT  0230
     *               F(3,30),TQ(3,30),WJ(30)                            EJOINT  0240
      COMMON/CEULER/ IEULER(30),HIR(3,3,30),ANG(3,30),ANGD(3,30),       EJOINT  0250
     *               FE(3,30),TQE(3,30),CONST(3,30)                     EJOINT  0260
      COMMON/FORCES/ PSF(7,30),BSF(4,20),SSF(10,20),BAGSF(3,20),        EJOINT  0270
     *               PRJNT(7,30),NPANEL(5),NPSF,NBSF,NSSF,NBGSF         EJOINT  0280
      COMMON/TEMPVI/ CREST,TTI(3),R1I(3),R2I(3),JSTOP(4,2,30)           EJOINT  0290
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         EJOINT  0300
     *               UNITL,UNITM,UNITT,GRAVTY(3)                        EJOINT  0310
      COMMON/TEMPVS/ DH1(3,3),DH4(3,3),TH(3,3),HIM(3,3),HIJ(3,3),       EJOINT  0320
     *               HDT(3,3),H2(3,3),SH(3),TM(3),TJ(3),WMJ(3),AD(3),   EJOINT  0330
     *               CV(3),CS(3),ANGL(3),HD3(3),CC(3),T9(3),LSKIP(3)    EJOINT  0340
      LOGICAL LSKIP                                                     EJOINT  0350
      IF (NJNT.LE.0) GO TO 99                                           EJOINT  0360
      CALL ELTIME(1,31)                                                 EJOINT  0370
      J1 = 1                                                            EJOINT  0380
      J2 = NJNT                                                         EJOINT  0390
      IF (NJ.EQ.0) GO TO 11                                             EJOINT  0400
      J1 = NJ                                                           EJOINT  0410
      J2 = NJ                                                           EJOINT  0420
   11 DO 98 J=J1,J2                                                     EJOINT  0430
      IF (IABS(IPIN(J)).NE.4) GO TO 98                                  EJOINT  0440
      M = IABS(JNT(J))                                                  EJOINT  0450
      CALL DOT33(D(1,1,M),HT(1,1,2*J-1),DH1)                            EJOINT  0460
      CALL DOT33(D(1,1,J+1),HT(1,1,2*J),DH4)                            EJOINT  0470
      CALL DOT33(DH4,DH1,TH)                                            EJOINT  0480
      DO 12 I=1,3                                                       EJOINT  0490
   12 ANG(I,J) = ANG(I,J) + CONST(I,J)                                  EJOINT  0500
```

```
      DOUBLE PRECISION FUNCTION EFUNCT (TH,THD,SPR,JSTOP)                    EFUNCT  0010
C                                                          REV 20 04/29/80    EFUNCT  0020
C     COMPUTES NONLINEAR SRRING TORQUE FOR EULER JOINTS.                      EFUNCT  0030
C                                                                             EFUNCT  0040
C     ARGUMENTS:                                                              EFUNCT  0050
C          TH    - THETA IS THE ANGLE OF THE EULER AXIS                       EFUNCT  0060
C          THD   - THETA DOT                                                  EFUNCT  0070
C          SPR   - ARRAY OF 5 VALUES DESCRIBING FUNCTION EVALUATION           EFUNCT  0080
C          JSTOP - INDICATOR TO BE SET TO ONE IF IN STOP                      EFUNCT  0090
C                                                                             EFUNCT  0100
      IMPLICIT REAL*8(A-H,O-Z)                                                EFUNCT  0110
      DIMENSION SPR(5)                                                        EFUNCT  0120
      JSTOP = 0                                                               EFUNCT  0130
      EFUNCT = TH*SPR(1)                                                      EFUNCT  0140
      TEN = 10.0                                                              EFUNCT  0150
      Q = DSIGN(TEN*THD,TH*THD)                                               EFUNCT  0160
      IF (Q.GT.1.0) Q = 1.0                                                   EFUNCT  0170
      IF (Q.LT.-1.0) Q = -1.0                                                 EFUNCT  0180
      X = 0.5*(1.0+SPR(4)+Q*(1.0-SPR(4)))                                     EFUNCT  0190
      IF (SPR(5).GT.0.0) GO TO 10                                             EFUNCT  0200
      EFUNCT = X*EFUNCT                                                       EFUNCT  0210
      GO TO 99                                                                EFUNCT  0220
   10 IF (DABS(TH).LT.SPR(5)) GO TO 99                                        EFUNCT  0230
      JSTOP = 1                                                               EFUNCT  0240
      Z = DABS(TH) - SPR(5)                                                   EFUNCT  0250
      EFUNCT = EFUNCT + DSIGN(X*(SPR(2)+Z*SPR(3))*Z**2,TH)                    EFUNCT  0260
   99 RETURN                                                                  EFUNCT  0270
      END                                                                     EFUNCT  0280
```

```
C                                     *          DXA                          EDEPTH  1510
C                                     C12 = XA'A---                           EDEPTH  1520
C                                     *          DU                           EDEPTH  1530
C                                                                             EDEPTH  1540
C                                     *               DXB                     EDEPTH  1550
C                                     C22 = (XB-M)'B---                       EDEPTH  1560
C                                     *               DU                      EDEPTH  1570
C                                                                             EDEPTH  1580
      C12 = 0.0                                                               EDEPTH  1590
      C22 = 0.0                                                               EDEPTH  1600
      DO 26 I=1,3                                                             EDEPTH  1610
      PXBU(I) = PXAU(I) + XL*(A(I,1)*PXAU(1)                                  EDEPTH  1620
     *     + A(I,2)*PXAU(2) + A(I,3)*PXAU(3) )                                EDEPTH  1630
      C12 = C12 + AXA(I)*PXAU(I)                                              EDEPTH  1640
   26 C22 = C22 + BXBM(I)*PXBU(I)                                             EDEPTH  1650
                                                                              EDEPTH  1660
C                                     SOLVE FOR DL AND DU                     EDEPTH  1670
C                                     C11*DL + C12*DU = C13                   EDEPTH  1680
C                                     C21*DL + C22*DU = C23                   EDEPTH  1690
C                                                                             EDEPTH  1700
      DET = C11*C22-C12*C21                                                   EDEPTH  1710
      DL = (C13*C22-C12*C23)/DET                                              EDEPTH  1720
      DU = (C11*C23-C13*C21)/DET                                              EDEPTH  1730
                                                                              EDEPTH  1740
C                                     INCREMENT L AND U                       EDEPTH  1750
C                                     TEST FOR CONVERGENCE                    EDEPTH  1760
C                                                                             EDEPTH  1770
      XL = XL + DL                                                            EDEPTH  1780
      XU = XU + DU                                                            EDEPTH  1790
      IF (DABS(DL/XL).GT.EPS(12)) GO TO 11                                    EDEPTH  1800
      IF (DABS(DU/XU).GT.EPS(12)) GO TO 11                                    EDEPTH  1810
   31 CONTINUE                                                                EDEPTH  1820
      RETURN                                                                  EDEPTH  1830
      END                                                                     EDEPTH  1840
```

```
      C23 = (1.0-C23)/2.0                                                    EDEPTH  1010
C                                      *          DXB                      EDEPTH  1020
C                                      SOLVE FOR ---                       EDEPTH  1030
C                                      *          DL                       EDEPTH  1040
C                                                                          EDEPTH  1050
C                                      * DXB                               EDEPTH  1060
C                                      C2--- = -AXA                        EDEPTH  1070
C                                      * DL                                EDEPTH  1080
C                                                                          EDEPTH  1090
      CALL DSMSOL(C2,3,3)                                                  EDEPTH  1100
C                                                                          EDEPTH  1110
C                                      CALCULATE                           EDEPTH  1120
C                                      DXA   DXB      DXB                  EDEPTH  1130
C                                      --- = --- + UB---                   EDEPTH  1140
C                                      DL    DL       DL                   EDEPTH  1150
C                                                                          EDEPTH  1160
C                                      *          DXA                      EDEPTH  1170
C                                      C11 = XA'A---                       EDEPTH  1180
C                                      *          DL                       EDEPTH  1190
C                                                                          EDEPTH  1200
C                                      *               DXB                 EDEPTH  1210
C                                      C21 = (XB-M)'B---                   EDEPTH  1220
C                                      *               DL                  EDEPTH  1230
C                                                                          EDEPTH  1240
      C11 = 0.0                                                            EDEPTH  1250
      C21 = 0.0                                                            EDEPTH  1260
      DO 25 I=1,3                                                          EDEPTH  1270
      PXAL(I) = B(I,1)*PXBL(1)                                             EDEPTH  1280
     *        + B(I,2)*PXBL(2)                                              EDEPTH  1290
     *        + B(I,3)*PXBL(3)                                              EDEPTH  1300
      PXAL(I) = PXBL(I) + XU*PXAL(I)                                       EDEPTH  1310
      C11 = C11 + AXA(I)*PXAL(I)                                           EDEPTH  1320
      C21 = C21 + BXBM(I)*PXBL(I)                                          EDEPTH  1330
   25 PXAU(I) = -BXBM(I)                                                   EDEPTH  1340
C                                                                          EDEPTH  1350
C                                      *          DXA                      EDEPTH  1360
C                                      SOLVE FOR ---                       EDEPTH  1370
C                                      *          DU                       EDEPTH  1380
C                                                                          EDEPTH  1390
C                                      * DXA                               EDEPTH  1400
C                                      C3--- = -B(XB-M)                    EDEPTH  1410
C                                      * DU                                EDEPTH  1420
C                                                                          EDEPTH  1430
      CALL DSMSOL(C3,3,3)                                                  EDEPTH  1440
C                                                                          EDEPTH  1450
C                                      CALCULATE                           EDEPTH  1460
C                                      DXB   DXA      DXA                  EDEPTH  1470
C                                      --- = --- + LA---                   EDEPTH  1480
C                                      DU    DU       DU                   EDEPTH  1490
C                                                                          EDEPTH  1500
```

```
   11 ITER = ITER+1                                                      EDEPTH  0510
      IF (NPRT(17).NE.0) WRITE (6,12) ITER,XL,XU,XA,XB                   EDEPTH  0520
   12 FORMAT(' EDEPTH ITER',I6,8G14.6)                                   EDEPTH  0530
      IF (ITER.LE.50) GO TO 14                                           EDEPTH  0540
      WRITE (6,13)                                                       EDEPTH  0550
   13 FORMAT(' EDEPTH ITERATION DID NOT CONVERGE')                       EDEPTH  0560
      GO TO 31                                                           EDEPTH  0570
C                                                                        EDEPTH  0580
CCC                                   FORM MATRICES                      EDEPTH  0590
CCC                                   C1 = LUAB + LA + UB                EDEPTH  0600
CCC                                   C2 = C1                            EDEPTH  0610
CCC                                   C3 = C1'                           EDEPTH  0620
C                                                                        EDEPTH  0630
   14 XLAU = XU*XL                                                       EDEPTH  0640
      DO 22 I=1,3                                                        EDEPTH  0650
      XBM(I) = 0.0                                                       EDEPTH  0660
      DO 22 J=1,3                                                        EDEPTH  0670
      C1(I,J) = XLAU*AB(I,J) + XL*A(I,J) + XU*B(I,J)                     EDEPTH  0680
      C2(I,J) = C1(I,J)                                                  EDEPTH  0690
      C3(J,I) = C1(I,J)                                                  EDEPTH  0700
   22 XBM(I) = XBM(I) - XL*A(I,J)*XM(J)                                  EDEPTH  0710
C                                                                        EDEPTH  0720
CCC                                   SOLVE FOR (XB-M)                   EDEPTH  0730
CCC                                   C1(XB-M) = -LAM                    EDEPTH  0740
C                                                                        EDEPTH  0750
      CALL DSMSOL(C1,3,3)                                                EDEPTH  0760
C                                                                        EDEPTH  0770
CCCCCCC                               EVALUATE                           EDEPTH  0780
CCCCCCC                               XB = (XB-M)+M                      EDEPTH  0790
CCCCCCC                               B(XB-M)                            EDEPTH  0800
CCCCCCC                               AXA                                EDEPTH  0810
CCCCCCC                               C13 = (1-XA'AXA)/2                 EDEPTH  0820
CCCCCCC                               C23 = (1-(XB-M)'B(XB-M))/2         EDEPTH  0830
C                                                                        EDEPTH  0840
      C13 = 0.0                                                          EDEPTH  0850
      C23 = 0.0                                                          EDEPTH  0860
      DO 23 I=1,3                                                        EDEPTH  0870
      XB(I) = XBM(I)+XM(I)                                               EDEPTH  0880
      BXBM(I) = B(I,1)*XBM(1)                                            EDEPTH  0890
     *        + B(I,2)*XBM(2)                                           EDEPTH  0900
     *        + B(I,3)*XBM(3)                                           EDEPTH  0910
   23 XA(I) = XB(I) + XU*BXBM(I)                                         EDEPTH  0920
      DO 24 I=1,3                                                        EDEPTH  0930
      AXA(I) = A(I,1)*XA(1)                                              EDEPTH  0940
     *       + A(I,2)*XA(2)                                             EDEPTH  0950
     *       + A(I,3)*XA(3)                                             EDEPTH  0960
      C13 = C13 + XA(I)*AXA(I)                                           EDEPTH  0970
      C23 = C23 + XBM(I)*BXBM(I)                                         EDEPTH  0980
   24 PXBL(I) = -AXA(I)                                                  EDEPTH  0990
      C13 = (1.0-C13)/2.0                                                EDEPTH  1000
```

```
      SUBROUTINE EDEPTH (A,B,XM,T,Y,XA,XB,XL,XU)                        EDEPTH  0010
C                                                        REV 19 08/05/78  EDEPTH  0020
CC    DETERMINES XA AND XB, THE POINTS OF MAXIMUM PENETRATION OF TWO     EDEPTH  0030
CC    INTERSECTING ELLIPSOIDS A AND B.                                   EDEPTH  0040
CC     ARGUMENTS A,B,XM,T AND X SAME AS FOR SUBROUTINE INTERS.           EDEPTH  0050
CC    ARGUMENTS XL AND XU, IF NONZERO, ARE FINAL RESULTS OF LAST CALL.   EDEPTH  0060
C                                                                        EDEPTH  0070
      IMPLICIT REAL*8 (A-H,O-Z)                                          EDEPTH  0080
      DIMENSION A(3,3),B(3,3),XM(3),Y(3),XA(3),XB(3)                     EDEPTH  0090
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,            EDEPTH  0100
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)         EDEPTH  0110
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                          EDEPTH  0120
     *               UNITL,UNITM,UNITT,GRAVTY(3)                         EDEPTH  0130
      DIMENSION C1(3,4),C2(3,4),C3(3,4),XBM(3),PXBL(3),PXAU(3),AB(3,3)   EDEPTH  0140
      DIMENSION AXA(3),BXBM(3),PXAL(3),PXBU(3)                           EDEPTH  0150
      EQUIVALENCE (XBM(1),C1(1,4)), (PXBL(1),C2(1,4)), (PXAU(1),C3(1,4)) EDEPTH  0160
C                                                                        EDEPTH  0170
CC                                    INITIAL GUESSES                    EDEPTH  0180
CC                                    XA = Y/T                           EDEPTH  0190
CC                                    XB = M+(Y-M)/T                     EDEPTH  0200
CC                                    L = -|XB-XA|/|AXA|                 EDEPTH  0210
CC                                    U = -|XB-XA|/|B(XB-M)|             EDEPTH  0220
C                                                                        EDEPTH  0230
      D1 = 0.0                                                           EDEPTH  0240
      D2 = 0.0                                                           EDEPTH  0250
      DO  9 I=1,3                                                        EDEPTH  0260
      XA(I) = Y(I)/T                                                     EDEPTH  0270
      XBM(I) = (Y(I)-XM(I))/T                                            EDEPTH  0280
      XB (I) = XBM(I)+XM(I)                                              EDEPTH  0290
    9 D1 = D1+(XB(I)-XA(I))**2                                           EDEPTH  0300
      IF (DABS(T-1.0).LE.EPS(6)) GO TO 31                                EDEPTH  0310
      ITER = 0                                                           EDEPTH  0320
      CALL MAT33 (A,B,AB)                                                EDEPTH  0330
      IF  (XL.NE.0.0)  GO TO 11                                          EDEPTH  0340
      IF  (XU.NE.0.0)  GO TO 11                                          EDEPTH  0350
      D3 = 0.0                                                           EDEPTH  0360
      DO 10 I=1,3                                                        EDEPTH  0370
      AXA(I) = A(I,1)*XA(1)                                              EDEPTH  0380
     *       + A(I,2)*XA(2)                                              EDEPTH  0390
     *       + A(I,3)*XA(3)                                              EDEPTH  0400
      D2 = D2 + AXA(I)**2                                                EDEPTH  0410
      BXBM(I) = B(I,1)*XBM(1)                                            EDEPTH  0420
     *        + B(I,2)*XBM(2)                                            EDEPTH  0430
     *        + B(I,3)*XBM(3)                                            EDEPTH  0440
   10 D3 = D3+BXBM(I)**2                                                 EDEPTH  0450
      XL = -DSQRT(D1/D2)                                                 EDEPTH  0460
      XU = -DSQRT(D1/D3)                                                 EDEPTH  0470
C                                                                        EDEPTH  0480
CC                                    START OF ITERATION                 EDEPTH  0490
C                                                                        EDEPTH  0500
```

```
      SUBROUTINE DZP(N,X,GG,E,R,M)                                          DZP      0010
C                                                          REV 19 08/05/78  DZP      0020
C     COMPUTES THE STATE VARIABLES (X) FROM THE PARAMETRIC FORM ASSUMED     DZP      0030
C     IN THE INTEGRATION ROUTINE DINT. ALSO EVALUATES THE EXPONENTIAL       DZP      0040
C     WEIGHTS (E) IF M IS NOT ZERO.                                         DZP      0050
C                                                                           DZP      0060
      IMPLICIT REAL*8 (A-H,O-Z)                                             DZP      0070
      DIMENSION X(1),GG(5,1),E(3,1)                                         DZP      0080
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                             DZP      0090
     *               UNITL,UNITM,UNITT,GRAVTY(3)                            DZP      0100
C                                                                           DZP      0110
      CALL ELTIME(1,5)                                                      DZP      0120
      IF(M.NE.0) GO TO 10                                                   DZP      0130
C                                                                           DZP      0140
C     COMPUTE STATE VARIABLES ONLY.                                         DZP      0150
C                                                                           DZP      0160
      DO 5  I=1,N                                                           DZP      0170
    5  X(I) = GG(1,I) + R*(GG(2,I)*E(1,I)                                   DZP      0180
     *                 + R*(GG(3,I)*E(2,I)                                 DZP      0190
     *                 + R*(GG(4,I)*E(3,I) )))                             DZP      0200
      GO TO 90                                                              DZP      0210
C                                                                           DZP      0220
C     COMPUTE EXPONENTIAL WEIGHTS AND STATE VARIABLES.                      DZP      0230
C                                                                           DZP      0240
   10 DO 50  I=1,N                                                          DZP      0250
      E(1,I) = 1.0                                                          DZP      0260
      E(2,I) = 0.5                                                          DZP      0270
      E(3,I) = THIRD                                                        DZP      0280
      IF (GG(5,I).EQ.0.0) GO TO 50                                          DZP      0290
      Z = R*GG(5,I)                                                         DZP      0300
      W = 0.                                                                DZP      0310
      IF (DABS(Z).GT.0.004) GO TO 20                                        DZP      0320
      W = 4.                                                                DZP      0330
      A = E(3,I)                                                            DZP      0340
      E(3,I) = 0.                                                           DZP      0350
   15 E(3,I) = E(3,I)+A                                                     DZP      0360
      A = A*Z/W                                                             DZP      0370
      W = W+1.0                                                             DZP      0380
      IF(E(3,I)+A.NE.E(3,I)) GO TO 15                                       DZP      0390
      E(2,I) = 0.5+0.5*Z*E(3,I)                                             DZP      0400
      E(1,I) = 1.+Z*E(2,I)                                                  DZP      0410
      GO TO 50                                                              DZP      0420
   20 IF(Z.GT.-40.) W = DEXP(Z)                                             DZP      0430
      E(1,I) = (W-1.)/Z                                                     DZP      0440
      E(2,I) = (E(1,I)-1.)/Z                                                DZP      0450
      E(3,I) = (2.*E(2,I)-1.)/Z                                             DZP      0460
   50  X(I) = GG(1,I) + R*(GG(2,I)*E(1,I)                                   DZP      0470
     *                 + R*(GG(3,I)*E(2,I)                                 DZP      0480
     *                 + R*(GG(4,I)*E(3,I) )))                             DZP      0490
C                                                                           DZP      0500
   90 CALL ELTIME(2,5)                                                      DZP      0510
      RETURN                                                                DZP      0520
      END                                                                   DZP      0530
```

```
      SUBROUTINE DSMSOL (A,KK,LL)                                       DSMSOL  0010
C                                                          REV 03 07/08/74 DSMSOL  0020
C     SOLVES A SET OF SIMULTANEOUS LINEAR EQUATIONS AX=B.                 DSMSOL  0030
C                                                                         DSMSOL  0040
C       ARGUMENTS:                                                        DSMSOL  0050
C         A: 2-DIMENSIONAL(KK,KK+1) MATRIX OF COEFFICIENTS.               DSMSOL  0060
C         KK: NUMBER OF EQUATIONS AND UNKNOWNS.                           DSMSOL  0070
C         LL: 1ST DIMENSION OF A IN CALLING PROGRAM.                      DSMSOL  0080
C                                                                         DSMSOL  0090
C       CALLING PROGRAM SETUP:                                            DSMSOL  0100
C         A(I,J) FOR I,J=1,KK                                             DSMSOL  0110
C         A(I,KK+1) = B(I) FOR I=1,KK                                     DSMSOL  0120
C         THE SOLUTION X IS RETURNED IN COLUMN KK+1 OF A.                 DSMSOL  0130
C         MATRIX A IS DESTROYED BY SUBROUTINE.                            DSMSOL  0140
C                                                                         DSMSOL  0150
      IMPLICIT REAL*8(A-H,O-Z)                                            DSMSOL  0160
      DIMENSION A(LL,1)                                                   DSMSOL  0170
      N  = KK                                                             DSMSOL  0180
      N1 = N+1                                                            DSMSOL  0190
      DO 50  L=1,N                                                        DSMSOL  0200
      L1 = L+1                                                            DSMSOL  0210
      BIG = 0.0                                                           DSMSOL  0220
      DO 25  I=L,N                                                        DSMSOL  0230
      IF (DABS(A(I,L)).LE.DABS(BIG)) GO TO 25                             DSMSOL  0240
      K = I                                                               DSMSOL  0250
      BIG = A(I,L)                                                        DSMSOL  0260
   25 CONTINUE                                                            DSMSOL  0270
      IF (BIG.NE.0.0) GO TO 30                                            DSMSOL  0280
      WRITE (6,26)                                                        DSMSOL  0290
   26 FORMAT('0 DSMSOL MATRIX SINGULAR, PROGRAM TERMINATED.')             DSMSOL  0300
      STOP  41                                                            DSMSOL  0310
   30 BIG = 1.0/BIG                                                       DSMSOL  0320
      DO 40  J=L,N1                                                       DSMSOL  0330
      B      = A(K,J)                                                     DSMSOL  0340
      A(K,J) = A(L,J)                                                     DSMSOL  0350
   40 A(L,J) = B*BIG                                                      DSMSOL  0360
      IF (L.EQ.N) GO TO 50                                                DSMSOL  0370
      DO 48  I=L1,N                                                       DSMSOL  0380
      IF (A(I,L).EQ.0.0) GO TO 48                                         DSMSOL  0390
      DO 45  J=L1,N1                                                      DSMSOL  0400
   45 A(I,J) = A(I,J)-A(I,L)*A(L,J)                                       DSMSOL  0410
   48 CONTINUE                                                            DSMSOL  0420
   50 CONTINUE                                                            DSMSOL  0430
      IF (N.EQ.1) GO TO 71                                                DSMSOL  0440
      N2 = N-1                                                            DSMSOL  0450
      DO 60  L=1,N2                                                       DSMSOL  0460
      I  = N-L                                                            DSMSOL  0470
      L1 = I+1                                                            DSMSOL  0480
      DO 60  J=L1,N                                                       DSMSOL  0490
   60 A(I,N1) = A(I,N1)-A(I,J)*A(J,N1)                                    DSMSOL  0500
                                                                          DSMSOL  0510
   71 CONTINUE                                                            DSMSOL  0520
      RETURN                                                              DSMSOL  0530
      END
```

```
      SUBROUTINE DSETQ(E,TH,ES,EC,D)                                    DSETQ   0010
C                                                          REV 19 08/05/78 DSETQ   0020
CC    COMPUTES NEW DIRECTION MATRIX (D), GIVEN ORIGINAL MATRIX (E)      DSETQ   0030
CC    AND INCREMENTAL MOTION EXPRESSED IN QUATERNION FORM.              DSETQ   0040
CC                                                                      DSETQ   0050
CC      ARGUMENTS:                                                      DSETQ   0060
CC                                                                      DSETQ   0070
CC           E : ORIGINAL DIRECTION COSINE MATRIX.                      DSETQ   0080
CC          TH : COMPONENTS OF Q ( UX SIN A/2, UY SIN A/2, UZ SIN A/2)  DSETQ   0090
CC          ES : SIN**2(A/2)                                            DSETQ   0100
CC          EC : COS (A/2)                                              DSETQ   0110
CC           D : NEW DIRECTION COSINE MATRIX.                           DSETQ   0120
C                                                                       DSETQ   0130
       IMPLICIT REAL*8(A-H,O-Z)                                         DSETQ   0140
      DIMENSION D(3,3),TH(3),S(3),TEMP(3,3),E(3,3)                      DSETQ   0150
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         DSETQ   0160
     *               UNITL,UNITM,UNITT,GRAVTY(3)                       DSETQ   0170
      CT = 1.0 - 2.0*ES                                                 DSETQ   0180
      DO 10 J=1,3                                                       DSETQ   0190
      S(1) = TH(2)*E(3,J) - TH(3)*E(2,J)                                DSETQ   0200
      S(2) = TH(3)*E(1,J) - TH(1)*E(3,J)                                DSETQ   0210
      S(3) = TH(1)*E(2,J) - TH(2)*E(1,J)                                DSETQ   0220
      DTT = TH(1)*E(1,J) + TH(2)*E(2,J) + TH(3)*E(3,J)                  DSETQ   0230
      DO 5 K=1,3                                                        DSETQ   0240
    5 D(K,J) = E(K,J)*CT + 2.0*(TH(K)*DTT - EC*S(K))                    DSETQ   0250
   10  CONTINUE                                                         DSETQ   0260
C                                                                       DSETQ   0270
CC    RENORMALIZATION OF DIRECTION COSINE MATRIX                        DSETQ   0280
CC    BY AVERAGING MATRIX AND TRANSPOSE OF ITS INVERSE.                 DSETQ   0290
C                                                                       DSETQ   0300
      DO 23 ITER=1,10                                                   DSETQ   0310
      CALL CFACTT(D,TEMP,DET)                                           DSETQ   0320
      DO 22 I=1,3                                                       DSETQ   0330
      DO 22 J=1,3                                                       DSETQ   0340
      D(I,J) = 0.5*(D(I,J)+TEMP(J,I)/DET)                               DSETQ   0350
   22 IF (DABS(D(I,J)).LT.EPS(15)) D(I,J)=0.0                           DSETQ   0360
      IF (DABS(DET-1.0).LT.EPS(6)) GO TO 24                             DSETQ   0370
   23 CONTINUE                                                          DSETQ   0380
      WRITE (6,27) DET                                                  DSETQ   0390
   27 FORMAT('0 DSETQ RENORMALIZATION DID NOT CONVERGE, DET =',1PD25.15) DSETQ   0400
   24 RETURN                                                            DSETQ   0410
       END                                                              DSETQ   0420
```

```
      SUBROUTINE DSETD(D,TH,T)                                          DSETD   0010
C                                                     REV 19 08/05/78   DSETD   0020
CC    UPDATES A DIRECTION COSINE MATRIX (D)                             DSETD   0030
CC    USING AN INCREMENTAL ANGULAR MOTION (TH).                         DSETD   0040
CC        ARGUMENTS D: 3X3 DIRECTION COSINE MATRIX TO BE UPDATED.       DSETD   0050
CC                  TH: 3 COMPONENTS OF INCREMENTAL ANGULAR MOTION      DSETD   0060
CC                      ABOUT LOCAL X,Y AND Z AXIS RESPECTIVELY.        DSETD   0070
CC                   T: MAGNITUDE OF VECTOR TH COMPUTED BY ROUTINE.     DSETD   0080
C                                                                       DSETD   0090
      IMPLICIT REAL*8(A-H,O-Z)                                          DSETD   0100
      DIMENSION D(3,3),TH(3),S(3),TEMP(3,3)                             DSETD   0110
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         DSETD   0120
     *               UNITL,UNITM,UNITT,GRAVTY(3)                        DSETD   0130
      T=DSQRT(TH(1)**2+TH(2)**2+TH(3)**2)                               DSETD   0140
      IF(T.EQ.0.)RETURN                                                 DSETD   0150
      ST=DSIN(T)                                                        DSETD   0160
      CT=DCOS(T)                                                        DSETD   0170
      STT=ST/T                                                          DSETD   0180
      CTT=STT**2/(1.+CT)                                                DSETD   0190
      DO 10 J=1,3                                                       DSETD   0200
      S(1)=-TH(3)*D(2,J)+TH(2)*D(3,J)                                   DSETD   0210
      S(2)= TH(3)*D(1,J)-TH(1)*D(3,J)                                   DSETD   0220
      S(3)=-TH(2)*D(1,J)+TH(1)*D(2,J)                                   DSETD   0230
      DTT=(TH(1)*D(1,J)+TH(2)*D(2,J)+TH(3)*D(3,J))*CTT                  DSETD   0240
      DO 5 K=1,3                                                        DSETD   0250
    5 D(K,J)=D(K,J)*CT-STT*S(K)+TH(K)*DTT                               DSETD   0260
   10 CONTINUE                                                          DSETD   0270
C                                                                       DSETD   0280
CC    RENORMALIZATION OF DIRECTION COSINE MATRIX                        DSETD   0290
CC    BY AVERAGING MATRIX AND TRANSPOSE OF ITS INVERSE.                 DSETD   0300
C                                                                       DSETD   0310
      DO 23 ITER=1,10                                                   DSETD   0320
      CALL CFACTT(D,TEMP,DET)                                           DSETD   0330
      DO 22 I=1,3                                                       DSETD   0340
      DO 22 J=1,3                                                       DSETD   0350
      D(I,J) = 0.5*(D(I,J)+TEMP(J,I)/DET)                               DSETD   0360
   22 IF (DABS(D(I,J)).LT.EPS(15)) D(I,J)=0.0                           DSETD   0370
      IF (DABS(DET-1.0).LT.EPS(6)) GO TO 24                             DSETD   0380
   23 CONTINUE                                                          DSETD   0390
      WRITE (6,27) DET                                                  DSETD   0400
   27 FORMAT('0 DSETD RENORMALIZATION DID NOT CONVERGE, DET =',1PD25.15) DSETD   0410
   24 RETURN                                                            DSETD   0420
      END                                                               DSETD   0430
```

```
      SUBROUTINE DRCYPR (D,A,ID)                                        DRCYPR  0010
C                                                          REV 19 08/05/78  DRCYPR  0020
C     SETS UP 3X3 DIRECTION COSINE MATRIX FOR GIVEN YAW,PITCH AND ROLL.    DRCYPR  0030
C                                                                          DRCYPR  0040
C       ARGUMENTS:                                                         DRCYPR  0050
C         D: 3X3 DIRECTION COSINE MATRIX TO BE COMPUTED.                   DRCYPR  0060
C         A: ARRAY OF LENGTH 3 CONTAINING ROTATATION ANGLES (DEGREES).     DRCYPR  0070
C        I1: AXIS OF ROTATION FOR 1ST ANGLE (1,2,3 = X,Y,Z)                DRCYPR  0080
C        I2: AXIS OF ROTATION FOR 2ND ANGLE (1,2,3 = X,Y,Z)                DRCYPR  0090
C        I3: AXIS OF ROTATION FOR 3RD ANGLE (1,2,3 = X,Y,Z)                DRCYPR  0100
C                                                                          DRCYPR  0110
      IMPLICIT REAL*8 (A-H,O-Z)                                            DRCYPR  0120
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                            DRCYPR  0130
     *               UNITL,UNITM,UNITT,GRAVTY(3)                         DRCYPR  0140
      DIMENSION D(3,3),A(3),ID(3),T(3,3),B(3),S(3)                         DRCYPR  0150
      IDSUM = ID(1) + ID(2) + ID(3)                                        DRCYPR  0160
      DO  12  I=1,3                                                        DRCYPR  0170
      B(I) = A(I)*RADIAN                                                   DRCYPR  0180
      DO  11  J=1,3                                                        DRCYPR  0190
   11 D(I,J) = 0.0                                                         DRCYPR  0200
   12 D(I,I) = 1.0                                                         DRCYPR  0210
      DO  30  N=1,3                                                        DRCYPR  0220
      IDN = IABS(ID(N))                                                    DRCYPR  0230
      M = 4 - IDN                                                          DRCYPR  0240
      IF  (ID(N).LT.0)  M = IDSUM - ID(N) - 2                              DRCYPR  0250
      IF  (B(M).EQ.0.0)  GO TO 30                                          DRCYPR  0260
      CALL ROT (T,IDN,B(M))                                                DRCYPR  0270
      DO  23  J=1,3                                                        DRCYPR  0280
      DO  21  K=1,3                                                        DRCYPR  0290
      S(K) = D(K,J)                                                        DRCYPR  0300
   21 D(K,J) = 0.0                                                         DRCYPR  0310
      DO  22  I=1,3                                                        DRCYPR  0320
      DO  22  K=1,3                                                        DRCYPR  0330
   22 D(I,J) = D(I,J) + T(I,K)*S(K)                                        DRCYPR  0340
   23 CONTINUE                                                             DRCYPR  0350
   30 CONTINUE                                                             DRCYPR  0360
      RETURN                                                               DRCYPR  0370
      END                                                                  DRCYPR  0380
```

```
      SUBROUTINE DRCIJK (D,ANG,ID,HT,J)                                 DRCIJK  0010
C                                                     REV 18 02/24/78   DRCIJK  0020
      IMPLICIT REAL*8 (A-H,O-Z)                                         DRCIJK  0030
      DIMENSION  D(9,22),HT(9,42),ANG(3,22),ID(4,22),T1(9),T2(9)        DRCIJK  0040
      M = ID(4,J)                                                       DRCIJK  0050
      IF  (M.NE.0)  GO TO 10                                            DRCIJK  0060
      CALL DRCYPR (D(1,J),ANG(1,J),ID(1,J))                             DRCIJK  0070
      GO TO 99                                                          DRCIJK  0080
   10 CALL DRCYPR (T1,ANG(1,J),ID(1,J))                                 DRCIJK  0090
      IF  (M.LT.0)  GO TO 20                                            DRCIJK  0100
      CALL MAT33 (T1,D(1,M),D(1,J))                                     DRCIJK  0110
      GO TO 99                                                          DRCIJK  0120
   20 M = -M                                                            DRCIJK  0130
      CALL DOT33 (HT(1,2*J-3),D(1,M),D(1,J))                            DRCIJK  0140
      CALL MAT33 (T1,D(1,J),T2)                                         DRCIJK  0150
      CALL MAT33 (HT(1,2*J-2),T2,D(1,J))                                DRCIJK  0160
   99 RETURN                                                            DRCIJK  0170
      END                                                               DRCIJK  0180
```

```
      SUBROUTINE DOT33 (A,B,C)                                          DOT33   0010
C                                                      REV 17 01/03/77  DOT33   0020
C     PERFORMS MATRIX MULTIPLICATION C = A'B                            DOT33   0030
C     WHERE A, B AND C ARE ALL 3X3 MATRICEES.                           DOT33   0040
C                                                                       DOT33   0050
      IMPLICIT REAL*8 (A-H,O-Z)                                         DOT33   0060
      DIMENSION  A(3,3) , B(3,3) , C(3,3)                               DOT33   0070
      DO  10  I=1,3                                                     DOT33   0080
      DO  10  J=1,3                                                     DOT33   0090
   10 C(I,J) = A(1,I)*B(1,J) + A(2,I)*B(2,J) + A(3,I)*B(3,J)            DOT33   0100
      RETURN                                                            DOT33   0110
      END                                                               DOT33   0120
```

```
      SUBROUTINE DOT31 (A,B,C)                                          DOT31   0010
C                                                     REV 17 01/03/77   DOT31   0020
C     PERFORMS MATRIX MULTIPLICATION C = A'B                            DOT31   0030
C     WHERE A IS A 3X3 MATRIX, AND B AND C ARE VECTORS OF LENGTH 3.     DOT31   0040
C                                                                       DOT31   0050
      IMPLICIT REAL*8 (A-H,O-Z)                                         DOT31   0060
      DIMENSION  A(3,3) , B(3) , C(3)                                   DOT31   0070
      C(1) = A(1,1)*B(1) + A(2,1)*B(2) + A(3,1)*B(3)                    DOT31   0080
      C(2) = A(1,2)*B(1) + A(2,2)*B(2) + A(3,2)*B(3)                    DOT31   0090
      C(3) = A(1,3)*B(1) + A(2,3)*B(2) + A(3,3)*B(3)                    DOT31   0100
      RETURN                                                            DOT31   0110
      END                                                               DOT31   0120
```

```
      SUBROUTINE DOTT33 (A,B,C)                                          DOTT33  0010
C                                                    REV 17 01/03/77     DOTT33  0020
C     PERFORMS MATRIX MULTIPLICATION C = AB'                             DOTT33  0030
C     WHERE A, B AND C ARE ALL 3X3 MATRICEES.                            DOTT33  0040
C                                                                        DOTT33  0050
      IMPLICIT REAL*8 (A-H,O-Z)                                          DOTT33  0060
      DIMENSION  A(3,3) , B(3,3) , C(3,3)                                DOTT33  0070
      DO  10  I=1,3                                                      DOTT33  0080
      DO  10  J=1,3                                                      DOTT33  0090
   10 C(I,J) = A(I,1)*B(J,1) + A(I,2)*B(J,2) + A(I,3)*B(J,3)             DOTT33  0100
      RETURN                                                             DOTT33  0110
      END                                                                DOTT33  0120
```

```
      SUBROUTINE DOTT31 (A,B,C)                                          DOTT31  0010
C                                                        REV 17 12/20/76  DOTT31  0020
C     PERFORMS MATRIX MULTIPLICATION C = AB'                             DOTT31  0030
C     WHERE C IS A 3X3 MATRIX, AND A AND B ARE VECTORS OF LENGTH 3.      DOTT31  0040
C                                                                        DOTT31  0050
      IMPLICIT REAL*8 (A-H,O-Z)                                          DOTT31  0060
      DIMENSION  A(3) , B(3) , C(3,3)                                    DOTT31  0070
      DO  10  I=1,3                                                      DOTT31  0080
      DO  10  J=1,3                                                      DOTT31  0090
   10 C(I,J) = A(I)*B(J)                                                 DOTT31  0100
      RETURN                                                             DOTT31  0110
      END                                                                DOTT31  0120
```

```
      GG5 = DEXP(-1600.0*H)                                              DINT    1510
      DO  63  I=1,NEQ                                                    DINT    1520
      F(3,I) = GG(3,I) + GG4*GG(4,I)                                     DINT    1530
      F(4,I) = GG(4,I)                                                   DINT    1540
      F(5,I) = GG(5,I)                                                   DINT    1550
      Y(3,I) = Y(1,I)                                                    DINT    1560
      Y(4,I) = Y(2,I)                                                    DINT    1570
      Y(5,I) = GG5*U(3,I)                                                DINT    1580
63    U(5,I) = GG5*U(4,I)                                                DINT    1590
      CALL QSET(F,Y,VAR,DER,NQUAT)                                       DINT    1600
      CALL PDAUX (VAR,DER,M,K)                                           DINT    1610
      DO  64  I=1,NEQ                                                    DINT    1620
      F(1,I) = VAR(I)                                                    DINT    1630
64    F(2,I) = DER(I)                                                    DINT    1640
      HS = H                                                             DINT    1650
      IF  (ICNT.LT.IDBL)  GO TO 65                                       DINT    1660
      ICNT = 0                                                           DINT    1670
      H = DMIN1(2.0*H,HMAX)                                              DINT    1680
      HPRINT = DMIN1(2.0*HPRINT,HMAX)                                    DINT    1690
65    CALL UPDATE(2)                                                     DINT    1700
      CALL OUTPUT(1)                                                     DINT    1710
      IF  (TPRINT-TIME.GE.EPS(8))  GO TO 12                              DINT    1720
      CALL ELTIME(2,3)                                                   DINT    1730
      RETURN                                                             DINT    1740
      END                                                                DINT    1750
```

```
      TE = 0.0                                                                  DINT    1010
      TY = 0.0                                                                  DINT    1020
      I2 = II+2                                                                 DINT    1030
      DO  45  I=II,I2                                                           DINT    1040
      Z = GG(5,I)*(VAR(I)-GG(1,I)) + GG(2,I) + H*(GG(3,I)+H*GG(4,I))            DINT    1050
      TE = TE + (DER(I)-Z)**2                                                   DINT    1060
      TYD = TT + TX*GG(5,I)**2                                                  DINT    1070
      IF  (TYD.EQ.0.0)  TYD = 1.0                                               DINT    1080
45    TY = TY + (DER(I)-Z)**2/TYD                                               DINT    1090
      TM = 1000.0*TIME                                                          DINT    1100
      IF  (NPRT(25).NE.0)  WRITE (6,46) TM,SEGT(JJ),REGT(JJ),TT,TE,TY,          DINT    1110
     *                                  (XTEST(I),I=II,I2)                      DINT    1120
46    FORMAT ('0 DINT CONV. TEST',F10.3,2X,A4,2X,A8,6G12.4)                     DINT    1130
      IF  (TT.LT.XTEST(II))  GO TO 47                                           DINT    1140
      IF  (XTEST(II+1).GT.0.0 .AND. TE.LT.XTEST(II+1))  GO TO 47                DINT    1150
      IF  (TY.GT.XTEST(II+2))  GO TO 48                                         DINT    1160
47    CONTINUE                                                                  DINT    1170
      FAIL = 0.0                                                                DINT    1180
48    CALL ADJUST   (4,D1)                                                      DINT    1190
      IF  (FAIL.EQ.0.0)  GO TO 60                                               DINT    1200
      IF  (L.EQ.NDINT)   GO TO 49                                               DINT    1210
      CALL CMPUTE (K,1,D1)                                                      DINT    1220
      IF  (K.LT.0)       GO TO 50                                               DINT    1230
      CALL ADJUST   (5,D1)                                                      DINT    1240
49    CONTINUE                                                                  DINT    1250
      IF  (NPRT(25).EQ.0)  WRITE (6,46) TM,SEGT(JJ),REGT(JJ),TT,TE,TY,          DINT    1260
     *                                  (XTEST(I),I=II,I2)                      DINT    1270
50    WRITE  (6,51)  TIME,H                                                     DINT    1280
51    FORMAT('0 TEST FAILED AT TIME = ',F10.6,' FOR H = ',F10.6)                DINT    1290
      ICNT = 0                                                                  DINT    1300
      IDBL = IDBL+2                                                             DINT    1310
      IF  (IDBL.GT.6)  IDBL = 6                                                 DINT    1320
      IF  (K.GE.0)  GO TO 58                                                    DINT    1330
      IF  (H.GT.HMIN+EPS(8))  GO TO 59                                          DINT    1340
      WRITE  (6,52)                                                             DINT    1350
52    FORMAT('0 PROGRAM TERMINATED: PDAUX NEG SQRT. H < HMIN+EPS8.'/            DINT    1360
     *       '  RERUN PROGRAM WITH SMALLER HMIN ON INPUT CARD A.4')             DINT    1370
      STOP 31                                                                   DINT    1380
58    IF  (H.LE.HMIN+EPS(8))  GO TO 61                                          DINT    1390
      IF  (NPRT(26).EQ.2)  CALL OUTPUT(1)                                       DINT    1400
59    TIME = TSTART                                                             DINT    1410
      H = 0.5*H                                                                 DINT    1420
      HPRINT = 0.5*HPRINT                                                       DINT    1430
      K = 2                                                                     DINT    1440
      GO TO 16                                                                  DINT    1450
60    IF  (H.GT.0.74*HPRINT)  ICNT = ICNT+1                                     DINT    1460
61    K = 4                                                                     DINT    1470
      M = 0                                                                     DINT    1480
      IF  (H.GT.HMIN .AND. IDBL.GT.2)  IDBL = IDBL-1                            DINT    1490
      GG4 = 2.0*H                                                               DINT    1500
```

```
      IF  (ISTEP.NE.0 .AND. NPRT(26).EQ.2)  CALL OUTPUT(1)                DINT    0510
      DO  14  I=1,NEQ                                                     DINT    0520
      F(1,I) = VAR(I)                                                     DINT    0530
      F(2,I) = DER(I)                                                     DINT    0540
      DO  14  J=3,5                                                       DINT    0550
      F(J,I) = 0.0                                                        DINT    0560
      U(J,I) = 0.0                                                        DINT    0570
   14 Y(J,I) = 0.0                                                        DINT    0580
      IF  (ISTEP.EQ.0)  GO TO 65                                          DINT    0590
      K = 1                                                               DINT    0600
C                                                                         DINT    0610
C     ADJUST H (CURRENT TIME STEP) IF IT WILL ADVANCE T BEYOND TPRINT.    DINT    0620
C                                                                         DINT    0630
   15 IF  (H+EPS(8).GE.TPRINT-TIME)  H = TPRINT-TIME                      DINT    0640
C                                                                         DINT    0650
C     BACKUP ENTRY POINT IF H HAS BEEN HALVED.                            DINT    0660
C                                                                         DINT    0670
   16 D1 = 0.5*H                                                          DINT    0680
      CALL TRIGFS                                                         DINT    0690
      TSTART = TIME                                                       DINT    0700
      DO  20  I=1,NEQ                                                     DINT    0710
      U(3,I) = Y(5,I)                                                     DINT    0720
      U(4,I) = U(5,I)                                                     DINT    0730
      DO  20  J=1,5                                                       DINT    0740
   20 GG(J,I) = F(J,I)                                                    DINT    0750
      CALL CMPUTE (K,1,D1)                                                DINT    0760
      IF (K.LT.0)  GO TO 50                                               DINT    0770
      CALL ADJUST   (1,D1)                                                DINT    0780
      K = 2                                                               DINT    0790
      CALL CMPUTE (K,0,D1)                                                DINT    0800
      IF (K.LT.0)  GO TO 50                                               DINT    0810
      CALL ADJUST   (2,D1)                                                DINT    0820
      NQUAT = K                                                           DINT    0830
      K = 3                                                               DINT    0840
      CALL CMPUTE (K,1, H)                                                DINT    0850
      IF (K.LT.0)  GO TO 50                                               DINT    0860
      CALL ADJUST   (3,D1)                                                DINT    0870
      DO  49  L=1,NDINT                                                   DINT    0880
      M = 1                                                               DINT    0890
      IF  (L.EQ.1)  M = 0                                                 DINT    0900
      IF  (NPRT(26).NE.2)  CALL OUTPUT(0)                                 DINT    0910
      CALL CMPUTE (K,M, H)                                                DINT    0920
      IF (K.LT.0)  GO TO 50                                               DINT    0930
      FAIL = 1.0                                                          DINT    0940
      JJ = 0                                                              DINT    0950
      DO  47  II=1,NEQ,3                                                  DINT    0960
      JJ = JJ+1                                                           DINT    0970
      IF  (XTEST(II).LE.0.0)  GO TO 47                                    DINT    0980
      TT = DER(II)**2 + DER(II+1)**2 + DER(II+2)**2                       DINT    0990
      TX = VAR(II)**2 + VAR(II+1)**2 + VAR(II+2)**2                       DINT    1000
```

```
      SUBROUTINE DINT                                                    DINT    0010
C                                                          REV 19 09/18/79 DINT    0020
      IMPLICIT REAL*8 (A-H,O-Z)                                          DINT    0030
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,            DINT    0040
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)         DINT    0050
      COMMON/INTEST/ SGTEST(3,4,30),XTEST(360  ),SEGT(120),REGT(120)     DINT    0060
C     NOTE: XTEST SINGLY DIMENSIONED HERE.                               DINT    0070
      REAL    SEGT                                                       DINT    0080
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                          DINT    0090
     *               UNITL,UNITM,UNITT,GRAVTY(3)                         DINT    0100
      COMMON/CDINT/ UU(4),GH(3,4),                                       DINT    0110
     *              E(3,240), F(5,240),GG(5,240),Y(5,240),U(5,240),      DINT    0120
     *              H,HPRINT,HS,TPRINT,TSTART,ICNT,IDBL,IFLAG            DINT    0130
      COMMON/COMAIN/ VAR(240),DER(240),DT,HO,HMAX,HMIN,RSTIME,           DINT    0140
     *               ISTEP,NSTEPS,NDINT,NEQ,IRSIN,IRSOUT                 DINT    0150
      CALL ELTIME(1,3)                                                   DINT    0160
      IF  (ISTEP.NE.0)  GO TO 11                                         DINT    0170
C                                                                        DINT    0180
C     IN=0: INITIAL CALL TO INTEGRATOR - INITIALIZE AND RESET PARAMETERS DINT    0190
C     NOTE: FOR EARLIER VERSIONS OF CVS, THE VARIABLE 'IN'(ISTEP IN THE  DINT    0200
C           CALLING PROGRAM) RAN FRON 1 TO NSTEPS+1, NOW IT RUNS FROM    DINT    0210
C           0 TO NSTEPS.                                                 DINT    0220
C                                                                        DINT    0230
      TPRINT = TIME                                                      DINT    0240
      IDBL = 2                                                           DINT    0250
      K = 0                                                              DINT    0260
      GO TO 13                                                           DINT    0270
C                                                                        DINT    0280
C     IN#0: ADVANCE TPRINT - TIME TO RETURN TO CALLING PROGRAM.          DINT    0290
C                                                                        DINT    0300
  11  TPRINT = TPRINT + DT                                               DINT    0310
      H = HPRINT                                                         DINT    0320
C                                                                        DINT    0330
C     ENTRY TO ADVANCE INTEGRATOR                                        DINT    0340
C                                                                        DINT    0350
  12  K = 1                                                              DINT    0360
      CALL UPDATE(K)                                                     DINT    0370
C                                                                        DINT    0380
C      NEGATIVE K FROM UPDATE IS INDICATOR TO RESET INTEGRARAOR.         DINT    0390
C                                                                        DINT    0400
      IF  (K.EQ.1)  GO TO 15                                             DINT    0410
C                                                                        DINT    0420
C      RESET OR INITIALIZE INTEGRATOR.                                   DINT    0430
C                                                                        DINT    0440
  13  H = HO                                                             DINT    0450
      HPRINT = HO                                                        DINT    0460
      HS = 0.0                                                           DINT    0470
      ICNT = -2                                                          DINT    0480
      IF  (ISTEP.EQ.0  .OR. NPRT(26).EQ.2)  CALL OUTPUT(0)               DINT    0490
      CALL PDAUX (VAR,DER,NEQ,K)                                         DINT    0500
```

```
      SUBROUTINE DHHPIN(DD,BN,L,M,N)                                    DHHPIN  0010
C                                                     REV 19 08/05/78   DHHPIN  0020
C     SETS DD = D(L) IF JOINT M IS NOT PINNED                           DHHPIN  0030
C       OR DD = (I-HH.)(D(L)) IF PINNED                                 DHHPIN  0040
C                                                                       DHHPIN  0050
      IMPLICIT REAL*8 (A-H,O-Z)                                         DHHPIN  0060
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),DHHPIN  0070
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       DHHPIN  0080
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(3,60),HA(3,60),HB(3,60), DHHPIN  0090
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),     DHHPIN  0100
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)    DHHPIN  0110
      COMMON/CEULER/ IEULER(30),HIR(3,3,30),ANG(3,30),ANGD(3,30),       DHHPIN  0120
     *               FE(3,30),TQE(3,30),CONST(3,30)                     DHHPIN  0130
      DIMENSION DD(3,3),BN(3)                                           DHHPIN  0140
      DO 10 J=1,3                                                       DHHPIN  0150
      BN(J) = 0.0                                                       DHHPIN  0160
      DO 10 I=1,3                                                       DHHPIN  0170
   10 DD(I,J) = D(I,J,L)                                                DHHPIN  0180
      LGO = IPIN(M)+5                                                   DHHPIN  0190
      TSIGN = -1.0                                                      DHHPIN  0200
      GO TO (20,90,90,90,90,30,90,90,90),LGO                            DHHPIN  0210
   20 IF (IEULER(M).GE.7) GO TO 90                                      DHHPIN  0220
      IF (IEULER(M).GE.4) GO TO 30                                      DHHPIN  0230
      TSIGN = 1.0                                                       DHHPIN  0240
      DO 21 J=1,3                                                       DHHPIN  0250
      DO 21 I=1,3                                                       DHHPIN  0260
   21 DD(I,J) = 0.0                                                     DHHPIN  0270
   30 DO 31 J=1,3                                                       DHHPIN  0280
      BN(J) = HB(1,N)*D(1,J,L) + HB(2,N)*D(2,J,L) + HB(3,N)*D(3,J,L)    DHHPIN  0290
      DO 31 I=1,3                                                       DHHPIN  0300
   31 DD(I,J) = DD(I,J) + TSIGN*BN(J)*HB(I,N)                           DHHPIN  0310
   90 RETURN                                                            DHHPIN  0320
      END                                                               DHHPIN  0330
```

```
      NNS = 2*NS+N                                                      DAUX55  0510
      IJK(IS+1,NNS) = IJ+1                                              DAUX55  0520
      IJK(NNS,IS+1) = IJ+2                                              DAUX55  0530
      IJ = IJ+2                                                         DAUX55  0540
   15 CONTINUE                                                          DAUX55  0550
   19 IF (NQ.EQ.0) GO TO 30                                             DAUX55  0560
      DO 25 N=1,NQ                                                      DAUX55  0570
      IF (KQTYPE(N).LT.0) GO TO 25                                      DAUX55  0580
      LN = 0                                                            DAUX55  0590
      IF (I.EQ.KQ1(N)) LN = 2*N-1                                       DAUX55  0600
      IF (I.EQ.KQ2(N)) LN = 2*N                                         DAUX55  0610
      IF (LN.EQ.0) GO TO 25                                             DAUX55  0620
      DO 20 J=1,3                                                       DAUX55  0630
      DO 20 K=1,3                                                       DAUX55  0640
      C(J,K,IJ+1) =  A13(J,K,LN)                                        DAUX55  0650
      C(J,K,IJ+2) =  A23(J,K,LN)                                        DAUX55  0660
      C(J,K,IJ+3) = -B31(J,K,LN)                                        DAUX55  0670
   20 C(J,K,IJ+4) = -B32(J,K,LN)                                        DAUX55  0680
      NNS = 2*NS+NFLX+N                                                 DAUX55  0690
      IJK(IS  ,NNS) = IJ+1                                              DAUX55  0700
      IJK(IS+1,NNS) = IJ+2                                              DAUX55  0710
      IJK(NNS,IS  ) = IJ+3                                              DAUX55  0720
      IJK(NNS,IS+1) = IJ+4                                              DAUX55  0730
      IJ = IJ+4                                                         DAUX55  0740
   25 CONTINUE                                                          DAUX55  0750
   30 IF (NJNT.EQ.0) GO TO 98                                           DAUX55  0760
      DO 65 N=1,NJNT                                                    DAUX55  0770
      IF (JNT(N).EQ.0) GO TO 65                                         DAUX55  0780
      LN = 0                                                            DAUX55  0790
      IF (I.EQ.IABS(JNT(N))) LN = 2*N-1                                 DAUX55  0800
      IF (I.EQ.N+1)          LN = 2*N                                   DAUX55  0810
      IF (LN.EQ.0) GO TO 65                                             DAUX55  0820
      SET = 1.0                                                         DAUX55  0830
      IF (I.EQ.N+1) SET = -1.0                                          DAUX55  0840
      DO 40 J=1,3                                                       DAUX55  0850
      DO 39 K=1,3                                                       DAUX55  0860
      C(J,K,IJ+1) = 0.0                                                 DAUX55  0870
      C(J,K,IJ+3) = 0.0                                                 DAUX55  0880
      C(J,K,IJ+2) =  B12(K,J,LN)                                        DAUX55  0890
   39 C(J,K,IJ+4) = -B12(J,K,LN)                                        DAUX55  0900
      C(J,J,IJ+1) =  SET                                                DAUX55  0910
   40 C(J,J,IJ+3) = -SET                                                DAUX55  0920
      NNS = NQ2S + N                                                    DAUX55  0930
      IJK(IS  ,NNS) = IJ+1                                              DAUX55  0940
      IJK(IS+1,NNS) = IJ+2                                              DAUX55  0950
      IJK(NNS,IS  ) = IJ+3                                              DAUX55  0960
      IJK(NNS,IS+1) = IJ+4                                              DAUX55  0970
      IJ = IJ+4                                                         DAUX55  0980
      IF (IPIN(N).EQ.0 .OR. IPIN(N).GE.2) GO TO 65                      DAUX55  0990
      DO 60 J=1,3                                                       DAUX55  1000
      DO 60 K=1,3                                                       DAUX55  1010
      C(J,K,IJ+1) =  SET*A22(J,K,LN)                                    DAUX55  1020
   60 C(J,K,IJ+2) = -SET*A22(K,J,LN)                                    DAUX55  1030
      NNS = NQ2S + NJNT + N                                             DAUX55  1040
      IJK(IS+1,NNS) = IJ+1                                              DAUX55  1050
      IJK(NNS,IS+1) = IJ+2                                              DAUX55  1060
      IJ = IJ+2                                                         DAUX55  1070
   65 CONTINUE                                                          DAUX55  1080
   98 IS = IS+1                                                         DAUX55  1090
   99 CONTINUE                                                          DAUX55  1100
      CALL ELTIME(2,30)                                                 DAUX55  1110
      RETURN                                                            DAUX55  1120
      END                                                               DAUX55  1130
```

```
      SUBROUTINE DAUX55                                                 DAUX55  0010
C                                                          REV 19 09/05/78  DAUX55  0020
      IMPLICIT REAL*8(A-H,O-Z)                                          DAUX55  0030
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           DAUX55  0040
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)        DAUX55  0050
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30), DAUX55  0060
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       DAUX55  0070
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(3,60),HA(3,60),HB(3,60), DAUX55  0080
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),     DAUX55  0090
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)    DAUX55  0100
      COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60), DAUX55  0110
     *               F(3,30),TQ(3,30),WJ(30)                            DAUX55  0120
      COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),   DAUX55  0130
     *               HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12),DAUX55  0140
     *               RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),              DAUX55  0150
     *               KQ1(12),KQ2(12),KQTYPE(12)                         DAUX55  0160
      COMMON/FLXBLE/ HF(4,12,8),B42(3,3,24),V4(3,8),NFLEX(3,8)          DAUX55  0170
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         DAUX55  0180
     *               UNITL,UNITM,UNITT,GRAVTY(3)                        DAUX55  0190
      COMMON/TEMPVS/ C(3,3,400),RHS(3,54),IJK(54,54),IJ,NQ2S            DAUX55  0200
      CALL ELTIME(1,30)                                                 DAUX55  0210
      IS = 0                                                            DAUX55  0220
      DO  99  I=1,NGRND                                                 DAUX55  0230
       IF (ISING(I).LE.0) GO TO 99                                      DAUX55  0240
      IS = IS+1                                                         DAUX55  0250
      IJ = IJ+1                                                         DAUX55  0260
      IJK(IS  ,IS  ) = I                                                DAUX55  0270
      IJK(IS+1,IS+1) = I+1                                              DAUX55  0280
      DO 11 J=1,3                                                       DAUX55  0290
      U1(J,IS  ) = U1(J,I) + W(I)*GRAVTY(J)/G                           DAUX55  0300
      U2(J,IS+1) = U2(J,I)                                              DAUX55  0310
      U1(J,I) = 0.0                                                     DAUX55  0320
      U2(J,I) = 0.0                                                     DAUX55  0330
      DO 10 K=1,3                                                       DAUX55  0340
      C(J,K,IJ  ) = 0.0                                                 DAUX55  0350
   10 C(J,K,IJ+1) = 0.0                                                 DAUX55  0360
      C(J,J,IJ  ) = W(I)/G                                              DAUX55  0370
   11 C(J,J,IJ+1) = PHI(J,I)                                            DAUX55  0380
      IJ = IJ+1                                                         DAUX55  0390
      IF (NFLX.EQ.0) GO TO 19                                           DAUX55  0400
      DO 15 N=1,NFLX                                                    DAUX55  0410
      LN = 0                                                            DAUX55  0420
      IF (NFLEX(1,N).EQ.I) LN = 3*N-2                                   DAUX55  0430
      IF (NFLEX(2,N).EQ.I) LN = 3*N-1                                   DAUX55  0440
      IF (NFLEX(3,N).EQ.I) LN = 3*N                                     DAUX55  0450
      IF (LN.EQ.0) GO TO 15                                             DAUX55  0460
      DO 14 J=1,3                                                       DAUX55  0470
      DO 14 K=1,3                                                       DAUX55  0480
      C(J,K,IJ+1) =  B42(K,J,LN)                                        DAUX55  0490
   14 C(J,K,IJ+2) = -B42(J,K,LN)                                        DAUX55  0500
```

```
39 CONTINUE                                                                    DAUX44  1010
40 IF (NJNT.EQ.0) GO TO 90                                                     DAUX44  1020
   DO 59 M=1,NJNT                                                              DAUX44  1030
   IF (JNT(M).EQ.0) GO TO 59                                                   DAUX44  1040
   DO 58 II=1,3                                                                DAUX44  1050
   LM = 0                                                                      DAUX44  1060
   IF (NFLEX(II,L).EQ.IABS(JNT(M))) LM = 2*M-1                                 DAUX44  1070
   IF (NFLEX(II,L).EQ.M+1)          LM = 2*M                                   DAUX44  1080
   IF (LM.EQ.0) GO TO 58                                                       DAUX44  1090
   IL = NFLEX(II,L)                                                            DAUX44  1100
   IF  (ISING(IL).NE.0)  GO TO 58                                              DAUX44  1110
   NSM = 2*NS+NFLX+NQ+M                                                        DAUX44  1120
   JK = IJK(NSL,NSM)                                                           DAUX44  1130
   KJ = IJK(NSM,NSL)                                                           DAUX44  1140
   IF (JK.GT.0) GO TO 42                                                       DAUX44  1150
   IJK(NSL,NSM) = IJ+1                                                         DAUX44  1160
   IJK(NSM,NSL) = IJ+2                                                         DAUX44  1170
   JK = IJ+1                                                                   DAUX44  1180
   KJ = IJ+2                                                                   DAUX44  1190
   IJ = IJ+2                                                                   DAUX44  1200
   DO 41 I=1,3                                                                 DAUX44  1210
   DO 41 J=1,3                                                                 DAUX44  1220
41 C(I,J,JK) = 0.0                                                             DAUX44  1230
42 LI = 3*L+II-3                                                               DAUX44  1240
   DO 44 I=1,3                                                                 DAUX44  1250
   DO 44 J=1,3                                                                 DAUX44  1260
   DO 43 K=1,3                                                                 DAUX44  1270
43 C(I,J,JK) = C(I,J,JK) + B42(I,K,LI)*RPHI(K,IL)*BI2(J,K,LM)                  DAUX44  1280
44 C(J,I,KJ) = C(I,J,JK)                                                       DAUX44  1290
   IF (IPIN(M).EQ.0 .OR. IPIN(M).GE.2) GO TO 58                                DAUX44  1300
   NSM = 2*NS+NFLX+NQ+NJNT+M                                                   DAUX44  1310
   JK = IJK(NSL,NSM)                                                           DAUX44  1320
   KJ = IJK(NSM,NSL)                                                           DAUX44  1330
   IF (JK.GT.0) GO TO 52                                                       DAUX44  1340
   IJK(NSL,NSM) = IJ+1                                                         DAUX44  1350
   IJK(NSM,NSL) = IJ+2                                                         DAUX44  1360
   JK = IJ+1                                                                   DAUX44  1370
   KJ = IJ+2                                                                   DAUX44  1380
   IJ = IJ+2                                                                   DAUX44  1390
   DO 51 I=1,3                                                                 DAUX44  1400
   DO 51 J=1,3                                                                 DAUX44  1410
51 C(I,J,JK) = 0.0                                                             DAUX44  1420
52 SET = 1.0                                                                   DAUX44  1430
   IF (IL.EQ.M+1) SET = -1.0                                                   DAUX44  1440
   DO 54 I=1,3                                                                 DAUX44  1450
   DO 54 J=1,3                                                                 DAUX44  1460
   DO 53 K=1,3                                                                 DAUX44  1470
53 C(I,J,JK) = C(I,J,JK) + SET*B42(I,K,LI)*RPHI(K,IL)*A22(K,J,LM)              DAUX44  1480
54 C(J,I,KJ) = C(I,J,JK)                                                       DAUX44  1490
   CONTINUE                                                                    DAUX44  1500
59 CONTINUE                                                                    DAUX44  1510
90 CONTINUE                                                                    DAUX44  1520
   CALL ELTIME(2,33)                                                           DAUX44  1530
99 RETURN                                                                      DAUX44  1540
   END                                                                         DAUX44  1550
```

```
      KJ = IJK(NSM,NSL)                                                 DAUX44  0510
      IF (JK.GT.0) GO TO 22                                             DAUX44  0520
      IJK(NSL,NSM) = IJ+1                                               DAUX44  0530
      IJK(NSM,NSL) = IJ+2                                               DAUX44  0540
      JK = IJ+1                                                         DAUX44  0550
      KJ = IJ+2                                                         DAUX44  0560
      IJ = IJ+2                                                         DAUX44  0570
      DO 21 I=1,3                                                       DAUX44  0580
      DO 21 J=1,3                                                       DAUX44  0590
   21 C(I,J,JK) = 0.0                                                   DAUX44  0600
   22 LI = 3*L+II-3                                                     DAUX44  0610
      MJ = 3*M+JJ-3                                                     DAUX44  0620
      DO 24 I=1,3                                                       DAUX44  0630
      DO 24 J=1,3                                                       DAUX44  0640
      DO 23 K=1,3                                                       DAUX44  0650
   23 C(I,J,JK) = C(I,J,JK) + B42(I,K,LI)*RPHI(K,IL)*B42(J,K,MJ)        DAUX44  0660
   24 C(J,I,KJ) = C(I,J,JK)                                             DAUX44  0670
   27 CONTINUE                                                          DAUX44  0680
   28 CONTINUE                                                          DAUX44  0690
   29 CONTINUE                                                          DAUX44  0700
   30 IF (NQ.EQ.0) GO TO 40                                             DAUX44  0710
      DO 39 M=1,NQ                                                      DAUX44  0720
      IF (KQTYPE(M).LT.0) GO TO 39                                      DAUX44  0730
      DO 38 II=1,3                                                      DAUX44  0740
      LM = 0                                                            DAUX44  0750
      IF (NFLEX(II,L).EQ.KQ1(M)) LM = 2*M-1                             DAUX44  0760
      IF (NFLEX(II,L).EQ.KQ2(M)) LM = 2*M                               DAUX44  0770
      IF (LM.EQ.0) GO TO 38                                             DAUX44  0780
      IL = NFLEX(II,L)                                                  DAUX44  0790
      IF  (ISING(IL).NE.0)  GO TO 38                                    DAUX44  0800
      NSM = 2*NS+NFLX+M                                                 DAUX44  0810
      JK = IJK(NSL,NSM)                                                 DAUX44  0820
      KJ = IJK(NSM,NSL)                                                 DAUX44  0830
      IF (JK.GT.0) GO TO 32                                             DAUX44  0840
      IJK(NSL,NSM) = IJ+1                                               DAUX44  0850
      IJK(NSM,NSL) = IJ+2                                               DAUX44  0860
      JK = IJ+1                                                         DAUX44  0870
      KJ = IJ+2                                                         DAUX44  0880
      IJ = IJ+2                                                         DAUX44  0890
      DO 31 I=1,3                                                       DAUX44  0900
      DO 31 J=1,3                                                       DAUX44  0910
      C(I,J,JK) = 0.0                                                   DAUX44  0920
   31 C(I,J,KJ) = 0.0                                                   DAUX44  0930
   32 LI = 3*L+II-3                                                     DAUX44  0940
      DO 33 I=1,3                                                       DAUX44  0950
      DO 33 J=1,3                                                       DAUX44  0960
      DO 33 K=1,3                                                       DAUX44  0970
      C(I,J,JK) = C(I,J,JK) + B42(I,K,LI)*RPHI(K,IL)*A23(K,J,LM)        DAUX44  0980
   33 C(I,J,KJ) = C(I,J,KJ) + B32(I,K,LM)*RPHI(K,IL)*B42(J,K,LI)        DAUX44  0990
   38 CONTINUE                                                          DAUX44  1000
```

```
      SUBROUTINE DAUX44                                                      DAUX44  0010
C                                                          REV 19 09/05/78   DAUX44  0020
      IMPLICIT REAL*8(A-H,O-Z)                                               DAUX44  0030
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,                DAUX44  0040
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)             DAUX44  0050
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),     DAUX44  0060
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)            DAUX44  0070
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(3,60),HA(3,60),HB(3,60),      DAUX44  0080
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),          DAUX44  0090
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)         DAUX44  0100
      COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60),     DAUX44  0110
     *               F(3,30),TQ(3,30),WJ(30)                                 DAUX44  0120
      COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),        DAUX44  0130
     *               HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12),     DAUX44  0140
     *               RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),                   DAUX44  0150
     *               KQ1(12),KQ2(12),KQTYPE(12)                              DAUX44  0160
      COMMON/FLXBLE/ HF(4,12,8),B42(3,3,24),V4(3,8),NFLEX(3,8)               DAUX44  0170
      COMMON/TEMPVS/ C(3,3,400),RHS(3,54),IJK(54,54),IJ,NQ2S                 DAUX44  0180
      IF (NFLX.EQ.0) GO TO 99                                                DAUX44  0190
      CALL ELTIME(1,33)                                                      DAUX44  0200
      DO 90 L=1,NFLX                                                         DAUX44  0210
      N1 = NFLEX(1,L)                                                        DAUX44  0220
      N2 = NFLEX(2,L)                                                        DAUX44  0230
      N3 = NFLEX(3,L)                                                        DAUX44  0240
      IJ = IJ+1                                                              DAUX44  0250
      DO 10 I=1,3                                                            DAUX44  0260
      DO 10 J=1,3                                                            DAUX44  0270
      C(I,J,IJ) = 0.0                                                        DAUX44  0280
      DO 10 K=1,3                                                            DAUX44  0290
   10 C(I,J,IJ) = C(I,J,IJ) + B42(I,K,3*L-2)*RPHI(K,N1)*B42(J,K,3*L-2)       DAUX44  0300
     *                      + B42(I,K,3*L-1)*RPHI(K,N2)*B42(J,K,3*L-1)       DAUX44  0310
     *                      + B42(I,K,3*L  )*RPHI(K,N3)*B42(J,K,3*L  )       DAUX44  0320
      NSL = 2*NS+L                                                           DAUX44  0330
      IJK(NSL,NSL) = IJ                                                      DAUX44  0340
      DO 20 I=1,3                                                            DAUX44  0350
      RHS(I,NSL) = -V4(I,L)                                                  DAUX44  0360
      DO 20 J=1,3                                                            DAUX44  0370
   20 RHS(I,NSL) = RHS(I,NSL) + B42(I,J,3*L-2)*U2(I,N1)                      DAUX44  0380
     *                        + B42(I,J,3*L-1)*U2(I,N2)                     DAUX44  0390
     *                        + B42(I,J,3*L  )*U2(I,N3)                     DAUX44  0400
      IF (L.EQ.NFLX) GO TO 30                                                DAUX44  0410
      LP1 = L+1                                                              DAUX44  0420
      DO 29 M=LP1,NFLX                                                       DAUX44  0430
      DO 28 II=1,3,2                                                         DAUX44  0440
      IL = NFLEX(II,L)                                                       DAUX44  0450
      IF  (ISING(IL).NE.0)  GO TO 28                                         DAUX44  0460
      DO 27 JJ=1,3,2                                                         DAUX44  0470
      IF (NFLEX(II,L).NE.NFLEX(JJ,M)) GO TO 27                               DAUX44  0480
      NSM = 2*NS+M                                                           DAUX44  0490
      JK = IJK(NSL,NSM)                                                      DAUX44  0500
```

```
C     FOR ANY M>N SUCH THAT K2(N) = K2(M)                                    DAUX33  2010
C                                                                            DAUX33  2020
C                                        -1                                  DAUX33  2030
C     C33(N,M) = C(N,M) + B31(N,K2)  M  (K2)A13(K2,M)                        DAUX33  2040
C                                        -1                                  DAUX33  2050
C                       + B32(N,K2)PHI   (K2)A23(K2,M)                       DAUX33  2060
C                                                                            DAUX33  2070
C                                        -1                                  DAUX33  2080
C     C33(M,N) = C(M,N) + B31(M,K2)  M  (K2)A13(K2,N)                        DAUX33  2090
C                                        -1                                  DAUX33  2100
C                       + B32(M,K2)PHI   (K2)A23(K2,N)                       DAUX33  2110
C                                                                            DAUX33  2120
      IJ = IJ+1                                                              DAUX33  2130
      IJK(MNS,NNS) = IJ                                                      DAUX33  2140
      IJK(NNS,MNS) = IJ+1                                                    DAUX33  2150
      DO 81 J=1,3                                                            DAUX33  2160
      DO 81 I=1,3                                                            DAUX33  2170
      C(I,J,IJ  ) = 0.0                                                      DAUX33  2180
   81 C(I,J,IJ+1) = 0.0                                                      DAUX33  2190
      IJ = IJ+1                                                              DAUX33  2200
   82 JJ = IJK(MNS,NNS)                                                      DAUX33  2210
      DO 84 I=1,3                                                            DAUX33  2220
      DO 84 J=1,3                                                            DAUX33  2230
      SUM = C(I,J,JJ)                                                        DAUX33  2240
      TUM = C(I,J,JJ+1)                                                      DAUX33  2250
      DO 83 K=1,3                                                            DAUX33  2260
      SUM = SUM + B31(I,K,2*N  )*  RW(  K2)*A13(K,J,2*M  )                   DAUX33  2270
     *          + B32(I,K,2*N  )*RPHI(K,K2)*A23(K,J,2*M  )                   DAUX33  2280
   83 TUM = TUM + B31(I,K,2*M  )*  RW(  K2)*A13(K,J,2*N  )                   DAUX33  2290
     *          + B32(I,K,2*M  )*RPHI(K,K2)*A23(K,J,2*N  )                   DAUX33  2300
      C(I,J,JJ  ) = SUM                                                      DAUX33  2310
   84 C(I,J,JJ+1) = TUM                                                      DAUX33  2320
   85 CONTINUE                                                               DAUX33  2330
   90 CONTINUE                                                               DAUX33  2340
   91 CALL ELTIME(2,19)                                                      DAUX33  2350
      RETURN                                                                 DAUX33  2360
      END                                                                    DAUX33  2370
```

```
      DO 74 I=1,3                                                        DAUX33  1510
      DO 74 J=1,3                                                        DAUX33  1520
      SUM = C(I,J,JJ)                                                    DAUX33  1530
      TUM = C(I,J,JJ+1)                                                  DAUX33  1540
      DO 73 K=1,3                                                        DAUX33  1550
      SUM = SUM + B31(I,K,2*N-1)*  RW(  K1)*A13(K,J,2*M  )               DAUX33  1560
     *          + B32(I,K,2*N-1)*RPHI(K,K1)*A23(K,J,2*M  )              DAUX33  1570
   73 TUM = TUM + B31(I,K,2*M  )*  RW(  K1)*A13(K,J,2*N-1)               DAUX33  1580
     *          + B32(I,K,2*M  )*RPHI(K,K1)*A23(K,J,2*N-1)              DAUX33  1590
      C(I,J,JJ  ) = SUM                                                  DAUX33  1600
   74 C(I,J,JJ+1) = TUM                                                  DAUX33  1610
   75 IF  (ISING(K2).NE.0)  GO TO 85                                     DAUX33  1620
      IF (K2.NE.KQ1(M))        GO TO 80                                  DAUX33  1630
      IF (IJK(MNS,NNS).NE.0) GO TO 77                                    DAUX33  1640
C                                                                        DAUX33  1650
C     FOR ANY M>N SUCH THAT K2(N) = K1(M)                                DAUX33  1660
C                                                                        DAUX33  1670
C                                          -1                            DAUX33  1680
C     C33(N,M) = C(N,M) + B31(N,K2)  M  (K2)A13(K1,M)                    DAUX33  1690
C                                          -1                            DAUX33  1700
C                       + B32(N,K2)PHI   (K2)A23(K1,M)                   DAUX33  1710
C                                                                        DAUX33  1720
C                                          -1                            DAUX33  1730
C     C33(M,N) = C(M,N) + B31(M,K1)  M  (K2)A13(K2,N)                    DAUX33  1740
C                                         -1                             DAUX33  1750
C                       + B32(M,K1)PHI   (K2)A23(K2,N)                   DAUX33  1760
C                                                                        DAUX33  1770
      IJ = IJ+1                                                          DAUX33  1780
      IJK(MNS,NNS) = IJ                                                  DAUX33  1790
      IJK(NNS,MNS) = IJ+1                                                DAUX33  1800
      DO 76 J=1,3                                                        DAUX33  1810
      DO 76 I=1,3                                                        DAUX33  1820
      C(I,J,IJ  ) = 0.0                                                  DAUX33  1830
   76 C(I,J,IJ+1) = 0.0                                                  DAUX33  1840
      IJ = IJ+1                                                          DAUX33  1850
   77 JJ = IJK(MNS,NNS)                                                  DAUX33  1860
      DO 79 I=1,3                                                        DAUX33  1870
      DO 79 J=1,3                                                        DAUX33  1880
      SUM = C(I,J,JJ)                                                    DAUX33  1890
      TUM = C(I,J,JJ+1)                                                  DAUX33  1900
      DO 78 K=1,3                                                        DAUX33  1910
      SUM = SUM + B31(I,K,2*N  )*  RW(  K2)*A13(K,J,2*M-1)               DAUX33  1920
     *          + B32(I,K,2*N  )*RPHI(K,K2)*A23(K,J,2*M-1)              DAUX33  1930
   78 TUM = TUM + B31(I,K,2*M-1)*  RW(  K2)*A13(K,J,2*N  )               DAUX33  1940
     *          + B32(I,K,2*M-1)*RPHI(K,K2)*A23(K,J,2*N  )              DAUX33  1950
      C(I,J,JJ  ) = SUM                                                  DAUX33  1960
   79 C(I,J,JJ+1) = TUM                                                  DAUX33  1970
   80 IF (K2.NE.KQ2(M))       GO TO 85                                   DAUX33  1980
      IF (IJK(MNS,NNS).NE.0) GO TO 82                                    DAUX33  1990
C                                                                        DAUX33  2000
```

```
C                                                                       DAUX33  1010
C                                        -1                             DAUX33  1020
C     C33(M,N) = C(M,N) + B31(M,K1)  M  (K1)A13(K1,N)                   DAUX33  1030
C                                        -1                             DAUX33  1040
C                       + B32(M,K1)PHI   (K1)A23(K1,N)                  DAUX33  1050
C                                                                       DAUX33  1060
      IJ = IJ+1                                                         DAUX33  1070
      IJK(MNS,NNS) = IJ                                                 DAUX33  1080
      IJK(NNS,MNS) = IJ+1                                               DAUX33  1090
      DO 66 J=1,3                                                       DAUX33  1100
      DO 66 I=1,3                                                       DAUX33  1110
      C(I,J,IJ  ) = 0.0                                                 DAUX33  1120
   66 C(I,J,IJ+1) = 0.0                                                 DAUX33  1130
      IJ = IJ+1                                                         DAUX33  1140
   67 JJ = IJK(MNS,NNS)                                                 DAUX33  1150
      DO 69 I=1,3                                                       DAUX33  1160
      DO 69 J=1,3                                                       DAUX33  1170
      SUM = C(I,J,JJ)                                                   DAUX33  1180
      TUM = C(I,J,JJ+1)                                                 DAUX33  1190
      DO 68 K=1,3                                                       DAUX33  1200
      SUM = SUM + B31(I,K,2*N-1)*  RW(  K1)*A13(K,J,2*M-1)              DAUX33  1210
     *          + B32(I,K,2*N-1)*RPHI(K,K1)*A23(K,J,2*M-1)              DAUX33  1220
   68 TUM = TUM + B31(I,K,2*M-1)*  RW(  K1)*A13(K,J,2*N-1)              DAUX33  1230
     *          + B32(I,K,2*M-1)*RPHI(K,K1)*A23(K,J,2*N-1)              DAUX33  1240
      C(I,J,JJ  ) = SUM                                                 DAUX33  1250
   69 C(I,J,JJ+1) = TUM                                                 DAUX33  1260
   70 IF (K1.NE.KQ2(M))         GO TO 75                                DAUX33  1270
      IF (IJK(MNS,NNS).NE.0) GO TO 72                                   DAUX33  1280
C                                                                       DAUX33  1290
C     FOR ANY M>N SUCH THAT K1(N) = K2(M)                               DAUX33  1300
C                                                                       DAUX33  1310
C                                        -1                             DAUX33  1320
C     C33(N,M) = C(N,M) + B31(N,K1)  M  (K1)A13(K2,M)                   DAUX33  1330
C                                        -1                             DAUX33  1340
C                       + B32(N,K1)PHI   (K1)A23(K2,M)                  DAUX33  1350
C                                                                       DAUX33  1360
C                                        -1                             DAUX33  1370
C     C33(M,N) = C(M,N) + B31(M,K2)  M  (K1)A13(K1,N)                   DAUX33  1380
C                                        -1                             DAUX33  1390
C                       + B32(M,K2)PHI   (K1)A23(K1,N)                  DAUX33  1400
C                                                                       DAUX33  1410
      IJ = IJ+1                                                         DAUX33  1420
      IJK(MNS,NNS) = IJ                                                 DAUX33  1430
      IJK(NNS,MNS) = IJ+1                                               DAUX33  1440
      DO 71 J=1,3                                                       DAUX33  1450
      DO 71 I=1,3                                                       DAUX33  1460
      C(I,J,IJ  ) = 0.0                                                 DAUX33  1470
   71 C(I,J,IJ+1) = 0.0                                                 DAUX33  1480
      IJ = IJ+1                                                         DAUX33  1490
   72 JJ = IJK(MNS,NNS)                                                 DAUX33  1500
```

```
C                                                                       DAUX33  0510
C              - B35(N,N)                                               DAUX33  0520
C                                                                       DAUX33  0530
      IJ = IJ+1                                                         DAUX33  0540
      IJK(NNS,NNS) = IJ                                                 DAUX33  0550
      IF (KQTYPE(N).EQ.2) GO TO 51                                      DAUX33  0560
      IF (KQTYPE(N).EQ.4) GO TO 51                                      DAUX33  0570
      DO 65 I=1,3                                                       DAUX33  0580
      DO 65 J=1,3                                                       DAUX33  0590
      SUM = -HHT(I,J,N)                                                 DAUX33  0600
      IF (I.EQ.J) SUM = 1.0+SUM                                         DAUX33  0610
      DO 64 K=1,3                                                       DAUX33  0620
   64 SUM = SUM + B31(I,K,2*N-1)*  RW(  K1)*A13(K,J,2*N-1)              DAUX33  0630
     *           + B31(I,K,2*N  )*  RW(  K2)*A13(K,J,2*N  )             DAUX33  0640
     *           + B32(I,K,2*N-1)*RPHI(K,K1)*A23(K,J,2*N-1)             DAUX33  0650
     *           + B32(I,K,2*N  )*RPHI(K,K2)*A23(K,J,2*N  )             DAUX33  0660
   65 C(I,J,IJ) = SUM                                                   DAUX33  0670
      GO TO 59                                                          DAUX33  0680
C                                                                       DAUX33  0690
C     FOR KQTYPE = 2 OR 4, SET C33(N,N) = B*I                           DAUX33  0700
C     WHERE B = SUM OF DIAGONAL ELEMENTS OF                             DAUX33  0710
C              -1                      -1                               DAUX33  0720
C     (B31)(M)  (A13) + (B32)(PHI)  (A23)                               DAUX33  0730
C                                                                       DAUX33  0740
   51 SUM = 0.0                                                         DAUX33  0750
      DO 55 I=1,3                                                       DAUX33  0760
      DO 55 K=1,3                                                       DAUX33  0770
   55 SUM = SUM + B31(I,K,2*N-1)*  RW(  K1)*A13(K,I,2*N-1)              DAUX33  0780
     *           + B31(I,K,2*N  )*  RW(  K2)*A13(K,I,2*N  )             DAUX33  0790
     *           + B32(I,K,2*N-1)*RPHI(K,K1)*A23(K,I,2*N-1)             DAUX33  0800
     *           + B32(I,K,2*N  )*RPHI(K,K2)*A23(K,I,2*N  )             DAUX33  0810
      DO 57 I=1,3                                                       DAUX33  0820
      DO 56 J=1,3                                                       DAUX33  0830
   56 C(I,J,IJ) = 0.0                                                   DAUX33  0840
   57 C(I,I,IJ) = SUM                                                   DAUX33  0850
   59 IF (N.EQ.NQ) GO TO 90                                             DAUX33  0860
      N1 = N+1                                                          DAUX33  0870
      DO 85 M=N1,NQ                                                     DAUX33  0880
      IF (KQTYPE(M).LT.0) GO TO 85                                      DAUX33  0890
      MNS = NQ2S - NQ + M                                               DAUX33  0900
      IF  (ISING(K1).NE.0)  GO TO 75                                    DAUX33  0910
      IF (K1.NE.KQ1(M))        GO TO 70                                 DAUX33  0920
      IF (IJK(MNS,NNS).NE.0) GO TO 67                                   DAUX33  0930
C                                                                       DAUX33  0940
C     FOR ANY M>N SUCH THAT K1(N) = K1(M)                               DAUX33  0950
C                                                                       DAUX33  0960
C                                          -1                           DAUX33  0970
C     C33(N,M) = C(N,M) + B31(N,K1)  M  (K1)A13(K1,M)                   DAUX33  0980
C                                           -1                          DAUX33  0990
C                       + B32(N,K1)PHI  (K1)A23(K1,M)                   DAUX33  1000
```

```
      SUBROUTINE DAUX33                                                      DAUX33  0010
C                                                          REV 19 09/05/78  DAUX33  0020
C     CALLED BY SUBROUTINE DAUX TO COMPUTE                                 DAUX33  0030
C                                                                          DAUX33  0040
C                      -1                      -1                          DAUX33  0050
C     (C33) = (B31)(M)  (A13) + (B32)(PHI)  (A23) - (B35)                  DAUX33  0060
C                                                                          DAUX33  0070
C                      -1                      -1                          DAUX33  0080
C     (R3)  = (B31)(M)  (U1)  + (B32)(PHI)  (U2)  - (V3)                   DAUX33  0090
C                                                                          DAUX33  0100
      IMPLICIT REAL*8 (A-H,O-Z)                                            DAUX33  0110
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,              DAUX33  0120
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)           DAUX33  0130
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),   DAUX33  0140
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)          DAUX33  0150
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(3,60),HA(3,60),HB(3,60),    DAUX33  0160
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),        DAUX33  0170
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)       DAUX33  0180
      COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60),   DAUX33  0190
     *               F(3,30),TQ(3,30),WJ(30)                               DAUX33  0200
      COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),      DAUX33  0210
     *               HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12),   DAUX33  0220
     *               RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),                 DAUX33  0230
     *               KQ1(12),KQ2(12),KQTYPE(12)                            DAUX33  0240
      COMMON/TEMPVS/ C(3,3,400),RHS(3,54),IJK(54,54),IJ,NQ2S               DAUX33  0250
      CALL ELTIME(1,19)                                                    DAUX33  0260
      DO 90 N=1,NQ                                                         DAUX33  0270
      IF (KQTYPE(N).LT.0) GO TO 90                                         DAUX33  0280
      K1 = KQ1(N)                                                          DAUX33  0290
      K2 = KQ2(N)                                                          DAUX33  0300
      NNS = NQ2S - NQ + N                                                  DAUX33  0310
C                                                                          DAUX33  0320
C                        -1                          -1                    DAUX33  0330
C     RHS(N) = B31(N,K1)M  (K1)U1(K1) + B32(N,K1)PHI  (K1)U2(K1)           DAUX33  0340
C                        -1                          -1                    DAUX33  0350
C            + B31(N,K2)M  (K2)U1(K2) + B32(N,K2)PHI  (K2)U2(K2)           DAUX33  0360
C                                                                          DAUX33  0370
C            - V3(N)                                                       DAUX33  0380
C                                                                          DAUX33  0390
      DO 63 I=1,3                                                          DAUX33  0400
      SUM = 0.0                                                            DAUX33  0410
      DO 62 K=1,3                                                          DAUX33  0420
   62 SUM = SUM + B31(I,K,2*N-1)*U1(K,K1) + B32(I,K,2*N-1)*U2(K,K1)        DAUX33  0430
     *           + B31(I,K,2*N  )*U1(K,K2) + B32(I,K,2*N  )*U2(K,K2)      DAUX33  0440
   63 RHS(I,NNS) = SUM - V3(I,N)                                           DAUX33  0450
C                                                                          DAUX33  0460
C                          -1                              -1              DAUX33  0470
C     C33(N,N) = B31(N,K1)M  (K1)A13(K1,N) + B32(N,K1)PHI  (K1)A23(K1,N)   DAUX33  0480
C                          -1                              -1              DAUX33  0490
C              + B31(N,K2)M  (K2)A13(K2,N) + B32(N,K2)PHI  (K2)A23(K2,N)   DAUX33  0500
```

```
      DO 47 J=1,3                                                        DAUX32  1010
      DO 47 I=1,3                                                        DAUX32  1020
      C(I,J,IJ  ) = 0.0                                                  DAUX32  1030
   47 C(I,J,IJ+1) = 0.0                                                  DAUX32  1040
      IJ = IJ+1                                                          DAUX32  1050
   48 JJ = IJK(KJNT,NNS)                                                 DAUX32  1060
      DO 50 I=1,3                                                        DAUX32  1070
      DO 50 J=1,3                                                        DAUX32  1080
      SUM = C(I,J,JJ)                                                    DAUX32  1090
      TUM = C(I,J,JJ+1)                                                  DAUX32  1100
      DO 49 K=1,3                                                        DAUX32  1110
      SUM = SUM + A22(K,I,2*L-1 ) * RPHI(K,K1) * A23(K,J,2*N-1 )         DAUX32  1120
   49 TUM = TUM + B32(I,K,2*N-1 ) * RPHI(K,K1) * A22(K,J,2*L-1 )         DAUX32  1130
      C(I,J,JJ)   = SUM                                                  DAUX32  1140
   50 C(I,J,JJ+1) = TUM                                                  DAUX32  1150
   51 IF (IABS(JNT(L)).NE.K2) GO TO 56                                   DAUX32  1160
      IF  (ISING(K2).NE.0)  GO TO 56                                     DAUX32  1170
C                                                                        DAUX32  1180
C     FOR ANY L SUCH THAT JNT(L) = K2                                    DAUX32  1190
C                                                                        DAUX32  1200
C                                -1                                      DAUX32  1210
C     C23(L,N) = B22(L,K2)PHI   (K2)A23(K2,N)                            DAUX32  1220
C                                                                        DAUX32  1230
C                                -1                                      DAUX32  1240
C     C32(N,L) = B32(N,K2)PHI   (K2)A22(K2,L)                            DAUX32  1250
C                                                                        DAUX32  1260
      KJNT = NQSJNT + L                                                  DAUX32  1270
      IF (IJK(KJNT,NNS).NE.0) GO TO 53                                   DAUX32  1280
      IJ = IJ+1                                                          DAUX32  1290
      IJK(KJNT,NNS) = IJ                                                 DAUX32  1300
      IJK(NNS,KJNT) = IJ+1                                               DAUX32  1310
      DO 52 J=1,3                                                        DAUX32  1320
      DO 52 I=1,3                                                        DAUX32  1330
      C(I,J,IJ  ) = 0.0                                                  DAUX32  1340
   52 C(I,J,IJ+1) = 0.0                                                  DAUX32  1350
      IJ = IJ+1                                                          DAUX32  1360
   53 JJ = IJK(KJNT,NNS)                                                 DAUX32  1370
      DO 55 I=1,3                                                        DAUX32  1380
      DO 55 J=1,3                                                        DAUX32  1390
      SUM = C(I,J,JJ)                                                    DAUX32  1400
      TUM = C(I,J,JJ+1)                                                  DAUX32  1410
      DO 54 K=1,3                                                        DAUX32  1420
      SUM = SUM + A22(K,I,2*L-1 ) * RPHI(K,K2) * A23(K,J,2*N   )         DAUX32  1430
   54 TUM = TUM + B32(I,K,2*N   ) * RPHI(K,K2) * A22(K,J,2*L-1 )         DAUX32  1440
      C(I,J,JJ)   = SUM                                                  DAUX32  1450
   55 C(I,J,JJ+1) = TUM                                                  DAUX32  1460
   56 CONTINUE                                                           DAUX32  1470
   60 CONTINUE                                                           DAUX32  1480
      CALL ELTIME(2,18)                                                  DAUX32  1490
      RETURN                                                             DAUX32  1500
                                                                         DAUX32  1510
      END
```

```
      DO 41 K=1,3                                                          DAUX32  0510
      SUM = SUM + A22(K,I,2*K1-2) * RPHI(K,K1) * A23(K,J,2*N-1 )           DAUX32  0520
   41 TUM = TUM + B32(I,K,2*N-1 ) * RPHI(K,K1) * A22(K,J,2*K1-2)           DAUX32  0530
      C(I,J,IJ  ) = -SUM                                                   DAUX32  0540
   42 C(I,J,IJ+1) = -TUM                                                   DAUX32  0550
      IJ = IJ+1                                                            DAUX32  0560
   43 IF (K2.LE.1)         GO TO 46                                        DAUX32  0570
      IF (IABS(JNT(K2-1)).EQ.0) GO TO 46                                   DAUX32  0580
      IF (IPIN(K2-1).EQ.0 .OR. IPIN(K2-1).GE.2) GO TO 46                   DAUX32  0590
      IF  (ISING(K2).NE.0)  GO TO 46                                       DAUX32  0600
C                                                                          DAUX32  0610
C                                    -1                                    DAUX32  0620
C     C23(K2-1,N) = B22(K2-1,K2)PHI  (K2)A23(K2,N)                         DAUX32  0630
C                                                                          DAUX32  0640
C                                   -1                                     DAUX32  0650
C     C32(N,K2-1) = B32(N,K2)PHI  (K2)A22(K2,K2-1)                         DAUX32  0660
C                                                                          DAUX32  0670
      KJNT = NQSJNT + K2 - 1                                               DAUX32  0680
      IJ = IJ+1                                                            DAUX32  0690
      IJK(KJNT,NNS) = IJ                                                   DAUX32  0700
      IJK(NNS,KJNT) = IJ+1                                                 DAUX32  0710
      DO 45 I=1,3                                                          DAUX32  0720
      DO 45 J=1,3                                                          DAUX32  0730
      SUM = 0.0                                                            DAUX32  0740
      TUM = 0.0                                                            DAUX32  0750
      DO 44 K=1,3                                                          DAUX32  0760
      SUM = SUM + A22(K,I,2*K2-2) * RPHI(K,K2) * A23(K,J,2*N   )           DAUX32  0770
   44 TUM = TUM + B32(I,K,2*N   ) * RPHI(K,K2) * A22(K,J,2*K2-2)           DAUX32  0780
      C(I,J,IJ  ) = -SUM                                                   DAUX32  0790
   45 C(I,J,IJ+1) = -TUM                                                   DAUX32  0800
      IJ = IJ+1                                                            DAUX32  0810
   46 IF (NJNT.LE.0)       GO TO 60                                        DAUX32  0820
      DO 56 L=1,NJNT                                                       DAUX32  0830
      IF (IPIN(L).EQ.0 .OR. IPIN(L).GE.2) GO TO 56                         DAUX32  0840
      IF (IABS(JNT(L)).NE.K1) GO TO 51                                     DAUX32  0850
      IF  (ISING(K1).NE.0)  GO TO 51                                       DAUX32  0860
C                                                                          DAUX32  0870
C     FOR ANY L SUCH THAT JNT(L) = K1                                      DAUX32  0880
C                                                                          DAUX32  0890
C                                   -1                                     DAUX32  0900
C     C23(L,N) = B22(L,K1)PHI  (K1)A23(K1,N)                               DAUX32  0910
C                                                                          DAUX32  0920
C                                   -1                                     DAUX32  0930
C     C32(N,L) = B32(N,K1)PHI  (K1)A22(K1,L)                               DAUX32  0940
C                                                                          DAUX32  0950
      KJNT = NQSJNT + L                                                    DAUX32  0960
      IF (IJK(KJNT,NNS).NE.0) GO TO 48                                     DAUX32  0970
      IJ = IJ+1                                                            DAUX32  0980
      IJK(KJNT,NNS) = IJ                                                   DAUX32  0990
      IJK(NNS,KJNT) = IJ+1                                                 DAUX32  1000
```

```
      SUBROUTINE DAUX32                                                      DAUX32  0010
C                                                          REV 19 09/05/78  DAUX32  0020
C     CALLED BY SUBROUTINE DAUX TO COMPUTE                                 DAUX32  0030
C                                                                          DAUX32  0040
C                         -1                                               DAUX32  0050
C     (C23) = (B22)(PHI)  (A23)                                            DAUX32  0060
C                                                                          DAUX32  0070
C                         -1                                               DAUX32  0080
C     (C32) = (B32)(PHI)  (A22)                                            DAUX32  0090
C                                                                          DAUX32  0100
      IMPLICIT REAL*8 (A-H,O-Z)                                            DAUX32  0110
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,              DAUX32  0120
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)           DAUX32  0130
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(3,60),HA(3,60),HB(3,60),    DAUX32  0140
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),        DAUX32  0150
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)       DAUX32  0160
      COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60),   DAUX32  0170
     *               F(3,30),TQ(3,30),WJ(30)                               DAUX32  0180
      COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),      DAUX32  0190
     *               HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12),   DAUX32  0200
     *               RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),                 DAUX32  0210
     *               KQ1(12),KQ2(12),KQTYPE(12)                            DAUX32  0220
      COMMON/TEMPVS/ C(3,3,400),RHS(3,54),IJK(54,54),IJ,NQ2S               DAUX32  0230
     *              ,DN(3,3),DM(3,3),BN(3)                                 DAUX32  0240
      CALL ELTIME(1,18)                                                    DAUX32  0250
      NQSJNT = NQ2S + NJNT                                                 DAUX32  0260
      DO 60 N=1,NQ                                                         DAUX32  0270
      IF (KQTYPE(N).LT.0) GO TO 60                                         DAUX32  0280
      K1 = KQ1(N)                                                          DAUX32  0290
      K2 = KQ2(N)                                                          DAUX32  0300
      NNS = NQ2S - NQ + N                                                  DAUX32  0310
      IF (K1.LE.1)        GO TO 43                                         DAUX32  0320
      IF (IABS(JNT(K1-1)).EQ.0) GO TO 43                                   DAUX32  0330
      IF (IPIN(K1-1).EQ.0 .OR. IPIN(K1-1).GE.2) GO TO 43                   DAUX32  0340
      IF  (ISING(K1).NE.0)  GO TO 43                                       DAUX32  0350
C                                                                          DAUX32  0360
C                                    -1                                    DAUX32  0370
C     C23(K1-1,N) = B22(K1-1,K1)PHI   (K1)A23(K1,N)                        DAUX32  0380
C                                                                          DAUX32  0390
C                                   -1                                     DAUX32  0400
C     C32(N,K1-1) = B32(N,K1)PHI   (K1)A22(K1,K1-1)                        DAUX32  0410
C                                                                          DAUX32  0420
      KJNT = NQSJNT + K1 - 1                                               DAUX32  0430
      IJ = IJ+1                                                            DAUX32  0440
      IJK(KJNT,NNS) = IJ                                                   DAUX32  0450
      IJK(NNS,KJNT) = IJ+1                                                 DAUX32  0460
      DO 42 I=1,3                                                          DAUX32  0470
      DO 42 J=1,3                                                          DAUX32  0480
      SUM = 0.0                                                            DAUX32  0490
      TUM = 0.0                                                            DAUX32  0500
```

```
C               + B32(N,K1)PHI   (K1)A21(K1,L)                          DAUX31  1010
C                                                                       DAUX31  1020
      MQ = NQ2S + L                                                     DAUX31  1030
      IF (IJK(MQ,NNS).NE.0) GO TO 18                                    DAUX31  1040
      IJ = IJ+1                                                         DAUX31  1050
      IJK(MQ,NNS) = IJ                                                  DAUX31  1060
      IJK(NNS,MQ) = IJ+1                                                DAUX31  1070
      DO 17 J=1,3                                                       DAUX31  1080
      DO 17 I=1,3                                                       DAUX31  1090
      C(I,J,IJ  ) = 0.0                                                 DAUX31  1100
   17 C(I,J,IJ+1) = 0.0                                                 DAUX31  1110
      IJ = IJ+1                                                         DAUX31  1120
   18 JJ = IJK(MQ,NNS)                                                  DAUX31  1130
      DO 20 I=1,3                                                       DAUX31  1140
      DO 20 J=1,3                                                       DAUX31  1150
      SUM = C(I,J,JJ) + RW(K1)*A13(I,J,2*N-1)                           DAUX31  1160
      TUM = C(I,J,JJ+1) + B31(I,J,2*N-1)*RW(K1)                         DAUX31  1170
      DO 19 K=1,3                                                       DAUX31  1180
      SUM = SUM + B12(I,K,2*L-1)*RPHI(K,K1)*A23(K,J,2*N-1)              DAUX31  1190
   19 TUM = TUM + B32(I,K,2*N-1)*RPHI(K,K1)*B12(J,K,2*L-1)              DAUX31  1200
      C(I,J,JJ)   = SUM                                                 DAUX31  1210
   20 C(I,J,JJ+1) = TUM                                                 DAUX31  1220
   21 IF (IABS(JNT(L)).NE.K2) GO TO 26                                  DAUX31  1230
      IF  (ISING(K2).NE.0)  GO TO 26                                    DAUX31  1240
C                                                                       DAUX31  1250
C     FOR ANY L SUCH THAT JNT(L) = K2                                   DAUX31  1260
C                                                                       DAUX31  1270
C                              -1                                       DAUX31  1280
C     C13(L,N) = B11(L,K2)M     (K2)A13(K2,N)                           DAUX31  1290
C                             -1                                        DAUX31  1300
C               + B12(L,K2)PHI   (K2)A23(K2,N)                          DAUX31  1310
C                                                                       DAUX31  1320
C                              -1                                       DAUX31  1330
C     C31(N,L) = B31(N,K2)M     (K2)A11(K2,L)                           DAUX31  1340
C                             -1                                        DAUX31  1350
C               + B32(N,K2)PHI   (K2)A21(K2,L)                          DAUX31  1360
C                                                                       DAUX31  1370
      MQ = NQ2S + L                                                     DAUX31  1380
      IF (IJK(MQ,NNS).NE.0) GO TO 23                                    DAUX31  1390
      IJ = IJ+1                                                         DAUX31  1400
      IJK(MQ,NNS) = IJ                                                  DAUX31  1410
      IJK(NNS,MQ) = IJ+1                                                DAUX31  1420
      DO 22 J=1,3                                                       DAUX3I  1430
      DO 22 I=1,3                                                       DAUX31  1440
      C(I,J,IJ  ) = 0.0                                                 DAUX31  1450
   22 C(I,J,IJ+1) = 0.0                                                 DAUX31  1460
      IJ = IJ+1                                                         DAUX31  1470
   23 JJ = IJK(MQ,NNS)                                                  DAUX31  1480
      DO 25 I=1,3                                                       DAUX31  1490
      DO 25 J=1,3                                                       DAUX31  1500
      SUM = C(I,J,JJ) + RW(K2)*A13(I,J,2*N)                             DAUX31  1510
      TUM = C(I,J,JJ+1) + B31(I,J,2*N)*RW(K2)                           DAUX31  1520
      DO 24 K=1,3                                                       DAUX31  1530
      SUM = SUM + B12(I,K,2*L-1)*RPHI(K,K2)*A23(K,J,2*N  )              DAUX31  1540
   24 TUM = TUM + B32(I,K,2*N  )*RPHI(K,K2)*B12(J,K,2*L-1)              DAUX31  1550
      C(I,J,JJ)   = SUM                                                 DAUX31  1560
   25 C(I,J,JJ+1) = TUM                                                 DAUX31  1570
   26 CONTINUE                                                          DAUX31  1580
   30 CONTINUE                                                          DAUX31  1590
   31 CALL ELTIME(2,17)                                                 DAUX31  1600
      RETURN                                                            DAUX31  1610
      END                                                               DAUX31  1620
```

```
      TUM = -B31(I,J,2*N-1)*RW(K1)                                      DAUX31  0510
      DO 11 K=1,3                                                       DAUX31  0520
      SUM = SUM + B12(I,K,2*K1-2)*RPHI(K,K1)*A23(K,J,2*N-1 )            DAUX31  0530
   11 TUM = TUM + B32(I,K,2*N-1 )*RPHI(K,K1)*B12(J,K,2*K1-2)            DAUX31  0540
      C(I,J,IJ)    = SUM                                                DAUX31  0550
   12 C(I,J,IJ+1) = TUM                                                 DAUX31  0560
      IJ = IJ+1                                                         DAUX31  0570
   13 IF (K2.LE.1) GO TO 16                                             DAUX31  0580
      IF (IABS(JNT(K2-1)).EQ.0) GO TO 16                                DAUX31  0590
      IF  (ISING(K2).NE.0)  GO TO 16                                    DAUX31  0600
C                                                                       DAUX31  0610
C                                    -1                                 DAUX31  0620
C     C13(K2-1,N) = B11(K2-1,K2)M    (K2)A13(K2,N)                      DAUX31  0630
C                                     -1                                DAUX31  0640
C                 + B12(K2-1,K2)PHI  (K2)A23(K2,N)                      DAUX31  0650
C                                                                       DAUX31  0660
C                                   -1                                  DAUX31  0670
C     C31(N,K2-1) = B31(N,K2)M     (K2)A11(K2,K2-1)                     DAUX31  0680
C                                  -1                                   DAUX31  0690
C                 + B32(N,K2)PHI  (K2)A21(K2,K2-1)                      DAUX31  0700
C                                                                       DAUX31  0710
      MQ = NQ2S + K2 - 1                                                DAUX31  0720
      IJ = IJ+1                                                         DAUX31  0730
      IJK(MQ,NNS) = IJ                                                  DAUX31  0740
      IJK(NNS,MQ) = IJ+1                                                DAUX31  0750
      DO 15 I=1,3                                                       DAUX31  0760
      DO 15 J=1,3                                                       DAUX31  0770
      SUM = -RW(K2)*A13(I,J,2*N)                                        DAUX31  0780
      TUM = -B31(I,J,2*N)*RW(K2)                                        DAUX31  0790
      DO 14 K=1,3                                                       DAUX31  0800
      SUM = SUM + B12(I,K,2*K2-2)*RPHI(K,K2)*A23(K,J,2*N    )           DAUX31  0810
   14 TUM = TUM + B32(I,K,2*N    )*RPHI(K,K2)*B12(J,K,2*K2-2)           DAUX31  0820
      C(I,J,IJ)    = SUM                                                DAUX31  0830
   15 C(I,J,IJ+1) = TUM                                                 DAUX31  0840
      IJ = IJ+1                                                         DAUX31  0850
   16 IF (NJNT.LE.0) GO TO 30                                           DAUX31  0860
      DO 26 L=1,NJNT                                                    DAUX31  0870
      IF (IABS(JNT(L)).NE.K1) GO TO 21                                  DAUX31  0880
      IF  (ISING(K1).NE.0)  GO TO 21                                    DAUX31  0890
C                                                                       DAUX31  0900
C     FOR ANY L SUCH THAT JNT(L) = K1                                   DAUX31  0910
C                                                                       DAUX31  0920
C                                   -1                                  DAUX31  0930
C     C13(L,N) = B11(L,K1)M     (K1)A13(K1,N)                           DAUX31  0940
C                                    -1                                 DAUX31  0950
C              + B12(L,K1)PHI   (K1)A23(K1,N)                           DAUX31  0960
C                                                                       DAUX31  0970
C                                   -1                                  DAUX31  0980
C     C31(N,L) = B31(N,K1)M     (K1)A11(K1,L)                           DAUX31  0990
C                                  -1                                   DAUX31  1000
```

```
      SUBROUTINE DAUX31                                                      DAUX31  0010
C                                                          REV 19 09/05/78  DAUX31  0020
C     CALLED BY SUBROUTINE DAUX TO COMPUTE                                 DAUX31  0030
C                                                                          DAUX31  0040
C                      -1                        -1                      DAUX31  0050
C     (C13) = (B11)(M)  (A13) + (B12)(PHI)  (A23)                          DAUX31  0060
C                                                                          DAUX31  0070
C                      -1                        -1                      DAUX31  0080
C     (C31) = (B31)(M)  (A11) + (B32)(PHI)  (A21)                          DAUX31  0090
C                                                                          DAUX31  0100
      IMPLICIT REAL*8 (A-H,O-Z)                                            DAUX31  0110
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,              DAUX31  0120
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)           DAUX31  0130
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(3,60),HA(3,60),HB(3,60),    DAUX31  0140
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),        DAUX31  0150
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)       DAUX31  0160
      COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60),   DAUX31  0170
     *               F(3,30),TQ(3,30),WJ(30)                               DAUX31  0180
      COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),      DAUX31  0190
     *               HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12),   DAUX31  0200
     *               RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),                 DAUX31  0210
     *               KQ1(12),KQ2(12),KQTYPE(12)                            DAUX31  0220
      COMMON/TEMPVS/ C(3,3,400),RHS(3,54),IJK(54,54),IJ,NQ2S               DAUX31  0230
      CALL ELTIME(1,17)                                                    DAUX31  0240
      DO 30 N=1,NQ                                                         DAUX31  0250
      IF (KQTYPE(N).LT.0) GO TO 30                                         DAUX31  0260
      K1 = KQ1(N)                                                          DAUX31  0270
      K2 = KQ2(N)                                                          DAUX31  0280
      NNS = NQ2S - NQ + N                                                  DAUX31  0290
      IF (K1.LE.1) GO TO 13                                                DAUX31  0300
      IF (IABS(JNT(K1-1)).EQ.0) GO TO 13                                   DAUX31  0310
      IF  (ISING(K1).NE.0)  GO TO 13                                       DAUX31  0320
C                                                                          DAUX31  0330
C                                  -1                                      DAUX31  0340
C     C13(K1-1,N) = B11(K1-1,K1)M     (K1)A13(K1,N)                        DAUX31  0350
C                                   -1                                     DAUX31  0360
C                 + B12(K1-1,K1)PHI   (K1)A23(K1,N)                        DAUX31  0370
C                                                                          DAUX31  0380
C                                 -1                                       DAUX31  0390
C     C31(N,K1-1) = B31(N,K1)M     (K1)A11(K1,K1-1)                        DAUX31  0400
C                                  -1                                      DAUX31  0410
C                 + B32(N,K1)PHI   (K1)A21(K1,K1-1)                        DAUX31  0420
C                                                                          DAUX31  0430
      MQ = NQ2S + K1 - 1                                                   DAUX31  0440
      IJ = IJ+1                                                            DAUX31  0450
      IJK(MQ,NNS) = IJ                                                     DAUX31  0460
      IJK(NNS,MQ) = IJ+1                                                   DAUX31  0470
      DO 12 I=1,3                                                          DAUX31  0480
      DO 12 J=1,3                                                          DAUX31  0490
      SUM = -RW(K1)*A13(I,J,2*N-1)                                         DAUX31  0500
```

```
      CALL DOT31 (D(1,I,N),HB(1,2*M-1),BN)                                  DAUX22  0510
      DO 53 J=1,3                                                           DAUX22  0520
      DO 53 I=1,3                                                           DAUX22  0530
   53 HH(I,J) = AN*BN(I)*BN(J)                                              DAUX22  0540
   64 DO 67 I=1,3                                                           DAUX22  0550
      RHS(I,MJNT) = -V2(I,M)                                                DAUX22  0560
      DO 66 J=1,3                                                           DAUX22  0570
      RHS(I,MJNT) = RHS(I,MJNT) + A22(J,I,2*M-1)*U2(J,N  )                  DAUX22  0580
     *                          - A22(J,I,2*M  )*U2(J,M+1)                 DAUX22  0590
      SN(I,J) = 0.0                                                         DAUX22  0600
      IF (TEST) GO TO 66                                                    DAUX22  0610
      DO 65 K=1,3                                                           DAUX22  0620
   65 SN(I,J) = SN(I,J) + A22(K,I,2*M-1) * RPHI(K,N  ) * A22(K,J,2*M-1)     DAUX22  0630
     *                  + A22(K,I,2*M  ) * RPHI(K,M+1) * A22(K,J,2*M  )     DAUX22  0640
   66 C(I,J,IJ) = SN(I,J) + HH(I,J)                                         DAUX22  0650
   67 IF (TEST) C(I,I,IJ) = AN                                              DAUX22  0660
      IF  (ISING(N).NE.0)  GO TO 90                                         DAUX22  0670
      IF (N.EQ.1) GO TO 80                                                  DAUX22  0680
      IF (IPIN(N-1).EQ.0 .OR. IPIN(N-1).GE.2) GO TO 80                      DAUX22  0690
      N1JNT = NQSJNT + N -1                                                 DAUX22  0700
      IJ = IJ+1                                                             DAUX22  0710
      IJK(MJNT,N1JNT) = IJ                                                  DAUX22  0720
      IJK(N1JNT,MJNT) = IJ+1                                                DAUX22  0730
      DO 77 I=1,3                                                           DAUX22  0740
      DO 77 J=1,3                                                           DAUX22  0750
      SN(I,J) = 0.0                                                         DAUX22  0760
      DO 76 K=1,3                                                           DAUX22  0770
   76 SN(I,J) = SN(I,J) + A22(K,I,2*M-1) * RPHI(K,N  ) * A22(K,J,2*N-2)     DAUX22  0780
      C(I,J,IJ)   = -SN(I,J)                                                DAUX22  0790
   77 C(J,I,IJ+1) = -SN(I,J)                                                DAUX22  0800
      IJ = IJ+1                                                             DAUX22  0810
   80 IF (M.EQ.NJNT) GO TO 90                                               DAUX22  0820
      M1 = M+1                                                              DAUX22  0830
      DO  88 L=M1,NJNT                                                      DAUX22  0840
      IF (IABS(JNT(L)).NE.N) GO TO 88                                       DAUX22  0850
      IF (IPIN(L).EQ.0 .OR. IPIN(L).GE.2) GO TO 88                          DAUX22  0860
      LJNT = NQSJNT + L                                                     DAUX22  0870
      IJ = IJ+1                                                             DAUX22  0880
      IJK(MJNT,LJNT) = IJ                                                   DAUX22  0890
      IJK(LJNT,MJNT) = IJ+1                                                 DAUX22  0900
      DO 87 I=1,3                                                           DAUX22  0910
      DO 87 J=1,3                                                           DAUX22  0920
      SN(I,J) = 0.0                                                         DAUX22  0930
      DO 86 K=1,3                                                           DAUX22  0940
   86 SN(I,J) = SN(I,J) + A22(K,I,2*M-1) * RPHI(K,N  ) * A22(K,J,2*L-1)     DAUX22  0950
      C(I,J,IJ)   = SN(I,J)                                                 DAUX22  0960
   87 C(J,I,IJ+1) = SN(I,J)                                                 DAUX22  0970
      IJ = IJ+1                                                             DAUX22  0980
   88 CONTINUE                                                              DAUX22  0990
   90 CONTINUE                                                              DAUX22  1000
      CALL ELTIME(2,16)                                                     DAUX22  1010
      RETURN                                                                DAUX22  1020
      END                                                                   DAUX22  1030
```

```
      SUBROUTINE DAUX22                                                      DAUX22  0010
C                                                          REV 19 09/05/78  DAUX22  0020
C     CALLED BY SUBROUTINE DAUX TO COMPUTE                                 DAUX22  0030
C                                                                          DAUX22  0040
C                                                                          DAUX22  0050
C                            -1                                            DAUX22  0060
C         (C22) = (B22)(PHI)  (A22) - (B24)                                DAUX22  0070
C                                                                          DAUX22  0080
C                            -1                                            DAUX22  0090
C         (R2)  = (B22)(PHI)  (U2)  - (V2)                                 DAUX22  0100
C                                                                          DAUX22  0110
      IMPLICIT REAL*8(A-H,O-Z)                                             DAUX22  0120
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,              DAUX22  0130
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)           DAUX22  0140
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),   DAUX22  0150
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)          DAUX22  0160
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(3,60),HA(3,60),HB(3,60),    DAUX22  0170
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),        DAUX22  0180
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)       DAUX22  0190
      COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60),   DAUX22  0200
     *               F(3,30),TQ(3,30),WJ(30)                               DAUX22  0210
      COMMON/CEULER/ IEULER(30),HIR(3,3,30),ANG(3,30),ANGD(3,30),          DAUX22  0220
     *               FE(3,30),TQE(3,30),CONST(3,30)                        DAUX22  0230
      COMMON/TEMPVS/ C(3,3,400),RHS(3,54),IJK(54,54),IJ,NQ2S               DAUX22  0240
     *              ,DN(3,3),DM(3,3),SN(3,3),SM(3,3),HH(3,3),BN(3)         DAUX22  0250
      LOGICAL TEST                                                         DAUX22  0260
      CALL ELTIME(1,16)                                                    DAUX22  0270
      NQSJNT = NQ2S + NJNT                                                 DAUX22  0280
      DO 90 M=1,NJNT                                                       DAUX22  0290
      MJNT = NQSJNT + M                                                    DAUX22  0300
      DO 60 I=1,3                                                          DAUX22  0310
   60 RHS(I,MJNT) = V2(I,M)                                                DAUX22  0320
      N = IABS(JNT(M))                                                     DAUX22  0330
      IF (N.EQ.0) GO TO 90                                                 DAUX22  0340
      IF (IPIN(M).EQ.0 .OR. IPIN(M).GE.2) GO TO 90                         DAUX22  0350
      IJ = IJ+1                                                            DAUX22  0360
      IJK(MJNT,MJNT) = IJ                                                  DAUX22  0370
      DO 61 J=1,3                                                          DAUX22  0380
      DO 61 I=1,3                                                          DAUX22  0390
   61 HH(I,J) = 0.0                                                        DAUX22  0400
      LGO = IPIN(M)+5                                                      DAUX22  0410
      TEST = .FALSE.                                                       DAUX22  0420
      GO TO (62,64,64,64,64,63,64,64,64),LGO                               DAUX22  0430
   62 IF (IEULER(M).GE.7) GO TO 64                                         DAUX22  0440
      TEST = IEULER(M).LT.4                                                DAUX22  0450
   63 AN = 0.0                                                             DAUX22  0460
      DO 51 J=1,3                                                          DAUX22  0470
   51 AN = AN + HB(J,2*M-1)**2 * RPHI(J,N  )                               DAUX22  0480
     *        + HB(J,2*M  )**2 * RPHI(J,M+1)                              DAUX22  0490
      IF (TEST) GO TO 64                                                   DAUX22  0500
```

```
      SN(I,J) = 0.0                                                                  DAUX12  0510
      DO 41 K=1,3                                                                    DAUX12  0520
   41 SN(I,J) = SN(I,J) + B12(I,K,2*M-1) * RPHI(K,N  ) * A22(K,J,2*N-2)              DAUX12  0530
      C(I,J,IJ  ) = -SN(I,J)                                                         DAUX12  0540
   42 C(J,I,IJ+1) = -SN(I,J)                                                         DAUX12  0550
      IJ = IJ+1                                                                      DAUX12  0560
   43 DO 49 L=N,NJNT                                                                 DAUX12  0570
      IF (L.EQ.M) GO TO 49                                                           DAUX12  0580
      IF (IABS(JNT(L)).NE.N ) GO TO 49                                               DAUX12  0590
      IF (IPIN(L).EQ.0 .OR. IPIN(L).GE.2) GO TO 49                                   DAUX12  0600
      MJNT = NQSJNT + L                                                              DAUX12  0610
      IJ = IJ+1                                                                      DAUX12  0620
      IJK(MQ,MJNT) = IJ                                                              DAUX12  0630
      IJK(MJNT,MQ) = IJ+1                                                            DAUX12  0640
      DO 48 I=1,3                                                                    DAUX12  0650
      DO 48 J=1,3                                                                    DAUX12  0660
      SN(I,J) = 0.0                                                                  DAUX12  0670
      DO 47 K=1,3                                                                    DAUX12  0680
   47 SN(I,J) = SN(I,J) + B12(I,K,2*M-1) * RPHI(K,N  ) * A22(K,J,2*L-1)              DAUX12  0690
      C(I,J,IJ  ) = SN(I,J)                                                          DAUX12  0700
   48 C(J,I,IJ+1) = SN(I,J)                                                          DAUX12  0710
      IJ = IJ +1                                                                     DAUX12  0720
   49 CONTINUE                                                                       DAUX12  0730
   50 IF  (M.EQ.NJNT)  GO TO 60                                                      DAUX12  0740
      IF  (ISING(M+1).NE.0)  GO TO 60                                                DAUX12  0750
      M1 = M+1                                                                       DAUX12  0760
      DO 59 L=M1,NJNT                                                                DAUX12  0770
      IF (IABS(JNT(L)).NE.M1) GO TO 59                                               DAUX12  0780
      IF (IPIN(L).EQ.0 .OR. IPIN(L).GE.2) GO TO 59                                   DAUX12  0790
      MJNT = NQSJNT + L                                                              DAUX12  0800
      IJ = IJ+1                                                                      DAUX12  0810
      IJK(MQ,MJNT) = IJ                                                              DAUX12  0820
      IJK(MJNT,MQ) = IJ+1                                                            DAUX12  0830
      DO 58 I=1,3                                                                    DAUX12  0840
      DO 58 J=1,3                                                                    DAUX12  0850
      SM(I,J) = 0.0                                                                  DAUX12  0860
      DO 57 K=1,3                                                                    DAUX12  0870
   57 SM(I,J) = SM(I,J) + B12(I,K,2*M  ) * RPHI(K,M+1) * A22(K,J,2*L-1)              DAUX12  0880
      C(I,J,IJ  ) = SM(I,J)                                                          DAUX12  0890
   58 C(J,I,IJ+1) = SM(I,J)                                                          DAUX12  0900
      IJ = IJ +1                                                                     DAUX12  0910
   59 CONTINUE                                                                       DAUX12  0920
   60 CONTINUE                                                                       DAUX12  0930
      CALL ELTIME(2,15)                                                              DAUX12  0940
      RETURN                                                                         DAUX12  0950
      END                                                                            DAUX12  0960
```

```
      SUBROUTINE DAUX12                                                      DAUX12  0010
C                                                          REV 19 09/05/78  DAUX12  0020
C     CALLED BY SUBROUTINE DAUX TO COMPUTE                                 DAUX12  0030
C                                                                          DAUX12  0040
C                            -1                                            DAUX12  0050
C         (C12) = (B12)(PHI)  (A22)                                        DAUX12  0060
C                                                                          DAUX12  0070
C                      T                                                   DAUX12  0080
C         (C21) = (C12)                                                    DAUX12  0090
C                                                                          DAUX12  0100
      IMPLICIT REAL*8(A-H,O-Z)                                             DAUX12  0110
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,              DAUX12  0120
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)           DAUX12  0130
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(3,60),HA(3,60),HB(3,60),    DAUX12  0140
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),        DAUX12  0150
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)       DAUX12  0160
      COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60),   DAUX12  0170
     *               F(3,30),TQ(3,30),WJ(30)                               DAUX12  0180
      COMMON/TEMPVS/ C(3,3,400),RHS(3,54),IJK(54,54),IJ,NQ2S               DAUX12  0190
     *              ,DN(3,3),DM(3,3),SN(3,3),SM(3,3),HH(3,3),BN(3)         DAUX12  0200
      CALL ELTIME(1,15)                                                    DAUX12  0210
      NQSJNT = NQ2S + NJNT                                                 DAUX12  0220
      DO 60 M=1,NJNT                                                       DAUX12  0230
      N = IABS(JNT(M))                                                     DAUX12  0240
      IF (N.EQ.0) GO TO 60                                                 DAUX12  0250
      MQ = NQ2S + M                                                        DAUX12  0260
      IF (IPIN(M).EQ.0  .OR. IPIN(M).GE.2) GO TO 37                        DAUX12  0270
      MJNT = NQSJNT + M                                                    DAUX12  0280
      IJ = IJ+1                                                            DAUX12  0290
      IJK(MQ,MJNT) = IJ                                                    DAUX12  0300
      IJK(MJNT,MQ) = IJ+1                                                  DAUX12  0310
      DO 36 I=1,3                                                          DAUX12  0320
      DO 36 J=1,3                                                          DAUX12  0330
      SN(I,J) = 0.0                                                        DAUX12  0340
      SM(I,J) = 0.0                                                        DAUX12  0350
      DO 35 K=1,3                                                          DAUX12  0360
      SN(I,J) = SN(I,J) + B12(I,K,2*M-1) * RPHI(K,N  ) * A22(K,J,2*M-1)    DAUX12  0370
   35 SM(I,J) = SM(I,J) + B12(I,K,2*M  ) * RPHI(K,M+1) * A22(K,J,2*M  )    DAUX12  0380
      C(I,J,IJ  ) = SN(I,J) - SM(I,J)                                      DAUX12  0390
   36 C(J,I,IJ+1) = C(I,J,IJ)                                              DAUX12  0400
      IJ = IJ+1                                                            DAUX12  0410
   37 IF  (ISING(N).NE.0)  GO TO 50                                        DAUX12  0420
      IF  (N.EQ.1)  GO TO 43                                               DAUX12  0430
      IF (IPIN(N-1).EQ.0 .OR. IPIN(N-1).GE.2) GO TO 43                     DAUX12  0440
      MJNT = NQSJNT + N-1                                                  DAUX12  0450
      IJ = IJ+1                                                            DAUX12  0460
      IJK(MQ,MJNT) = IJ                                                    DAUX12  0470
      IJK(MJNT,MQ) = IJ+1                                                  DAUX12  0480
      DO 42 I=1,3                                                          DAUX12  0490
      DO 42 J=1,3                                                          DAUX12  0500
```

```
      T1 = U1(I,N) - U1(I,M+1) - V1(I,M)                                    DAUX11  0510
      DO 15 J = 1,3                                                         DAUX11  0520
      T1 = T1 + B12(I,J,2*M-1)*U2(J,N) + B12(I,J,2*M)*U2(J,M+1)             DAUX11  0530
      IF (J.LT.I) GO TO 15                                                  DAUX11  0540
      T2 = 0.0                                                              DAUX11  0550
      IF (J.EQ.I) T2 = RW(N) + RW(M+1)                                      DAUX11  0560
      DO 14 K=1,3                                                           DAUX11  0570
   14 T2 = T2 + B12(I,K,2*M-1)*RPHI(K,N  )*B12(J,K,2*M-1)                   DAUX11  0580
     *       + B12(I,K,2*M  )*RPHI(K,M+1)*B12(J,K,2*M  )                    DAUX11  0590
      C(I,J,IJ) = T2                                                        DAUX11  0600
      C(J,I,IJ) = T2                                                        DAUX11  0610
   15 RHS(I,MQ) = T1                                                        DAUX11  0620
      IF  (ISING(N).NE.0)  GO TO 30                                         DAUX11  0630
      L = 0                                                                 DAUX11  0640
      IF (N.GT.1) L = IABS(JNT(N-1))                                        DAUX11  0650
      IF (L.EQ.0) GO TO 18                                                  DAUX11  0660
C                                                                           DAUX11  0670
C     IF (N > 1) AND (L = JNT(N-1) > 0)                                     DAUX11  0680
C                                                                           DAUX11  0690
C         SET  C11(M,N-1) =  -RW(N) + B12(M,N)PHI(N)'A21(N,N-1)             DAUX11  0700
C                                                                           DAUX11  0710
C                                    T                                      DAUX11  0720
C         AND  C11(N-I,M) = C(M,N-1)                                        DAUX11  0730
C                                                                           DAUX11  0740
      KJNT = NQ2S + N -1                                                    DAUX11  0750
      IJ = IJ+1                                                             DAUX11  0760
      IJK(MQ,KJNT) = IJ                                                     DAUX11  0770
      IJK(KJNT,MQ) = IJ+1                                                   DAUX11  0780
      DO 17 I=1,3                                                           DAUX11  0790
      DO 17 J=1,3                                                           DAUX11  0800
      C(I,J,IJ) = 0.0                                                       DAUX11  0810
      IF (I.EQ.J) C(I,J,IJ) = -RW(N)                                        DAUX11  0820
      DO 16 K=1,3                                                           DAUX11  0830
   16 C(I,J,IJ) = C(I,J,IJ) + B12(I,K,2*M-1)*RPHI(K,N)*B12(J,K,2*N-2)       DAUX11  0840
   17 C(J,I,IJ+1) = C(I,J,IJ)                                               DAUX11  0850
      IJ = IJ+1                                                             DAUX11  0860
   18 IF (M.EQ.NJNT) GO TO 30                                               DAUX11  0870
      M1 = M+1                                                              DAUX11  0880
      DO 21 L=M1,NJNT                                                       DAUX11  0890
      IF (IABS(JNT(L)).NE.N) GO TO 21                                       DAUX11  0900
C                                                                           DAUX11  0910
C     IF (L > M) AND (JNT(L) = N)                                           DAUX11  0920
C                                                                           DAUX11  0930
C         SET  C11(M,L) = RW(N) + B12(M,N)PHI(N)'A21(N,L)                   DAUX11  0940
C                                                                           DAUX11  0950
C                                    T                                      DAUX11  0960
C         AND  C11(L,M) = C11(M,L)                                          DAUX11  0970
C                                                                           DAUX11  0980
      KJNT = NQ2S + L                                                       DAUX11  0990
      IJ = IJ+1                                                             DAUX11  1000
      IJK(MQ,KJNT) = IJ                                                     DAUX11  1010
      IJK(KJNT,MQ) = IJ+1                                                   DAUX11  1020
      DO 20 I=1,3                                                           DAUX11  1030
      DO 20 J=1,3                                                           DAUX11  1040
      C(I,J,IJ) = 0.0                                                       DAUX11  1050
      IF (I.EQ.J) C(I,J,IJ) = RW(N)                                         DAUX11  1060
      DO 19 K=1,3                                                           DAUX11  1070
   19 C(I,J,IJ) = C(I,J,IJ) + B12(I,K,2*M-1)*RPHI(K,N)*B12(J,K,2*L-1)       DAUX11  1080
   20 C(J,I,IJ+1) = C(I,J,IJ)                                               DAUX11  1090
      IJ = IJ+1                                                             DAUX11  1100
   21 CONTINUE                                                              DAUX11  1110
   30 CONTINUE                                                              DAUX11  1120
      CALL ELTIME(2,14)                                                     DAUX11  1130
      RETURN                                                                DAUX11  1140
      END                                                                   DAUX11  1150
```

```
      SUBROUTINE DAUX11                                                      DAUX11  0010
C                                                          REV 19 09/05/78  DAUX11  0020
C     CALLED BY SUBROUTINE DAUX TO COMPUTE                                 DAUX11  0030
C                                                                          DAUX11  0040
C                          -1                      -1                      DAUX11  0050
C         (C11) = (B11)(M)  (A11) + (B12)(PHI)  (A21)                      DAUX11  0060
C                                                                          DAUX11  0070
C                         -1                      -1                       DAUX11  0080
C         (R1) = (B11)(M)  (U1) + (B12)(PHI)  (U2) - (V1)                  DAUX11  0090
C                                                                          DAUX11  0100
      IMPLICIT REAL*8(A-H,O-Z)                                             DAUX11  0110
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,              DAUX11  0120
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)           DAUX11  0130
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),   DAUX11  0140
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)          DAUX11  0150
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(3,60),HA(3,60),HB(3,60),    DAUX11  0160
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),        DAUX11  0170
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)       DAUX11  0180
      COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60),   DAUX11  0190
     *               F(3,30),TQ(3,30),WJ(30)                               DAUX11  0200
      COMMON/TEMPVS/ C(3,3,400),RHS(3,54),IJK(54,54),IJ,NQ2S               DAUX11  0210
     *              ,DN(3,3),DM(3,3),SN(3,3),SM(3,3),HH(3,3),BN(3)         DAUX11  0220
      CALL ELTIME(1,14)                                                    DAUX11  0230
      DO 30 M=1,NJNT                                                       DAUX11  0240
      N = IABS(JNT(M))                                                     DAUX11  0250
      MQ = NQ2S + M                                                        DAUX11  0260
      IJ = IJ+1                                                            DAUX11  0270
      IJK(MQ,MQ) = IJ                                                      DAUX11  0280
      IF (N.GT.0) GO TO 13                                                 DAUX11  0290
C                                                                          DAUX11  0300
C     IF (N < 1)  SET  C11(M,M) = I                                        DAUX11  0310
C                                                                          DAUX11  0320
C                 AND  RHS(M)   = V1(M)                                    DAUX11  0330
C                                                                          DAUX11  0340
      DO 12 I=1,3                                                          DAUX11  0350
      DO 11 J=1,3                                                          DAUX11  0360
   11 C(I,J,IJ) = 0.0                                                      DAUX11  0370
      C(I,I,IJ) = 1.0                                                      DAUX11  0380
   12 RHS(I,MQ) = V1(I,M)                                                  DAUX11  0390
      IJK(MQ,MQ) = -IJ                                                     DAUX11  0400
      GO TO 30                                                             DAUX11  0410
C                                                                          DAUX11  0420
C     IF (N > 0)  SET  RHS(M)   = U1(N) - U1(M+1) - V1(M)                  DAUX11  0430
C                              + B12(M,N)U2(N) + B12(M,M+1)U2(M+1)         DAUX11  0440
C                                                                          DAUX11  0450
C                 AND  C11(M,N) = RW(N) + RW(M+1)                          DAUX11  0460
C                              + B12(M,N  )PHI(N  )'A21(N  ,M)             DAUX11  0470
C                              + B12(M,M+1)PHI(M+1)'A21(M+1,M)             DAUX11  0480
C                                                                          DAUX11  0490
   13 DO 15 I=1,3                                                          DAUX11  0500
```

```
      SEGLA(I,N  ) = SEGLA(I,N  ) - RW(N  )*F(I,M)                          DAUX    2010
      SEGLA(I,M+1) = SEGLA(I,M+1) + RW(M+1)*F(I,M)                          DAUX    2020
      DO 72 J=1,3                                                           DAUX    2030
      WMEGD(I,N)   = WMEGD(I,N  ) - B12(J,I,2*M-1)*RPHI(I,N  )*F(J,M)       DAUX    2040
   72 WMEGD(I,M+1) = WMEGD(I,M+1) - B12(J,I,2*M  )*RPHI(I,M+1)*F(J,M)       DAUX    2050
C                                                                           DAUX    2060
C     ELIMINATE TQ                                                          DAUX    2070
C                                                                           DAUX    2080
   73 IF (IPIN(M).EQ.0 .OR. IPIN(M).GE.2) GO TO 75                          DAUX    2090
      L = NQ2S + NJNT + M                                                   DAUX    2100
      DO 74 I=1,3                                                           DAUX    2110
      DO 74 J=1,3                                                           DAUX    2120
      WMEGD(I,N  ) = WMEGD(I,N  ) - A22(I,J,2*M-1)*RPHI(I,N  )*RHS(J,L)     DAUX    2130
   74 WMEGD(I,M+1) = WMEGD(I,M+1) + A22(I,J,2*M  )*RPHI(I,M+1)*RHS(J,L)     DAUX    2140
   75 CONTINUE                                                              DAUX    2150
   80 IF (NQ.EQ.0) GO TO 83                                                 DAUX    2160
C                                                                           DAUX    2170
C     ELIMINATE QQ                                                          DAUX    2180
C                                                                           DAUX    2190
      DO 82 K=1,NQ                                                          DAUX    2200
      IF (KQTYPE(K).LT.0) GO TO 82                                          DAUX    2210
      N = KQ1(K)                                                            DAUX    2220
      M = KQ2(K)                                                            DAUX    2230
      DO 81 I=1,3                                                           DAUX    2240
      DO 81 J=1,3                                                           DAUX    2250
      SEGLA(I,N) = SEGLA(I,N) - A13(I,J,2*K-1)*RW(N)    *QQ(J,K)            DAUX    2260
      SEGLA(I,M) = SEGLA(I,M) - A13(I,J,2*K  )*RW(M)    *QQ(J,K)            DAUX    2270
      WMEGD(I,N) = WMEGD(I,N) - A23(I,J,2*K-1)*RPHI(I,N)*QQ(J,K)            DAUX    2280
   81 WMEGD(I,M) = WMEGD(I,M) - A23(I,J,2*K  )*RPHI(I,M)*QQ(J,K)            DAUX    2290
   82 CONTINUE                                                              DAUX    2300
   83 IF (NFLX.EQ.0) GO TO 90                                               DAUX    2310
C                                                                           DAUX    2320
C     ELIMINATE V4 (TORQUES FOR FLEXIBLE SEGMENTS)                          DAUX    2330
C                                                                           DAUX    2340
      DO 84 N=1,NFLX                                                        DAUX    2350
      N1 = NFLEX(1,N)                                                       DAUX    2360
      N2 = NFLEX(2,N)                                                       DAUX    2370
      N3 = NFLEX(3,N)                                                       DAUX    2380
      DO 84 I=1,3                                                           DAUX    2390
      DO 84 J=1,3                                                           DAUX    2400
      WMEGD(I,N1) = WMEGD(I,N1) - B42(J,I,3*N-2)*RPHI(I,N1)*V4(J,N)         DAUX    2410
      WMEGD(I,N2) = WMEGD(I,N2) - B42(J,I,3*N-1)*RPHI(I,N2)*V4(J,N)         DAUX    2420
   84 WMEGD(I,N3) = WMEGD(I,N3) - B42(J,I,3*N  )*RPHI(I,N3)*V4(J,N)         DAUX    2430
   90 DO 91 J=1,NGRND                                                       DAUX    2440
      DO 91 I=1,3                                                           DAUX    2450
      IF (DABS(WMEGD(I,J)).LE.EPS12) WMEGD(I,J) = 0.0                       DAUX    2460
   91 IF (DABS(SEGLA(I,J)).LE.EPS12) SEGLA(I,J) = 0.0                       DAUX    2470
C                                                                           DAUX    2480
C     OPTIONAL OUTPUT OF FUNCTIONS AND DERIVATIVES.                         DAUX    2490
C                                                                           DAUX    2500
      IF (NPRT(9).NE.0) CALL PRINT(6H DAUX )                                DAUX    2510
C                                                                           DAUX    2520
      CALL ELTIME(2,9)                                                      DAUX    2530
      RETURN                                                                DAUX    2540
      END                                                                   DAUX    2550
```

```
      DO 51 I=1,NJNT                                                    DAUX    1510
      NJ = NQ2S + I                                                     DAUX    1520
      NI = NJ+NJNT                                                      DAUX    1530
      DO 51 K=1,3                                                       DAUX    1540
      IF (DABS(RHS(K,NJ)).LT.EPS12) RHS(K,NJ) = 0.0                     DAUX    1550
      IF (DABS(RHS(K,NI)).LT.EPS12) RHS(K,NI) = 0.0                     DAUX    1560
      TQ(K,I) = TQ(K,I) - RHS(K,NI)                                     DAUX    1570
   51 F(K,I) = RHS(K,NJ)                                                DAUX    1580
   49 IF (NQ.EQ.0) GO TO 53                                             DAUX    1590
      DO 52 I=1,NQ                                                      DAUX    1600
      J = 2*NS + NFLX + I                                               DAUX    1610
      DO 52 K=1,3                                                       DAUX    1620
      IF (KQTYPE(I).LT.0)           RHS(K,J) = 0.0                      DAUX    1630
      IF (DABS(RHS(K,J)).LT.EPS12) RHS(K,J) = 0.0                       DAUX    1640
   52 QQ(K,I) = RHS(K,J)                                                DAUX    1650
   53 IF (NFLX.EQ.0) GO TO 70                                           DAUX    1660
      DO 54 I=1,NFLX                                                    DAUX    1670
      J = 2*NS + I                                                      DAUX    1680
      DO 54 K=1,3                                                       DAUX    1690
      IF (DABS(RHS(K,J)).LT.EPS12) RHS(K,J) = 0.0                       DAUX    1700
   54 V4(K,I) = RHS(K,J)                                                DAUX    1710
C                                                                       DAUX    1720
C     BACKUP SOLUTION FOR SEGLA AND WMEGD.                              DAUX    1730
C                                                                       DAUX    1740
   70 DO  71  J=1,NGRND                                                 DAUX    1750
      DO  71  I=1,3                                                     DAUX    1760
      SEGLA(I,J) = U1(I,J)                                              DAUX    1770
   71 WMEGD(I,J) = U2(I,J)                                              DAUX    1780
      IF (NS.EQ.0) GO TO 79                                             DAUX    1790
C                                                                       DAUX    1800
C     SET UP SEGLA & WMEGD FOR SINGULAR SEGMENTS.                       DAUX    1810
C                                                                       DAUX    1820
      IS = 0                                                            DAUX    1830
      DO 78 J=1,NGRND                                                   DAUX    1840
      IF (ISING(J).LE.0) GO TO 78                                       DAUX    1850
      IS = IS+2                                                         DAUX    1860
      DO 77 I=1,3                                                       DAUX    1870
      IF (DABS(RHS(I,IS-1)).LT.EPS12) RHS(I,IS-1) = 0.0                 DAUX    1880
      SEGLA(I,J) = SEGLA(I,J) + RHS(I,IS-1)                             DAUX    1890
      IF (DABS(RHS(I,IS  )).LT.EPS12) RHS(I,IS  ) = 0.0                 DAUX    1900
   77 WMEGD(I,J) = WMEGD(I,J) + RHS(I,IS)                               DAUX    1910
   78 CONTINUE                                                          DAUX    1920
   79 IF (NJNT.EQ.0) GO TO 80                                           DAUX    1930
C                                                                       DAUX    1940
C     ELIMINATE F                                                       DAUX    1950
C                                                                       DAUX    1960
      DO 75 M=1,NJNT                                                    DAUX    1970
      N = IABS(JNT(M))                                                  DAUX    1980
      IF (N.EQ.0) GO TO 73                                              DAUX    1990
      DO 72 I=1,3                                                       DAUX    2000
```

```
   11 FORMAT('0NS=',I6,',NFLX=',I6,',NQ=',I6,',NJNT=',I6,' AND NJ2='I6/     DAUX     1010
     *' THE VALUE OF NJ2 EXCEEDS THE ARRAY SIZES FOR RHS AND IJK IN SUBR    DAUX     1020
     *OUTINE DAUX. PROGRAM TERMINATED.')                                    DAUX     1030
      IF  (NJ2.GT.54)  STOP 34                                              DAUX     1040
      MJ2 = NJ2                                                             DAUX     1050
      DO 10 I=1,NJ2                                                         DAUX     1060
      DO 10 J=1,NJ2                                                         DAUX     1070
   10 IJK(I,J) = 0                                                          DAUX     1080
      IJ = 0                                                                DAUX     1090
C                                                                           DAUX     1100
C     ELMINATE SEGLA AND WMEGD FROM SYSTEM OF EQUATIONS.                    DAUX     1110
C                                                                           DAUX     1120
      IF (NS.GT.0)   CALL DAUX55                                            DAUX     1130
      IF (NJNT.EQ.0) GO TO 12                                               DAUX     1140
      IF (NFLX.GT.0) CALL DAUX44                                            DAUX     1150
      CALL DAUX11                                                           DAUX     1160
      CALL DAUX12                                                           DAUX     1170
      CALL DAUX22                                                           DAUX     1180
   12 IF (NQ.LE.0) GO TO 15                                                 DAUX     1190
      IF (NJNT.EQ.0) GO TO 13                                               DAUX     1200
      CALL DAUX31                                                           DAUX     1210
      CALL DAUX32                                                           DAUX     1220
   13 CALL DAUX33                                                           DAUX     1230
      DO 14 I=1,NQ                                                          DAUX     1240
   14 IF (KQTYPE(I).GE.4) MJ2 = -NJ2                                        DAUX     1250
   15 IF (NPRT(8).EQ.0) GO TO 28                                            DAUX     1260
   21 WRITE (6,22) (J,J=1,NJ2)                                              DAUX     1270
   22 FORMAT('1 DAUX PRINT OF IJK MATRIX'//6X,40I3)                         DAUX     1280
      DO 23 I=1,NJ2                                                         DAUX     1290
   23 WRITE (6,24) I,(IJK(I,J),J=1,NJ2)                                     DAUX     1300
   24 FORMAT(I3,3X,40I3)                                                    DAUX     1310
      WRITE (6,29)                                                          DAUX     1320
   29 FORMAT('0 DAUX PRINT OF RHS ARRAY'//)                                 DAUX     1330
      DO 30 K=1,NJ2                                                         DAUX     1340
   30 WRITE (6,27) K,(RHS(I,K),I=1,3)                                       DAUX     1350
      WRITE (6,25)                                                          DAUX     1360
   25 FORMAT('1 DAUX PRINT OF C ARRAY ELEMENTS'//)                          DAUX     1370
      DO 26 K=1,IJ                                                          DAUX     1380
   26 WRITE (6,27) K,((C(I,J,K),J=1,3),I=1,3)                               DAUX     1390
   27 FORMAT(I6,9G14.7)                                                     DAUX     1400
   28 IF (NPRT(8).EQ.-2) GO TO 31                                           DAUX     1410
C                                                                           DAUX     1420
C     SOLVE SYSTEM OF EQUATIONS FOR F,TQ,QQ & V4.                           DAUX     1430
C                                                                           DAUX     1440
      CALL FSMSOL (C,RHS,IJK,MJ2,IJ,54,400)                                 DAUX     1450
      IF (NPRT(8).EQ. 2) NPRT(8) = -2                                       DAUX     1460
      IF (NPRT(8).EQ.-2) GO TO 21                                           DAUX     1470
   31 IF (NPRT(8).EQ.-2) NPRT(8) = 0                                        DAUX     1480
      EPS12 = EPS(12)                                                       DAUX     1490
      IF (NJNT.EQ.0) GO TO 49                                               DAUX     1500
```

```
      IF  (ISING(J))  1,3,5                                                  DAUX    0510
    1 DO  2  I=1,3                                                           DAUX    0520
      U1(I,J) = SEGLA(I,J)                                                   DAUX    0530
    2 U2(I,J) = WMEGD(I,J)                                                   DAUX    0540
      GO TO 5                                                                DAUX    0550
    3 DO  4  I=1,3                                                           DAUX    0560
      U1(I,J) = U1(I,J)*RW(J) + GRAVTY(I)                                    DAUX    0570
    4 U2(I,J) = U2(I,J)*RPHI(I,J)                                            DAUX    0580
    5 CONTINUE                                                               DAUX    0590
C                                                                            DAUX    0600
C     SET UP BODY SEGMENT SYMMETRY                                           DAUX    0610
C         NSYM(J) = 0     3D MOTION                                          DAUX    0620
C         NSYM(J) = J     CENTRAL SEGMENT 2D MOTION, NO LATERAL MOTION       DAUX    0630
C         NSYM(J) = K     SEGMENT J SYMMETRIC TO SEGMENT K, ALL MOTION       DAUX    0640
C                         IN THE X-Z PLANE, NO LATERAL MOTION                DAUX    0650
C         NSYM(J) = -K    SEGMENT J MIRROR SYMMETRIC TO SEGMENT K, EQUAL     DAUX    0660
C                         BUT OPPOSITE LATERAL MOTION PERMITTED              DAUX    0670
C                                                                            DAUX    0680
      DO  20  J=1,NGRND                                                      DAUX    0690
      IF (NSYM(J).EQ.0) GO TO 20                                             DAUX    0700
      IF (NSYM(J).EQ.J) GO TO 19                                             DAUX    0710
      K = IABS(NSYM(J))                                                      DAUX    0720
      IF (K.LT.J) GO TO 16                                                   DAUX    0730
      U1(1,J) = 0.5*(U1(1,J) + U1(1,K))                                      DAUX    0740
      U1(3,J) = 0.5*(U1(3,J) + U1(3,K))                                      DAUX    0750
      U2(2,J) = 0.5*(U2(2,J) + U2(2,K))                                      DAUX    0760
      GO TO 17                                                               DAUX    0770
   16 U1(1,J) = U1(1,K)                                                      DAUX    0780
      U1(3,J) = U1(3,K)                                                      DAUX    0790
      U2(2,J) = U2(2,K)                                                      DAUX    0800
   17 IF (NSYM(J).GT.0) GO TO 19                                             DAUX    0810
      IF (K.LT.J) GO TO 18                                                   DAUX    0820
      U1(2,J) = 0.5*(U1(2,J) - U1(2,K))                                      DAUX    0830
      U2(1,J) = 0.5*(U2(1,J) - U2(1,K))                                      DAUX    0840
      U2(3,J) = 0.5*(U2(3,J) - U2(3,K))                                      DAUX    0850
      GO TO 20                                                               DAUX    0860
   18 U1(2,J) = -U1(2,K)                                                     DAUX    0870
      U2(1,J) = -U2(1,K)                                                     DAUX    0880
      U2(3,J) = -U2(3,K)                                                     DAUX    0890
      GO TO 20                                                               DAUX    0900
   19 U1(2,J) = 0.0                                                          DAUX    0910
      U2(1,J) = 0.0                                                          DAUX    0920
      U2(3,J) = 0.0                                                          DAUX    0930
   20 CONTINUE                                                               DAUX    0940
C                                                                            DAUX    0950
C     INITIALIZE IJK ARRAY AND IJ COUNTER TO ZERO.                           DAUX    0960
C                                                                            DAUX    0970
    8 NQ2S = 2*NS + NFLX + NQ                                                DAUX    0980
      NJ2  = NQ2S + 2*NJNT                                                   DAUX    0990
      IF (NJ2.GT.54) WRITE (6,11) NS,NFLX,NQ,NJNT,NJ2                        DAUX    1000
```

```
      SUBROUTINE DAUX(I1)                                                    DAUX     0010
C                                                          REV 20 05/18/80    DAUX     0020
C     COMPUTES DERIVATIVES FOR INTEGRATOR ROUTINE BY                          DAUX     0030
C      (1) SET UP INITIAL VALUES FOR ARRAY OF SYSTEM EQUATIONS.               DAUX     0040
C      (2) MODIFY ARRAYS BY CONSTRAINTS.                                      DAUX     0050
C      (3) SOLVE SYSTEM OF EQUATION FOR F,TQ,QQ AND V4.                       DAUX     0060
C      (4) EVALUATE DERIVATIVES SEGLA AND WMEGD.                              DAUX     0070
C                                                                             DAUX     0080
      IMPLICIT REAL*8(A-H,O-Z)                                               DAUX     0090
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,                DAUX     0100
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)             DAUX     0110
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),     DAUX     0120
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)            DAUX     0130
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(3,60),HA(3,60),HB(3,60),      DAUX     0140
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),          DAUX     0150
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)         DAUX     0160
      COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60),     DAUX     0170
     *               F(3,30),TQ(3,30),WJ(30)                                 DAUX     0180
      COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),        DAUX     0190
     *               HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12),     DAUX     0200
     *               RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),                   DAUX     0210
     *               KQ1(12),KQ2(12),KQTYPE(12)                              DAUX     0220
      COMMON/FLXBLE/ HF(4,12,8),B42(3,3,24),V4(3,8),NFLEX(3,8)               DAUX     0230
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                              DAUX     0240
     *               UNITL,UNITM,UNITT,GRAVTY(3)                             DAUX     0250
C                                                                            DAUX     0260
C     NOTE: DAUX SHARES /TEMPVS/ WITH DAUX11,12,22,31,32 &33.                DAUX     0270
C                                                                            DAUX     0280
      COMMON/TEMPVS/ C(3,3,400),RHS(3,54),IJK(54,54),IJ,NQ2S                 DAUX     0290
      CALL ELTIME(1,9)                                                       DAUX     0300
C                                                                            DAUX     0310
C     IF I1#0, U1 AND U2 HAVE BEEN SET UP BY CALLING ROUTINE.                DAUX     0320
C                                                                            DAUX     0330
      IF (I1.NE.0) GO TO  8                                                  DAUX     0340
C                                                                            DAUX     0350
C     SET UP INITIAL VALUES OF A & B ARRAYS AND U & V VECTORS.               DAUX     0360
C     MODIFY U1 & U2 ARRAYS BY CONTACT AND JOINT FORCES.                     DAUX     0370
C                                                                            DAUX     0380
      CALL SETUP1                                                            DAUX     0390
      CALL VEHPOS                                                            DAUX     0400
      CALL CHAIN                                                             DAUX     0410
      CALL CONTCT                                                            DAUX     0420
      CALL VISPR(0,0)                                                        DAUX     0430
      CALL EJOINT(0,0)                                                       DAUX     0440
      CALL SETUP2                                                            DAUX     0450
      IF (NFLX.GT.0) CALL FLXSEG                                             DAUX     0460
C                                                                            DAUX     0470
C     MODIFY U1,U2 AND ADD G TO U1.                                          DAUX     0480
C                                                                            DAUX     0490
      DO  5  J=1,NGRND                                                       DAUX     0500
```

```
      SUBROUTINE CROSS(A,B,C)                                          CROSS   0010
C                                                   REV 03 05/31/73    CROSS   0020
C     COMPUTES VECTOR CROSS PRODUCT  C = A X B.                        CROSS   0030
C                                                                      CROSS   0040
C       ARGUMENTS                                                      CROSS   0050
C         A,B,C: VECTORS OF LENGTH 3 WHERE C=AXB.                      CROSS   0060
C                                                                      CROSS   0070
      IMPLICIT REAL*8 (A-H,O-Z)                                        CROSS   0080
      DIMENSION A(3),B(3),C(3)                                         CROSS   0090
       C(1) = A(2)*B(3) - A(3)*B(2)                                    CROSS   0100
       C(2) = A(3)*B(1) - A(1)*B(3)                                    CROSS   0110
       C(3) = A(1)*B(2) - A(2)*B(1)                                    CROSS   0120
      RETURN                                                           CROSS   0130
      END                                                              CROSS   0140
```

```
      M2 = MBLT(2,I,J)                                                  CONTCT  0510
      M3 = MBLT(3,I,J)                                                  CONTCT  0520
      NT = NTBLT(I,J)                                                   CONTCT  0530
      JT = NTAB(NT)                                                     CONTCT  0540
      TAB(JT) = 0.0                                                     CONTCT  0550
      NF = NTAB(NT+5)                                                   CONTCT  0560
      IF (NF.NE.0) JT = NTAB(NT+6)                                      CONTCT  0570
      IF (NF.NE.0) TAB(JT) = 0.0                                        CONTCT  0580
   29 CALL BELTRT(M2,M3,M1,J,NT)                                        CONTCT  0590
   30 CONTINUE                                                          CONTCT  0600
C                                                                       CONTCT  0610
C     CALL SEGSEG ROUTINE FOR EACH ALLOWED SEGMENT-SEGMENT CONTACT.     CONTCT  0620
C                                                                       CONTCT  0630
   41 DO 50 J=1,NSEG                                                    CONTCT  0640
      IF(MNSEG(J).EQ.0) GO TO 50                                        CONTCT  0650
      KSEG = MNSEG(J)                                                   CONTCT  0660
      DO 49 I=1,KSEG                                                    CONTCT  0670
      NSSF = NSSF+1                                                     CONTCT  0680
      M1 = MSEG(1,I,J)                                                  CONTCT  0690
      M2 = MSEG(2,I,J)                                                  CONTCT  0700
      M3 = MSEG(3,I,J)                                                  CONTCT  0710
      NT = NTSEG(I,J)                                                   CONTCT  0720
      JT = NTAB(NT)                                                     CONTCT  0730
      TAB(JT) = 0.0                                                     CONTCT  0740
   49 CALL SEGSEG(J,M1,M2,M3,NT)                                        CONTCT  0750
   50 CONTINUE                                                          CONTCT  0760
C                                                                       CONTCT  0770
C     CALL AIRBAG ROUTINE FOR ALLOWED BAG-SEGMENT CONTACTS, IF ANY.     CONTCT  0780
C                                                                       CONTCT  0790
      IF (NBAG.NE.0) CALL AIRBAG                                        CONTCT  0800
C                                                                       CONTCT  0810
C     CALL WINDY ROUTINE FOR WIND FORCES ON EACH SEGMENT.               CONTCT  0820
C                                                                       CONTCT  0830
      DO 60 J=1,NSEG                                                    CONTCT  0840
      IF (MWSEG(1,J).EQ.0) GO TO 60                                     CONTCT  0850
      M1 = MWSEG(2,J)                                                   CONTCT  0860
      M2 = MWSEG(3,J)                                                   CONTCT  0870
      M3 = MWSEG(4,J)                                                   CONTCT  0880
      NT = MWSEG(5,J)                                                   CONTCT  0890
      CALL WINDY (J,M1,M2,M3,NT)                                        CONTCT  0900
   60 CONTINUE                                                          CONTCT  0910
C                                                                       CONTCT  0920
C     CALL WINDY FOR FORCE FUNCE FUNCTION CALCULATIONS.                 CONTCT  0930
C                                                                       CONTCT  0940
      NFORCE = NFVSEG(6)                                                CONTCT  0950
      IF  (NFORCE.GT.0)  CALL WINDY (-1,M1,M2,M3,NT)                    CONTCT  0960
C                                                                       CONTCT  0970
C     CALL HBELT ROUTINE FOR EACH HARNESS-BELT SYSTEM.                  CONTCT  0980
C                                                                       CONTCT  0990
      IF  (NHRNSS.LE.0)  GO TO 80                                       CONTCT  1000
      J1 = 1                                                            CONTCT  1010
      KNL0 = 0                                                          CONTCT  1020
      DO  70  I=1,NHRNSS                                                CONTCT  1030
      IF  (NBLTPH(I).LE.0)  GO TO 70                                    CONTCT  1040
      J2 = J1 + NBLTPH(I) - 1                                           CONTCT  1050
      CALL HBELT (J1,J2,KNL0,0)                                         CONTCT  1060
      J1 = J2+1                                                         CONTCT  1070
   70 CONTINUE                                                          CONTCT  1080
C                                                                       CONTCT  1090
C     CALL SPDAMP FOR SPRING DAMPER FORCES, IF ANY                      CONTCT  1100
C                                                                       CONTCT  1110
   80 IF  (NSD.NE.0)  CALL SPDAMP                                       CONTCT  1120
      CALL ELTIME (2,12)                                                CONTCT  1130
      RETURN                                                            CONTCT  1140
      END                                                               CONTCT  1150
```

```
      SUBROUTINE CONTCT                                                      CONTCT  0010
C                                                          REV 20 05/18/80  CONTCT  0020
C     CONTROLS THE CALLING OF SUBROUTINES REQUIRED TO COMPUTE THOSE        CONTCT  0030
C     EXTERNAL FORCES AND TORQUES ACTING ON THE BODY SEGMENTS.             CONTCT  0040
C                                                                          CONTCT  0050
      IMPLICIT REAL*8 (A-H,O-Z)                                            CONTCT  0060
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,              CONTCT  0070
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)           CONTCT  0080
      COMMON/JBARTZ/ MNPL(   30),MNBLT(    8),MNSEG(   30),MNBAG(    6),  CONTCT  0090
     *               MPL(3,5,30),MBLT(3,5,8),MSEG(3,5,30),MBAG(3,10,6),    CONTCT  0100
     *               NTPL( 5,30),NTBLT( 5,8),NTSEG( 5,30)                  CONTCT  0110
      COMMON/FORCES/ PSF(7,30),BSF(4,20),SSF(10,20),BAGSF(3,20),           CONTCT  0120
     *               PRJNT(7,30),NPANEL(5),NPSF,NBSF,NSSF,NBGSF            CONTCT  0130
      COMMON/TABLES/ MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(500),TAB(2600)   CONTCT  0140
      COMMON/HRNESS/ BAR(15,100),BB(100),BBDOT(100),PLOSS(2,100),          CONTCT  0150
     *               XLONG(20),HTIME(2),IBAR(5,100),NL(2,100),             CONTCT  0160
     *               NPTSPB(20),NPTPLY(20),NTHRNS(20),NBLTPH(5)            CONTCT  0170
      COMMON/WINDFR/ WTIME(30),QFU(3,5),QFV(3,5),                          CONTCT  0180
     *               IWIND(30),MWSEG(5,30),NFVSEG(6),NFVNT(5)              CONTCT  0190
      CALL ELTIME(1,12)                                                    CONTCT  0200
      NPSF = 0                                                             CONTCT  0210
      NBSF = 0                                                             CONTCT  0220
      NSSF = 0                                                             CONTCT  0230
      IF (NPL.LE.0) GO TO 21                                               CONTCT  0240
C                                                                          CONTCT  0250
C     CALL PLELP ROUTINE FOR EACH ALLOWED PLANE-SEGMENT CONTACT.           CONTCT  0260
C                                                                          CONTCT  0270
      DO 20 J=1,NPL                                                        CONTCT  0280
      IF(MNPL(J).EQ.0) GO TO 20                                            CONTCT  0290
      KPL = MNPL(J)                                                        CONTCT  0300
      DO 19 I=1,KPL                                                        CONTCT  0310
      NPSF = NPSF+1                                                        CONTCT  0320
      M1 = MPL(1,I,J)                                                      CONTCT  0330
      M2 = MPL(2,I,J)                                                      CONTCT  0340
      M3 = MPL(3,I,J)                                                      CONTCT  0350
      NT = NTPL(I,J)                                                       CONTCT  0360
      JT = NTAB(NT)                                                        CONTCT  0370
      TAB(JT) = 0.0                                                        CONTCT  0380
   19 CALL PLELP(M2,M3,M1,J,NT)                                            CONTCT  0390
   20 CONTINUE                                                             CONTCT  0400
   21 IF(NBLT.LE.0) GO TO 41                                               CONTCT  0410
C                                                                          CONTCT  0420
C     CALL BELTRT ROUTINE FOR EACH ALLOWED BELT-SEGMENT CONTACT.           CONTCT  0430
C                                                                          CONTCT  0440
      DO 30 J=1,NBLT                                                       CONTCT  0450
      IF(MNBLT(J).EQ.0) GO TO 30                                           CONTCT  0460
      KBLT = MNBLT(J)                                                      CONTCT  0470
      DO 29 I=1,KBLT                                                       CONTCT  0480
      NBSF = NBSF+1                                                        CONTCT  0490
      M1 = MBLT(1,I,J)                                                     CONTCT  0500
```

```
      SUBROUTINE CMPUTE (K,M,FT)                                        CMPUTE  0010
C                                                     REV 19 09/18/79   CMPUTE  0020
      IMPLICIT REAL*8 (A-H,O-Z)                                         CMPUTE  0030
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           CMPUTE  0040
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)        CMPUTE  0050
      COMMON/CDINT/ UU(4),GH(3,4),                                      CMPUTE  0060
     *              E(3,240), F(5,240),GG(5,240),Y(5,240),U(5,240),     CMPUTE  0070
     *              H,HPRINT,HS,TPRINT,TSTART,ICNT,IDBL,IFLAG           CMPUTE  0080
      COMMON/COMAIN/ VAR(240),DER(240),DT,H0,HMAX,HMIN,RSTIME,          CMPUTE  0090
     *               ISTEP,NSTEPS,NDINT,NEQ,IRSIN,IRSOUT                CMPUTE  0100
      TIME = TSTART + FT                                                CMPUTE  0110
      CALL DZP (NEQ,VAR,GG,E,FT,M)                                      CMPUTE  0120
      IF  (NPRT(26).EQ.2)  CALL OUTPUT(0)                               CMPUTE  0130
      CALL PDAUX (VAR,DER,NEQ,K)                                        CMPUTE  0140
      IF  (NPRT(26).EQ.2)  CALL OUTPUT(1)                               CMPUTE  0150
      RETURN                                                            CMPUTE  0160
      END                                                               CMPUTE  0170
```

```
   19 FORMAT(7X,'FUNCTION IS CONSTANT',F12.6)                                CINPUT  0510
      GO TO 11                                                               CINPUT  0520
C                                                                            CINPUT  0530
C       5TH ORDER POLYNOMIAL ...  1ST FUNCTION                               CINPUT  0540
C       INPUT CARD E.3                                                       CINPUT  0550
C                                                                            CINPUT  0560
   20 J2 = J1+5                                                              CINPUT  0570
      READ(5,15)(TAB(J),J = J1,J2)                                           CINPUT  0580
      WRITE(6,21)  (TAB(J),J = J1,J2)                                        CINPUT  0590
   21 FORMAT(7X,'FIRST PART OF FUNCTION - 5TH DEGREE POLYNOMIAL'//           CINPUT  0600
     *          8X,'A0',13X,'A1',13X,'A2',13X,'A3',13X,'A4',13X,'A5',13X/    CINPUT  0610
     *          6F15.6//)                                                    CINPUT  0620
      J1 = J2+1                                                              CINPUT  0630
      GO TO 25                                                               CINPUT  0640
C                                                                            CINPUT  0650
C          TABLE LOAD ...  1ST FUNCTION                                      CINPUT  0660
C          INPUT CARDS E.4.A-E.4.N                                           CINPUT  0670
C                                                                            CINPUT  0680
   22 READ(5,23)  NPI                                                        CINPUT  0690
   23 FORMAT (12I6)                                                          CINPUT  0700
      TAB(J1) = NPI                                                          CINPUT  0710
      J1 = J1+1                                                              CINPUT  0720
      J2 = J1+2*NPI-1                                                        CINPUT  0730
      READ(5,15)(TAB(J),J = J1,J2)                                           CINPUT  0740
      WRITE (6,24)  NPI, (TAB(J)   ,J = J1, J2)                              CINPUT  0750
   24 FORMAT(7X,'FIRST PART OF FUNCTION - ',I4,' TABULAR POINTS'//           CINPUT  0760
     *          8X,'D',16X,'F(D)'  /(F15.6,F15.4))                           CINPUT  0770
      J1 = J2+1                                                              CINPUT  0780
C                                                                            CINPUT  0790
C        CHECK FOR SECOND FUNCTION                                           CINPUT  0800
C                                                                            CINPUT  0810
   25 IF(D2)  28,11,26                                                       CINPUT  0820
C                                                                            CINPUT  0830
C       SECOND FUNCTION ...  5TH ORDER POLYNOMIAL                            CINPUT  0840
C       INPUT CARD E.3                                                       CINPUT  0850
C                                                                            CINPUT  0860
   26 J2 = J1+5                                                              CINPUT  0870
      READ(5,15)(TAB(J),J = J1,J2)                                           CINPUT  0880
      WRITE (6,27)  (TAB(J),J = J1,J2)                                       CINPUT  0890
   27 FORMAT(7X,'SECOND PART OF FUNCTION - 5TH DEGREE POLYNOMIAL'//          CINPUT  0900
     *          8X,'B0',13X,'B1',13X,'B2',13X,'B3',13X,'B4',13X,'B5',13X/    CINPUT  0910
     *          6F15.6//)                                                    CINPUT  0920
      J1 = J2+1                                                              CINPUT  0930
      GO TO 11                                                               CINPUT  0940
C                                                                            CINPUT  0950
C       SECOND FUNCTION ...  TABLE LOAD                                      CINPUT  0960
C       INPUT CARDS E.4.A-E.4.N                                              CINPUT  0970
C                                                                            CINPUT  0980
   28 READ(5,23)  NPI                                                        CINPUT  0990
      TAB(J1) = NPI                                                          CINPUT  1000
      J1 = J1+1                                                              CINPUT  1010
      J2 = J1+2*NPI-1                                                        CINPUT  1020
      READ(5,15)(TAB(J),J = J1,J2)                                           CINPUT  1030
      WRITE(6,29) NPI, (TAB(J), J = J1,J2)                                   CINPUT  1040
   29 FORMAT(7X,'SECOND PART OF FUNCTION - ',I4,' TABULAR POINTS'//          CINPUT  1050
     *          8X,'D',16X,'F(D)'  /(F15.6,F15.4))                           CINPUT  1060
      J1 = J2+1                                                              CINPUT  1070
      GO TO 11                                                               CINPUT  1080
   30 MXTB1 = J1-1                                                           CINPUT  1090
      CALL KINPUT                                                            CINPUT  1100
      CALL FINPUT                                                            CINPUT  1110
      CALL HINPUT                                                            CINPUT  1120
      RETURN                                                                 CINPUT  1130
      END                                                                    CINPUT  1140
```

```
      SUBROUTINE CINPUT                                                 CINPUT  0010
C                                                          REV 19 08/05/78  CINPUT  0020
C     INPUT CARDS E.1 - E.4 FOR THE FORCE-DEFLECTION, INERTIAL SPIKE,   CINPUT  0030
C     R FACTOR, G FACTOR AND FRICTION COEFFICIENT FUNCTION DEFINITIONS  CINPUT  0040
C                                                                       CINPUT  0050
      IMPLICIT REAL*8(A-H,O-Z)                                          CINPUT  0060
      COMMON/TABLES/ MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(500),TAB(2600) CINPUT 0070
      COMMON/TEMPVS/JTITLE(5,51),NF(5),NS(3),KTITLE(31)                 CINPUT  0080
      REAL JTITLE,KTITLE                                                CINPUT  0090
C                                                                       CINPUT  0100
      IS = 0                                                            CINPUT  0110
      DO 10 I = 1,50                                                    CINPUT  0120
   10 NTI(I) = 0                                                        CINPUT  0130
      J1 = 1                                                            CINPUT  0140
C                                                                       CINPUT  0150
C     INPUT CARD E.1 - FUNCTION NO. AND TITLE, IF NO. > 50 SKIP OUT.    CINPUT  0160
C                                                                       CINPUT  0170
   11 READ (5,12) I,(KTITLE(J  ),J = 1,5)                               CINPUT  0180
   12 FORMAT (I4,4X,5A4)                                                CINPUT  0190
      IF (I.GT.50) GO TO 30                                             CINPUT  0200
      DO 13 J = 1,5                                                     CINPUT  0210
   13 JTITLE(J,I) = KTITLE(J)                                           CINPUT  0220
C                                                                       CINPUT  0230
C     HAS FUNCTION NO. BEEN ALREADY USED?                               CINPUT  0240
C                                                                       CINPUT  0250
      IF  (NTI(I).NE.0)  WRITE(6,14) I                                  CINPUT  0260
   14 FORMAT('0 FUNCTION NO.',I4,' HAS ALREADY BEEN INPUTTED AND WILL BE CINPUT 0270
     *REPLACED BY NEXT FUNCTION')                                       CINPUT  0280
      NTI(I) = J1                                                       CINPUT  0290
      J2 = J1+4                                                         CINPUT  0300
C                                                                       CINPUT  0310
C     INPUT CARD E.2                                                    CINPUT  0320
C                                                                       CINPUT  0330
      READ (5,15) (TAB(J),J = J1,J2)                                    CINPUT  0340
   15 FORMAT (6F12.0)                                                   CINPUT  0350
      IS = 1-IS                                                         CINPUT  0360
      IF (IS.EQ.0) WRITE (6,16)                                         CINPUT  0370
   16 FORMAT(/////)                                                     CINPUT  0380
      WRITE (6,17) IS,I,(JTITLE(J,I),J=1,5),I,NTI(I),(TAB(J),J=J1,J2)   CINPUT  0390
   17 FORMAT(I1,'FUNCTION NO.',I4,4X,5A4,20X,'NTI(',I2,') =',I5,45X     CINPUT  0400
     * 'CARDS E'//10X,'D0',13X,'D1',13X,'D2',13X,'D3',13X,'D4'/5F15.4//) CINPUT 0410
      D0 = TAB(J1)                                                      CINPUT  0420
      D1 = TAB(J1+1)                                                    CINPUT  0430
      D2 = TAB(J1+2)                                                    CINPUT  0440
      J1 = J2+1                                                         CINPUT  0450
      IF (D1)  22,18,20                                                 CINPUT  0460
C                                                                       CINPUT  0470
C     FUNCTION IS CONSTANT  D2 FOR ALL D.                               CINPUT  0480
C                                                                       CINPUT  0490
   18 WRITE (6,19)  D2                                                  CINPUT  0500
```

```
      SUBROUTINE CHAIN                                                  CHAIN   0010
C                                                          REV 19 09/05/78  CHAIN   0020
C     COMPUTES THE LINEAR POSITION AND VELOCITY IN INERTIAL REFERENCE    CHAIN   0030
C     OF BODY SEGMENTS FROM THOSE OF THE REFERENCE SEGMENTS              CHAIN   0040
C     (I.E., SEGMENT NO. 1 AND EACH SEGMENT J FOR WHICH JNT(J)=0).       CHAIN   0050
C                                                                        CHAIN   0060
      IMPLICIT REAL*8(A-H,O-Z)                                           CHAIN   0070
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,            CHAIN   0080
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)         CHAIN   0090
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30), CHAIN   0100
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)        CHAIN   0110
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(3,60),HA(3,60),HB(3,60),  CHAIN   0120
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),      CHAIN   0130
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)     CHAIN   0140
      COMMON/TEMPVS/ T1(3),T2(3),T3(3),T4(3)                             CHAIN   0150
      CALL ELTIME(1,11)                                                  CHAIN   0160
C                                                                        CHAIN   0170
C     COMPUTE SEGMENT POSITIONS BY                                       CHAIN   0180
C      PM = PN + DN'*RNJ - DM'*RMJ                                       CHAIN   0190
C                                                                        CHAIN   0200
C     COMPUTE SEGMENT VELOCITIES BY                                      CHAIN   0210
C      VM = VN + DN'*WN X RNJ - DM'*WM X RMJ                             CHAIN   0220
C                                                                        CHAIN   0230
      IF (NSEG.EQ.1) GO TO 71                                            CHAIN   0240
      DO 70 J=2,NSEG                                                     CHAIN   0250
      I = IABS(JNT(J-1))                                                 CHAIN   0260
      IF (I.EQ.0) GO TO 70                                               CHAIN   0270
      IF  (ISING(J).LT.0)  GO TO 70                                      CHAIN   0280
      K = 2*J-3                                                          CHAIN   0290
      CALL CROSS (WMEG(1,I),SR(1,K),T1)                                  CHAIN   0300
      CALL DOT31 (D(1,1,I),T1,T3)                                        CHAIN   0310
      CALL CROSS (WMEG(1,J),SR(1,K+1),T2)                                CHAIN   0320
      CALL DOT31 (D(1,1,J),T2,T4)                                        CHAIN   0330
      CALL DOT31 (D(1,1,I),SR(1,K),T1)                                   CHAIN   0340
      CALL DOT31 (D(1,1,J),SR(1,K+1),T2)                                 CHAIN   0350
      DO 60  L=1,3                                                       CHAIN   0360
      SEGLP(L,J) = SEGLP(L,I) + T1(L) - T2(L)                            CHAIN   0370
   60 SEGLV(L,J) = SEGLV(L,I) + T3(L) - T4(L)                            CHAIN   0380
   70 CONTINUE                                                           CHAIN   0390
C                                                                        CHAIN   0400
C     OPTIONAL OUTPUT                                                    CHAIN   0410
C                                                                        CHAIN   0420
   71 IF  (NPRT(20).NE.0)  WRITE (6,90)  TIME                            CHAIN   0430
     *                       ,((SEGLP(I,J),I=1,3),J=1,NSEG)              CHAIN   0440
     *                       ,((SEGLV(I,J),I=1,3),J=1,NSEG)              CHAIN   0450
   90 FORMAT('0 LINEAR POSITIONS AND VELOCITIES OF BODY SEGMENTS FROM CH  CHAIN   0460
     *AIN FOR TIME =',F12.6/(9F13.5))                                    CHAIN   0470
      CALL ELTIME(2,11)                                                  CHAIN   0480
      RETURN                                                             CHAIN   0490
      END                                                                CHAIN   0500
```

```
      SUBROUTINE CFACTT(A,B,D)                                          CFACTT  0010
C                                                        REV 03 05/31/73  CFACTT  0020
C        GIVEN 3X3 MATRIX A                                              CFACTT  0030
C        COMPUTE B TRANSPOSE OF COFACTORS (SIGNED MINORS)                CFACTT  0040
C        AND D THE VALUE OF THE DETERMINANT OF A.                        CFACTT  0050
C        INVERSE OF A IS B(J,K)/D.                                       CFACTT  0060
C                                                                        CFACTT  0070
      IMPLICIT REAL*8 (A-H,O-Z)                                          CFACTT  0080
      DIMENSION A(3,3),B(3,3)                                            CFACTT  0090
       M = 4                                                             CFACTT  0100
       L = 2                                                             CFACTT  0110
       N = 3                                                             CFACTT  0120
       D = 0.0                                                           CFACTT  0130
      DO 20 J=1,3                                                        CFACTT  0140
       B(J,J) = A(L,L)*A(N,N)-A(L,N)*A(N,L)                              CFACTT  0150
       IF (J.EQ.3) GO TO 20                                              CFACTT  0160
       L = N                                                             CFACTT  0170
       N = J                                                             CFACTT  0180
       KK = J+1                                                          CFACTT  0190
      DO 15 K=KK,3                                                       CFACTT  0200
       M = M-1                                                           CFACTT  0210
       B(K,J) = A(K,M)*A(M,J)-A(K,J)*A(M,M)                              CFACTT  0220
   15  B(J,K) = A(J,M)*A(M,K)-A(J,K)*A(M,M)                              CFACTT  0230
   20 D = D+A(1,J)*B(J,1)                                                CFACTT  0240
       RETURN                                                           CFACTT  0250
      END                                                               CFACTT  0260
```

```
      SUBROUTINE BLKDTA                                                 BLKDTA  0010
C                                                          REV 19 08/05/78  BLKDTA  0020
CC    THIS SUBROUTINE REPLACES THE BLOCK DATA SUBPROGRAM OF PREVIOUS    BLKDTA  0030
C     VERSIONS OF CVS-III TO INITIALIZE COMMON/CNSNTS/ IN A MANNER      BLKDTA  0040
C     THAT IS INDEPENDENT OF THE COMPUTER SYSTEM BEING UTILIZED.        BLKDTA  0050
C                                                                       BLKDTA  0060
      IMPLICIT REAL*8 (A-H,O-Z)                                         BLKDTA  0070
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         BLKDTA  0080
     *               UNITL,UNITM,UNITT,GRAVTY(3)                        BLKDTA  0090
      COMMON/TEMPVS/ ZERO,ONE,THREE,TEN,ONE80                           BLKDTA  0100
      DATA  UM/8H LBS    / , UT/8H SEC    / , UL/8H IN     /            BLKDTA  0110
      ZERO  = 0.0                                                       BLKDTA  0120
      ONE   = 1.0                                                       BLKDTA  0130
      UNITM = UM                                                        BLKDTA  0140
      UNITT = UT                                                        BLKDTA  0150
      UNITL = UL                                                        BLKDTA  0160
      G     = 386.088D0                                                 BLKDTA  0170
      GRAVTY(1) = ZERO                                                  BLKDTA  0180
      GRAVTY(2) = ZERO                                                  BLKDTA  0190
      GRAVTY(3) = G                                                     BLKDTA  0200
      THREE = 3.0                                                       BLKDTA  0210
      TEN   = 10.0                                                      BLKDTA  0220
      ONE80 = 180.0                                                     BLKDTA  0230
      PI = DATAN2(ZERO,-ONE)                                            BLKDTA  0240
      RADIAN = PI/ONE80                                                 BLKDTA  0250
      THIRD  = ONE/THREE                                                BLKDTA  0260
      EPS(1) = ONE/TEN                                                  BLKDTA  0270
      DO  10  I=2,24                                                    BLKDTA  0280
   10 EPS(I) = EPS(I-1)/TEN                                             BLKDTA  0290
      RETURN                                                            BLKDTA  0300
      END                                                               BLKDTA  0310
```

```
C                                                                       BINPUT  3010
   65 NS = 0                                                            BINPUT  3020
      DO  68  I=1,NSEG                                                  BINPUT  3030
      ISING(I) = 0                                                      BINPUT  3040
      RW(I) = 0.0                                                       BINPUT  3050
      IF (W(I).EQ.0.0) ISING(I) = 1                                     BINPUT  3060
      DO 66 K=1,3                                                       BINPUT  3070
      IF (PHI(K,I).EQ.0.0) ISING(I) = 1                                 BINPUT  3080
   66 RPHI(K,I) = 0.0                                                   BINPUT  3090
      IF (ISING(I).EQ.1) NS = NS+1                                      BINPUT  3100
      IF (ISING(I).EQ.1) GO TO 68                                       BINPUT  3110
      RW(I) = G/W(I)                                                    BINPUT  3120
      DO 67 K=1,3                                                       BINPUT  3130
   67 RPHI(K,I) = 1.0/PHI(K,I)                                          BINPUT  3140
   68 CONTINUE                                                          BINPUT  3150
C                                                                       BINPUT  3160
C     SET UP ELLIPSOID MATRIX AND INVERSE (ASSUME YAW,PITCH,ROLL = 0)   BINPUT  3170
C     FOR 1ST NSEG ELLIPSOIDS IN BD(7-15) AND BD(16-24).                BINPUT  3180
C                                                                       BINPUT  3190
   69 DO  71  J=1,NSEG                                                  BINPUT  3200
      DO 70 I=7,24                                                      BINPUT  3210
   70 BD(I,J) = 0.0                                                     BINPUT  3220
      DO 71 I=1,3                                                       BINPUT  3230
      BD(4*I+3,J) = 1.0/BD(I,J)**2                                      BINPUT  3240
   71 BD(4*I+12,J) = BD(I,J)**2                                         BINPUT  3250
      RETURN                                                            BINPUT  3260
      END                                                               BINPUT  3270
```

```
      READ (5,54) NFX,(KNT(J),J=1,NFX)                                  BINPUT  2510
   54 FORMAT(18I4)                                                      BINPUT  2520
      IF (NFX.NE.NFLX) WRITE (6,55) NFX,NFLX                            BINPUT  2530
   55 FORMAT('0INPUT ERROR ON CARD B.7.A, NFX =',I4, ' BUT NFLX =',I4/  BINPUT  2540
     *       ' AS COMPUTED FROM CARDS B.3. PROGRAM TERMINATED.')        BINPUT  2550
      IF  (NFX.NE.NFLX)  STOP 4                                         BINPUT  2560
      WRITE (6,56)                                                      BINPUT  2570
   56 FORMAT('1',120X,'CARDS B.7')                                      BINPUT  2580
      DO 60 JJ=1,NFX                                                    BINPUT  2590
      DO  57  K=1,NFLX                                                  BINPUT  2600
      IF (KNT(JJ).EQ.NFLEX(2,K)) GO TO 59                               BINPUT  2610
   57 CONTINUE                                                          BINPUT  2620
      WRITE (6,58) KNT(JJ)                                              BINPUT  2630
   58 FORMAT('0INPUT ERROR ON CARD B.7.J, SEGMENT NO.',I4,' IS NOT AN IN BINPUT  2640
     *TERIOR SEGMENT OF A FLEXIBLE ELEMENT FROM DATA ON CARDS B.3.'/     BINPUT  2650
     *       ' PROGRAM TERMINATED.')                                    BINPUT  2660
      STOP 5                                                            BINPUT  2670
C                                                                       BINPUT  2680
C     CARDS B.7.J  HF ARRAY FOR SEGMENT KNT(JJ)                         BINPUT  2690
C                                                                       BINPUT  2700
   59 READ (5,29) ((HF(I,J,K),J=1,12),I=1,4)                            BINPUT  2710
   60 WRITE (6,61) KNT(JJ),K,(NFLEX(I,K),I=1,3),                        BINPUT  2720
     *            ((HF(I,J,K),J=1,12),I=1,4)                          BINPUT  2730
   61 FORMAT('0 HF ARRAY FOR INTERIOR SEGMENT NO.',I4,20X,              BINPUT  2740
     *       '(NFLEX(I,',I1,'),I=1,3) =',3I6//                        BINPUT  2750
     *       (3X,4F10.4,3X,4F10.4,3X,4F10.4) )                        BINPUT  2760
   62 IF (NJNT.EQ.0) GO TO 65                                           BINPUT  2770
C                                                                       BINPUT  2780
C     CHANGE SPRING AND VISC FROM DEG TO RAD                            BINPUT  2790
C                                                                       BINPUT  2800
      DO  64  I=1,NJNT                                                  BINPUT  2810
      J1 = 3*I-2                                                        BINPUT  2820
      J2 = 3*I-1                                                        BINPUT  2830
      IF (IABS(IPIN(I)).EQ.4) J2= 3*I                                   BINPUT  2840
      DO 63 J=J1,J2                                                     BINPUT  2850
      SPRING(1,J) = SPRING(1,J)/RADIAN                                  BINPUT  2860
      SPRING(2,J) = SPRING(2,J)/RADIAN**2                               BINPUT  2870
      SPRING(3,J) = SPRING(3,J)/RADIAN**3                               BINPUT  2880
      SPRING(5,J) = SPRING(5,J)*RADIAN                                  BINPUT  2890
   63 CONTINUE                                                          BINPUT  2900
      IF (IABS(IPIN(I)).NE.4) J2 = J1                                   BINPUT  2910
      DO 64 J=J1,J2                                                     BINPUT  2920
      VISC  (1,J) = VISC  (1,J)/RADIAN                                  BINPUT  2930
   64 VISC  (3,J) = VISC  (3,J)*RADIAN                                  BINPUT  2940
                                                                        BINPUT  2950
C     W ARRAY HAS BEEN SUPPLIED IN LBS. SET UP RECIPROCAL MASS (RW)     BINPUT  2960
C     AND MOMENT OF INERTIA (RPHI) ARRAYS. HOWEVER, IF W OR ANY ELEMENT BINPUT  2970
C     OF PHI IS ZERO, SEGMENT WILL BE CONSIDERED SINGULAR (ISING=1) AND BINPUT  2980
C     ALL RECIPROCALS WILL BE ZERO SO AS TO NULLIFY COMPUTATIONS IN THE BINPUT  2990
C     DAUX ROUTINES. NS IS THE NUMBER OF SINGULAR SEGMENTS.             BINPUT  3000
```

```
      J1 = 3*J-2                                                        BINPUT  2010
      J2 = 3*J-1                                                        BINPUT  2020
      J3 = 3*J                                                          BINPUT  2030
      WRITE (6,43) J,JOINT(J),((SPRING(I,JJ),I=1,5),JJ=J1,J2)           BINPUT  2040
   42 IF (IABS(IPIN(J)).EQ.4) WRITE (6,44) (SPRING(I,J3),I=1,5)         BINPUT  2050
   43 FORMAT(I3,1X,A4,2(3X,3F12.3,2F10.3))                              BINPUT  2060
   44 FORMAT(11X,3F12.3,2F10.3)                                         BINPUT  2070
C                                                                       BINPUT  2080
C     PRINT CARDS B.5.J  FOR EACH JOINT.                                BINPUT  2090
C                                                                       BINPUT  2100
   45 WRITE (6,46) (UNITL,UNITM,UNITT,I=1,2),(UNITL,UNITM,I=1,2),UNITT  BINPUT  2110
   46 FORMAT(///120X,'CARDS B.5'/                                       BINPUT  2120
     *38X,'JOINT VISCOUS CHARACTERISTICS AND LOCK-UNLOCK CONDITIONS'//   BINPUT  2130
     *14X,'VISCOUS',9X,'COULOMB',7X,'FULL FRICTION',5X,'MAX TORQUE FOR',  BINPUT  2140
     *4X,'MIN TORQUE FOR',4X,'MIN. ANG. VELOCITY',6X,'IMPULSE'/          BINPUT  2150
     *2X,'JOINT',5X,'COEFFICIENT',4X,'FRICTION COEF. ANGULAR VELOCITY',  BINPUT  2160
     *4X,'A LOCKED JOINT',4X,'UNLOCKED JOINT',4X,'FOR UNLOCKED JOINT',   BINPUT  2170
     *4X,'RESTITUTION'/                                                 BINPUT  2180
     *8X,'(',3A4,'/DEG)  (',2A4,')',6X,'(DEG/',A4,')',10X,'(',2A4,')',  BINPUT  2190
     *8X,'(',2A4,')',10X,'(RAD/',A4,')',8X,'COEFFICIENT'/ )             BINPUT  2200
      IF (NJNT.EQ.0) GO TO 50                                           BINPUT  2210
      DO  47  J=1,NJNT                                                  BINPUT  2220
      J1 = 3*J-2                                                        BINPUT  2230
      J2 = 3*J-1                                                        BINPUT  2240
      J3 = 3*J                                                          BINPUT  2250
      WRITE (6,48) J,JOINT(J),(VISC(I,J1),I=1,7)                        BINPUT  2260
   47 IF (IABS(IPIN(J)).EQ.4) WRITE (6,49) ((VISC(I,JJ),I=1,7),JJ=J2,J3) BINPUT  2270
   48 FORMAT(I3,1X,A4,F13.3,2F15.2,F22.2,F18.2,F20.2,F17.3)              BINPUT  2280
   49 FORMAT(      8X,F13.3,2F15.2,F22.2,F18.2,F20.2,F17.3)             BINPUT  2290
C                                                                       BINPUT  2300
C     PRINT CARDS  B.6.I.  FOR EACH SEGMENT.                            BINPUT  2310
C                                                                       BINPUT  2320
   50 WRITE (6,51) (UNITT,UNITL,UNITT,I=1,2)                            BINPUT  2330
   51 FORMAT('1',120X,'CARDS B.6'/                                      BINPUT  2340
     *     20X,'SEGMENT INTEGRATION CONVERGENCE TEST INPUT'//           BINPUT  2350
     *      17X,'ANGULAR VELOCITIES',  11X,'LINEAR VELOCITIES',         BINPUT  2360
     *      10X,'ANGULAR ACCELERATIONS',9X,'LINEAR ACCELERATIONS'/      BINPUT  2370
     *     21X,'(RAD/',A4,')', 18X,'(',A4,'/',A4,')',                   BINPUT  2380
     *     17X,'(RAD/',A4,'**2)', 16X,'(',A4,'/',A4,'**2)'/             BINPUT  2390
     *      ' SEGMENT', 4('       MAG.     ABS.     REL.')   /         BINPUT  2400
     *      ' NO. SYM', 4('       TEST    ERROR    ERROR')   /)        BINPUT  2410
      DO 52 I=1,NSEG                                                    BINPUT  2420
   52 WRITE (6,53) I,SEG(I),((SGTEST(J,K,I),J=1,3),K=1,4)               BINPUT  2430
   53 FORMAT(I3,1X,A4,4(F11.3,F9.3,F9.4) )                              BINPUT  2440
      IF (NFLX.EQ.0) GO TO 62                                           BINPUT  2450
C                                                                       BINPUT  2460
C     INPUT AND PRINT CARDS B.7                                         BINPUT  2470
C     CARD B.7.A  NFX: NO. OF INTERIOR SEGMENTS OF FLEXIBLE ELEMENTS.   BINPUT  2480
C                 KNT(J),J=1,NFX: THE SEGMENT NUMBERS.                  BINPUT  2490
C                                                                       BINPUT  2500
```

```
     *             2X, 'PRIN. AXIS(DEG) - SEG(J+1)'/                          BINPUT  1510
     * '  J SYM PLOT JNT PIN', 2(6X,'X',8X,'Y',8X,'Z',3X),                    BINPUT  1520
     *             2(5X,'YAW',5X,'PITCH',5X,'ROLL',1X) /)                     BINPUT  1530
      IF (NJNT.EQ.0) GO TO 40                                                 BINPUT  1540
      DO  34  J=1,NJNT                                                        BINPUT  1550
      WRITE (6,35) J,JOINT(J),JS(J),JNT(J),IPIN(J),(SR(I,2*J-1),I=1,3),       BINPUT  1560
     *             (SR(I,2*J),I=1,3),(YPR1(I,J),I=1,3),(YPR2(I,J),I=1,3)      BINPUT  1570
      IF (IABS(IPIN(J)).LT.4) GO TO 34                                        BINPUT  1580
      IEULER(J) = 8                                                           BINPUT  1590
      IF (IPIN(J).EQ.4) GO TO 34                                              BINPUT  1600
      IEULER(J) = 11 + IPIN(J)                                                BINPUT  1610
      IPIN(J) = -4                                                            BINPUT  1620
   34 CONTINUE                                                                BINPUT  1630
   35 FORMAT( I3,1X,A4,2X,A1,2X,2I3,2(1X,3F9.3),2(1X,3F9.2) )                 BINPUT  1640
C                                                                             BINPUT  1650
C     SET UP HT MATRIX FROM YPR1 & YPR2 INPUT.                                BINPUT  1660
C     HA IS 3RD COLUMN & HB IS 2ND COLUMN OF HT.                              BINPUT  1670
C                                                                             BINPUT  1680
      IF  (NPRT(23).NE.0)  WRITE (6,36)                                       BINPUT  1690
   36 FORMAT('1 HT ARRAY AS COMPUTED FROM YPR1 & YPR2 INPUT.')                BINPUT  1700
      DO  38  J=1,NJNT                                                        BINPUT  1710
      CALL DRCYPR (TMP1,YPR1(1,J),IDYPR(1,J))                                 BINPUT  1720
      CALL DRCYPR (TMP2,YPR2(1,J),IDYPR(4,J))                                 BINPUT  1730
      DO 37 I=1,3                                                             BINPUT  1740
      ANGD(I,J) = 0.0                                                         BINPUT  1750
      HA(I,2*J-1) = 0.0                                                       BINPUT  1760
      HA(I,2*J  ) = 0.0                                                       BINPUT  1770
      IF (IABS(IPIN(J)).GE.4) CONST(I,J) = YPR3(I,J)*RADIAN                   BINPUT  1780
      IF (IABS(IPIN(J)).GE.4) ANG(I,J) = ANG(I,J)*RADIAN - CONST(I,J)         BINPUT  1790
      HB(I,2*J-1) = TMP1(2,I)                                                 BINPUT  1800
      HB(I,2*J  ) = TMP2(2,I)                                                 BINPUT  1810
      DO 37 K=1,3                                                             BINPUT  1820
      HT(I,K,2*J-1) = TMP1(K,I)                                               BINPUT  1830
   37 HT(I,K,2*J  ) = TMP2(K,I)                                               BINPUT  1840
   38 IF  (NPRT(23).NE.0)  WRITE (6,39) J,JOINT(J),                           BINPUT  1850
     *    ((HT(I,K,2*J-1),K=1,3),(HT(I,K,2*J),K=1,3),I=1,3)                   BINPUT  1860
   39 FORMAT('0',I4,2X,A4,3X,3F12.6,3X,3F12.6/(14X,3F12.6,3X,3F12.6))         BINPUT  1870
C                                                                             BINPUT  1880
C     PRINT CARDS  B.4.J  FOR EACH JOINT.                                     BINPUT  1890
C                                                                             BINPUT  1900
   40 WRITE (6,41) UNITL,UNITM,UNITL,UNITM                                    BINPUT  1910
   41 FORMAT('1 JOINT TORQUE CHARACTERISTICS',90X,'CARDS B.4'//               BINPUT  1920
     *23X,'FLEXURAL SPRING CHARACTERISTICS',28X,'TORSIONAL SPRING' ,          BINPUT  1930
     *' CHARACTERISTICS'//                                                    BINPUT  1940
     *15X,'SPRING COEF. (',2A4,'/DEG**J)',6X,'ENERGY     JOINT',             BINPUT  1950
     * 7X,'SPRING COEF. (',2A4,'/DEG**J)',6X,'ENERGY     JOINT'/    '  J     BINPUT  1960
     *OINT ',2(8X,'LINEAR    QUADRATIC     CUBIC    DISSIPATION  STOP ')     BINPUT  1970
     */8X,2(8X,'(J=1)',7X,'(J=2)',7X,'(J=3)',7X,'COEF.     (DEG)')/)          BINPUT  1980
      IF (NJNT.EQ.0) GO TO 45                                                 BINPUT  1990
      DO  42  J=1,NJNT                                                        BINPUT  2000
```

```
C     INPUT CARDS  B.4.J  FOR EACH JOINT.                                    BINPUT  1010
C                                                                           BINPUT  1020
      DO  23  J=1,NJNT                                                      BINPUT  1030
      READ (5,24) (SPRING(I,3*J-2),I=1,5),(SPRING(I,3*J-1),I=1,5)           BINPUT  1040
   23 IF (IABS(IPIN(J)).GE.4) READ (5,24) (SPRING(I,3*J  ),I=1,5)           BINPUT  1050
     *                                   ,(ANG(I,J),I=1,3)                  BINPUT  1060
   24 FORMAT(2(4F6.0,F12.0))                                                BINPUT  1070
                                                                            BINPUT  1080
C     INPUT CARDS  B.5.J  FOR EACH JOINT.                                   BINPUT  1090
C                                                                           BINPUT  1100
      DO  25  J=1,NJNT                                                      BINPUT  1110
      READ (5,26) (VISC(I,3*J-2),I=1,7)                                     BINPUT  1120
      IF (IABS(IPIN(J)).LT.4) GO TO 25                                      BINPUT  1130
      READ (5,26) (VISC(I,3*J-1),I=1,7)                                     BINPUT  1140
      READ (5,26) (VISC(I,3*J  ),I=1,7)                                     BINPUT  1150
   25 CONTINUE                                                              BINPUT  1160
   26 FORMAT(5F6.0,18X,2F6.0)                                               BINPUT  1170
                                                                            BINPUT  1180
C     INPUT CARDS  B.6.I  FOR EACH SEGMENT.                                 BINPUT  1190
C                                                                           BINPUT  1200
   27 DO  28  I=1,NSEG                                                      BINPUT  1210
   28 READ (5,29) ((SGTEST(J,K,I),J=1,3),K=1,4)                             BINPUT  1220
   29 FORMAT(12F6.0)                                                        BINPUT  1230
                                                                            BINPUT  1240
C     PRINT CARD B.1                                                        BINPUT  1250
C                                                                           BINPUT  1260
      WRITE (6,30) BDYTTL,NSEG,NJNT,UNITM,UNITT,UNITL,UNITL,UNITL,UNITM     BINPUT  1270
   30 FORMAT('1 CRASH VICTIM',5X,5A4,I5,' SEGMENTS',I5,' JOINTS',55X,       BINPUT  1280
     *       'CARD  B.1'//25X,'PRINCIPAL MOMENTS OF INERTIA',               BINPUT  1290
     *       14X,'SEGMENT CONTACT ELLIPSOID',28X,'CARDS B.2'/               BINPUT  1300
     *       3X,'SEGMENT',6X,'WEIGHT',7X,'(',A4,'-',A4,'**2-',A4,')',       BINPUT  1310
     *       11X,'SEMIAXES (',A4,')',12X,'CENTER (',A4,')',                 BINPUT  1320
     *       11X,'PRINCIPAL AXES (DEG)'/                                    BINPUT  1330
     *       '  I  SYM PLOT   (',A4,')',7X,'X',8X,'Y',8X,'Z ',              BINPUT  1340
     *       2(9X,'X',7X,'Y',7X,'Z'),8X,'YAW',5X,'PITCH',5X,'ROLL'/)        BINPUT  1350
C                                                                           BINPUT  1360
C     PRINT CARDS  B.2.I  FOR EACH SEGMENT.                                 BINPUT  1370
C                                                                           BINPUT  1380
      DO  31  I=1,NSEG                                                      BINPUT  1390
   31 WRITE (6,32) I,SEG(I),CGS(I),W(I),(PHI(J,I),J=1,3),                   BINPUT  1400
     *              (BD(J,I),J=1,6),(YPRPMI(J,I),J=1,3)                    BINPUT  1410
   32 FORMAT(I3,1X,A4,2X,A1,F11.3,2X,3F9.4,2(2X,3F8.3),1X,3F9.2)            BINPUT  1420
                                                                            BINPUT  1430
C     PRINT CARDS  B.3.J  FOR EACH JOINT.                                   BINPUT  1440
C                                                                           BINPUT  1450
      WRITE (6,33) UNITL,UNITL                                              BINPUT  1460
   33 FORMAT(///120X,'CARDS B.3'/                                           BINPUT  1470
     * 3X,'JOINT',15X,'LOCATION(',A4,') - SEG(JNT)',                       BINPUT  1480
     *             3X,'LOCATION(',A4,') - SEG(J+1)',                      BINPUT  1490
     *             2X, 'PRIN. AXIS(DEG) - SEG(JNT)',                      BINPUT  1500
```

```
      NFLX = 0                                                          BINPUT  0510
      IF (NJNT.EQ.0) GO TO 27                                           BINPUT  0520
      DO  14  J=1,NJNT                                                  BINPUT  0530
      READ (5,15) JOINT(J),JS(J),JNT(J),IPIN(J),(SR(I,2*J-1),I=1,3),    BINPUT  0540
     *            (SR(I,2*J),I=1,3),(YPR1(I,J),I=1,3),(YPR2(I,J),I=1,3),BINPUT  0550
     *            (YPR3(I,J),I=1,3),(IDYPR(I,J),I=1,6)                  BINPUT  0560
      ID1 = IDYPR(1,J)                                                  BINPUT  0570
      ID4 = IDYPR(4,J)                                                  BINPUT  0580
      DO  14  I=1,3                                                     BINPUT  0590
      IF  (ID1.EQ.0)  IDYPR(I  ,J) = 4-I                                BINPUT  0600
   14 IF  (ID4.EQ.0)  IDYPR(I+3,J) = 4-I                                BINPUT  0610
   15 FORMAT(A4,1X,A1,2I4,6F6.0/14X,9F6.0,6I2)                          BINPUT  0620
                                                                        BINPUT  0630
C                                                                       BINPUT  0640
C     COMPUTE NFLX AND NFLEX ARRAY FROM NEGATIVE VALUES OF JNT(J).      BINPUT  0650
C     NFLX WILL BE NUMBER OF CONSTRAINT TORQUES FOR FLEXIBLE SEGMENTS.  BINPUT  0660
C     NFLEX(1, ) REFERENCE SEGMENT (LOWEST NUMBERED SEGMENT OF CHAIN)   BINPUT  0670
C     NFLEX(2, ) INTERIOR SEGMENT NUMBERS                               BINPUT  0680
C     NFLEX(3, ) TERMINATING SEGMENT (HIGHEST NUMBERED SEGMENT IN CHAIN)BINPUT  0690
C     VALUES OF NFLEX NEED NOT BE SEQUENTIAL BUT MUST BE ORDERED.       BINPUT  0700
C     FLEXIBLE SEGMENT MUST BE SIMPLE CHAIN, I.E., BRANCHING SEGMENTS   BINPUT  0710
C     CANNOT BE ATTACHED TO INTERIOR SEGMENTS BUT MAY BE ATTACHED TO    BINPUT  0720
C     REFERENCE OR TERMINATING SEGMENTS.                                BINPUT  0730
C                                                                       BINPUT  0740
      DO  16  J=1,NJNT                                                  BINPUT  0750
   16 KNT(J) = JNT(J)                                                   BINPUT  0760
      DO  22  J=1,NJNT                                                  BINPUT  0770
      IF (KNT(J).GE.0) GO TO 22                                         BINPUT  0780
      NFA = NFLX+1                                                      BINPUT  0790
      IT = J+1                                                          BINPUT  0800
      IF (IT.GT.NJNT) GO TO 18                                          BINPUT  0810
      JP1 = J+1                                                         BINPUT  0820
      DO  17  L=JP1,NJNT                                                BINPUT  0830
      IF (IABS(KNT(L)).NE.IT) GO TO 17                                  BINPUT  0840
      KL = KNT(L)                                                       BINPUT  0850
      KNT(L) = 0                                                        BINPUT  0860
      IF (KL.GT.0) GO TO 18                                             BINPUT  0870
      NFLX = NFLX+1                                                     BINPUT  0880
      NFLEX(1,NFLX) = IABS(KNT(J))                                      BINPUT  0890
      NFLEX(2,NFLX) = IT                                                BINPUT  0900
      IT = L+1                                                          BINPUT  0910
   17 CONTINUE                                                          BINPUT  0920
   18 IF (NFLX.GE.NFA) GO TO 20                                         BINPUT  0930
      WRITE (6,19)                                                      BINPUT  0940
   19 FORMAT('0ERROR IN DEFINING FLEXIBLE SEGMENTS, ONLY ONE NEGATIVE JN BINPUT  0950
     *T IN STRING. PROGRAM TERMINATED.')                                BINPUT  0960
      STOP 3                                                            BINPUT  0970
   20 DO  21  K=NFA,NFLX                                                BINPUT  0980
   21 NFLEX(3,K) = IT                                                   BINPUT  0990
   22 CONTINUE                                                          BINPUT  1000
C
```

```
      SUBROUTINE BINPUT                                                 BINPUT  0010
C                                                          REV 20 05/06/80  BINPUT  0020
C     READS THE INPUT CARDS THAT CONTAINS THE PHYSICAL DIMENSIONS AND   BINPUT  0030
C     CHARACTERISTICS OF THE CRASH VICTIM'S BODY SEGMENTS AND JOINTS.   BINPUT  0040
C                                                                       BINPUT  0050
      IMPLICIT REAL*8(A-H,O-Z)                                          BINPUT  0060
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           BINPUT  0070
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)        BINPUT  0080
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(3,60),HA(3,60),HB(3,60), BINPUT  0090
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),     BINPUT  0100
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)    BINPUT  0110
      COMMON/CNTSRF/ PL(17,30),BELT(20,8),TPTS(6,8),BD(24,40)           BINPUT  0120
      COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),           BINPUT  0130
     *               BLTTTL(5,8),PLTTL(5,30),BAGTTL(5,6),SEG(30),       BINPUT  0140
     *               JOINT(30),CGS(30),JS(30)                           BINPUT  0150
      REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT      BINPUT  0160
      LOGICAL*1 CGS,JS                                                  BINPUT  0170
      COMMON/INTEST/ SGTEST(3,4,30),XTEST(3,120),SEGT(120),REGT(120)    BINPUT  0180
      REAL    SEGT                                                      BINPUT  0190
      COMMON/FLXBLE/ HF(4,12,8),B42(3,3,24),V4(3,8),NFLEX(3,8)          BINPUT  0200
      COMMON/CEULER/ IEULER(30),HIR(3,3,30),ANG(3,30),ANGD(3,30),       BINPUT  0210
     *               FE(3,30),TQE(3,30),CONST(3,30)                     BINPUT  0220
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         BINPUT  0230
     *               UNITL,UNITM,UNITT,GRAVTY(3)                        BINPUT  0240
      COMMON/RSAVE/  XSG(3,20,3),DPMI(3,3,30),LPMI(30),NSG(7),MSG(20,7) BINPUT  0250
      COMMON/TEMPVS/ YPR1(3,30),YPR2(3,30),YPR3(3,30),YPRPMI(3,30),     BINPUT  0260
     *               T1(6),TMP1(3,3),TMP2(3,3),KNT(30),IDYPR(6,30)      BINPUT  0270
      CALL ELTIME(1, 2)                                                 BINPUT  0280
C                                                                       BINPUT  0290
C     INPUT CARD B.1                                                    BINPUT  0300
C                                                                       BINPUT  0310
      READ (5,11) NSEG,NJNT,BDYTTL                                      BINPUT  0320
   11 FORMAT (2I6,8X,5A4)                                               BINPUT  0330
C                                                                       BINPUT  0340
C     INPUT CARDS  B.2.I  FOR EACH SEGMENT.                             BINPUT  0350
C                                                                       BINPUT  0360
      DO  12  I=1,NSEG                                                  BINPUT  0370
      READ (5,13) SEG(I),CGS(I),W(I),(PHI(J,I),J=1,3),                  BINPUT  0380
     *            (BD(J,I),J=1,6),LPMI(I)                             BINPUT  0390
   13 FORMAT(A4,1X,A1,10F6.0,I4)                                        BINPUT  0400
      DO  81  J=1,3                                                     BINPUT  0410
      IDYPR(J,I) = 4-J                                                  BINPUT  0420
   81 YPRPMI(J,I) = 0.0                                                 BINPUT  0430
      IF  (LPMI(I).EQ.0)  GO TO 12                                      BINPUT  0440
      READ (5,82) (YPRPMI(J,I),J=1,3)                                   BINPUT  0450
   82 FORMAT(12X,3F6.0)                                                 BINPUT  0460
   12 CALL DRCYPR (DPMI(1,1,I),YPRPMI(1,I),IDYPR(1,I))                  BINPUT  0470
C                                                                       BINPUT  0480
C     INPUT CARDS  B.3.J  FOR EACH JOINT.                               BINPUT  0490
C                                                                       BINPUT  0500
```

```
C     COMPUTE COMPONENTS OF RELATIVE VELOCITY IN TANGENT PLANE,          BGG     1510
C     FRICTION FORCE AND TOTAL FORCE VECTOR.                             BGG     1520
C                                                                        BGG     1530
      S3 = 0.0                                                           BGG     1540
      DO 78 K=1,3                                                        BGG     1550
      T6(K) = T5(K)-SUM*PLANE(K,3)                                       BGG     1560
   78 S3 = S3+T6(K)**2                                                   BGG     1570
      SQ3 = DSQRT(S3)                                                    BGG     1580
      IF (SQ3.LT.S3TEST) SQ3=S3TEST/(2.0-SQ3/S3TEST)                     BGG     1590
      FF = VSCS*DSQRT(FM)/SQ3                                            BGG     1600
      DO 79 K=1,3                                                        BGG     1610
   79 FORCE(K) = FORCE(K)-FF*T6(K)                                       BGG     1620
C                                                                        BGG     1630
C     COMPUTE FRB: FORCE ON REACTION SURFACE IN ITS LOCAL REFERENCE.     BGG     1640
C            TORQ: TORQUE ON AIRBAG IN AIRBAG REFERENCE.                 BGG     1650
C             TQB: TORQUE ON REACTION SURFACE IN ITS LOCAL REFERENCE.    BGG     1660
C             FRA: FORCE ON AIRBAG IN INERTIAL REFERENCE.                BGG     1670
C                                                                        BGG     1680
       CALL DOT31(DAB,FORCE,FRB)                                         BGG     1690
       CALL CROSS(YFA,FORCE,TORQ)                                        BGG     1700
       CALL CROSS(FRB,YFB,TQB)                                           BGG     1710
       CALL DOT31(DA,FORCE,FRA)                                          BGG     1720
   99  RETURN                                                            BGG     1730
       END                                                               BGG     1740
```

```
      AREA=PI                                                          BGG     1010
      DO 70 L=1,2                                                      BGG     1020
      RA=RCRT(A,PLANE,CPA,L)                                           BGG     1030
      RB=RCRT(BA,PLANE,CBB,L)                                          BGG     1040
      IF(PP.GT.RA)RA=PP                                                BGG     1050
      R=(RA-RB)*.5                                                     BGG     1060
      RC=(RA+RB)*.5                                                    BGG     1070
      VP=PP/(RA+RB)                                                    BGG     1080
      VD=VP                                                            BGG     1090
      ALP=RC*DSQRT(VP*(2.-VP))                                         BGG     1100
      IF(R.GE.0.)GO TO 60                                              BGG     1110
      AB=RA+RB-PP                                                      BGG     1120
      BET=(RA**2-RB**2+AB**2)*.5/AB                                    BGG     1130
      ALP=DSQRT(RA**2-BET**2)                                          BGG     1140
       R=0.                                                            BGG     1150
      VD=1.-BET/RA                                                     BGG     1160
      VP=(PP+BET-RA)/RB                                                BGG     1170
60    VLM(L)=RB*(RB*VP)**2*(1.-VP/3.)+RA*(RA*VD)**2*(1.-VD/3.)         BGG     1180
      IF(R.GT.0.)VLM(L)=VLM(L)-ALP*R*R*(PI-2.*(DARSIN(1.-VP)+          BGG     1190
     X  (1.-VP)*ALP/RC))                                               BGG     1200
      VLM(L)=VLM(L)*PI                                                 BGG     1210
      AREA=AREA*ALP                                                    BGG     1220
70    IP=1                                                             BGG     1230
      VOL=(VLM(1)+VLM(2))*.5                                           BGG     1240
     IF  (IFULL.EQ.0)  GO TO 99                                        BGG     1250
C                                                                      BGG     1260
C     SET UP FORCE VECTOR ALONG LINE OF MAXIMUM PENETRATION.           BGG     1270
C                                                                      BGG     1280
      CALL DOT31(DAB,CBB,ZBB)                                          BGG     1290
      DO 76 K=1,3                                                      BGG     1300
      YFA(K)=CPB(K)+BFA(K)                                             BGG     1310
      YFB(K)=ZBB(K)+BFB(K)                                             BGG     1320
     FORCE(K) = -AREA*PLANE(K,3)                                       BGG     1330
76   T1(K) = VA(K)-VB(K)                                              BGG     1340
C                                                                      BGG     1350
C     COMPUTE ANGULAR VELOCITY COMPONENTS,RELATIVE VELOCITY, COMPONENTS BGG     1360
C     OF RELATIVE VELOCITY ALONG MAX PENETRATION LINE AND MAGNITUDE OF BGG     1370
C     FORCE.                                                           BGG     1380
C                                                                      BGG     1390
      CALL MAT31(DA,T1,T2)                                             BGG     1400
      CALL CROSS(WA,YFA,T1)                                            BGG     1410
      CALL CROSS(WB,YFB,T3)                                            BGG     1420
      CALL MAT31(DAB,T3,T4)                                            BGG     1430
      FM = 0.0                                                         BGG     1440
      SUM = 0.0                                                        BGG     1450
      DO 77 K=1,3                                                      BGG     1460
      T5(K) = T2(K)+T1(K)-T4(K)                                        BGG     1470
      SUM = SUM+T5(K)*PLANE(K,3)                                       BGG     1480
77    FM = FM+FORCE(K)**2                                              BGG     1490
C                                                                      BGG     1500
```

```
      DO 5 J=1,3                                                        BGG     0510
      BA(I,4)=BA(I,4)+DA(I,J)*(ZB(J)-ZA(J))                             BGG     0520
      DAB(I,J)=0.                                                       BGG     0530
      DO 5 K=1,3                                                        BGG     0540
    5 DAB(I,J)=DAB(I,J)+DA(I,K)*DB(J,K)                                 BGG     0550
                                                                        BGG     0560
C                                                                       BGG     0570
C     COMPUTE DISTANCE BETWEEN ELLIPSOID CENTERS AND                    BGG     0580
C     CONVERT ELLIPSOID MATRIX OF OBJECT TO AIRBAG REFERENCE.           BGG     0590
C                                                                       BGG     0600
      DO 10 I=1,3                                                       BGG     0610
      DO 10 J=1,3                                                       BGG     0620
      TEMP(I,J) = 0.0                                                   BGG     0630
      BA(I,4)=BA(I,4)+DAB(I,J)*BFB(J)                                   BGG     0640
      DO 10 K=1,3                                                       BGG     0650
   10 TEMP(I,J) = TEMP(I,J) + B(I,K)*DAB(J,K)                           BGG     0660
      CALL MAT33(DAB,TEMP,BA)                                           BGG     0670
C                                                                       BGG     0680
C     CHECK FOR INTERSECTION AND DETERMINE POINTS OF MAXIMUM PENETRATION BGG     0690
C                                                                       BGG     0700
      TB = 1.0                                                          BGG     0710
      CALL INTERS (A,BA,BA(1,4),TB,Y,TV(1),T1)                          BGG     0720
      IF (TB.GT.1.0) RETURN                                             BGG     0730
      CALL EDEPTH (A,BA,BA(1,4),TB,Y,CPA,CPB,TV(2),TV(3))               BGG     0740
C                                                                       BGG     0750
C     SET UP ORTHOGONAL SYSTEM USING VECTOR BETWEEN POINTS              BGG     0760
C     OF MAXIMUM PENETRATION AS ONE AXIS.                               BGG     0770
C                                                                       BGG     0780
      P = 0.                                                            BGG     0790
      DO 20  I=1,3                                                      BGG     0800
      PLANE(I,3) = CPA(I)-CPB(I)                                        BGG     0810
   20 P = PLANE(I,3)**2+P                                               BGG     0820
      IF  (P.LT.EPS(6))  GO TO 99                                       BGG     0830
      PP = DSQRT(P)                                                     BGG     0840
      DO 25  I=1,3                                                      BGG     0850
   25 TEMP(I,1) = PLANE(I,3)/PP                                         BGG     0860
      CALL ORTHO(PLANE,TEMP,4)                                          BGG     0870
C                                                                       BGG     0880
C     DEFINE PLANES AT MAXIMUM PENETRATION POINTS.                      BGG     0890
C                                                                       BGG     0900
   35 DO 40  I=1,3                                                      BGG     0910
      PLANE(4,I) = 0.0                                                  BGG     0920
      DO 40  J=1,3                                                      BGG     0930
   40 PLANE(4,I) = PLANE(4,I)+PLANE(J,I)*CPB(J)                         BGG     0940
      DO 45 K=1,3                                                       BGG     0950
   45 CBB(K)=CPB(K)-BA(K,4)                                             BGG     0960
C                                                                       BGG     0970
C     ESTIMATES OF VOLUME AND AREA BASED ON RADII OF CURVATURE          BGG     0980
C        AND PENETRATION.                                               BGG     0990
C                                                                       BGG     1000
      IP=2
```

```
      SUBROUTINE BGG(A,ZA,DA,BFA,VA,WA,                                  BGG     0010
     *               B,ZB,DB,BFB,VB,WB,                                  BGG     0020
     *               VSCS,IFULL,TV,FRA,TORQ,TQB,VOL)                     BGG     0030
C                                                        REV 19 08/05/78  BGG     0040
C     COMPUTES THE VOLUME OF INTERSECTION OF AN ELLIPSOIDAL AIRBAG       BGG     0050
C     WITH AN ELLIPSOIDAL BODY SEGMENT OR REACTION PANEL.                BGG     0060
C     ALSO COMPUTES THE FORCE PER UNIT PRESSURE AND TORQUE PER UNIT      BGG     0070
C     PRESSURE ON BOTH THE BAG AND THE INTERSECTING OBJECT.              BGG     0080
C                                                                        BGG     0090
C     ARGUMENTS:                                                         BGG     0100
C      AIRBAG INPUTS : A(3,3) - ELLIPSOID MATRIX                         BGG     0110
C                      ZA(3)  - C.G.                                     BGG     0120
C                      DA(3,3)- DIRECTION COSINE MATRIX                  BGG     0130
C                      BFA(3) - OFFSET                                   BGG     0140
C                      VA(3)  - CG VELOCITY(INERTIAL REF.)               BGG     0150
C                      WA(3)  - ANGULAR VELOCITY (LOCAL REF.)            BGG     0160
C                                                                        BGG     0170
C      CONTACT SURFACE B(3,3) - ELLIPSOID MATRIX                         BGG     0180
C                      ZB(3)  - C.G.                                     BGG     0190
C                      DB(3,3)- DIRECTION COSINE MATRIX                  BGG     0200
C                      BFB(3) - OFFSET                                   BGG     0210
C                      VB(3)  - CG VELOCITY (INERTIAL REF.)              BGG     0220
C                      WB(3)  - ANGULAR VELOCITY (LOCAL REF.)            BGG     0230
C                      VSCS   - COEFFICIENT OF SLIDING FRICTION          BGG     0240
C                      IFULL  - IF ZERO, COMPUTE VOL ONLY.               BGG     0250
C                      TV(3)  - MEMORY FOR SUBROUTINES INTERS & EDEPTH.  BGG     0260
C                                                                        BGG     0270
C            OUTPUT  : FRA(3) - FORCE ON BAG                             BGG     0280
C                      TORQ(3)- TORQUE ON BAG                            BGG     0290
C                      TOB(3) - TORQUE ON CONTACT SURFACE                BGG     0300
C                      VOL    - VOLUME OF INTERSECTION                   BGG     0310
      IMPLICIT REAL*8 (A-H,O-Z)                                          BGG     0320
      DIMENSION A(3,3),ZA(3),DA(3,3),BFA(3),VA(3),WA(3),B(3,3),ZB(3),    BGG     0330
     *          DB(3,3),BFB(3),VB(3),WB(3),FRA(3),TORQ(3),TQB(3),TV(3)   BGG     0340
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                          BGG     0350
     *               UNITL,UNITM,UNITT,GRAVTY(3)                         BGG     0360
      COMMON/TEMPVS/ DUMMY(200),DAB(3,3),BA(3,4),TEMP(3,3),Y(3),CPA(3),  BGG     0370
     *               CPB(3),PLANE(4,3),FORCE(3),CBB(3),VLM(3),FRB(3),    BGG     0380
     *      YFA(3),YFB(3),ZBB(3),T1(3),T2(3),T3(3),T4(3),T5(3),T6(3)     BGG     0390
C     NOTE: DUMMY IS USED BY SUBROUTINES AIRBAG AND AIRBGG.              BGG     0400
C                                                                        BGG     0410
C     INITIALIZATION                                                     BGG     0420
C                                                                        BGG     0430
      S3TEST = 10.0                                                      BGG     0440
       VOL=0.                                                            BGG     0450
       DO 5 I=1,3                                                        BGG     0460
      FRA(I)  = 0.0                                                      BGG     0470
      TORQ(I) = 0.0                                                      BGG     0480
      TQB (I) = 0.0                                                      BGG     0490
       BA(I,4)=-BFA(I)                                                   BGG     0500
```

```
      BELT(10,M) = 0.0                                                  BELTRT  1010
   17 BSF(1,NBSF) = SA                                                  BELTRT  1020
      BSF(2,NBSF) = FA                                                  BELTRT  1030
      BSF(3,NBSF) = SB                                                  BELTRT  1040
      BSF(4,NBSF) = FB                                                  BELTRT  1050
      IF  (FA+FB.LE.0.0) GO TO 31                                       BELTRT  1060
C                                                                       BELTRT  1070
C     COMPUTE FORCE VECTORS.                                            BELTRT  1080
C                                                                       BELTRT  1090
      DO  20  K=1,3                                                     BELTRT  1100
      UVA (K) = FA*UVA(K)                                               BELTRT  1110
   20 UVB(K) = FB*UVB(K)                                                BELTRT  1120
C                                                                       BELTRT  1130
C     CONVERT FORCES TO INERTIAL REFERENCE AND ADD TO U1 ARRAY.         BELTRT  1140
C                                                                       BELTRT  1150
      CALL DOT31(D(1,1,I),UVA,TT )                                      BELTRT  1160
      CALL DOT31(D(1,1,I),UVB,TTT)                                      BELTRT  1170
      DO 30 K=1,3                                                       BELTRT  1180
      TTT(K) = TTT(K)+TT(K)                                             BELTRT  1190
   30 U1(K,I) = U1(K,I)+TTT(K)                                          BELTRT  1200
C                                                                       BELTRT  1210
C     CONVERT TORQUES TO LOCAL REFERENCE AND ADD TO U2 ARRAY.           BELTRT  1220
C                                                                       BELTRT  1230
      CALL CROSS(APA,UVA,TT)                                            BELTRT  1240
      CALL CROSS(APB,UVB,TTT)                                           BELTRT  1250
      DO 40 K=1,3                                                       BELTRT  1260
   40 U2(K,I) = U2(K,I)+(TT(K)+TTT(K))                                  BELTRT  1270
   31 CONTINUE                                                          BELTRT  1280
      CALL ELTIME(2,22)                                                 BELTRT  1290
      RETURN                                                            BELTRT  1300
      END                                                               BELTRT  1310
```

```
      IF (BELT(11,M).LT.0.0) BELT(11,M)= -BELT(11,M)-TL                  BELTRT  0510
      IF (BELT(11,M).LT.0.0) BELT(11,M)=0.0                              BELTRT  0520
      BELT(12,M) = TLA+TLA/TL*BELT(11,M)                                 BELTRT  0530
      BELT(13,M) = TLB+TLB/TL*BELT(11,M)                                 BELTRT  0540
      B1213 = BELT(12,M) + BELT(13,M)                                    BELTRT  0550
      BELT(10,M) = B1213                                                 BELTRT  0560
      WRITE (6,14) M, B1213, BELT(12,M), BELT(13,M), UNITL, APA, APB     BELTRT  0570
   14 FORMAT('0 INITIAL LENGTHS OF BELT NO.',I3,' AND ITS SEGMENTS ARE', BELTRT  0580
     *       3F12.4,1X,A4/'0 INITIAL TANGENT POINTS ARE',2(3X,3F12.3))   BELTRT  0590
C                                                                        BELTRT  0600
C     CONVERT TANGENT POINTS TO INERTIAL REFERENCE AND STORE.            BELTRT  0610
C                                                                        BELTRT  0620
   11 CALL DOT31 (D(1,1,I),APA,TPTS(1,M))                                BELTRT  0630
      CALL DOT31 (D(1,1,I),APB,TPTS(4,M))                                BELTRT  0640
      DO  12  K=1,3                                                      BELTRT  0650
      TPTS(K  ,M) = TPTS(K  ,M) + SEGLP(K,I)                             BELTRT  0660
   12 TPTS(K+3,M) = TPTS(K+3,M) + SEGLP(K,I)                             BELTRT  0670
      SDOT = 0.0                                                         BELTRT  0680
      NCF = NTAB(NT+5)                                                   BELTRT  0690
      IF  (NCF.NE.0)  GO TO 15                                           BELTRT  0700
C                                                                        BELTRT  0710
C     ZERO BELT FRICTION; COMPUTE STRAIN AND FORCE OF ENTIRE BELT.       BELTRT  0720
C                                                                        BELTRT  0730
      B1213 = BELT(12,M)+BELT(13,M)                                      BELTRT  0740
      S = (TL-B1213)/B1213                                               BELTRT  0750
      SA = S                                                             BELTRT  0760
      SB = S                                                             BELTRT  0770
      IF  (S.LT.0.0)  S = 0.0                                            BELTRT  0780
      CALL FRCDFL (S,SDOT,NT,1,FA,ELOSS)                                 BELTRT  0790
      FB = FA                                                            BELTRT  0800
      GO TO 17                                                           BELTRT  0810
C                                                                        BELTRT  0820
C     FULL BELT FRICTION; COMPUTE STRAIN AND FORCE OF EACH PART OF BELT. BELTRT  0830
C                                                                        BELTRT  0840
   15 IF  (TL.GT.BELT(10,M))  GO TO 16                                   BELTRT  0850
      FA = 0.0                                                           BELTRT  0860
      FB = 0.0                                                           BELTRT  0870
      SA = (TL-BELT(10,M))/BELT(10,M)                                    BELTRT  0880
      SB = SA                                                            BELTRT  0890
      BELT(12,M) = TLA                                                   BELTRT  0900
      BELT(13,M) = TLB                                                   BELTRT  0910
      GO TO 17                                                           BELTRT  0920
   16 S = (TLA-BELT(12,M))/BELT(12,M)                                    BELTRT  0930
      SA = S                                                             BELTRT  0940
      IF  (S.LT.0.0)  S = 0.0                                            BELTRT  0950
      CALL FRCDFL (S,SDOT,NT,1,FA,ELOSS)                                 BELTRT  0960
      S = (TLB-BELT(13,M))/BELT(13,M)                                    BELTRT  0970
      SB = S                                                             BELTRT  0980
      IF  (S.LT.0.0)  S = 0.0                                            BELTRT  0990
      CALL FRCDFL (S,SDOT,NT+6,1,FB,ELOSS)                               BELTRT  1000
```

```
      SUBROUTINE BELTRT(I,II,MM,M,NT)                                    BELTRT  0010
C                                                          REV 20 05/23/80 BELTRT  0020
C     THE ROUTINE CALLS SUBROUTINE BELTG TO COMPUTE THE TANGENT POINTS    BELTRT  0030
C     AND BELT LENGTHS AND APPLIES THE RESTRAINT FORCES TO THE U1 ARRAY   BELTRT  0040
C     AND BELT TORQUES TO THE U2 ARRAY FOR ELLIPSOID(II) ATTACHED TO      BELTRT  0050
C     BODY SEGMENT (I) BY BELT (M) ATTACHED TO SEGMENT (MM).              BELTRT  0060
C                                                                         BELTRT  0070
      IMPLICIT REAL*8(A-H,O-Z)                                            BELTRT  0080
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,             BELTRT  0090
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)          BELTRT  0100
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),  BELTRT  0110
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)         BELTRT  0120
      COMMON/CNTSRF/ PL(17,30),BELT(20,8),TPTS(6,8),BD(24,40)             BELTRT  0130
      COMMON/TABLES/ MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(500),TAB(2600)  BELTRT  0140
      COMMON/FORCES/ PSF(7,30),BSF(4,20),SSF(10,20),BAGSF(3,20),          BELTRT  0150
     *               PRJNT(7,30),NPANEL(5),NPSF,NBSF,NSSF,NBGSF           BELTRT  0160
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                           BELTRT  0170
     *               UNITL,UNITM,UNITT,GRAVTY(3)                          BELTRT  0180
C     NOTE: BELTRT AND BELTG SHARE FIRST PART OF TEMPVS                   BELTRT  0190
      COMMON/TEMPVS/ APA(3),UVA(3),DLGA,UAA,APB(3),UVB(3),DLGB,UBB        BELTRT  0200
      DIMENSION TA(3),TB(3),ZA(3),ZB(3),TT(3),TTT(3)                      BELTRT  0210
C                                                                         BELTRT  0220
      CALL ELTIME(1,22)                                                   BELTRT  0230
C                                                                         BELTRT  0240
C     CONVERT SEGMENT POSITION TO SEGMENT REFERENCE.                      BELTRT  0250
C                                                                         BELTRT  0260
      MA = MM                                                             BELTRT  0270
      MB = MM                                                             BELTRT  0280
      CALL DOT31 (D(1,1,MA),BELT(1,M),TA)                                 BELTRT  0290
      CALL DOT31 (D(1,1,MB),BELT(4,M),TB)                                 BELTRT  0300
      DO  10  K=1,3                                                       BELTRT  0310
      TA(K) = SEGLP(K,MA) + TA(K) - SEGLP(K,I)                            BELTRT  0320
   10 TB(K) = SEGLP(K,MB) + TB(K) - SEGLP(K,I)                            BELTRT  0330
      CALL MAT31 (D(1,1,I),TA,ZA)                                         BELTRT  0340
      CALL MAT31 (D(1,1,I),TB,ZB)                                         BELTRT  0350
      DO  13  K=1,3                                                       BELTRT  0360
      ZA(K) = ZA(K) - BD(K+3,II)                                          BELTRT  0370
   13 ZB(K) = ZB(K) - BD(K+3,II)                                          BELTRT  0380
C                                                                         BELTRT  0390
C     COMPUTE NEW BELT LENGTHS AND EXPANSION.                             BELTRT  0400
C                                                                         BELTRT  0410
      CALL BELTG (ZA, ZB, BELT(7,M), BD(1,II))                            BELTRT  0420
      TLA = DLGA+UAA                                                      BELTRT  0430
      TLB = DLGB+UBB                                                      BELTRT  0440
      TL  = TLA+TLB                                                       BELTRT  0450
      IF  (TIME.NE.0.0)  GO TO 11                                         BELTRT  0460
C                                                                         BELTRT  0470
C     IF TIME=0, COMPUTE INITIAL BELT LENGTHS                             BELTRT  0480
C     AND STORE RESULTS IN BELT ARRAY.                                    BELTRT  0490
C                                                                         BELTRT  0500
```

```
C                                                                       BELTG   1510
   70 UAA=0.                                                            BELTG   1520
       UBB=0.                                                           BELTG   1530
       DO 80 K=1,3                                                      BELTG   1540
      APA(K) = APA(K)+XE(K)                                             BELTG   1550
      APB(K) = APB(K)+XE(K)                                             BELTG   1560
       UVA(K)=ZA(K)-APA(K)                                              BELTG   1570
       UVB(K)=ZB(K)-APB(K)                                              BELTG   1580
       APA(K)=APA(K)+BD(K+3)                                            BELTG   1590
       APB(K)=APB(K)+BD(K+3)                                            BELTG   1600
       UAA=UAA+UVA(K)**2                                                BELTG   1610
       UBB=UBB+UVB(K)**2                                                BELTG   1620
   80  CONTINUE                                                         BELTG   1630
       UAA=DSQRT(UAA)                                                   BELTG   1640
       UBB=DSQRT(UBB)                                                   BELTG   1650
       DO 90 K=1,3                                                      BELTG   1660
       UVA(K)=UVA(K)/UAA                                                BELTG   1670
       UVB(K)=UVB(K)/UBB                                                BELTG   1680
   90  CONTINUE                                                         BELTG   1690
C                                                                       BELTG   1700
C      OPTIONAL OUTPUT                                                  BELTG   1710
C                                                                       BELTG   1720
       IF (NPRT(15).EQ.0) GO TO 99                                      BELTG   1730
       WRITE(6,50)                                                      BELTG   1740
   50 FORMAT(1X,'BELT RESTRAINT')                                       BELTG   1750
       WRITE(6,60) APA,UVA,DLGA,UAA                                     BELTG   1760
       WRITE(6,60) APB,UVB,DLGB,UBB                                     BELTG   1770
   60 FORMAT (1X,1P8D15.5)                                              BELTG   1780
   99 CONTINUE                                                          BELTG   1790
       RETURN                                                           BELTG   1800
       END                                                              BELTG   1810
```

```
      B(1) = B(1)/YAY1                                                  BELTG   1010
      B(2) = B(2)/YAY1                                                  BELTG   1020
      B(3) = B(3)/YAY1                                                  BELTG   1030
C                                                                       BELTG   1040
C     COMPUTE ANGLES FROM FIXED POINT TO POSSIBLE TANGENT POINTS.       BELTG   1050
C                                                                       BELTG   1060
      UCUVA = UC(1)*UVA(1) + UC(2)*UVA(2) + UC(3)*UVA(3)                BELTG   1070
      UCUVB = UC(1)*UVB(1) + UC(2)*UVB(2) + UC(3)*UVB(3)                BELTG   1080
      UCACA = UC(1)*ACA(1) + UC(2)*ACA(2) + UC(3)*ACA(3)                BELTG   1090
      UCACB = UC(1)*ACB(1) + UC(2)*ACB(2) + UC(3)*ACB(3)                BELTG   1100
      UPUVA = UP(1)*UVA(1) + UP(2)*UVA(2) + UP(3)*UVA(3)                BELTG   1110
      UPUVB = UP(1)*UVB(1) + UP(2)*UVB(2) + UP(3)*UVB(3)                BELTG   1120
      UPACA = UP(1)*ACA(1) + UP(2)*ACA(2) + UP(3)*ACA(3)                BELTG   1130
      UPACB = UP(1)*ACB(1) + UP(2)*ACB(2) + UP(3)*ACB(3)                BELTG   1140
      TH1 = DATAN2(UPUVA-UPACA,UCUVA-UCACA)                             BELTG   1150
      TH2 = DATAN2(UPUVA+UPACA,UCUVA+UCACA)                             BELTG   1160
      TH3 = DATAN2(UPUVB-UPACB,UCUVB-UCACB)                             BELTG   1170
      TH4 = DATAN2(UPUVB+UPACB,UCUVB+UCACB)                             BELTG   1180
      TWOPI = 2.0*PI                                                    BELTG   1190
      IF (TH1.LT.0.0) TH1 = TWOPI + TH1                                 BELTG   1200
      IF (TH2.LT.0.0) TH2 = TWOPI + TH2                                 BELTG   1210
      IF (TH3.LT.0.0) TH3 = TWOPI + TH3                                 BELTG   1220
      IF (TH4.LT.0.0) TH4 = TWOPI + TH4                                 BELTG   1230
C                                                                       BELTG   1240
C     CHOOSE PROPER TANGENT POINTS AND BELT ARC LENGTHS.                BELTG   1250
C                                                                       BELTG   1260
      THMIN = DMIN1(TH1,TH2,TH3,TH4)                                    BELTG   1270
      IF (THMIN.EQ.TH1.AND.DMIN1(TH2,TH3,TH4).NE.TH4) GO TO 61          BELTG   1280
      IF (THMIN.EQ.TH2.AND.DMAX1(TH1,TH3,TH4).EQ.TH4) GO TO 61          BELTG   1290
      IF (THMIN.EQ.TH3.AND.DMIN1(TH1,TH2,TH4).NE.TH2) GO TO 63          BELTG   1300
      IF (THMIN.EQ.TH4.AND.DMAX1(TH1,TH2,TH3).EQ.TH2) GO TO 63          BELTG   1310
      GO TO 70                                                          BELTG   1320
   61 THA = TH1                                                         BELTG   1330
      THB = TWOPI-TH4                                                   BELTG   1340
      DO 62 K=1,3                                                       BELTG   1350
      APA(K) = UVA(K)-ACA(K)                                            BELTG   1360
   62 APB(K) = UVB(K)+ACB(K)                                            BELTG   1370
      GO TO 65                                                          BELTG   1380
   63 THA = TWOPI-TH2                                                   BELTG   1390
      THB = TH3                                                         BELTG   1400
      DO 64 K=1,3                                                       BELTG   1410
      APA(K) = UVA(K)+ACA(K)                                            BELTG   1420
   64 APB(K) = UVB(K)-ACB(K)                                            BELTG   1430
   65 CONTINUE                                                          BELTG   1440
      EPS1 = EPS(1)                                                     BELTG   1450
      DLGA = DABS(ELONG(B(1),B(2),B(3),EPS1,THA))                       BELTG   1460
      DLGB = DABS(ELONG(B(1),B(2),B(3),EPS1,THB))                       BELTG   1470
C                                                                       BELTG   1480
C     CALCULATE BELT LENGTHS AND UNIT VECTORS                           BELTG   1490
C     FROM TANGENT POINTS TO ANCHOR POINTS.                             BELTG   1500
```

```
      XE(K)  = XE(K)*GG                                                  BELTG   0510
      UC(K)  = ZC(K)-XE(K)                                               BELTG   0520
      APA(K) = UC(K)                                                     BELTG   0530
   15 APB(K) = UC(K)                                                     BELTG   0540
      YAY = GG*BET                                                       BELTG   0550
      YAY1 = 1.0-YAY                                                     BELTG   0560
      IF (YAY1.LE.EPS(6)) GO TO 70                                       BELTG   0570
C                                                                        BELTG   0580
C     CALCULATE POSSIBLE TANGENT POINTS FROM                             BELTG   0590
C       UVA,UVB: VECTORS FROM ELLIPSE CENTER TO MIDPOINT OF              BELTG   0600
C                LINE CONNECTING POSSIBLE TANGENT POINTS.                BELTG   0610
C       ACA,ACB: VECTORS FROM UVA,UVB TO TANGENT POINTS (POSITIVE).      BELTG   0620
C                                                                        BELTG   0630
      CALL MAT31 (BD(7),ZA,AX)                                           BELTG   0640
      CALL MAT31 (BD(7),ZB,BX)                                           BELTG   0650
      ZZA = AX(1)*ZA(1)+AX(2)*ZA(2)+AX(3)*ZA(3)                          BELTG   0660
      ZZB = BX(1)*ZB(1)+BX(2)*ZB(2)+BX(3)*ZB(3)                          BELTG   0670
      C2A = YAY1/(ZZA-YAY)                                               BELTG   0680
      C2B = YAY1/(ZZB-YAY)                                               BELTG   0690
      CALL CROSS(TC,AX,ACA)                                              BELTG   0700
      CALL CROSS(TC,BX,ACB)                                              BELTG   0710
      TTA = 0.0                                                          BELTG   0720
      TTB = 0.0                                                          BELTG   0730
      DO 21 I=1,3                                                        BELTG   0740
      DO 21 J=1,3                                                        BELTG   0750
      K = 3*J+I+3                                                        BELTG   0760
      TTA = TTA + ACA(I)*BD(K)*ACA(J)                                    BELTG   0770
   21 TTB = TTB + ACB(I)*BD(K)*ACB(J)                                    BELTG   0780
      C3A = DSQRT((1.0-C2A)*YAY1/TTA)                                    BELTG   0790
      C3B = DSQRT((1.0-C2B)*YAY1/TTB)                                    BELTG   0800
      TT = DSQRT(UC(1)**2 + UC(2)**2 + UC(3)**2)                         BELTG   0810
      DO 24 K=1,3                                                        BELTG   0820
      UVA(K) = C2A*(ZA(K)-XE(K))                                         BELTG   0830
      UVB(K) = C2B*(ZB(K)-XE(K))                                         BELTG   0840
      ACA(K) = C3A*ACA(K)                                                BELTG   0850
      ACB(K) = C3B*ACB(K)                                                BELTG   0860
      UC(K) = UC(K)/TT                                                   BELTG   0870
   24 B(K) = 0.0                                                         BELTG   0880
C                                                                        BELTG   0890
C     OBTAIN EQUATION OF ELLIPSE                                         BELTG   0900
C        B1*X**2 + 2*B2*X*Y + B3*Y**2 = 1                                BELTG   0910
C     IN UC,UP COORDINATES WHERE UC POINTS TO FIXED POINT.               BELTG   0920
C                                                                        BELTG   0930
      CALL CROSS(TC,UC,UP)                                               BELTG   0940
      DO 22 I=1,3                                                        BELTG   0950
      DO 22 J=1,3                                                        BELTG   0960
      K = 3*J+I+3                                                        BELTG   0970
      B(1) = B(1) + UC(I)*BD(K)*UC(J)                                    BELTG   0980
      B(2) = B(2) + UC(I)*BD(K)*UP(J)                                    BELTG   0990
   22 B(3) = B(3) + UP(I)*BD(K)*UP(J)                                    BELTG   1000
```

```
      SUBROUTINE BELTG (ZA,ZB,ZC,BD)                                    BELTG   0010
C                                                          REV 19 08/05/78 BELTG   0020
C     COMPUTE TANGENT POINTS, UNIT VECTORS FROM TANGENT POINTS TO       BELTG   0030
C     ANCHOR POINTS AND LENGTHS OF THE BELT SEGMENTS.                   BELTG   0040
C                                                                       BELTG   0050
C     ARGUMENTS:                                                        BELTG   0060
C                                                                       BELTG   0070
C         ZA,ZB - ANCHOR POINTS RELATIVE TO ELLIPSOID CENTER.           BELTG   0080
C         ZC    - FIXED POINT OF BELT ON SEGMENT ELLIPSOID.             BELTG   0090
C         BD    - SEGMENT ELLIPSOID SEMIAXES AND CENTER.                BELTG   0100
C                                                                       BELTG   0110
C     RESULTS ARE RETURNED TO CALLING ROUTINE VIA COMMON/TEMPVS/.       BELTG   0120
C                                                                       BELTG   0130
      IMPLICIT REAL*8 (A-H,O-Z)                                         BELTG   0140
      DIMENSION ZA(3),ZB(3),ZC(3),BD(24)                                BELTG   0150
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           BELTG   0160
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)        BELTG   0170
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         BELTG   0180
     *               UNITL,UNITM,UNITT,GRAVTY(3)                        BELTG   0190
C     NOTE: BELTRT AND BELTG SHARE FIRST PART OF TEMPVS                 BELTG   0200
      COMMON/TEMPVS/ APA(3),UVA(3),DLGA,UAA,APB(3),UVB(3),DLGB,UBB      BELTG   0210
     *              ,TA(3),TB(3),TC(3),UP(3),B(3)                       BELTG   0220
     *              ,UC(3),AX(3),XE(3),BX(3),ACA(3),ACB(3)              BELTG   0230
C                                                                       BELTG   0240
C     COMPUTE                                                           BELTG   0250
C       TC: NORMALIZED VECTOR OF BELT PLANE DETERMINED                  BELTG   0260
C           BY ANCHOR POINTS AND FIXED POINT.                           BELTG   0270
C                                                                       BELTG   0280
      DO 10 K=1,3                                                       BELTG   0290
      TA(K) = ZC(K)-ZA(K)                                               BELTG   0300
   10 TB(K) = ZC(K)-ZB(K)                                               BELTG   0310
      CALL CROSS(TB,TA,TC)                                              BELTG   0320
      S = DSQRT(TC(1)**2 + TC(2)**2 + TC(3)**2)                         BELTG   0330
      TC(1) = TC(1)/S                                                   BELTG   0340
      TC(2) = TC(2)/S                                                   BELTG   0350
      TC(3) = TC(3)/S                                                   BELTG   0360
C                                                                       BELTG   0370
C     GET DISTANCE OF BELT PLANE TO CENTER OF ELLIPSIOD.                BELTG   0380
C                                                                       BELTG   0390
      BET = TC(1)*ZC(1)+TC(2)*ZC(2)+TC(3)*ZC(3)                         BELTG   0400
C                                                                       BELTG   0410
C     COMPUTE                                                           BELTG   0420
C       XE: CENTER OF ELLIPSE DETERMINED BY INTERSECTION                BELTG   0430
C           OF BELT PLANE AND SEGMENT ELLIPSOID.                        BELTG   0440
C                                                                       BELTG   0450
      CALL MAT31 (BD(16),TC,XE)                                         BELTG   0460
      GG = BET/(TC(1)*XE(1)+TC(2)*XE(2)+TC(3)*XE(3))                    BELTG   0470
      DLGA = 0.0                                                        BELTG   0480
      DLGB = 0.0                                                        BELTG   0490
      DO 15 K=1,3                                                       BELTG   0500
```

```
      PCYV(J) = CYV(J)                                                    AIRBG3  0510
      PCYMIN(J) = CYMIN(J)                                                AIRBG3  0520
      PVBAG(J) = VBAG(J)                                                  AIRBG3  0530
   22 CALL AIRBGG(J)                                                      AIRBG3  0540
      VBAG(J) = CYV(J) + VOLBP(J)                                         AIRBG3  0550
      IF  (SCALE(J).EQ.1.0)  GO TO 23                                     AIRBG3  0560
      SCALE(J) = (VBAG(J)/VBAGG(J))**THIRD                                AIRBG3  0570
      IF  (SCALE(J).LT.1.0)  GO TO 24                                     AIRBG3  0580
      SCALE(J) = 1.0                                                      AIRBG3  0590
      GO TO 22                                                            AIRBG3  0600
   23 IFULL(J) = -1                                                       AIRBG3  0610
      CYMOUT(J) = 0.0                                                     AIRBG3  0620
      PSW1 = (VBAG (J)-VBAGG(J))*PCYV(J)/PCYMIN(J)                        AIRBG3  0630
      PSW2 = (VBAGG(J)-PVBAG(J))* CYV(J)/ CYMIN(J)                        AIRBG3  0640
      SWITCH(J) = (PSW1+PSW2)/(VBAG(J)-PVBAG(J))                          AIRBG3  0650
      BAGPV(J) = CYPA(J)*(CYMIN(J)*SWITCH(J))**CYK(J)                     AIRBG3  0660
      PD(J) = BAGPV(J)/(CYV(J)**CYK(J)) - CYPA(J)                         AIRBG3  0670
   24 DO  25  K=1,3                                                       AIRBG3  0680
      BD(K,JB) = SCALE(J)*AB(K,J)                                         AIRBG3  0690
      IF  (SCALE(J).EQ.0.0)  GO TO 25                                     AIRBG3  0700
      BD(4*K+12,JB) = BD(K,JB)**2                                         AIRBG3  0710
      BD(4*K+ 3,JB) = 1.0/BD(4*K+12,JB)                                   AIRBG3  0720
   25 TMP(K) = DEPLOY(K,J) + BD(1,JB)*DPVCTR(K,J)                         AIRBG3  0730
      CALL PANEL (DBR(1,1,J),TMP,JB)                                      AIRBG3  0740
C                                                                         AIRBG3  0750
C     SET UP BAGSF ARRAY FOR OUTPUT.                                      AIRBG3  0760
C                                                                         AIRBG3  0770
   31 BAGSF(1,NBGSF+1) = CYP(J)                                           AIRBG3  0780
      BAGSF(2,NBGSF+1) = CYT(J)                                           AIRBG3  0790
      BAGSF(3,NBGSF+1) =  PD(J)                                           AIRBG3  0800
      CALL DOT31 (D(1,1,JB),BD(4,JB),TMP)                                 AIRBG3  0810
      DO  32  K=1,3                                                       AIRBG3  0820
      BAGSF(K,NBGSF+3) = BD(K,JB)                                         AIRBG3  0830
   32 TMP(K) = TMP(K) + SEGLP(K,JB) - SEGLP(K,NVEH)                       AIRBG3  0840
      CALL MAT31 (D(1,1,NVEH),TMP,BAGSF(1,NBGSF+2))                       AIRBG3  0850
      CALL YPRDEG (D(1,1,JB),BAGSF(1,NBGSF+4))                            AIRBG3  0860
      NBGSF = NBGSF + 5 + NPANEL(J) + MNBAG(J)                            AIRBG3  0870
      GO TO 50                                                            AIRBG3  0880
C                                                                         AIRBG3  0890
C     START OF INTEGRATION STEP WITH IFULL =-1, RESET INTEGRATOR.         AIRBG3  0900
C                                                                         AIRBG3  0910
   41 IFULL(J) = 1                                                        AIRBG3  0920
      IRESET = -1                                                         AIRBG3  0930
   49 PYMOUT(J) = CYMOUT(J)                                               AIRBG3  0940
   50 CONTINUE                                                            AIRBG3  0950
   99 CALL ELTIME(2,29)                                                   AIRBG3  0960
      RETURN                                                              AIRBG3  0970
      END                                                                 AIRBG3  0980
```

```
      SUBROUTINE AIRBG3(IRESET)                                          AIRBG3  0010
C                                                        REV 20 04/11/80  AIRBG3  0020
C                                                                         AIRBG3  0030
C     THIS SUBROUTINE IS CALLED BY SUBROUTINE UPDATE AT START (IRESET=1)  AIRBG3  0040
C     AND END (IRESET=2) OF EACH INTEGRATION STEP TO DETERMINE IF EACH    AIRBG3  0050
C     AIRBAG HAS BEEN FULLY INFLATED.                                     AIRBG3  0060
C                                                                         AIRBG3  0070
      IMPLICIT REAL*8 (A-H,O-Z)                                           AIRBG3  0080
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,             AIRBG3  0090
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)          AIRBG3  0100
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),  AIRBG3  0110
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)         AIRBG3  0120
      COMMON/JBARTZ/ MNPL(   30),MNBLT(    8),MNSEG(   30),MNBAG(    6),  AIRBG3  0130
     *               MPL(3,5,30),MBLT(3,5,8),MSEG(3,5,30),MBAG(3,10,6),   AIRBG3  0140
     *               NTPL( 5,30),NTBLT( 5,8),NTSEG( 5,30)                 AIRBG3  0150
      COMMON/FORCES/ PSF(7,30),BSF(4,20),SSF(10,20),BAGSF(3,20),          AIRBG3  0160
     *               PRJNT(7,30),NPANEL(5),NPSF,NBSF,NSSF,NBGSF           AIRBG3  0170
      COMMON/CNTSRF/ PL(17,30),BELT(20,8),TPTS(6,8),BD(24,40)             AIRBG3  0180
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                           AIRBG3  0190
     *               UNITL,UNITM,UNITT,GRAVTY(3)                          AIRBG3  0200
      COMMON/ABDATA/ ZDEP(3,5),DBR(3,3,5),DPVCTR(3,5),DEPLOY(3,5),        AIRBG3  0210
     *               AB(3,5),B(9,4,5),ZR(3,4,5),BFB(3,4,5),DRR(9,4,5),    AIRBG3  0220
     *               VBAGG(5),VSCS(5),SPRK(5),CK(5),CMASS(5),CYMIN(5),    AIRBG3  0230
     *               CYMOUT(5),BAGPV(5),PD(5),VBAG(5),VOLBP(5),           AIRBG3  0240
     *               PCYV(5),PCYMIN(5),PVBAG(5),TV1(3,4,5),TV2(3,10,5),   AIRBG3  0250
     *               SWITCH(5),PYMOUT(5),SCALE(5),PREVT,IFULL(6)          AIRBG3  0260
      COMMON/CYDATA/ CYTD(5),CYPA(5),CYSP(5),CYTO(5),CYVO(5),CYCD(5),     AIRBG3  0270
     *               CYK(5),CYR(5),CYAT(5),CYPV(5),CYCDO(5),CYAO(5),      AIRBG3  0280
     *               CYPO(5),CYSS(5),CYLO(5),CYC(5),CYRHOO(5),CYVMAX(5),  AIRBG3  0290
     *               CYORFC(5),CYRHO(5),CYT(5),CYP(5),CYV(5)              AIRBG3  0300
      COMMON/TEMPVS/ TMP(9),TMP1(3)                                       AIRBG3  0310
      CALL ELTIME(1,29)                                                   AIRBG3  0320
      JRESET = IRESET                                                     AIRBG3  0330
      IF  (JRESET.EQ.1)  PREVT = TIME                                     AIRBG3  0340
      NBGSF = 0                                                           AIRBG3  0350
      DO  50  J=1,NBAG                                                    AIRBG3  0360
      IF  (MNBAG(J).EQ.0.)  GO TO 50                                      AIRBG3  0370
      JB = NVEH + J                                                       AIRBG3  0380
      JFULL = IFULL(J) + 2                                                AIRBG3  0390
      IF  (JFULL.LT.1 .OR. JFULL.GT.3)  GO TO 11                          AIRBG3  0400
      GO TO (13,14),JRESET                                                AIRBG3  0410
   11 WRITE  (6,12)  TIME                                                 AIRBG3  0420
   12 FORMAT ('0 ERROR IN SUBROUTINE AIRBG3 AT TIME =',F10.6)             AIRBG3  0430
      STOP 32                                                             AIRBG3  0440
   13 GO TO (41,49,49),JFULL                                              AIRBG3  0450
   14 GO TO (11,21,31),JFULL                                              AIRBG3  0460
C                                                                         AIRBG3  0470
C     END OF INTEGRATION STEP WHEN IFULL=0. TEST FOR FULL INFLATION.      AIRBG3  0480
C                                                                         AIRBG3  0490
   21 PD(J) = 0.0                                                         AIRBG3  0500
```

```
      RPHI(I,JB) = 1.0/PHI(I,JB)                                          AIRBG1  1510
      DO  36  K=1,4                                                       AIRBG1  1520
   36 SGTEST(I,K,JB) = 0.0                                                AIRBG1  1530
      DO  35  I=7,24                                                      AIRBG1  1540
   35 BD(I,JB) = 0.0                                                      AIRBG1  1550
      IFULL(J) = 0                                                        AIRBG1  1560
      CYMOUT(J) = 0.0                                                     AIRBG1  1570
      PYMOUT(J) = 0.0                                                     AIRBG1  1580
      DO  38  I=1,3                                                       AIRBG1  1590
      DO  37  K=1,4                                                       AIRBG1  1600
   37 TV1(I,K,J) = 0.0                                                    AIRBG1  1610
      DO  38  K=1,10                                                      AIRBG1  1620
   38 TV2(I,K,J) = 0.0                                                    AIRBG1  1630
C                                                                         AIRBG1  1640
C     AIR CYLINDER INITIALIZATION                                         AIRBG1  1650
C                                                                         AIRBG1  1660
      CYP0(J) = CYSP(J)+CYPA(J)                                           AIRBG1  1670
      CYSS(J) = DSQRT(CYK(J)*CYR(J)*CYT0(J)*G)                            AIRBG1  1680
      CYL0(J) = CYV0(J)/CYAT(J)                                           AIRBG1  1690
      CYK1    = CYK(J)-1.0                                                AIRBG1  1700
      CYK2    = 0.5*(CYK(J)+1.0)                                          AIRBG1  1710
      CYK3    = CYK2**(-CYK2/CYK1)                                        AIRBG1  1720
      CYC(J)  = 0.5*CYK1*CYSS(J)*CYCD(J)/CYL0(J)*CYK3                     AIRBG1  1730
      CYRHO0(J) = CYP0(J)/(CYR(J)*CYT0(J))                                AIRBG1  1740
      CYVMAX(J) = CYV0(J)/CYK(J)*CYP0(J)/CYPA(J)                          AIRBG1  1750
      CYORFC(J) = CYCD0(J)*CYA0(J)*G*DSQRT(2.0*CYPA(J)*CYK(J))/CYSS(J)    AIRBG1  1760
      IF (NPRT(22).NE.0) WRITE (6,39)                                     AIRBG1  1770
     *     (SEGLP(I,JB),I=1,3),(SEGLV(I,JB),I=1,3),(WMEG(I,JB),I=1,3),    AIRBG1  1780
     *     VBAGG(J),CYP0(J),CYSS(J),CYC(J),CYRHO0(J),CYVMAX(J),CYORFC(J)  AIRBG1  1790
   39 FORMAT('0 AIRBAG SINPUT'/(1X,9G14.6))                               AIRBG1  1800
   40 CONTINUE                                                            AIRBG1  1810
      PREVT = 0.0                                                         AIRBG1  1820
      RETURN                                                              AIRBG1  1830
      END                                                                 AIRBG1  1840
```

```
C     INCLUDES THE DEPLOYMENT POINT.                                          AIRBG1  1010
C                                                                             AIRBG1  1020
      READ (5,11) (B(I,K,J),I=1,3),(BFB(I,K,J),I=1,3),                        AIRBG1  1030
     *            (ZR(I,K,J),I=1,3),YP                                        AIRBG1  1040
   11 FORMAT(6F12.0)                                                          AIRBG1  1050
      WRITE (6,12) K,(B(I,K,J),I=1,3),(BFB(I,K,J),I=1,3),                     AIRBG1  1060
     *             (ZR(I,K,J),I=1,3),YP                                       AIRBG1  1070
   12 FORMAT('0 PANEL NO.',I4//                                               AIRBG1  1080
     * 24X,'PANEL ELLIPSOID SEMIAXES',43X,'C.G. OFFSET'/6X,6G20.9//           AIRBG1  1090
     * 29X,'PANEL LOCATION',32X,'YAW',16X,'PITCH',15X,'ROLL'/6X,6G20.9)        AIRBG1  1100
C                                                                             AIRBG1  1110
C     CONVERT B FROM ELLIPSOID SEMIAXES TO MATRIX                             AIRBG1  1120
C                                                                             AIRBG1  1130
      DO 21 I=1,3                                                             AIRBG1  1140
   21 TMP(I) = B(I,K,J)                                                       AIRBG1  1150
      DO 22 I=1,9                                                             AIRBG1  1160
   22 B(I,K,J) = 0.0                                                          AIRBG1  1170
      DO 23 I=1,3                                                             AIRBG1  1180
   23 B(4*I-3,K,J) = 1.0/TMP(I)**2                                            AIRBG1  1190
      CALL DRCYPR (DRR(1,K,J),YP,IDYPR)                                       AIRBG1  1200
      CALL MAT33 (B(1,K,J),DRR(1,K,J),TMP)                                    AIRBG1  1210
      CALL DOT33 (DRR(1,K,J),TMP,B(1,K,J))                                    AIRBG1  1220
      CALL DOT31 (DRR(1,K,J),BFB(1,K,J),TMP)                                  AIRBG1  1230
      DO  24  I=1,3                                                           AIRBG1  1240
   24 BFB(I,K,J) = TMP(I) + ZR(I,K,J)                                         AIRBG1  1250
   25 CONTINUE                                                                AIRBG1  1260
C                                                                             AIRBG1  1270
C     COMPUTE GEOMETRY OF DEPLOYMENT POINT ON FIRST PANEL.                    AIRBG1  1280
C                                                                             AIRBG1  1290
      CALL DRCYPR (DBR(1,1,J),YB,IDYPR)                                       AIRBG1  1300
      CALL DOT31 (DRR(1,1,J),ZDEP(1,J),DEPLOY(1,J))                           AIRBG1  1310
      DO  31  I=1,3                                                           AIRBG1  1320
      DPVCTR(I,J) = -DBR(1,I,J)                                               AIRBG1  1330
   31 DEPLOY(I,J) = DEPLOY(I,J) + BFB(I,1,J)                                  AIRBG1  1340
      CALL PANEL (DBR(1,1,J),DEPLOY(1,J),JB)                                  AIRBG1  1350
C                                                                             AIRBG1  1360
C     INITIALIZATION OF AIRBAG GEOMETRY.                                      AIRBG1  1370
C                                                                             AIRBG1  1380
      VBAGG(J) = 4.0/3.0*PI*AB(1,J)*AB(2,J)*AB(3,J)                           AIRBG1  1390
      PHI(1,JB) = (AB(2,J)**2+AB(3,J)**2)/5.0                                 AIRBG1  1400
      PHI(2,JB) = (AB(3,J)**2+AB(1,J)**2)/5.0                                 AIRBG1  1410
      PHI(3,JB) = (AB(1,J)**2+AB(2,J)**2)/5.0                                 AIRBG1  1420
      JNT(JB-1) = 0                                                           AIRBG1  1430
      IPIN(JB-1) = 0                                                          AIRBG1  1440
      SEG(JB) = BAG(J)                                                        AIRBG1  1450
      IF  (NBAG.EQ.1)  SEG(JB) = BAG(6)                                       AIRBG1  1460
      ISING(JB) = -1                                                          AIRBG1  1470
      RW(JB) = G/W(JB)                                                        AIRBG1  1480
      DO  36  I=1,3                                                           AIRBG1  1490
      BD(I,JB) = 0.0                                                          AIRBG1  1500
```

```
      IF  (L-1.GT.NJNT)  JNT (L-1) = 0                                  AIRBG1  0510
      IF  (L-1.GT.NJNT)  IPIN(L-1) = 0                                  AIRBG1  0520
      DO  19  I=1,3                                                     AIRBG1  0530
      SEGLP(I,L) = SEGLP(I,K)                                           AIRBG1  0540
      SEGLV(I,L) = SEGLV(I,K)                                           AIRBG1  0550
      SEGLA(I,L) = SEGLA(I,K)                                           AIRBG1  0560
      WMEG (I,L) = WMEG (I,K)                                           AIRBG1  0570
      WMEGD(I,L) = WMEGD(I,K)                                           AIRBG1  0580
      PHI  (I,L) = PHI  (I,K)                                           AIRBG1  0590
      RPHI (I,L) = RPHI (I,K)                                           AIRBG1  0600
      DO  18  J=1,3                                                     AIRBG1  0610
      D(I,J,L) = D(I,J,K)                                               AIRBG1  0620
   18 SGTEST(I,J,L) = SGTEST(I,J,K)                                     AIRBG1  0630
   19 SGTEST(I,4,L) = SGTEST(I,4,K)                                     AIRBG1  0640
      NGRND = NSEG + NBAG + 2                                           AIRBG1  0650
      DO 40 J=1,NBAG                                                    AIRBG1  0660
      JB = NVEH + J                                                     AIRBG1  0670
C                                                                       AIRBG1  0680
C     READ AND PRINT CARDS D.4.A -D.4.F FOR THE JTH AIRBAG.             AIRBG1  0690
C                                                                       AIRBG1  0700
      READ (5,13) (BAGTTL(I,J),I = 1,5),NPANEL(J),                      AIRBG1  0710
     *            (AB(I,J),I=1,3) , (BD(I,JB),I=4,6),                   AIRBG1  0720
     *             YB,(ZDEP(I,J),I=1,3),                                AIRBG1  0730
     *            W(JB),CYTD(J),CYPA(J),CYSP(J),CYT0(J),CYV0(J),        AIRBG1  0740
     *            CYCD(J),CYK(J),CYR(J),CYAT(J),CYPV(J),CYCD0(J),       AIRBG1  0750
     *             CYA0(J),SPRK(J),VSCS(J),CK(J),CMASS(J)               AIRBG1  0760
   13 FORMAT (5A4,I4/(6F12.0))                                          AIRBG1  0770
      IF (MOD(J,2).EQ.1) WRITE (6,15)                                   AIRBG1  0780
   15 FORMAT('1 AIRBAG INPUTS',105X,'CARDS D.4')                        AIRBG1  0790
      WRITE (6,14) J,(BAGTTL(I,J),I = 1,5 ),                            AIRBG1  0800
     *            (AB(I,J),I=1,3) , (BD(I,JB),I=4,6),                   AIRBG1  0810
     *             YB,(ZDEP(I,J),I=1,3),                                AIRBG1  0820
     *            W(JB),CYTD(J),CYPA(J),CYSP(J),CYT0(J),CYV0(J),        AIRBG1  0830
     *            CYCD(J),CYK(J),CYR(J),CYAT(J),CYPV(J),CYCD0(J),       AIRBG1  0840
     *             CYA0(J),SPRK(J),VSCS(J),CK(J),CMASS(J)               AIRBG1  0850
   14 FORMAT('0 AIRBAG NO.',I4,4X,5A4//                                 AIRBG1  0860
     * 29X,'AIR BAG SEMIAXES',46X,'C.G. OFFSET'/6X,6G20.9//             AIRBG1  0870
     * 15X,'YAW',16X,'PITCH',15X,'ROLL',30X,'DEPLOYMENT POINT'          AIRBG1  0880
     *                                          /6X,6G20.9//            AIRBG1  0890
     * 15X,'XBM',16X,'CYTD',16X,'CYPA',16X,'CYSP',16X,'CYT0',16X,'CYV0' AIRBG1  0900
     *                                          /6X,6G20.9//            AIRBG1  0910
     * 14X,'CYCD',17X,'CYK',17X,'CYR',16X,'CYAT',16X,'CYPV',16X,'CYCD0' AIRBG1  0920
     *                                          /6X,6G20.9//            AIRBG1  0930
     *14X,'CYA0',16X,'SPRK',16X,'VSCS',17X,'CK',17X,'CMASS'/6X,5G20.9)   AIRBG1  0940
      KP = NPANEL(J)                                                    AIRBG1  0950
      DO  25  K=1,KP                                                    AIRBG1  0960
C                                                                       AIRBG1  0970
C     READ AND PRINT CARDS D.4.G AND D.4.H FOR THE KTH PANEL TO         AIRBG1  0980
C     CONTACT THE JTH AIRBAG. THESE PANELS ARE APPROXIMATED BY          AIRBG1  0990
C     ELLIPSOIDS. THE FIRST PANEL (K=1) IS THE REACTION PANEL THAT      AIRBG1  1000
```

```
      SUBROUTINE AIRBG1                                                 AIRBG1  0010
C                                                          REV 20 04/11/80 AIRBG1  0020
CC    READS AND PRINTS THE INPUT CARDS THAT DESCRIBE THE PHYSICAL       AIRBG1  0030
CC    DIMENSIONS AND GAS DYNAMICS OF THE AIRBAG RESTRAINTS AND          AIRBG1  0040
CC    PERFORMS INITIALIZATION REQUIRED BY THE AIRBAG ROUTINE.           AIRBG1  0050
C                                                                       AIRBG1  0060
      IMPLICIT REAL*8 (A-H,O-Z)                                         AIRBG1  0070
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           AIRBG1  0080
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)        AIRBG1  0090
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30), AIRBG1  0100
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       AIRBG1  0110
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(3,60),HA(3,60),HB(3,60), AIRBG1  0120
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),     AIRBG1  0130
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)    AIRBG1  0140
      COMMON/FORCES/ PSF(7,30),BSF(4,20),SSF(10,20),BAGSF(3,20),        AIRBG1  0150
     *               PRJNT(7,30),NPANEL(5),NPSF,NBSF,NSSF,NBGSF         AIRBG1  0160
      COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),           AIRBG1  0170
     *               BLTTTL(5,8),PLTTL(5,30),BAGTTL(5,6),SEG(30),       AIRBG1  0180
     *               JOINT(30),CGS(30),JS(30)                           AIRBG1  0190
      REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT      AIRBG1  0200
      LOGICAL*1 CGS,JS                                                  AIRBG1  0210
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         AIRBG1  0220
     *               UNITL,UNITM,UNITT,GRAVTY(3)                        AIRBG1  0230
      COMMON/CNTSRF/ PL(17,30),BELT(20,8),TPTS(6,8),BD(24,40)           AIRBG1  0240
      COMMON/INTEST/ SGTEST(3,4,30),XTEST(3,120),SEGT(120),REGT(120)    AIRBG1  0250
      REAL    SEGT                                                      AIRBG1  0260
      COMMON/ABDATA/ ZDEP(3,5),DBR(3,3,5),DPVCTR(3,5),DEPLOY(3,5),      AIRBG1  0270
     *               AB(3,5),B(9,4,5),ZR(3,4,5),BFB(3,4,5),DRR(9,4,5),  AIRBG1  0280
     *               VBAGG(5),VSCS(5),SPRK(5),CK(5),CMASS(5),CYMIN(5),  AIRBG1  0290
     *               CYMOUT(5),BAGPV(5),PD(5),VBAG(5),VOLBP(5),         AIRBG1  0300
     *               PCYV(5),PCYMIN(5),PVBAG(5),TV1(3,4,5),TV2(3,10,5), AIRBG1  0310
     *               SWITCH(5),PYMOUT(5),SCALE(5),PREVT,IFULL(6)        AIRBG1  0320
      COMMON/CYDATA/ CYTD(5),CYPA(5),CYSP(5),CYT0(5),CYV0(5),CYCD(5),   AIRBG1  0330
     *               CYK(5),CYR(5),CYAT(5),CYPV(5),CYCD0(5),CYA0(5),    AIRBG1  0340
     *               CYP0(5),CYSS(5),CYL0(5),CYC(5),CYRHO0(5),CYVMAX(5),AIRBG1  0350
     *               CYORFC(5),CYRHO(5),CYT(5),CYP(5),CYV(5)            AIRBG1  0360
      COMMON/TEMPVS/ TMP(9),TMP1(3)                                     AIRBG1  0370
      DIMENSION YB(3),YP(3),IDYPR(3)                                    AIRBG1  0380
      REAL BAG(6)                                                       AIRBG1  0390
      DATA BAG/4HBAG1,4HBAG2,4HBAG3,4HBAG4,4HBAG5,4HBAG /               AIRBG1  0400
      DATA IDYPR/3,2,1/                                                 AIRBG1  0410
C                                                                       AIRBG1  0420
CC    MAKE ROOM FOR BAG DATA IN SEGMENT ARRAYS BETWEEN VEH AND GRND.    AIRBG1  0430
C                                                                       AIRBG1  0440
      L = NSEG + NBAG + 2                                               AIRBG1  0450
      K = NSEG + 2                                                      AIRBG1  0460
      W(L)      = W(K)                                                  AIRBG1  0470
      RW(L)     = RW(K)                                                 AIRBG1  0480
      SEG(L)    = SEG(K)                                                AIRBG1  0490
      ISING(L)  = ISING(K)                                              AIRBG1  0500
```

```
      CYMIN(J) = CYV0(J)*(CYRHO0(J)-CYRHO(J))                                  AIRBGG  0510
      CYV(J)     = CYVMAX(J)*(1.0-Q2)                                          AIRBGG  0520
      IF  (TIME.LT.CYTD(J))  GO TO 31                                          AIRBGG  0530
      IF  (BD(1,JB).EQ.0.0)  GO TO 31                                          AIRBGG  0540
      IF  (TIME.LE.0.0)      GO TO 31                                          AIRBGG  0550
      VOLB = 0.0                                                               AIRBGG  0560
C                                                                              AIRBGG  0570
C     COMPUTE AIRBAG ELLIPSOID MATRIX AND ZERO BAG FORCE AND TORQUE.           AIRBGG  0580
C                                                                              AIRBGG  0590
      IF  (IFULL(J).NE.0)  GO TO 21                                            AIRBGG  0600
      SAB = SCALE(J)*AB(1,J)                                                   AIRBGG  0610
      DO  19  I=1,3                                                            AIRBGG  0620
   19 TMP(I) = DEPLOY(I,J) + SAB*DPVCTR(I,J)                                   AIRBGG  0630
      CALL DOT31 (D(1,1,NVEH),TMP,SEGLP(1,JB))                                AIRBGG  0640
      DO  20  I=1,3                                                            AIRBGG  0650
   20 SEGLP(I,JB) = SEGLP(I,JB) + SEGLP(I,NVEH)                                AIRBGG  0660
   21 DO  23  I=1,3                                                            AIRBGG  0670
      FORCE(I,J) = 0.0                                                         AIRBGG  0680
   23 TORA (I,J) = 0.0                                                         AIRBGG  0690
C                                                                              AIRBGG  0700
C     COMPUTE FORCE,TORQUE AND VOLUME OF INTERSECTION                          AIRBGG  0710
C     OF AIRBAG WITH REACTION PANEL ELLIPSOIDS.                                AIRBGG  0720
C                                                                              AIRBGG  0730
      KP = NPANEL(J)                                                           AIRBGG  0740
      DO  26  K=1,KP                                                           AIRBGG  0750
      CALL BGG(                                                                AIRBGG  0760
     *   BD(7,JB),SEGLP(1,JB),D(1,1,JB),BD(4,JB),SEGLV(1,JB),WMEG(1,JB),      AIRBGG  0770
     *      B(1,K,J),SEGLP(1,NVEH),D(1,1,NVEH),BFB(1,K,J),SEGLV(1,NVEH),      AIRBGG  0780
     *      WMEG(1,NVEH),VSCS(J),IFULL(J),TV1(1,K,J),                          AIRBGG  0790
     *      FRA(1,K),TORQ,TQB,VOLP(K,J))                                       AIRBGG  0800
      VOLBP(J) = VOLBP(J) + VOLP(K,J)                                          AIRBGG  0810
      DO  26  I=1,3                                                            AIRBGG  0820
      FORCE(I,J) = FORCE(I,J) + FRA(I,K)                                       AIRBGG  0830
   26 TORA (I,J) = TORA (I,J) + TORQ(I)                                        AIRBGG  0840
C                                                                              AIRBGG  0850
C     COMPUTE FORCE,TORQUE AND VOLUME OF INTERSECTION                          AIRBGG  0860
C     OF AIRBAG WITH CONTACTING SEGMENT ELLIPSOIDS.                            AIRBGG  0870
C                                                                              AIRBGG  0880
      KBAG = MNBAG(J)                                                          AIRBGG  0890
      DO  30  I=1,KBAG                                                         AIRBGG  0900
      M  = MBAG(2,I,J)                                                         AIRBGG  0910
      MM = MBAG(3,I,J)                                                         AIRBGG  0920
      CALL BGG(                                                                AIRBGG  0930
     *   BD(7,JB),SEGLP(1,JB),D(1,1,JB),BD(4,JB),SEGLV(1,JB),WMEG(1,JB),      AIRBGG  0940
     *      BD(7,MM),SEGLP(1,M),D(1,1,M),BD(4,MM),SEGLV(1,M),WMEG(1,M),       AIRBGG  0950
     *      VSCS(J),IFULL(J),TV2(1,I,J),FRB(1,I),TORQ,TQB(1,I),VOL(I))        AIRBGG  0960
      IF  (VOL(I).EQ.0.0)  GO TO 30                                            AIRBGG  0970
      VOLB = VOLB + VOL(I)                                                     AIRBGG  0980
      DO  28  K=1,3                                                            AIRBGG  0990
      FORCE(K,J) = FORCE(K,J) + FRB(K,I)                                       AIRBGG  1000
   28 TORA (K,J) = TORA (K,J) + TORQ(K)                                        AIRBGG  1010
   30 CONTINUE                                                                 AIRBGG  1020
      VOLBP(J) = VOLBP(J) + VOLB                                               AIRBGG  1030
   31 RETURN                                                                   AIRBGG  1040
      END                                                                      AIRBGG  1050
```

```
      SUBROUTINE AIRBGG(J)                                                  AIRBGG  0010
C                                                          REV 20 04/11/80  AIRBGG  0020
C     CALLED BY SUBROUTINES AIRBAG AND AIRBG3 TO COMPUTE VOLUMES OF        AIRBGG  0030
C     INTERSECTION BETWEEN AIRBAGS AND PANELS AND SEGMENTS.                AIRBGG  0040
C                                                                          AIRBGG  0050
      IMPLICIT REAL*8 (A-H,O-Z)                                            AIRBGG  0060
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,              AIRBGG  0070
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)           AIRBGG  0080
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),   AIRBGG  0090
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)          AIRBGG  0100
      COMMON/JBARTZ/ MNPL(   30),MNBLT(    8),MNSEG(   30),MNBAG(    6),  AIRBGG  0110
     *               MPL(3,5,30),MBLT(3,5,8),MSEG(3,5,30),MBAG(3,10,6),    AIRBGG  0120
     *               NTPL( 5,30),NTBLT( 5,8),NTSEG( 5,30)                  AIRBGG  0130
      COMMON/FORCES/ PSF(7,30),BSF(4,20),SSF(10,20),BAGSF(3,20),           AIRBGG  0140
     *               PRJNT(7,30),NPANEL(5),NPSF,NBSF,NSSF,NBGSF            AIRBGG  0150
      COMMON/CNTSRF/ PL(17,30),BELT(20,8),TPTS(6,8),BD(24,40)              AIRBGG  0160
      COMMON/ABDATA/ ZDEP(3,5),DBR(3,3,5),DPVCTR(3,5),DEPLOY(3,5),         AIRBGG  0170
     *               AB(3,5),B(9,4,5),ZR(3,4,5),BFB(3,4,5),DRR(9,4,5),     AIRBGG  0180
     *               VBAGG(5),VSCS(5),SPRK(5),CK(5),CMASS(5),CYMIN(5),     AIRBGG  0190
     *               CYMOUT(5),BAGPV(5),PD(5),VBAG(5),VOLBP(5),            AIRBGG  0200
     *               PCYV(5),PCYMIN(5),PVBAG(5),TV1(3,4,5),TV2(3,10,5),    AIRBGG  0210
     *               SWITCH(5),PYMOUT(5),SCALE(5),PREVT,IFULL(6)           AIRBGG  0220
      COMMON/CYDATA/ CYTD(5),CYPA(5),CYSP(5),CYT0(5),CYV0(5),CYCD(5),      AIRBGG  0230
     *               CYK(5),CYR(5),CYAT(5),CYPV(5),CYCD0(5),CYA0(5),       AIRBGG  0240
     *               CYP0(5),CYSS(5),CYL0(5),CYC(5),CYRHO0(5),CYVMAX(5),   AIRBGG  0250
     *               CYORFC(5),CYRHO(5),CYT(5),CYP(5),CYV(5)               AIRBGG  0260
      COMMON/TEMPVS/ TMP(9),TMP1(3),TORQ(3),FORCE(3,5),TORA(3,5),          AIRBGG  0270
     *           TQB(3,10),FRB(3,10),VOL(10),DELF(3),VOLP(4,5),FRA(4,5)    AIRBGG  0280
C     NOTE: THIS COMMON/TEMPVS/ IS SHARED BY AIRBAG AND AIRBGG.            AIRBGG  0290
      JB = NVEH + J                                                        AIRBGG  0300
      VOLBP(J) = 0.0                                                       AIRBGG  0310
C                                                                          AIRBGG  0320
C     COMPUTE THERMODYNAMIC PROPERTIES OF AIRBAG                           AIRBGG  0330
C         CYRHO  : DENSITY                                                 AIRBGG  0340
C         CYT    : TEMPERATURE                                             AIRBGG  0350
C         CYP    : PRESSURE                                                AIRBGG  0360
C         CYMIN  : MASS FLOW INTO BAG                                      AIRBGG  0370
C         VBCALC : CALCULATED VOLUME                                       AIRBGG  0380
C                                                                          AIRBGG  0390
      Q  = 1.0                                                             AIRBGG  0400
      Q1 = 1.0                                                             AIRBGG  0410
      Q2 = 1.0                                                             AIRBGG  0420
      IF  (TIME.LE.CYTD(J))  GO TO 13                                      AIRBGG  0430
      Q  = 1.0 + CYC(J)*(TIME-CYTD(J))                                     AIRBGG  0440
      CYK1 = 2.0/(CYK(J)-1.0)                                              AIRBGG  0450
      Q1 = 1.0/Q**CYK1                                                     AIRBGG  0460
      Q2 = 1.0/Q**(CYK(J)*CYK1)                                            AIRBGG  0470
   13 CYRHO(J) = CYRHO0(J)*Q1                                              AIRBGG  0480
      CYT(J)   = CYT0(J)/Q**2                                              AIRBGG  0490
      CYP(J)   = CYP0(J)*Q2                                                AIRBGG  0500
```

```
51    TMP(K) = BFB(K,1,J) + ZDEP(K,J)                                    AIRBAG  1010
      CALL DOT31 (D(1,1,NVEH),TMP,TMP1)                                  AIRBAG  1020
      DO  52  K=1,3                                                      AIRBAG  1030
      DELF(K) = TMP1(K) + SEGLP(K,NVEH) - SEGLP(K,JB)                    AIRBAG  1040
52    TMP(K) = BD(K+3,JB)                                                AIRBAG  1050
      TMP(1) = TMP(1) + BD(1,JB)                                         AIRBAG  1060
      CALL DOT31 (D(1,1,JB),TMP,TMP1)                                    AIRBAG  1070
      DO  53  K=1,3                                                      AIRBAG  1080
      DELF(K) = SPRK(J)*(DELF(K)-TMP1(K))                                AIRBAG  1090
      BAGSF(K,NBGSF+5) = DELF(K)                                         AIRBAG  1100
53    FORCE(K,J) = FORCE(K,J) + DELF(K)                                  AIRBAG  1110
      CALL MAT31 (D(1,1,JB),DELF,TMP1)                                   AIRBAG  1120
      CALL CROSS (TMP,TMP1,DELF)                                         AIRBAG  1130
      DO  54  K=1,3                                                      AIRBAG  1140
54    TORA(K,J) = TORA(K,J) + DELF(K)                                    AIRBAG  1150
55    XDD = CYMIN(J) - CYMOUT(J) + W(JB)                                 AIRBAG  1160
      FMASS = CMASS(J)*XDD/G                                             AIRBAG  1170
      TMASS = CMASS(J)*(XDD+W(JB)*2.0/3.0)/G                             AIRBAG  1180
      DO  56  I=1,3                                                      AIRBAG  1190
56    TMP(I) = WMEG(I,JB)*PHI(I,JB)                                      AIRBAG  1200
      CALL CROSS (WMEG(1,JB),TMP,TMP1)                                   AIRBAG  1210
      DO  57  I=1,3                                                      AIRBAG  1220
      SEGLA(I,JB) = FORCE(I,J)/FMASS + GRAVTY(I)                         AIRBAG  1230
57    WMEGD(I,JB) = (TORA(I,J)/TMASS-TMP1(I))*RPHI(I,JB)                 AIRBAG  1240
69    NBGSF = NBGSF + 5 + NPANEL(J) + MNBAG(J)                           AIRBAG  1250
70    CONTINUE                                                           AIRBAG  1260
      CALL ELTIME(2,24)                                                  AIRBAG  1270
      RETURN                                                             AIRBAG  1280
      END                                                                AIRBAG  1290
```

```
      BAGPV(J) = CYPA(J)*((CYMIN(J)-CYMOUT(J))*SWITCH(J))**CYK(J)          AIRBAG  0510
C                                                                          AIRBAG  0520
C     BAG IS FULLY INFLATED, COMPUTE DIFFERENTIAL PRESSURE                 AIRBAG  0530
C                                                                          AIRBAG  0540
      PD(J) = BAGPV(J)/(VBAG(J)-VOLBP(J))**CYK(J) - CYPA(J)                AIRBAG  0550
      JB = NVEH + J                                                        AIRBAG  0560
      KP = NPANEL(J)                                                       AIRBAG  0570
      KBAG = MNBAG(J)                                                      AIRBAG  0580
C                                                                          AIRBAG  0590
C     OPTIONAL DIAGNOSTIC OUTPUT                                           AIRBAG  0600
C                                                                          AIRBAG  0610
      IF  (NPRT(21).NE.0)  WRITE  (6,41)                                   AIRBAG  0620
     *  ((FRB(I,K),I=1,3),(TQB(I,K),I=1,3),K=1,KBAG),(FORCE(I,J),I=1,3),    AIRBAG  0630
     *  (TORA(I,J),I=1,3),TORQ,((FRA(I,K),I=1,3),VOLP(K,J),K=1,KP),         AIRBAG  0640
     *  (VOL(K),K=1,KBAG),VOLBP(J),CYMOUT(J),BAGPV(J),PD(J)                 AIRBAG  0650
  41  FORMAT ('0AIRBAG CONTCT'/(1X,9G14.6))                                AIRBAG  0660
      IF  (PD(J).LT.0.0)  PD(J) = 0.0                                      AIRBAG  0670
      IF  (PD(J).EQ.0.0)  GO TO 46                                         AIRBAG  0680
C                                                                          AIRBAG  0690
C     SET UP BAGSF ARRAY FOR OUTPUT ROUTINE                                AIRBAG  0700
C                                                                          AIRBAG  0710
      KBGSF = NBGSF+5                                                      AIRBAG  0720
      DO  42  K=1,KP                                                       AIRBAG  0730
      KBGSF = KBGSF+1                                                      AIRBAG  0740
      DO  42  I=1,3                                                        AIRBAG  0750
  42  BAGSF(I,KBGSF) = PD(J)*FRA(I,K)                                      AIRBAG  0760
      DO  45  I=1,KBAG                                                     AIRBAG  0770
      KBGSF = KBGSF+1                                                      AIRBAG  0780
      IF  (VOL(I).EQ.0.0)  GO TO 45                                        AIRBAG  0790
      M = MBAG(2,I,J)                                                      AIRBAG  0800
C                                                                          AIRBAG  0810
C     FINAL COMPUTATIONS OF FORCE AND TORQUE ON AIRBAG                     AIRBAG  0820
C                                                                          AIRBAG  0830
      DO  44  K=1,3                                                        AIRBAG  0840
      FRB(K,I) = PD(J)*FRB(K,I)                                            AIRBAG  0850
      BAGSF(K,KBGSF) = FRB(K,I)                                            AIRBAG  0860
      U1(K,M) = U1(K,M) - FRB(K,I)                                         AIRBAG  0870
  44  U2(K,M) = U2(K,M) + PD(J)*TQB(K,I)                                   AIRBAG  0880
  45  CONTINUE                                                             AIRBAG  0890
  46  DO  47  K=1,3                                                        AIRBAG  0900
      FORCE(K,J) = PD(J)*FORCE(K,J)                                        AIRBAG  0910
  47  TORA (K,J) = PD(J)*TORA (K,J)                                        AIRBAG  0920
      IF  (VOLP(1,J).NE.0.0)  GO TO 55                                     AIRBAG  0930
C                                                                          AIRBAG  0940
C     AIRBAG IS NOT INTERSECTING PRIMARY REACTION PANEL.                   AIRBAG  0950
C     COMPUTE ARTIFICIAL FORCE AND TORQUE WITH A LINEAR SPRING FUNCTION    AIRBAG  0960
C     IN AN ATTEMPT TO TIE +X SEMIAXIS ENDPOINT OF AIRBAG TO DEPLOYMENT    AIRBAG  0970
C     POINT ON REACTION PANEL.                                             AIRBAG  0980
C                                                                          AIRBAG  0990
      DO  51  K=1,3                                                        AIRBAG  1000
```

```
      SUBROUTINE  AIRBAG                                                    AIRBAG  0010
C                                                          REV 20 04/11/80  AIRBAG  0020
C     AIRBAG ROUTINE CALLED BY SUBROUTINE CONTCT TO DETERMINE THE INTER-   AIRBAG  0030
C     ACTION OF THE BAG WITH REACTION PANELS AND BODY SEGMENTS BY USE OF   AIRBAG  0040
C     SUBROUTINE BGG. THE DIFFERENTIAL PRESSURE,FORCE AND TORQUE ON THE    AIRBAG  0050
C     BAG IS EVALUATED AND THE RESULTING FORCE AND TORQUE ON THE BODY      AIRBAG  0060
C     SEGMENTS ARE ADDED TO THE U1 AND U2 ARRAYS.                          AIRBAG  0070
C                                                                          AIRBAG  0080
      IMPLICIT REAL*8 (A-H,O-Z)                                            AIRBAG  0090
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,              AIRBAG  0100
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)           AIRBAG  0110
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),   AIRBAG  0120
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)          AIRBAG  0130
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(3,60),HA(3,60),HB(3,60),    AIRBAG  0140
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),        AIRBAG  0150
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)       AIRBAG  0160
      COMMON/JBARTZ/ MNPL(   30),MNBLT(    8),MNSEG(   30),MNBAG(    6),  AIRBAG  0170
     *               MPL(3,5,30),MBLT(3,5,8),MSEG(3,5,30),MBAG(3,10,6),    AIRBAG  0180
     *               NTPL( 5,30),NTBLT( 5,8),NTSEG( 5,30)                  AIRBAG  0190
      COMMON/FORCES/ PSF(7,30),BSF(4,20),SSF(10,20),BAGSF(3,20),           AIRBAG  0200
     *               PRJNT(7,30),NPANEL(5),NPSF,NBSF,NSSF,NBGSF            AIRBAG  0210
      COMMON/CNTSRF/ PL(17,30),BELT(20,8),TPTS(6,8),BD(24,40)              AIRBAG  0220
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                            AIRBAG  0230
     *               UNITL,UNITM,UNITT,GRAVTY(3)                           AIRBAG  0240
      COMMON/ABDATA/ ZDEP(3,5),DBR(3,3,5),DPVCTR(3,5),DEPLOY(3,5),         AIRBAG  0250
     *               AB(3,5),B(9,4,5),ZR(3,4,5),BFB(3,4,5),DRR(9,4,5),     AIRBAG  0260
     *               VBAGG(5),VSCS(5),SPRK(5),CK(5),CMASS(5),CYMIN(5),     AIRBAG  0270
     *               CYMOUT(5),BAGPV(5),PD(5),VBAG(5),VOLBP(5),            AIRBAG  0280
     *               PCYV(5),PCYMIN(5),PVBAG(5),TV1(3,4,5),TV2(3,10,5),    AIRBAG  0290
     *               SWITCH(5),PYMOUT(5),SCALE(5),PREVT,IFULL(6)           AIRBAG  0300
      COMMON/CYDATA/ CYTD(5),CYPA(5),CYSP(5),CYTO(5),CYVO(5),CYCD(5),      AIRBAG  0310
     *               CYK(5),CYR(5),CYAT(5),CYPV(5),CYCDO(5),CYAO(5),       AIRBAG  0320
     *               CYPO(5),CYSS(5),CYLO(5),CYC(5),CYRHOO(5),CYVMAX(5),   AIRBAG  0330
     *               CYORFC(5),CYRHO(5),CYT(5),CYP(5),CYV(5)               AIRBAG  0340
      COMMON/TEMPVS/ TMP(9),TMP1(3),TORQ(3),FORCE(3,5),TORA(3,5),          AIRBAG  0350
     *          TQB(3,10),FRB(3,10),VOL(10),DELF(3),VOLP(4,5),FRA(4,5)     AIRBAG  0360
C     NOTE: THIS COMMON/TEMPVS/ IS SHARED BY AIRBAG AND AIRBGG.            AIRBAG  0370
      CALL ELTIME(1,24)                                                    AIRBAG  0380
      DELT = TIME-PREVT                                                    AIRBAG  0390
      NBGSF = 0                                                            AIRBAG  0400
      DO  70  J=1,NBAG                                                     AIRBAG  0410
      IF  (MNBAG(J).EQ.0)  GO TO 70                                        AIRBAG  0420
      IF  (IFULL(J).LE.0)  GO TO 69                                        AIRBAG  0430
      CALL AIRBGG(J)                                                       AIRBAG  0440
C                                                                          AIRBAG  0450
C     COMPUTE  CMOUT: MASS FLOW OUT OF BAG                                 AIRBAG  0460
C              BAGPV: UNDISTORTED BAG VOLUME                               AIRBAG  0470
C                                                                          AIRBAG  0480
   40 IF  (PD(J).GT.CYPV(J))  CYMOUT(J) = PYMOUT(J)                        AIRBAG  0490
     *                        + DELT*CYORFC(J)*DSQRT(PD(J))                AIRBAG  0500
```

```
      GG(5,I+2) = ZA                                                    ADJUST  0510
      GG(5,I+1) = ZA                                                    ADJUST  0520
   20 GG(5,I  ) = ZA                                                    ADJUST  0530
      GO TO (99,21,99,23,25),M                                          ADJUST  0540
C                                                                       ADJUST  0550
C     M = 2:                                                            ADJUST  0560
C                                                                       ADJUST  0570
   21 DO  22  I=1,NEQ                                                   ADJUST  0580
      ZA = GG(5,I)                                                      ADJUST  0590
      Y1 = Y(4,I)  - ZA*Y(3,I)                                          ADJUST  0600
      Y2 = GG(2,I) - ZA*GG(1,I)                                         ADJUST  0610
      Y3 = DER(I)  - ZA*VAR(I)                                          ADJUST  0620
      GG(3,I) = -Y1*GH(1,1) + Y2*GH(2,1) + Y3*GH(3,1)                   ADJUST  0630
      GG(4,I) =  Y1*GH(1,2) - Y2*GH(2,2) + Y3*GH(3,2)                   ADJUST  0640
      Y(1,I) = 0.5*(Y(1,I)+VAR(I))                                      ADJUST  0650
   22 Y(2,I) = 0.5*(Y(2,I)+DER(I))                                      ADJUST  0660
      GO TO 99                                                          ADJUST  0670
C                                                                       ADJUST  0680
C     M = 3,4:                                                          ADJUST  0690
C                                                                       ADJUST  0700
   23 DO  24  I=1,NEQ                                                   ADJUST  0710
      ZA = GG(5,I)                                                      ADJUST  0720
      Y1 = GG(2,I) - ZA*GG(1,I)                                         ADJUST  0730
      Y2 = Y(2,I)  - ZA*Y(1,I)                                          ADJUST  0740
      Y3 = DER(I)  - ZA*VAR(I)                                          ADJUST  0750
      GG(3,I) = -Y1*GH(1,3) + Y2*GH(2,3) - Y3*GH(3,3)                   ADJUST  0760
      GG(4,I) =  Y1*GH(1,4) - Y2*GH(2,4) + Y3*GH(3,4)                   ADJUST  0770
      U(1,I) = VAR(I)                                                   ADJUST  0780
   24 U(2,I) = DER(I)                                                   ADJUST  0790
      GO TO 99                                                          ADJUST  0800
C                                                                       ADJUST  0810
C     M = 5:                                                            ADJUST  0820
C                                                                       ADJUST  0830
   25 DO  26  I=1,NEQ                                                   ADJUST  0840
      ZA = GG(5,I)                                                      ADJUST  0850
      Y1 = GG(2,I) - ZA*GG(1,I)                                         ADJUST  0860
      Y2 = DER(I)  - ZA*VAR(I)                                          ADJUST  0870
      Y3 = U(2,I)  - ZA*U(1,I)                                          ADJUST  0880
      GG(3,I) = -Y1*GH(1,3) + Y2*GH(2,3) - Y3*GH(3,3)                   ADJUST  0890
      GG(4,I) =  Y1*GH(1,4) - Y2*GH(2,4) + Y3*GH(3,4)                   ADJUST  0900
      Y(1,I) = VAR(I)                                                   ADJUST  0910
   26 Y(2,I) = DER(I)                                                   ADJUST  0920
   99 RETURN                                                            ADJUST  0930
      END                                                               ADJUST  0940
```

```
      JCARD = 0                                                          EQUILB  0510
      IF  (NVAR.LT.1 .OR. NVAR.GT.10)  GO TO 65                          EQUILB  0520
      IF  (NCON.LT.0 .OR. NCON.GT.5 )  GO TO 65                          EQUILB  0530
      WRITE  (6,52)                                                      EQUILB  0540
   52 FORMAT('0',4X,'J',4X,'NTV',3X,'NI1',3X,'NSG',8X,'GX',12X,'XDEV',   EQUILB  0550
     *7X,'JPL',3X,'JSG',3X,'NAV',3X,'KSG(I,J),I=1,NAV',28X,'CARDS G.5'/) EQUILB  0560
      ICARD = 5                                                          EQUILB  0570
      DO  58  J=1,NVAR                                                   EQUILB  0580
      JCARD = J                                                          EQUILB  0590
      READ   (5,53)    NTV(J),NI1(J),NSG(J),GX(J),XDEV(J),               EQUILB  0600
     *              JPL(J),JSG(J),IAV,(KSG(I,J),I=1,IAV)                EQUILB  0610
   53 FORMAT(3I4,2F8.0,8I4)                                              EQUILB  0620
      NAV(J) = IAV                                                       EQUILB  0630
      WRITE  (6,54)  J,NTV(J),NI1(J),NSG(J),GX(J),XDEV(J),               EQUILB  0640
     *              JPL(J),JSG(J),IAV,(KSG(I,J),I=1,IAV)                EQUILB  0650
   54 FORMAT(4I6,2F15.6,8I6)                                             EQUILB  0660
      IF  (NTV(J).LT.1 .OR. NTV(J).GT.2    )  GO TO 65                   EQUILB  0670
      IF  (NI1(J).LT.1 .OR. NI1(J).GT.3    )  GO TO 65                   EQUILB  06B0
      IF  (NSG(J).LT.1 .OR. NSG(J).GT.NSEG)  GO TO 65                    EQUILB  0690
      IF  (NAV(J).LT.0 .OR. NAV(J).GT.5    )  GO TO 65                   EQUILB  0700
      IF  (JPL(J).LT.1 .OR. JPL(J).GT.NPL  )  GO TO 65                   EQUILB  0710
      IF  (JSG(J).LT.1 .OR. JSG(J).GT.NSEG)  GO TO 65                    EQUILB  0720
      K = JPL(J)                                                         EQUILB  0730
      NNPL = MNPL(K)                                                     EQUILB  0740
      IF  (NNPL.LT.1 .OR. NNPL.GT.5)  GO TO 65                           EQUILB  0750
      DO  55  I=1,NNPL                                                   EQUILB  0760
      IF  (JSG(J).NE.MPL(2,I,K))  GO TO 55                               EQUILB  0770
      JSG(J) = I                                                         EQUILB  0780
      GO TO 56                                                           EQUILB  0790
   55 CONTINUE                                                           EQUILB  0800
      GO TO 65                                                           EQUILB  0810
   56 IF  (NAV(J).LE.0)  GO TO 58                                        EQUILB  0820
      DO  57  I=1,IAV                                                    EQUILB  0830
      IF  (KSG(I,J).LT.1 .OR. KSG(I,J).GT.NSEG)  GO TO 65                EQUILB  0840
   57 CONTINUE                                                           EQUILB  0850
   58 CONTINUE                                                           EQUILB  0860
      IF  (NCON.LE.0)  GO TO 17                                          EQUILB  0870
      WRITE  (6,59)                                                      EQUILB  0880
   59 FORMAT('0',4X,'I',4X,'IPL',3X,'ISG',2X,'LTYPE',2X,'INDGX',         EQUILB  0890
     *       87X,'CARDS G.6'/)                                           EQUILB  0900
      ICARD = 6                                                          EQUILB  0910
      DO  64  I=1,NCON                                                   EQUILB  0920
      JCARD = I                                                          EQUILB  0930
      READ   (5,60)    IPL(I),ISG(I),LTYPE(I),INDGX(I)                   EQUILB  0940
      WRITE  (6,61)  I,IPL(I),ISG(I),LTYPE(I),INDGX(I)                   EQUILB  0950
   60 FORMAT(4I4)                                                        EQUILB  0960
   61 FORMAT(5I6)                                                        EQUILB  0970
      IF  (  IPL(I).LT.1 .OR.   IPL(I).GT.NPL )  GO TO 65                EQUILB  09B0
      IF  (  ISG(I).LT.1 .OR.   ISG(I).GT.NSEG)  GO TO 65                EQUILB  0990
      IF  (LTYPE(I).LT.3 .OR. LTYPE(I).GT.4    )  GO TO 65               EQUILB  1000
```

```
      IF  (INDGX(I).LT.0 .OR. INDGX(I).GT.NVAR)  GO TO 65                EQUILB  1010
      J = IPL(I)                                                         EQUILB  1020
      NNPL = MNPL(J)                                                     EQUILB  1030
      IF  (NNPL.LT.1 .OR. NNPL.GT.5)  GO TO 65                           EQUILB  1040
      DO  62  K=1,NNPL                                                   EQUILB  1050
      IF  (ISG(I).NE.MPL(2,K,J))  GO TO 62                               EQUILB  1060
      ISG(I) = K                                                         EQUILB  1070
      GO TO 63                                                           EQUILB  1080
   62 CONTINUE                                                           EQUILB  1090
      GO TO 65                                                           EQUILB  1100
   63 IF  (INDGX(I).LE.0)  GO TO 64                                      EQUILB  1110
      K = INDGX(I)                                                       EQUILB  1120
      IF  (IPL(I).NE.JPL(K) .OR. ISG(I).NE.JSG(K))  GO TO 65             EQUILB  1130
   64 CONTINUE                                                           EQUILB  1140
      GO TO 17                                                           EQUILB  1150
C                                                                        EQUILB  1160
C     INPUT ERROR - PRINT MESSAGE AND TERMINATE PROGRAM.                 EQUILB  1170
C                                                                        EQUILB  1180
   65 WRITE  (6,66)  ICARD,JCARD                                         EQUILB  1190
   66 FORMAT('0  INPUT ERROR ON CARD G.',I2,'.',I2,                      EQUILB  1200
     *       '.  PROGRAM TERMINATED.')                                   EQUILB  1210
      STOP 26                                                            EQUILB  1220
C                                                                        EQUILB  1230
C     DATA INITIALIZATION.                                               EQUILB  1240
C                                                                        EQUILB  1250
   17 NQORG = NQ                                                         EQUILB  1260
      DO  19  K=1,NVAR                                                   EQUILB  1270
      J = JPL(K)                                                         EQUILB  1280
      I = JSG(K)                                                         EQUILB  1290
      M1(K) = MPL(1,I,J)                                                 EQUILB  1300
      M2(K) = MPL(2,I,J)                                                 EQUILB  1310
      M3(K) = MPL(3,I,J)                                                 EQUILB  1320
      MT(K) = NTPL (I,J)                                                 EQUILB  1330
      JX(K) = 1                                                          EQUILB  1340
      DXP(K) = 0.0                                                       EQUILB  1350
      I1 = NI1(K)                                                        EQUILB  1360
      I2 = NSG(K)                                                        EQUILB  1370
      IF  (NTV(K).EQ.1)  X(K) = SEGLP(I1,I2)                             EQUILB  1380
      IF  (NTV(K).EQ.2)  X(K) =   YPR(I1,I2)                             EQUILB  1390
      SX (K) =   X(K)                                                    EQUILB  1400
      SGX(K) =  GX(K)                                                    EQUILB  1410
      IF  (NAV(K).LE.0)  GO TO 19                                        EQUILB  1420
      IAV = NAV(K)                                                       EQUILB  1430
      DO  18  L=1,IAV                                                    EQUILB  1440
      J2 = KSG(L,K)                                                      EQUILB  1450
      IF  (NTV(K).EQ.1)  DPN(L,K) = SEGLP(I1,I2) - SEGLP(I1,J2)          EQUILB  1460
   18 IF  (NTV(K).EQ.2)  DPN(L,K) =   YPR(I1,I2) -   YPR(I1,J2)         EQUILB  1470
   19 CONTINUE                                                           EQUILB  1480
      IF  (NPRT(27).EQ.0)  GO TO 20                                      EQUILB  1490
C                                                                        EQUILB  1500
```

```
C     LET'S SEE WHAT USER INPUT LOOKS LIKE.                                  EQUILB  1510
C                                                                            EQUILB  1520
      CALL OUTPUT(0)                                                         EQUILB  1530
      CALL DAUX(0)                                                           EQUILB  1540
      CALL PRINT(6H USER )                                                   EQUILB  1550
      CALL OUTPUT(1)                                                         EQUILB  1560
C                                                                            EQUILB  1570
C     START FDF FORCE -> CONSTRAINT FORCE ITERATION                          EQUILB  1580
C                                                                            EQUILB  1590
   20 PENDOT = 0.0                                                           EQUILB  1600
      DO  50  JITTER=1,10                                                    EQUILB  1610
C                                                                            EQUILB  1620
C     ITERATE INPUT (X) SUCH THAT F(X) = G(X)                                EQUILB  1630
C                                                                            EQUILB  1640
      MVAR = 2                                                               EQUILB  1650
      IF  (NVAR.EQ.1)  MVAR = 1                                              EQUILB  1660
      DO  32  M=1,2                                                          EQUILB  1670
      DO  32  I=MVAR,NVAR                                                    EQUILB  1680
      DO  32  J=1,I                                                          EQUILB  1690
      NITER = 10                                                             EQUILB  1700
      IF  (DXP(J).EQ.0.0)  NITER = 50                                        EQUILB  1710
      DX(J) = 0.25                                                           EQUILB  1720
      N1 = M1(J)                                                             EQUILB  1730
      N2 = M2(J)                                                             EQUILB  1740
      N3 = M3(J)                                                             EQUILB  1750
      NP = JPL(J)                                                            EQUILB  1760
      NT = MT(J)                                                             EQUILB  1770
      I1 = NI1(J)                                                            EQUILB  1780
      I2 = NSG(J)                                                            EQUILB  1790
      IAV = NAV(J)                                                           EQUILB  1800
      IF  (NTV(J).NE.2)  GO TO 15                                            EQUILB  1810
      CALL DRCIJK (D,YPR,IYPR,HT,I2)                                         EQUILB  1820
      IF  (NAV(J).LE.0)  GO TO 15                                            EQUILB  1830
      DO  14  K=1,IAV                                                        EQUILB  1840
      J2 = KSG(K,J)                                                          EQUILB  1850
   14 CALL DRCIJK (D,YPR,IYPR,HT,J2)                                         EQUILB  1860
   15 DO  29  ITER=1,NITER                                                   EQUILB  1870
      CALL CHAIN                                                             EQUILB  1880
C                                                                            EQUILB  1890
C     SHORTENED VERSION OF PLELP.                                            EQUILB  1900
C                                                                            EQUILB  1910
      CALL DOTT33(D(1,1,N2),D(1,1,N1),DMNT)                                  EQUILB  1920
      DO  12  L=1,3                                                          EQUILB  1930
   12 XMN(L) = SEGLP(L,N2) - SEGLP(L,N1)                                     EQUILB  1940
      CALL MAT31(D(1,1,N2),XMN,XMM)                                          EQUILB  1950
      CALL MAT31(DMNT,PL(1,NP),TM)                                           EQUILB  1960
      BET = PL(4,NP)                                                         EQUILB  1970
      DO  13  L=1,3                                                          EQUILB  1980
   13 BET = BET - TM(L)*(BD(L+3,N3)+XMM(L))                                  EQUILB  1990
      BTS = XDY(TM,BD(16,N3),TM)                                             EQUILB  2000
```

```
      PEN1 = PEN                                                                EQUILB  2010
      PEN = BET + DSQRT(BTS)                                                    EQUILB  2020
      FX1(J) = FX(J)                                                            EQUILB  2030
      FXJ = 0.0                                                                 EQUILB  2040
      IF  (PEN.GT.0.0)  CALL FRCDFL (PEN,PENDOT,NT,1,FXJ,ELOSS)                 EQUILB  2050
      FX(J) = FXJ                                                               EQUILB  2060
      IF  (JX(J)-2)  23,21,25                                                   EQUILB  2070
   21 IF  (FX(J)*FX1(J).GT.0.0)  GO TO 22                                       EQUILB  2080
      IF  (FX1(J).EQ.0.0)  JX(J) = 1                                            EQUILB  2090
      FX(J) = FX1(J)                                                            EQUILB  2100
      PEN = PEN1                                                                EQUILB  2110
      DX(J) = 0.5*DX(J)                                                         EQUILB  2120
      X(J)  = X(J) - DX(J)                                                      EQUILB  2130
      GO TO 27                                                                  EQUILB  2140
   22 F2 = FX(J) - GX(J)                                                        EQUILB  2150
      F1 = FX1(J) - GX(J)                                                       EQUILB  2160
      IF  (F1*F2.LE.0.0)  GO TO 24                                              EQUILB  2170
      IF  (DABS(F2).LT.DABS(F1)) GO TO 23                                       EQUILB  2180
   26 FX(J) = FX1(J)                                                            EQUILB  2190
      DX(J) = -DX(J)                                                            EQUILB  2200
      PEN = PEN1                                                                EQUILB  2210
      X(J) = X(J) + 2.0*DX(J)                                                   EQUILB  2220
      GO TO 27                                                                  EQUILB  2230
   23 JX(J) = 1                                                                 EQUILB  2240
      IF  (PEN.GT.0.0)  JX(J) = 2                                               EQUILB  2250
      IF  (ITER.GT.1 .AND. PEN.LT.0.0 .AND. PEN.LT.PEN1)  GO TO 26              EQUILB  2260
      X(J) = X(J) + DX(J)                                                       EQUILB  2270
      GO TO 27                                                                  EQUILB  2280
   24 DXP(J) = DX(J)/(FX(J)-FX1(J))                                             EQUILB  2290
      JX(J) = 3                                                                 EQUILB  2300
   25 IF  (DABS(FX(J)-GX(J)).LT.EPS(6))  GO TO 30                               EQUILB  2310
      IF  (PEN.LT.0.0)  CALL FRCDFL (-PEN,PENDOT,NT,1,FXJ,ELOSS)                EQUILB  2320
      IF  (PEN.LT.0.0)  FX(J) = -FXJ                                            EQUILB  2330
      X(J) = X(J) - DXP(J)*(FX(J)-GX(J))                                        EQUILB  2340
   27 IF  (XDEV(J).LE.0.0)  GO TO 42                                            EQUILB  2350
      IF  (DABS(X(J)-SX(J)).LE.XDEV(J))  GO TO 42                               EQUILB  2360
      WRITE  (6,41)  J,X(J),SX(J),XDEV(J)                                       EQUILB  2370
   41 FORMAT('0 PROGRAM IS BEING TERMINATED IN SUBROUTINE EQUILB.'//            EQUILB  2380
     *       '  ITERATION FOR VARIABLE NO.',I3,' IS NOT CONVERGING.'//          EQUILB  2390
     *       '  VALUE OF X IS OUT OF RANGE. VALUES OF X,SX,XDEV ARE'//          EQUILB  2400
     *        3G20.8)                                                          EQUILB  2410
      STOP 27                                                                   EQUILB  2420
   42 IF  (NTV(J).EQ.1)  SEGLP(I1,I2) = X(J)                                    EQUILB  2430
      IF  (NTV(J).EQ.2)    YPR(I1,I2) = X(J)                                    EQUILB  2440
      IF  (NTV(J).EQ.2)  CALL DRCIJK (D,YPR,IYPR,HT,I2)                         EQUILB  2450
      IF  (NAV(J).LE.0)  GO TO 29                                               EQUILB  2460
      DO  28  K=1,IAV                                                           EQUILB  2470
      J2 = KSG(K,J)                                                             EQUILB  2480
      IF  (NTV(J).EQ.1)  SEGLP(I1,J2) = X(J) - DPN(K,J)                         EQUILB  2490
      IF  (NTV(J).EQ.2)    YPR(I1,J2) = X(J) - DPN(K,J)                         EQUILB  2500
```

```
   28  IF  (NTV(J).EQ.2)  CALL DRCIJK (D,YPR,IYPR,HT,J2)                    EQUILB  2510
   29  CONTINUE                                                             EQUILB  2520
   30  IF  (NPRT(27).NE.0)  WRITE  (6,31)  M,I,J,ITER,X(J),FX(J)            EQUILB  2530
   31  FORMAT(4I3,4X,2F12.6)                                                EQUILB  2540
   32  CONTINUE                                                             EQUILB  2550
       IF  (NQ.LE.0)  GO TO 40                                              EQUILB  2560
C                                                                           EQUILB  2570
C      COMPUTE VEHICLE COORDINATES FOR FIXED POINT CONSTRAINTS.             EQUILB  2580
C                                                                           EQUILB  2590
       DO  35  K=1,NQ                                                       EQUILB  2600
       IF  (KQTYPE(K).NE.1)  GO TO 35                                       EQUILB  2610
       IF  (KQ2(K).NE.NVEH)  GO TO 35                                       EQUILB  2620
       L = KQ1(K)                                                           EQUILB  2630
       CALL DOT31(D(1,1,L),RK1(1,K),T)                                      EQUILB  2640
       DO  34  I=1,3                                                        EQUILB  2650
   34  T(I) = T(I) + SEGLP(I,L) - SEGLP(I,NVEH)                             EQUILB  2660
       CALL MAT31(D(1,1,NVEH),T,RK2(1,K))                                   EQUILB  2670
   35  CONTINUE                                                             EQUILB  2680
   40  IF  (NPRT(27).EQ.0)  GO TO 36                                        EQUILB  2690
C                                                                           EQUILB  2700
C      SOLVE SYSTEM EQUATIONS WITH CONSTRAINTS OFF.                         EQUILB  2710
C                                                                           EQUILB  2720
       CALL OUTPUT(0)                                                       EQUILB  2730
       CALL DAUX(0)                                                         EQUILB  2740
       CALL PRINT(6HEQUIL2)                                                 EQUILB  2750
       CALL OUTPUT(1)                                                       EQUILB  2760
C                                                                           EQUILB  2770
C      SET UP CONSTRAINTS TO PRODUCE ZERO ACCELERATIONS.                    EQUILB  2780
C                                                                           EQUILB  2790
   36  NQ = NQORG                                                           EQUILB  2800
       IF  (NCON.LE.0)  GO TO 81                                            EQUILB  2810
       DO  37  I=1,NCON                                                     EQUILB  2820
       NQ = NQ+1                                                            EQUILB  2830
       J = IPL(I)                                                           EQUILB  2840
       K = ISG(I)                                                           EQUILB  2850
       NT = NTPL(K,J)                                                       EQUILB  2860
       NTNQ(I) = NTAB(NT+1)                                                 EQUILB  2870
       NTAB(NT+1) = -NQ                                                     EQUILB  2880
       KQ1(NQ) = MPL(2,K,J)                                                 EQUILB  2890
       KQ2(NQ) = MPL(1,K,J)                                                 EQUILB  2900
   37  KQTYPE(NQ) = LTYPE(I)                                                EQUILB  2910
C                                                                           EQUILB  2920
C      SOLVE SYSTEM EQUATIONS WITH CONSTRAINTS ON.                          EQUILB  2930
C                                                                           EQUILB  2940
       CALL OUTPUT(0)                                                       EQUILB  2950
       CALL DAUX(0)                                                         EQUILB  2960
       IF  (NPRT(27).NE.0.AND.JITTER.EQ.1)  CALL PRINT(6HEQUIL1)            EQUILB  2970
C                                                                           EQUILB  2980
C      FETCH CONSTRAINTS FORCES NORMAL TO PLANE SURFACES.                   EQUILB  2990
C      STORE FRICTION FORCE AND TURN OFF CONSTRAINTS.                       EQUILB  3000
```

```
C                                                                       EQUILB  3010
      CONV = 1.0                                                        EQUILB  3020
      DO  39  I=1,NCON                                                  EQUILB  3030
      MQ = NQORG+I                                                      EQUILB  3040
      J = IPL(I)                                                        EQUILB  3050
      K = ISG(I)                                                        EQUILB  3060
      NT = NTPL(K,J)                                                    EQUILB  3070
      NTAB(NT+1) = NTNQ(I)                                              EQUILB  3080
      M = MPL(2,K,J)                                                    EQUILB  3090
      N = MPL(1,K,J)                                                    EQUILB  3100
      CALL DOT31(D(1,1,N),PL(1,J),TEMP)                                 EQUILB  3110
      T(I) = TEMP(1)*QQ(1,MQ) + TEMP(2)*QQ(2,MQ)+ TEMP(3)*QQ(3,MQ)      EQUILB  3120
      I1 = INDGX(I)                                                     EQUILB  3130
      IF  (I1.GT.0 .AND. DABS(GX(I1)+T(I)).GT.EPS(2))  CONV = 0.0       EQUILB  3140
      IF  (I1.GT.0)  GX(I1) = 0.5*(GX(I1)-T(I))                         EQUILB  3150
      DO  38  L=1,3                                                     EQUILB  3160
 38   TEMP(L) = QQ(L,MQ) - T(I)*TEMP(L)                                 EQUILB  3170
      LT = NTAB(NT)                                                     EQUILB  3180
 39   CALL MAT31(D(1,1,M),TEMP,TAB(LT+19))                              EQUILB  3190
      NQ = NQORG                                                        EQUILB  3200
      IF  (CONV.EQ.1.0)  GO TO 81                                       EQUILB  3210
 50   CONTINUE                                                          EQUILB  3220
C                                                                       EQUILB  3230
C     PRINT INPUT AND CHANGES MADE.                                     EQUILB  3240
C                                                                       EQUILB  3250
 81   IF  (NJNT.LE.0)  GO TO 86                                         EQUILB  3260
      CALL OUTPUT(0)                                                    EQUILB  3270
      CALL DAUX(0)                                                      EQUILB  3280
      IPRINT = 0                                                        EQUILB  3290
      DO  84  J=1,NJNT                                                  EQUILB  3300
      IF  (IPIN(J).GE.0)  GO TO 84                                      EQUILB  3310
      IF  (VISC(4,3*J-2).GT.0.0)  GO TO 84                              EQUILB  3320
      IF  (IPIN(J).EQ.-1)  T1 = DABS(XDY(HB(1,2*J),D(1,1,J+1),TQ(1,J)))  EQUILB  3330
      IF  (IPIN(J).LE.-2)  T1 = DSQRT(TQ(1,J)**2+TQ(2,J)**2+TQ(3,J)**2)  EQUILB  3340
      VISC(4,3*J-2) = 1.5*T1                                            EQUILB  3350
      IF  (IPRINT.EQ.0)  WRITE  (6,82)                                  EQUILB  3360
 82   FORMAT('0  THE FOLLOWING VALUES FOR THE MAX TORQUE FOR A LOCKED JO  EQUILB  3370
     *INT ON CARDS B.5 HAVE BEEN SET UP BY SUBROUTINE EQUILB:'//         EQUILB  3380
     *       '     J SYM    IPIN    T1=VISC(4)' /)                      EQUILB  3390
      IPRINT = 1                                                        EQUILB  3400
      WRITE  (6,83)  J,JOINT(J),IPIN(J),VISC(4,3*J-2)                   EQUILB  3410
 83   FORMAT(I6,1X,A4,I6,F15.6)                                         EQUILB  3420
 84   CONTINUE                                                          EQUILB  3430
 86   IF  (NQ.LE.0)  GO TO 91                                           EQUILB  3440
      IPRINT = 0                                                        EQUILB  3450
      DO  89  K=1,NQ                                                    EQUILB  3460
      IF  (KQTYPE(K).NE.1)  GO TO 89                                    EQUILB  3470
      IF  (KQ2(K).NE.NVEH)  GO TO 89                                    EQUILB  3480
      IF  (IPRINT.EQ.0)  WRITE  (6,87)                                  EQUILB  3490
 87   FORMAT('0  THE FOLLOWING VALUES FOR RK2 ON CARDS D.6 FOR FIXED POI  EQUILB  3500
```

```
     *NT CONSTRAINTS HAVE BEEN CHANGED BY SUBROUTINE EQUILB:'//          EQUILB  3510
     *       5X,'K',3X,'KQTYPE',4X,'KQ1',5X,'KQ2',8X,'RK2(X)',          EQUILB  3520
     *       9X,'RK2(Y)',9X,'RK2(Z)'/)                                  EQUILB  3530
      IPRINT = 1                                                        EQUILB  3540
      WRITE  (6,88)  K,KQTYPE(K),KQ1(K),KQ2(K),(RK2(I,K),I=1,3)        EQUILB  3550
   88 FORMAT(I6,3I8,3F15.6)                                             EQUILB  3560
   89 CONTINUE                                                          EQUILB  3570
   91 WRITE  (6,92)                                                     EQUILB  3580
   92 FORMAT('0    THE FOLLOWING VARIABLES ON CARDS G.2 AND G.3 ',      EQUILB  3590
     *       'HAVE BEEN CHANGED BY SUBROUTINE EQUILB:'//)               EQUILB  3600
      DO  95  J=1,NVAR                                                  EQUILB  3610
      I0 = NTV(J)                                                       EQUILB  3620
      I1 = NI1(J)                                                       EQUILB  3630
      I2 = NSG(J)                                                       EQUILB  3640
      WRITE  (6,93)  WORD(I0),I1,I2,SX(J),X(J),BLANK,J,SGX(J),GX(J)     EQUILB  3650
   93 FORMAT(4X,A6,'(',I2,',',I2,') FROM',F12.6,' TO',F12.6,            EQUILB  3660
     *          A4,'AND GX(',I2,') FROM',F12.6,' TO',F12.6)            EQUILB  3670
      IF  (NAV(J).LE.0)  GO TO 95                                       EQUILB  3680
      IAV = NAV(J)                                                      EQUILB  3690
      DO  94  I=1,IAV                                                   EQUILB  3700
      J2 = KSG(I,J)                                                     EQUILB  3710
      ZSX = SX(J) - DPN(I,J)                                            EQUILB  3720
      ZXX =  X(J) - DPN(I,J)                                            EQUILB  3730
   94 WRITE  (6,93)  WORD(I0),I1,J2,ZSX,ZXX                             EQUILB  3740
   95 CONTINUE                                                          EQUILB  3750
      CALL ELTIME (2,35)                                                EQUILB  3760
      RETURN                                                            EQUILB  3770
      END                                                               EQUILB  3780
```

```
      SUBROUTINE EULRAD(D,A,IC)                                          EULRAD  0010
C                                                          REV 20 05/02/80 EULRAD  0020
C     COMPUTES EULER ANGLES PRECESSION, NUTATION, AND SPIN IN RADIANS     EULRAD  0030
C     AND PLACES THEM INTO THE A ARRAY FOR GIVEN DIRECTION COSINE MATRIX  EULRAD  0040
C                                                                         EULRAD  0050
C       ASSUMES  D = D(S)D(N)D(P) , WHERE                                 EULRAD  0060
C                                                                         EULRAD  0070
C            CS SS 0           1  0  0           CP SP 0                  EULRAD  0080
C     D(S)=-SS CS 0  , D(N)= 0  CN SN  , D(P)=-SP CP 0                    EULRAD  0090
C            0  0  1           0 -SN CN           0  0  1                 EULRAD  0100
C                                                                         EULRAD  0110
C     AND P=A(1),  N=A(2), S=A(3)                                         EULRAD  0120
C                                                                         EULRAD  0130
C     ROUTINE WILL ALWAYS WORK IN THE MEMORY MODE, I.E., WILL PRODUCE A   EULRAD  0140
C     NEW SET OF A'S THAT DIFFER THE LEAST FROM THE INPUTTED A ARRAY.     EULRAD  0150
C     TO USE IN NON-MEMORY MODE,SET ALL A'S TO ZERO, CALL WITH IC = 8.    EULRAD  0160
C                                                                         EULRAD  0170
C     NEW N IS ALWAYS COMPUTED.                                           EULRAD  0180
C     IF N OR PI-N < 10**-6, IC IS USED TO RESOLVE AMBIGUITES ON P & S,   EULRAD  0190
C     EXCEPT FOR IC = 2 OR 8 WHERE THEY ARE NOT CHANGED.                  EULRAD  0200
C                                                                         EULRAD  0210
      IMPLICIT REAL*8(A-H,O-Z)                                            EULRAD  0220
      DIMENSION A(3),D(3,3),T(6)                                          EULRAD  0230
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                           EULRAD  0240
     *               UNITL,UNITM,UNITT,GRAVTY(3)                        EULRAD  0250
      TWOPI = 2.0*PI                                                      EULRAD  0260
      IF  (D(3,3).GT. 1.0)  D(3,3) =  1.0                                 EULRAD  0270
      IF  (D(3,3).LT.-1.0)  D(3,3) = -1.0                                 EULRAD  0280
      B = DARCOS(D(3,3))                                                  EULRAD  0290
      T(2) =  B-A(2)                                                      EULRAD  0300
      T(5) = -B-A(2)                                                      EULRAD  0310
      Z = 0.0                                                             EULRAD  0320
      IF  (   B.LT.EPS(6)) Z =  1.0                                       EULRAD  0330
      IF  (PI-B.LT.EPS(6)) Z = -1.0                                       EULRAD  0340
      IF  (Z.NE.0.0)  GO TO 11                                            EULRAD  0350
      T(1) = DATAN2(D(3,1),-D(3,2)) - A(1)                                EULRAD  0360
      T(4) = T(1) + PI                                                    EULRAD  0370
      T(3) = DATAN2(D(1,3), D(2,3)) - A(3)                                EULRAD  0380
      T(6) = T(3) + PI                                                    EULRAD  0390
      GO TO 26                                                            EULRAD  0400
   11 T(1) = DATAN2(D(1,2)-Z*D(2,1) , D(1,1)+Z*D(2,2))- A(1) - Z*A(3)     EULRAD  0410
      T(3) = T(1)                                                         EULRAD  0420
      GO TO (21,22,23,23,22,21,22,22) , IC                                EULRAD  0430
C                                                                         EULRAD  0440
C     SET T(1) = 0 EXCEPT FOR IC=3,4                                      EULRAD  0450
C     SET T(3) = 0 EXCEPT FOR IC=1,6                                      EULRAD  0460
C                                                                         EULRAD  0470
   21 T(1) = 0.0                                                          EULRAD  0480
      GO TO 25                                                            EULRAD  0490
   22 T(1) = 0.0                                                          EULRAD  0500
```

```
23  T(3) = 0.0                                                                    EULRAD  0510
25  T(4) = T(1)                                                                   EULRAD  0520
    T(6) = T(3)                                                                   EULRAD  0530
26  TMAX = 0.0                                                                    EULRAD  0540
    J    = 3                                                                      EULRAD  0550
    DO  30  I=1,6                                                                 EULRAD  0560
    T(I) = DMOD(T(I),TWOPI)                                                       EULRAD  0570
    IF  (DABS(T(I)).GT.PI   )  T(I) = T(I) - DSIGN(TWOPI,T(I))                    EULRAD  0580
    IF  (DABS(T(I)).LT.TMAX)  GO TO 30                                            EULRAD  0590
    TMAX = DABS(T(I))                                                             EULRAD  0600
    IF  (I.GT.3)  J = 0                                                           EULRAD  0610
30  CONTINUE                                                                      EULRAD  0620
    IF  (Z.LT.0.0)  T(J+3) = -T(J+3)                                              EULRAD  0630
    DO  40  I=1,3                                                                 EULRAD  0640
    IJ = I+J                                                                      EULRAD  0650
40  A(I) = A(I) + T(IJ)                                                           EULRAD  0660
    RETURN                                                                        EULRAD  0670
    END                                                                           EULRAD  0680
```

```
      DOUBLE PRECISION FUNCTION EVALFD (D,N,L)                          EVALFD  0010
C                                                      REV 20 04/30/80  EVALFD  0020
C     EVALUATE FUNCTION THAT IS DEFINED AT LOCATION N OF TAB ARRAY      EVALFD  0030
C     FOR ABSCISSA VALUE D.  EVALUATES DERIVATIVE, FUNCTION OR INTEGRAL EVALFD  0040
C     AS L EQUALS 0, 1, OR 2. TAB ARRAY IS DEFINED AS FOLLOWS:          EVALFD  0050
C        TAB(N)    -  D0 (D0 MUST BE NON-NEGATIVE)                     EVALFD  0060
C        TAB(N+1)  -  D1 (F1 DEFINED FOR D0 < D < D1)                  EVALFD  0070
C        TAB(N+2)  -  D2 (F2 DEFINED FOR D1 < D < D2)                  EVALFD  0080
C        TAB(N+3)  -  (NOT CURRENTLY USED)                             EVALFD  0090
C        TAB(N+4)  -  (NOT CURRENTLY USED)                             EVALFD  0100
C        TAB(N+5)  -  START OF DEFINITION OF 1ST PART OF FUNCTION (F1) EVALFD  0110
C        WHICH IS FOLLOWED BY DEFINITION OF 2ND PART OF FUNCTION (F2),  EVALFD  0120
C                              IF ANY.                                  EVALFD  0130
C        2ND PART OF FUNCTION EXISTS IF D2 IS NON-ZERO.                EVALFD  0140
C        SIGN OF D1 DETERMINES FORM OF DEFINITION FOR 1ST PART OF      EVALFD  0150
C                              THE FUNCTION.                            EVALFD  0160
C                                                                       EVALFD  0170
C             D1 ZERO INDICATES THAT FUNCTION IS CONSTANT D2 FOR ALL D. EVALFD  0180
C                                                                       EVALFD  0190
C             D1 POSITIVE INDICATES THAT TAB(N+5)-TAB(N+10) CONTAINS    EVALFD  0200
C             A0,A1,...A5.   THE COEFFICIENTS OF A 5TH ORDER POLYNOMIAL.EVALFD  0210
C                                                                       EVALFD  0220
C             D1 NEGATIVE INDICATES THAT TAB(N+5) CONTAINS NP (REAL)    EVALFD  0230
C             FOLLOWED BY   D(1), F(1), D(2), F(2) ..., D(NP), F(NP)    EVALFD  0240
C                                                                       EVALFD  0250
C             WARNING- TABULAR FUNCTION MUST BE DEFINED FOR WHOLE RANGE,EVALFD  0260
C             THAT IS, FROM D0 TO D1 INCLUSIVE,OR D1 TO D2 INCLUSIVE.   EVALFD  0270
C                                                                       EVALFD  0280
C                                                                       EVALFD  0290
C        SIMILARLY, THE SIGN OF D2 (IF NON-ZERO) DETERMINES FORM OF     EVALFD  0300
C        DEFINITION OF 2ND PART OF FUNCTION, IF ANY.                    EVALFD  0310
C                                                                       EVALFD  0320
C                                                                       EVALFD  0330
C     IF D < D0                    FUNCTION = 0                       EVALFD  0340
C     IF D > |D1|  AND  D2=0       FUNCTION = F1(|D1|)                EVALFD  0350
C     IF D > |D2|  AND  D2#0       FUNCTION = F2(|D2|)                EVALFD  0360
C                                                                       EVALFD  0370
      IMPLICIT REAL*8(A-H,O-Z)                                          EVALFD  0380
      COMMON/TABLES/ MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(500),TAB(2600)EVALFD  0390
      F = 0.0                                                           EVALFD  0400
      IOUTR = 0                                                         EVALFD  0410
      D0 = TAB(N)                                                       EVALFD  0420
      D1 = TAB(N+1)                                                     EVALFD  0430
      D2 = TAB(N+2)                                                     EVALFD  0440
      IF  (D0.EQ.0.0 .AND. D1.EQ.0.0)  GO TO 23                         EVALFD  0450
      IF  (D.LT.D0)  GO TO 40                                           EVALFD  0460
      IF  (D1.NE.0.0)  GO TO 26                                         EVALFD  0470
   23 IF  (L-1)  40,24,25                                               EVALFD  0480
   24 F = D2                                                            EVALFD  0490
      GO TO 40                                                          EVALFD  0500
```

```
   25 F= (D-D0)*D2                                                      EVALFD  0510
      GO TO 40                                                          EVALFD  0520
C                                                                       EVALFD  0530
C     COMPUTE INDEX OF F1 DEFINITION                                    EVALFD  0540
C                                                                       EVALFD  0550
   26 NP = N+5                                                          EVALFD  0560
      IF (L.EQ.2) GO TO 41                                              EVALFD  0570
C                                                                       EVALFD  0580
C    DERIVATIVES AND FUNCTIONS HERE, INTEGRALS HAVE OTHER LOGIC         EVALFD  0590
C                                                                       EVALFD  0600
      IF (D.LT.DABS(D1)) GO TO 31                                       EVALFD  0610
      IF (D2.NE.0.0) GO TO 32                                           EVALFD  0620
C                                                                       EVALFD  0630
C     D .GE.|D1| , D2 = 0                                               EVALFD  0640
C                                                                       EVALFD  0650
   30 IF (D1.LE.0.0) GO TO 33                                           EVALFD  0660
C                                                                       EVALFD  0670
C     IOUTR.EQ.1 INDICATES D BEYOND RANGE. DERIVATIVE = 0.              EVALFD  0680
C     IOUTR.EQ.0 INDICATES D.LE.|D1|. COMPUTE POLY DERIVATIVE           EVALFD  0690
C                                                                       EVALFD  0700
      IF (D.GT.DABS(D1)) IOUTR = 1                                      EVALFD  0710
      X = D1                                                            EVALFD  0720
      GO TO 37                                                          EVALFD  0730
C                                                                       EVALFD  0740
C     D0 < D < |D1|                                                     EVALFD  0750
C                                                                       EVALFD  0760
   31 IF (D1.LT.0.0) GO TO 35                                           EVALFD  0770
      X = D                                                             EVALFD  0780
      GO TO 37                                                          EVALFD  0790
C                                                                       EVALFD  0800
C     D .GE. |D1|,  D2 NON-ZERO,  USE F2                                EVALFD  0810
C                                                                       EVALFD  0820
   32 MP = 6                                                            EVALFD  0830
C                                                                       EVALFD  0840
C     COMPUTE INDEX OF F2 DEFINITION                                    EVALFD  0850
C                                                                       EVALFD  0860
      IF (D1.LT.0.0) MP = 2.0 * TAB(NP)+1.0                             EVALFD  0870
      NP = NP+MP                                                        EVALFD  0880
      IF (D.LT.DABS(D2)) GO TO 34                                       EVALFD  0890
   29 IF (D2.LT.0.0) GO TO 33                                           EVALFD  0900
C                                                                       EVALFD  0910
C     IOUTR.EQ.1 INDICATES D BEYOND RANGE. DERIVATIVE = 0.              EVALFD  0920
C     IOUTR.EQ.0 INDICATES D.LE.|D2|. COMPUTE POLY DERIVATIVE           EVALFD  0930
C                                                                       EVALFD  0940
      IF (D.GT.DABS(D2)) IOUTR = 1                                      EVALFD  0950
C                                                                       EVALFD  0960
C     D .GE. D2 (POSITIVE), EVALUATE F2 FOR D2                          EVALFD  0970
C                                                                       EVALFD  0980
      X = D2                                                            EVALFD  0990
      GO TO 37                                                          EVALFD  1000
```

```
C                                                                       EVALFD  1010
C     D EXCEEDS TABULAR DEFINITION, SET F = F(NP)                       EVALFD  1020
C     IF TABLE DEFINITION EXTENDS BEYOND RANGE, USE TABLE VALUES        EVALFD  1030
C                                                                       EVALFD  1040
   33 MB = TAB(NP)                                                      EVALFD  1050
      NB = NP+MB+MB                                                     EVALFD  1060
      IF (D .LE. TAB(NB-1)) GO TO 35                                    EVALFD  1070
      IF (L.EQ.1) F=TAB(NB)                                             EVALFD  1080
      GO TO 40                                                          EVALFD  1090
C                                                                       EVALFD  1100
C     |D1| .LE. D < |D2|                                               EVALFD  1110
C                                                                       EVALFD  1120
   34 IF (D2.LT.0.0) GO TO 35                                           EVALFD  1130
      X = D                                                             EVALFD  1140
      GO TO 37                                                          EVALFD  1150
C                                                                       EVALFD  1160
C     EVALUATE F FROM TABULAR DEFINITION                                EVALFD  1170
C                                                                       EVALFD  1180
   35 MB = TAB(NP)                                                      EVALFD  1190
      K1 = NP+3                                                         EVALFD  1200
      K2 = NP+MB+MB                                                     EVALFD  1210
      DO 36 K=K1,K2,2                                                   EVALFD  1220
      IF (D.GT.TAB(K)) GO TO 36                                         EVALFD  1230
      IF (L-1) 28,27,40                                                 EVALFD  1240
C                                                                       EVALFD  1250
C     EVALUATE DERIVATIVE FROM TABLE                                    EVALFD  1260
C                                                                       EVALFD  1270
  28   F = (TAB(K+1)-TAB(K-1))/(TAB(K)-TAB(K-2))                        EVALFD  1280
      GO TO 40                                                          EVALFD  1290
C                                                                       EVALFD  1300
C     EVALUATE FUNCTION FROM TABLE.                                     EVALFD  1310
C                                                                       EVALFD  1320
  27  R2 = TAB(K)-TAB(K-2)                                              EVALFD  1330
      R1 = (D-TAB(K-2))/R2                                              EVALFD  1340
      R2 = (TAB(K)-D )/R2                                               EVALFD  1350
      F = R1*TAB(K+1)+R2*TAB(K-1)                                       EVALFD  1360
      GO TO 40                                                          EVALFD  1370
   36 CONTINUE                                                          EVALFD  1380
      IF (L.EQ.1) F = TAB(K2)                                           EVALFD  1390
      GO TO 40                                                          EVALFD  1400
   37 IF (IOUTR.EQ.1 .AND. L.EQ.0 ) GO TO 40                            EVALFD  1410
      IF (L-1) 38,39,40                                                 EVALFD  1420
C                                                                       EVALFD  1430
C     EVALUATE DERIVATIVE OF 5TH DEGREE POLYNOMIAL                      EVALFD  1440
C                                                                       EVALFD  1450
  38  F = TAB(NP+1)+X*(2.0*TAB(NP+2)+X*(3.0*TAB(NP+3)+X*(4.0*TAB(NP+4)+ EVALFD  1460
     *       X*5.0*TAB(NP+5))))                                         EVALFD  1470
      GO TO 40                                                          EVALFD  1480
C                                                                       EVALFD  1490
C     EVALUATE 5TH DEGREE POLYNOMIAL                                    EVALFD  1500
```

```
C                                                                       EVALFD  1510
   39 F =      TAB(NP) + X*(TAB(NP+1)+X*(TAB(NP+2)                      EVALFD  1520
     *    +X*(TAB(NP+3)+X*(TAB(NP+4)+X*TAB(NP+5)))))                   EVALFD  1530
      GO TO 40                                                          EVALFD  1540
C                                                                       EVALFD  1550
C     L=2: COMPUTE INTEGRAL OF FUNCTION FROM D0 TO D.                   EVALFD  1560
C                                                                       EVALFD  1570
   41 IF (D.EQ.D0) GO TO 40                                             EVALFD  1580
      X0 = D0                                                           EVALFD  1590
      X1 = D1                                                           EVALFD  1600
      DO 50 I=1,2                                                       EVALFD  1610
      IF (X1) 43,49,42                                                  EVALFD  1620
   42 A0 = TAB(NP   )                                                   EVALFD  1630
      A1 = TAB(NP+1)/2.0                                                EVALFD  1640
      A2 = TAB(NP+2)/3.0                                                EVALFD  1650
      A3 = TAB(NP+3)/4.0                                                EVALFD  1660
      A4 = TAB(NP+4)/5.0                                                EVALFD  1670
      A5 = TAB(NP+5)/6.0                                                EVALFD  1680
      NP = NP+6                                                         EVALFD  1690
      X = X0                                                            EVALFD  1700
      IF (X.NE.0.0) F=F-X*(A0+X*(A1+X*(A2+X*(A3+X*(A4+X*A5)))))         EVALFD  1710
      X = DMIN1(D,X1)                                                   EVALFD  1720
      IF (X.NE.0.0) F=F+X*(A0+X*(A1+X*(A2+X*(A3+X*(A4+X*A5)))))         EVALFD  1730
      IF(D.LE.X1)  GO TO 40                                             EVALFD  1740
      IF(I.EQ.1.AND.D2.NE.0.0)  GO TO 49                                EVALFD  1750
C                                                                       EVALFD  1760
C     NOTE -  NP WAS UPDATED NP=NP+6 BEFORE THIS, READY FOR SECOND PASS EVALFD  1770
C                                                                       EVALFD  1780
      F = F + (D-X1)*(TAB(NP-6)+X1*(TAB(NP-5)+X1*(TAB(NP-4)             EVALFD  1790
     *          +X1*(TAB(NP-3)+X1*(TAB(NP-2)+X1*TAB(NP-1))))))         EVALFD  1800
      GO TO 40                                                          EVALFD  1810
   43 MB = TAB(NP)                                                      EVALFD  1820
      K1 = NP+3                                                         EVALFD  1830
      K2 = NP+MB+MB                                                     EVALFD  1840
      NP = K2+1                                                         EVALFD  1850
      DL = DMIN1(D,DABS(X1))                                            EVALFD  1860
      DO 44 K=K1,K2,2                                                   EVALFD  1870
      IF (X0.GE.TAB(K)) GO TO 44                                        EVALFD  1880
      Z1 = DMAX1(X0,TAB(K-2))                                           EVALFD  1890
      Z2 = DMIN1(DL,TAB(K))                                             EVALFD  1900
      FYX = TAB(K-1)*TAB(K) - TAB(K+1)*TAB(K-2)                         EVALFD  1910
      FY = TAB(K+1) - TAB(K-1)                                          EVALFD  1920
      F = F +(FYX  + 0.5*FY*(Z1+Z2)) *(Z2-Z1)/ (TAB(K)-TAB(K-2))        EVALFD  1930
      IF (Z2.NE.DL) GO TO 44                                            EVALFD  1940
      IF(I.EQ.1.AND.D2.NE.0.0)  GO TO 49                                EVALFD  1950
      IF(Z2. EQ. D) GO TO 40                                            EVALFD  1960
      F = F +(D-Z2)*(FYX+Z2*FY)/ (TAB(K)-TAB(K-2))                      EVALFD  1970
      GO TO 40                                                          EVALFD  1980
   44 CONTINUE                                                          EVALFD  1990
   49 X0 = DABS(D1)                                                     EVALFD  2000

   50 X1 = D2                                                           EVALFD  2010
   40 EVALFD = F                                                        EVALFD  2020
      RETURN                                                            EVALFD  2030
      END                                                               EVALFD  2040
```

```
      SUBROUTINE FDINIT                                                 FDINIT  0010
C                                                          REV 20 05/14/80  FDINIT  0020
C     REPLACES CODE PREVIOUSLY IN SUBROUTINES FINPUT AND HINPUT.        FDINIT  0030
C     FROM FIVE FUNCTION NUMBERS IN NF ARRAY                            FDINIT  0040
C      1. SET UP KTITLE                                                 FDINIT  0050
C      2. SET UP NTAB AND TAB ARRAYS                                    FDINIT  0060
C      3. INCREMENT COUNTERS MXNTB AND MXTB2                            FDINIT  0070
C                                                                       FDINIT  0080
      IMPLICIT REAL*8 (A-H,O-Z)                                         FDINIT  0090
      COMMON/TABLES/ MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(500),TAB(2600)FDINIT  0100
      COMMON/TEMPVS/ JTITLE(5,51),NF(5),MS(3),KTITLE(31)                FDINIT  0110
C     NOTE: THIS IS SHARED BY SUBS CINPUT, FINPUT, HINPUT AND FDINIT.   FDINIT  0120
      REAL JTITLE,KTITLE                                                FDINIT  0130
      J1 = MXTB2 + 1                                                    FDINIT  0140
      NT = MXNTB + 1                                                    FDINIT  0150
      NTAB(NT) = J1                                                     FDINIT  0160
      NT = NT+1                                                         FDINIT  0170
      DO 56 L=1,5                                                       FDINIT  0180
      NX = IABS(NF(L))                                                  FDINIT  0190
      NTAB(NT) = 0                                                      FDINIT  0200
      IF (NX.EQ.0) GO TO 56                                             FDINIT  0210
      NTAB(NT) = ISIGN(NTI(NX),NF(L))                                   FDINIT  0220
      DO 51 KK = 1,5                                                    FDINIT  0230
      KJ = 5*L+KK+1                                                     FDINIT  0240
   51 KTITLE(KJ) = JTITLE(KK,NX)                                        FDINIT  0250
      IF (NTI(NX).NE.0) GO TO 56                                        FDINIT  0260
      WRITE(6,54) NX                                                    FDINIT  0270
   54 FORMAT ('0 FUNCTION NO.',I4,' HAS NOT BEEN DEFINED. ',            FDINIT  0280
     *        '  PROGRAM TERMINATED.')                                 FDINIT  0290
      STOP 15                                                           FDINIT  0300
   56 NT = NT+1                                                         FDINIT  0310
C                                                                       FDINIT  0320
C     INITIALIZE TAB ARRAY TO ZERO EXCEPT FOR DMAX, DINER, FDMAX.       FDINIT  0330
C                                                                       FDINIT  0340
      J2 = J1+29                                                        FDINIT  0350
      DO 57 JJ=J1,J2                                                    FDINIT  0360
   57 TAB(JJ) = 0.0                                                     FDINIT  0370
      NX = NTAB(NT-5)                                                   FDINIT  0380
      IF (NX.LT.0) GO TO 58                                             FDINIT  0390
      TAB(J1+8) = DABS(TAB(NX+1))                                       FDINIT  0400
      IF (TAB(NX+2).NE.0.0) TAB(J1+8) = DABS(TAB(NX+2))                 FDINIT  0410
      DX = TAB(J1+8)                                                    FDINIT  0420
      TAB(J1+10) = EVALFD(DX,NX,1)                                      FDINIT  0430
      NX = NTAB(NT-4)                                                   FDINIT  0440
      IF (NX.LE.0) GO TO 58                                             FDINIT  0450
      TAB(J1+9) = DABS(TAB(NX+1))                                       FDINIT  0460
      IF (TAB(NX+2).NE.0.0) TAB(J1+9) = DABS(TAB(NX+2))                 FDINIT  0470
   58 J1 = J2+1                                                         FDINIT  0480
      MXNTB = NT-1                                                      FDINIT  0490
      MXTB2 = J1-1                                                      FDINIT  0500
      IF  (MXTB2.GT.2600)  WRITE (6,62) MXTB2                           FDINIT  0510
   62 FORMAT ('0 ERROR IN SUBROUTINE FDINIT, SIZE OF TAB ARRAY =',I8//  FDINIT  0520
     *        '  PROGRAM TERMINATED.')                                 FDINIT  0530
      IF (MXNTB.GT.500)  WRITE (6,63) MXNTB                             FDINIT  0540
   63 FORMAT ('0 ERROR IN SUBROUTINE FDINIT, SIZE OF NTAB ARRAY =',I8// FDINIT  0550
     *        '  PROGRAM TERMINATED.')                                 FDINIT  0560
      IF  (MXTB2.GT.2600 .OR. MXNTB.GT.500)  STOP 16                    FDINIT  0570
      RETURN                                                            FDINIT  0580
      END                                                               FDINIT  0590
```

```
      SUBROUTINE FINPUT                                                 FINPUT  0010
C                                                          REV 20 05/27/80  FINPUT  0020
C     INPUT CARDS F.1-F.5 SPECIFYING THE ALLOWED CONTACTS OF THE CRASH     FINPUT  0030
C     VICTIM BODY SEGMENTS WITH VEHICLE PANELS, BELTS, AIRBAGS AND OTHER   FINPUT  0040
C     BODY SEGMENTS ALONG WITH THE ASSOCIATED FUNCTIONS TO BE USED FOR     FINPUT  0050
C     EACH CONTACT.                                                        FINPUT  0060
C     ALSO SETS UP TABLES TO CONTROL TIME HISTORY INFORMATION FOR          FINPUT  0070
C     EACH FUNCTION FOR EACH ALLOWED CONTACT.                              FINPUT  0080
C                                                                          FINPUT  0090
      IMPLICIT REAL*8(A-H,O-Z)                                           FINPUT  0100
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,            FINPUT  0110
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)         FINPUT  0120
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(3,60),HA(3,60),HB(3,60),  FINPUT  0130
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),      FINPUT  0140
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)     FINPUT  0150
      COMMON/JBARTZ/ MNPL(   30),MNBLT(    8),MNSEG(   30),MNBAG(    6),FINPUT  0160
     *               MPL(3,5,30),MBLT(3,5,8),MSEG(3,5,30),MBAG(3,10,6),  FINPUT  0170
     *               NTPL( 5,30),NTBLT( 5,8),NTSEG( 5,30)                FINPUT  0180
      COMMON/TABLES/ MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(500),TAB(2600) FINPUT  0190
      COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),            FINPUT  0200
     *               BLTTTL(5,8),PLTTL(5,30),BAGTTL(5,6),SEG(30),        FINPUT  0210
     *               JOINT(30),CGS(30),JS(30)                            FINPUT  0220
      REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT       FINPUT  0230
      LOGICAL*1 CGS,JS                                                   FINPUT  0240
      COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),    FINPUT  0250
     *               HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12), FINPUT  0260
     *               RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),               FINPUT  0270
     *               KQ1(12),KQ2(12),KQTYPE(12)                          FINPUT  0280
      COMMON/WINDFR/ WTIME(30),QFU(3,5),QFV(3,5),                        FINPUT  0290
     *               IWIND(30),MWSEG(5,30),NFVSEG(6),NFVNT(5)            FINPUT  0300
      COMMON/TEMPVS/JTITLE(5,51),NF(5),MS(3),KTITLE(31)                  FINPUT  0310
C                                                                        FINPUT  0320
      REAL  JTITLE,KTITLE,BLANK,SURFCE(2,3)                              FINPUT  0330
      DATA  BLANK/4H    /                                                FINPUT  0340
      DATA  SURFCE/4H  PL,4HANE ,4H  BE,4HLT  ,4H SEG,4HMENT/            FINPUT  0350
C                                                                        FINPUT  0360
      MXNTI = 50                                                         FINPUT  0370
      MXNTB = 0                                                          FINPUT  0380
      MXTB2 = MXTB1                                                      FINPUT  0390
C                                                                        FINPUT  0400
C     INPUT ALLOWED CONTACTS AND FUNCTIONS BY REF. NO.                   FINPUT  0410
C                                                                        FINPUT  0420
      WRITE (6,31)                                                       FINPUT  0430
   31 FORMAT('1 ALLOWED CONTACTS AND ASSOCIATED FUNCTIONS')              FINPUT  0440
      DO 61 I=1,4                                                        FINPUT  0450
      IJK = 0                                                            FINPUT  0460
      GO TO (32,34,35,36),I                                              FINPUT  0470
   32 IF (NPL.LE.0) GO TO 61                                             FINPUT  0480
C                                                                        FINPUT  0490
C     INPUT NO. OF SEGMENTS TO CONTACT EACH PLANE.                       FINPUT  0500
```

```
C     INPUT CARD F.1.A                                                    FINPUT  0510
C                                                                         FINPUT  0520
      READ  (5,33) (MNPL(J),J=1,NPL)                                      FINPUT  0530
   33 FORMAT(18I4)                                                        FINPUT  0540
      NJJ = NPL                                                           FINPUT  0550
      GO TO 37                                                            FINPUT  0560
   34 IF (NBLT.LE.0) GO TO 61                                             FINPUT  0570
C                                                                         FINPUT  0580
C     INPUT NO. OF SEGMENTS TO CONTACT EACH BELT.                         FINPUT  0590
C     INPUT CARD F.2.A                                                    FINPUT  0600
C                                                                         FINPUT  0610
      READ  (5,33) (MNBLT(J),J=1,NBLT)                                    FINPUT  0620
      NJJ = NBLT                                                          FINPUT  0630
      GO TO 37                                                            FINPUT  0640
   35 IF (NSEG.LE.0) GO TO 61                                             FINPUT  0650
C                                                                         FINPUT  0660
C     INPUT NO. OF SEGMENTS TO CONTACT EACH SEGMENT.                      FINPUT  0670
C     INPUT CARD F.3.A                                                    FINPUT  0680
C                                                                         FINPUT  0690
      READ  (5,33) (MNSEG(J),J=1,NSEG)                                    FINPUT  0700
      NJJ = NSEG                                                          FINPUT  0710
      NSEG1 = NSEG+1                                                      FINPUT  0720
      DO  26  J=NSEG1,NGRND                                               FINPUT  0730
   26 MNSEG(J) = 0                                                        FINPUT  0740
      GO TO 37                                                            FINPUT  0750
   36  IF (NJNT.LE.0) GO TO 61                                            FINPUT  0760
C                                                                         FINPUT  0770
C     INPUT CARD F.4.A                                                    FINPUT  0780
C     SUPPLY IGLOB(J)=1 FOR EACH GLOBALGRAPHIC JOINT J=1,NJNT             FINPUT  0790
C                                                                         FINPUT  0800
       READ (5,33) (IGLOB(J),J=1,NJNT)                                    FINPUT  0810
      NJJ = NJNT                                                          FINPUT  0820
C                                                                         FINPUT  0830
C     START OF LOOP TO READ CONTACTS FOR PLANES (I=1), BELTS (I=2),       FINPUT  0840
C     SEGMENTS (I=3) AND FUNCTIONS FOR GLOBALGRAPHIC JOINTS (I=4).        FINPUT  0850
C                                                                         FINPUT  0860
   37 DO 60 J=1,NJJ                                                       FINPUT  0870
      IF (I.EQ.1) NK = MNPL(J)                                            FINPUT  0880
      IF (I.EQ.2) NK = MNBLT(J)                                           FINPUT  0890
      IF (I.EQ.3) NK = MNSEG(J)                                           FINPUT  0900
      IF (I.EQ.4) NK = IGLOB(J)                                           FINPUT  0910
      IF (NK.LE.0) GO TO 60                                               FINPUT  0920
      DO 59 K=1,NK                                                        FINPUT  0930
      IF (IJK.EQ.0) WRITE (6,38) I                                        FINPUT  0940
   38 FORMAT('0',119X,'CARDS F.',I1)                                      FINPUT  0950
      IF (IJK.EQ.0 .AND. I.NE.4) WRITE (6,39) SURFCE(1,I),SURFCE(2,I)     FINPUT  0960
   39 FORMAT('0',3X,2A4,8X,'SEGMENT',2X,'FORCE DEFLECTION',6X,'INERTIAL   FINPUT  0970
     *SPIKE',10X,'R FACTOR',13X,'G FACTOR',10X,'FRICTION COEF.')          FINPUT  0980
      IF (IJK.EQ.0 .AND. I.EQ.4) WRITE (6,40)                             FINPUT  0990
   40 FORMAT('0',5X,'JOINT (GLOBALGRAPHIC)',2X,'TORQUE DEFLECTION',6X,'H  FINPUT  1000
```

```
     *ERRON FORMULA',10X,'R FACTOR',13X,'G FACTOR',10X,'FRICTION COEF.')   FINPUT  1010
      IJK = 1                                                              FINPUT  1020
C                                                                          FINPUT  1030
C     INPUT CONTACT SURFACE NO., SEGMENT NO., AND FUNCTION NOS.            FINPUT  1040
C     INPUT CARD F.(I).(K)                                                 FINPUT  1050
C                                                                          FINPUT  1060
      READ  (5,33) NJ,MS,NF                                                FINPUT  1070
      WRITE (6,41) NJ,MS,NF                                                FINPUT  1080
   41 FORMAT('0',I7,'-',I3,I11,'-',I3,I8,4I21)                             FINPUT  1090
      IF (NJ.NE.J) WRITE (6,42)                                            FINPUT  1100
   42 FORMAT(' CONTACT INPUT ERROR. PROGRAM TERMINATED.')                  FINPUT  1110
      IF  (NJ.NE.J)  STOP 14                                               FINPUT  1120
      NLT = 1                                                              FINPUT  1130
      DO 43 JJ = 1,31                                                      FINPUT  1140
   43 KTITLE(JJ) = BLANK                                                   FINPUT  1150
      GO TO (44,46,48,49),I                                                FINPUT  1160
C                                                                          FINPUT  1170
C     PLACE SEGMENT NO. AND INDEX TO NTAB ARRAY INTO M- AND NT- ARRAYS.    FINPUT  1180
C                                                                          FINPUT  1190
   44 MPL(1,K,J) = MS(1)                                                   FINPUT  1200
      MPL(2,K,J) = MS(2)                                                   FINPUT  1210
      MPL(3,K,J) = MS(3)                                                   FINPUT  1220
      NTPL(K,J)  = MXNTB+1                                                 FINPUT  1230
      DO 45 JJ = 1,5                                                       FINPUT  1240
   45 KTITLE(JJ) = PLTTL (JJ,J)                                            FINPUT  1250
      GO TO 50                                                             FINPUT  1260
   46 MBLT(1,K,J) = MS(1)                                                  FINPUT  1270
      MBLT(2,K,J) = MS(2)                                                  FINPUT  1280
      MBLT(3,K,J) = MS(3)                                                  FINPUT  1290
      NTBLT(K,J) = MXNTB+1                                                 FINPUT  1300
      DO 47 JJ = 1,5                                                       FINPUT  1310
   47 KTITLE(JJ) = BLTTTL (JJ,J)                                           FINPUT  1320
C                                                                          FINPUT  1330
C     SET UP TWO TABLES FOR FULL BELT FRICTION                             FINPUT  1340
C                                                                          FINPUT  1350
      IF (NF(5).NE.0)  NLT = 2                                             FINPUT  1360
      GO TO 50                                                             FINPUT  1370
   48 MSEG(1,K,J) = MS(1)                                                  FINPUT  1380
      MSEG(2,K,J) = MS(2)                                                  FINPUT  1390
      MSEG(3,K,J) = MS(3)                                                  FINPUT  1400
      NTSEG(K,J) = MXNTB+1                                                 FINPUT  1410
      KTITLE (3) = SEG(J)                                                  FINPUT  1420
      GO TO 50                                                             FINPUT  1430
C                                                                          FINPUT  1440
C       NOTE:  GLOBALGRAPHIC JOINT WILL SAVE NT IN IGLOB ARRAY             FINPUT  1450
C                                                                          FINPUT  1460
   49 IGLOB(J)   = MXNTB+1                                                 FINPUT  1470
      KTITLE(2) = JOINT(J)                                                 FINPUT  1480
C                                                                          FINPUT  1490
C     SET UP POINTERS TO TAB ARRAY IN NTAB ARRAY.                          FINPUT  1500
```

```
C                                                                       FINPUT  1510
   50 NFJ = MS(2)                                                       FINPUT  1520
      IF (NFJ.GT.0) KTITLE(6) = SEG(NFJ)                                FINPUT  1530
      DO 51 JJ=1,NLT                                                    FINPUT  1540
   51 CALL FDINIT                                                       FINPUT  1550
      WRITE (6,53) KTITLE                                               FINPUT  1560
   53 FORMAT(1X,5A4,1X,A4,5(1X,5A4))                                    FINPUT  1570
      IF (NF(1).NE.0) GO TO 56                                          FINPUT  1580
C                                                                       FINPUT  1590
C     IF FORCE DEFLECTION FUNCTION NO. IS ZERO,                         FINPUT  1600
C     SET UP FOR ROLLING CONSTRAINT                                     FINPUT  1610
C                                                                       FINPUT  1620
      NQ = NQ+1                                                         FINPUT  1630
      NTAB(MXNTB-4) = -NQ                                               FINPUT  1640
      KQTYPE(NQ) = -4                                                   FINPUT  1650
      KQ1(NQ) = MS(2)                                                   FINPUT  1660
      KQ2(NQ) = MS(1)                                                   FINPUT  1670
      IF (I.NE.3) GO TO 56                                              FINPUT  1680
      KQ1(NQ) = J                                                       FINPUT  1690
      KQ2(NQ) = MS(2)                                                   FINPUT  1700
   56 CONTINUE                                                          FINPUT  1710
   59 CONTINUE                                                          FINPUT  1720
   60 CONTINUE                                                          FINPUT  1730
   61 CONTINUE                                                          FINPUT  1740
C                                                                       FINPUT  1750
C     INPUT CARD F.5 - JOINT FUNCTIONS TO BE USED.                      FINPUT  1760
C                                                                       FINPUT  1770
      IF (NJNT.LE.0) GO TO 81                                           FINPUT  1780
      IF  (NJNTF.NE.0)  GO TO 76                                        FINPUT  1790
      DO  75  J=1,NJNT                                                  FINPUT  1800
   75 JOINTF(J) = 0                                                     FINPUT  1810
      GO TO 81                                                          FINPUT  1820
   76 READ  (5,33)  (JOINTF(J),J=1,NJNT)                               FINPUT  1830
      IJK = 0                                                           FINPUT  1840
      DO 80 J=1,NJNT                                                    FINPUT  1850
      IF (JOINTF(J).EQ.0) GO TO 80                                      FINPUT  1860
      IF (IJK.EQ.0) WRITE (6,77)                                        FINPUT  1870
   77 FORMAT('1',119X,'CARD F.5'/                                       FINPUT  1880
     *       ' THE FOLLOWING JOINT RESTORING FORCE FUNCTIONS AS DEFINED FINPUT  1890
     *ON CARDS E.7 WILL BE USED.'//4X,'JOINT',10X,'FUNCTION'//)         FINPUT  1900
      JF = JOINTF(J)                                                    FINPUT  1910
      IJK = 1                                                           FINPUT  1920
      WRITE (6,78) J,JOINT(J),JF,(JTITLE(I,JF),I=1,5)                   FINPUT  1930
   78 FORMAT(I6,'-',A4,I10,'-',5A4)                                     FINPUT  1940
      IF (NTI(JF).EQ.0) WRITE (6,42)                                    FINPUT  1950
      IF (NTI(JF).EQ.0) STOP 17                                         FINPUT  1960
   80 CONTINUE                                                          FINPUT  1970
C                                                                       FINPUT  1980
C     INPUT CONTACT SEGMENTS FOR AIRBAG, IF ANY.                        FINPUT  1990
C                                                                       FINPUT  2000
```

```
   81 IF (NBAG.LE.0) GO TO 69                                                    FINPUT  2010
      IJK = 0                                                                    FINPUT  2020
      DO 68 J=1,NBAG                                                             FINPUT  2030
C                                                                                FINPUT  2040
C     INPUT CARD F.6.(J)                                                         FINPUT  2050
C                                                                                FINPUT  2060
      READ (5,63) K,NK,(MBAG(2,I,J),MBAG(3,I,J),I=1,NK)                          FINPUT  2070
   63 FORMAT(2I4,20I2)                                                           FINPUT  2080
      MNBAG(J) = NK                                                              FINPUT  2090
      IF (NK.EQ.0) GO TO 68                                                      FINPUT  2100
      IF (IJK.EQ.0) WRITE (6,64)                                                 FINPUT  2110
   64 FORMAT(////5X,'AIRBAG',4X,'VS.',4X,'SEGMENTS',90X,'CARDS F.6')             FINPUT  2120
      IF (K.NE.J) WRITE (6,42)                                                   FINPUT  2130
      IF (K.NE.J) STOP 20                                                        FINPUT  2140
      WRITE (6,65) J,(MBAG(2,I,J),MBAG(3,I,J),I=1,NK)                            FINPUT  2150
   65 FORMAT('0      NO.',I2,12X,10(I3,'-',I3))                                 FINPUT  2160
      DO 66 I=1,NK                                                               FINPUT  2170
      K = MBAG(2,I,J)                                                            FINPUT  2180
   66 KTITLE(I) = SEG(K)                                                         FINPUT  2190
      WRITE (6,67) (BAGTTL(I,J),I=1,5),(KTITLE(I),I=1,NK)                        FINPUT  2200
   67 FORMAT(1X,5A4,10(3X,A4))                                                   FINPUT  2210
   68 CONTINUE                                                                   FINPUT  2220
                                                                                 FINPUT  2230
C                                                                                FINPUT  2240
C     INPUT CARDS F.7.A-F.7.B FOR SUBROUTINE WINDY.                              FINPUT  2250
C                                                                                FINPUT  2260
   69 DO  85  J=1,NGRND                                                          FINPUT  2270
   85 MWSEG(1,J) = 0                                                             FINPUT  2280
      IF (NWINDF.EQ.0) GO TO 99                                                  FINPUT  2290
      READ  (5,33)  (MWSEG(1,J),J=1,NSEG)                                        FINPUT  2300
      IPAGE = 0                                                                  FINPUT  2310
      DO 73 J=1,NSEG                                                             FINPUT  2320
      IWIND(J) = 0                                                               FINPUT  2330
      WTIME(J) = 0.0                                                             FINPUT  2340
      IF (MWSEG(1,J).EQ.0) GO TO 73                                              FINPUT  2350
      IF (IPAGE.EQ.0) WRITE (6,70)                                               FINPUT  2360
   70 FORMAT('1 SEGMENT WIND FORCES',99X,'CARDS F.7'//                           FINPUT  2370
     *       '  SEGMENT-ELLIPSOID  SEGMENT-PLANE',                              FINPUT  2380
     *      17X,'WIND FORCE FUNCTION')                                           FINPUT  2390
      IPAGE = 1                                                                  FINPUT  2400
      READ  (5,33) (MWSEG(I,J),I=1,5)                                            FINPUT  2410
      WRITE (6,71) (MWSEG(I,J),I=1,5)                                            FINPUT  2420
   71 FORMAT(1H0,I7,2H -,I3,I14,2H -,I3,I30)                                     FINPUT  2430
      IF (MWSEG(1,J).NE.J) WRITE (6,42)                                          FINPUT  2440
      IF (MWSEG(1,J).NE.J) STOP 21                                               FINPUT  2450
      M3 = MWSEG(3,J)                                                            FINPUT  2460
      M4 = MWSEG(4,J)                                                            FINPUT  2470
      M5 = MWSEG(5,J)                                                            FINPUT  2480
      WRITE (6,72) SEG(J),SEG(M3),(PLTTL(I,M4),I=I,5)                            FINPUT  2490
     *                            ,(JTITLE(I,M5),I=1,5)                        FINPUT  2500
   72 FORMAT(5X,A4,15X,A4,1H-,5A4,2X,5A4)                                        FINPUT  2510
   73 CONTINUE                                                                   FINPUT  2520
   99 RETURN                                                                     FINPUT  2530
      END
```

```
      SUBROUTINE FLXSEG                                                 FLXSEG  0010
C                                                        REV 19 08/05/78  FLXSEG  0020
      IMPLICIT REAL*8(A-H,O-Z)                                          FLXSEG  0030
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           FLXSEG  0040
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)        FLXSEG  0050
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30), FLXSEG  0060
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       FLXSEG  0070
      COMMON/FLXBLE/ HF(4,12,8),B42(3,3,24),V4(3,8),NFLEX(3,8)          FLXSEG  0080
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         FLXSEG  0090
     *               UNITL,UNITM,UNITT,GRAVTY(3)                        FLXSEG  0100
      COMMON/TEMPVS/ TT(3,3), THN(4), CN1(3,3), CN(3,3), WNM1(3),       FLXSEG  0110
     *               THND(4), PTD(3), WCSN(3), RHSN(3), RHS1(3),        FLXSEG  0120
     *               RHS2(3), GF(3,4), GC(3,3), CGC(3,3), THA(3),       FLXSEG  0130
     *              THAD(3), THADEG(3), DN2N1(3,3), RMG(3)              FLXSEG  0140
      DIMENSION IDYPR(3)                                                FLXSEG  0150
      DATA      IDYPR/3,2,1/                                            FLXSEG  0160
      IF (NFLX.EQ.0) GO TO 99                                           FLXSEG  0170
      CALL ELTIME(1,34)                                                 FLXSEG  0180
      IFX = 1                                                           FLXSEG  0190
   11 N1 = NFLEX(1,IFX)                                                 FLXSEG  0200
      N3 = NFLEX(3,IFX)                                                 FLXSEG  0210
      CALL DOTT33(D(1,1,N3),D(1,1,N1),TT)                               FLXSEG  0220
      THN(1) =  DATAN2(TT(1,2),TT(1,1))                                 FLXSEG  0230
      THN(2) = -DARSIN(TT(1,3))                                         FLXSEG  0240
      THN(3) =  DATAN2(TT(2,3),TT(3,3))                                 FLXSEG  0250
      THN(4) =  1.0                                                     FLXSEG  0260
      CT22 = 1.0-TT(1,3)**2                                             FLXSEG  0270
      CT2  = DSQRT(CT22)                                                FLXSEG  0280
      ST2  = -TT(1,3)                                                   FLXSEG  0290
      CT1  =  TT(1,1)/CT2                                               FLXSEG  0300
      ST1  =  TT(1,2)/CT2                                               FLXSEG  0310
      CN1(1,1) = -TT(1,1)*TT(1,3)/CT22                                  FLXSEG  0320
      CN1(1,2) = -TT(1,2)*TT(1,3)/CT22                                  FLXSEG  0330
      CN1(1,3) =  1.0                                                   FLXSEG  0340
      CN1(2,1) = -ST1                                                   FLXSEG  0350
      CN1(2,2) =  CT1                                                   FLXSEG  0360
      CN1(2,3) =  0.0                                                   FLXSEG  0370
      CN1(3,1) =  TT(1,1)/CT22                                          FLXSEG  0380
      CN1(3,2) =  TT(1,2)/CT22                                          FLXSEG  0390
      CN1(3,3) =  0.0                                                   FLXSEG  0400
      CALL DOT31(TT,WMEG(1,N3),WNM1)                                    FLXSEG  0410
      DO 12 I=1,3                                                       FLXSEG  0420
   12 WNM1(I) = WNM1(I) - WMEG(I,N1)                                    FLXSEG  0430
      CALL MAT31(CN1,WNM1,THND)                                         FLXSEG  0440
      THND(4) =  0.0                                                    FLXSEG  0450
      CALL CROSS(WMEG(1,N1),WNM1,WCSN)                                  FLXSEG  0460
      RHSN(1) = ( (-THND(1)*ST1*ST2 + THND(2)*CT1/CT2)*WNM1(1)          FLXSEG  0470
     *          +( THND(1)*CT1*ST2 + THND(2)*ST1/CT2)*WNM1(2) )/CT2     FLXSEG  0480
      RHSN(2) =  -THND(1)*(CT1*WNM1(1) + ST1*WNM1(2))                   FLXSEG  0490
      RHSN(3) = ( (-THND(1)*ST1 + THND(2)*CT1*ST2/CT2)*WNM1(1)          FLXSEG  0500
```

```
     *           +( THND(1)*CT1 + THND(2)*ST1*ST2/CT2)*WNM1(2) )/CT2        FLXSEG  0510
  13 N2 = NFLEX(2,IFX)                                                       FLXSEG  0520
     M = 0                                                                   FLXSEG  0530
     DO 15 I=1,3                                                             FLXSEG  0540
     DO 14 J=1,4                                                             FLXSEG  0550
     JM = J+M                                                                FLXSEG  0560
     GF(I,J) = 0.0                                                           FLXSEG  0570
     DO 14 K=1,4                                                             FLXSEG  0580
  14 GF(I,J) = GF(I,J) + HF(K,JM,IFX)*THN(K)                                 FLXSEG  0590
  15 M = M+4                                                                 FLXSEG  0600
     DO 17 I=1,3                                                             FLXSEG  0610
     THA(I)  = 0.0                                                           FLXSEG  0620
     THAD(I) = 0.0                                                           FLXSEG  0630
     DO 16 J=1,4                                                             FLXSEG  0640
     THA (I) = THA (I) + GF(I,J)*THN (J)                                     FLXSEG  0650
  16 THAD(I) = THAD(I) + GF(I,J)*THND(J)                                     FLXSEG  0660
     THA (I) = 0.5*THA(I)                                                    FLXSEG  0670
  17 THADEG(I) = THA(I)/RADIAN                                               FLXSEG  0680
     CALL DRCYPR (DN2N1,THADEG,IDYPR)                                        FLXSEG  0690
     CALL MAT33(DN2N1,D(1,1,N1),D(1,1,N2))                                   FLXSEG  0700
     CSC = DCOS(THA(2))                                                      FLXSEG  0710
     CSS = DSIN(THA(2))                                                      FLXSEG  0720
     CN(1,1) =    0.0                                                        FLXSEG  0730
     CN(2,1) =    0.0                                                        FLXSEG  0740
     CN(3,1) =    1.0                                                        FLXSEG  0750
     CN(1,2) =  -DSIN(THA(1))                                                FLXSEG  0760
     CN(2,2) =   DCOS(THA(1))                                                FLXSEG  0770
     CN(3,2) =    0.0                                                        FLXSEG  0780
     CN(1,3) =   CSC*CN(2,2)                                                 FLXSEG  0790
     CN(2,3) =  -CSC*CN(1,2)                                                 FLXSEG  0800
     CN(3,3) =  -CSS                                                         FLXSEG  0810
     CALL MAT33(GF, CN1, GC)                                                 FLXSEG  0820
     CALL MAT33(CN, GC, CGC)                                                 FLXSEG  0830
     CALL DOT33 (D(1,1,N1),CGC,B42(1,1,3*IFX-2))                             FLXSEG  0840
     CALL DOTT33(B42(1,1,3*IFX-2),TT,B42(1,1,3*IFX))                         FLXSEG  0850
     DO 20 I=1,3                                                             FLXSEG  0860
     DO 20 J=1,3                                                             FLXSEG  0870
     B42(I,J,3*IFX-2) = B42(I,J,3*IFX-2) - D(J,I,N1)                         FLXSEG  0880
     B42(I,J,3*IFX-1) = D(J,I,N2)                                            FLXSEG  0890
  20 B42(I,J,3*IFX  ) = -B42(I,J,3*IFX)                                      FLXSEG  0900
C                                                                            FLXSEG  0910
C    COMPUTE V4                                                              FLXSEG  0920
C                                                                            FLXSEG  0930
     CALL MAT31(CGC,WNM1,RHS1)                                               FLXSEG  0940
     DO 21 I=1,3                                                             FLXSEG  0950
  21 RMG(I) = RHS1(I) + WMEG(I,N1)                                           FLXSEG  0960
     CALL MAT31(DN2N1,RMG,WMEG(1,N2))                                        FLXSEG  0970
     CALL CROSS(WMEG(1,N1),RHS1,RHS2)                                        FLXSEG  0980
     CALL MAT31(CGC,WCSN,RHS1)                                               FLXSEG  0990
     DO 25 I=1,3                                                             FLXSEG  1000
```

```
25 RHS1(I) = RHS2(I) - RHS1(I)                                                  FLXSEG  1010
   CALL MAT31(GC,WNM1,RHS2)                                                     FLXSEG  1020
   RHS1(1) = RHS1(1) - THAD(1)*(CN(2,2)*RHS2(2)-CN(1,2)*CSC*RHS2(3))            FLXSEG  1030
  *                  - THAD(2)*CN(2,2)*CSS*RHS2(3)                            FLXSEG  1040
   RHS1(2) = RHS1(2) + THAD(1)*(CN(1,2)*RHS2(2)+CN(2,2)*CSC*RHS2(3))            FLXSEG  1050
  *                  + THAD(2)*CN(1,2)*CSS*RHS2(3)                            FLXSEG  1060
   RHS1(3) = RHS1(3) - THAD(2)*CSC*RHS2(3)                                      FLXSEG  1070
   CALL MAT31(GF, RHSN, RHS2)                                                   FLXSEG  1080
   M = 1                                                                        FLXSEG  1090
   DO 30 I=1,3                                                                  FLXSEG  1100
   DO  26  J=1,3                                                                FLXSEG  1110
   PTD(J) = 0.0                                                                 FLXSEG  1120
   DO  26  K=1,3                                                                FLXSEG  1130
   KK = K+M-1                                                                   FLXSEG  1140
26 PTD(J) = PTD(J) + HF(J,KK,IFX)*THND(K)                                       FLXSEG  1150
   RHS2(I) = RHS2(I) + XDY(PTD,CN1,WNM1)                                        FLXSEG  1160
30 M = M+4                                                                      FLXSEG  1170
   CALL MAT31(CN, RHS2, PTD)                                                    FLXSEG  1180
   DO 35 I=1,3                                                                  FLXSEG  1190
35 RHS1(I) = RHS1(I) + PTD(I)                                                   FLXSEG  1200
   CALL DOT31(D(1,1,N1),RHS1,V4(1,IFX))                                         FLXSEG  1210
   IF (IFX.EQ.NFLX) GO TO 98                                                    FLXSEG  1220
   IFX = IFX+1                                                                  FLXSEG  1230
   IF (NFLEX(1,IFX).EQ.N1 .AND. NFLEX(3,IFX).EQ.N3) GO TO 13                    FLXSEG  1240
   GO TO 11                                                                     FLXSEG  1250
98 CALL ELTIME(2,34)                                                            FLXSEG  1260
99 RETURN                                                                       FLXSEG  1270
   END                                                                          FLXSEG  1280
```

```
      DOUBLE PRECISION FUNCTION FNTERP(THETA,PHI,NT)                    FNTERP  0010
C                                                          REV 19 08/05/78 FNTERP  0020
C     COMPUTES THE RESTORING TORQUE OF A JOINT AS A FUNCTION OF THE      FNTERP  0030
C     FLEXURE ANGLE (THETA) AND THE AZIMUTH ANGLE (PHI) AS DEFINED BY    FNTERP  0040
C     FUNCTION NO. NT                                                    FNTERP  0050
C                                                                        FNTERP  0060
C     ASSUMES  0  < THETA < PI                                           FNTERP  0070
C             -PI <  PHI  < PI                                           FNTERP  0080
C              DATA IN TAB ARRAY CONTAINS NTHETA,NPHI FOLLOWED BY        FNTERP  0090
C              TWO DIMENSIONAL ARRAY OF FUNCTIONAL VALUES (NTHETA > 0)   FNTERP  0100
C              OR POLYNOMIAL COEFFICIENTS (NTHETA < 0) FOR EQUALLY       FNTERP  0110
C              SPACED VALUES OF PHI.                                     FNTERP  0120
C                                                                        FNTERP  0130
C                   THETA(I) = (I-1)*PI/(NTHETA-1) FOR I=1,NTHETA        FNTERP  0140
C                   PHI(J) = -PI + (J-1)*2*PI/NPHI FOR J=1,NPHI          FNTERP  0150
C                       F(THETA,PI) = F(THETA,-PI)                       FNTERP  0160
C                                                                        FNTERP  0170
C     SUBROUTINE EVALUATES G1(THETA) = F(THETA,PHI(J)  )                 FNTERP  0180
C                          G2(THETA) = F(THETA,PHI(J+1))                 FNTERP  0190
C                FOR  PHI(J) < PHI < PHI(J+1)                            FNTERP  0200
C     BY LINEAR INTERPOLATION OR POLYNOMIAL EVALUATION AND THEN LINEAR   FNTERP  0210
C     INTERPOLATES BETWEEN G1 AND G2 TO OBTAIN F(THETA,PHI).             FNTERP  0220
C     IF F < 0, F IS SET TO ZERO, THEREFORE A DEAD BAND IS OBTAINED      FNTERP  0230
C     BY NEGATIVE VALUES IN THE TABLE.                                   FNTERP  0240
C                                                                        FNTERP  0250
      IMPLICIT REAL*8 (A-H,O-Z)                                          FNTERP  0260
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                          FNTERP  0270
     *               UNITL,UNITM,UNITT,GRAVTY(3)                       FNTERP  0280
      COMMON/TABLES/ MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(500),TAB(2600) FNTERP  0290
      IERROR = 0                                                         FNTERP  0300
      IF (PHI.LT.-PI)    IERROR = 1                                      FNTERP  0310
      IF (PHI.GT. PI)    IERROR = 2                                      FNTERP  0320
      IF (THETA.LT.0.0) IERROR = 3                                       FNTERP  0330
      IF (THETA.GT.PI ) IERROR = 4                                       FNTERP  0340
      IF (IERROR.NE.0)  WRITE (6,11) IERROR,THETA,PHI,NT                 FNTERP  0350
   11 FORMAT('0 IMPROPER ARGUMENTS TO FUNCTION FNTERP. ERROR CODE =',I4/ FNTERP  0360
     *       '0 THETA =',G25.15, ' PHI =',G25.15,' NT =',I6)             FNTERP  0370
      IF (IERROR.NE.0) STOP 36                                           FNTERP  0380
      NF = NTI(NT) + 5                                                   FNTERP  0390
      NTHETA = TAB(NF)                                                   FNTERP  0400
      NPHI   = TAB(NF+1)                                                 FNTERP  0410
C                                                                        FNTERP  0420
C     DETERMINE INDEX AND INTERPOLATION PARAMETERS FOR PHI.              FNTERP  0430
C                                                                        FNTERP  0440
      XNP = (PHI+PI)/(2.0*PI)*TAB(NF+1)                                  FNTERP  0450
      NP1 = XNP                                                          FNTERP  0460
      NP2 = NP1+1                                                        FNTERP  0470
      IF (NP2.GE.NPHI) NP2 = 0                                           FNTERP  0480
      RP2 = XNP - DFLOAT(NP1)                                            FNTERP  0490
      RP1 = 1.0 - RP2                                                    FNTERP  0500
```

```
      NTH = IABS(NTHETA)                                                FNTERP  0510
      IP1 = NF+1+NP1*NTH                                                FNTERP  0520
      IP2 = NF+1+NP2*NTH                                                FNTERP  0530
C                                                                       FNTERP  0540
C     DETERMINE INDEX AND INTERPOLATION PARAMETERS FOR THETA.           FNTERP  0550
C                                                                       FNTERP  0560
      IF (NTHETA.LT.0) GO TO 20                                         FNTERP  0570
      XNT = THETA/PI*(TAB(NF)-1.0)                                      FNTERP  0580
      NT1 = XNT                                                         FNTERP  0590
      RT2 = XNT - DFLOAT(NT1)                                           FNTERP  0600
      RT1 = 1.0 - RT2                                                   FNTERP  0610
      IT1 = IP1 + NT1                                                   FNTERP  0620
      IT2 = IP2 + NT1                                                   FNTERP  0630
      G1  = RT1*TAB(IT1+1) + RT2*TAB(IT1+2)                             FNTERP  0640
      G2  = RT1*TAB(IT2+1) + RT2*TAB(IT2+2)                             FNTERP  0650
      GO TO 23                                                          FNTERP  0660
C                                                                       FNTERP  0670
C     COMPUTE FOR POLYNOMIALS IN THETA FOR FIXED PHI.                   FNTERP  0680
C                                                                       FNTERP  0690
   20 NPOLY = -NTHETA-1                                                 FNTERP  0700
      IT1 = IP1 + NPOLY + 2                                             FNTERP  0710
      IT2 = IP2 + NPOLY + 2                                             FNTERP  0720
      THETA1 = THETA - TAB(IP1+1)                                       FNTERP  0730
      THETA2 = THETA - TAB(IP2+1)                                       FNTERP  0740
      G1 = 0.0                                                          FNTERP  0750
      G2 = 0.0                                                          FNTERP  0760
      DO 21 I=1,NPOLY                                                   FNTERP  0770
      IT1 = IT1-1                                                       FNTERP  0780
      IT2 = IT2-1                                                       FNTERP  0790
      G1 = THETA1*(TAB(IT1)+G1)                                         FNTERP  0800
   21 G2 = THETA2*(TAB(IT2)+G2)                                         FNTERP  0810
   23 FNTERP = RP1*G1 + RP2*G2                                          FNTERP  0820
      IF (FNTERP.LT.0.0) FNTERP = 0.0                                   FNTERP  0830
      RETURN                                                            FNTERP  0840
      END                                                               FNTERP  0850
```

```
      SUBROUTINE FRCDFL (D,RATE,M,N,FRCDF,ELOSS)                        FRCDFL  0010
C                                                        REV 19 10/19/79  FRCDFL  0020
C     EVALUATE FORCE DEFLECTION FUNCTION AT POINT D, WHERE DEFINITION    FRCDFL  0030
C     OF FUNCTION IS CONTROLLED BY M INDEX OF NTAB ARRAY.               FRCDFL  0040
C     DERIVATIVE, FUNCTION OR INTEGRAL IS EVALUATED AS N = 0,1 OR 2.    FRCDFL  0050
C        NTAB(M)    -  INDEX TO TAB ARRAY FOR REAL DATA                 FRCDFL  0060
C        NTAB(M+1)  -  INDEX TO TAB ARRAY FOR BASE FUNCTION             FRCDFL  0070
C        NTAB(M+2)  -  INDEX TO TAB ARRAY FOR INERTIAL FUNCTION, IF ANY FRCDFL  0080
C                                                                       FRCDFL  0090
C     ASSUMES  0 < DG < DCUBIC < DREF < DMAX                            FRCDFL  0100
C               BUT ANY < MAY BE LESS THAN OR EQUAL TO                  FRCDFL  0110
C                                                                       FRCDFL  0120
      IMPLICIT REAL*8(A-H,O-Z)                                          FRCDFL  0130
      COMMON/TABLES/ MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(500),TAB(2600) FRCDFL  0140
      F = 0.0                                                           FRCDFL  0150
      ELOSS = 0.0                                                       FRCDFL  0160
      L = NTAB(M)                                                       FRCDFL  0170
      TAB(L) = D                                                        FRCDFL  0180
      IF (D.LT.0.0) GO TO 99                                            FRCDFL  0190
      DMAX = TAB(L+8)                                                   FRCDFL  0200
      IF (D.LT.DMAX) GO TO 10                                           FRCDFL  0210
C                                                                       FRCDFL  0220
C     DMAX < D , USE MAX VALUE                                          FRCDFL  0230
C                                                                       FRCDFL  0240
      IF (N-1)  99,9,99                                                 FRCDFL  0250
    9 FDMAX = TAB(L+10)                                                 FRCDFL  0260
      F = FDMAX                                                         FRCDFL  0270
      GO TO 40                                                          FRCDFL  0280
   10 DREF = TAB(L+7)                                                   FRCDFL  0290
      IF (D.GE.DREF) GO TO 30                                           FRCDFL  0300
      DCUBIC = TAB(L+6)                                                 FRCDFL  0310
      IF (DCUBIC.GE.DREF) GO TO 20                                      FRCDFL  0320
      IF (D.LE.DCUBIC) GO TO 20                                         FRCDFL  0330
C                                                                       FRCDFL  0340
C     DCUBIC < D < DREF , USE CUBIC                                     FRCDFL  0350
C                                                                       FRCDFL  0360
      LC = L+14                                                         FRCDFL  0370
      DC0 = TAB(L+18)                                                   FRCDFL  0380
      X = D-DC0                                                         FRCDFL  0390
      IF (N-1) 12,11,99                                                 FRCDFL  0400
C                                                                       FRCDFL  0410
C     USE CUBIC DEFINITION                                              FRCDFL  0420
C                                                                       FRCDFL  0430
   11 F = TAB(LC) + X *(TAB(LC+1)+X*(TAB(LC+2)+X*TAB(LC+3)))            FRCDFL  0440
      GO TO 40                                                          FRCDFL  0450
C                                                                       FRCDFL  0460
C     USE DERIVATIVE OF CUBIC                                           FRCDFL  0470
C                                                                       FRCDFL  0480
   12 F = TAB(LC+1)+X*(2.0*TAB(LC+2)+X*3.0*TAB(LC+3))                   FRCDFL  0490
      GO TO 99                                                          FRCDFL  0500
```

```
   20 DG = TAB(L+5)                                                      FRCDFL  0510
      IF (D.LE.DG) GO TO 40                                              FRCDFL  0520
C                                                                        FRCDFL  0530
C     DG < D < DCUBIC , USE QUADRATIC                                    FRCDFL  0540
C                                                                        FRCDFL  0550
      LQ = L+11                                                          FRCDFL  0560
      X = D-DG                                                           FRCDFL  0570
      IF (N-1) 22,21,99                                                  FRCDFL  0580
C                                                                        FRCDFL  0590
C     USE QUADRATIC DEFINITION                                           FRCDFL  0600
C                                                                        FRCDFL  0610
   21 F = TAB(LQ)+X*(TAB(LQ+1)+X*TAB(LQ+2))                              FRCDFL  0620
      GO TO 40                                                           FRCDFL  0630
C                                                                        FRCDFL  0640
C     USE DERIVATIVE OF QUADRATIC.                                       FRCDFL  0650
C                                                                        FRCDFL  0660
   22 F = TAB(LQ+1)+X*2.0*TAB(LQ+2)                                      FRCDFL  0670
      GO TO 99                                                           FRCDFL  0680
C                                                                        FRCDFL  0690
C     DREF < D < DMAX, USE BASE FUNCTION                                 FRCDFL  0700
C                                                                        FRCDFL  0710
   30 IF (N-1) 31,31,99                                                  FRCDFL  0720
   31 NB = NTAB(M+1)                                                     FRCDFL  0730
C                                                                        FRCDFL  0740
C     EVALUATE BASE FUNCTION                                             FRCDFL  0750
C                                                                        FRCDFL  0760
      IF (NB.GT.0) F = EVALFD(D,NB,N)                                    FRCDFL  0770
      NI = NTAB(M+2)                                                     FRCDFL  0780
C                                                                        FRCDFL  0790
C     ADD INERTIAL FUNCTION , IF ANY                                     FRCDFL  0800
C                                                                        FRCDFL  0810
      IF (NI.GT.0) F = F+EVALFD(D,NI,N)                                  FRCDFL  0820
   40 IF  (N.NE.1)  GO TO 99                                             FRCDFL  0830
C                                                                        FRCDFL  0840
C     COMPUTE AND ADD RATE DEPENDENT FUNCTIONS, IF ANY.                  FRCDFL  0850
C                                                                        FRCDFL  0860
C     CURRENT RESTRICTIONS:                                              FRCDFL  0870
C                                                                        FRCDFL  0880
C        1) COMPUTED FOR N=1 (FUNCTION) ONLY.                            FRCDFL  0890
C                                                                        FRCDFL  0900
C        2) FUNCTION NOS. M+2,M+3 AND M+4 (USED FOR INERTIAL SPIKE,      FRCDFL  0910
C           R FACTOR AND G FACTOR FUNCTIONS) MUST BE NEGATIVE OR ZERO,   FRCDFL  0920
C           I.E., THESE FUNCTIONS CANNOT BE USED IN CONJUNCTION WITH     FRCDFL  0930
C           THE RATE DEPENDENT FUNCTIONS.                                FRCDFL  0940
C                                                                        FRCDFL  0950
C        3) ASSUMES THE FUNCTIONAL FORM                                  FRCDFL  0960
C                                                                        FRCDFL  0970
C                 F(D,D') = F1(D) + F2(D)*F3(D') + F4(D')                FRCDFL  0980
C                                                                        FRCDFL  0990
C           WHERE F1(D ) IS DEFINED BY FUNCTION NTAB(M+1)>0,             FRCDFL  1000
```

```
C                       I.E., NORMAL FORCE DEFLECTION FUNCTION WITH NO     FRCDFL  1010
C                       INERTIAL SPIKE FUNCTION AND DEFAULT VALUES         FRCDFL  1020
C                       R=1 AND G=0 (UNLOADING AND RELOADING SAME AS       FRCDFL  1030
C                       ORIGINAL LOADING);                                 FRCDFL  1040
C                                                                          FRCDFL  1050
C                 F2(D ) IS DEFINED BY FUNCTION NTAB(M+2)<0,               FRCDFL  1060
C                        IF NTAB(M+2)=0, F2(D )=0;                         FRCDFL  1070
C                                                                          FRCDFL  1080
C                 F3(D') IS DEFINED BY FUNCTION NTAB(M+3)<0,               FRCDFL  1090
C                        IF NTAB(M+3)=0, F3(D')=0;                         FRCDFL  1100
C                                                                          FRCDFL  1110
C           AND   F4(D') IS DEFINED BY FUNCTION NTAB(M+4)<0,               FRCDFL  1120
C                        IF NTAB(M+4)=0, F4(D')=0.                         FRCDFL  1130
C                                                                          FRCDFL  1140
C           NOTE: FUNCTIONAL FORM CAN BE CHANGED BY REVISING PROGRAM       FRCDFL  1150
C           BETWEEN STATEMENTS 40 AND 99.                                  FRCDFL  1160
C                                                                          FRCDFL  1170
      F2 = 0.0                                                             FRCDFL  1180
      F3 = 0.0                                                             FRCDFL  1190
      F4 = 0.0                                                             FRCDFL  1200
      N2 = -NTAB(M+2)                                                      FRCDFL  1210
      N3 = -NTAB(M+3)                                                      FRCDFL  1220
      N4 = -NTAB(M+4)                                                      FRCDFL  1230
      IF  (N2.GT.0)  F2 = EVALFD (D,    N2,N)                            FRCDFL  1240
      IF  (N3.GT.0)  F3 = EVALFD (RATE,N3,N)                             FRCDFL  1250
      IF  (N4.GT.0)  F4 = EVALFD (RATE,N4,N)                             FRCDFL  1260
      F = F + F2*F3 + F4                                                   FRCDFL  1270
      ELOSS = RATE*(F2*F3+F4)                                              FRCDFL  1280
   99 FRCDF = F                                                            FRCDFL  1290
      RETURN                                                               FRCDFL  1300
      END                                                                  FRCDFL  1310
```

```
      SUBROUTINE FSMSOL (C,R,NN,MX,MAXN,JN,MAXDIM)                      FSMSOL  0010
C                                                    REV 20 04/11/80    FSMSOL  0020
C        SOLVES A SET OF SIMULTANEOUS EQUATIONS  OF SIZE 3*MM           FSMSOL  0030
C        WHERE THE MATRIX CONSISTS OF A SET OF 3*3 SUBMATRICES          FSMSOL  0040
C        STORED IN  C(3,3,IJ). THE LOCATION OF THE I,J ELEMENT          FSMSOL  0050
C        IS  STORED IN   NN(I,J).  I.E.  IJ= NN(I,J)                    FSMSOL  0060
C                                                                       FSMSOL  0070
C        A NEGATIVE IJ IMPLIES THAT C( , ,|IJ|) IS AN                   FSMSOL  0080
C        IDENTITY AND THE RIGHT SIDE IS ZERO. A NEGATIVE                FSMSOL  0090
C        IJ WILL ONLY OCCUR ON A DIAGONAL ENTRY OF NN.                  FSMSOL  0100
C                                                                       FSMSOL  0110
C            THE  BASIC EQUATION IS   CX=R                              FSMSOL  0120
C                                                                       FSMSOL  0130
C           DURING THE SOLUTION THE C MATRIX IS DESTROYED ,IT MAY       FSMSOL  0140
C           BE NECESSARY TO ADD TO THE C ARRAY.                         FSMSOL  0150
C           THE SOLUTION IS STORED IN R.                                FSMSOL  0160
C                                                                       FSMSOL  0170
C         INPUT                                                         FSMSOL  0180
C                                                                       FSMSOL  0190
C              C(3,3,K)  GIVEN ARRAY                                    FSMSOL  0200
C              R(3,MM)   GIVEN RIGHT HAND SIDE                          FSMSOL  0210
C              NN(JJ,JJ) GIVEN ARRAY CONTAINING LOCATIONS OF I,J,ELEMENT FSMSOL  0220
C              MX        SIZE OF SYSTEM OF SUBMATRICES (POSITIVE INDICATES FSMSOL  0230
C                        THAT C MATRIX IS SYMMETRIC, NEGATIVE IT IS NOT.) FSMSOL  0240
C              MAXN      LARGEST VALUE IN NN ARRAY                      FSMSOL  0250
C              JN        DIMENSION OF NN                                FSMSOL  0260
C              MAXDIM    THIRD DIMENSION OF C IN CALLING ROUTINE        FSMSOL  0270
C                                                                       FSMSOL  0280
      IMPLICIT REAL*8 (A-H,O-Z)                                         FSMSOL  0290
      DIMENSION C(3,3,1),R(3,1),NN(JN,1)                                FSMSOL  0300
      CALL ELTIME(1,20)                                                 FSMSOL  0310
      MM = IABS(MX)                                                     FSMSOL  0320
      IF  (MM.LE.0)  GO TO 99                                           FSMSOL  0330
      MM1 = MM-1                                                        FSMSOL  0340
      MP1 = MM+1                                                        FSMSOL  0350
      DO  50  II=1,MM                                                   FSMSOL  0360
      I = MP1-II                                                        FSMSOL  0370
C                                                                       FSMSOL  0380
C     START PIVOT AT BOTTOM - FIND PIVOT - INVERT.                      FSMSOL  0390
C                                                                       FSMSOL  0400
      L = NN(I,I)                                                       FSMSOL  0410
      IF  (L.LE.0)  GO TO 50                                            FSMSOL  0420
      DO  14  M=1,3                                                     FSMSOL  0430
      B = 1.0/C(M,M,L)                                                  FSMSOL  0440
      C(M,M,L) = 1.0                                                    FSMSOL  0450
      C(M,1,L) = B*C(M,1,L)                                             FSMSOL  0460
      C(M,2,L) = B*C(M,2,L)                                             FSMSOL  0470
      C(M,3,L) = B*C(M,3,L)                                             FSMSOL  0480
      R(M,I)   = B*R(M,I)                                               FSMSOL  0490
      DO  13  N=1,3                                                     FSMSOL  0500
```

```
      IF  (N.EQ.M)  GO TO 13                                                    FSMSOL  0510
      B = C(N,M,L)                                                              FSMSOL  0520
      C(N,M,L) = 0.0                                                            FSMSOL  0530
      C(N,1,L) = C(N,1,L) - B*C(M,1,L)                                          FSMSOL  0540
      C(N,2,L) = C(N,2,L) - B*C(M,2,L)                                          FSMSOL  0550
      C(N,3,L) = C(N,3,L) - B*C(M,3,L)                                          FSMSOL  0560
      R(N,I)   = R(N,I)   - B*R(M,I)                                            FSMSOL  0570
   13 CONTINUE                                                                  FSMSOL  0580
   14 CONTINUE                                                                  FSMSOL  0590
C                                                                               FSMSOL  0600
CC    CHECK IF DONE.                                                            FSMSOL  0610
C                                                                               FSMSOL  0620
      IF  (I.EQ.1)  GO TO 50                                                    FSMSOL  0630
      IM1 = I-1                                                                 FSMSOL  0640
C                                                                               FSMSOL  0650
CC    CALCULATE PIVOT ROW.                                                      FSMSOL  0660
C                                                                               FSMSOL  0670
      DO  20  J=1,IM1                                                           FSMSOL  0680
      IF  (NN(I,J).EQ.0)  GO TO 20                                              FSMSOL  0690
      M = NN(I,J)                                                               FSMSOL  0700
      DO  15  N=1,3                                                             FSMSOL  0710
      A = C(1,1,L)*C(1,N,M) + C(1,2,L)*C(2,N,M) + C(1,3,L)*C(3,N,M)             FSMSOL  0720
      B = C(2,1,L)*C(1,N,M) + C(2,2,L)*C(2,N,M) + C(2,3,L)*C(3,N,M)             FSMSOL  0730
      D = C(3,1,L)*C(1,N,M) + C(3,2,L)*C(2,N,M) + C(3,3,L)*C(3,N,M)             FSMSOL  0740
      C(1,N,M) = A                                                              FSMSOL  0750
      C(2,N,M) = B                                                              FSMSOL  0760
   15 C(3,N,M) = D                                                              FSMSOL  0770
   20 CONTINUE                                                                  FSMSOL  0780
C                                                                               FSMSOL  0790
CC    DONE WITH PIVOT ROW - ZERO COLUMN I ABOVE DIAGONAL.                       FSMSOL  0800
CC                                                                              FSMSOL  0810
CC    1,1                                                                       FSMSOL  0820
CC        .                                                                     FSMSOL  0830
CC           .                                                                  FSMSOL  0840
CC             K,K .  . K,J .  . K,I          C    = C    - C   *C             FSMSOL  0850
CC               .    .    .      .            KJ     KJ     KI  IJ            FSMSOL  0860
CC               .    .    .      .                                             FSMSOL  0870
CC             J,K .  . J,J .  . J,I          C    = C    - C   *C             FSMSOL  0880
CC               .      .    .    .            JK     JK     JI  IK            FSMSOL  0890
CC               .      .    .    .                                             FSMSOL  0900
CC             I,K .  . I,J .  . I,I          C    = 0                         FSMSOL  0910
CC                                  .          KI                              FSMSOL  0920
CC                                     .                                        FSMSOL  0930
CC                                       M,M                                   FSMSOL  0940
C                                                                               FSMSOL  0950
      DO  40  K=1,IM1                                                           FSMSOL  0960
      KI = NN(K,I)                                                              FSMSOL  0970
      IK = NN(I,K)                                                              FSMSOL  0980
      IF  (KI.EQ.0 .AND. IK.EQ.0)  GO TO 40                                     FSMSOL  0990
      DO  30  J=K,IM1                                                           FSMSOL  1000
```

```
      IJ = NN(I,J)                                                      FSMSOL  1010
      JI = NN(J,I)                                                      FSMSOL  1020
      IF  (KI.EQ.0  .OR. IJ.EQ.0)  GO TO 24                             FSMSOL  1030
      KJ = NN(K,J)                                                      FSMSOL  1040
      IF  (KJ.NE.0)  GO TO 22                                           FSMSOL  1050
      MAXN = MAXN+1                                                     FSMSOL  1060
      IF  (MAXN.GT.MAXDIM)  GO TO 41                                    FSMSOL  1070
      KJ = MAXN                                                         FSMSOL  1080
      NN(K,J) = KJ                                                      FSMSOL  1090
      DO  21  M=1,3                                                     FSMSOL  1100
      DO  21  N=1,3                                                     FSMSOL  1110
   21 C(N,M,KJ) = 0.0                                                   FSMSOL  1120
   22 DO  23  M=1,3                                                     FSMSOL  1130
      DO  23  N=1,3                                                     FSMSOL  1140
   23 C(N,M,KJ) = C(N,M,KJ) - C(N,1,KI)*C(1,M,IJ)                       FSMSOL  1150
     *                      - C(N,2,KI)*C(2,M,IJ)                      FSMSOL  1160
     *                      - C(N,3,KI)*C(3,M,IJ)                      FSMSOL  1170
   24 IF  (J.EQ.K)  GO TO 30                                            FSMSOL  1180
      IF  (JI.EQ.0  .OR. IK.EQ.0)  GO TO 30                             FSMSOL  1190
      JK = NN(J,K)                                                      FSMSOL  1200
      IF  (JK.NE.0)  GO TO 26                                           FSMSOL  1210
      MAXN = MAXN+1                                                     FSMSOL  1220
      IF  (MAXN.GT.MAXDIM)  GO TO 41                                    FSMSOL  1230
      JK = MAXN                                                         FSMSOL  1240
      NN(J,K) = JK                                                      FSMSOL  1250
      DO  25  M=1,3                                                     FSMSOL  1260
      DO  25  N=1,3                                                     FSMSOL  1270
   25 C(N,M,JK) = 0.0                                                   FSMSOL  1280
   26 IF  (MX.LT.0)  GO TO 28                                           FSMSOL  1290
      DO  27  M=1,3                                                     FSMSOL  1300
      DO  27  N=1,3                                                     FSMSOL  1310
   27 C(N,M,JK) = C(M,N,KJ)                                             FSMSOL  1320
      GO TO 30                                                          FSMSOL  1330
   28 DO  29  M=1,3                                                     FSMSOL  1340
      DO  29  N=1,3                                                     FSMSOL  1350
   29 C(N,M,JK) = C(N,M,JK) - C(N,1,JI)*C(1,M,IK)                       FSMSOL  1360
     *                      - C(N,2,JI)*C(2,M,IK)                      FSMSOL  1370
     *                      - C(N,3,JI)*C(3,M,IK)                      FSMSOL  1380
   30 CONTINUE                                                          FSMSOL  1390
      IF  (KI.EQ.0)  GO TO 40                                           FSMSOL  1400
      DO  35  N=1,3                                                     FSMSOL  1410
   35 R(N,K) = R(N,K) - C(N,1,KI)*R(1,I)                                FSMSOL  1420
     *                - C(N,2,KI)*R(2,I)                               FSMSOL  1430
     *                - C(N,3,KI)*R(3,I)                               FSMSOL  1440
   40 CONTINUE                                                          FSMSOL  1450
   50 CONTINUE                                                          FSMSOL  1460
      GO TO 51                                                          FSMSOL  1470
   41 WRITE  (6,49)  MAXDIM,(L,L=1,MM)                                  FSMSOL  1480
      DO  42  I=1,MM                                                    FSMSOL  1490
   42 WRITE  (6,43)  I,(NN(I,L),L=1,MM)                                 FSMSOL  1500
```

```
   43  FORMAT(I3,3X,40I3,3X/6X,40I3)                                          FSMSOL  1510
       WRITE  (6,44)                                                          FSMSOL  1520
   44  FORMAT('1 FSMSOL PRINT OF RHS ARRAY'//)                                FSMSOL  1530
       DO  45  K=1,MM                                                         FSMSOL  1540
   45  WRITE  (6,46)  K,(R(I,K),I=1,3)                                        FSMSOL  1550
   46  FORMAT(I6,9G14.7)                                                      FSMSOL  1560
       WRITE  (6,47)                                                          FSMSOL  1570
   47  FORMAT('1 FSMSOL PRINT OF C ARRAY ELEMENTS'//)                         FSMSOL  1580
       DO  48  K=1,MAXN                                                       FSMSOL  1590
   48  WRITE  (6,46)  K,((C(I,L,K),L=1,3),I=1,3)                              FSMSOL  1600
   49  FORMAT('1 MAXIMUM DIMENSION OF',I4,' ON C ARRAY HAS BEEN EXCEEDED      FSMSOL  1610
      *IN SUBROUTINE FSMSOL.'//' IF 400, CALL IS FROM SUBROUTINE DAUX. IF     FSMSOL  1620
      *200 CALL IS FROM SUBROUTINE HPTURB.'//' PROGRAM IS BEING TERMINATE     FSMSOL  1630
      *D. COMPLETE PRINT-OUT OF IJK, RHS AND C ARRAYS FOLLOW.'//              FSMSOL  1640
      *' FSMSOL PRINT OF IJK MATRIX'//(6X,40I3))                             FSMSOL  1650
       STOP 35                                                                FSMSOL  1660
C                                                                             FSMSOL  1670
C      BACKDOWN SOLUTION                                                      FSMSOL  1680
C                                                                             FSMSOL  1690
   51  IF  (MM.EQ.1)  GO TO 99                                                FSMSOL  1700
       DO  90  J=1,MM1                                                        FSMSOL  1710
       IP = J+1                                                               FSMSOL  1720
       DO  80  I=IP,MM                                                        FSMSOL  1730
       IF  (NN(I,J).EQ.0)  GO TO 80                                           FSMSOL  1740
       IJ = NN(I,J)                                                           FSMSOL  1750
       DO  75  N=1,3                                                          FSMSOL  1760
   75  R(N,I) = R(N,I) - C(N,1,IJ)*R(1,J)                                     FSMSOL  1770
      *                - C(N,2,IJ)*R(2,J)                                     FSMSOL  1780
      *                - C(N,3,IJ)*R(3,J)                                     FSMSOL  1790
   80  CONTINUE                                                               FSMSOL  1800
   90  CONTINUE                                                               FSMSOL  1810
   99  CALL ELTIME(2,20)                                                      FSMSOL  1820
       RETURN                                                                 FSMSOL  1830
       END                                                                    FSMSOL  1840
```

```
      SUBROUTINE GLOBAL (J,HD3,DH1,TQC,T9,ANGL)                                 GLOBAL  0010
C                                                          REV 19 10/19/79      GLOBAL  0020
      IMPLICIT REAL*8 (A-H,O-Z)                                                 GLOBAL  0030
      DIMENSION HD3(3),DH1(3,3),T9(3),ANGL(3),CC(3)                             GLOBAL  0040
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(3,60),HA(3,60),HB(3,60),         GLOBAL  0050
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),             GLOBAL  0060
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)            GLOBAL  0070
      COMMON/TABLES/ MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(500),TAB(2600)        GLOBAL  0080
      COMMON/TEMPVI/ CREST,TTI(3),R1I(3),R2I(3),JSTOP(4,2,30)                   GLOBAL  0090
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                                 GLOBAL  0100
     *               UNITL,UNITM,UNITT,GRAVTY(3)                                GLOBAL  0110
      IF (DABS(HD3(3)).GT.1.0-EPS(6)) GO TO 34                                  GLOBAL  0120
      ANGL(1) = DARCOS(HD3(3))                                                  GLOBAL  0130
      NT = IGLOB(J)                                                             GLOBAL  0140
      NT1 = NTAB(NT+2)                                                          GLOBAL  0150
      CALL HERRON(HD3,NT1,THET0,THET0P)                                         GLOBAL  0160
      JSTOP(4,1,J) = 0                                                          GLOBAL  0170
      IF (ANGL(1).LE.THET0) GO TO 34                                            GLOBAL  0180
      JSTOP(4,1,J) = 1                                                          GLOBAL  0190
      MT = NTAB(NT+5)                                                           GLOBAL  0200
      CREST = TAB(MT+3)                                                         GLOBAL  0210
      STH2 = 1.0-HD3(3)**2                                                      GLOBAL  0220
      STH = DSQRT(STH2)                                                         GLOBAL  0230
      CTH = HD3(3)/STH                                                          GLOBAL  0240
      CST = DSQRT(STH2+THET0P**2)                                               GLOBAL  0250
      DR  = (ANGL(1)-THET0)*STH/CST                                             GLOBAL  0260
      LT  = NTAB(NT)                                                            GLOBAL  0270
      TAB(LT) = DR                                                              GLOBAL  0280
      NTAB(NT+2) = 0                                                            GLOBAL  0290
      DRDOT = 0.0                                                               GLOBAL  0300
      CALL FRCDFL (DR,DRDOT,NT,1,TQF,ELOSS)                                     GLOBAL  0310
      NTAB(NT+2) = NT1                                                          GLOBAL  0320
      TQC = TQF/CST                                                             GLOBAL  0330
      CC(1) = -HD3(2)+HD3(1)*CTH*THET0P                                         GLOBAL  0340
      CC(2) =  HD3(1)+HD3(2)*CTH*THET0P                                         GLOBAL  0350
      CC(3) =                -STH*THET0P                                        GLOBAL  0360
      DO 28 L=1,3                                                               GLOBAL  0370
   28 T9(L) = CC(1)*DH1(L,1) + CC(2)*DH1(L,2) + CC(3)*DH1(L,3)                  GLOBAL  0380
   34 RETURN                                                                    GLOBAL  0390
      END                                                                       GLOBAL  0400
```

```
      SUBROUTINE HBELT (J1,J2,KNL0,IND)                                  HBELT   0010
C                                                        REV 20 05/18/80  HBELT   0020
C     ARGUMENTS:                                                         HBELT   0030
C         J1,J2 - FIRST AND LAST INDEX FOR BELTS.                        HBELT   0040
C          KNL0 - ZERO VALUE FOR KNL INDEX.                              HBELT   0050
C          IND  - 0: CALL IS FROM SUBROUTINE CONTCT                      HBELT   0060
C                 1: CALL IS FROM SUBROUTINE UPDATE                      HBELT   0070
C                                                                        HBELT   0080
      IMPLICIT REAL*8 (A-H,O-Z)                                          HBELT   0090
      COMMON/CNTSRF/ PL(17,30),BELT(20,8),TPTS(6,8),BD(24,40)            HBELT   0100
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30), HBELT   0110
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       HBELT   0120
      COMMON/TABLES/ MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(500),TAB(2600) HBELT   0130
      COMMON/FORCES/ PSF(7,30),BSF(4,20),SSF(10,20),BAGSF(3,20),         HBELT   0140
     *               PRJNT(7,30),NPANEL(5),NPSF,NBSF,NSSF,NBGSF         HBELT   0150
      COMMON/HRNESS/ BAR(15,100),BB(100),BBDOT(100),PLOSS(2,100),        HBELT   0160
     *               XLONG(20),HTIME(2),IBAR(5,100),NL(2,100),          HBELT   0170
     *               NPTSPB(20),NPTPLY(20),NTHRNS(20),NBLTPH(5)         HBELT   0180
C     THIS COMMON/TEMPVS/ IS SHARED BY HPTURB, HBPLAY, HBELT AND HSETC.  HBELT   0190
      COMMON/TEMPVS/ B(3,3,3),S(3,3),T(3),R(3),V(3),T1(3),T2(3),         HBELT   0200
     *               E(3,3,50),EDOT(3,50),FCE(3,50),FR(3,50),ZR(3,50),  HBELT   0210
     *               TR(3,50),U(3,50),PTLOSS(2,50),BL(50),FB(50),FP(50), HBELT   0220
     *               OLDBB(100),RHS(3,54),C(3,3,200),IJK(54,54)         HBELT   0230
      CALL ELTIME (1,38)                                                 HBELT   0240
      NTP = 0                                                            HBELT   0250
      K2 = 0                                                             HBELT   0260
      DO  31  JB=J1,J2                                                   HBELT   0270
      IF  (IND.EQ.0)  NBSF = NBSF + 1                                    HBELT   0280
      IF  (NPTPLY(JB).LE.0)  GO TO 31                                    HBELT   0290
C                                                                        HBELT   0300
C     FIRST LOOP ON K                                                    HBELT   0310
C     COMPUTE Z(K),ZR(K),E3(K),U(K-1),BL(K-1),FB(K-1)                    HBELT   0320
C     NEED    NL(K),BB(K-1)                                              HBELT   0330
C     NOTE: AN INDEX K-1 REFERS TO BELT SEGMENT BETWEEN K-1 AND K.       HBELT   0340
C                                                                        HBELT   0350
      K1 = K2 + 1                                                        HBELT   0360
      K2 = K2 + NPTPLY(JB)                                               HBELT   0370
      DO  20  K=K1,K2                                                    HBELT   0380
      KNL = KNL0 + K                                                     HBELT   0390
      KI = NL(1,KNL)                                                     HBELT   0400
C                                                                        HBELT   0410
C     HERE K   IS INDEX OF POINTS IN PLAY ON EACH HARNESS                HBELT   0420
C          KNL IS INDEX OF ALL POINTS IN PLAY                            HBELT   0430
C          KI  IS INDEX OF ALL POINTS                                    HBELT   0440
C                                                                        HBELT   0450
      KS   = IABS(IBAR(1,KI))                                            HBELT   0460
      IF  (KS.GT.100)  NTP = 1                                           HBELT   0470
      IF  (KS.GT.100)  KS = MOD(KS,100)                                  HBELT   0480
      KE = IBAR(2,KI)                                                    HBELT   0490
      CALL DOT31 (D(1,1,KS),BAR(4,KI),T1)                                HBELT   0500
```

```
      CALL DOT31 (D(1,1,KS),BAR(7,KI),T2)                                    HBELT   0510
      DO  11  J=1,3                                                          HBELT   0520
      R(J) = V(J)                                                            HBELT   0530
      V(J) = BAR(J+3,KI) + BAR(J+6,KI)                                       HBELT   0540
      TR(J,K) = T1(J)                                                        HBELT   0550
      ZR(J,K) = T1(J) + T2(J)                                                HBELT   0560
      S (J,2) = S(J,1)                                                       HBELT   0570
   11 S (J,1) = SEGLP(J,KS) + ZR(J,K)                                        HBELT   0580
      CALL CROSS (WMEG(1,KS),V,T)                                            HBELT   0590
      IF  (KE.EQ.0)  GO TO 12                                                HBELT   0600
      CALL MAT31 (BD(7,KE),BAR(4,KI),T2)                                     HBELT   0610
      CALL DOT31 (D(1,1,KS),T2,T1)                                           HBELT   0620
   12 DO  13  J=1,3                                                          HBELT   0630
      T(J) = T(J) + BAR(J+12,KI)                                             HBELT   0640
   13 E(J,3,K) = T1(J)                                                       HBELT   0650
      CALL DOT31 (D(1,1,KS),T,V)                                             HBELT   0660
      DO  14  J=1,3                                                          HBELT   0670
   14 V(J) = V(J) + SEGLV(J,KS)                                              HBELT   0680
      FB(K) = 0.0                                                            HBELT   0690
      FP(K) = 0.0                                                            HBELT   0700
      IF  (K.EQ.K1)  GO TO 20                                                HBELT   0710
      DO  15  J=1,3                                                          HBELT   0720
   15 U(J,K-1) = S(J,1) - S(J,2)                                             HBELT   0730
      BL(K-1) = DSQRT(U(1,K-1)**2 + U(2,K-1)**2 + U(3,K-1)**2)               HBELT   0740
      DO  16  J=1,3                                                          HBELT   0750
   16 U(J,K-1) = U(J,K-1)/BL(K-1)                                            HBELT   0760
      STRAIN = (BL(K-1)/BB(KNL-1)) - 1.0                                     HBELT   0770
      IF  (STRAIN.LT.0.0)  STRAIN = 0.0                                      HBELT   0780
      NT = NL(2,KNL)                                                         HBELT   0790
      BLDOT = U(1,K-1)*(V(1)-R(1))                                           HBELT   0800
     *      + U(2,K-1)*(V(2)-R(2))                                           HBELT   0810
     *      + U(3,K-1)*(V(3)-R(3))                                           HBELT   0820
      STRDOT = (BB(KNL-1)*BLDOT-BL(K-1)*BBDOT(KNL-1))/BB(KNL-1)**2           HBELT   0830
      CALL FRCDFL (STRAIN,STRDOT,NT,0,FPK,ELOSS)                             HBELT   0840
      CALL FRCDFL (STRAIN,STRDOT,NT,1,FBK,ELOSS)                             HBELT   0850
      PTLOSS(1,K-1) = BB(KNL-1)*ELOSS                                        HBELT   0860
      FP(K-1) = FPK                                                          HBELT   0870
      FB(K-1) = FBK                                                          HBELT   0880
      IF  (IND.NE.0)  GO TO 20                                               HBELT   0890
      IBSF = 0                                                               HBELT   0900
      IF  (K.EQ.K1+1)  IBSF = 1                                              HBELT   0910
      IF  (K.EQ.K2  )  IBSF = 3                                              HBELT   0920
      IF  (IBSF.EQ.0)  GO TO 20                                              HBELT   0930
      BSF(IBSF  ,NBSF) = STRAIN                                              HBELT   0940
      BSF(IBSF+1,NBSF) = FBK                                                 HBELT   0950
   20 CONTINUE                                                               HBELT   0960
C                                                                            HBELT   0970
C     SECOND LOOP ON K                                                       HBELT   0980
C     COMPUTE FCE(K),E1(K),E2(K),EDOT(K),FR(K),U1(KS),U2(KS)                 HBELT   0990
C     NEED    FB(K&K-1),U(K&K-1),ZR(K),E3(K)                                 HBELT   1000
```

```
C                                                                       HBELT   1010
      DO  30  K=K1,K2                                                   HBELT   1020
      KNL = KNL0 + K                                                    HBELT   1030
      KI = NL(1,KNL)                                                    HBELT   1040
      KS   = IABS(IBAR(1,KI))                                           HBELT   1050
      IF  (KS.GT.100)  KS = MOD(KS,100)                                 HBELT   1060
      DO  21  J=1,3                                                     HBELT   1070
      FCE(J,K) = FB(K)*U(J,K)                                           HBELT   1080
   21 IF  (K.NE.K1)  FCE(J,K) = FCE(J,K) - FB(K-1)*U(J,K-1)             HBELT   1090
      NT = IBAR(3,KI)                                                   HBELT   1100
      NF = NTAB(NT+5)                                                   HBELT   1110
      IF  (NF.EQ.0 .AND. IND.EQ.0)  GO TO 30                            HBELT   1120
      IF  (IBAR(4,KI).EQ.0)  GO TO 22                                   HBELT   1130
      CALL DOT31 (D(1,1,KS),BAR(10,KI),T1)                              HBELT   1140
      GO TO 24                                                          HBELT   1150
   22 DO  23  J=1,3                                                     HBELT   1160
      T1(J) = 0.0                                                       HBELT   1170
      IF  (K.NE.K2)  T1(J) = U(J,K)                                     HBELT   1180
   23 IF  (K.NE.K1)  T1(J) = T1(J) + U(J,K-1)                           HBELT   1190
   24 CALL CROSS (T1,E(1,3,K),E(1,1,K))                                 HBELT   1200
      CALL CROSS (E(1,3,K),E(1,1,K),E(1,2,K))                           HBELT   1210
      DO  25  J=1,3                                                     HBELT   1220
      EDOT(J,K) = DSQRT(E(1,J,K)**2 + E(2,J,K)**2 + E(3,J,K)**2)        HBELT   1230
      DO  25  I=1,3                                                     HBELT   1240
   25 E(I,J,K) = E(I,J,K)/EDOT(J,K)                                     HBELT   1250
      CALL DOT31 (E(1,1,K),FCE(1,K),FR(1,K))                            HBELT   1260
   30 CONTINUE                                                          HBELT   1270
   31 CONTINUE                                                          HBELT   1280
      IF  (NTP.LE.0)  GO TO 41                                          HBELT   1290
C                                                                       HBELT   1300
C     SUM FCE,FR FOR TIE-POINTS                                         HBELT   1310
C                                                                       HBELT   1320
      KNL1 = KNL0 + 2                                                   HBELT   1330
      KNL2 = KNL0 + K2                                                  HBELT   1340
      DO  40  KNL=KNL1,KNL2                                             HBELT   1350
      KI = NL(1,KNL)                                                    HBELT   1360
      KS   = IABS(IBAR(1,KI))                                           HBELT   1370
      IF  (KS.LT.100)  GO TO 40                                         HBELT   1380
      KS1 = KS/100                                                      HBELT   1390
      KH = KNL - KNL0                                                   HBELT   1400
      MH = 0                                                            HBELT   1410
      DO  38  JNL=KNL1,KNL                                              HBELT   1420
      KI = NL(1,JNL-1)                                                  HBELT   1430
      KS   = IABS(IBAR(1,KI))                                           HBELT   1440
      IF  (KS.LT.100)  GO TO 38                                         HBELT   1450
      KS2 = KS/100                                                      HBELT   1460
      IF  (KS2.NE.KS1)  GO TO 38                                        HBELT   1470
      JH = JNL-1 - KNL0                                                 HBELT   1480
      IF  (MH.EQ.0)  MH = JH                                            HBELT   1490
      DO  37  J=1,3                                                     HBELT   1500
```

```
      IF  (MH.EQ.JH)  FCE(J,MH) = FCE(J,MH) + FCE(J,KH)                  HBELT   1510
   37 FCE(J,JH) = FCE(J,MH)                                              HBELT   1520
      CALL DOT31 (E(1,1,JH),FCE(1,JH),FR(1,JH))                          HBELT   1530
   38 CONTINUE                                                           HBELT   1540
      IF  (MH.EQ.0)  GO TO 40                                            HBELT   1550
      KI = NL(1,KNL)                                                     HBELT   1560
      IBAR(1,KI) = -IABS(IBAR(1,KI))                                     HBELT   1570
      DO  39  J=1,3                                                      HBELT   1580
   39 FCE(J,KH) = FCE(J,MH)                                              HBELT   1590
      CALL DOT31 (E(1,1,KH),FCE(1,KH),FR(1,KH))                          HBELT   1600
   40 CONTINUE                                                           HBELT   1610
C                                                                        HBELT   1620
C     IF CALL IS FROM SUBROUTINE CONTCT,                                 HBELT   1630
C     ADD FORCES (FCE) MODIFIED BY FRICTION TO U1,U2 ARRAYS.             HBELT   1640
C                                                                        HBELT   1650
   41 IF  (IND.NE.0)  GO TO 52                                           HBELT   1660
      K2 = 0                                                             HBELT   1670
      DO  51  JB=J1,J2                                                   HBELT   1680
      IF  (NPTPLY(JB).LE.0)  GO TO 51                                    HBELT   1690
      K1 = K2 + 1                                                        HBELT   1700
      K2 = K2 + NPTPLY(JB)                                               HBELT   1710
      DO  50  K=K1,K2                                                    HBELT   1720
      KNL = KNL0 + K                                                     HBELT   1730
      KI = NL(1,KNL)                                                     HBELT   1740
      IF  (IBAR(1,KI).LT.0)  GO TO 50                                    HBELT   1750
      KS = IBAR(1,KI)                                                    HBELT   1760
      IF  (KS.GT.100)  KS = MOD(KS,100)                                  HBELT   1770
      NT = IBAR(3,KI)                                                    HBELT   1780
      NF = NTAB(NT+5)                                                    HBELT   1790
      IF  (NF.EQ.0)  GO TO 43                                            HBELT   1800
      DO  42  J=1,3                                                      HBELT   1810
   42 T1(J) = FR(J,K)                                                    HBELT   1820
      FR1 = TAB(NF+2)*DABS(T1(3))                                        HBELT   1830
      FR2 = TAB(NF+4)*DABS(T1(3))                                        HBELT   1840
      IF  (DABS(T1(1)).GT.FR1)  T1(1) = DSIGN(FR1,T1(1))                 HBELT   1850
      IF  (DABS(T1(2)).GT.FR2)  T1(2) = DSIGN(FR2,T1(2))                 HBELT   1860
      CALL MAT31 (E(1,1,K),T1,FCE(1,K))                                  HBELT   1870
   43 CALL CROSS (ZR(1,K),FCE(1,K),T2)                                   HBELT   1880
      CALL MAT31 (D(1,1,KS),T2,T1)                                       HBELT   1890
      DO  44  J=1,3                                                      HBELT   1900
      U1(J,KS) = U1(J,KS) + FCE(J,K)                                     HBELT   1910
   44 U2(J,KS) = U2(J,KS) + T1(J)                                        HBELT   1920
   50 CONTINUE                                                           HBELT   1930
   51 CONTINUE                                                           HBELT   1940
   52 KNL0 = KNL0 + K2                                                   HBELT   1950
      CALL ELTIME (2,38)                                                 HBELT   1960
      RETURN                                                             HBELT   1970
      END                                                                HBELT   1980
```

```
      SUBROUTINE HBPLAY                                                 HBPLAY  0010
C                                                          REV 20 06/11/80  HBPLAY  0020
      IMPLICIT REAL*8 (A-H,O-Z)                                         HBPLAY  0030
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           HBPLAY  0040
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)        HBPLAY  0050
      COMMON/CNTSRF/ PL(17,30),BELT(20,8),TPTS(6,8),BD(24,40)           HBPLAY  0060
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30), HBPLAY  0070
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       HBPLAY  0080
      COMMON/HRNESS/ BAR(15,100),BB(100),BBDOT(100),PLOSS(2,100),       HBPLAY  0090
     *               XLONG(20),HTIME(2),IBAR(5,100),NL(2,100),          HBPLAY  0100
     *               NPTSPB(20),NPTPLY(20),NTHRNS(20),NBLTPH(5)         HBPLAY  0110
C     THIS COMMON/TEMPVS/ IS SHARED BY HPTURB, HBPLAY, HBELT AND HSETC. HBPLAY  0120
      COMMON/TEMPVS/ B(3,3,3),S(3,3),T(3),R(3),V(3),T1(3),T2(3),        HBPLAY  0130
     *               E(3,3,50),EDOT(3,50),FCE(3,50),FR(3,50),ZR(3,50),  HBPLAY  0140
     *               TR(3,50),U(3,50),PTLOSS(2,50),BL(50),FB(50),FP(50), HBPLAY  0150
     *               OLDBB(100),RHS(3,54),C(3,3,200),IJK(54,54)         HBPLAY  0160
      IF  (NHRNSS.LE.0)  GO TO 99                                       HBPLAY  0170
C                                                                       HBPLAY  0180
C     SAVE PREVIOUS NL,BB AND PLOSS ARRAYS.                             HBPLAY  0190
C     USE IJK,OLDBB AND PTLOSS AS TEMP STORAGE.                         HBPLAY  0200
C                                                                       HBPLAY  0210
      DO  10  I=1,100                                                   HBPLAY  0220
      IJK(I,1) = NL(1,I)                                                HBPLAY  0230
      PTLOSS(I,1) = PLOSS(1,I)                                          HBPLAY  0240
   10 OLDBB(I) = BB(I)                                                  HBPLAY  0250
      JNL = 1                                                           HBPLAY  0260
      J1 = 1                                                            HBPLAY  0270
      K1 = 1                                                            HBPLAY  0280
      LL = 0                                                            HBPLAY  0290
      DO  90  NH=1,NHRNSS                                               HBPLAY  0300
      IF  (NBLTPH(NH).LE.0)  GO TO 90                                   HBPLAY  0310
      J2 = J1 + NBLTPH(NH) - 1                                          HBPLAY  0320
      DO  80  NB=J1,J2                                                  HBPLAY  0330
      L1 = LL                                                           HBPLAY  0340
      IF  (NPTSPB(NB).LE.0)  GO TO 80                                   HBPLAY  0350
      K2 = K1 + NPTSPB(NB) - 1                                          HBPLAY  0360
      KB = 0                                                            HBPLAY  0370
      DO  30  K=K1,K2                                                   HBPLAY  0380
      KB = KB + 1                                                       HBPLAY  0390
C                                                                       HBPLAY  0400
C     HERE K IS INDEX OF ALL POINTS                                     HBPLAY  0410
C         KB IS INDEX OF POINTS ON A SINGLE BELT                        HBPLAY  0420
C         LL IS INDEX OF ALL POINTS IN PLAY                             HBPLAY  0430
C         JB IS INDEX OF PREVIOUS POINT ON BELT IN PLAY                 HBPLAY  0440
C                                                                       HBPLAY  0450
      KS = IABS(IBAR(1,K))                                              HBPLAY  0460
      IF  (KS.GT.100)  KS = MOD(KS,100)                                 HBPLAY  0470
      KE = IBAR(2,K)                                                    HBPLAY  0480
      CALL DOT31 (D(1,1,KS),BAR(4,K),T1)                                HBPLAY  0490
      CALL DOT31 (D(1,1,KS),BAR(7,K),T2)                                HBPLAY  0500
```

```
      DO  11  J=1,3                                                    HBPLAY  0510
   11 U(J,KB) = SEGLP(J,KS) + T1(J) + T2(J)                            HBPLAY  0520
      IF  (K.EQ.K1)  GO TO 30                                          HBPLAY  0530
      LL = LL + 1                                                      HBPLAY  0540
   12 JJ = NL(1,LL)                                                    HBPLAY  0550
      JB = JJ - K1 + 1                                                 HBPLAY  0560
      DSS = 0.0                                                        HBPLAY  0570
      DO  13  J=1,3                                                    HBPLAY  0580
      ZR(J,KB) = U(J,KB) - U(J,JB)                                     HBPLAY  0590
   13 DSS = DSS + ZR(J,KB)**2                                          HBPLAY  0600
      BL(LL) = DSQRT(DSS)                                              HBPLAY  0610
      IF  (JJ.EQ.K1 .OR. IABS(IBAR(1,JJ)).GT.100)  GO TO 30            HBPLAY  0620
      JS = IBAR(1,JJ)                                                  HBPLAY  0630
      JE = IBAR(2,JJ)                                                  HBPLAY  0640
      IF  (JE.LE.0)  GO TO 30                                          HBPLAY  0650
      CALL MAT31 (BD(7,JE),BAR(4,JJ),T2)                               HBPLAY  0660
      CALL DOT31 (D(1,1,JS),T2,R)                                      HBPLAY  0670
      DPR = 0.0                                                        HBPLAY  0680
      DO  17  J=1,3                                                    HBPLAY  0690
   17 DPR = DPR + R(J)*(ZR(J,KB)/BL(LL) - ZR(J,JB)/BL(LL-1))           HBPLAY  0700
      IF  (DPR.LT.0.0)  GO TO 30                                       HBPLAY  0710
      LL = LL - 1                                                      HBPLAY  0720
      GO TO 12                                                         HBPLAY  0730
   30 NL(1,LL+1) = K                                                   HBPLAY  0740
      L2 = L1 + 1                                                      HBPLAY  0750
      LL = LL + 1                                                      HBPLAY  0760
      L3 = LL-1                                                        HBPLAY  0770
      DO  31  J=L2,LL                                                  HBPLAY  0780
   31 NL(2,J) = NTHRNS(NB)                                             HBPLAY  0790
      IF  (XLONG(NB).EQ.0.0)  GO TO 35                                 HBPLAY  0800
C                                                                      HBPLAY  0810
C     FIRST TIME IN ROUTINE, SET INITIAL BB ARRAY.                     HBPLAY  0820
C     INPUT XLONG MUST BE NON-ZERO TO TRIGGER THIS TEST.               HBPLAY  0830
C                                                                      HBPLAY  0840
      XLG = 0.0                                                        HBPLAY  0850
      DO  32  J=L2,L3                                                  HBPLAY  0860
   32 XLG = XLG + BL(J)                                                HBPLAY  0870
      XLG = 1.0 + XLONG(NB)/XLG                                        HBPLAY  0880
      DO  33  J=L2,L3                                                  HBPLAY  0890
   33 BB(J) = XLG*BL(J)                                                HBPLAY  0900
      XLONG(NB) = 0.0                                                  HBPLAY  0910
      GO TO 52                                                         HBPLAY  0920
C                                                                      HBPLAY  0930
C     DETERMINE IF NEW NL ARRAY IS DIFFERENT FROM PREVIOUS NL ARRAY.   HBPLAY  0940
C     IF SO, RECOMPUTE BB ELEMENTS FOR POINTS THAT ARE DIFFERENT.      HBPLAY  0950
C                                                                      HBPLAY  0960
   35 IF  (NL(1,L2).EQ.IJK(JNL,1))  GO TO 61                           HBPLAY  0970
      WRITE (6,62)                                                     HBPLAY  0980
   62 FORMAT ('0 LOGIC ERROR IN SUB HBPLAY. PROGRAM TERMINATED.')      HBPLAY  0990
      STOP 42                                                          HBPLAY  1000
```

```
   61  LTEST = 0                                                              HBPLAY  1010
       M = L2                                                                 HBPLAY  1020
       N = JNL                                                                HBPLAY  1030
   36  IF  (NL(1,M+1)-IJK(N+1,1))  39,37,41                                   HBPLAY  1040
   37  BB(M) = OLDBB(N)                                                       HBPLAY  1050
       PLOSS(1,M) = PTLOSS(N,1)                                               HBPLAY  1060
   38  M = M+1                                                                HBPLAY  1070
       N = N+1                                                                HBPLAY  1080
       IF  (M-LL)  36,51,51                                                   HBPLAY  1090
C                                                                             HBPLAY  1100
C      POINT M+1 IS NEW.                                                      HBPLAY  1110
C                                                                             HBPLAY  1120
   39  M0 = M                                                                 HBPLAY  1130
       N0 = N                                                                 HBPLAY  1140
       LTEST = 1                                                              HBPLAY  1150
   40  M = M+1                                                                HBPLAY  1160
       GO TO 43                                                               HBPLAY  1170
C                                                                             HBPLAY  1180
C       POINT N+1 IS DROPPED.                                                 HBPLAY  1190
C                                                                             HBPLAY  1200
   41  M0 = M                                                                 HBPLAY  1210
       N0 = N                                                                 HBPLAY  1220
       LTEST = 1                                                              HBPLAY  1230
   42  N = N+1                                                                HBPLAY  1240
   43  IF  (NL(1,M+1)-IJK(N+1,1))  40,44,42                                   HBPLAY  1250
C                                                                             HBPLAY  1260
C      POINTS N0 TO N+1 ARE BEING REPLACED WITH POINTS M0 TO M+1.             HBPLAY  1270
C                                                                             HBPLAY  1280
   44  SUMBL = 0.0                                                            HBPLAY  1290
       DO  45  J=M0,M                                                         HBPLAY  1300
   45  SUMBL = SUMBL + BL(J)                                                  HBPLAY  1310
       SUMPL = 0.0                                                            HBPLAY  1320
       SUMBB = 0.0                                                            HBPLAY  1330
       DO  46  J=N0,N                                                         HBPLAY  1340
       SUMPL = SUMPL + PTLOSS(J,1)                                            HBPLAY  1350
   46  SUMBB = SUMBB + OLDBB(J)                                               HBPLAY  1360
       RATPL = SUMPL/SUMBL                                                    HBPLAY  1370
       RATIO = SUMBB/SUMBL                                                    HBPLAY  1380
       DO  47  J=M0,M                                                         HBPLAY  1390
       PLOSS(1,J) = RATPL*BL(J)                                               HBPLAY  1400
   47  BB(J) = RATIO*BL(J)                                                    HBPLAY  1410
       GO TO 38                                                               HBPLAY  1420
   51  JNL = N+1                                                              HBPLAY  1430
       IF  (LTEST.EQ.0)  GO TO 79                                             HBPLAY  1440
C                                                                             HBPLAY  1450
C      PRINT NEW POINT ARRAY IF DIFFERENT.                                    HBPLAY  1460
C                                                                             HBPLAY  1470
   52  NPTS = LL - L1                                                         HBPLAY  1480
       USEC = 1000.0*TIME                                                     HBPLAY  1490
       WRITE  (6,53)  USEC,NH,NB,NPTS,NTHRNS(NB)                              HBPLAY  1500
   53  FORMAT ('0 HBPLAY TIME =',F10.3,' MSEC. NH,NB,NPTS NT=',4I6)           HBPLAY  1510
       WRITE  (6,54)  (NL(1,J),J=L2,LL)                                       HBPLAY  1520
   54  FORMAT ('  NL(1)=',15I8/(8X,15I8))                                     HBPLAY  1530
       WRITE  (6,55)  (BB(J),J=L2,L3)                                         HBPLAY  1540
   55  FORMAT ('  BB   =',6X,14F8.3/(6X,15F8.3))                              HBPLAY  1550
   79  K1 = K2 + 1                                                            HBPLAY  1560
   80  NPTPLY(NB) = LL - L1                                                   HBPLAY  1570
       J1 = J2 + 1                                                            HBPLAY  1580
   90  CONTINUE                                                               HBPLAY  1590
   99  RETURN                                                                 HBPLAY  1600
       END                                                                    HBPLAY  1610
```

```
      SUBROUTINE HEDING (LINES,LPP)                                          HEDING  0010
C                                                          REV 20 05/18/80    HEDING  0020
      IMPLICIT REAL*8 (A-H,O-Z)                                              HEDING  0030
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,                HEDING  0040
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)             HEDING  0050
      COMMON/JBARTZ/ MNPL(   30),MNBLT(    8),MNSEG(   30),MNBAG(    6),   HEDING  0060
     *               MPL(3,5,30),MBLT(3,5,8),MSEG(3,5,30),MBAG(3,10,6),     HEDING  0070
     *               NTPL( 5,30),NTBLT( 5,8),NTSEG( 5,30)                    HEDING  0080
      COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),                HEDING  0090
     *               BLTTTL(5,8),PLTTL(5,30),BAGTTL(5,6),SEG(30),            HEDING  0100
     *               JOINT(30),CGS(30),JS(30)                                HEDING  0110
      REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT           HEDING  0120
      LOGICAL*1 CGS,JS                                                       HEDING  0130
      COMMON/FORCES/ PSF(7,30),BSF(4,20),SSF(10,20),BAGSF(3,20),             HEDING  0140
     *               PRJNT(7,30),NPANEL(5),NPSF,NBSF,NSSF,NBGSF              HEDING  0150
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                              HEDING  0160
     *               UNITL,UNITM,UNITT,GRAVTY(3)                             HEDING  0170
      COMMON/RSAVE/  XSG(3,20,3),DPMI(3,3,30),LPMI(30),NSG(7),MSG(20,7)      HEDING  0180
      COMMON/DAMPER/ APSDM(3,20),APSDN(3,20),ASD(5,20),MSDM(20),MSDN(20)     HEDING  0190
      COMMON/HRNESS/ BAR(15,100),BB(100),BBDOT(100),PLOSS(2,100),            HEDING  0200
     *               XLONG(20),HTIME(2),IBAR(5,100),NL(2,100),               HEDING  0210
     *               NPTSPB(20),NPTPLY(20),NTHRNS(20),NBLTPH(5)              HEDING  0220
C     NOTE: SUBROUTINES POSTPR & HEDING SHARE THIS COMMON/TEMPVS/.            HEDING  0230
C     SEE COMMENT IN POSTPR ABOUT FIRST DIMENSION OF PLDATA.                  HEDING  0240
      REAL   HEAD , PHED , BLANK , PLDATA , USEC , ZTTH                      HEDING  0250
      COMMON/TEMPVS/ TDATA(14,50),HEAD(20),NOPL(100),MOPL(100),              HEDING  0260
     *               PLDATA(97,20) , USEC(45) , ZTTH(14,45,2)                HEDING  0270
      LOGICAL   LOLD , LNEW                                                  HEDING  0280
      DIMENSION PHED(5),HEDJ(4,2),HEADJJ(4,2)                                HEDING  0290
      DATA HEDJ/8HIPIN  FL,8HEXURE  A,8HZIMUTH  ,8HTORSION ,                 HEDING  0300
     *          8HIEULER  ,8HPREC.  N,8HUTATION ,8H  SPIN  /                 HEDING  0310
      DATA BLANK/4H    /                                                     HEDING  0320
      DATA PHED/4HSPRF,4HPNL1,4HPNL2,4HPNL3,4HPNL4/                          HEDING  0330
      NPRT4 = NPRT(4) + 4                                                    HEDING  0340
      IF  (NPRT4.LE.0 .OR. NPRT4.GT.8)  STOP 40                              HEDING  0350
      GO TO  (11,11,82,12,12,11,11,12) , NPRT4                               HEDING  0360
   11 LOLD = .FALSE.                                                         HEDING  0370
      LNEW = .TRUE.                                                          HEDING  0380
      GO TO 13                                                               HEDING  0390
   12 LOLD = .TRUE.                                                          HEDING  0400
      LNEW = .FALSE.                                                         HEDING  0410
   13 MT = 20                                                                HEDING  0420
      NLINES = MOD(LINES-1,LPP)+1                                            HEDING  0430
      XPAGE = 0.01*FLOAT((LINES + LPP-1)/LPP)                                HEDING  0440
C                                                                            HEDING  0450
C     NOTE: MT WILL BE THE PAGE OR OUTPUT UNIT COUNTER                       HEDING  0460
C           NT WILL BE THE ACTUAL OUTPUT UNIT NUMBER                         HEDING  0470
C           IT WILL BE THE INDEX TO THE DATA ARRAY                           HEDING  0480
C           NLINES WILL BE THE NUMBER OF LINES TO BE PRINTED                 HEDING  0490
C                                                                            HEDING  0500
```

```
C                                                                       HEDING  0510
C     EVERY LPP LINES PRINT HEADINGS FOR 7 TYPES OF OUTPUT ABOVE.       HEDING  0520
C                                                                       HEDING  0530
      DO 20 K=1,7                                                       HEDING  0540
      IF (NSG(K).LE.0) GO TO 20                                         HEDING  0550
      KSG = NSG(K)                                                      HEDING  0560
      J3  = 3                                                           HEDING  0570
      IF (K.EQ.7) J3 = 2                                                HEDING  0580
      DO 19 J1=1,KSG,J3                                                 HEDING  0590
      MT = MT + 1                                                       HEDING  0600
      NT = MT                                                           HEDING  0610
      IF  (LNEW)  NT = 6                                                HEDING  0620
      IT = MT - 20                                                      HEDING  0630
      PAGE = FLOAT(MT) + XPAGE                                          HEDING  0640
      WRITE (NT,21) DATE,PAGE,COMENT,VPSTTL,BDYTTL                      HEDING  0650
      IF (K.EQ.1) WRITE (NT,22)                                         HEDING  0660
      IF (K.EQ.2) WRITE (NT,23) UNITL,UNITT                             HEDING  0670
      IF (K.EQ.3) WRITE (NT,24) UNITL                                   HEDING  0680
      IF (K.EQ.4) WRITE (NT,25) UNITT                                   HEDING  0690
      IF (K.EQ.5) WRITE (NT,26) UNITT                                   HEDING  0700
      IF (K.EQ.6) WRITE (NT,27)                                         HEDING  0710
      IF (K.EQ.7) WRITE (NT,28)                                         HEDING  0720
      J2 = MIN0(J1+J3-1,KSG)                                            HEDING  0730
      DO 14 J=J1,J2                                                     HEDING  0740
      KK = MSG(J,K)                                                     HEDING  0750
      HEAD(J) = SEG(KK)                                                 HEDING  0760
      IF (K.LT.7) GO TO 14                                              HEDING  0770
      KK = IABS(KK)                                                     HEDING  0780
      HEAD(J) = JOINT(KK)                                               HEDING  0790
      JJ2 = J-J1+1                                                      HEDING  0800
      K2 = 1                                                            HEDING  0810
      IF (MSG(J,K).LT.0) K2 = 2                                         HEDING  0820
      DO  35  K1=1,4                                                    HEDING  0830
   35 HEADJJ(K1,JJ2) = HEDJ(K1,K2)                                      HEDING  0840
   14 CONTINUE                                                          HEDING  0850
      IF (K.LE.3) WRITE (NT,29) (BLANK,(XSG(I,J,K),I=1,3),J=J1,J2)      HEDING  0860
      IF (K.LE.6) WRITE (NT,30) (BLANK,MSG(J,K),HEAD(J),J=J1,J2)        HEDING  0870
      IF (K.LE.5) WRITE (NT,31) (BLANK,J=J1,J2)                         HEDING  0880
      IF (K.EQ.6) WRITE (NT,32) (BLANK,J=J1,J2)                         HEDING  0890
      IF (K.LT.7) GO TO 15                                              HEDING  0900
      WRITE (NT,33) (BLANK,MSG(J,K),HEAD(J),J=J1,J2)                    HEDING  0910
      WRITE (NT,36) (BLANK,UNITL,UNITM,J=J1,J2)                         HEDING  0920
      WRITE (NT,37) (BLANK,(HEADJJ(K1,J),K1=1,4),J=1,JJ2)               HEDING  0930
   15 WRITE (NT,38)                                                     HEDING  0940
      IF  (.NOT.LNEW)  GO TO 19                                         HEDING  0950
      IF  (K.EQ.7)  GO TO 17                                            HEDING  0960
      JJ = 4*(J2-J1+1)                                                  HEDING  0970
      DO  16  I=1,NLINES                                                HEDING  0980
   16 WRITE (NT,39)  USEC(I),(ZTTH(J,I,IT),J=1,JJ)                      HEDING  0990
      GO TO 19                                                          HEDING  1000
```

```
   17 JJ = 7*(J2-J1+1)                                                    HEDING  1010
      DO  18  I=1,NLINES                                                  HEDING  1020
   18 WRITE (NT,40)  USEC(I),(ZTTH(J,I,IT),J=1,JJ)                        HEDING  1030
   19 CONTINUE                                                            HEDING  1040
   20 CONTINUE                                                            HEDING  1050
   21 FORMAT('1',18X,'DATE:',3X,3A4,68X,'PAGE:',F6.2/                     HEDING  1060
     *       8X,'RUN DESCRIPTION:',3X,20A4/27X,20A4/                      HEDING  1070
     *       3X,'VEHICLE DECELERATION:',3X,20A4/                          HEDING  1080
     *       11X,'CRASH VICTIM:',3X,5A4 )                                 HEDING  1090
   22 FORMAT(' '/27X,                                                     HEDING  1100
     * 'SEGMENT LINEAR ACCELERATIONS (G''S) IN LOCAL REFERENCE'/)         HEDING  1110
   23 FORMAT(' '/27X,                                                     HEDING  1120
     *'SEGMENT LINEAR VELOCITIES (',A4,'/',A4,') IN VEHICLE REFERENCE'/)  HEDING  1130
   24 FORMAT(' '/27X,                                                     HEDING  1140
     *'SEGMENT LINEAR DISPLACEMENTS (',A4,') IN VEHICLE REFERENCE'/)     HEDING  1150
   25 FORMAT(' '//27X,                                                    HEDING  1160
     *'SEGMENT ANGULAR ACCELERATIONS (REV/',A4,'**2) IN LOCAL REFERENCE'  HEDING  1170
     */)                                                                  HEDING  1180
   26 FORMAT(' '//27X,                                                    HEDING  1190
     *'SEGMENT ANGULAR VELOCITIES (REV/',A4,') IN VEHICLE REFERENCE'/)    HEDING  1200
   27 FORMAT(' '//27X,                                                    HEDING  1210
     * 'SEGMENT ANGULAR DISPLACEMENTS (DEG) IN VEHICLE REFERENCE'/)       HEDING  1220
   28 FORMAT(' '/27X,'JOINT PARAMETERS'/)                                 HEDING  1230
   29 FORMAT(9X,3(A4,3X,'POINT (',F6.2,',',F6.2,',',F6.2,') ON ') )       HEDING  1240
   30 FORMAT('     TIME ',3(A4,9X,'SEGMENT NO.',I3,' - ',A4,5X)    )      HEDING  1250
   31 FORMAT('    (MSEC)',3(A4,5X,'X',8X,'Y',8X,'Z',7X,'RES',1X)  )      HEDING  1260
   32 FORMAT('    (MSEC)',3(A4,4X,'YAW',5X,'PITCH',5X,'ROLL',5X,'RES ') ) HEDING  1270
   33 FORMAT(9X,2(A1,21X,'JOINT NO.',I3,' - ',A4,20X)  )                  HEDING  1280
   36 FORMAT('     TIME ',2(A1,'STATE',5X,'JOINT ANGLES (DEG)',8X,        HEDING  1290
     *                    'TOTAL TORQUE (',2A4,') ') )                   HEDING  1300
   37 FORMAT('    (MSEC)',2(A1,4A8,4X,'SPRING  VISCOUS    RES. ') )      HEDING  1310
   38 FORMAT(1X)                                                          HEDING  1320
   39 FORMAT(F9.3,3(3X,4F9.3) )                                           HEDING  1330
   40 FORMAT(F9.3,2(F5.0,3F9.3,2X,3F9.3))                                 HEDING  1340
C                                                                         HEDING  1350
C     PLANE FORCES HEADINGS                                               HEDING  1360
C                                                                         HEDING  1370
      MPSF = 0                                                            HEDING  1380
      IF (NPL.EQ.0) GO TO 52                                              HEDING  1390
      DO  42  J=1,NPL                                                     HEDING  1400
      IF (MNPL(J).EQ.0) GO TO 42                                          HEDING  1410
      KPL = MNPL(J)                                                       HEDING  1420
      DO 41 I=1,KPL                                                       HEDING  1430
      MPSF = MPSF+1                                                       HEDING  1440
      NOPL(MPSF) = J                                                      HEDING  1450
   41 MOPL(MPSF) = MPL(2,I,J)                                             HEDING  1460
   42 CONTINUE                                                            HEDING  1470
      IF (MPSF.EQ.0) GO TO 52                                             HEDING  1480
      DO 44 J1=1,MPSF,2                                                   HEDING  1490
      J2 = MIN0(J1+1,MPSF)                                                HEDING  1500
```

```
      MT = MT + 1                                                          HEDING  1510
      NT = MT                                                              HEDING  1520
      IF  (LNEW)  NT = 6                                                   HEDING  1530
      IT = MT - 20                                                         HEDING  1540
      PAGE = FLOAT(MT) + XPAGE                                             HEDING  1550
      WRITE (NT,21) DATE,PAGE,COMENT,VPSTTL,BDYTTL                         HEDING  1560
      WRITE (NT,45)                                                        HEDING  1570
      N1 = NOPL(J1)                                                        HEDING  1580
      N2 = NOPL(J2)                                                        HEDING  1590
      M1 = MOPL(J1)                                                        HEDING  1600
      M2 = MOPL(J2)                                                        HEDING  1610
      IF  (J1.EQ.J2)  WRITE  (NT,46)                                       HEDING  1620
     *               BLANK,N1,( PLTTL(I,N1),I=1,5),M1,SEG(M1)              HEDING  1630
      IF  (J1.NE.J2)  WRITE  (NT,46)                                       HEDING  1640
     *               BLANK,N1,( PLTTL(I,N1),I=1,5),M1,SEG(M1),             HEDING  1650
     *               BLANK,N2,( PLTTL(I,N2),I=1,5),M2,SEG(M2)              HEDING  1660
      WRITE (NT,47) (BLANK,UNITL,J=J1,J2)                                  HEDING  1670
      WRITE (NT,48) (BLANK,J=J1,J2)                                        HEDING  1680
      WRITE (NT,49) (BLANK,UNITL,UNITM,UNITM,UNITM,J=J1,J2)                HEDING  1690
      WRITE (NT,38)                                                        HEDING  1700
      IF  (.NOT.LNEW)  GO TO 44                                            HEDING  1710
      JJ = 7*(J2-J1+1)                                                     HEDING  1720
      DO  43  I=1,NLINES                                                   HEDING  1730
   43 WRITE (NT,50)  USEC(I),(ZTTH(J,I,IT),J=1,JJ)                         HEDING  1740
   44 CONTINUE                                                             HEDING  1750
   45 FORMAT(27X,'CONTACT FORCES - VEHICLE PANELS VS. SEGMENTS' )          HEDING  1760
   46 FORMAT(' '/8X,2(A4,'  PANEL',I3,' (',5A4,') VS. SEGMENT',I3,         HEDING  1770
     *          ' (',A4,') ') )                                           HEDING  1780
   47 FORMAT(' ',8X,A4,'DEFL-  NORMAL  FRICTION RESULTANT  CONTACT LOCAT   HEDING  1790
     *ION (',A4,')',A2,'DEFL-  NORMAL  FRICTION RESULTANT  CONTACT LOCAT   HEDING  1800
     *ION (',A4,')')                                                       HEDING  1810
   48 FORMAT('    TIME',2(A4,'ECTION   FORCE    FORCE    FORCE     (VEHI   HEDING  1820
     *CLE REFERENCE)' ) )                                                  HEDING  1830
   49 FORMAT('   (MSEC)',2(A3,'(',A4,')',2X,'(',A4,')',4X,'(',A4,')',3X,   HEDING  1840
     *       '(',A4,')     X       Y       Z  ') )                        HEDING  1850
   50 FORMAT(F9.3,2(F9.3,3F9.2,3F8.3) )                                    HEDING  1860
   51 FORMAT(3X,'(MSEC)',4(A1,9X,'X',8X,'Y',8X,'Z',1X))                    HEDING  1870
C                                                                          HEDING  1880
C     BELT FORCES HEADINGS                                                 HEDING  1890
C                                                                          HEDING  1900
   52 MBSF = 0                                                             HEDING  1910
      IF  (NBLT.EQ.0)  GO TO 83                                            HEDING  1920
      DO  54  J=1,NBLT                                                     HEDING  1930
      IF (MNBLT(J).EQ.0) GO TO 54                                          HEDING  1940
      MBSF = MBSF+1                                                        HEDING  1950
      NOPL(MBSF) = J                                                       HEDING  1960
      MOPL(MBSF) = MBLT(2,1,J)                                             HEDING  1970
   54 CONTINUE                                                             HEDING  1980
      IF  (MBSF.EQ.0)  GO TO 83                                            HEDING  1990
      DO 56 J1=1,MBSF,2                                                    HEDING  2000
```

```
      J2 = MIN0(J1+1,MBSF)                                                    HEDING  2010
      MT = MT + 1                                                             HEDING  2020
      NT = MT                                                                 HEDING  2030
      IF  (LNEW)  NT = 6                                                      HEDING  2040
      IT = MT - 20                                                            HEDING  2050
      PAGE = FLOAT(MT) + XPAGE                                                HEDING  2060
      WRITE (NT,21) DATE,PAGE,COMENT,VPSTTL,BDYTTL                            HEDING  2070
      WRITE (NT,57)                                                           HEDING  2080
      N1 = NOPL(J1)                                                           HEDING  2090
      N2 = NOPL(J2)                                                           HEDING  2100
      M1 = MOPL(J1)                                                           HEDING  2110
      M2 = MOPL(J2)                                                           HEDING  2120
      IF  (J1.EQ.J2)  WRITE  (NT,58)                                          HEDING  2130
     *               BLANK,N1,(BLTTTL(I,N1),I=1,5),M1,SEG(M1)                 HEDING  2140
      IF  (J1.NE.J2)  WRITE  (NT,58)                                          HEDING  2150
     *               BLANK,N1,(BLTTTL(I,N1),I=1,5),M1,SEG(M1),                HEDING  2160
     *               BLANK,N2,(BLTTTL(I,N2),I=1,5),M2,SEG(M2)                 HEDING  2170
      WRITE (NT,59) (BLANK,J=J1,J2)                                           HEDING  2180
      WRITE (NT,60) (BLANK,J=J1,J2)                                           HEDING  2190
      WRITE (NT,61) (BLANK,UNITL,UNITL,UNITM,UNITL,UNITL,UNITM,J=J1,J2)       HEDING  2200
      WRITE (NT,38)                                                           HEDING  2210
      IF  (.NOT.LNEW)  GO TO 56                                               HEDING  2220
      JJ = 4*(J2-J1+1)                                                        HEDING  2230
      DO  55  I=1,NLINES                                                      HEDING  2240
   55 WRITE (NT,62)  USEC(I),(ZTTH(J,I,IT),J=1,JJ)                            HEDING  2250
   56 CONTINUE                                                                HEDING  2260
   57 FORMAT('0',26X,'CONTACT FORCES - BELTS VS. SEGMENTS')                   HEDING  2270
   58 FORMAT(' ',7X,2(A4,'   BELT',I3,' (',5A4,') VS. SEGMENT',I3,            HEDING  2280
     *          ' (',A4,') ') )                                              HEDING  2290
   59 FORMAT(' ',2X,2(A4,11X,'ANCHOR  POINT  A',14X,'ANCHOR  POINT  B'))     HEDING  2300
   60 FORMAT(4X,'TIME',2(A4,5X,'STRAIN',7X,'FORCE',12X,                       HEDING  2310
     *                         'STRAIN',7X,'FORCE', 3X) )                    HEDING  2320
   61 FORMAT(3X,'(MSEC)',2(A4,2X,'(',A4,'/',A4,')',4X,'(',A4,')',9X,          HEDING  2330
     *                           '(',A4,'/',A4,')',4X,'(',A4,')',3X) )       HEDING  2340
   62 FORMAT(F9.3,4(F15.6,F12.2,3X) )                                         HEDING  2350
C                                                                             HEDING  2360
C     HARNESS BELT ENDPOINTS FORCES HEADINGS                                  HEDING  2370
C                                                                             HEDING  2380
   83 IF  (NHRNSS.LE.0)  GO TO 91                                             HEDING  2390
      MBSF = 0                                                                HEDING  2400
      J1 = 1                                                                  HEDING  2410
      K1 = 1                                                                  HEDING  2420
      DO  85  I=1,NHRNSS                                                      HEDING  2430
      IF  (NBLTPH(I).LE.0)  GO TO 85                                          HEDING  2440
      J2 = J1 + NBLTPH(I) - 1                                                 HEDING  2450
      DO  84  J=J1,J2                                                         HEDING  2460
      MBSF = MBSF + 1                                                         HEDING  2470
      IF  (NPTSPB(J).LE.0)  GO TO 84                                          HEDING  2480
      K2 = K1 + NPTSPB(J) - 1                                                 HEDING  2490
      NOPL(2*MBSF-1) = J                                                      HEDING  2500
```

```
      NOPL(2*MBSF   ) = I                                                    HEDING  2510
      MOPL(2*MBSF-1) = K1                                                    HEDING  2520
      MOPL(2*MBSF   ) = K2                                                   HEDING  2530
      K1 = K2 + 1                                                            HEDING  2540
   84 CONTINUE                                                               HEDING  2550
      J1 = J2 + 1                                                            HEDING  2560
   85 CONTINUE                                                               HEDING  2570
      DO  87  J1=1,MBSF,2                                                    HEDING  2580
      J2 = MIN0(J1+1,MBSF)                                                   HEDING  2590
      MT = MT + 1                                                            HEDING  2600
      NT = MT                                                                HEDING  2610
      IF  (LNEW)  NT = 6                                                     HEDING  2620
      IT = MT - 20                                                           HEDING  2630
      PAGE = FLOAT(MT) + XPAGE                                               HEDING  2640
      WRITE (NT,21) DATE,PAGE,COMENT,VPSTTL,BDYTTL                           HEDING  2650
      WRITE (NT,88)                                                          HEDING  2660
      WRITE (NT,89) (BLANK,NOPL(2*J-1),NOPL(2*J),J=J1,J2)                    HEDING  2670
      WRITE (NT,90) (BLANK,MOPL(2*J-1),MOPL(2*J),J=J1,J2)                    HEDING  2680
      WRITE (NT,60) (BLANK,J=J1,J2)                                          HEDING  2690
      WRITE (NT,61) (BLANK,UNITL,UNITL,UNITM,UNITL,UNITL,UNITM,J=J1,J2)      HEDING  2700
      WRITE (NT,38)                                                          HEDING  2710
      IF  (.NOT.LNEW)  GO TO 87                                              HEDING  2720
      JJ = 4*(J2-J1+1)                                                       HEDING  2730
      DO  86  I=1,NLINES                                                     HEDING  2740
   86 WRITE (NT,62)  USEC(I),(ZTTH(J,I,IT),J=1,JJ)                           HEDING  2750
   87 CONTINUE                                                               HEDING  2760
   88 FORMAT('0',26X,'HARNESS SYSTEM BELT ENDPOINT FORCES')                  HEDING  2770
   89 FORMAT(9X,2(A4,11X,'BELT NO.',I4,' OF HARNESS NO.',I3,15X))            HEDING  2780
   90 FORMAT(9X,2(A4,6X,'POINT NO.',I5,16X,'POINT NO.',I5,6X))               HEDING  2790
C                                                                            HEDING  2800
C     SPRING DAMPER FORCES HEADINGS                                          HEDING  2810
C                                                                            HEDING  2820
   91 IF  (NSD.LE.0)  GO TO 63                                               HEDING  2830
      DO  94  J1=1,NSD,4                                                     HEDING  2840
      J2 = MIN0(J1+3,NSD)                                                    HEDING  2850
      MT = MT + 1                                                            HEDING  2860
      NT = MT                                                                HEDING  2870
      IF  (LNEW)  NT = 6                                                     HEDING  2880
      IT = MT - 20                                                           HEDING  2890
      PAGE = FLOAT(MT) + XPAGE                                               HEDING  2900
      WRITE (NT,21) DATE,PAGE,COMENT,VPSTTL,BDYTTL                           HEDING  2910
      WRITE (NT,95) (BLANK,J,J=J1,J2)                                        HEDING  2920
      DO  92  J=J1,J2                                                        HEDING  2930
      M1 = MSDM(J)                                                           HEDING  2940
      N1 = MSDN(J)                                                           HEDING  2950
C     POSSIBLE OVERFLOW INTO NOPL ARRAY IS INTENTIONAL.                      HEDING  2960
      HEAD(2*J-1) = SEG(M1)                                                  HEDING  2970
   92 HEAD(2*J   ) = SEG(N1)                                                 HEDING  2980
      WRITE (NT,96)(BLANK,MSDM(J),HEAD(2*J-1),MSDN(J),HEAD(2*J),J=J1,J2)     HEDING  2990
      WRITE (NT,97) (BLANK,J=J1,J2)                                          HEDING  3000
```

```
      WRITE (NT,98) (BLANK,UNITL,UNITM,J=J1,J2)                          HEDING  3010
      WRITE (NT,38)                                                      HEDING  3020
      IF  (.NOT.LNEW)  GO TO 94                                          HEDING  3030
      JJ = 2*(J2-J1+1)                                                   HEDING  3040
      DO  93  I=1,NLINES                                                 HEDING  3050
   93 WRITE (NT,99)  USEC(I),(ZTTH(J,I,IT),J=1,JJ)                       HEDING  3060
   94 CONTINUE                                                           HEDING  3070
   95 FORMAT('0',26X,'SPRING DAMPER FORCES'/                             HEDING  3080
     *       9X,4(A3,3X,'SPRING DAMPER NO.',I3,4X))                     HEDING  3090
   96 FORMAT(9X,4(A3,'SEG',I3,'(',A4,') - SEG',I3,'(',A4,')'))           HEDING  3100
   97 FORMAT(4X,'TIME',1X,4(A3,5X,'LENGTH',7X,'FORCE',4X))               HEDING  3110
   98 FORMAT(3X,'(MSEC)',4(A3,5X,'(',A4,')',6X,'(',A4,')',4X))           HEDING  3120
   99 FORMAT (F9.3,4(F14.3,F12.2,4X))                                    HEDING  3130
C                                                                        HEDING  3140
C     SEGMENT FORCES HEADINGS                                            HEDING  3150
C                                                                        HEDING  3160
   63 MSSF = 0                                                           HEDING  3170
      DO  65  J=1,NSEG                                                   HEDING  3180
      IF (MNSEG(J).EQ.0) GO TO 65                                        HEDING  3190
      LSEG = MNSEG(J)                                                    HEDING  3200
      DO 64  I=1,LSEG                                                    HEDING  3210
      MSSF = MSSF+1                                                      HEDING  3220
      NOPL(MSSF) = J                                                     HEDING  3230
   64 MOPL(MSSF) = MSEG(2,I,J)                                           HEDING  3240
   65 CONTINUE                                                           HEDING  3250
      IF (MSSF.EQ.0) GO TO 70                                            HEDING  3260
      DO 67 J=1,MSSF                                                     HEDING  3270
      MT = MT + 1                                                        HEDING  3280
      NT = MT                                                            HEDING  3290
      IF  (LNEW)  NT = 6                                                 HEDING  3300
      IT = MT - 20                                                       HEDING  3310
      PAGE = FLOAT(MT) + XPAGE                                           HEDING  3320
      WRITE (NT,21) DATE,PAGE,COMENT,VPSTTL,BDYTTL                       HEDING  3330
      N1 = NOPL(J)                                                       HEDING  3340
      M1 = MOPL(J)                                                       HEDING  3350
      WRITE (NT,68) N1,SEG(N1),M1,SEG(M1),UNITL,N1,M1                    HEDING  3360
     *             ,UNITL,UNITM,UNITM,UNITM                              HEDING  3370
      IF  (.NOT.LNEW)  GO TO 67                                          HEDING  3380
      DO  66  I=1,NLINES                                                 HEDING  3390
   66 WRITE (NT, 69)  USEC(I),(ZTTH(JJ,I,IT),JJ=1,10)                    HEDING  3400
   67 CONTINUE                                                           HEDING  3410
   68 FORMAT('0',26X,'CONTACT FORCES - SEGMENT NO.',I3,' (',A4,          HEDING  3420
     *                        ') VS. SEGMENT NO.',I3,' (',A4,')'//      HEDING  3430
     *       13X,'DEFL-  NORMAL  FRICTION RESULTANT',                    HEDING  3440
     *       14X,'CONTACT LOCATION (',A4,')'/                            HEDING  3450
     *       4X,'TIME    ECTION',3(3X,'FORCE',1X),                       HEDING  3460
     *       2(' SEG.',I3,' LOCAL REFERENCE ')/                          HEDING  3470
     *       3X,'(MSEC)',3X,'(',A4,')', 3(3X,'(',A4,')'),                HEDING  3480
     *       2(5X,'X',7X,'Y',7X,'Z',4X)/1X)                              HEDING  3490
   69 FORMAT(2F9.3,3F9.2,3F8.3,2X,3F8.3)                                 HEDING  3500
```

```
C                                                                       HEDING  3510
C     AIRBAG FORCES HEADINGS                                            HEDING  3520
C                                                                       HEDING  3530
   70 IF (NBAG.EQ.0) GO TO 82                                           HEDING  3540
      DO  77  J=1,NBAG                                                  HEDING  3550
      IF (MNBAG(J).EQ.0) GO TO 77                                       HEDING  3560
      MT = MT + 1                                                       HEDING  3570
      NT = MT                                                           HEDING  3580
      IF  (LNEW)  NT = 6                                                HEDING  3590
      IT = MT - 20                                                      HEDING  3600
      PAGE = FLOAT(MT) + XPAGE                                          HEDING  3610
      WRITE (NT,21) DATE,PAGE,COMENT,VPSTTL,BDYTTL                      HEDING  3620
      WRITE (NT,78) J,(BAGTTL(I,J),I=1,5)                               HEDING  3630
      IF  (.NOT.LNEW)  GO TO 72                                         HEDING  3640
      DO  71  I=1,NLINES                                                HEDING  3650
   71 WRITE (NT, 79)  USEC(I),(ZTTH(JJ,I,IT),JJ=1,12)                   HEDING  3660
   72 KBAG = 0                                                          HEDING  3670
      KP = NPANEL(J)+1                                                  HEDING  3680
      DO 73 K=1,KP                                                      HEDING  3690
      KBAG = KBAG+1                                                     HEDING  3700
   73 HEAD(KBAG) = PHED(K)                                              HEDING  3710
      KP = MNBAG(J)                                                     HEDING  3720
      DO 74 K=1,KP                                                      HEDING  3730
      KBAG = KBAG+1                                                     HEDING  3740
      M = MBAG(2,K,J)                                                   HEDING  3750
   74 HEAD(KBAG) = SEG(M)                                               HEDING  3760
      DO 76 J1=1,KBAG,4                                                 HEDING  3770
      J2 = MIN0(J1+3,KBAG)                                              HEDING  3780
      MT = MT + 1                                                       HEDING  3790
      NT = MT                                                           HEDING  3800
      IF  (LNEW)  NT = 6                                                HEDING  3810
      IT = MT - 20                                                      HEDING  3820
      PAGE = FLOAT(MT) + XPAGE                                          HEDING  3830
      WRITE (NT,21) DATE,PAGE,COMENT,VPSTTL,BDYTTL                      HEDING  3840
      WRITE (NT,80)UNITM;J,(BAGTTL(I,J),I=1,5),(BLANK,J,HEAD(K),K=J1,J2)HEDING  3850
      WRITE (NT,51) (BLANK,K=J1,J2)                                     HEDING  3860
      WRITE (NT,38)                                                     HEDING  3870
      IF  (.NOT.LNEW)  GO TO 76                                         HEDING  3880
      JJ = 3*(J2-J1+1)                                                  HEDING  3890
      DO  75  I=1,NLINES                                                HEDING  3900
   75 WRITE (NT, 81)  USEC(I),(ZTTH(K,I,IT),K=1,JJ)                     HEDING  3910
   76 CONTINUE                                                          HEDING  3920
   77 CONTINUE                                                          HEDING  3930
   78 FORMAT('0',26X,'PARAMETERS FOR AIRBAG NO.',I2,4X,5A4//            HEDING  3940
     *        16X,'SUPPLY CYLINDER   STATIC'/                          HEDING  3950
     *        4X,'TIME',8X,'PRES.',4X,'TEMP.',4X,'PRES.',12X,'AIRBAG', HEDING  3960
     *  3X,'CENTER',14X,'AIRBAG  SEMIAXES',12X,'ORIENTATION (DEG.)'/   HEDING  3970
     *  3X,'(MSEC)',7X,'(PSIG)  (DEG.R)   (PSIG)',8X,'X',8X,'Y',8X,'Z', HEDING  3980
     *  11X,'A',8X,'B',8X,'C',10X,'YAW',4X,'PITCH',5X,'ROLL'/ )        HEDING  3990
   79 FORMAT (F9.3,3X,3F9.2,2(3X,3F9.3),3X,3F9.2)                       HEDING  4000
   80 FORMAT('0',26X,'CONTACT FORCES (',A4,') ON AIRBAG NO.',I2,4X,5A4// HEDING  4010
     *      /4X,'TIME',4(A1,11X,'AIRBAG',I2,' VS. ',A4,1X))           HEDING  4020
                                                                       HEDING  4030
   81 FORMAT (F9.3,4(3X,3F9.2))                                        HEDING  4040
   82 RETURN                                                           HEDING  4050
      END
```

```
      SUBROUTINE HERRON(HD3,NT1,THETO,THETOP)                                  HERRON  0010
C                                                          REV 19 08/05/78    HERRON  0020
C     COMPUTES THETO - ANGLE OF JOINT STOP                                     HERRON  0030
C              THETOP- DERIVATIVE OF THETO WITH RESPECT TO PHI                 HERRON  0040
C                                                                             HERRON  0050
C          FROM HD3   - COMPONENTS OF VECTOR DEFINING PHI                     HERRON  0060
C               NT1   - INDEX TO TAB ARRAY DEFINING FUNCTION                  HERRON  0070
C                                                                             HERRON  0080
      IMPLICIT REAL*8(A-H,O-Z)                                                HERRON  0090
      COMMON/TABLES/ MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(500),TAB(2600)      HERRON  0100
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                               HERRON  0110
     *               UNITL,UNITM,UNITT,GRAVTY(3)                              HERRON  0120
      DIMENSION HD3(3)                                                        HERRON  0130
      IF (TAB(NT1+1).LE.0.0) GO TO 30                                         HERRON  0140
      IF (TAB(NT1+2).LE.0.0) GO TO 30                                         HERRON  0150
C                                                                             HERRON  0160
C     THETO = P1(CP) + SP*P2(CP)                                              HERRON  0170
C                                                                             HERRON  0180
C     THETOP = -SP*P1'(CP) + CP*P2(CP) - SP**2*P2'(CP)                        HERRON  0190
C                                                                             HERRON  0200
C          WHERE P1(X),P2(X) ARE THE TWO 5TH ORDER POLYNOMIALS DEFINED        HERRON  0210
C                            IN TAB(NT1+5) AND TAB(NT1+11)                    HERRON  0220
C               P1'(X),P2'(X) ARE THEIR DERIVATIVES WITH RESPECT TO PHI       HERRON  0230
C            AND SP,CP ARE SIN(PHI) AND COS(PHI)                              HERRON  0240
C                                                                             HERRON  0250
      STH2 = 1.0-HD3(3)**2                                                    HERRON  0260
      STH  = DSQRT(STH2)                                                      HERRON  0270
      CP   = HD3(1)/STH                                                       HERRON  0280
      SP   = HD3(2)/STH                                                       HERRON  0290
      P1   = TAB(NT1+5 )+ CP*(TAB(NT1+6 )                                     HERRON  0300
     *                    + CP*(TAB(NT1+7 )                                  HERRON  0310
     *                    + CP*(TAB(NT1+8 )                                  HERRON  0320
     *                    + CP*(TAB(NT1+9 )                                  HERRON  0330
     *                    + CP*(TAB(NT1+10) )))))                            HERRON  0340
      P2   = TAB(NT1+11)+ CP*(TAB(NT1+12)                                     HERRON  0350
     *                    + CP*(TAB(NT1+13)                                  HERRON  0360
     *                    + CP*(TAB(NT1+14)                                  HERRON  0370
     *                    + CP*(TAB(NT1+15)                                  HERRON  0380
     *                    + CP*(TAB(NT1+16) )))))                            HERRON  0390
      P1P  = TAB(NT1+6 )+ CP*(2.0*TAB(NT1+7 )                                 HERRON  0400
     *                    + CP*(3.0*TAB(NT1+8 )                              HERRON  0410
     *                    + CP*(4.0*TAB(NT1+9 )                              HERRON  0420
     *                    + CP*(5.0*TAB(NT1+10) ))))                         HERRON  0430
      P2P  = TAB(NT1+12)+ CP*(2.0*TAB(NT1+13)                                 HERRON  0440
     *                    + CP*(3.0*TAB(NT1+14)                              HERRON  0450
     *                    + CP*(4.0*TAB(NT1+15)                              HERRON  0460
     *                    + CP*(5.0*TAB(NT1+16) ))))                         HERRON  0470
      THETO  = P1 + SP*P2                                                     HERRON  0480
      THETOP = CP*P2 - SP*(P1P + SP*P2P)                                      HERRON  0490
      GO TO 99                                                                HERRON  0500
C                                                                             HERRON  0510
C     EVALUATE THETO AND THETOP FROM REGULAR FUNCTION DEFINITION WHERE        HERRON  0520
C     THETO (ORDINATE) IS A FUNCTION OF PHI (ABSCISSA)  (0 < PHI < 2*PI)      HERRON  0530
C                                                                             HERRON  0540
   30 PHI = DATAN2(HD3(2),HD3(1))                                             HERRON  0550
      IF (PHI.LT.0.0) PHI = PHI + 2.0*PI                                      HERRON  0560
      THETO  = EVALFD(PHI,NT1,1)                                              HERRON  0570
      THETOP = EVALFD(PHI,NT1,0)                                              HERRON  0580
   99 RETURN                                                                  HERRON  0590
      END                                                                     HERRON  0600
```

```
      SUBROUTINE HICCSI(NPTS)                                                  HICCSI  0010
C                                                          REV 18 07/26/78    HICCSI  0020
C                                                                             HICCSI  0030
C     COMPUTES HIC, HSI AND CSI FOR CVS PROGRAM.                              HICCSI  0040
C                                                                             HICCSI  0050
C     ASSUMES Z ARRAY CONTAINS                                                HICCSI  0060
C     Z(I,1 ),I=1,NPTS : TIME POINTS (SECONDS)                                HICCSI  0070
C     Z(I,JH),I=1,NPTS :  HEAD RESULTANT ACCELERATIONS (G'S)                  HICCSI  0080
C     Z(I,JC),I=1,NPTS : CHEST RESULTANT ACCELERATIONS (G'S)                  HICCSI  0090
C                                                                             HICCSI  0100
C     NOTE:                                                                   HICCSI  0110
C     IF JDTPTS(1)=0,  HEAD RESULTANT IS NOT AVAILABLE (JH=NULL,JC=2).        HICCSI  0120
C     IF JDTPTS(2)=0, CHEST RESULTANT IS NOT AVAILABLE (JH=2,JC=NULL).        HICCSI  0130
C     OTHERWISE, JH=2 AND JC=3.                                               HICCSI  0140
C                                                                             HICCSI  0150
      COMMON/CDINT/ JDTPTS(18),Z(400,25)                                      HICCSI  0160
      DIMENSION AREA(400)                                                     HICCSI  0170
      IF  (NPTS.LT.25)  GO TO 25                                              HICCSI  0180
      WRITE  (6,14)                                                           HICCSI  0190
   14 FORMAT (1H1, '   HIC, HSI AND CSI RESULTS')                             HICCSI  0200
      JH  = 2                                                                 HICCSI  0210
      JC  = 3                                                                 HICCSI  0220
      IF  (JDTPTS(1).EQ.0)  JC = 2                                            HICCSI  0230
      CSI = 0.0                                                               HICCSI  0240
      HSI = 0.0                                                               HICCSI  0250
      HIC = 0.0                                                               HICCSI  0260
      CMX = Z(1,JC)                                                           HICCSI  0270
      HMX = Z(1,JH)                                                           HICCSI  0280
      IF  (JDTPTS(2).EQ.0)  GO TO 16                                          HICCSI  0290
C                                                                             HICCSI  0300
C     COMPUTE CSI - CHEST SEVERITY INDEX                                      HICCSI  0310
C                                                                             HICCSI  0320
      H1  = SQRT(Z(1,JC)) * Z(1,JC)**2                                        HICCSI  0330
      DO  15  I=2,NPTS                                                        HICCSI  0340
      H2  = SQRT(Z(I,JC)) * Z(I,JC)**2                                        HICCSI  0350
      DT  = Z(I,1) - Z(I-1,1)                                                 HICCSI  0360
      CSI = CSI + 0.5*DT*(H1+H2)                                              HICCSI  0370
      IF  (CMX.GT.Z(I,JC))  GO TO 15                                          HICCSI  0380
      CMX = Z(I,JC)                                                           HICCSI  0390
      CMT = Z(I,1)                                                            HICCSI  0400
   15 H1  = H2                                                                HICCSI  0410
      CSI = 0.001*CSI                                                         HICCSI  0420
   16 IF  (JDTPTS(1).EQ.0)  GO TO 23                                          HICCSI  0430
C                                                                             HICCSI  0440
C     COMPUTE HSI - HEAD SEVERITY INDEX - AND AREA TABLE                      HICCSI  0450
C                                                                             HICCSI  0460
      AREA(1) = 0.0                                                           HICCSI  0470
      H1  = SQRT(Z(1,JH)) * Z(1,JH)**2                                        HICCSI  0480
      DO  17  I=2,NPTS                                                        HICCSI  0490
      H2  = SQRT(Z(I,JH)) * Z(I,JH)**2                                        HICCSI  0500
```

```
      DT   = 0.5*(Z(I,1) - Z(I-1,1))                                          HICCSI  0510
      AREA(I) = AREA(I-1) + DT*(Z(I-1,JH)+Z(I,JH))                            HICCSI  0520
      HSI = HSI + DT*(H1+H2)                                                  HICCSI  0530
      IF  (HMX.GT.Z(I,JH))  GO TO 17                                          HICCSI  0540
      HMX = Z(I,JH)                                                           HICCSI  0550
      HMT = Z(I,1)                                                            HICCSI  0560
   17 H1  = H2                                                                HICCSI  0570
      HSI = 0.001*HSI                                                         HICCSI  0580
C                                                                             HICCSI  0590
C      COMPUTE HIC - HEAD INJURY CRITERION - AND TIME DURATION HT1,HT2        HICCSI  0600
C                                                                             HICCSI  0610
      DO  19  K=2,NPTS                                                        HICCSI  0620
      DO  18  L=K,NPTS                                                        HICCSI  0630
      DT   = Z(L,1)  -  Z(K-1,1)                                              HICCSI  0640
      DH   = AREA(L) - AREA(K-1)                                              HICCSI  0650
      HT   = DH/DT                                                            HICCSI  0660
      HM   = DT*SQRT(HT)*HT**2                                                HICCSI  0670
      IF  (HM.LE.HIC)  GO TO 18                                               HICCSI  0680
      HIC = HM                                                                HICCSI  0690
      HT1 = Z(K-1,1)                                                          HICCSI  0700
      HT2 = Z(L,1)                                                            HICCSI  0710
      HA2 = Z(L,JH)                                                           HICCSI  0720
      HA1 = Z(K-1,JH)                                                         HICCSI  0730
      AVE = HT                                                                HICCSI  0740
   18 CONTINUE                                                                HICCSI  0750
   19 CONTINUE                                                                HICCSI  0760
      HIC = 0.001*HIC                                                         HICCSI  0770
      WRITE  (6,21)  HIC,HT1,HT2,HA1,HA2,AVE                                  HICCSI  0780
   21 FORMAT (1H0, '   HEAD INJURY CRITERION'//                               HICCSI  0790
     *          '     HIC = ', F8.2,                                         HICCSI  0800
     *             9X, 'TIME DURATION = ', F9.3, '  TO ', F9.3, ' MSEC'/     HICCSI  0810
     *    20X, 'WITH HEAD RESULTANTS = ', F9.3, ' AND ', F9.3, ' G''S'//     HICCSI  0820
     *14X,'AVERAGE HEAD RESULTANT FOR TIME DURATION = ', F9.3, ' G''S')       HICCSI  0830
      WRITE  (6,22)  HSI,HMX,HMT                                              HICCSI  0840
   22 FORMAT (1H0, '   HEAD SEVERITY INDEX'//                                 HICCSI  0850
     *          '     HSI = ', F8.2//                                        HICCSI  0860
     *    '     MAX  HEAD RESULTANT = ', F9.3, ' G''S AT ', F9.3, ' MSEC')    HICCSI  0870
   23 IF  (JDTPTS(2).EQ.0)  GO TO 25                                          HICCSI  0880
      WRITE  (6,24)  CSI,CMX,CMT                                              HICCSI  0890
   24 FORMAT (1H0, '   CHEST SEVERITY INDEX'//                                HICCSI  0900
     *          '     CSI = ', F8.2//                                        HICCSI  0910
     *    '     MAX CHEST RESULTANT = ', F9.3, ' G''S AT ', F9.3, ' MSEC')    HICCSI  0920
                                                                              HICCSI  0930
   25 RETURN                                                                  HICCSI  0940
      END
```

```
      SUBROUTINE HINPUT                                                      HINPUT  0010
C                                                          REV 19 10/23/79   HINPUT  0020
C     CONTROLS THE INPUT OF CARDS F.8.A - F.8.D CONTAINING THE SETUP AND     HINPUT  0030
C     CONTROL OF THE HARNESS BELT SYSTEM.                                    HINPUT  0040
C                                                                            HINPUT  0050
C                                                                            HINPUT  0060
      IMPLICIT REAL*8(A-H,O-Z)                                               HINPUT  0070
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,                HINPUT  0080
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)             HINPUT  0090
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                              HINPUT  0100
     *               UNITL,UNITM,UNITT,GRAVTY(3)                             HINPUT  0110
      COMMON/HRNESS/ BAR(15,100),BB(100),BBDOT(100),PLOSS(2,100),            HINPUT  0120
     *               XLONG(20),HTIME(2),IBAR(5,100),NL(2,100),               HINPUT  0130
     *               NPTSPB(20),NPTPLY(20),NTHRNS(20),NBLTPH(5)              HINPUT  0140
      COMMON/TABLES/ MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(500),TAB(2600)     HINPUT  0150
      COMMON/CNTSRF/ PL(17,30),BELT(20,8),TPTS(6,8),BD(24,40)                HINPUT  0160
      COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),                HINPUT  0170
     *               BLTTTL(5,8),PLTTL(5,30),BAGTTL(5,6),SEG(30),            HINPUT  0180
     *               JOINT(30),CGS(30),JS(30)                                HINPUT  0190
      REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT           HINPUT  0200
      LOGICAL*1 CGS,JS                                                       HINPUT  0210
C     THIS COMMON/TEMTVS/ IS SHARED BY CINPUT, FINPUT, HINPUT AND FDINIT     HINPUT  0220
      COMMON/TEMPVS/ JTITLE(5,51),NF(5),MS(3),KTITLE(31)                     HINPUT  0230
      REAL JTITLE,KTITLE                                                     HINPUT  0240
      IF  (NHRNSS.EQ.0)  GO TO 99                                            HINPUT  0250
C                                                                            HINPUT  0260
C     INPUT CARD F.8.A                                                       HINPUT  0270
C         (NOTE: NHRNSS NOW SUPPLIED ON INPUT CARD D.1)                      HINPUT  0280
C         NBLTPH - NO. OF BELTS PER HARNESS                                  HINPUT  0290
C                                                                            HINPUT  0300
      READ  (5,11)  (NBLTPH(I),I=1,NHRNSS)                                  HINPUT  0310
   11 FORMAT(18I4)                                                           HINPUT  0320
      WRITE (6,12) NHRNSS,(NBLTPH(I),I=1,NHRNSS)                             HINPUT  0330
   12 FORMAT('1 HARNESS-BELT SYSTEM INPUT',93X,'CARDS F.8'//                 HINPUT  0340
     *       '  NO. OF HARNESSES =',I4//                                    HINPUT  0350
     *       '  NO. OF BELTS PER HARNESS =',5I6)                            HINPUT  0360
      J1 = 1                                                                 HINPUT  0370
      K1 = 1                                                                 HINPUT  0380
      DO 90 I=1,NHRNSS                                                       HINPUT  0390
      IF (NBLTPH(I).LE.0) GO TO 90                                           HINPUT  0400
      J2 = J1 + NBLTPH(I) -1                                                 HINPUT  0410
C                                                                            HINPUT  0420
C     INPUT CARD F.8.B - NPTSPB - NO. OF POINTS PER BELT.                    HINPUT  0430
C                                                                            HINPUT  0440
      READ (5,11) (NPTSPB(J),J=J1,J2)                                        HINPUT  0450
      WRITE (6,13) I,(NPTSPB(J),J=J1,J2)                                     HINPUT  0460
   13 FORMAT('0 FOR HARNESS NO.',I3,' NO. OF POINTS PER BELT =',20I4)        HINPUT  0470
      DO 80 J=J1,J2                                                          HINPUT  0480
      IF (NPTSPB(J).EQ.0) GO TO 80                                           HINPUT  0490
C                                                                            HINPUT  0500
```

```
C     INPUT CARD F.8.C - 5 FUNCTION NOS AND LENGTH OF EACH BELT.
C
      READ  (5,14)     NF,XLONG(J)
   14 FORMAT(5I4,F12.6)
      WRITE (6,15) I,J,NF,XLONG(J),UNITL
   15 FORMAT('0 HARNESS NO.',I3,' BELT NO.',I3,' FUNCTION NOS.',5I6,
     *       '  REFERENCE SLACK = ',F9.3,1X,A4/)
      IF  (XLONG(J).EQ.0.0)  XLONG(J) = EPS(24)
      WRITE  (6,16)
   16 FORMAT ('0    K    KS    KE    NT    NPD   NDR    FUNCTION NOS.',
     *        66X,'CARDS F.8.D'/)
C
C     SET UP POINTERS IN NTAB AND INITIAL VALUES OF TAB FOR BELT J
C         AS WAS DONE FOR OTHER CONTACTS IN SUBROUTINE FINPUT.
C
      NTHRNS(J) = MXNTB+1
      CALL FDINIT
      K2 = K1 + NPTSPB(J) - 1
      DO 70 K=K1,K2
C
C     INPUT CARD F.8.D
C
      READ  (5,21)  KS,KE,NPD,NDR,NF, (BAR(L,K),L=1,3)
   21 FORMAT (9I4,3F12.0)
      READ  (5,22)  (BAR(L,K),L=7,12)
   22 FORMAT (6F12.0)
      IBAR(1,K) = KS
      IBAR(2,K) = KE
      IBAR(4,K) = NPD
      IBAR(5,K) = NDR
      IBAR(3,K) = MXNTB+1
      CALL FDINIT
      SQRER = 1.0
      IF  (KE.NE.0)  SQRER = DSQRT(XDY(BAR(1,K),BD(7,KE),BAR(1,K)))
      DO  26  L=1,3
      IF  (KE.NE.0)  BAR(L+6,K) = BD(L+3,KE)
   26 BAR(L+3,K) = BAR(L,K)/SQRER
      WRITE  (6,31)  K,(IBAR(L,K),L=1,5),NF
   31 FORMAT (11I6)
   70 CONTINUE
      WRITE  (6,71)  UNITL,UNITL,UNITL,UNITL
   71 FORMAT ('0',12X,'BASE REFERENCE (',     A4,')',
     *             7X,'ADJUSTED REFERENCE (', A4,')',
     *            11X,'OFFSET (',             A4,')',
     *            11X,'PREFERRED DIRECTION (',A4,')'/
     *        5X,'K', 4(8X,'X',8X,'Y',8X,'Z',3X) /)
      WRITE  (6,72)  (K,(BAR(L,K),L=1,12),K=K1,K2)
   72 FORMAT (I6,3X,3F9.3,3X,3F9.3,3X,3F9.3,3X,3F9.3)
      K1 = K2+1
   80 CONTINUE
      J1 = J2+1
   90 CONTINUE
      DO  92  K=1,100
      BBDOT(K) = 0.0
      DO  91  J=1,2
   91 PLOSS(J,K) = 0.0
      DO  92  J=1,3
   92 BAR(J+12,K) = 0.0
   99 RETURN
      END
```

```
HINPUT  0510
HINPUT  0520
HINPUT  0530
HINPUT  0540
HINPUT  0550
HINPUT  0560
HINPUT  0570
HINPUT  0580
HINPUT  0590
HINPUT  0600
HINPUT  0610
HINPUT  0620
HINPUT  0630
HINPUT  0640
HINPUT  0650
HINPUT  0660
HINPUT  0670
HINPUT  0680
HINPUT  0690
HINPUT  0700
HINPUT  0710
HINPUT  0720
HINPUT  0730
HINPUT  0740
HINPUT  0750
HINPUT  0760
HINPUT  0770
HINPUT  0780
HINPUT  0790
HINPUT  0800
HINPUT  0810
HINPUT  0820
HINPUT  0830
HINPUT  0840
HINPUT  0850
HINPUT  0860
HINPUT  0870
HINPUT  0880
HINPUT  0890
HINPUT  0900
HINPUT  0910
HINPUT  0920
HINPUT  0930
HINPUT  0940
HINPUT  0950
HINPUT  0960
HINPUT  0970
HINPUT  0980
HINPUT  0990
HINPUT  1000
HINPUT  1010
HINPUT  1020
HINPUT  1030
HINPUT  1040
HINPUT  1050
HINPUT  1060
HINPUT  1070
HINPUT  1080
HINPUT  1090
HINPUT  1100
```

```
      SUBROUTINE HPTURB                                                      HPTURB  0010
C                                                          REV 20 04/11/80   HPTURB  0020
      IMPLICIT REAL*8 (A-H,O-Z)                                              HPTURB  0030
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,                HPTURB  0040
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)             HPTURB  0050
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                              HPTURB  0060
     *               UNITL,UNITM,UNITT,GRAVTY(3)                             HPTURB  0070
      COMMON/CNTSRF/ PL(17,30),BELT(20,8),TPTS(6,8),BD(24,40)                HPTURB  0080
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),     HPTURB  0090
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)            HPTURB  0100
      COMMON/RSAVE/  XSG(3,20,3),DPMI(3,3,30),LPMI(30),NSG(7),MSG(20,7)      HPTURB  0110
      COMMON/HRNESS/ BAR(15,100),BB(100),BBDOT(100),PLOSS(2,100),            HPTURB  0120
     *               XLONG(20),HTIME(2),IBAR(5,100),NL(2,100),               HPTURB  0130
     *               NPTSPB(20),NPTPLY(20),NTHRNS(20),NBLTPH(5)              HPTURB  0140
C     THIS COMMON/TEMPVS/ IS SHARED BY HPTURB, HBPLAY, HBELT AND HSETC.      HPTURB  0150
      COMMON/TEMPVS/ B(3,3,3),S(3,3),T(3),R(3),V(3),T1(3),T2(3),             HPTURB  0160
     *               E(3,3,50),EDOT(3,50),FCE(3,50),FR(3,50),ZR(3,50),       HPTURB  0170
     *               TR(3,50),U(3,50),PTLOSS(2,50),BL(50),FB(50),FP(50),     HPTURB  0180
     *               OLDBB(100),RHS(3,54),C(3,3,200),IJK(54,54)              HPTURB  0190
      DIMENSION BLOSS(2,20),HLOSS(2,5)                                       HPTURB  0200
      EQUIVALENCE (BLOSS(1,1),C(1,1,1)) , (HLOSS(1,1),C(1,1,10))             HPTURB  0210
      LOGICAL LAST                                                           HPTURB  0220
      DATA MAXITR/10/                                                        HPTURB  0230
      CALL ELTIME (1,39)                                                     HPTURB  0240
      CALL HBPLAY                                                            HPTURB  0250
      DHT = 0.0                                                              HPTURB  0260
      IF  (TIME.NE.0.0)  DHT = TIME - HTIME(1)                               HPTURB  0270
      HTIME(1) = TIME                                                        HPTURB  0280
      DO  11  J=1,100                                                        HPTURB  0290
      PTLOSS(J,1) = 0.0                                                      HPTURB  0300
      OLDBB(J) = BB(J)                                                       HPTURB  0310
      DO  11  I=1,3                                                          HPTURB  0320
   11 BAR(I,J) = BAR(I+3,J)                                                  HPTURB  0330
      TSEC = 1000.0*TIME                                                     HPTURB  0340
      IF  (NPRT(28).NE.0)  WRITE (6,12) TSEC,UNITL,UNITM,UNITL,              HPTURB  0350
     *                                  UNITL,UNITM,UNITL,UNITM              HPTURB  0360
   12 FORMAT('1   HARNESS BELT RESULTS FOR TIME =',F9.3,' MSEC.'///          HPTURB  0370
     *       36X,'BELT STRAIN',68X,'PENETRATION'/                            HPTURB  0380
     *       '     POINT   POINT  SEGMENT LENGTH  ENERGY LOSS',5X,           HPTURB  0390
     *       'REFERENCE POINT (',A4,')',13X,'BELT FORCES  ('A4,')',          HPTURB  0400
     *       9X,'ENERGY LOSS'/                                               HPTURB  0410
     *       '       NO.     INDEX    NO.   (',A4,')  (',2A4,')',7X,         HPTURB  0420
     *       'X',8X,'Y',8X,'Z',13X,'X',10X,'Y',10X,'Z',8X,'(',2A4,')'/)      HPTURB  0430
      J1 = 1                                                                 HPTURB  0440
      K0 = 1                                                                 HPTURB  0450
      KNL0 = 0                                                               HPTURB  0460
      DO  61  NH=1,NHRNSS                                                    HPTURB  0470
      IF  (NBLTPH(NH).LE.0)  GO TO 61                                        HPTURB  0480
      ITER = 1                                                               HPTURB  0490
      KNL1 = KNL0                                                            HPTURB  0500
```

```
C                                                                       HPTURB  0510
C     START OF   DO  59  ITER=1,MAXITR   LOOP                          HPTURB  0520
C                                                                       HPTURB  0530
   13 NJ2 = 54                                                          HPTURB  0540
      DO  14  I=1,NJ2                                                   HPTURB  0550
      DO  14  J=1,NJ2                                                   HPTURB  0560
   14 IJK(I,J) = 0                                                      HPTURB  0570
      KNL0 = KNL1                                                       HPTURB  0580
      J2   = J1 + NBLTPH(NH) - 1                                        HPTURB  0590
      NTP  = 0                                                          HPTURB  0600
      IJ   = 0                                                          HPTURB  0610
      CALL HBELT (J1,J2,KNL0,1)                                         HPTURB  0620
      KH0  = 0                                                          HPTURB  0630
      KNL0 = KNL1                                                       HPTURB  0640
      DO  15  NB=J1,J2                                                  HPTURB  0650
      IF  (NPTPLY(NB).LE.0)  GO TO 15                                   HPTURB  0660
      NPTS = NPTPLY(NB)                                                 HPTURB  0670
      CALL HSETC (NPTS,KH0,KNL0,NTP,IJ)                                 HPTURB  0680
      KH0 = KH0 + NPTS                                                  HPTURB  0690
      KNL0 = KNL0 + NPTS                                                HPTURB  0700
   15 CONTINUE                                                          HPTURB  0710
C                                                                       HPTURB  0720
C     SET UP C AND IJK ELEMENTS FOR TIE-POINTS.                         HPTURB  0730
C                                                                       HPTURB  0740
      KNL0 = KNL1                                                       HPTURB  0750
      KNLK = KNL0 + 1                                                   HPTURB  0760
      K1   = KNLK                                                       HPTURB  0770
      DO  22  NB=J1,J2                                                  HPTURB  0780
      IF  (NPTPLY(NB).LE.0)  GO TO 22                                   HPTURB  0790
      K2   = K1 + NPTPLY(NB) - 1                                        HPTURB  0800
      DO  21  KNL=K1,K2                                                 HPTURB  0810
      KI   = NL(1,KNL)                                                  HPTURB  0820
      KS   = IABS(IBAR(1,KI))                                           HPTURB  0830
      IF  (KS.LT.100)  GO TO 21                                         HPTURB  0840
      KS1  = KS/100                                                     HPTURB  0850
      DO  16  K=KNLK,KNL                                                HPTURB  0860
      KK   = K                                                          HPTURB  0870
      KI   = NL(1,K)                                                    HPTURB  0880
      KS   = IABS(IBAR(1,KI))                                           HPTURB  0890
      IF  (KS.LT.100)  GO TO 16                                         HPTURB  0900
      KS2  = KS/100                                                     HPTURB  0910
      IF  (KS2.EQ.KS1)  GO TO 17                                        HPTURB  0920
   16 CONTINUE                                                          HPTURB  0930
   17 IF  (KK.EQ.KNL)  GO TO 21                                         HPTURB  0940
      KK1 = KK  - KNL0                                                  HPTURB  0950
      KK2 = KNL - KNL0                                                  HPTURB  0960
      IQ1 = MAX0(1,KK2-1)                                               HPTURB  0970
      IQ2 = MIN0(KK2+1,KH0)                                             HPTURB  0980
      DO  18  IQ=IQ1,IQ2                                                HPTURB  0990
      IF  (IJK(KK2,IQ).EQ.0)  GO TO 18                                  HPTURB  1000
```

```
      IJK(KK1,IQ) = IJK(KK2,IQ)                                          HPTURB  1010
      IJK(KK2,IQ) = 0                                                    HPTURB  1020
   18 CONTINUE                                                           HPTURB  1030
      IJK(KK2,KK2) = IJ+1                                                HPTURB  1040
      IJK(KK2,KK1) = IJ+2                                                HPTURB  1050
      DO  20  J=1,3                                                      HPTURB  1060
      DO  19  I=1,3                                                      HPTURB  1070
      C(I,J,IJ+1) =  0.0                                                 HPTURB  1080
   19 C(I,J,IJ+2) =  0.0                                                 HPTURB  1090
      C(J,J,IJ+1) =  1.0                                                 HPTURB  1100
   20 C(J,J,IJ+2) = -1.0                                                 HPTURB  1110
      IJ   = IJ + 2                                                      HPTURB  1120
   21 CONTINUE                                                           HPTURB  1130
      K1   = K2 + 1                                                      HPTURB  1140
   22 CONTINUE                                                           HPTURB  1150
      MJ2 = -(KH0+NTP)                                                   HPTURB  1160
      IF  (NPRT(28).LT.3)  GO TO 29                                      HPTURB  1170
      NJ2 = -MJ2                                                         HPTURB  1180
      DO  25  J=1,NJ2                                                    HPTURB  1190
   25 WRITE  (6,26)  J,(RHS(I,J),I=1,3),(IJK(J,I),I=1,NJ2)               HPTURB  1200
   26 FORMAT (I6,3F12.6,20I4/(42X,20I4))                                 HPTURB  1210
      DO  27  KLM=1,IJ                                                   HPTURB  1220
   27 WRITE  (6,28)  KLM,((C(J,I,KLM),I=1,3),J=1,3)                      HPTURB  1230
   28 FORMAT (I6,9F12.6)                                                 HPTURB  1240
   29 CALL FSMSOL (C,RHS,IJK,MJ2,IJ,54,200)                              HPTURB  1250
      IF  (NPRT(28).LT.3)  GO TO 31                                      HPTURB  1260
      DO  30  J=1,NJ2                                                    HPTURB  1270
   30 WRITE  (6,26)  J,(RHS(I,J),I=1,3),(IJK(J,I),I=1,NJ2)               HPTURB  1280
   31 ONE = 1.0                                                          HPTURB  1290
      DELMAX = 0.0                                                       HPTURB  1300
      SCALE = 1.0                                                        HPTURB  1310
      DO  44  IT=1,2                                                     HPTURB  1320
      K1 = K0                                                            HPTURB  1330
      KH = 0                                                             HPTURB  1340
      KR = NTP                                                           HPTURB  1350
      DO  43  NB=J1,J2                                                   HPTURB  1360
      IF  (NPTPLY(NB).LE.0)  GO TO 43                                    HPTURB  1370
      K2 = K1 + NPTPLY(NB) - 1                                           HPTURB  1380
      DO  42  K=K1,K2                                                    HPTURB  1390
      KH = KH + 1                                                        HPTURB  1400
      KR = KR + 1                                                        HPTURB  1410
C                                                                        HPTURB  1420
CC    HERE K IS INDEX OF ALL POINTS IN PLAY                              HPTURB  1430
CC        KH IS INDEX OF ALL POINTS IN PLAY ON A SINGLE HARNESS          HPTURB  1440
CC        KR IS INDEX OF RHS ARRAY ELEMENTS                              HPTURB  1450
C                                                                        HPTURB  1460
      KI = NL(1,K)                                                       HPTURB  1470
      KS   = IABS(IBAR(1,KI))                                            HPTURB  1480
      IF  (KS.GT.100)  KS = MOD(KS,100)                                  HPTURB  1490
      IF  (IBAR(5,KI).EQ.0)  GO TO 32                                    HPTURB  1500
```

```
      CALL MAT31 (D(1,1,KS),RHS(1,KR),R)                                    HPTURB  1510
      GO TO 37                                                              HPTURB  1520
C                                                                           HPTURB  1530
C     NOTE: ENDPOINTS (K = K1 & K2) MUST BE TYPE 5.                         HPTURB  1540
C                                                                           HPTURB  1550
   32 CALL DOT31 (E(1,1,KH),RHS(1,KR),T1)                                   HPTURB  1560
      IF  (IT.EQ.2)  GO TO 33                                               HPTURB  1570
      DELMAX = DMAX1(DELMAX,DABS(T1(2)/DMIN1(BB(K),BB(K-1))))               HPTURB  1580
      GO TO 34                                                              HPTURB  1590
   33 BB(K  ) = BB(K  ) + SCALE*T1(2)                                       HPTURB  1600
      BB(K-1) = BB(K-1) - SCALE*T1(2)                                       HPTURB  1610
   34 DO  35  J=1,3                                                         HPTURB  1620
   35 T2(J) = T1(1)*E(J,1,KH) + T1(3)*E(J,3,KH)                             HPTURB  1630
      CALL MAT31 (D(1,1,KS),T2,R)                                           HPTURB  1640
      IF  (NPRT(28).GE.3)  WRITE (6,36) K,T1,T2,R                           HPTURB  1650
   36 FORMAT ('0',I6,3(3X,3F12.6))                                          HPTURB  1660
   37 IF  (IT.EQ.2)  GO TO 39                                               HPTURB  1670
      DO  38  J=1,3                                                         HPTURB  1680
   38 DELMAX = DMAX1(DELMAX,DABS(R(J)/DMAX1(EPS(1),DABS(BAR(J+3,KI)))))     HPTURB  1690
      GO TO 42                                                              HPTURB  1700
   39 DO  40  J=1,3                                                         HPTURB  1710
   40 BAR(J+3,KI) = BAR(J+3,KI) + SCALE*R(J)                                HPTURB  1720
      KE = IBAR(2,KI)                                                       HPTURB  1730
      IF  (KE.EQ.0)  GO TO 42                                               HPTURB  1740
      RER = XDY(BAR(4,KI),BD(7,KE),BAR(4,KI))                               HPTURB  1750
      IF  (RER.LE.1.0)  GO TO 42                                            HPTURB  1760
      SQRER = 1.0/DSQRT(RER)                                                HPTURB  1770
      DO  41  J=1,3                                                         HPTURB  1780
   41 BAR(J+3,KI) = SQRER*BAR(J+3,KI)                                       HPTURB  1790
   42 CONTINUE                                                              HPTURB  1800
      K1 = K2 + 1                                                           HPTURB  1810
   43 CONTINUE                                                              HPTURB  1820
      IF  (IT.EQ.2)  GO TO 44                                               HPTURB  1830
      IF  (DELMAX.NE.0.0)  SCALE = DMIN1(ONE,EPS(1)/DELMAX)                 HPTURB  1840
   44 CONTINUE                                                              HPTURB  1850
      IF  (NPRT(28).GE.2)  WRITE  (6,45)  ITER,DELMAX,SCALE                 HPTURB  1860
   45 FORMAT ('0  ITER =',I6,'  DELMAX =',F15.6,'  SCALE =',F15.6)          HPTURB  1870
      LAST = DELMAX.LE.EPS(2) .OR. ITER.EQ.MAXITR                           HPTURB  1880
      IF  (.NOT.LAST)  GO TO 52                                             HPTURB  1890
      KH = 0                                                                HPTURB  1900
      K1 = K0                                                               HPTURB  1910
      HLOSS(1,NH) = 0.0                                                     HPTURB  1920
      HLOSS(2,NH) = 0.0                                                     HPTURB  1930
      DO  51  NB=J1,J2                                                      HPTURB  1940
      BLOSS(1,NB) = 0.0                                                     HPTURB  1950
      BLOSS(2,NB) = 0.0                                                     HPTURB  1960
      IF  (NPTPLY(NB).LE.0)  GO TO 51                                       HPTURB  1970
      K2 = K1 + NPTPLY(NB) - 1                                              HPTURB  1980
      KK1 = NL(1,K1)                                                        HPTURB  1990
      KK2 = NL(1,K2)                                                        HPTURB  2000
```

```
      DO  46  K=KK1,KK2                                                  HPTURB  2010
      DO  46  J=1,3                                                      HPTURB  2020
   46 BAR(J+12,K) = 0.0                                                  HPTURB  2030
      IF  (DHT.EQ.0.0)  GO TO 49                                         HPTURB  2040
      DO  48  K=K1,K2                                                    HPTURB  2050
      KH = KH + 1                                                        HPTURB  2060
      KI = NL(1,K)                                                       HPTURB  2070
      PLOSS(2,KI ) = PLOSS(2,KI ) + DHT*PTLOSS(2,KH)                     HPTURB  2080
      IF  (K.EQ.K1)  GO TO 47                                            HPTURB  2090
      BBDOT(K-1) = (BB(K-1)-OLDBB(K-1))/DHT                              HPTURB  2100
      PLOSS(1,K-1) = PLOSS(1,K-1) + DHT*PTLOSS(1,KH-1)                   HPTURB  2110
      BLOSS(1,NB) = BLOSS(1,NB) + PLOSS(1,K-1)                           HPTURB  2120
   47 DO  48  J=1,3                                                      HPTURB  2130
   48 BAR(J+12,KI) = (BAR(J+3,KI)-BAR(J,KI))/DHT                         HPTURB  2140
      BBDOT(K2) = 0.0                                                    HPTURB  2150
      PLOSS(1,K2) = 0.0                                                  HPTURB  2160
   49 K1 = K2+1                                                          HPTURB  2170
      DO  50  K=KK1,KK2                                                  HPTURB  2180
   50 BLOSS(2,NB) = BLOSS(2,NB) + PLOSS(2,K)                             HPTURB  2190
      HLOSS(1,NH) = HLOSS(1,NH) + BLOSS(1,NB)                            HPTURB  2200
      HLOSS(2,NH) = HLOSS(2,NH) + BLOSS(2,NB)                            HPTURB  2210
   51 CONTINUE                                                           HPTURB  2220
   52 IF  (NPRT(28).EQ.0)  GO TO 59                                      HPTURB  2230
      IF  (.NOT.LAST .AND. IABS(NPRT(28)).EQ.1)  GO TO 59                HPTURB  2240
      K1 = K0                                                            HPTURB  2250
      KH = 0                                                             HPTURB  2260
      DO  57  NB=J1,J2                                                   HPTURB  2270
      IF  (NPTPLY(NB).LE.0)  GO TO 57                                    HPTURB  2280
      WRITE (6,53)  NB,NH                                                HPTURB  2290
   53 FORMAT ('0  BELT NO.',I4,' OF HARNESS NO.',I4)                     HPTURB  2300
      K2 = K1 + NPTPLY(NB) - 1                                           HPTURB  2310
      DO  54  K=K1,K2                                                    HPTURB  2320
      KH = KH + 1                                                        HPTURB  2330
      KI = NL(1,K)                                                       HPTURB  2340
      KS = IBAR(1,KI)                                                    HPTURB  2350
      BK = 0.0                                                           HPTURB  2360
      IF  (K.NE.K1)  BK = BB(K-1)                                        HPTURB  2370
      PLS = 0.0                                                          HPTURB  2380
      IF  (K.NE.K1)  PLS = PLOSS(1,K-1)                                  HPTURB  2390
      T(1) = BAR(4,KI)                                                   HPTURB  2400
      T(2) = BAR(5,KI)                                                   HPTURB  2410
      T(3) = BAR(6,KI)                                                   HPTURB  2420
      KJ = MOD(IABS(KS),100)                                             HPTURB  2430
      IF  (LPMI(KJ).NE.0)  CALL DOT31 (DPMI(1,1,KJ),BAR(4,KI),T)         HPTURB  2440
   54 WRITE (6,55) K,KI,KS,BK,PLS,(T(J),J=1,3),                          HPTURB  2450
     *            (FCE(J,KH),J=1,3),PLOSS(2,KI)                        HPTURB  2460
   55 FORMAT (3I8,F10.3,F12.3,2X,3F9.3,3X,3F11.3,3X,F12.3)               HPTURB  2470
      IF  (LAST)  WRITE (6,56)  BLOSS(1,NB),BLOSS(2,NB)                  HPTURB  2480
   56 FORMAT ('0    TOTAL BELT ENERGY LOSS',7X,F12.3,68X,F12.3)          HPTURB  2490
      K1 = K2 + 1                                                        HPTURB  2500
```

```
   57 CONTINUE                                                          HPTURB  2510
      IF  (LAST)  WRITE (6,58)  HLOSS(1,NH),HLOSS(2,NH)                 HPTURB  2520
   58 FORMAT ('0 TOTAL HARNESS ENERGY LOSS',7X,F12.3,68X,F12.3)         HPTURB  2530
   59 ITER = ITER + 1                                                   HPTURB  2540
C                                                                       HPTURB  2550
C       END OF   DO  59  ITER=1,MAXITR   LOOP                           HPTURB  2560
C                                                                       HPTURB  2570
      IF  (.NOT.LAST)  GO TO 13                                         HPTURB  2580
      IF  (ITER.GT.MAXITR)  WRITE  (6,60)  MAXITR,TSEC,DELMAX,SCALE     HPTURB  2590
   60 FORMAT ('0 HPTURB ITER =',I4,' AT TIME =',F8.3,                   HPTURB  2600
     *        ' MSEC.  DELMAX =',F10.6,'  SCALE =',F10.6)               HPTURB  2610
      J1 = J2 + 1                                                       HPTURB  2620
      K0 = K1                                                           HPTURB  2630
   61 CONTINUE                                                          HPTURB  2640
      IF  (NPRT(28).LT.0)  NPRT(28) = 0                                 HPTURB  2650
      CALL ELTIME (2,39)                                                HPTURB  2660
      RETURN                                                            HPTURB  2670
      END                                                               HPTURB  2680
```

```
      SUBROUTINE HSETC (NPTS,KH0,KNL0,NTP,IJ)                                HSETC   0010
C                                                          REV 20 06/18/80    HSETC   0020
      IMPLICIT REAL*8 (A-H,O-Z)                                              HSETC   0030
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),     HSETC   0040
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)            HSETC   0050
      COMMON/TABLES/ MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(500),TAB(2600)     HSETC   0060
      COMMON/HRNESS/ BAR(15,100),BB(100),BBDOT(100),PLOSS(2,100),            HSETC   0070
     *               XLONG(20),HTIME(2),IBAR(5,100),NL(2,100),               HSETC   0080
     *               NPTSPB(20),NPTPLY(20),NTHRNS(20),NBLTPH(5)              HSETC   0090
C     THIS COMMON/TEMPVS/ IS SHARED BY HPTURB, HBPLAY, HBELT AND HSETC.      HSETC   0100
      COMMON/TEMPVS/ B(3,3,3),S(3,3),T(3),R(3),V(3),T1(3),T2(3),             HSETC   0110
     *               E(3,3,50),EDOT(3,50),FCE(3,50),FR(3,50),ZR(3,50),       HSETC   0120
     *               TR(3,50),U(3,50),PTLOSS(2,50),BL(50),FB(50),FP(50),     HSETC   0130
     *               OLDBB(100),RHS(3,54),C(3,3,200),IJK(54,54)              HSETC   0140
      DIMENSION  KM(3),MK(2)                                                 HSETC   0150
      ONE = 1.0                                                              HSETC   0160
      KNL = KNL0                                                             HSETC   0170
      KH = KH0                                                               HSETC   0180
      K1 = KH0 + NTP + 1                                                     HSETC   0190
      K2 = KH0 + NTP + NPTS                                                  HSETC   0200
      DO  60  K=K1,K2                                                        HSETC   0210
C                                                                            HSETC   0220
C     HERE K   IS INDEX OF IJK AND RHS ARRAYS                                HSETC   0230
C          KH  IS INDEX OF POINTS IN PLAY ON EACH HARNESS                    HSETC   0240
C          KNL IS INDEX OF ALL POINTS IN PLAY                                HSETC   0250
C          KI  IS INDEX OF ALL POINTS                                        HSETC   0260
C                                                                            HSETC   0270
      KH = KH + 1                                                            HSETC   0280
      KNL = KNL + 1                                                          HSETC   0290
C                                                                            HSETC   0300
C     ZERO C(K,K) , C(K,K-1) , C(K,K+1) & RHS(K); SET IJK(K,K) = IJ          HSETC   0310
C                                                                            HSETC   0320
      KM(1) = K+1                                                            HSETC   0330
      KM(2) = K-1                                                            HSETC   0340
      KM(3) = K                                                              HSETC   0350
      IF  (K.EQ.K2)  KM(1) = 0                                               HSETC   0360
      IF  (K.EQ.K1)  KM(2) = 0                                               HSETC   0370
      KK = IJ                                                                HSETC   0380
      DO  12  L=1,3                                                          HSETC   0390
      RHS(L,K) = 0.0                                                         HSETC   0400
      IF  (KM(L).EQ.0)  GO TO 12                                             HSETC   0410
      KK = KK+1                                                              HSETC   0420
      DO  11  I=1,3                                                          HSETC   0430
      DO  11  J=1,3                                                          HSETC   0440
   11 C(I,J,KK) = 0.0                                                        HSETC   0450
   12 CONTINUE                                                               HSETC   0460
      IJ = IJ+1                                                              HSETC   0470
      IJK(K,K  ) = IJ                                                        HSETC   0480
C                                                                            HSETC   0490
C     COMPUTE CNORM; IF ZERO, SET C(K,K) = I                                 HSETC   0500
```

```
C                                                                       HSETC   0510
      CNORM = 0.0                                                       HSETC   0520
      IF  (K.NE.K2)  CNORM = FB(KH)/BL(KH)                              HSETC   0530
      IF  (K.NE.K1)  CNORM = CNORM + FB(KH-1)/BL(KH-1)                  HSETC   0540
      KI = NL(1,KNL)                                                    HSETC   0550
      IF  (IABS(IBAR(1,KI)).GT.100)  GO TO 14                           HSETC   0560
      IF  (CNORM.NE.0.0)  GO TO 14                                      HSETC   0570
      KK = IJK(K,K)                                                     HSETC   0580
      DO  13  I=1,3                                                     HSETC   0590
   13 C(I,I,KK) = ONE                                                   HSETC   0600
      GO TO 60                                                          HSETC   0610
   14 KK = IBAR(3,KI)                                                   HSETC   0620
      NFD = NTAB(KK+1)                                                  HSETC   0630
      NFR = NTAB(KK+5)                                                  HSETC   0640
C                                                                       HSETC   0650
C     SET UP B(3,3,3) AND S(3,3)                                        HSETC   0660
C                                                                       HSETC   0670
      MK(1) = KH                                                        HSETC   0680
      MK(2) = KH-1                                                      HSETC   0690
      IF  (K.EQ.K2)  MK(1) = 0                                          HSETC   0700
      IF  (K.EQ.K1)  MK(2) = 0                                          HSETC   0710
      DO  18 M=1,2                                                      HSETC   0720
      KK = MK(M)                                                        HSETC   0730
      IF  (KK.NE.0 .AND. CNORM.NE.0.0)  GO TO 16                        HSETC   0740
      DO  15  I=1,3                                                     HSETC   0750
      S(I,M) = 0.0                                                      HSETC   0760
      DO  15  J=1,3                                                     HSETC   0770
   15 B(I,J,M) = 0.0                                                    HSETC   0780
      GO TO 18                                                          HSETC   0790
   16 CALL DOT31 (E(1,1,KH),U(1,KK),T)                                  HSETC   0800
      KIM = KNL + 1 - M                                                 HSETC   0810
      FB1 = FB(KK)/BL(KK)                                               HSETC   0820
      FB2 = FP(KK)/BB(KIM) - FB1                                        HSETC   0830
      FB3 = FP(KK)*BL(KK)/BB(KIM)**2                                    HSETC   0840
      DO  17  I=1,3                                                     HSETC   0850
      SGN = ONE                                                         HSETC   0860
      IF  (FR(I,KH).LT.0.0)  SGN = -ONE                                 HSETC   0870
      S(I,M) = SGN*(FB3*T(I))                                           HSETC   0880
      DO  17  J=1,3                                                     HSETC   0890
   17 B(I,J,M) = SGN*(FB1*E(J,I,KH) + FB2*T(I)*U(J,KK))                 HSETC   0900
   18 CONTINUE                                                          HSETC   0910
      DO  19  I=1,3                                                     HSETC   0920
      S(I,3) = -(S(I,1) + S(I,2))                                       HSETC   0930
      DO  19  J=1,3                                                     HSETC   0940
   19 B(I,J,3) = -(B(I,J,1) + B(I,J,2))                                 HSETC   0950
      IF  (NFR.EQ.0)    GO TO 20                                        HSETC   0960
      R(1) = TAB(NFR+2)                                                 HSETC   0970
      R(2) = TAB(NFR+4)                                                 HSETC   0980
   20 R(3) = 0.0                                                        HSETC   0990
      DO  50  M=1,3                                                     HSETC   1000
```

```
      RH = 0.0                                                                  HSETC   1010
      IF  (M.EQ.3)       GO TO 31                                               HSETC   1020
      IF  (NFR.EQ.0)     GO TO 48                                               HSETC   1030
C                                                                               HSETC   1040
C     CONSTRAINTS 1 AND 2                                                       HSETC   1050
C                                                                               HSETC   1060
      SGN   = -ONE                                                              HSETC   1070
      FR3   = DABS(FR(M,KH)) - R(M)*DABS(FR(3,KH))                              HSETC   1080
      IF  (IBAR(1,KI).GT.0)  RH = FR3                                           HSETC   1090
      IF  (FR3.LE.0.0)  GO TO 48                                                HSETC   1100
      GO TO 40                                                                  HSETC   1110
C                                                                               HSETC   1120
C     CONSTRAINT NO. 3                                                          HSETC   1130
C                                                                               HSETC   1140
   31 IF  (NFD.EQ.0)     GO TO 48                                               HSETC   1150
      IF  (IBAR(1,KI).LT.0)  GO TO 40                                           HSETC   1160
      SGN   = ONE                                                               HSETC   1170
      RMAG2 = TR(1,KH)**2 + TR(2,KH)**2 + TR(3,KH)**2                           HSETC   1180
      RMAG  = DSQRT(RMAG2)                                                      HSETC   1190
      RER2  = TR(1,KH)*E(1,3,KH)+ TR(2,KH)*E(2,3,KH)+ TR(3,KH)*E(3,3,KH)        HSETC   1200
      RER2  = EDOT(3,KH)*RER2                                                   HSETC   1210
      RER   = DSQRT(RER2)                                                       HSETC   1220
      PEN   = RMAG/RER - RMAG                                                   HSETC   1230
      RRDOT = BAR(4,KI)*BAR(13,KI)                                              HSETC   1240
     *      + BAR(5,KI)*BAR(14,KI)                                             HSETC   1250
     *      + BAR(6,KI)*BAR(15,KI)                                             HSETC   1260
      KS = IABS(IBAR(1,KI))                                                     HSETC   1270
      IF  (KS.GT.100)  KS = MOD(KS,100)                                         HSETC   1280
      CALL DOT31 (D(1,1,KS),BAR(13,KI),T)                                       HSETC   1290
      ERDOT = E(1,3,KH)*T(1) + E(2,3,KH)*T(2) + E(3,3,KH)*T(3)                  HSETC   1300
      C1    = PEN/RMAG2                                                         HSETC   1310
      C2    = RMAG*EDOT(3,KH)/(RER*RER2)                                        HSETC   1320
      PDOT  = C1*RRDOT - C2*ERDOT                                               HSETC   1330
      CALL FRCDFL (PEN,PDOT,NFD,0,FDP,ELOSS)                                    HSETC   1340
      CALL FRCDFL (PEN,PDOT,NFD,1,FD ,ELOSS)                                    HSETC   1350
      RH    = FD + FR(3,KH)                                                     HSETC   1360
      PTLOSS(2,KH) = ELOSS                                                      HSETC   1370
      C1    = FDP*C1                                                            HSETC   1380
      C2    = FDP*C2                                                            HSETC   1390
      SGNB3 = -DSIGN(ONE,FR(3,KH))                                              HSETC   1400
      DO  32  J=1,3                                                             HSETC   1410
   32 B(3,J,3) = SGNB3*B(3,J,3) - C1*TR(J,KH) + C2*E(J,3,KH)                    HSETC   1420
   40 DO  47  LL=1,3                                                            HSETC   1430
      L = 4 - LL                                                                HSETC   1440
      IF  (KM(L).EQ.0)  GO TO 47                                                HSETC   1450
      DO  42  J=1,3                                                             HSETC   1460
   42 V(J) = R(M)*B(3,J,L) + SGN*B(M,J,L)                                       HSETC   1470
      KL = KM(L)                                                                HSETC   1480
      KML = KNL + KL - K                                                        HSETC   1490
      KIL = NL(1,KML)                                                           HSETC   1500
```

```
      IF  (IBAR(5,KIL).NE.0)  GO TO 43                                  HSETC   1510
      KHL = KH + KL - K                                                 HSETC   1520
      CALL DOT31 (E(1,1,KHL),V,T)                                       HSETC   1530
      T(2) = R(M)*S(3,L) + SGN*S(M,L)                                   HSETC   1540
      CALL MAT31 (E(1,1,KHL),T,V)                                       HSETC   1550
   43 IF  (LL.NE.1)  GO TO 44                                           HSETC   1560
      VE = V(1)*E(1,M,KH) + V(2)*E(2,M,KH) + V(3)*E(3,M,KH)             HSETC   1570
      EV = 1.0                                                          HSETC   1580
      IF  (IABS(IBAR(1,KI)).LT.100)                                     HSETC   1590
     *          EV = DSIGN(ONE,VE)/DSQRT(V(1)**2+V(2)**2+V(3)**2)       HSETC   1600
      RH = EV*RH                                                        HSETC   1610
   44 IF  (IJK(K,KL).NE.0)  GO TO 45                                    HSETC   1620
      IJ = IJ+1                                                         HSETC   1630
      IJK(K,KL) = IJ                                                    HSETC   1640
   45 KK = IJK(K,KL)                                                    HSETC   1650
      DO  46  J=1,3                                                     HSETC   1660
      VEV = EV*V(J)                                                     HSETC   1670
      DO  46  I=1,3                                                     HSETC   1680
   46 C(I,J,KK) = C(I,J,KK) + E(I,M,KH)*VEV                             HSETC   1690
   47 CONTINUE                                                          HSETC   1700
      DO  41  I=1,3                                                     HSETC   1710
   41 RHS(I,K) = RHS(I,K) + RH*E(I,M,KH)                                HSETC   1720
      GO TO 50                                                          HSETC   1730
   48 IF  (IBAR(1,KI).LE.0)  GO TO 50                                   HSETC   1740
      KK = IJK(K,K)                                                     HSETC   1750
      DO  49  I=1,3                                                     HSETC   1760
      DO  49  J=1,3                                                     HSETC   1770
   49 C(I,J,KK) = C(I,J,KK) + E(I,M,KH)*E(J,M,KH)                       HSETC   1780
   50 CONTINUE                                                          HSETC   1790
   60 CONTINUE                                                          HSETC   1800
      RETURN                                                            HSETC   1810
      END                                                               HSETC   1820
```

```
      SUBROUTINE IMPLS2(MODE,J,H)                                          IMPLS2  0010
C                                                          REV 19 09/05/78  IMPLS2  0020
C     CALLED BY SUBROUTINE UPDATE WHEN JOINT J LOCKS TO APPLY IMPULSE      IMPLS2  0030
C     TO SET P.(D(M)'W(M) - D(N)'W(N)) = 0                                 IMPLS2  0040
C                                                                          IMPLS2  0050
C     ARGUMENTS:                                                           IMPLS2  0060
C          MODE -  0: FULL LOCK          P = I                             IMPLS2  0070
C                  1: AXIS (H) FREE      P = I-HH'                         IMPLS2  0080
C                 -1: AXIS (H) LOCKED    P = HH'                           IMPLS2  0090
C                                                                          IMPLS2  0100
C          J     - JOINT IDENTIFICATION NUMBER                             IMPLS2  0110
C                                                                          IMPLS2  0120
C          H     - AXIS VECTOR                                             IMPLS2  0130
C                                                                          IMPLS2  0140
C                                                                          IMPLS2  0150
      IMPLICIT REAL*8(A-H,O-Z)                                             IMPLS2  0160
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,              IMPLS2  0170
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)           IMPLS2  0180
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),   IMPLS2  0190
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)          IMPLS2  0200
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(3,60),HA(3,60),HB(3,60),    IMPLS2  0210
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),        IMPLS2  0220
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)       IMPLS2  0230
      COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60),   IMPLS2  0240
     *               F(3,30),TQ(3,30),WJ(30)                               IMPLS2  0250
      COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),      IMPLS2  0260
     *               HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12),   IMPLS2  0270
     *               RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),                 IMPLS2  0280
     *               KQ1(12),KQ2(12),KQTYPE(12)                            IMPLS2  0290
      COMMON/FLXBLE/ HF(4,12,8),B42(3,3,24),V4(3,8),NFLEX(3,8)             IMPLS2  0300
      COMMON/TEMPVS/ SM(3),SN(3),TM(3,3),TN(3,3),T(3,4),TT(3,4)            IMPLS2  0310
      DIMENSION      TWA(3,3,30),TLA(3,3,30),H(3)                          IMPLS2  0320
      CALL ELTIME(1,28)                                                    IMPLS2  0330
      M = JNT(J)                                                           IMPLS2  0340
      N = J+1                                                              IMPLS2  0350
      DO 20 L=1,3                                                          IMPLS2  0360
      DO 12 K=1,NGRND                                                      IMPLS2  0370
      DO 12 I=1,3                                                          IMPLS2  0380
      U1(I,K) = 0.0                                                        IMPLS2  0390
   12 U2(I,K) = 0.0                                                        IMPLS2  0400
      DO 13 K=1,NJNT                                                       IMPLS2  0410
      DO 13 I=1,3                                                          IMPLS2  0420
      V1(I,K) = 0.0                                                        IMPLS2  0430
   13 V2(I,K) = 0.0                                                        IMPLS2  0440
      IF (NQ.LE.0) GO TO 15                                                IMPLS2  0450
      DO 14 K=1,NQ                                                         IMPLS2  0460
      DO 14 I=1,3                                                          IMPLS2  0470
   14 V3(I,K) = 0.0                                                        IMPLS2  0480
   15 IF (NFLX.EQ.0) GO TO 18                                              IMPLS2  0490
      DO 19 K=1,NFLX                                                       IMPLS2  0500
```

```
      DO 19 I=1,3                                                       IMPLS2  0510
   19 V4(I,K) = 0.0                                                     IMPLS2  0520
   18 DO 16 I=1,3                                                       IMPLS2  0530
      U2(I,M) = RPHI(I,M)*D(I,L,M)                                      IMPLS2  0540
   16 U2(I,N) =-RPHI(I,N)*D(I,L,N)                                      IMPLS2  0550
      CALL DAUX(L)                                                      IMPLS2  0560
      DO 17 K=1,NGRND                                                   IMPLS2  0570
      DO 17 I=1,3                                                       IMPLS2  0580
      TLA(I,L,K) = SEGLA(I,K)                                           IMPLS2  0590
   17 TWA(I,L,K) = WMEGD(I,K)                                           IMPLS2  0600
   20 CONTINUE                                                          IMPLS2  0610
      CALL DOT33(D(1,1,M),TWA(1,1,M),TM)                                IMPLS2  0620
      CALL DOT33(D(1,1,N),TWA(1,1,N),TN)                                IMPLS2  0630
      CALL DOT31(D(1,1,M),WMEG(1,M), SM)                                IMPLS2  0640
      CALL DOT31(D(1,1,N),WMEG(1,N), SN)                                IMPLS2  0650
      DO 22 I=1,3                                                       IMPLS2  0660
      DO 21 K=1,3                                                       IMPLS2  0670
      T(I,K) = TM(I,K) - TN(I,K)                                        IMPLS2  0680
   21 TT(I,K) = T(I,K)                                                  IMPLS2  0690
      T(I,4) = SN(I) - SM(I)                                            IMPLS2  0700
   22 TT(I,4) = H(I)                                                    IMPLS2  0710
      IF (MODE.GE.0) CALL DSMSOL(T,3,3)                                 IMPLS2  0720
      IF (MODE.GT.0) CALL DSMSOL(TT,3,3)                                IMPLS2  0730
      IF (MODE) 24,29,25                                                IMPLS2  0740
   24 ST = 0.0                                                          IMPLS2  0750
      STT = XDY(H,T,H)                                                  IMPLS2  0760
      GO TO 26                                                          IMPLS2  0770
   25 ST = 1.0                                                          IMPLS2  0780
      STT = -(H(1)*TT(1,4) + H(2)*TT(2,4) + H(3)*TT(3,4))               IMPLS2  0790
   26 STT =  (H(1)* T(1,4) + H(2)* T(2,4) + H(3)* T(3,4))/STT           IMPLS2  0800
      DO 27 I=1,3                                                       IMPLS2  0810
   27 T(I,4) = ST*T(I,4) + STT*TT(I,4)                                  IMPLS2  0820
   29 DO 30 K=1,NGRND                                                   IMPLS2  0830
      DO 30 I=1,3                                                       IMPLS2  0840
      DO 30 L=1,3                                                       IMPLS2  0850
      SEGLV(I,K) = SEGLV(I,K) + T(L,4)*TLA(I,L,K)                       IMPLS2  0860
   30 WMEG (I,K) = WMEG (I,K) + T(L,4)*TWA(I,L,K)                       IMPLS2  0870
      IF  (NPRT(3).NE.0) CALL PRINT(6HIMPLS2)                           IMPLS2  0880
      CALL ELTIME(2,28)                                                 IMPLS2  0890
      RETURN                                                            IMPLS2  0900
      END                                                               IMPLS2  0910
```

```
      SUBROUTINE IMPULS(I1,I2,I3)                                          IMPULS  0010
C                                                          REV 19 09/05/78  IMPULS  0020
C         ARGUMENTS: I1 = 1 - IMPULS FOR PLELP.                             IMPULS  0030
C                     3 - IMPULS FOR SEGSEG.                               IMPULS  0040
C                     4 - IMPULS FOR VISPR OR EJOINT                       IMPULS  0050
C                I2 = INDEX OF CONTACTING SEGMENT OR JOINT AXIS            IMPULS  0060
C                I3 = INDEX OF PLANE, SEGMENT OR JOINT AXIS                IMPULS  0070
C       .                                                                  IMPULS  0080
C                                                                          IMPULS  0090
      IMPLICIT REAL*8 (A-H,O-Z)                                            IMPULS  0100
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,              IMPULS  0110
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)           IMPULS  0120
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),   IMPULS  0130
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)          IMPULS  0140
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(3,60),HA(3,60),HB(3,60),    IMPULS  0150
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),        IMPULS  0160
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)       IMPULS  0170
      COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60),   IMPULS  0180
     *               F(3,30),TQ(3,30),WJ(30)                               IMPULS  0190
      COMMON/JBARTZ/ MNPL(   30),MNBLT(    8),MNSEG(   30),MNBAG(    6),  IMPULS  0200
     *               MPL(3,5,30),MBLT(3,5,8),MSEG(3,5,30),MBAG(3,10,6),    IMPULS  0210
     *               NTPL( 5,30),NTBLT( 5,8),NTSEG( 5,30)                  IMPULS  0220
      COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),      IMPULS  0230
     *               HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12),   IMPULS  0240
     *               RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),                 IMPULS  0250
     *               KQ1(12),KQ2(12),KQTYPE(12)                            IMPULS  0260
      COMMON/FLXBLE/ HF(4,12,8),B42(3,3,24),V4(3,8),NFLEX(3,8)             IMPULS  0270
      COMMON/TEMPVI/ CREST,TTI(3),R1I(3),R2I(3),JSTOP(4,2,30)              IMPULS  0280
      COMMON/TABLES/ MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(500),TAB(2600)   IMPULS  0290
      DIMENSION TEMP(3),DWR1(3),DWR2(3),DWR3(3),DWR4(3),VREL(3),DV(3)      IMPULS  0300
      IF (TIME.EQ.0.0) GO TO 99                                            IMPULS  0310
C                                                                          IMPULS  0320
C     SPECIAL SETUP FOR CALL TO SUBROUTINE DAUX                            IMPULS  0330
C     REPLACE SETUP WITH U1,U2,V1,V2,V3 = 0.                               IMPULS  0340
C     ASSUME OTHER ARRAYS FROM PREVIOUS CALL TO DAUX.                      IMPULS  0350
C                                                                          IMPULS  0360
      CALL ELTIME(1,27)                                                    IMPULS  0370
      CALL OUTPUT(0)                                                       IMPULS  0380
      KQTEST = 0                                                           IMPULS  0390
      NT = 0                                                               IMPULS  0400
      IF (I1.EQ.1) NT = NTPL (I2,I3)                                       IMPULS  0410
      IF (I1.EQ.3) NT = NTSEG(I2,I3)                                       IMPULS  0420
      IF (NT.EQ.0) GO TO 29                                                IMPULS  0430
      KQ = -NTAB(NT+1)                                                     IMPULS  0440
      IF (KQ.LE.0) GO TO 29                                                IMPULS  0450
      KQTYPE(KQ) = IABS(KQTYPE(KQ))                                        IMPULS  0460
      CALL DAUX(0)                                                         IMPULS  0470
   29 IF (NQ.LE.0) GO TO 31                                                IMPULS  0480
      DO 30 J=1,NQ                                                         IMPULS  0490
      DO 30 I=1,3                                                          IMPULS  0500
```

```
   30 V3(I,J) = 0.0                                                      IMPULS  0510
   31 DO 32 J=1,NGRND                                                    IMPULS  0520
      DO 32 I=1,3                                                        IMPULS  0530
      U1(I,J) = 0.0                                                      IMPULS  0540
   32 U2(I,J) = 0.0                                                      IMPULS  0550
      IF (NJNT.LE.0) GO TO 21                                            IMPULS  0560
      DO 33 J=1,NJNT                                                     IMPULS  0570
      DO 33 I=1,3                                                        IMPULS  0580
      V1(I,J) = 0.0                                                      IMPULS  0590
   33 V2(I,J) = 0.0                                                      IMPULS  0600
   21 IF (NFLX.EQ.0) GO TO 23                                            IMPULS  0610
      DO 22 J=1,NFLX                                                     IMPULS  0620
      DO 22 I=1,3                                                        IMPULS  0630
   22 V4(I,J) = 0.0                                                      IMPULS  0640
C                                                                        IMPULS  0650
C     REPLACE CALLS TO CONTACT AND VISPR WITH SINGLE CALL                IMPULS  0660
C     AT FIRST CONTACT IF NOT CONSTRAINT.                                IMPULS  0670
C                                                                        IMPULS  0680
   23 IF (I1.NE.1) GO TO 34                                              IMPULS  0690
      NT = NTPL(I2,I3)                                                   IMPULS  0700
      M1 = MPL(1,I2,I3)                                                  IMPULS  0710
      M2 = MPL(2,I2,I3)                                                  IMPULS  0720
      M3 = MPL(3,I2,I3)                                                  IMPULS  0730
      CALL PLELP(M2,M3,M1,I3,NT)                                         IMPULS  0740
      IF (NTAB(NT+1).LT.0) GO TO 37                                      IMPULS  0750
      K1 = M2                                                            IMPULS  0760
      K2 = M1                                                            IMPULS  0770
      GO TO 39                                                           IMPULS  0780
   34 IF (I1.NE.3) GO TO 35                                              IMPULS  0790
      NT = NTSEG(I2,I3)                                                  IMPULS  0800
      M1 = MSEG(1,I2,I3)                                                 IMPULS  0810
      M2 = MSEG(2,I2,I3)                                                 IMPULS  0820
      M3 = MSEG(3,I2,I3)                                                 IMPULS  0830
      CALL SEGSEG(I3,M1,M2,M3,NT)                                        IMPULS  0840
      IF (NTAB(NT+1).LT.0) GO TO 37                                      IMPULS  0850
      K1 = I3                                                            IMPULS  0860
      K2 = M2                                                            IMPULS  0870
      GO TO 39                                                           IMPULS  0880
   35 IF (I1.NE.4) WRITE (6,36) I1,I2,I3                                 IMPULS  0890
   36 FORMAT('0 IMPROPER ARGUMENTS TO SUBROUTINE IMPULS'/                IMPULS  0900
     *       '  ARGUMENTS = ', 3I6 /                                     IMPULS  0910
     *       '  PROGRAM TERMINATED! )                                    IMPULS  0920
      IF  (I1.NE.4)  STOP 33                                             IMPULS  0930
C                                                                        IMPULS  0940
C     RECALL VISPR FOR JOINT STOP.                                       IMPULS  0950
C                                                                        IMPULS  0960
      IF (IABS(IPIN(I3)).NE.4) GO TO 25                                  IMPULS  0970
      CALL EJOINT(I2,I3)                                                 IMPULS  0980
      GO TO 26                                                           IMPULS  0990
   25 CALL VISPR(I2,I3)                                                  IMPULS  1000
```

```
   26 K1 = IABS(JNT(I3))                                                  IMPULS  1010
      K2 = I3+1                                                           IMPULS  1020
      GO TO 39                                                            IMPULS  1030
C                                                                         IMPULS  1040
C     SET UP SPECIAL U1,U2 FOR FIRST CONTACT OF CONSTRAINT.               IMPULS  1050
C                                                                         IMPULS  1060
   37 KQ = -NTAB(NT+1)                                                    IMPULS  1070
      KQTEST = 1                                                          IMPULS  1080
      KQTYPE(KQ) = -IABS(KQTYPE(KQ))                                      IMPULS  1090
      K1 = KQ1(KQ)                                                        IMPULS  1100
      K2 = KQ2(KQ)                                                        IMPULS  1110
      IF (K1.GT.NSEG) GO TO 38                                            IMPULS  1120
      CALL MAT31(A13(1,1,2*KQ-1),QQ(1,KQ),U1(1,K1))                       IMPULS  1130
      CALL MAT31(A23(1,1,2*KQ-1),QQ(1,KQ),U2(1,K1))                       IMPULS  1140
   38 IF (K2.GT.NSEG) GO TO 39                                            IMPULS  1150
      CALL MAT31(A13(1,1,2*KQ  ),QQ(1,KQ),U1(1,K2))                       IMPULS  1160
      CALL MAT31(A23(1,1,2*KQ  ),QQ(1,KQ),U2(1,K2))                       IMPULS  1170
C                                                                         IMPULS  1180
C     FINAL SETUP OF U1 AND U2                                            IMPULS  1190
C                                                                         IMPULS  1200
   39 DO 40 J=1,NGRND                                                     IMPULS  1210
      DO 40 I=1,3                                                         IMPULS  1220
      U1(I,J) = U1(I,J)*RW(J)                                             IMPULS  1230
   40 U2(I,J) = U2(I,J)*RPHI(I,J)                                         IMPULS  1240
      CALL DAUX(I1)                                                       IMPULS  1250
      IF (KQTEST.EQ.1) KQTYPE(KQ) = IABS(KQTYPE(KQ))                      IMPULS  1260
      IF (NPRT(10).NE.0) CALL PRINT(6HPREIMP)                             IMPULS  1270
      IF (I1.GT.3) GO TO 51                                               IMPULS  1280
      IF (NPRT(10).NE.0) WRITE (6,42) R1I,R2I                             IMPULS  1290
   42 FORMAT ('0'/(6G20.8))                                               IMPULS  1300
      CALL CROSS(WMEG (1,K1),R1I(1),TEMP)                                 IMPULS  1310
      CALL DOT31(D(1,1,K1),TEMP,DWR1(1))                                  IMPULS  1320
      CALL CROSS(WMEG (1,K2),R2I(1),TEMP)                                 IMPULS  1330
      CALL DOT31(D(1,1,K2),TEMP,DWR2(1))                                  IMPULS  1340
      CALL CROSS(WMEGD(1,K1),R1I(1),TEMP)                                 IMPULS  1350
      CALL DOT31(D(1,1,K1),TEMP,DWR3(1))                                  IMPULS  1360
      CALL CROSS(WMEGD(1,K2),R2I(1),TEMP)                                 IMPULS  1370
      CALL DOT31(D(1,1,K2),TEMP,DWR4(1))                                  IMPULS  1380
      TVREL = 0.0                                                         IMPULS  1390
      TDV   = 0.0                                                         IMPULS  1400
      DO 50 I=1,3                                                         IMPULS  1410
      VREL(I) = SEGLV(I,K1)+DWR1(I) - SEGLV(I,K2)-DWR2(I)                 IMPULS  1420
      DV  (I) = SEGLA(I,K1)+DWR3(I) - SEGLA(I,K2)-DWR4(I)                 IMPULS  1430
      TVREL = TVREL + TTI(I)*VREL(I)                                      IMPULS  1440
   50 TDV   = TDV   + TTI(I)*DV  (I)                                      IMPULS  1450
      GO TO 53                                                            IMPULS  1460
   51 CALL DOT31(D(1,1,K1),WMEG (1,K1),DWR1(1))                           IMPULS  1470
      CALL DOT31(D(1,1,K2),WMEG (1,K2),DWR2(1))                           IMPULS  1480
      CALL DOT31(D(1,1,K1),WMEGD(1,K1),DWR3(1))                           IMPULS  1490
      CALL DOT31(D(1,1,K2),WMEGD(1,K2),DWR4(1))                           IMPULS  1500
```

```
      TVREL = 0.0                                                          IMPULS  1510
      TDV   = 0.0                                                          IMPULS  1520
      DO 52 I=1,3                                                          IMPULS  1530
      VREL(I) = DWR1(I) - DWR2(I)                                          IMPULS  1540
      DV  (I) = DWR3(I) - DWR4(I)                                          IMPULS  1550
      TVREL = TVREL + TTI(I)*VREL(I)                                       IMPULS  1560
   52 TDV   = TDV   + TTI(I)*DV  (I)                                       IMPULS  1570
   53 ALPHA = 0.0                                                          IMPULS  1580
C                                                                          IMPULS  1590
C     NOTE: CREST IS SUPPLIED AS (1+E)/2 WHERE E IS THE CLASSICAL          IMPULS  1600
C     COEFFICIENT OF RESTITUTION BUT WITH A RANGE OF -1 TO +1.             IMPULS  1610
C     CREST HAS A RANGE OF 0 TO +1 WHERE 0 (E=-1) REPRESENTS NO IMPULSE.   IMPULS  1620
C                                                                          IMPULS  1630
      IF (TDV.NE.0.0) ALPHA = -2.0*CREST*TVREL/TDV                         IMPULS  1640
      IF (NPRT(10).NE.0) WRITE (6,42) DWR1,DWR2,DWR3,DWR4,                 IMPULS  1650
     *                                TTI,VREL,DV,                         IMPULS  1660
     *                                TVREL,TDV,CREST,ALPHA                IMPULS  1670
      DO 60 J=1,NGRND                                                      IMPULS  1680
      DO 60 I=1,3                                                          IMPULS  1690
      SEGLV(I,J) = SEGLV(I,J) + ALPHA*SEGLA(I,J)                           IMPULS  1700
   60 WMEG (I,J) = WMEG (I,J) + ALPHA*WMEGD(I,J)                           IMPULS  1710
      IF  (NPRT(10).NE.0)  CALL OUTPUT(1)                                  IMPULS  1720
      IF  (NPRT( 3).NE.0)  CALL PRINT(6HIMPULS)                            IMPULS  1730
      CALL ELTIME(2,27)                                                    IMPULS  1740
   99 RETURN                                                               IMPULS  1750
      END                                                                  IMPULS  1760
```

```
      SUBROUTINE INITAL                                                 INITAL  0010
C                                                        REV 20 01/22/80  INITAL  0020
C     PERFORMS CARD INPUT AND COMPUTATIONS FOR INITIAL                  INITAL  0030
C     POSITIONING OF THE CRASH VICTIM'S BODY SEGMENTS.                  INITAL  0040
C                                                                       INITAL  0050
      IMPLICIT REAL*8(A-H,O-Z)                                          INITAL  0060
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           INITAL  0070
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)        INITAL  0080
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30), INITAL  0090
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       INITAL  0100
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(3,60),HA(3,60),HB(3,60), INITAL  0110
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),     INITAL  0120
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)    INITAL  0130
      COMMON/VPOSTN/ ZPLT(3),SPLT(3),AXV(3,6),VATAB(6,101,6),           INITAL  0140
     *               VT0(6),VDT(6),TIMEV(6),OMEGV(6),NVTAB(6),INDXV(6)  INITAL  0150
      COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),           INITAL  0160
     *               BLTTTL(5,8),PLTTL(5,30),BAGTTL(5,6),SEG(30),       INITAL  0170
     *               JOINT(30),CGS(30),JS(30)                           INITAL  0180
      REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT      INITAL  0190
      LOGICAL*1 CGS,JS                                                  INITAL  0200
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         INITAL  0210
     *               UNITL,UNITM,UNITT,GRAVTY(3)                        INITAL  0220
      COMMON/TEMPVS/ TMP(18),WMGDEG(3,30),T(3),S(3),A(3,2),Z(3,3)       INITAL  0230
C                NOTE : CHAIN ALSO USES TEMPVS.                         INITAL  0240
      DIMENSION  YPR(3,30) , IYPR(4,30)                                 INITAL  0250
C                                                                       INITAL  0260
C     INPUT CARD G.1.A (PLOT COORDINATES OF VEHICLE REFERENCE ORIGIN)   INITAL  0270
C                                                                       INITAL  0280
      READ(5,22) ZPLT,I1,J1,I2,J2,I3                                    INITAL  0290
   22 FORMAT(3F10.0,5I4)                                                INITAL  0300
      S(1) = 10.0                                                       INITAL  0310
      S(2) =  6.0                                                       INITAL  0320
      S(3) =  1.0                                                       INITAL  0330
C                                                                       INITAL  0340
C     IF J1#0, INPUT CARD G.1.B (PLOT SCALING INPUT)                    INITAL  0350
C                                                                       INITAL  0360
      IF  (J1.NE.0)  READ (5,22) S                                      INITAL  0370
      SPLT(1) = 1.0/S(3)                                                INITAL  0380
      SPLT(2) = 1.0/S(3)                                                INITAL  0390
      SPLT(3) = -(S(1)/S(2))/S(3)                                       INITAL  0400
      WRITE  (6,23)  ZPLT,I1,J1,I2,J2,I3,S                              INITAL  0410
   23 FORMAT('1 SUBROUTINE INITAL INPUT',85X,'CARD G.1'//               INITAL  0420
     *   '     ZPLT(X)   ZPLT(Y)   ZPLT(Z)   I1    J1    I2    J2    I3',  INITAL  0430
     *   '     SPLT(1)   SPLT(2)   SPLT(3)'/3F10.0,5I6,3F10.2)          INITAL  0440
C                                                                       INITAL  0450
C     INPUT CARDS G.2.A - G.2.N                                         INITAL  0460
C                                                                       INITAL  0470
C     INITIAL LINEAR POSITION (IN) AND (IF I3=1) VELOCITY (IN/SEC)      INITAL  0480
C     OF EACH BASE BODY SEGMENT. IF I3=0, VELOCITY WILL BE SET TO       INITAL  0490
C     INITIAL VELOCITY OF VEHICLE.. INPUTS IN INERTIAL REFERENCE.       INITAL  0500
```

```
C                                                                       INITAL  0510
      DO 37 J=1,NSEG                                                    INITAL  0520
      IF(J.GT.1.AND.IABS(JNT(J-1)).GT.0) GO TO 37                       INITAL  0530
      READ(5,24) (SEGLP(I,J),I=1,3),(SEGLV(I,J),I=1,3)                 INITAL  0540
   24 FORMAT (6F10.0 , 4I3)                                             INITAL  0550
      IF(I3.GT.0) GO TO 37                                              INITAL  0560
      DO 36 I=1,3                                                       INITAL  0570
   36 SEGLV(I,J) = SEGLV(I,NVEH)                                        INITAL  0580
   37 CONTINUE                                                          INITAL  0590
C                                                                       INITAL  0600
C     INPUT CARDS G.3.A - G.3.N                                         INITAL  0610
C                                                                       INITAL  0620
C     FOR EACH BODY SEGMENT SUPPLY YAW, PITCH AND ROLL (DEGREES)        INITAL  0630
C     AND (IF I3=1) THE ANGULAR VELOCITY IN LOCAL REFERENCE (DEG/SEC).  INITAL  0640
C     IF I3=0, THE ANGULAR VELOCITY (BLANK ON INPUT CARDS) WILL BE SET  INITAL  0650
C     EQUAL TO THE INITIAL ANGULAR VELOCITY OF THE VEHICLE.             INITAL  0660
C                                                                       INITAL  0670
      FIRST = 0.0                                                       INITAL  0680
      DO 40 J=1,NSEG                                                    INITAL  0690
      READ (5,24) (YPR(I,J),I=1,3),(WMGDEG(I,J),I=1,3),(IYPR(I,J),I=1,4) INITAL  0700
      ID1 = IYPR(1,J)                                                   INITAL  0710
      DO 38 I=1,3                                                       INITAL  0720
      IF (ID1.EQ.0) IYPR(I,J) = I                                       INITAL  0730
   38 WMEG(I,J) = WMGDEG(I,J)*RADIAN                                    INITAL  0740
      IF  (ID1.GE.0)  GO TO 60                                          INITAL  0750
C                                                                       INITAL  0760
C     READ CARD G.3.J2 FOR SEGMENT NO. J WHEN IYPR(1,J) IS NEGATIVE.    INITAL  0770
C                                                                       INITAL  0780
      READ  (5,24)  A,II,IK,JJ,JK                                       INITAL  0790
      IJ = II                                                           INITAL  0800
      LK = IK                                                           INITAL  0810
      DO  54  K=1,2                                                     INITAL  0820
      IF  (IJ.GT.0)  GO TO 52                                           INITAL  0830
      DO  51  I=1,3                                                     INITAL  0840
   51 Z(I,LK) = A(I,K)                                                  INITAL  0850
      GO TO 53                                                          INITAL  0860
   52 DA1 = A(1,K)*RADIAN                                               INITAL  0870
      DA2 = A(2,K)*RADIAN                                               INITAL  0880
      SA1 = DSIN(DA1)                                                   INITAL  0890
      SA2 = DSIN(DA2)                                                   INITAL  0900
      CA1 = DCOS(DA1)                                                   INITAL  0910
      CA2 = DCOS(DA2)                                                   INITAL  0920
      IJ1 = IJ+1                                                        INITAL  0930
      IJ2 = IJ+2                                                        INITAL  0940
      IF  (IJ1.GT.3)  IJ1= IJ1-3                                        INITAL  0950
      IF  (IJ2.GT.3)  IJ2= IJ2-3                                        INITAL  0960
      SGN = 1.0                                                         INITAL  0970
      IF  (SA1.LT.0.0 .AND. CA2.LT.0.0)  SGN = -1.0                     INITAL  0980
      Z(IJ ,LK) = SGN*SA1*CA2                                           INITAL  0990
      Z(IJ1,LK) = SGN*SA1*SA2                                           INITAL  1000
```

```
      Z(IJ2,LK) = SGN*CA1*CA2                                                    INITAL  1010
   53 IJ = JJ                                                                    INITAL  1020
   54 LK = JK                                                                    INITAL  1030
      ZDOTIJ = Z(1,IK)*Z(1,JK) + Z(2,IK)*Z(2,JK) + Z(3,IK)*Z(3,JK)              INITAL  1040
      ZDOTII = Z(1,IK)*Z(1,IK) + Z(2,IK)*Z(2,IK) + Z(3,IK)*Z(3,IK)              INITAL  1050
      RATIO = ZDOTIJ/ZDOTII                                                     INITAL  1060
      DO  55  I=1,3                                                             INITAL  1070
   55 Z(I,JK) = Z(I,JK) - RATIO*Z(I,IK)                                         INITAL  1080
      LK = 6-IK-JK                                                              INITAL  1090
      IT = MOD(JK-IK+3,3)                                                       INITAL  1100
      IF  (IT.EQ.1)  CALL CROSS(Z(1,IK),Z(1,JK),Z(1,LK))                        INITAL  1110
      IF  (IT.EQ.2)  CALL CROSS(Z(1,JK),Z(1,IK),Z(1,LK))                        INITAL  1120
      DO  57  K=1,3                                                             INITAL  1130
      IYPR(K,J) = 4-K                                                           INITAL  1140
      SUM = 0.0                                                                 INITAL  1150
      DO  56  I=1,3                                                             INITAL  1160
   56 SUM = SUM + Z(I,K)**2                                                     INITAL  1170
      SQUM = DSQRT(SUM)                                                         INITAL  1180
      DO  57  I=1,3                                                             INITAL  1190
   57 D(K,I,J) = Z(I,K)/SQUM                                                    INITAL  1200
      CALL YPRDEG (D(1,1,J),YPR(1,J))                                           INITAL  1210
      IF  (FIRST.EQ.0.0)  WRITE (6,58)                                          INITAL  1220
   58 FORMAT('0 INITIAL ANGULAR ROTATIONS COMPUTED FROM CARDS G.3.J2'//          INITAL  1230
     *       '  SEGMENT',10X,'SEGMENT PRIMARY AXIS',                            INITAL  1240
     *       12X,'SEGMENT SECONDARY AXIS',30X,'ANGULAR ROTATIONS (DEG)'/       INITAL  1250
     *       '  NO. SEG',9X,'A1',8X,'A2',8X,'A3',11X,'B1',8X,'B2',8X,          INITAL  1260
     *       'B3',7X,'II  IK  JJ  JK',9X,'YAW',6X,'PITCH',5X,'ROLL'/)          INITAL  1270
      FIRST = 1.0                                                               INITAL  1280
      WRITE (6,59) J,SEG(J),A,II,IK,JJ,JK,(YPR(I,J),I=1,3)                      INITAL  1290
   59 FORMAT (I4,1X,A4,3X,3F10.3,3X,3F10.3,3X,4I4,3X,3F10.3)                    INITAL  1300
   60 M = IYPR(4,J)                                                             INITAL  1310
      IF (M.EQ.0) M=NGRND                                                       INITAL  1320
      IF  (M.GE.J .AND. M.LE.NSEG)  STOP 24                                     INITAL  1330
      IF  (M.LT.0 .AND. -M.NE.IABS(JNT(J-1)))  STOP 25                          INITAL  1340
      CALL DRCIJK (D,YPR,IYPR,HT,J)                                             INITAL  1350
      IF (I3.GT.0) GO TO 40                                                     INITAL  1360
      CALL DOT31(D(1,1,NVEH),WMEG(1,NVEH),T)                                    INITAL  1370
      CALL MAT31(D(1,1,J),T,WMEG(1,J))                                          INITAL  1380
      DO 39 I=1,3                                                               INITAL  1390
   39 WMGDEG(I,J) = WMEG(I,J)/RADIAN                                            INITAL  1400
   40 CONTINUE                                                                  INITAL  1410
      CALL VEHPOS                                                               INITAL  1420
      CALL CHAIN                                                                INITAL  1430
C                                                                               INITAL  1440
C     OUTPUT INITIAL BODY SEGMENT POSITIONS.                                    INITAL  1450
C                                                                               INITAL  1460
      WRITE (6,42) UNITL,UNITL,UNITT                                            INITAL  1470
   42 FORMAT('0 INITIAL POSITIONS (INERTIAL REFERENCE)',70X,'CARDS G.2'/         INITAL  1480
     *      /'  SEGMENT',11X,'LINEAR POSITION (',A4,')',                       INITAL  1490
     *       14X,'LINEAR VELOCITY ('A4,'/',A4,')'/                            INITAL  1500
```

```
  *       '  NO. SEG',2(9X,'X',11X,'Y',11X,'Z',5X) )                          INITAL  1510
   WRITE (6,43) (J,SEG(J),(SEGLP(I,J),I=1,3),(SEGLV(I,J),I=1,3)               INITAL  1520
  *             ,J=1,NSEG)                                                    INITAL  1530
43 FORMAT(I4,1X,A4,3X,3F12.5,3X,3F12.5)                                       INITAL  1540
   WRITE (6,44) UNITT                                                         INITAL  1550
44 FORMAT('0 INITIAL ANGULAR ROTATION AND VELOCITY',71X,'CARDS G.3'//         INITAL  1560
  *       '  SEGMENT',11X,'ANGULAR ROTATION (DEG)'                           INITAL  1570
  *       14X,'ANGULAR VELOCITY (DEG/',A4,')'/                               INITAL  1580
  *       '  NO. SEG',8X,'YAW',8X,'PITCH',7X,'ROLL',                         INITAL  1590
  *       13X,'X',11X,'Y',11X,'Z',15X,'IYPR' )                               INITAL  1600
   WRITE (6,46) (J,SEG(J),(YPR(I,J),I=1,3),(WMGDEG(I,J),I=1,3),               INITAL  1610
  *             (IYPR(I,J),I=1,4),J=1,NSEG)                                   INITAL  1620
46 FORMAT(I4,1X,A4,3X,3F12.5,3X,3F12.5,3X,4I4)                                INITAL  1630
   IF (I3.EQ.0) WRITE (6,45)                                                  INITAL  1640
45 FORMAT('0 LINEAR AND ANGULAR VELOCITIES HAVE BEEN SET EQUAL TO THE        INITAL  1650
  * INITIAL VEHICLE VELOCITIES.')                                             INITAL  1660
   IF  (NHRNSS.NE.0)  CALL HBPLAY                                             INITAL  1670
   IF  (I1.EQ.15)  CALL EQUILB (YPR,IYPR)                                     INITAL  1680
   CALL ROTATE                                                                INITAL  1690
   CALL ELTIME(2,2)                                                           INITAL  1700
   RETURN                                                                     INITAL  1710
   END                                                                        INITAL  1720
```

```
      SUBROUTINE INTERS(A,B,XM,T,X,V,AX)                                INTERS  0010
C                                                        REV 19 08/05/78  INTERS  0020
C     DETERMINES INTERSECTION OF ELLIPSOIDS                              INTERS  0030
C         X'AX = 1                                                       INTERS  0040
C         (X'-M')B(X-M) = 1                                              INTERS  0050
C     WHERE A AND B ARE ELLIPSOID MATRICES                               INTERS  0060
C     IF T ENTERS AS +1.0 , A IS EXTERNAL TO B AND                       INTERS  0070
C                 AS -1.0 , A IS INTERNAL TO B.                          INTERS  0080
C                                                                        INTERS  0090
C     IF V ENTERS AS NON-ZERO, WILL USE PREVIOUS VALUE FOR START.        INTERS  0100
C     (AX) RETURNS AS (A)*(X).                                           INTERS  0110
C                                                                        INTERS  0120
C     RETURNS T>1 - NO INTERSECTION                                      INTERS  0130
C             T<1 - INTERSECTION IN WHICH CASE X WILL                    INTERS  0140
C                   CONTAIN COORDINATES OF CONTACT OF                    INTERS  0150
C                   CONTRACTED ELLIPSOIDS.                               INTERS  0160
C                                                                        INTERS  0170
      IMPLICIT REAL*8 (A-H,O-Z)                                          INTERS  0180
      DIMENSION A(3,3),B(3,3),XM(3),X(3)                                 INTERS  0190
      DIMENSION C(3,4),Z(3),BM(3),AX(3),AM(3)                            INTERS  0200
      EQUIVALENCE (Z(1),C(1,4))                                          INTERS  0210
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                          INTERS  0220
     *               UNITL,UNITM,UNITT,GRAVTY(3)                         INTERS  0230
C                                  INITIALIZATION                        INTERS  0240
C                                      EVALUATE BM,M'AM,M'BM             INTERS  0250
C                                      SET N=0, V=M'BM/M'AM              INTERS  0260
      N = 0                                                              INTERS  0270
      BMM = 0.0                                                          INTERS  0280
      AMM = 0.0                                                          INTERS  0290
      DO 11 I=1,3                                                        INTERS  0300
      BM(I) = 0.0                                                        INTERS  0310
      AM(I) = 0.0                                                        INTERS  0320
      DO 10 J=1,3                                                        INTERS  0330
      IF (DABS(A(I,J)).LT.EPS(20)) A(I,J) = 0.0                          INTERS  0340
      AM(I) = AM(I) + A(I,J)*XM(J)                                       INTERS  0350
      IF (DABS(B(I,J)).LT.EPS(20)) B(I,J) =0.0                           INTERS  0360
   10 BM(I) = BM(I) + B(I,J)*XM(J)                                       INTERS  0370
      BMM = BMM + XM(I)*BM(I)                                            INTERS  0380
   11 AMM = AMM + XM(I)*AM(I)                                            INTERS  0390
      IF (V.EQ.0.0) V=T*DSQRT(BMM/AMM)                                   INTERS  0400
      IDONE = 0                                                          INTERS  0410
C                                  NEWTON-RAPHSON ITERATION FOR          INTERS  0420
C                                  G(V) = FA(V)-FB(V) = 0                INTERS  0430
C                                      SOLVE (VA+B)X = BM  FOR X         INTERS  0440
      ITER = 0                                                           INTERS  0450
   20 ITER = ITER+1                                                      INTERS  0460
      DO 22 I=1,3                                                        INTERS  0470
      DO 21 J=1,3                                                        INTERS  0480
   21 C(I,J) = V*A(I,J) + B(I,J)                                         INTERS  0490
   22 Z(I)   = BM(I)                                                     INTERS  0500
```

```
      CALL DSMSOL(C,3,3)                                                INTERS  0510
C                                       EVALUATE AX                     INTERS  0520
C                                                FA(V) = X'AX           INTERS  0530
C                                                FB(V) = -V(X'-M')AX    INTERS  0540
      FA = 0.0                                                          INTERS  0550
      FB = 0.0                                                          INTERS  0560
      CALL MAT31(A,Z,AX)                                                INTERS  0570
      DO 30 I=1,3                                                       INTERS  0580
      X(I) = Z(I)                                                       INTERS  0590
      FA = FA+X(I)*AX(I)                                                INTERS  0600
   30 FB = FB+(X(I)-XM(I))*AX(I)                                        INTERS  0610
      FB = -V*FB                                                        INTERS  0620
      IF (T.LT.0.0) FA = 1.0/FA                                         INTERS  0630
      IF (IDONE.EQ.1) GO TO 60                                          INTERS  0640
C                                       TEST FOR INTERSECTION           INTERS  0650
      IF (FA-FB) 32,60,31                                               INTERS  0660
C                                       IF FA>FB>1, NO INTERSECTION     INTERS  0670
   31 IF (T.GT.0.0.AND.FB.LT.1.0) GO TO 40                              INTERS  0680
      IF (T.LT.0.0.AND.FA.GT.1.0) GO TO 40                              INTERS  0690
      IF (N.EQ.0) GO TO 60                                              INTERS  0700
      GO TO 62                                                          INTERS  0710
C                                       IF FA<FB<1 , INTERSECTION       INTERS  0720
   32 IF (T.GT.0.0.AND.FB.LE.1.0) N=1                                   INTERS  0730
      IF (T.LT.0.0.AND.FA.GE.1.0) N=1                                   INTERS  0740
C                                       SOLVE (VA+B)Z = AX  FOR Z       INTERS  0750
   40 DO 42 I=1,3                                                       INTERS  0760
      DO 41 J=1,3                                                       INTERS  0770
   41 C(I,J) = V*A(I,J) + B(I,J)                                        INTERS  0780
   42 Z(I)   = AX(I)                                                    INTERS  0790
      CALL DSMSOL(C,3,3)                                                INTERS  0800
C                                       F'A(V) = -2X'AZ                 INTERS  0810
      CALL MAT31 (A,Z,AX)                                               INTERS  0820
      FPA = X(1)*AX(1)                                                  INTERS  0830
     *    + X(2)*AX(2)                                                 INTERS  0840
     *    + X(3)*AX(3)                                                 INTERS  0850
      FPA = -(FPA+FPA)                                                  INTERS  0860
C                                       DV = -G(V)/G'(V)                INTERS  0870
      DV = 1.0 + V                                                      INTERS  0880
      IF (T.LT.0.0) DV = V-FA**2                                        INTERS  0890
      DV = (FB-FA)/(DV*FPA)                                             INTERS  0900
      IF (ITER.GE.50) GO TO 62                                          INTERS  0910
C                                       TEST FOR CONVERGENCE            INTERS  0920
      IF (T*(V+DV).LE.0.0) DV = -0.5*V                                  INTERS  0930
      V   = V+DV                                                        INTERS  0940
      DV = DABS(DV/V)                                                   INTERS  0950
      IF (DV.LE.EPS(12)) IDONE=1                                        INTERS  0960
      GO TO 20                                                          INTERS  0970
C                                     FA(V) = FB(B), RETURN             INTERS  0980
   60 IF (T.LT.0.0) FA = 1.0/FB                                         INTERS  0990
      T = DSQRT(FA)                                                     INTERS  1000
      IF (FA.GT.1.0) GO TO 61                                           INTERS  1010
      N = 1                                                             INTERS  1020
      GO TO 71                                                          INTERS  1030
   61 IF (N.EQ.0) GO TO 71                                              INTERS  1040
   62 WRITE (6,63)                                                      INTERS  1050
   63 FORMAT(' INTERS ITERATION DID NOT CONVERGE')                      INTERS  1060
   71 CONTINUE                                                          INTERS  1070
      RETURN                                                            INTERS  1080
      END                                                               INTERS  1090
```

```
      SUBROUTINE KINPUT                                                 KINPUT  0010
C                                                          REV 19 09/18/79  KINPUT  0020
C     PERFORMS THE FOLLOWING CARD INPUT AFTER CARDS E.1-E.4 (SUBROUTINE    KINPUT  0030
C     CINPUT) AND BEFORE CARDS F.1-F.5 (SUBROUTINE FINPUT).               KINPUT  0040
C         CARD E.5 - NWINDF: NO. OF WIND FORCE FUNCTIONS ON CARDS E.6     KINPUT  0050
C                  - NJNTF : NO. OF JOINT FORCE FUNCTIONS ON CARDS E.7    KINPUT  0060
C         CARDS E.6 - DEFINITIONS OF WIND FORCE FUNCTIONS                 KINPUT  0070
C         CARDS E.7 - DEFINITIONS OF JOINT RESTORING FORCE FUNCTIONS      KINPUT  0080
C                                                                         KINPUT  0090
      IMPLICIT REAL*8(A-H,O-Z)                                          KINPUT  0100
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           KINPUT  0110
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)        KINPUT  0120
      COMMON/TABLES/ MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(500),TAB(2600) KINPUT  0130
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         KINPUT  0140
     *               UNITL,UNITM,UNITT,GRAVTY(3)                        KINPUT  0150
      COMMON/TEMPVS/ JTITLE(5,51),NF(5),MS(3),KTITLE(31),TH(50)         KINPUT  0160
C     NOTE: TEMPVS IS SHARED HERE WITH SUBROUTINES CINPUT AND FINPUT.   KINPUT  0170
      REAL BLANK,JTITLE,KTITLE                                          KINPUT  0180
      DATA BLANK/4H    /                                                KINPUT  0190
   11 FORMAT(2I6)                                                       KINPUT  0200
      J1 = MXTB1+1                                                      KINPUT  0210
      IF (NWINDF.LE.0) GO TO 31                                         KINPUT  0220
      DO 30 K=1,NWINDF                                                  KINPUT  0230
C                                                                       KINPUT  0240
C     INPUT CARD E.6.A - FUNCTION NO. AND TITLE                         KINPUT  0250
C                                                                       KINPUT  0260
      READ (5,12) I,(KTITLE(J),J=1,5)                                   KINPUT  0270
   12 FORMAT(I4,4X,5A4)                                                 KINPUT  0280
      WRITE (6,13) I,(KTITLE(J),J=1,5),I,J1                             KINPUT  0290
   13 FORMAT('1 WIND FORCE FUNCTION NO.',I4,4X,5A4,10X,'NTI(',I2,') =', KINPUT  0300
     *       I5,43X,'CARDS E.6'//)                                      KINPUT  0310
      IF (I.LE.0.OR.I.GT.50) WRITE (6,14)                               KINPUT  0320
   14 FORMAT('0 IMPROPER FUNCTION NO. PROGRAM TERMINATED.')             KINPUT  0330
      IF (I.LE.0.OR.I.GT.50) STOP 11                                    KINPUT  0340
      IF (NTI(I).NE.0) WRITE (6,15) I                                   KINPUT  0350
   15 FORMAT('0 FUNCTION NO.',I4,' HAS ALREADY BEEN INPUTTED AND WILL BE KINPUT  0360
     * REPLACED BY THIS FUNCTION.')                                     KINPUT  0370
      NTI(I) = J1                                                       KINPUT  0380
      DO 16 J=1,5                                                       KINPUT  0390
   16 JTITLE(J,I) = KTITLE(J)                                           KINPUT  0400
      J2 = J1+4                                                         KINPUT  0410
C                                                                       KINPUT  0420
C     INPUT CARD E.6.B - DO THRU D4 (FOR NOW A BLANK CARD)              KINPUT  0430
C                                                                       KINPUT  0440
      READ  (5,17) (TAB(J),J=J1,J2)                                     KINPUT  0450
      WRITE (6,18) (TAB(J),J=J1,J2)                                     KINPUT  0460
   17 FORMAT(6F12.0)                                                    KINPUT  0470
   18 FORMAT(10X,'D0',13X,'D1',13X,'D2',13X,'D3',13X,'D4'/5F15.4//)     KINPUT  0480
      J1 = J2+1                                                         KINPUT  0490
C                                                                       KINPUT  0500
```

```
C      INPUT CARD E.6.C - NTMPTS                                                KINPUT  0510
C                                                                               KINPUT  0520
       READ  (5,11) NTMPTS                                                      KINPUT  0530
       WRITE (6,19) NTMPTS                                                      KINPUT  0540
    19 FORMAT('0 WIND FORCE TABLES FOR ',I6,' TIME POINTS.'//                   KINPUT  0550
      *       11X,'T',14X,'FX(T)',15X,'FY(T)',15X,'FZ(T)'  /)                   KINPUT  0560
       TAB(J1) = NTMPTS                                                         KINPUT  0570
       J1 = J1+1                                                                KINPUT  0580
       J2 = J1+4*NTMPTS-1                                                       KINPUT  0590
C                                                                               KINPUT  0600
C      INPUT CARDS E.6.D-E.6.N - NTMPTS CARDS OF T,FX(T),FY(T),FZ(T)            KINPUT  0610
C                                                                               KINPUT  0620
       READ  (5,20) (TAB(J),J=J1,J2)                                            KINPUT  0630
       WRITE (6,21) (TAB(J),J=J1,J2)                                            KINPUT  0640
    20 FORMAT(4F12.0)                                                           KINPUT  0650
    21 FORMAT(3X,F12.6,3G20.6)                                                  KINPUT  0660
       J1 = J2+1                                                                KINPUT  0670
    30 CONTINUE                                                                 KINPUT  0680
    31 IF (NJNTF.LE.0) GO TO 51                                                 KINPUT  0690
       DO 50 K=1,NJNTF                                                          KINPUT  0700
C                                                                               KINPUT  0710
C      INPUT CARD E.7.A - FUNCTION NO. AND TITLE                                KINPUT  0720
C                                                                               KINPUT  0730
       READ  (5,12) I,(KTITLE(J),J=1,5)                                         KINPUT  0740
       WRITE (6,32) I,(KTITLE(J),J=1,5),I,J1                                    KINPUT  0750
    32 FORMAT('1 JOINT FORCE FUNCTION NO.',I4,4X,5A4,10X,'NTI(',I2,') =',       KINPUT  0760
      *        I5,42X,'CARDS E.7'//)                                            KINPUT  0770
       IF (I.LE.0.OR.I.GT.50) WRITE (6,14)                                      KINPUT  0780
       IF (I.LE.0.OR.I.GT.50) STOP 12                                           KINPUT  0790
       IF (NTI(I).NE.0) WRITE (6,15) I                                          KINPUT  0800
       NTI(I) = J1                                                              KINPUT  0810
       DO 33 J=1,5                                                              KINPUT  0820
    33 JTITLE(J,I) = KTITLE(J)                                                  KINPUT  0830
C                                                                               KINPUT  0840
C      INPUT CARD E.7.B - D0,D1,D2,D3,D4 (FOR NOW A BLANK CARD).                KINPUT  0850
C                                                                               KINPUT  0860
       J2 = J1+4                                                                KINPUT  0870
       READ  (5,17) (TAB(J),J=J1,J2)                                            KINPUT  0880
       WRITE (6,18) (TAB(J),J=J1,J2)                                            KINPUT  0890
       J1 = J2+1                                                                KINPUT  0900
C                                                                               KINPUT  0910
C      INPUT CARD E.7.C - NTHETA,NPHI                                           KINPUT  0920
C                                                                               KINPUT  0930
       READ  (5,11) NTHETA,NPHI                                                 KINPUT  0940
       TAB(J1  ) = NTHETA                                                       KINPUT  0950
       TAB(J1+1) = NPHI                                                         KINPUT  0960
       J1 = J1+2                                                                KINPUT  0970
       IF (NTHETA.LT.0) GO TO 38                                                KINPUT  0980
       DO 35 J=1,NTHETA                                                         KINPUT  0990
    35 TH(J) = DFLOAT(J-1)*180.0/DFLOAT(NTHETA-1)                               KINPUT  1000
```

```
      WRITE (6,36) NTHETA,NPHI,(TH(J),J=2,NTHETA)                          KINPUT  1010
   36 FORMAT('0 FUNCTION IS TABULAR FOR' ,I3,' X',I3,' VALUES OF THETA A   KINPUT  1020
     *ND PHI'//30X,'THETA'/5X,'PHI',5X,'THETA0',F16.3,4F20.3/              KINPUT  1030
     *     (15X,5F20.3))                                                  KINPUT  1040
   37 FORMAT(F9.2,F10.3,5G20.7/(19X,5G20.7))                              KINPUT  1050
      GO TO 40                                                            KINPUT  1060
   38 NPOLY = -NTHETA -1                                                  KINPUT  1070
      WRITE (6,39) NPOLY,NPHI,(BLANK,J,J=1,NPOLY)                         KINPUT  1080
   39 FORMAT('0 FUNCTION IS COEFFICIENTS OF' ,I3,' ORDER POLYNOMIALS IN   KINPUT  1090
     *(THETA-THETA0) FOR',I3,' VALUES OF PHI.'//                          KINPUT  1100
     *  27X,'COEFFICIENTS OF (THETA-THETA0)**N'/                          KINPUT  1110
     *  5X,'PHI',5X,'THETA0',7X,5(A4,'N =',I2,11X)/(26X,A4,'N =',I2,11X,  KINPUT  1120
     *  A4,'N =',I2,11X,A4,'N =',I2,11X,A4,'N =',I2,11X,A4,'N =',I2)  )   KINPUT  1130
   40 WRITE (6,21)                                                        KINPUT  1140
      DO 49 I=1,NPHI                                                      KINPUT  1150
      PHIDEG = DFLOAT(I-1)*360.0/DFLOAT(NPHI) - 180.0                     KINPUT  1160
C                                                                         KINPUT  1170
C     INPUT CARDS E.7.D - E.7.N NPHI SETS WITH NTHETA ITEMS PER SET.      KINPUT  1180
C        EACH SET I IS FOR PHI(I) = -180 +(I-1)*360/NPHI DEGREES AND      KINPUT  1190
C        ASSUMES DATA FOR PHI(NPHI+1) = 180 IS SAME AS PHI(1) = -180.     KINPUT  1200
C                                                                         KINPUT  1210
      J2 = J1 + IABS(NTHETA) -1                                           KINPUT  1220
      READ  (5,17) (TAB(J),J=J1,J2)                                       KINPUT  1230
      WRITE (6,37) PHIDEG,(TAB(J),J=J1,J2)                                KINPUT  1240
      IF (NTHETA.LT.0) TAB(J1) = TAB(J1)*RADIAN                           KINPUT  1250
      IF (NTHETA.LT.0) GO TO 49                                           KINPUT  1260
C                                                                         KINPUT  1270
C     FOR TABULAR DATA, FILL IN ZERO VALUES WITH INTERPOLATED NEGATIVE    KINPUT  1280
C     VALUES. OVERWRITE VALUE IN FIRST COLUMN (SUPPLIED AS THETA0) WITH   KINPUT  1290
C     VALUE FOR THETA = 0 AND ALL OTHER ZERO VALUES.                      KINPUT  1300
C                                                                         KINPUT  1310
      THETA0 = TAB(J1)                                                    KINPUT  1320
      IF (THETA0.EQ.0.0) GO TO 49                                         KINPUT  1330
      JJ = THETA0*DFLOAT(NTHETA-1)/180.0 + 1.0 + EPS(6)                   KINPUT  1340
      JJ1 = J1+JJ                                                         KINPUT  1350
      IERROR = 0                                                          KINPUT  1360
      IF (JJ1.GT.J2) IERROR = 1                                           KINPUT  1370
      IF (TAB(JJ1).LE.0.0) IERROR = 2                                     KINPUT  1380
      IF (IERROR.NE.0) GO TO 46                                           KINPUT  1390
      DO 45 J=1,JJ                                                        KINPUT  1400
      J1J = J1+J-1                                                        KINPUT  1410
      IF (J.NE.1.AND.TAB(J1J).GT.0.0) IERROR = 3                          KINPUT  1420
   45 TAB(J1J) = TAB(JJ1)*(TH(J)-THETA0)/(TH(JJ+1)-THETA0)                KINPUT  1430
   46 IF (IERROR.NE.0) WRITE (6,47) IERROR                                KINPUT  1440
   47 FORMAT('0 INPUT ERROR. INCONSISTENT VALUE OF THETA0. IERROR =',I2,  KINPUT  1450
     *       ' PROGRAM TERMINATED.')                                      KINPUT  1460
      IF (IERROR.NE.0) STOP 13                                            KINPUT  1470
   49 J1 = J2+1                                                           KINPUT  1480
   50 CONTINUE                                                            KINPUT  1490
   51 MXTB1 = J1-1                                                        KINPUT  1500
      RETURN                                                              KINPUT  1510
      END                                                                 KINPUT  1520
```

```
      SUBROUTINE LINAXS(X0,Y0,THETA,NINTVS,TOTLGT)                      LINAXS  0010
C                                                        REV 18 02/28/78  LINAXS  0020
CC    PURPOSE : PREPARE A LINEAR AXIS ON A PLOT.                        LINAXS  0030
CC                                                                      LINAXS  0040
CC    DESCRIPTION OF PARAMETERS:                                        LINAXS  0050
CC       X0,Y0 - STARTING POINT (IN INCHES, REL TO PLOTTER ORIGIN).     LINAXS  0060
CC                                                                      LINAXS  0070
CC       THETA - ANGLE OF AXIS, IN DEGREES.                             LINAXS  0080
CC                                                                      LINAXS  0090
CC       NINTVS- MAGNITUDE = NO. OF INTERVALS DELINEATED BY TIC MARKS.  LINAXS  0100
CC             - SIGN DETERMINES WHETHER TIC MARKS ARE PLACED ON        LINAXS  0110
CC               POSITIVE OR NEGATIVE SIDE OF AXIS, RESPECTIVELY        LINAXS  0120
CC               (POSITIVE SIDE IS TO LEFT OF DIRECTION OF TRAVEL).     LINAXS  0130
CC                                                                      LINAXS  0140
CC       TOTLGT- TOTAL LENGTH OF AXIS, IN INCHES.                       LINAXS  0150
CC                                                                      LINAXS  0160
CC    SUBROUTINES REQUIRED : SIN, COS, PLOT (NOTE: SINGLE PRECISION).   LINAXS  0170
CC                                                                      LINAXS  0180
CC    AUTHOR: W. D. FRYER, CALSPAN (MARCH 1967).                        LINAXS  0190
CC                                                                      LINAXS  0200
CC    PLAGIARIZED FROM CALSPAN SUBROUTINE LIBRARY (NO. CU 0035).        LINAXS  0210
C                                                                       LINAXS  0220
      THR = 1.7453293E-2 * THETA                                        LINAXS  0230
      SINT = SIN(THR)                                                   LINAXS  0240
      COST = COS(THR)                                                   LINAXS  0250
C                                                                       LINAXS  0260
      DL = ABS(TOTLGT/ FLOAT(NINTVS))                                   LINAXS  0270
      DX = DL*COST                                                      LINAXS  0280
      DY = DL*SINT                                                      LINAXS  0290
C                                                                       LINAXS  0300
      TICX = -0.12* SINT                                                LINAXS  0310
      TICY = 0.12* COST                                                 LINAXS  0320
      IF(NINTVS.GT.0) GO TO 30                                          LINAXS  0330
      TICX = -TICX                                                      LINAXS  0340
      TICY = -TICY                                                      LINAXS  0350
C                                                                       LINAXS  0360
   30 X = X0                                                            LINAXS  0370
      Y = Y0                                                            LINAXS  0380
C                                                                       LINAXS  0390
      CALL PLOT (X +TICX,Y+TICY,3)                                      LINAXS  0400
      CALL PLOT (X,Y,2)                                                 LINAXS  0410
      NINT = IABS(NINTVS)                                               LINAXS  0420
      DO 40 I=1,NINT                                                    LINAXS  0430
      X = X+DX                                                          LINAXS  0440
      Y = Y+DY                                                          LINAXS  0450
      CALL PLOT(X,Y,2)                                                  LINAXS  0460
      CALL PLOT(X+TICX,Y+TICY,2)                                        LINAXS  0470
   40 CALL PLOT(X,Y,2)                                                  LINAXS  0480
C                                                                       LINAXS  0490
      RETURN                                                            LINAXS  0500
C                                                                       LINAXS  0510
      END
```

```
      SUBROUTINE LOGAXS(X0,Y0,THETA,NDEC,EXTENT)                        LOGAXS  0010
C                                                    REV 19 09/18/79    LOGAXS  0020
CC    PURPOSE : PREPARE LOGARITHMIC AXIS ON A PLOT.                     LOGAXS  0030
C                                                                       LOGAXS  0040
CC    DESCRIPTION OF PARAMETERS:                                        LOGAXS  0050
C                                                                       LOGAXS  0060
CC       X0,Y0 - STARTING POINT (IN INCHES, REL TO PLOTTER ORIGIN).     LOGAXS  0070
C                                                                       LOGAXS  0080
CC       THETA - ANGLE OF AXIS (DEGREES).                               LOGAXS  0090
C                                                                       LOGAXS  0100
CC       NDECS - MAGNITUDE OF NDECS SPECIFIES NO. OF DECADES.           LOGAXS  0110
CC             - SIGN DETERMINES WHETHER TIC MARKS ARE TO BE PLACED     LOGAXS  0120
CC               ON POS. OR NEG. SIDE OF AXIS, RESP. (POS. SIDE IS      LOGAXS  0130
CC               TO LEFT OF PREDOMINANT DIRECTION OF TRAVEL).           LOGAXS  0140
C                                                                       LOGAXS  0150
CC       EXTENT- MAGNITUDE OF EXTENT SETS OVER-ALL LENGTH OF AXIS       LOGAXS  0160
CC               IN INCHES. IF EXTENT IS POSITIVE, TIC MARKS ARE        LOGAXS  0170
CC               SPACED NORMALLY (LARGE INTERVALS FIRST). IF EXTENT     LOGAXS  0180
CC               IS NEGATIVE, TIC MARKS ARE SPACED IN REVERSE ORDER     LOGAXS  0190
CC               (SMALL INTERVALS FIRST).                               LOGAXS  0200
C                                                                       LOGAXS  0210
CC    SUBROUTINES REQUIRED : SIN, COS, PLOT (NOTE: SINGLE PRECISION).   LOGAXS  0220
C                                                                       LOGAXS  0230
CC    AUTHOR: W. D. FRYER, CALSPAN (MARCH 1967).                        LOGAXS  0240
C                                                                       LOGAXS  0250
CC    PLAGIARIZED FROM CALSPAN SUBROUTINE LIBRARY (NO. CU 0036).        LOGAXS  0260
C                                                                       LOGAXS  0270
      LOGICAL REVERS                                                    LOGAXS  0280
      REAL XL(18),XL0(19)                                               LOGAXS  0290
      EQUIVALENCE (XL0(2),XL(1))                                        LOGAXS  0300
      DATA XL0/ 0.0    , 0.17609, 0.30103, 0.39794, 0.47712, 0.54407,   LOGAXS  0310
     * 0.60206, 0.65321, 0.69897, 0.74036, 0.77815, 0.81291, 0.84510,   LOGAXS  0320
     * 0.87506, 0.90309, 0.92942, 0.95424, 0.97772, 1.0    /            LOGAXS  0330
      DATA RPD /1.7453293E-2/                                           LOGAXS  0340
C                                                                       LOGAXS  0350
C                                                                       LOGAXS  0360
      REVERS = .FALSE.                                                  LOGAXS  0370
      IF(EXTENT.LT.0.0) REVERS = .TRUE.                                 LOGAXS  0380
C                                                                       LOGAXS  0390
      NODEC = IABS(NDEC)                                                LOGAXS  0400
      SPDEC = ABS(EXTENT) / FLOAT(NODEC)                                LOGAXS  0410
      THR = THETA*RPD                                                   LOGAXS  0420
      COST = COS(THR)                                                   LOGAXS  0430
      SINT = SIN(THR)                                                   LOGAXS  0440
C                                                                       LOGAXS  0450
   30 TICX1 =-0.05*SINT                                                 LOGAXS  0460
      TICY1 = 0.05*COST                                                 LOGAXS  0470
      TICXA = -0.12*SINT                                                LOGAXS  0480
      TICXB = -0.20*SINT                                                LOGAXS  0490
      TICYA = 0.12*COST                                                 LOGAXS  0500
```

```
      TICYB = 0.20*COST                                                 LOGAXS  0510
      IF(NDEC.GT.0) GO TO 50                                            LOGAXS  0520
C                                                                       LOGAXS  0530
   40 TICX1 = -TICX1                                                    LOGAXS  0540
      TICY1 = - TICY1                                                   LOGAXS  0550
      TICX2 = -TICX2                                                    LOGAXS  0560
      TICXA = - TICXA                                                   LOGAXS  0570
      TICYA = -TICYA                                                    LOGAXS  0580
      TICXB = -TICXB                                                    LOGAXS  0590
      TICYB = - TICYB                                                   LOGAXS  0600
C                                                                       LOGAXS  0610
   50 COST = COST*SPDEC                                                 LOGAXS  0620
      SINT = SINT* SPDEC                                                LOGAXS  0630
      TICX2 = TICXA                                                     LOGAXS  0640
      TICY2 = TICYA                                                     LOGAXS  0650
C                                                                       LOGAXS  0660
      XD = X0                                                           LOGAXS  0670
      YD = Y0                                                           LOGAXS  0680
      ND = 1                                                            LOGAXS  0690
      N = 0                                                             LOGAXS  0700
C                                                                       LOGAXS  0710
C*****GO TO START POS.*****                                             LOGAXS  0720
      CALL PLOT(X0+TICXB,Y0+TICYB,3)                                    LOGAXS  0730
      CALL PLOT(X0,Y0,2)                                                LOGAXS  0740
C                                                                       LOGAXS  0750
   60 N = N+1                                                           LOGAXS  0760
      Q = XL(N)                                                         LOGAXS  0770
      IF(.NOT. REVERS) GO TO 65                                         LOGAXS  0780
      M = 18-N                                                          LOGAXS  0790
      Q = 1.0-XL(M)                                                     LOGAXS  0800
   65 X = XD + Q*COST                                                   LOGAXS  0810
      Y = YD + Q*SINT                                                   LOGAXS  0820
      CALL PLOT(X,Y,2)                                                  LOGAXS  0830
      CALL PLOT(X+TICX1,Y+TICY1,2)                                      LOGAXS  0840
      CALL PLOT (X,Y,2 )                                                LOGAXS  0850
C                                                                       LOGAXS  0860
      N = N+1                                                           LOGAXS  0870
      Q = XL(N)                                                         LOGAXS  0880
      IF(.NOT. REVERS) GO TO 75                                         LOGAXS  0890
      M = 18-N                                                          LOGAXS  0900
      Q = 1.0 - XL(M)                                                   LOGAXS  0910
   75 X = XD + Q*COST                                                   LOGAXS  0920
      Y = YD + Q*SINT                                                   LOGAXS  0930
      CALL PLOT(X,Y,2)                                                  LOGAXS  0940
      CALL PLOT (X+TICX2,Y+TICY2,2)                                     LOGAXS  0950
      CALL PLOT(X,Y,2)                                                  LOGAXS  0960
C                                                                       LOGAXS  0970
      IF(N-16) 60,80,100                                                LOGAXS  0980
C                                                                       LOGAXS  0990
   80 TICX2 = TICXB                                                     LOGAXS  1000
      TICY2 = TICYB                                                     LOGAXS  1010
      GO TO 60                                                          LOGAXS  1020
C                                                                       LOGAXS  1030
  100 IF(ND .EQ. NODEC) GO TO 200                                       LOGAXS  1040
      TICX2 = TICXA                                                     LOGAXS  1050
      TICY2 = TICYA                                                     LOGAXS  1060
      N = 0                                                             LOGAXS  1070
      XD = X                                                            LOGAXS  1080
      YD = Y                                                            LOGAXS  1090
      ND = ND+1                                                         LOGAXS  1100
      GO TO 60                                                          LOGAXS  1110
C                                                                       LOGAXS  1120
  200 RETURN                                                            LOGAXS  1130
      END                                                               LOGAXS  1140
```

```
      FUNCTION LTIME(N)                                                 LTIME   0010
C                                                        REV 01 11/29/72  LTIME   0020
C     TEMPORARY FORTRAN VERSION OF S/370 ASSEMBLER LANGUAGE ROUTINE FROM  LTIME   0030
C     CALSPAN LIBRARY THAT MEASURES ELAPSED CPU TIME IN UNITS OF 0.01     LTIME   0040
C     SECONDS. IT SHOULD BE REPLACED WITH AN EQUIVALENT ROUTINE BY THE    LTIME   0050
C     USER TO ENABLE SUBROUTINE ELTIME TO PERFORM ON HIS COMPUTER.        LTIME   0060
C                                                                       LTIME   0070
C     ORIGINAL CALSPAN ROUTINE PERFORMS AS FOLLOWS:                     LTIME   0080
C         IT = LTIME(0) GIVES ELAPSED CPU TIME (INTEGER NUMBER OF 0.01  LTIME   0090
C                       SECOND UNITS) SINCE SUBROUTINE REFERENCE WAS    LTIME   0100
C                       RESET, AND RESETS THIS REFERENCE.               LTIME   0110
C         IT = LTIME(1) SAME, EXCEPT THAT THE REFERENCE IS NOT RESET.   LTIME   0120
C                                                                       LTIME   0130
      DATA KTIME/0/                                                     LTIME   0140
      KTIME = KTIME+1                                                   LTIME   0150
      LTIME = KTIME                                                     LTIME   0160
      IF (N.EQ.0) KTIME = 0                                             LTIME   0170
      RETURN                                                            LTIME   0180
      END                                                               LTIME   0190
```

```
      SUBROUTINE MAT31 (A,B,C)                                          MAT31   0010
C                                                    REV 17 01/03/77    MAT31   0020
C     PERFORMS MATRIX MULTIPLICATION C = AB                             MAT31   0030
C     WHERE A IS A 3X3 MATRIX, AND B AND C ARE VECTORS OF LENGTH 3.     MAT31   0040
C                                                                       MAT31   0050
      IMPLICIT REAL*8 (A-H,O-Z)                                         MAT31   0060
      DIMENSION  A(3,3) , B(3) , C(3)                                   MAT31   0070
      C(1) = A(1,1)*B(1) + A(1,2)*B(2) + A(1,3)*B(3)                    MAT31   0080
      C(2) = A(2,1)*B(1) + A(2,2)*B(2) + A(2,3)*B(3)                    MAT31   0090
      C(3) = A(3,1)*B(1) + A(3,2)*B(2) + A(3,3)*B(3)                    MAT31   0100
      RETURN                                                            MAT31   0110
      END                                                               MAT31   0120
```

```
      SUBROUTINE MAT33 (A,B,C)                                          MAT33   0010
C                                                    REV 17 01/03/77    MAT33   0020
C     PERFORMS MATRIX MULTIPLICATION C = AB                             MAT33   0030
C     WHERE A, B AND C ARE ALL 3X3 MATRICEES.                           MAT33   0040
C                                                                       MAT33   0050
      IMPLICIT REAL*8 (A-H,O-Z)                                         MAT33   0060
      DIMENSION  A(3,3) , B(3,3) , C(3,3)                               MAT33   0070
      DO  10  I=1,3                                                     MAT33   0080
      DO  10  J=1,3                                                     MAT33   0090
   10 C(I,J) = A(I,1)*B(1,J) + A(I,2)*B(2,J) + A(I,3)*B(3,J)            MAT33   0100
      RETURN                                                            MAT33   0110
      END                                                               MAT33   0120
```

```
      SUBROUTINE ORTHO(P,X,L)                                          ORTHO   0010
C                                                        REV 03 05/31/73 ORTHO   0020
C     GENERATES A SET OF RIGHT HANDED ORTHONORMAL VECTORS (P),          ORTHO   0030
C     GIVEN ONE OF THE VECTORS (X), WHERE                               ORTHO   0040
C          P - LX3 MATRIX OF 3 ORTHONORMAL VECTORS TO BE GENERATED.     ORTHO   0050
C          X - GIVEN VECTOR.                                           ORTHO   0060
C          L - 1ST SUBSCRIPT OF P IN CALLING PROGRAM.                  ORTHO   0070
C                                                                      ORTHO   0080
      IMPLICIT REAL*8(A-H,O-Z)                                         ORTHO   0090
      DIMENSION P(L,3),X(3)                                            ORTHO   0100
       M=2                                                             ORTHO   0110
       N=3                                                             ORTHO   0120
       TEST=0.                                                         ORTHO   0130
       DO 5 I=1,3                                                      ORTHO   0140
       P(I,3)=X(I)                                                     ORTHO   0150
       D=1.-X(I)**2                                                    ORTHO   0160
       IF(D.LE.TEST)GO TO 4                                            ORTHO   0170
       TEST=D                                                          ORTHO   0180
       D=DSQRT(D)                                                      ORTHO   0190
       P(I,1)=D                                                        ORTHO   0200
       P(I,2)=0.                                                       ORTHO   0210
       P(M,2)=X(N)/D                                                   ORTHO   0220
       P(N,2)=-X(M)/D                                                  ORTHO   0230
       P(M,1)=X(I)*P(N,2)                                              ORTHO   0240
       P(N,1)=-X(I)*P(M,2)                                             ORTHO   0250
    4  M=N                                                             ORTHO   0260
       N=I                                                             ORTHO   0270
    5  CONTINUE                                                        ORTHO   0280
       RETURN                                                          ORTHO   0290
       END                                                             ORTHO   0300
```

```
      SUBROUTINE OUTPUT(IJK)                                                OUTPUT  0010
C                                                          REV 20 05/18/80  OUTPUT  0020
C     CONTROLS TABULATED OUTPUT ON FORTRAN UNITS (STARTING WITH NO. 21)     OUTPUT  0030
C     OF SELECTED OPTIONAL SEGMENT LINEAR AND ANGULAR ACCELERATIONS,        OUTPUT  0040
C     VELOCITIES AND DISPLACEMENTS, JOINT PARAMETERS AND SELECTED DATA      OUTPUT  0050
C     FROM ALL ALLOWED CONTACT FORCE COMPUTATIONS BETWEEN BODY SEGMENTS     OUTPUT  0060
C     AND VEHICLE COMPONENTS.                                               OUTPUT  0070
C                                                                           OUTPUT  0080
      IMPLICIT REAL*8 (A-H,O-Z)                                             OUTPUT  0090
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,               OUTPUT  0100
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)            OUTPUT  0110
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),    OUTPUT  0120
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)           OUTPUT  0130
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(3,60),HA(3,60),HB(3,60),     OUTPUT  0140
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),         OUTPUT  0150
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)        OUTPUT  0160
      COMMON/JBARTZ/ MNPL(   30),MNBLT(   8),MNSEG(   30),MNBAG(    6),    OUTPUT  0170
     *               MPL(3,5,30),MBLT(3,5,8),MSEG(3,5,30),MBAG(3,10,6),     OUTPUT  0180
     *               NTPL( 5,30),NTBLT( 5,8),NTSEG( 5,30)                   OUTPUT  0190
      COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),               OUTPUT  0200
     *               BLTTTL(5,8),PLTTL(5,30),BAGTTL(5,6),SEG(30),           OUTPUT  0210
     *               JOINT(30),CGS(30),JS(30)                               OUTPUT  0220
      REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT          OUTPUT  0230
      LOGICAL*1 CGS,JS                                                      OUTPUT  0240
      COMMON/FORCES/ PSF(7,30),BSF(4,20),SSF(10,20),BAGSF(3,20),            OUTPUT  0250
     *               PRJNT(7,30),NPANEL(5),NPSF,NBSF,NSSF,NBGSF             OUTPUT  0260
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                             OUTPUT  0270
     *               UNITL,UNITM,UNITT,GRAVTY(3)                            OUTPUT  0280
      COMMON/RSAVE/  XSG(3,20,3),DPMI(3,3,30),LPMI(30),NSG(7),MSG(20,7)     OUTPUT  0290
      COMMON/COMAIN/VAR(240),DER(240),DT,H0,HMAX,HMIN,RSTIME,               OUTPUT  0300
     *              ISTEP,NSTEPS,NDINT,NEQ,IRSIN,IRSOUT                     OUTPUT  0310
      COMMON/DAMPER/ APSDM(3,20),APSDN(3,20),ASD(5,20),MSDM(20),MSDN(20)    OUTPUT  0320
      COMMON/HRNESS/ BAR(15,100),BB(100),BBDOT(100),PLOSS(2,100),           OUTPUT  0330
     *               XLONG(20),HTIME(2),IBAR(5,100),NL(2,100),              OUTPUT  0340
     *               NPTSPB(20),NPTPLY(20),NTHRNS(20),NBLTPH(5)             OUTPUT  0350
      COMMON/TEMPVS/ TDATA(14,50),ACC(7,20),T1(3),T2(3),T3(3),T4(9)         OUTPUT  0360
      LOGICAL  LTAPE8 , LTHIST                                              OUTPUT  0370
      DATA LINES/0/,LPP/45/                                                 OUTPUT  0380
C                                                                           OUTPUT  0390
      IF (IJK.NE.0) GO TO 13                                                OUTPUT  0400
C                                                                           OUTPUT  0410
C     SET ALL FORCE ARRAYS TO ZERO.                                         OUTPUT  0420
C                                                                           OUTPUT  0430
      DO  11  I=1,760                                                       OUTPUT  0440
   11 PSF(I,1) = 0.0                                                        OUTPUT  0450
      GO TO 66                                                              OUTPUT  0460
C                                                                           OUTPUT  0470
C     LTHIST = TRUE MEANS PRINT LINE OF TIME HISTORY DATA FOR THIS          OUTPUT  0480
C                         TIME POINT ON EACH OUTPUT UNIT (NT).              OUTPUT  0490
C                                                                           OUTPUT  0500
```

```
C     LTAPE8 = TRUE MEANS WRITE TIME HISTORY DATA ON TAPE 8.                    OUTPUT  0510
C                                                                               OUTPUT  0520
   13 NPRT4 = NPRT(4) + 4                                                       OUTPUT  0530
      IF  (NPRT4.LE.0 .OR. NPRT4.GT.8)  STOP 37                                 OUTPUT  0540
      GO TO  (66,66,66,16,15,14,14,15) , NPRT4                                  OUTPUT  0550
   14 LTHIST = .FALSE.                                                          OUTPUT  0560
      LTAPE8 = .TRUE.                                                           OUTPUT  0570
      GO TO 17                                                                  OUTPUT  0580
   15 LTHIST = .TRUE.                                                           OUTPUT  0590
      LTAPE8 = .TRUE.                                                           OUTPUT  0600
      GO TO 17                                                                  OUTPUT  0610
   16 LTHIST = .TRUE.                                                           OUTPUT  0620
      LTAPE8 = .FALSE.                                                          OUTPUT  0630
   17 CALL ELTIME (1,8)                                                         OUTPUT  0640
      IF  (LINES.NE.0)  GO TO 21                                                OUTPUT  0650
      PREVT = -999.0                                                            OUTPUT  0660
      IF  (IRSIN.NE.0)  GO TO 10                                                OUTPUT  0670
C                                                                               OUTPUT  0680
C     1ST TIME IN ROUTINE, READ CARD INPUT FOR OUTPUT CONTROL.                  OUTPUT  0690
C                                                                               OUTPUT  0700
C      1. NO. OF SEGMENT LINEAR  ACCELERATIONS,SEGMENT NOS. AND LOCATION        OUTPUT  0710
C      2. NO. OF SEGMENT LINEAR  VELOCITIES   ,SEGMENT NOS. AND LOCATION        OUTPUT  0720
C      3. NO. OF SEGMENT LINEAR  DISPLACEMENTS,SEGMENT NOS. AND LOCATION        OUTPUT  0730
C      4. NO. OF SEGMENT ANGULAR ACCELERATIONS AND SEGMENT NOS.                 OUTPUT  0740
C      5. NO. OF SEGMENT ANGULAR VELOCITIES    AND SEGMENT NOS.                 OUTPUT  0750
C      6. NO. OF SEGMENT ANGULAR DISPLACEMENTS AND SEGMENT NOS.                 OUTPUT  0760
C      7. NO. OF JOINT PARAMETERS AND JOINT NOS.                                OUTPUT  0770
C                                                                               OUTPUT  0780
      DO 20 K=1,7                                                               OUTPUT  0790
C                                                                               OUTPUT  0800
C     INPUT CARDS H.(K).(J) FOR K=1,3                                           OUTPUT  0810
C                                                                               OUTPUT  0820
      IF (K.LE.3) READ (5,18) KSG,(MSG(J,K),(XSG(I,J,K),I=1,3),J=1,KSG)         OUTPUT  0830
   18 FORMAT(2I6,3F12.6/(I12,3F12.6))                                           OUTPUT  0840
C                                                                               OUTPUT  0850
C     INPUT CARDS H.(K)     FOR K=4,7                                           OUTPUT  0860
C                                                                               OUTPUT  0870
      IF (K.GT.3) READ (5,19) KSG,(MSG(J,K),J=1,KSG)                            OUTPUT  0880
   19 FORMAT(12I6/(I12,10I6))                                                   OUTPUT  0890
      IF (K.NE.7 .OR. KSG.EQ.0) GO TO 20                                        OUTPUT  0900
      DO 12 J=1,KSG                                                             OUTPUT  0910
      L = MSG(J,K)                                                              OUTPUT  0920
      IF (IABS(IPIN(L)).EQ.4) MSG(J,K) = -L                                     OUTPUT  0930
   12 CONTINUE                                                                  OUTPUT  0940
   20 NSG(K) = KSG                                                              OUTPUT  0950
   10 IF  (.NOT.LTAPE8)  GO TO 21                                               OUTPUT  0960
      WRITE  (8)         NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,NPANEL,            OUTPUT  0970
     *                   MNPL,MNBLT,MNSEG,MNBAG,MPL,MBLT,MSEG,MBAG             OUTPUT  0980
      WRITE  (8)         DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,        OUTPUT  0990
     *                   SEG,JOINT,UNITL,UNITM,UNITT,NSG,MSG,XSG,              OUTPUT  1000
```

```
     *                    NHRNSS,NBLTPH,NPTSPB,NSD,MSDM,MSDN                  OUTPUT  1010
   21 LINES = LINES + 1                                                      OUTPUT  1020
      IF  (MOD(LINES,LPP).EQ.1 .AND. LTHIST)  CALL HEDING (LINES,LPP)        OUTPUT  1030
      NT = 20                                                               OUTPUT  1040
      USEC = 1000.0*TIME                                                    OUTPUT  1050
C                                                                           OUTPUT  1060
C     COMPUTE AND PRINT DATA FOR 7 TYPES OF OUTPUT ABOVE                    OUTPUT  1070
C                                                                           OUTPUT  1080
      DO 44 K=1,7                                                           OUTPUT  1090
      IF (NSG(K).LE.0) GO TO 44                                             OUTPUT  1100
      KSG = NSG(K)                                                          OUTPUT  1110
      J3  = 3                                                               OUTPUT  1120
      IF (K.EQ.7) J3 = 2                                                    OUTPUT  1130
      DO 43 J1=1,KSG,J3                                                     OUTPUT  1140
      J2 = MIN0(J1+J3-1,KSG)                                                OUTPUT  1150
      NT = NT+1                                                             OUTPUT  1160
      DO 38 J=J1,J2                                                         OUTPUT  1170
      L = IABS(MSG(J,K))                                                    OUTPUT  1180
      GO TO (22,24,26,29,31,34,35),K                                        OUTPUT  1190
C                                                                           OUTPUT  1200
C     1. SEGMENT LINEAR ACCELERATIONS IN LOCAL REFERENCE                    OUTPUT  1210
C                                                                           OUTPUT  1220
   22 CALL CROSS (WMEG(1,L),XSG(1,J,K),T1)                                  OUTPUT  1230
      CALL CROSS (WMEG(1,L),T1,T2)                                          OUTPUT  1240
      CALL CROSS (WMEGD(1,L),XSG(1,J,K),T3)                                 OUTPUT  1250
      CALL MAT31(D(1,1,L),SEGLA(1,L),T4)                                    OUTPUT  1260
      DO 23 I=1,3                                                           OUTPUT  1270
      ACC(I,J) = (T4(I)+T3(I)+T2(I))/G                                      OUTPUT  1280
   23 T1(I) = ACC(I,J)                                                      OUTPUT  1290
      IF  (LPMI(L).NE.0)  CALL DOT31 (DPMI(1,1,L),T1,ACC(1,J))              OUTPUT  1300
      GO TO 33                                                              OUTPUT  1310
C                                                                           OUTPUT  1320
C     2. SEGMENT LINEAR VELOCITIES IN VEHICLE REFERENCE                     OUTPUT  1330
C                                                                           OUTPUT  1340
   24 CALL CROSS (WMEG(1,L),XSG(1,J,K),T1)                                  OUTPUT  1350
      CALL DOT31(D(1,1,L),T1,T2)                                            OUTPUT  1360
      DO 25 I=1,3                                                           OUTPUT  1370
   25 T3(I) = T2(I) + SEGLV(I,L) - SEGLV(I,NVEH)                            OUTPUT  1380
      GO TO 28                                                              OUTPUT  1390
C                                                                           OUTPUT  1400
C     3. SEGMENT LINEAR DISPLACEMENTS IN VEHICLE REFERENCE                  OUTPUT  1410
C                                                                           OUTPUT  1420
   26 IF  (LPMI(L).EQ.0)  GO TO 76                                          OUTPUT  1430
      CALL DOT33 (DPMI(1,1,L),D(1,1,L),T4)                                  OUTPUT  1440
      CALL DOT31 (T4,XSG(1,J,K),T1)                                         OUTPUT  1450
      GO TO 77                                                              OUTPUT  1460
   76 CALL DOT31 (D(1,1,L),XSG(1,J,K),T1)                                   OUTPUT  1470
   77 DO 27 I=1,3                                                           OUTPUT  1480
   27 T3(I) = T1(I) + SEGLP(I,L) - SEGLP(I,NVEH)                            OUTPUT  1490
   28 CALL MAT31 (D(1,1,NVEH),T3,ACC(1,J))                                  OUTPUT  1500
```

```
      GO TO 33                                                                  OUTPUT  1510
C                                                                               OUTPUT  1520
C     4. SEGMENT ANGULAR ACCELERATIONS IN LOCAL REFERENCE                       OUTPUT  1530
C                                                                               OUTPUT  1540
   29 DO 30 I=1,3                                                               OUTPUT  1550
      ACC(I,J) = WMEGD(I,L)/(2.0*PI)                                            OUTPUT  1560
   30 T1(I) = ACC(I,J)                                                          OUTPUT  1570
      IF  (LPMI(L).NE.0)  CALL DOT31 (DPMI(1,1,L),T1,ACC(1,J))                  OUTPUT  1580
      GO TO 33                                                                  OUTPUT  1590
C                                                                               OUTPUT  1600
C     5. SEGMENT ANGULAR VELOCITIES IN VEHICLE REFERENCE                        OUTPUT  1610
C                                                                               OUTPUT  1620
   31 CALL DOT31 (D(1,1,L),WMEG(1,L),T1)                                        OUTPUT  1630
      CALL MAT31 (D(1,1,NVEH),T1,T2)                                            OUTPUT  1640
      DO 32 I=1,3                                                               OUTPUT  1650
   32 ACC(I,J) = (T2(I)-WMEG(I,NVEH))/(2.0*PI)                                  OUTPUT  1660
   33 ACC(4,J) = DSQRT(ACC(1,J)**2+ACC(2,J)**2+ACC(3,J)**2)                     OUTPUT  1670
      GO TO 38                                                                  OUTPUT  1680
C                                                                               OUTPUT  1690
C     6. SEGMENT ANGULAR DISPLACEMENTS IN VEHICLE REFERENCE                     OUTPUT  1700
C                                                                               OUTPUT  1710
   34 IF  (LPMI(L).EQ.0)  GO TO 36                                              OUTPUT  1720
      CALL DOT33(DPMI(1,1,L),D(1,1,L),T4)                                       OUTPUT  1730
      CALL DOTT33(T4,D(1,1,NVEH),T1)                                            OUTPUT  1740
      GO TO 37                                                                  OUTPUT  1750
   36 CALL DOTT33 (D(1,1,L),D(1,1,NVEH),T1)                                     OUTPUT  1760
   37 CALL YPRDEG(T1,ACC(1,J))                                                  OUTPUT  1770
      TRACE = 0.5*(T1(1)+T2(2)+T3(3)-1.0)                                       OUTPUT  1780
      IF (TRACE.GT. 1.0) TRACE =  1.0                                           OUTPUT  1790
      IF (TRACE.LT.-1.0) TRACE = -1.0                                           OUTPUT  1800
      ACC(4,J) = DARCOS(TRACE)/RADIAN                                           OUTPUT  1810
      GO TO 38                                                                  OUTPUT  1820
C                                                                               OUTPUT  1830
C     7. JOINT PARAMETERS                                                       OUTPUT  1840
C                                                                               OUTPUT  1850
   35 ACC(1,J) = PRJNT(1,L)                                                     OUTPUT  1860
      ACC(2,J) = PRJNT(2,L)/RADIAN                                              OUTPUT  1870
      ACC(3,J) = PRJNT(3,L)/RADIAN                                              OUTPUT  1880
      ACC(4,J) = PRJNT(4,L)/RADIAN                                              OUTPUT  1890
      ACC(5,J) = DSQRT(PRJNT(5,L))                                              OUTPUT  1900
      ACC(6,J) = DSQRT(PRJNT(6,L))                                              OUTPUT  1910
      ACC(7,J) = DSQRT(PRJNT(7,L))                                              OUTPUT  1920
   38 CONTINUE                                                                  OUTPUT  1930
      IF  (.NOT.LTAPE8)  GO TO 40                                               OUTPUT  1940
      KK = 0                                                                    OUTPUT  1950
      I2 = 4                                                                    OUTPUT  1960
      IF  (K.EQ.7)  I2 = 7                                                      OUTPUT  1970
      DO  39  J=J1,J2                                                           OUTPUT  1980
      DO  39  I=1,I2                                                            OUTPUT  1990
      KK = KK+1                                                                 OUTPUT  2000
```

```
   39 TDATA(KK,NT-20) = ACC(I,J)                                          OUTPUT  2010
   40 IF  (.NOT.LTHIST)  GO TO 43                                         OUTPUT  2020
      IF (K.LE.6) WRITE (NT,41) USEC,((ACC(I,J),I=1,4),J=J1,J2)           OUTPUT  2030
   41 FORMAT(F9.3,3(3X,4F9.3) )                                           OUTPUT  2040
      IF (K.EQ.7) WRITE (NT,42) USEC,((ACC(I,J),I=1,7),J=J1,J2)           OUTPUT  2050
   42 FORMAT(F9.3,2(F5.0,3F9.3,2X,3F9.3))                                 OUTPUT  2060
   43 CONTINUE                                                            OUTPUT  2070
   44 CONTINUE                                                            OUTPUT  2080
C                                                                         OUTPUT  2090
C     PRINT PLANE FORCES.                                                 OUTPUT  2100
C                                                                         OUTPUT  2110
      MPSF = 0                                                            OUTPUT  2120
      IF  (NPL.EQ.0)  GO TO 49                                            OUTPUT  2130
      DO  45  J=1,NPL                                                     OUTPUT  2140
   45 MPSF = MPSF + MNPL(J)                                               OUTPUT  2150
      IF  (MPSF.EQ.0)  GO TO 49                                           OUTPUT  2160
      DO 47 J1=1,MPSF,2                                                   OUTPUT  2170
      J2 = MIN0(J1+1,MPSF)                                                OUTPUT  2180
      NT = NT+1                                                           OUTPUT  2190
      IF  (.NOT.LTAPE8)  GO TO 47                                         OUTPUT  2200
      KK = 0                                                              OUTPUT  2210
      DO  46  J=J1,J2                                                     OUTPUT  2220
      DO  46  I=1,7                                                       OUTPUT  2230
      KK = KK+1                                                           OUTPUT  2240
   46 TDATA(KK,NT-20) = PSF(I,J)                                          OUTPUT  2250
   47 IF  (LTHIST)  WRITE (NT,48) USEC,((PSF(I,J),I=1,7),J=J1,J2)         OUTPUT  2260
   48 FORMAT(F9.3,2(F9.3,3F9.2,3F8.3) )                                   OUTPUT  2270
C                                                                         OUTPUT  2280
C     PRINT BELT FORCES                                                   OUTPUT  2290
C                                                                         OUTPUT  2300
   49 MBSF = 0                                                            OUTPUT  2310
      IF  (NBLT.EQ.0)  GO TO 67                                           OUTPUT  2320
      DO  50  J=1,NBLT                                                    OUTPUT  2330
   50 MBSF = MBSF + MNBLT(J)                                              OUTPUT  2340
      IF  (MBSF.EQ.0)  GO TO 67                                           OUTPUT  2350
      DO 52 J1=1,MBSF,2                                                   OUTPUT  2360
      J2 = MIN0(J1+1,MBSF)                                                OUTPUT  2370
      NT = NT+1                                                           OUTPUT  2380
      IF  (.NOT.LTAPE8)  GO TO 52                                         OUTPUT  2390
      KK = 0                                                              OUTPUT  2400
      DO  51  J=J1,J2                                                     OUTPUT  2410
      DO  51  I=1,4                                                       OUTPUT  2420
      KK = KK+1                                                           OUTPUT  2430
   51 TDATA(KK,NT-20) = BSF(I,J)                                          OUTPUT  2440
   52 IF  (LTHIST)  WRITE (NT,53) USEC,((BSF(I,J),I=1,4),J=J1,J2)         OUTPUT  2450
   53 FORMAT(F9.3,4(F15.6,F12.2,3X) )                                     OUTPUT  2460
C                                                                         OUTPUT  2470
C     PRINT HARNESS-BELT ENDPOINT FORCES (STORED IN BSF ARRAY).           OUTPUT  2480
C                                                                         OUTPUT  2490
   67 IF  (NHRNSS.LE.0)  GO TO 71                                         OUTPUT  2500
```

```
      MBSF1 = MBSF + 1                                                  OUTPUT  2510
      DO  68  I=1,NHRNSS                                                OUTPUT  2520
   68 MBSF = MBSF + NBLTPH(I)                                           OUTPUT  2530
      DO  70  J1=MBSF1,MBSF,2                                           OUTPUT  2540
      J2 = MIN0(J1+1,MBSF)                                              OUTPUT  2550
      NT = NT+1                                                         OUTPUT  2560
      IF  (.NOT.LTAPE8)  GO TO 70                                       OUTPUT  2570
      KK = 0                                                            OUTPUT  2580
      DO  69  J=J1,J2                                                   OUTPUT  2590
      DO  69  I=1,4                                                     OUTPUT  2600
      KK = KK+1                                                         OUTPUT  2610
   69 TDATA(KK,NT-20) = BSF(I,J)                                        OUTPUT  2620
   70 IF  (LTHIST)  WRITE (NT,53) USEC,((BSF(I,J),I=1,4),J=J1,J2)       OUTPUT  2630
C                                                                       OUTPUT  2640
C     PRINT SPRING DAMPER FORCES (STORED IN BSF ARRAY).                 OUTPUT  2650
C                                                                       OUTPUT  2660
   71 IF  (NSD.LE.0)  GO TO 54                                          OUTPUT  2670
      MBSF1 = MBSF + 1                                                  OUTPUT  2680
      MBSF = MBSF + (NSD+1)/2                                           OUTPUT  2690
      DO  73  J1=MBSF1,MBSF,2                                           OUTPUT  2700
      J2 = MIN0(J1+1,MBSF)                                              OUTPUT  2710
      NT = NT+1                                                         OUTPUT  2720
      IF  (.NOT.LTAPE8)  GO TO 73                                       OUTPUT  2730
      KK = 0                                                            OUTPUT  2740
      DO  72  J=J1,J2                                                   OUTPUT  2750
      DO  72  I=1,4                                                     OUTPUT  2760
      KK = KK+1                                                         OUTPUT  2770
   72 TDATA(KK,NT-20) = BSF(I,J)                                        OUTPUT  2780
   73 IF  (LTHIST)  WRITE (NT,74) USEC,((BSF(I,J),I=1,4),J=J1,J2)       OUTPUT  2790
   74 FORMAT (F9.3,4(F14.3,F12.2,4X))                                   OUTPUT  2800
C                                                                       OUTPUT  2810
C     PRINT SEGMENT CONTACT FORCES.                                     OUTPUT  2820
C                                                                       OUTPUT  2830
   54 MSSF = 0                                                          OUTPUT  2840
      DO  55  J=1,NSEG                                                  OUTPUT  2850
   55 MSSF = MSSF + MNSEG(J)                                            OUTPUT  2860
      IF  (MSSF.EQ.0)  GO TO 59                                         OUTPUT  2870
      DO 57 J=1,MSSF                                                    OUTPUT  2880
      NT = NT+1                                                         OUTPUT  2890
      IF  (.NOT.LTAPE8)  GO TO 57                                       OUTPUT  2900
      DO  56  I=1,10                                                    OUTPUT  2910
   56 TDATA(I,NT-20) = SSF(I,J)                                         OUTPUT  2920
   57 IF  (LTHIST)  WRITE (NT,58) USEC,(SSF(I,J),I=1,10)                OUTPUT  2930
   58 FORMAT(2F9.3,3F9.2,3F8.3,2X,3F8.3)                                OUTPUT  2940
C                                                                       OUTPUT  2950
C     PRINT AIRBAG FORCES                                               OUTPUT  2960
C                                                                       OUTPUT  2970
   59 IF (NBAG.EQ.0) GO TO 65                                           OUTPUT  2980
      K1 = 1                                                            OUTPUT  2990
      DO 64 J=1,NBAG                                                    OUTPUT  3000
```

```
      IF (MNBAG(J).EQ.0) GO TO 64                                          OUTPUT  3010
      KBAG = MNBAG(J)+NPANEL(J)+5                                          OUTPUT  3020
      DO 63 J1=1,KBAG,4                                                    OUTPUT  3030
      J2 = MIN0(J1+3,KBAG)                                                 OUTPUT  3040
      K2 = K1+J2-J1                                                        OUTPUT  3050
      NT = NT+1                                                            OUTPUT  3060
      IF  (.NOT.LTAPE8)  GO TO 61                                          OUTPUT  3070
      KK = 0                                                               OUTPUT  3080
      DO  60  K=K1,K2                                                      OUTPUT  3090
      DO  60  I=1,3                                                        OUTPUT  3100
      KK = KK+1                                                            OUTPUT  3110
   60 TDATA(KK,NT-20) = BAGSF(I,K)                                         OUTPUT  3120
   61 IF  (.NOT.LTHIST)  GO TO 63                                          OUTPUT  3130
      IF  (J1.EQ.1)  WRITE (NT,75) USEC,((BAGSF(I,K),I=1,3),K=K1,K2)       OUTPUT  3140
      IF  (J1.NE.1)  WRITE (NT,62) USEC,((BAGSF(I,K),I=1,3),K=K1,K2)       OUTPUT  3150
   75 FORMAT (F9.3,3X,3F9.2,2(3X,3F9.3),3X,3F9.2)                          OUTPUT  3160
   62 FORMAT(F9.3,4(3X,3F9.2))                                             OUTPUT  3170
   63 K1 = K2+1                                                            OUTPUT  3180
   64 CONTINUE                                                             OUTPUT  3190
   65 NT = NT-20                                                           OUTPUT  3200
      IF  (LTAPE8)  WRITE (8) NT,USEC,((TDATA(I,J),I=1,14),J=1,NT)         OUTPUT  3210
      PREVT = TIME                                                         OUTPUT  3220
      CALL ELTIME(2,8)                                                     OUTPUT  3230
   66 RETURN                                                               OUTPUT  3240
      END                                                                  OUTPUT  3250
```

```
      SUBROUTINE PANEL (DRR,ZR,JB)                                      PANEL   0010
C                                                          REV 19 08/05/78 PANEL   0020
C     COMPUTES AIRBAG PARAMETERS DURING INFLATION OF BAG.                 PANEL   0030
C                                                                         PANEL   0040
C         GIVEN: DRR - DC MATRIX RELATIVE TO VEHICLE                      PANEL   0050
C                ZR - CG LOCATION IN VEHICLE REFERENCE                    PANEL   0060
C                                                                         PANEL   0070
C       COMPUTE: SEGLP,SEGLV,SEGLA,D,WMEG & WMEGD FOR SEGMENT JB.         PANEL   0080
C                                                                         PANEL   0090
      IMPLICIT REAL*8 (A-H,O-Z)                                           PANEL   0100
      DIMENSION DRR(3,3),ZR(3),T1(3),T2(3)                                PANEL   0110
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,             PANEL   0120
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)          PANEL   0130
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),  PANEL   0140
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)         PANEL   0150
      CALL MAT33 (DRR,D(1,1,NVEH),D(1,1,JB))                              PANEL   0160
      CALL MAT31 (DRR,WMEG(1,NVEH),WMEG(1,JB))                            PANEL   0170
      CALL DOT31 (D(1,1,NVEH),ZR,SEGLP(1,JB))                             PANEL   0180
      CALL CROSS (WMEG(1,NVEH),ZR,T1)                                     PANEL   0190
      CALL DOT31 (D(1,1,NVEH),T1,SEGLV(1,JB))                             PANEL   0200
      CALL CROSS (WMEG(1,NVEH),T1,T2)                                     PANEL   0210
      CALL DOT31 (D(1,1,NVEH),T2,SEGLA(1,JB))                             PANEL   0220
      DO  10  I=1,3                                                       PANEL   0230
      SEGLP(I,JB) = SEGLP(I,JB) + SEGLP(I,NVEH)                           PANEL   0240
      SEGLV(I,JB) = SEGLV(I,JB) + SEGLV(I,NVEH)                           PANEL   0250
      SEGLA(I,JB) = SEGLA(I,JB) + SEGLA(I,NVEH)                           PANEL   0260
   10 WMEGD(I,JB) = WMEGD(I,NVEH)                                         PANEL   0270
      RETURN                                                              PANEL   0280
      END                                                                 PANEL   0290
```

```
      SUBROUTINE PDAUX (VAR,DER,NEQ,KDINT)                                  PDAUX   0010
C                                                          REV 20 12/18/79  PDAUX   0020
C     PURPOSE IS TO ACT AS INTERFACE BETWEEN INTEGRATOR AND DAUX TO        PDAUX   0030
C     ACCOMODATE VARIABLE NUMBER OF FUNCTIONS TO BE INTEGRATED.            PDAUX   0040
C                                                                          PDAUX   0050
C     ARGUMENTS:                                                           PDAUX   0060
C        VAR - ARRAY OF NEQ STATE VARIABLES UPDATED BY DINT.               PDAUX   0070
C        DER - ARRAY OF NEQ DERIVATIVES TO BE SUPPLIED BY DAUX.            PDAUX   0080
C        NEQ - NUMBER OF STATE VARIABLES AND DERIVATIVES.                  PDAUX   0090
C      KDINT - INTEGRATION STEP NUMBER IN DINT.                            PDAUX   0100
C                                                                          PDAUX   0110
      IMPLICIT REAL*8 (A-H,O-Z)                                            PDAUX   0120
      DIMENSION VAR(3,1),DER(3,1)                                          PDAUX   0130
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,              PDAUX   0140
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)           PDAUX   0150
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),   PDAUX   0160
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)          PDAUX   0170
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(3,60),HA(3,60),HB(3,60),    PDAUX   0180
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),        PDAUX   0190
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)       PDAUX   0200
      COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),              PDAUX   0210
     *               BLTTTL(5,8),PLTTL(5,30),BAGTTL(5,6),SEG(30),          PDAUX   0220
     *               JOINT(30),CGS(30),JS(30)                              PDAUX   0230
      REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT         PDAUX   0240
      LOGICAL*1 CGS,JS                                                     PDAUX   0250
      COMMON/INTEST/ SGTEST(3,4,30),XTEST(3,120),SEGT(120),REGT(120)       PDAUX   0260
      REAL    SEGT                                                         PDAUX   0270
      COMMON/FLXBLE/ HF(4,12,8),B42(3,3,24),V4(3,8),NFLEX(3,8)             PDAUX   0280
      COMMON/TEMPVS/ T(3,30),VXT(3)                                        PDAUX   0290
      DIMENSION SD(3,3,30) , E1(30) , NTST(30) , LSEG(30) , RGTTL(4)       PDAUX   0300
      LOGICAL  LSEG                                                        PDAUX   0310
      DATA NTST/30*0/                                                      PDAUX   0320
      DATA RGTTL/8HANG VEL ,8HLIN VEL ,8HANG ACC ,8HLIN ACC /              PDAUX   0330
      CALL ELTIME(1,6)                                                     PDAUX   0340
      MBAG = NGRND                                                         PDAUX   0350
      KK = KDINT                                                           PDAUX   0360
      IF (NTST(1).NE.0) GO TO 10                                           PDAUX   0370
      DO 5 M=1,MBAG                                                        PDAUX   0380
      LSEG(M) = M.GT.1 .AND. ISING(M).GE.0 .AND. JNT(M-1).NE.0             PDAUX   0390
    5 NTST(M) = M                                                          PDAUX   0400
      NTST(NGRND) = -NGRND                                                 PDAUX   0410
      LSEG(NGRND) = .TRUE.                                                 PDAUX   0420
      IF (NFLX.EQ.0) GO TO 10                                              PDAUX   0430
      DO 6 J=1,NFLX                                                        PDAUX   0440
      M = NFLEX(2,J)                                                       PDAUX   0450
    6 NTST(M) = -M                                                         PDAUX   0460
   10 IF  (KDINT.EQ.4)  GO TO 48                                           PDAUX   0470
      IF  (KDINT.GT.0)  GO TO 20                                           PDAUX   0480
C                                                                          PDAUX   0490
C     KDINT=0 IMPLIES INITIAL CALL FROM DINT. PDAUX TO SUPPLY INITIAL      PDAUX   0500
```

```
C     VALUES TO STATE VARIABLES AND COMPUTE VALUE OF NEQ.                PDAUX   0510
C                                                                        PDAUX   0520
C                                                                        PDAUX   0530
C       (A) SET Q TO IDENTITY QUATERNION                                 PDAUX   0540
C                                                                        PDAUX   0550
      N = 0                                                              PDAUX   0560
      DO 12 M=1,MBAG                                                     PDAUX   0570
      IF (NTST(M).LT.0) GO TO 12                                         PDAUX   0580
      N = N+1                                                            PDAUX   0590
      REGT(N) = RGTTL(1)                                                 PDAUX   0600
      SEGT(N) = SEG(M)                                                   PDAUX   0610
      E1(N) = 1.0                                                        PDAUX   0620
      DO 11 I=1,3                                                        PDAUX   0630
      XTEST(I,N) = SGTEST(I,1,M)**2                                      PDAUX   0640
   11 VAR(I,N) = 0.0                                                     PDAUX   0650
   12 CONTINUE                                                           PDAUX   0660
C                                                                        PDAUX   0670
C       (B) SEGLP OF REFERENCE SEGMENTS                                  PDAUX   0680
C                                                                        PDAUX   0690
      DO 14 M=1,MBAG                                                     PDAUX   0700
      IF  (LSEG(M))  GO TO 14                                            PDAUX   0710
      N = N+1                                                            PDAUX   0720
      REGT(N) = RGTTL(2)                                                 PDAUX   0730
      SEGT(N) = SEG(M)                                                   PDAUX   0740
      DO 13 I=1,3                                                        PDAUX   0750
      XTEST(I,N) = SGTEST(I,2,M)**2                                      PDAUX   0760
   13 VAR(I,N) = SEGLP(I,M)                                              PDAUX   0770
   14 CONTINUE                                                           PDAUX   0780
C                                                                        PDAUX   0790
C       (C) WMEG                                                         PDAUX   0800
C                                                                        PDAUX   0810
      DO 16 M=1,MBAG                                                     PDAUX   0820
      IF (NTST(M).LT.0) GO TO 16                                         PDAUX   0830
      N = N+1                                                            PDAUX   0840
      REGT(N) = RGTTL(3)                                                 PDAUX   0850
      SEGT(N) = SEG(M)                                                   PDAUX   0860
      DO 15 I=1,3                                                        PDAUX   0870
      XTEST(I,N) = SGTEST(I,3,M)**2                                      PDAUX   0880
   15 VAR(I,N) = WMEG(I,M)                                               PDAUX   0890
   16 CONTINUE                                                           PDAUX   0900
C                                                                        PDAUX   0910
C       (D) SEGLV OF REFERENCE SEGMENTS                                  PDAUX   0920
C                                                                        PDAUX   0930
      DO 18 M=1,MBAG                                                     PDAUX   0940
      IF  (LSEG(M))  GO TO 18                                            PDAUX   0950
      N = N+1                                                            PDAUX   0960
      REGT(N) = RGTTL(4)                                                 PDAUX   0970
      SEGT(N) = SEG(M)                                                   PDAUX   0980
      DO 17 I=1,3                                                        PDAUX   0990
      XTEST(I,N) = SGTEST(I,4,M)**2                                      PDAUX   1000
```

```
   17 VAR(I,N) = SEGLV(I,M)                                              PDAUX   1010
   18 CONTINUE                                                           PDAUX   1020
      NEQ = 3*N                                                          PDAUX   1030
      GO TO 40                                                           PDAUX   1040
   20 IF (KDINT.NE.1) GO TO 30                                           PDAUX   1050
C                                                                        PDAUX   1060
C     KDINT = 1, 1ST STEP IN ADVANCING INTEGRATING INTERVAL,             PDAUX   1070
C                SAVE DC MATRICES IF TIME HAS ADVANCED.                  PDAUX   1080
C                                                                        PDAUX   1090
      N = 0                                                              PDAUX   1100
      DO 22 M=1,MBAG                                                     PDAUX   1110
      IF (NTST(M).LT.0) GO TO 22                                         PDAUX   1120
      N = N+1                                                            PDAUX   1130
      DO 21 J=1,3                                                        PDAUX   1140
      DO 21 I=1,3                                                        PDAUX   1150
   21 SD(I,J,N) = D(I,J,M)                                               PDAUX   1160
   22 CONTINUE                                                           PDAUX   1170
C                                                                        PDAUX   1180
C     KDINT > 0,1 - FETCH SAVED DC MATRICES AND UPDATE BY CURRENT THETA. PDAUX   1190
C                                                                        PDAUX   1200
C       (A) UPDATE D BY Q                                                PDAUX   1210
C                                                                        PDAUX   1220
   30 N = 0                                                              PDAUX   1230
      DO 32 M=1,MBAG                                                     PDAUX   1240
      IF (NTST(M).LT.0) GO TO 32                                         PDAUX   1250
      N = N+1                                                            PDAUX   1260
      EDOTE = VAR(1,N)**2 + VAR(2,N)**2 + VAR(3,N)**2                    PDAUX   1270
      IF (EDOTE.GE.1.0) KDINT = -KDINT                                   PDAUX   1280
      IF (KDINT.LE.0)  GO TO 99                                          PDAUX   1290
      E1(N) = DSQRT(1.0-EDOTE)                                           PDAUX   1300
      CALL DSETQ(SD(1,1,N),VAR(1,N),EDOTE,E1(N),D(1,1,M))                PDAUX   1310
   32 CONTINUE                                                           PDAUX   1320
C                                                                        PDAUX   1330
C     KDINT > 0 - STORE STATE VARIABLES INTO PROGRAM ARRAYS.             PDAUX   1340
C                                                                        PDAUX   1350
C       (B) SEGLP OF REFERENCE SEGMENTS                                  PDAUX   1360
C                                                                        PDAUX   1370
      DO 35 M=1,MBAG                                                     PDAUX   1380
      IF  (LSEG(M))  GO TO 35                                            PDAUX   1390
      N = N+1                                                            PDAUX   1400
      DO 34 I=1,3                                                        PDAUX   1410
   34 SEGLP(I,M) = VAR(I,N)                                              PDAUX   1420
   35 CONTINUE                                                           PDAUX   1430
C                                                                        PDAUX   1440
C       (C) WMEG                                                         PDAUX   1450
C                                                                        PDAUX   1460
      DO 31 M=1,MBAG                                                     PDAUX   1470
      IF (NTST(M).LT.0) GO TO 31                                         PDAUX   1480
      N = N+1                                                            PDAUX   1490
      DO 36 I=1,3                                                        PDAUX   1500
```

```
   36 WMEG(I,M) = VAR(I,N)                                              PDAUX   1510
   31 CONTINUE                                                          PDAUX   1520
C                                                                       PDAUX   1530
C       (D) SEGLV OF REFERENCE SEGMENTS                                 PDAUX   1540
C                                                                       PDAUX   1550
      DO 38 M=1,MBAG                                                    PDAUX   1560
      IF  (LSEG(M))  GO TO 38                                           PDAUX   1570
      N = N+1                                                           PDAUX   1580
      DO 37 I=1,3                                                       PDAUX   1590
   37 SEGLV(I,M) = VAR(I,N)                                             PDAUX   1600
   38 CONTINUE                                                          PDAUX   1610
C                                                                       PDAUX   1620
C     CALL DAUX ROUTINE TO COMPUTE DERIVATIVES                          PDAUX   1630
C                                                                       PDAUX   1640
   40 CALL DAUX(0)                                                      PDAUX   1650
C                                                                       PDAUX   1660
C     STORE DERIVATIVES FOR INTEGRATING SUBROUTINE.                     PDAUX   1670
C                                                                       PDAUX   1680
C       (A) DERIVATIVE OF Q                                             PDAUX   1690
C                                                                       PDAUX   1700
      N = 0                                                             PDAUX   1710
      DO 39 M=1,MBAG                                                    PDAUX   1720
      IF (NTST(M).LT.0) GO TO 39                                        PDAUX   1730
      N = N+1                                                           PDAUX   1740
      CALL CROSS(VAR(1,N),WMEG(1,M),VXT)                                PDAUX   1750
      DO 41 I=1,3                                                       PDAUX   1760
   41 DER(I,N) = 0.5*(E1(N)*WMEG(I,M) + VXT(I) )                        PDAUX   1770
   39 CONTINUE                                                          PDAUX   1780
      NQUAT = N                                                         PDAUX   1790
C                                                                       PDAUX   1800
C       (B) SEGLV OF REFERENCE SEGMENTS                                 PDAUX   1810
C                                                                       PDAUX   1820
      DO 43 M=1,MBAG                                                    PDAUX   1830
      IF  (LSEG(M))  GO TO 43                                           PDAUX   1840
      N = N+1                                                           PDAUX   1850
      DO 42 I=1,3                                                       PDAUX   1860
   42 DER(I,N) = SEGLV(I,M)                                             PDAUX   1870
   43 CONTINUE                                                          PDAUX   1880
C                                                                       PDAUX   1890
C       (C) WMEGD                                                       PDAUX   1900
C                                                                       PDAUX   1910
      DO 47 M=1,MBAG                                                    PDAUX   1920
      IF (NTST(M).LT.0) GO TO 47                                        PDAUX   1930
      N = N+1                                                           PDAUX   1940
      DO 44 I=1,3                                                       PDAUX   1950
   44 DER(I,N) = WMEGD(I,M)                                             PDAUX   1960
   47 CONTINUE                                                          PDAUX   1970
C                                                                       PDAUX   1980
C       (D) SEGLA OF REFERENCE SEGMENTS                                 PDAUX   1990
C                                                                       PDAUX   2000
```

```
      DO 46 M=1,MBAG                                                    PDAUX   2010
      IF  (LSEG(M))  GO TO 46                                           PDAUX   2020
      N = N+1                                                           PDAUX   2030
      DO 45 I=1,3                                                       PDAUX   2040
   45 DER(I,N) = SEGLA(I,M)                                             PDAUX   2050
   46 CONTINUE                                                          PDAUX   2060
      IF (KDINT.NE.4) GO TO 99                                          PDAUX   2070
   48 N = 0                                                             PDAUX   2080
      DO 51 M=1,MBAG                                                    PDAUX   2090
      IF (NTST(M).LT.0) GO TO 51                                        PDAUX   2100
      N = N+1                                                           PDAUX   2110
      E1(N) = 1.0                                                       PDAUX   2120
      DO 50 I=1,3                                                       PDAUX   2130
      DER(I,N) = 0.5*WMEG(I,M)                                          PDAUX   2140
   50 VAR(I,N) = 0.0                                                    PDAUX   2150
   51 CONTINUE                                                          PDAUX   2160
   99 IF  (KDINT.EQ.2)  KDINT = NQUAT                                   PDAUX   2170
      CALL ELTIME(2,6)                                                  PDAUX   2180
      RETURN                                                            PDAUX   2190
      END                                                               PDAUX   2200
```

```
      SUBROUTINE PLELP(M,MM,N,NN,NT)                                    PLELP   0010
C                                                        REV 20 04/11/80  PLELP   0020
C     COMPUTES FORCES (WHICH ARE ADDED TO U1 ARRAY)                      PLELP   0030
C      AND TORQUES    (WHICH ARE ADDED TO U2 ARRAY)                      PLELP   0040
C     OF ELLIPSOID (MM) ATTACHED TO BODY SEGMENT (M)                     PLELP   0050
C     INTERSECTING PLANE (NN) ATTACHED TO SEGMENT (N).                   PLELP   0060
C                                                                        PLELP   0070
      IMPLICIT REAL *8(A-H,O-Z)                                          PLELP   0080
      COMMON/TABLES/ MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(500),TAB(2600) PLELP   0090
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30), PLELP   0100
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       PLELP   0110
      COMMON/FORCES/ PSF(7,30),BSF(4,20),SSF(10,20),BAGSF(3,20),         PLELP   0120
     *               PRJNT(7,30),NPANEL(5),NPSF,NBSF,NSSF,NBGSF         PLELP   0130
      COMMON/CNTSRF/ PL(17,30),BELT(20,8),TPTS(6,8),BD(24,40)            PLELP   0140
      COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),    PLELP   0150
     *               HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12), PLELP   0160
     *               RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),              PLELP   0170
     *               KQ1(12),KQ2(12),KQTYPE(12)                         PLELP   0180
C     THIS COMMON/TEMPVS/ IS SHARED BY PLELP, PLSEGF AND SEGSEG.         PLELP   0190
      COMMON/TEMPVS/DMNT(3,3),TEMP(3,3),B(3,3),XMN(3),RLN(3),XMM(3),     PLELP   0200
     *              TM(3),R(3),RM(3),DMNWN(3),RLM(3),RN(3),VMN(3),VR(3), PLELP   0210
     *              WMN(3),WCM(3),WCN(3),VREL(3),FFM(3),FR(3),TQM(3),    PLELP   0220
     *              TQN(3),TQNT(3),T(3),H(3),T1(3),T2(3),RMD(3),RND(3),  PLELP   0230
     *              TD(3),TT4(3,4),TT5(3,4),T3(3),T4(3),P,AMR,FM,CF,     PLELP   0240
     *              VRM,VRT,VRTS,VRTEST,TF,ELOSS,MCF,NCF                 PLELP   0250
      CALL ELTIME(1,21)                                                  PLELP   0260
C                                                                        PLELP   0270
C     COMPUTE PENETRATION DISTANCE, IF NEGATIVE, RETURN.                 PLELP   0280
C                                                                        PLELP   0290
      CALL DOTT33(D(1,1,M),D(1,1,N),DMNT)                                PLELP   0300
      DO 10 I=1,3                                                        PLELP   0310
   10 XMN(I) = SEGLP(I,M) - SEGLP(I,N)                                   PLELP   0320
      CALL MAT31(D(1,1,M),XMN,XMM)                                       PLELP   0330
      CALL MAT31(DMNT,PL(1,NN),TM)                                       PLELP   0340
      BET = PL(4,NN)                                                     PLELP   0350
      DO 11 I=1,3                                                        PLELP   0360
   11 BET = BET - TM(I)*(BD(I+3,MM)+XMM(I))                              PLELP   0370
      CALL MAT31(BD(16,MM),TM,T4)                                        PLELP   0380
      BTS = TM(1)*T4(1) + TM(2)*T4(2) + TM(3)*T4(3)                      PLELP   0390
      BTE = -DSQRT(BTS)                                                  PLELP   0400
      P = BET - BTE                                                      PLELP   0410
      PSF(1,NPSF) = P                                                    PLELP   0420
      MCF = NTAB(NT+1)                                                   PLELP   0430
      NCF = -MCF                                                         PLELP   0440
      IF (NCF.GT.0) CFQQ(NCF) = -999.                                    PLELP   0450
      IF (P.LE.0.0) GO TO 99                                             PLELP   0460
C                                                                        PLELP   0470
C     IF COMPLETE PENETRATION, RETURN                                    PLELP   0480
C                                                                        PLELP   0490
      IF (BET+BTE.GT.0.0) GO TO 99                                       PLELP   0500
```

```
C                                                                       PLELP   0510
C     COMPUTE TG - THE POINT IN SEGMENT REFERENCE AT WHICH THE CONTACT   PLELP   0520
C                  FORCES ARE TO BE APPLIED WHICH LIES ON THE SCALED     PLELP   0530
C                  LINE BETWEEN THE POINT OF MAXIMUM PENETRATION (RHO=0) PLELP   0540
C                  AND THE CENTER OF THE INTERSECTION ELLIPSE (RHO=1).   PLELP   0550
C       AND TEMP - THE SAME POINT IN VEHICLE REFERENCE.                  PLELP   0560
C                                                                       PLELP   0570
      RHO = 0.0                                                         PLELP   0580
      IF (MCF.GT.0) RHO = TAB(MCF+4)                                    PLELP   0590
      BETE = (1.0+RHO*P/BTE)/BTE                                        PLELP   0600
      AMR  = -1.0/BTE                                                   PLELP   0610
      DO 13 I=1,3                                                       PLELP   0620
      RM(I) = BETE*T4(I)                                                PLELP   0630
      RLM(I) = RM(I) + BD(I+3,MM)                                       PLELP   0640
   13 RN(I)  = RLM(I) + XMM(I)                                          PLELP   0650
      CALL DOT31(DMNT,RN,RLN)                                           PLELP   0660
C                                                                       PLELP   0670
C     IF BOUNDARY PLANE IS GIVEN, COMPUTE DISTANCE FROM POINT TO PLANE, PLELP   0680
C         IF NEGATIVE OR > LIMIT, RETURN.                               PLELP   0690
C                                                                       PLELP   0700
      DO 14 I=8,13,5                                                    PLELP   0710
      IF (PL(I+4,NN).LE.0.0) GO TO 14                                   PLELP   0720
      DIST = RLN(1)*PL(I  ,NN)                                          PLELP   0730
     *     + RLN(2)*PL(I+1,NN)                                          PLELP   0740
     *     + RLN(3)*PL(I+2,NN) - PL(I+3,NN)                             PLELP   0750
      IF (DIST.LE.0.0 .OR. DIST.GT.PL(I+4,NN)) GO TO 99                 PLELP   0760
   14 CONTINUE                                                          PLELP   0770
      CALL PLSEGF(M,N,NT)                                               PLELP   0780
      IF (MCF.LT.0) GO TO 30                                            PLELP   0790
C                                                                       PLELP   0800
C     STORE RESULTS FOR OUTPUT ROUTINE.                                 PLELP   0810
C                                                                       PLELP   0820
      PSF(2,NPSF) = FM                                                  PLELP   0830
      TF2FM2 = TF**2 - FM**2                                            PLELP   0840
      IF (TF2FM2.LE.0.0) TF2FM2 = 0.0                                   PLELP   0850
      PSF(3,NPSF) = DSQRT(TF2FM2)                                       PLELP   0860
      PSF(4,NPSF) = TF                                                  PLELP   0870
      DO 24 I=1,3                                                       PLELP   0880
   24 PSF(I+4,NPSF) = RLN(I)                                            PLELP   0890
      GO TO 99                                                          PLELP   0900
   30 DO 31 I=1,3                                                       PLELP   0910
      PSF(I+1,NPSF) = T(I)                                              PLELP   0920
   31 PSF(I+4,NPSF) = RLN(I)                                            PLELP   0930
      CALL CROSS(WMN,TM,T1)                                             PLELP   0940
      CALL MAT31(BD(16,MM),T1,T2)                                       PLELP   0950
      TMT = TM(1)*T2(1) + TM(2)*T2(2) + TM(3)*T2(3)                     PLELP   0960
      TMT = TMT/BTS                                                     PLELP   0970
      TRT = RHO*(VRM+P*TMT)/BTS                                         PLELP   0980
      DO 32 I=1,3                                                       PLELP   0990
      T3(I) = VR(I) + VRM*TM(I)                                         PLELP   1000
   32 RMD(I) = BETE*T2(I) - TMT*RM(I) - TRT*T4(I)                       PLELP   1010
      CALL CROSS(DMNWN,T3,T1)                                           PLELP   1020
      CALL CROSS(WMN,RMD,T3)                                            PLELP   1030
      SQQ(NCF) = 0.0                                                    PLELP   1040
      DO 36 I=1,3                                                       PLELP   1050
      SQQ(NCF) = SQQ(NCF) + TM(I)*(T3(I)+2.0*T1(I))                     PLELP   1060
   36 T3(I) = T3(I) + T1(I)                                             PLELP   1070
      CALL DOT31(D(1,1,M),T3,RQQ(1,NCF))                                PLELP   1080
   99 CALL ELTIME(2,21)                                                 PLELP   1090
      RETURN                                                            PLELP   1100
      END                                                               PLELP   1110
```

```
      SUBROUTINE PLSEGF(M,N,NT)                                          PLSEGF  0010
C                                                          REV 19 10/19/79 PLSEGF  0020
      IMPLICIT REAL*8 (A-H,O-Z)                                          PLSEGF  0030
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30), PLSEGF  0040
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       PLSEGF  0050
      COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),    PLSEGF  0060
     *               HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12), PLSEGF  0070
     *               RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),              PLSEGF  0080
     *               KQ1(12),KQ2(12),KQTYPE(12)                         PLSEGF  0090
      COMMON/TEMPVI/ CREST,TTI(3),R1I(3),R2I(3),JSTOP(4,2,30)            PLSEGF  0100
      COMMON/TABLES/ MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(500),TAB(2600) PLSEGF  0110
C     THIS COMMON/TEMPVS/ IS SHARED BY PLELP, PLSEGF AND SEGSEG.         PLSEGF  0120
      COMMON/TEMPVS/DMNT(3,3),TEMP(3,3),B(3,3),XMN(3),RLN(3),XMM(3),     PLSEGF  0130
     *              TM(3),R(3),RM(3),DMNWN(3),RLM(3),RN(3),VMN(3),VR(3), PLSEGF  0140
     *              WMN(3),WCM(3),WCN(3),VREL(3),FFM(3),FR(3),TQM(3),    PLSEGF  0150
     *              TQN(3),TQNT(3),T(3),H(3),T1(3),T2(3),RMD(3),RND(3),  PLSEGF  0160
     *              TD(3),TT4(3,4),TT5(3,4),T3(3),T4(3),P,AMR,FM,CF,     PLSEGF  0170
     *              VRM,VRT,VRTS,VRTEST,TF,ELOSS,MCF,NCF                 PLSEGF  0180
      VRTEST = 2.0                                                       PLSEGF  0190
      CALL MAT31(DMNT,WMEG(1,N),DMNWN)                                   PLSEGF  0200
      DO 15 I=1,3                                                        PLSEGF  0210
      VMN(I) = SEGLV(I,M) - SEGLV(I,N)                                   PLSEGF  0220
   15 WMN(I) = DMNWN(I) - WMEG(I,M)                                      PLSEGF  0230
      CALL DOT31(D(1,1,M),TM,T)                                          PLSEGF  0240
      CALL MAT31(D(1,1,M),VMN,VR)                                        PLSEGF  0250
      CALL CROSS(WMEG(1,M),RLM,WCM)                                      PLSEGF  0260
      CALL CROSS(DMNWN,RN,WCN)                                           PLSEGF  0270
      VRM = 0.0                                                          PLSEGF  0280
      DO 16 I=1,3                                                        PLSEGF  0290
      VR(I) = VR(I) + WCM(I) - WCN(I)                                    PLSEGF  0300
   16 VRM = VRM + VR(I)*TM(I)                                            PLSEGF  0310
      VRT = 0.0                                                          PLSEGF  0320
      DO 17 I=1,3                                                        PLSEGF  0330
      VREL(I) = VR(I) - VRM*TM(I)                                        PLSEGF  0340
   17 VRT = VRT + VREL(I)**2                                             PLSEGF  0350
      VRT = DSQRT(VRT)                                                   PLSEGF  0360
      CF = EVALFD (P,NTAB(NT+5),1)                                       PLSEGF  0370
      LT = NTAB(NT)                                                      PLSEGF  0380
      TAB(LT) = P                                                        PLSEGF  0390
      FM = 1.0                                                           PLSEGF  0400
      PDOT = -VRM                                                        PLSEGF  0410
      ELOSS = 0.0                                                        PLSEGF  0420
      IF (MCF.GT.0) CALL FRCDFL(P,PDOT,NT,1,FM,ELOSS)                    PLSEGF  0430
      VRTS = VRT                                                         PLSEGF  0440
      IF (VRT.LT.VRTEST) VRT = VRTEST/(2.0-VRT/VRTEST)                   PLSEGF  0450
      FF = -DABS(FM)*CF/VRT                                              PLSEGF  0460
      IF (NCF.GT.0.AND.KQTYPE(NCF).EQ.6)  FF=0.0                         PLSEGF  0470
      FS = (VRTS-VRT)/VRT                                                PLSEGF  0480
      IF (NCF.GT.0.AND.KQTYPE(NCF).EQ.6)  FS=0.0                         PLSEGF  0490
      TF = 0.0                                                           PLSEGF  0500
```

```
      L = LT+18                                                          PLSEGF  0510
      DO 18 I=1,3                                                        PLSEGF  0520
      L = L+1                                                            PLSEGF  0530
      FFM(I) = FM*TM(I) + FF*VREL(I) + FS*TAB(L)                         PLSEGF  0540
      TF = TF + FFM(I)**2                                                PLSEGF  0550
      TTI(I) = T(I)                                                      PLSEGF  0560
      R1I(I) = RLM(I)                                                    PLSEGF  0570
   18 R2I(I) = RLN(I)                                                    PLSEGF  0580
      TF = DSQRT(TF)                                                     PLSEGF  0590
      MT = NTAB(NT+5)                                                    PLSEGF  0600
      CREST = TAB(MT+3)                                                  PLSEGF  0610
      CALL DOT31 (D(1,1,M),FFM,FR)                                       PLSEGF  0620
      IF (MCF.LE.0) GO TO 21                                             PLSEGF  0630
      CALL CROSS (RLM,FFM,TQM)                                           PLSEGF  0640
      CALL CROSS (RN,FFM,TQNT)                                           PLSEGF  0650
      CALL DOT31 (DMNT,TQNT,TQN)                                         PLSEGF  0660
      DO 19 I=1,3                                                        PLSEGF  0670
      U1(I,M) = U1(I,M) + FR(I)                                          PLSEGF  0680
      U1(I,N) = U1(I,N) - FR(I)                                          PLSEGF  0690
      U2(I,M) = U2(I,M) + TQM(I)                                         PLSEGF  0700
   19 U2(I,N) = U2(I,N) - TQN(I)                                         PLSEGF  0710
      IF (NCF.LE.0) GO TO 23                                             PLSEGF  0720
   21 DO 22 I=1,3                                                        PLSEGF  0730
      HQQ(I,NCF) = FR(I)/TF                                              PLSEGF  0740
      TQQ(I,NCF) = T(I)                                                  PLSEGF  0750
      RK1(I,NCF) = RLM(I)                                                PLSEGF  0760
   22 RK2(I,NCF) = RLN(I)                                                PLSEGF  0770
      CFQQ(NCF)  = CF                                                    PLSEGF  0780
      MT = NTAB(NT+5)                                                    PLSEGF  0790
      IF (KQTYPE(NCF).EQ.3) CFQQ(NCF) = TAB(MT+4)                        PLSEGF  0800
   23 RETURN                                                             PLSEGF  0810
      END                                                                PLSEGF  0820
```

```
      SUBROUTINE PLTXYZ(P,C)                                                  PLTXYZ  0010
C                                                          REV 20 05/06/80    PLTXYZ  0020
C     STORES PLOT CHARACTER (C) INTO PLOTYZ, PLOTXZ AND PLOTXY ARRAYS         PLTXYZ  0030
C     IN VEHICLE REFERENCE FOR POINT (P) GIVEN IN INERTIAL REFERENCE.         PLTXYZ  0040
C                                                                             PLTXYZ  0050
      IMPLICIT REAL*8 (A-H,O-Z)                                               PLTXYZ  0060
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,                 PLTXYZ  0070
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)              PLTXYZ  0080
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),      PLTXYZ  0090
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)             PLTXYZ  0100
      COMMON/VPOSTN/ ZPLT(3),SPLT(3),AXV(3,6),VATAB(6,101,6),                 PLTXYZ  0110
     *               VT0(6),VDT(6),TIMEV(6),OMEGV(6),NVTAB(6),INDXV(6)        PLTXYZ  0120
      COMMON/TEMPVS/ DUM(101),PLOTYZ(96,55),PLOTXZ(96,55),PLOTXY(96,55)       PLTXYZ  0130
      LOGICAL*1      C,PLOTYZ,PLOTXZ,PLOTXY                                   PLTXYZ  0140
      DIMENSION P(3),TMP(3),XYZ(3)                                            PLTXYZ  0150
      DATA  NPLTZ/96/ , NPLTX/55/                                             PLTXYZ  0160
C                                                                             PLTXYZ  0170
C     CONVERT P FROM INERTIAL TO VEHICLE REFERENCE BY                         PLTXYZ  0180
C         XYZ = DVEH(P-XCOMP)                                                 PLTXYZ  0190
C                                                                             PLTXYZ  0200
      DO 10 I=1,3                                                             PLTXYZ  0210
   10 TMP(I) = P(I) - SEGLP(I,NVEH)                                           PLTXYZ  0220
      CALL MAT31(D(1,1,NVEH),TMP,XYZ)                                         PLTXYZ  0230
C                                                                             PLTXYZ  0240
C     CONVERT XYZ INTO PLOT CORDINATES IX,IY,IZ AND                           PLTXYZ  0250
C     IF WITHIN PLOT LIMITS, STORE C IN PLOTYZ, PLOTXZ AND PLOTXY.            PLTXYZ  0260
C                                                                             PLTXYZ  0270
      IX = SPLT(1)*XYZ(1) + ZPLT(1) + 0.5                                     PLTXYZ  0280
      IZ = SPLT(3)*XYZ(3) + ZPLT(3) + 0.5                                     PLTXYZ  0290
      IF  (IZ.LT.1 .OR. IZ.GT.NPLTZ)  GO TO 11                                PLTXYZ  0300
      IY = SPLT(2)*XYZ(2) + ZPLT(2) + 0.5                                     PLTXYZ  0310
      IF  (IY.GE.1 .AND. IY.LE.NPLTX)  PLOTYZ(IZ,IY) = C                      PLTXYZ  0320
      IF  (IX.GE.1 .AND. IX.LE.NPLTX)  PLOTXZ(IZ,IX) = C                      PLTXYZ  0330
   11 IY = -SPLT(3)*XYZ(2) + ZPLT(2) + 0.5                                    PLTXYZ  0340
      IF  (IY.LT.1 .OR. IY.GT.NPLTZ)  GO TO 99                                PLTXYZ  0350
      IF  (IX.GE.1 .AND. IX.LE.NPLTX)  PLOTXY(IY,IX) = C                      PLTXYZ  0360
   99 RETURN                                                                  PLTXYZ  0370
      END                                                                     PLTXYZ  0380
```

```
      SUBROUTINE POSTPR(PRDT)                                                  POSTPR  0010
C                                                              REV 20 05/18/80  POSTPR  0020
C     CONTROLS GENERATION OF PRINTED TABULAR TIME HISTORIES                    POSTPR  0030
C     AND PLOTS BY THE VALUE OF NPRT(4) AS FOLLOWS:                            POSTPR  0040
C                                                                              POSTPR  0050
C      VALUE OF      TIME                                                      POSTPR  0060
C      NPRT(4)    HISTORIES   PLOTS                                           POSTPR  0070
C                                                                              POSTPR  0080
C        +4           **        NO                                            POSTPR  0090
C        +3          YES       YES                                            POSTPR  0100
C        +2          YES        NO                                            POSTPR  0110
C        +1           **       YES                                            POSTPR  0120
C         0           **        NO                                            POSTPR  0130
C        -1           NO       YES                                            POSTPR  0140
C        -2          YES        NO                                            POSTPR  0150
C        -3          YES       YES                                            POSTPR  0160
C                                                                              POSTPR  0170
C      **  TIME HISTORIES WERE PRINTED BY SUBROUTINE OUTPUT.                   POSTPR  0180
C                                                                              POSTPR  0190
      COMMON/CDINT/  JDTPTS(18) , Z(400,25)                                    POSTPR  0200
C     NOTE: THIS OVERWRITES COMMON /CDINT/.                                    POSTPR  0210
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,                  POSTPR  0220
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)               POSTPR  0230
      REAL*8 TIME                                                              POSTPR  0240
      COMMON/FORCES/ PSF(7,30),BSF(4,20),SSF(10,20),BAGSF(3,20),               POSTPR  0250
     *               PRJNT(7,30),NPANEL(5),NPSF,NBSF,NSSF,NBGSF                POSTPR  0260
      REAL*8 PSF,BSF,SSF,BAGSF,PRJNT                                           POSTPR  0270
      COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),                  POSTPR  0280
     *               BLTTTL(5,8),PLTTL(5,30),BAGTTL(5,6),SEG(30),              POSTPR  0290
     *               JOINT(30),CGS(30),JS(30)                                  POSTPR  0300
      REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT             POSTPR  0310
      LOGICAL*1 CGS,JS                                                         POSTPR  0320
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                                POSTPR  0330
     *               UNITL,UNITM,UNITT,GRAVTY(3)                               POSTPR  0340
      REAL*8 PI,RADIAN,G,THIRD,EPS,UNITL,UNITM,UNITT,GRAVTY                    POSTPR  0350
      COMMON/JBARTZ/ MNPL(   30),MNBLT(    8),MNSEG(   30),MNBAG(    6),     POSTPR  0360
     *               MPL(3,5,30),MBLT(3,5,8),MSEG(3,5,30),MBAG(3,10,6),        POSTPR  0370
     *               NTPL( 5,30),NTBLT( 5,8),NTSEG( 5,30)                      POSTPR  0380
      COMMON/DAMPER/ APSDM(3,20),APSDN(3,20),ASD(5,20),MSDM(20),MSDN(20)       POSTPR  0390
      REAL*8 APSDM,APSDN,ASD                                                   POSTPR  0400
      COMMON/HRNESS/ BAR(15,100),BB(100),BBDOT(100),PLOSS(2,100),              POSTPR  0410
     *               XLONG(20),HTIME(2),IBAR(5,100),NL(2,100),                 POSTPR  0420
     *               NPTSPB(20),NPTPLY(20),NTHRNS(20),NBLTPH(5)                POSTPR  0430
      REAL*8 BAR,BB,BBDOT,PLOSS,XLONG,HTIME                                    POSTPR  0440
      COMMON/RSAVE/  XSG(3,20,3),DPMI(3,3,30),LPMI(30),NSG(7),MSG(20,7)        POSTPR  0450
      REAL*8 XSG,DPMI,TDATA,UMSEC,PRDT,TEST                                    POSTPR  0460
C     NOTE: SUBROUTINES POSTPR & HEDING SHARE THIS COMMON/TEMPVS/.             POSTPR  0470
C     THE FIRST DIMENSION OF XLAB,YLAB,PLB1 AND PLB2 SHOULD BE THE SAME        POSTPR  0480
C     AS THE VALUE ASSIGNED TO NW60 WHICH IS THE NUMBER OF WORDS THAT          POSTPR  0490
C     IS NECESSARY TO CONTAIN 60 CONSECUTIVE CHARACTERS DEPENDING ON THE       POSTPR  0500
```

```
C     COMPUTER SYSTEM THIS PROGRAM IS OPERATING ON. THE VALUE OF NW60      POSTPR  0510
C     SHOULD BE 15 ON IBM 360 AND 370, 10 ON UNIVAC 1108, 6 ON CDC 6600.   POSTPR  0520
C     THE LAST TERM IN FORMAT 13 BELOW SHOULD BE 15A4(IBM), 10A6(UNIVAC)   POSTPR  0530
C     OR 6A10(CDC). ALSO, THE FIRST DIMENSION OF PLDATA IN SUBROUTINE      POSTPR  0540
C     HEDING SHOULD BE 97(IBM), 77(UNIVAC) OR 53(CDC).                     POSTPR  0550
C                                                                          POSTPR  0560
      COMMON/TEMPVS/ TDATA(14,50),HEDATA(220),                             POSTPR  0570
     *               X0(20),XN(20),XL(20),XS(20),XLAB(15,20),PLB1(15,20),  POSTPR  0580
     *               Y0(20),YN(20),YL(20),YS(20),YLAB(15,20),PLB2(15,20),  POSTPR  0590
     *               NYP(20),MX(2,20),MY(2,10,20),NX(20),NY(20),           POSTPR  0600
     *               NXLAB(20),NYLAB(20),NPLB1(20),NPLB2(20),              POSTPR  0610
     *               USEC(45) , ZTTH(14,45,2)                              POSTPR  0620
      LOGICAL LTABH,LPLOT                                                  POSTPR  0630
      DATA  LPP/45/ , NZD1/400/                                            POSTPR  0640
      DATA NW60/15/                                                        POSTPR  0650
      LTABH = .FALSE.                                                      POSTPR  0660
      LPLOT = .FALSE.                                                      POSTPR  0670
      NPRT4 = IABS(NPRT(4))                                                POSTPR  0680
      IF  (NPRT4.EQ.1)  LPLOT = .TRUE.                                     POSTPR  0690
      IF  (NPRT4.EQ.3)  LPLOT = .TRUE.                                     POSTPR  0700
      IF  (NPRT4.EQ.2)  LTABH = .TRUE.                                     POSTPR  0710
      IF  (NPRT4.EQ.3)  LTABH = .TRUE.                                     POSTPR  0720
C                                                                          POSTPR  0730
C     READ INPUT CARD H.8.A TO CONTROL COMPUTATION OF HIC, HSI & CSI.      POSTPR  0740
C                                                                          POSTPR  0750
      READ  (5,11)  JDTPTS                                                 POSTPR  0760
      NDPT = 0                                                             POSTPR  0770
      IF  (JDTPTS(1).NE.0)  NDPT = NDPT + 1                                POSTPR  0780
      IF  (JDTPTS(2).NE.0)  NDPT = NDPT + 1                                POSTPR  0790
      IF  (.NOT.LPLOT .AND. .NOT.LTABH .AND. NDPT.EQ.0) GO TO 99           POSTPR  0800
      CALL ELTIME (1,36)                                                   POSTPR  0810
      IF  (.NOT.LPLOT)  GO TO 20                                           POSTPR  0820
C                                                                          POSTPR  0830
C     READ INDICES OF VARIABLES TO BE PLOTTED AND                          POSTPR  0840
C     ARGUMENTS TO SUBROUTINE SLPLOT ON CARDS I.                           POSTPR  0850
C                                                                          POSTPR  0860
C         INPUT CARD I.1.                                                  POSTPR  0870
C                                                                          POSTPR  0880
      READ  (5,11)  NPLT , (NYP(K),K=1,NPLT)                               POSTPR  0890
   11 FORMAT (18I4)                                                        POSTPR  0900
      IF  (NPLT.LE.0)  LPLOT = .FALSE.                                     POSTPR  0910
      IF  (.NOT.LPLOT)  GO TO 20                                           POSTPR  0920
      DO  15  K=1,NPLT                                                     POSTPR  0930
      NYPLT = NYP(K)                                                       POSTPR  0940
C                                                                          POSTPR  0950
C         INPUT CARD I.2.K                                                 POSTPR  0960
C                                                                          POSTPR  0970
      READ  (5,11)  MX(1,K), MX(2,K), (MY(1,J,K), MY(2,J,K), J=1,NYPLT)    POSTPR  0980
C                                                                          POSTPR  0990
C         INPUT CARD I.3.K                                                 POSTPR  1000
```

```
C                                                                       POSTPR  1010
      READ  (5,12)  NX(K), X0(K), XN(K), XL(K), XS(K)                   POSTPR  1020
   12 FORMAT (I4 , 4X , 4F8.0 )                                         POSTPR  1030
C                                                                       POSTPR  1040
C         INPUT CARD I.4.K                                              POSTPR  1050
C                                                                       POSTPR  1060
      READ  (5,12)  NY(K), Y0(K), YN(K), YL(K), YS(K)                   POSTPR  1070
C                                                                       POSTPR  1080
C         INPUT CARD I.5.K                                              POSTPR  1090
C                                                                       POSTPR  1100
      READ  (5,13)  NXLAB(K), (XLAB(I,K),I=1,NW60)                      POSTPR  1110
   13 FORMAT (I4 , 4X , 15A4)                                           POSTPR  1120
C                                                                       POSTPR  1130
C     NOTE - ABOVE FORMAT ASSUMES 4 ALPHANUMERIC CHARACTERS FOR SINGLE  POSTPR  1140
C     PRECISION WORDS ON IBM 360 AND 370 COMPUTERS. THE 15A4 TERM IN THE POSTPR  1150
C     FORMAT WILL HAVE TO BE CHANGED ON NON-IBM COMPUTERS TO PRODUCE A  POSTPR  1160
C     CONTINUOUS STRING OF 60 CHARACTERS IN CORE MEMORY.                POSTPR  1170
C                                                                       POSTPR  1180
C         INPUT CARD I.6.K                                              POSTPR  1190
C                                                                       POSTPR  1200
      READ  (5,13)  NYLAB(K), (YLAB(I,K),I=1,NW60)                      POSTPR  1210
C                                                                       POSTPR  1220
C         INPUT CARD I.7.K                                              POSTPR  1230
C                                                                       POSTPR  1240
      READ  (5,13)  NPLB1(K), (PLB1(I,K),I=1,NW60)                      POSTPR  1250
C                                                                       POSTPR  1260
C         INPUT CARD I.8.K                                              POSTPR  1270
C                                                                       POSTPR  1280
   15 READ  (5,13)  NPLB2(K), (PLB2(I,K),I=1,NW60)                      POSTPR  1290
C                                                                       POSTPR  1300
C     READ TIME HISTORY DATA FROM TAPE 8.                               POSTPR  1310
C                                                                       POSTPR  1320
   20 NPTS = 0                                                          POSTPR  1330
      LINES = 0                                                         POSTPR  1340
      IF  (NPRT(4).GT.0)  REWIND 8                                      POSTPR  1350
      READ  (8,END=29)  NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,NPANEL,      POSTPR  1360
     *                  MNPL,MNBLT,MNSEG,MNBAG,MPL,MBLT,MSEG,MBAG       POSTPR  1370
      READ  (8,END=29)  DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,  POSTPR  1380
     *                  SEG,JOINT,UNITL,UNITM,UNITT,NSG,MSG,XSG,        POSTPR  1390
     *                  NHRNSS,NBLTPH,NPTSPB,NSD,MSDM,MSDN              POSTPR  1400
   21 READ  (8,END=29)  NT, UMSEC,((TDATA(I,J),I=1,14),J=1,NT)          POSTPR  1410
      NPTS = NPTS + 1                                                   POSTPR  1420
      Z(NPTS,1) = UMSEC                                                 POSTPR  1430
      IF  (NDPT.EQ.0)  GO TO 22                                         POSTPR  1440
C                                                                       POSTPR  1450
C     STORE DATA FOR HIC, HSI AND CSI.                                  POSTPR  1460
C                                                                       POSTPR  1470
      JJ = 1                                                            POSTPR  1480
      DO  61  I=1,2                                                     POSTPR  1490
      IF  (JDTPTS(I).EQ.0)  GO TO 61                                    POSTPR  1500
```

```
      JJ = JJ +1                                                          POSTPR  1510
      JD = JDTPTS(I) - 1                                                  POSTPR  1520
      JE = 4*MOD(JD,3) + 4                                                POSTPR  1530
      JP = JD/3 + 1                                                       POSTPR  1540
      Z(NPTS,JJ) = TDATA(JE,JP)                                           POSTPR  1550
   61 CONTINUE                                                            POSTPR  1560
   22 IF  (.NOT.LPLOT)  GO TO 25                                          POSTPR  1570
C                                                                         POSTPR  1580
C     STORE DATA FOR PLOTTING                                             POSTPR  1590
C                                                                         POSTPR  1600
      JY = NDPT + 1                                                       POSTPR  1610
      DO  24  K=1,NPLT                                                    POSTPR  1620
      JE = IABS(MX(2,K))                                                  POSTPR  1630
      IF  (JE.EQ.0)  GO TO 23                                             POSTPR  1640
      JY = JY + 1                                                         POSTPR  1650
      JP = MX(1,K) - 20                                                   POSTPR  1660
      Z(NPTS,JY) = TDATA(JE,JP)                                           POSTPR  1670
   23 NYPLT = NYP(K)                                                      POSTPR  1680
      DO  24  J=1,NYPLT                                                   POSTPR  1690
      JY = JY + 1                                                         POSTPR  1700
      JP = MY(1,J,K) - 20                                                 POSTPR  1710
      JE = IABS(MY(2,J,K))                                                POSTPR  1720
      Z(NPTS,JY) = UMSEC                                                  POSTPR  1730
   24 IF  (JE.NE.0)  Z(NPTS,JY) = TDATA(JE,JP)                            POSTPR  1740
   25 IF  (.NOT.LTABH)  GO TO 21                                          POSTPR  1750
C                                                                         POSTPR  1760
C     STORE DATA TO PRINT TABULAR TIME HISTORIES                          POSTPR  1770
C                                                                         POSTPR  1780
      TEST = DMOD(UMSEC,PRDT)                                             POSTPR  1790
      TEST = DMIN1(TEST,DABS(PRDT-TEST))                                  POSTPR  1800
      IF  (NPRT(26).EQ.0 .AND. TEST.GT.EPS(4))  GO TO 21                  POSTPR  1810
      LINES = LINES + 1                                                   POSTPR  1820
      NTTH = MOD(LINES-1,LPP) + 1                                         POSTPR  1830
      USEC(NTTH) = UMSEC                                                  POSTPR  1840
      DO  26  J=1,NT                                                      POSTPR  1850
      DO  26  I=1,14                                                      POSTPR  1860
   26 ZTTH(I,NTTH,J) = TDATA(I,J)                                         POSTPR  1870
      IF  (NTTH.EQ.LPP)  CALL HEDING (LINES,LPP)                          POSTPR  1880
      GO TO 21                                                            POSTPR  1890
   29 IF  (.NOT.LTABH .OR. LINES.EQ.0)  GO TO 30                          POSTPR  1900
      IF  (NTTH.NE.LPP)  CALL HEDING (LINES,LPP)                          POSTPR  1910
   30 IF  (NDPT.NE.0)  CALL HICCSI(NPTS)                                  POSTPR  1920
      IF  (.NOT.LPLOT)  GO TO 98                                          POSTPR  1930
C                                                                         POSTPR  1940
C     PLOT DATA VIA SUBROUTINE SLPLOT.                                    POSTPR  1950
C                                                                         POSTPR  1960
C     INCLUDE ANY PROGRAM STATEMENTS HERE REQUIRED BY YOUR COMPUTER AND   POSTPR  1970
C     PLOTTING SYSTEMS FOR PLOT INITIALIZATION (E.G., CALL PLOTS).        POSTPR  1980
C                                                                         POSTPR  1990
      JZ = NDPT+1                                                         POSTPR  2000
```

```
      DO  50  K=1,NPLT                                                          POSTPR  2010
      JX = 1                                                                    POSTPR  2020
      IF  (MX(2,K).EQ.0)  GO TO 42                                              POSTPR  2030
      JZ = JZ + 1                                                               POSTPR  2040
      JX = JZ                                                                   POSTPR  2050
      IF  (Z(1,JX).EQ.0.0 .OR. MX(2,K).GE.0)  GO TO 42                          POSTPR  2060
      DO  41  I=2,NPTS                                                          POSTPR  2070
   41 Z(I,JX) = Z(I,JX) - Z(1,JX)                                               POSTPR  2080
      Z(1,JX) = 0.0                                                             POSTPR  2090
   42 NYPLT = NYP(K)                                                            POSTPR  2100
      DO  44  J=1,NYPLT                                                         POSTPR  2110
      JY = JZ + J                                                               POSTPR  2120
      IF  (Z(1,JY).EQ.0.0 .OR. MY(2,J,K).GE.0)  GO TO 44                        POSTPR  2130
      DO  43  I=2,NPTS                                                          POSTPR  2140
   43 Z(I,JY) = Z(I,JY) - Z(1,JY)                                               POSTPR  2150
      Z(1,JY) = 0.0                                                             POSTPR  2160
   44 CONTINUE                                                                  POSTPR  2170
      NXK = NX(K)                                                               POSTPR  2180
      NYK = NY(K)                                                               POSTPR  2190
      XOK = XO(K)                                                               POSTPR  2200
      YOK = YO(K)                                                               POSTPR  2210
      XNK = XN(K)                                                               POSTPR  2220
      YNK = YN(K)                                                               POSTPR  2230
      XLK = XL(K)                                                               POSTPR  2240
      YLK = YL(K)                                                               POSTPR  2250
      XSK = XS(K)                                                               POSTPR  2260
      YSK = YS(K)                                                               POSTPR  2270
      NXLABK = NXLAB(K)                                                         POSTPR  2280
      NYLABK = NYLAB(K)                                                         POSTPR  2290
      NPLB1K = NPLB1(K)                                                         POSTPR  2300
      NPLB2K = NPLB2(K)                                                         POSTPR  2310
      CALL SLPLOT(Z(1,JX  ), NXK, XOK, XNK, XLK, XSK, XLAB(1,K), NXLABK,        POSTPR  2320
     *            Z(1,JZ+1), NYK, YOK, YNK, YLK, YSK, YLAB(1,K), NYLABK,        POSTPR  2330
     *             NPTS,NYPLT,NZD1,PLB1(1,K),NPLB1K,PLB2(1,K),NPLB2K)           POSTPR  2340
C                                                                               POSTPR  2350
C     INSERT ANY CODE REQUIRED BY YOUR SYSTEM TO ADVANCE PLOT PAGES HERE        POSTPR  2360
C                                                                               POSTPR  2370
   50 JZ = JZ + NYPLT                                                           POSTPR  2380
C                                                                               POSTPR  2390
C     INSERT ANY PLOT TERMINATION CODE REQUIRED BY YOUR SYSTEM HERE.            POSTPR  2400
C                                                                               POSTPR  2410
      CALL PLOT (0.0,0.0,999)                                                   POSTPR  2420
   98 CALL ELTIME (2,36)                                                        POSTPR  2430
   99 RETURN                                                                    POSTPR  2440
      END                                                                       POSTPR  2450
```

```
      SUBROUTINE PRINT(SUB)                                                  PRINT   0010
C                                                          REV 20 04/11/80    PRINT   0020
C     SUBROUTINE TO PRINT SEGMENT LINEAR AND ANGULAR                          PRINT   0030
C     POSITIONS, VELOCITIES AND ACCELERATIONS FOR A GIVEN TIME.               PRINT   0040
C                                                                             PRINT   0050
C     ARGUMENTS                                                               PRINT   0060
C        SUB: CALLING SUBROUTINE NAME                                         PRINT   0070
C                                                                             PRINT   0080
      IMPLICIT REAL*8(A-H,O-Z)                                                PRINT   0090
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,                 PRINT   0100
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)              PRINT   0110
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),      PRINT   0120
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)             PRINT   0130
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(3,60),HA(3,60),HB(3,60),       PRINT   0140
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),           PRINT   0150
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)          PRINT   0160
      COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60),      PRINT   0170
     *               F(3,30),TQ(3,30),WJ(30)                                  PRINT   0180
      COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),         PRINT   0190
     *               HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12),      PRINT   0200
     *               RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),                    PRINT   0210
     *               KQ1(12),KQ2(12),KQTYPE(12)                               PRINT   0220
      COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),                 PRINT   0230
     *               BLTTTL(5,8),PLTTL(5,30),BAGTTL(5,6),SEG(30),             PRINT   0240
     *               JOINT(30),CGS(30),JS(30)                                 PRINT   0250
      REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT            PRINT   0260
      LOGICAL*1 CGS,JS                                                        PRINT   0270
      COMMON/CEULER/ IEULER(30),HIR(3,3,30),ANG(3,30),ANGD(3,30),             PRINT   0280
     *               FE(3,30),TQE(3,30),CONST(3,30)                           PRINT   0290
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                               PRINT   0300
     *               UNITL,UNITM,UNITT,GRAVTY(3)                              PRINT   0310
      COMMON/RSAVE/  XSG(3,20,3),DPMI(3,3,30),LPMI(30),NSG(7),MSG(20,7)       PRINT   0320
      COMMON/TEMPVS/ YPR(3),T1(3),T2(3),HH(3),T3(3,3)                         PRINT   0330
C                                                                             PRINT   0340
      IPC = 1                                                                 PRINT   0350
      TMSEC = 1000.0*TIME                                                     PRINT   0360
      WRITE (6,11) IPC,SUB,TMSEC                                              PRINT   0370
   11 FORMAT (I1,6X,A6,' FUNCTIONS IN INERTIAL REFERENCE FOR TIME=',          PRINT   0380
     *        F10.3,' MSEC'//)                                               PRINT   0390
      WRITE (6,12) UNITT,UNITT                                                PRINT   0400
   12 FORMAT(19X,'ANGULAR ROTATION (DEG)',                                    PRINT   0410
     *       12X,'ANGULAR VELOCITY (RAD/',A4,')',                             PRINT   0420
     *       12X,'ANGULAR ACCELERATION (RAD/',A4,'**2)'/                      PRINT   0430
     *       ' SEGMENT',9X,'YAW' ,7X,'PITCH',7X,'ROLL',                       PRINT   0440
     *       11X,'X',11X,'Y',11X,'Z',15X,'X',13X,'Y',13X,'Z'/)                PRINT   0450
      MBAG = NVEH + NBAG                                                      PRINT   0460
      DO  20  I=1,MBAG                                                        PRINT   0470
      IF  (LPMI(I).EQ.0)  GO TO 19                                            PRINT   0480
      CALL DOT33 (DPMI(1,1,I),D(1,1,I),T3)                                    PRINT   0490
      CALL YPRDEG (T3,YPR)                                                    PRINT   0500
```

```
      GO TO 20                                                          PRINT   0510
   19 CALL YPRDEG (D(1,1,I),YPR)                                        PRINT   0520
   20 WRITE  (6,31)  I,SEG(I),YPR,(WMEG(K,I),K=1,3),(WMEGD(K,I),K=1,3)  PRINT   0530
      WRITE  (6,22)  UNITL,UNITL,UNITT                                  PRINT   0540
   22 FORMAT(///18X,'LINEAR POSITION (',A4,')',                         PRINT   0550
     *          13X,'LINEAR VELOCITY (',A4,'/',A4,')',                 PRINT   0560
     *          16X,'LINEAR ACCELERATIONS (G''S)'/                     PRINT   0570
     *       ' SEGMENT',10X,'X',10X,'Y',10X,'Z',                       PRINT   0580
     *       13X,'X',11X,'Y',11X,'Z',15X,'X',13X,'Y',13X,'Z'/)         PRINT   0590
      DO  30  I=1,MBAG                                                  PRINT   0600
      DO  29  K=1,3                                                     PRINT   0610
   29 T1(K) = SEGLA(K,I)/G                                              PRINT   0620
   30 WRITE (6,31) I,SEG(I),(SEGLP(K,I),K=1,3),(SEGLV(K,I),K=1,3),T1    PRINT   0630
   31 FORMAT(I3,1X,A4,3X,3F11.4,3X,3F12.5,3X,3F14.6)                    PRINT   0640
      IF (NSEG.GT.6) WRITE (6,32)                                       PRINT   0650
   32 FORMAT('1')                                                       PRINT   0660
      WRITE (6,33) UNITL,UNITT,UNITT                                    PRINT   0670
   33 FORMAT(///18X,'U1 ARRAY (',A4,'/',A4,'**2)',                      PRINT   0680
     *          14X,'U2 ARRAY (RAD/',A4,'**2)'/                        PRINT   0690
     *          15X,'EXTERNAL LINEAR ACCELERATIONS',                   PRINT   0700
     *           8X,'EXTERNAL ANGULAR ACCELERATIONS'/                  PRINT   0710
     * ' SEGMENT',10X,'X',10X,'Y',10X,'Z',13X,'X',11X,'Y',11X,'Z'/)     PRINT   0720
      DO 34 I=1,NSEG                                                    PRINT   0730
   34 WRITE (6,31) I,SEG(I),(U1(K,I),K=1,3),(U2(K,I),K=1,3)             PRINT   0740
      IF (NJNT.LE.0) GO TO 39                                           PRINT   0750
      WRITE (6,35) UNITM,UNITL,UNITM,UNITT                              PRINT   0760
   35 FORMAT(///24X,'JOINT FORCES (',A4,')',                            PRINT   0770
     *          15X,'JOINT TORQUES (',2A4,')',                         PRINT   0780
     *           9X,'RELATIVE ANGULAR'/                                PRINT   0790
     *' JOINT  IPIN',9X,'X',10X,'Y',10X,'Z',13X,'X',11X,'Y',11X,'Z',    PRINT   0800
     *           7X,'VELOCITY (RAD/',A4,')'/)                          PRINT   0810
      DO 36 J=1,NJNT                                                    PRINT   0820
      IPINJ = IPIN(J)                                                   PRINT   0830
      IF (IABS(IPIN(J)).EQ.4) IPINJ = IEULER(J)                         PRINT   0840
   36 WRITE (6,37) J,JOINT(J),IPINJ,(F(K,J),K=1,3),(TQ(K,J),K=1,3),WJ(J) PRINT   0850
   37 FORMAT(I3,1X,A4,I4,3X,3F11.4,3X,3F12.5,F17.6)                     PRINT   0860
   39 IF (NQ.LE.0) GO TO 99                                             PRINT   0870
      WRITE (6,41) UNITM,UNITL                                          PRINT   0880
   41 FORMAT(///' OTHER CONSTRAINT FORCES'//                            PRINT   0890
     *       '  NO.  TYPE   SEG1  SEG2',                               PRINT   0900
     *       15X,'CONSTRAINT FORCE (',A4,')',                           PRINT   0910
     *       16X,'DISTANCE (',A4,')'/)                                  PRINT   0920
      DO 50 J=1,NQ                                                      PRINT   0930
      IF (KQTYPE(J).LT.0) GO TO 50                                      PRINT   0940
      M = KQ1(J)                                                        PRINT   0950
      N = KQ2(J)                                                        PRINT   0960
      CALL DOT31(D(1,1,M),RK1(1,J),T1)                                  PRINT   0970
      CALL DOT31(D(1,1,N),RK2(1,J),T2)                                  PRINT   0980
      S1 = 0.0                                                          PRINT   0990
      DO 42 I=1,3                                                       PRINT   1000
      HH(I) = SEGLP(I,M)+T1(I) - SEGLP(I,N)-T2(I)                       PRINT   1010
   42 S1 = S1 + HH(I)**2                                                PRINT   1020
      SQS1 = DSQRT(S1)                                                  PRINT   1030
      WRITE (6,43) J,KQTYPE(J),SEG(M),SEG(N),(QQ(I,J),I=1,3),SQS1       PRINT   1040
   43 FORMAT(I4,I6,4X,A4,2X,A4,3X,3G15.7,6X,G15.7)                      PRINT   1050
   50 CONTINUE                                                          PRINT   1060
   99 IF  (NPRT(28).LE.0)  NPRT(28) = -1                                PRINT   1070
      RETURN                                                            PRINT   1080
      END                                                               PRINT   1090
```

```
      SUBROUTINE PRIPLT                                                 PRIPLT  0010
C                                                          REV 20 05/05/80  PRIPLT  0020
C     PRODUCES PRINTER PLOT OF Y-Z PLANE VIEW AND X-Z PLANE VIEW OF       PRIPLT  0030
C     BODY SEGMENT CGS, JOINTS AND SELECTED POINTS OF VEHICLE COMPONENTS  PRIPLT  0040
C                                                                         PRIPLT  0050
      IMPLICIT REAL*8 (A-H,O-Z)                                         PRIPLT  0060
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           PRIPLT  0070
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)        PRIPLT  0080
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30), PRIPLT  0090
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       PRIPLT  0100
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(3,60),HA(3,60),HB(3,60), PRIPLT  0110
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),     PRIPLT  0120
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)    PRIPLT  0130
      COMMON/JBARTZ/ MNPL(   30),MNBLT(    8),MNSEG(   30),MNBAG(    6), PRIPLT  0140
     *               MPL(3,5,30),MBLT(3,5,8),MSEG(3,5,30),MBAG(3,10,6), PRIPLT  0150
     *               NTPL( 5,30),NTBLT( 5,8),NTSEG( 5,30)               PRIPLT  0160
      COMMON/CNTSRF/ PL(17,30),BELT(20,8),TPTS(6,8),BD(24,40)           PRIPLT  0170
      COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),           PRIPLT  0180
     *               BLTTTL(5,8),PLTTL(5,30),BAGTTL(5,6),SEG(30),       PRIPLT  0190
     *               JOINT(30),CGS(30),JS(30)                           PRIPLT  0200
      COMMON/HRNESS/ BAR(15,100),BB(100),BBDOT(100),PLOSS(2,100),       PRIPLT  0210
     *               XLONG(20),HTIME(2),IBAR(5,100),NL(2,100),          PRIPLT  0220
     *               NPTSPB(20),NPTPLY(20),NTHRNS(20),NBLTPH(5)         PRIPLT  0230
      REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT      PRIPLT  0240
      LOGICAL*1 CGS,JS                                                  PRIPLT  0250
      COMMON/TEMPVS/ TEMP1(3),TEMP(3),TEMP2(3),CJOINT(3,30),BSN(2),     PRIPLT  0260
     *               PLOTYZ(96,55),PLOTXZ(96,55),PLOTXY(96,55)          PRIPLT  0270
      LOGICAL*1 PLOTYZ,PLOTXZ,PLOTXY,CHARS(7),BLANK,BCHAR               PRIPLT  0280
      LOGICAL   NPRT5,NPRT6,NPRT7                                       PRIPLT  0290
      DATA CHARS/'.','+','@','I','_','-','*'/, BLANK/' '/               PRIPLT  0300
      DATA ISTEP/0/ , NPLTI/96/ , NPLTJ/55/                             PRIPLT  0310
C                                                                       PRIPLT  0320
C     DETERMINE IF PLOTTING IS TO BE DONE FOR THIS TIME STEP.           PRIPLT  0330
C                                                                       PRIPLT  0340
      ISTEP = ISTEP+1                                                   PRIPLT  0350
      NPRT5 = (NPRT(5).EQ.1)                                            PRIPLT  0360
      IF (NPRT(5).GT.1) NPRT5 = (MOD(ISTEP,NPRT(5)).EQ.1)               PRIPLT  0370
      NPRT6 = (NPRT(6).EQ.1)                                            PRIPLT  0380
      IF (NPRT(6).GT.1) NPRT6 = (MOD(ISTEP,NPRT(6)).EQ.1)               PRIPLT  0390
      NPRT7 = (NPRT(7).EQ.1)                                            PRIPLT  0400
      IF (NPRT(7).GT.1) NPRT7 = (MOD(ISTEP,NPRT(7)).EQ.1)               PRIPLT  0410
      IF  (.NOT.NPRT5 .AND. .NOT.NPRT6 .AND. .NOT.NPRT7)  GO TO 99      PRIPLT  0420
      CALL ELTIME(1, 4)                                                 PRIPLT  0430
C                                                                       PRIPLT  0440
C     BLANK OUT PLOT ARRAYS.                                            PRIPLT  0450
C                                                                       PRIPLT  0460
      DO  10  J=1,NPLTJ                                                 PRIPLT  0470
      PLOTYZ(1,J) = CHARS(6)                                            PRIPLT  0480
      PLOTXZ(1,J) = CHARS(6)                                            PRIPLT  0490
      PLOTXY(1,J) = CHARS(6)                                            PRIPLT  0500
```

```
      DO  10  I=2,NPLTI                                                  PRIPLT  0510
      PLOTYZ(I,J) = BLANK                                                PRIPLT  0520
      PLOTXZ(I,J) = BLANK                                                PRIPLT  0530
   10 PLOTXY(I,J) = BLANK                                                PRIPLT  0540
C                                                                        PRIPLT  0550
C     PLOT VEHICLE REFERENCE ORIGIN USING SYMBOL(*).                     PRIPLT  0560
C                                                                        PRIPLT  0570
      CALL PLTXYZ (SEGLP(1,NVEH),CHARS(7))                               PRIPLT  0580
C                                                                        PRIPLT  0590
C     PLOT CG OF BODY SEGMENTS USING SEGMENT SYMBOLS.                    PRIPLT  0600
C                                                                        PRIPLT  0610
      DO 20  I=1,NSEG                                                   PRIPLT  0620
   20 CALL PLTXYZ(SEGLP(1,I),CGS(I))                                     PRIPLT  0630
C                                                                        PRIPLT  0640
C     COMPUTE AND PLOT JOINT LOCATIONS USING JOINT SYMBOLS.              PRIPLT  0650
C                                                                        PRIPLT  0660
      IF  (NJNT.EQ.0)  GO TO 40                                          PRIPLT  0670
      DO  31  J=1,NJNT                                                   PRIPLT  0680
      I = IABS(JNT(J))                                                   PRIPLT  0690
      IF  (I.LE.0)  GO TO 31                                             PRIPLT  0700
      CALL DOT31(D(1,1,I),SR(1,2*J-1),TEMP)                              PRIPLT  0710
      DO 30 L=1,3                                                        PRIPLT  0720
   30 CJOINT(L,J) = TEMP(L)+SEGLP(L,I)                                   PRIPLT  0730
      CALL PLTXYZ (CJOINT(1,J),JS(J))                                    PRIPLT  0740
   31 CONTINUE                                                           PRIPLT  0750
      IF  (NPRT(13).NE.0)  WRITE (6,32) ((CJOINT(I,J),I=1,3),J=1,NJNT)   PRIPLT  0760
   32 FORMAT ('0 JOINT POSITIONS'/(1X,9F14.4))                           PRIPLT  0770
C                                                                        PRIPLT  0780
C     PLOT BELT ANCHOR, FIXED AND TANGENT POINTS USING SYMBOL(.).        PRIPLT  0790
C                                                                        PRIPLT  0800
   40 IF  (NBLT.LE.0)  GO TO 50                                          PRIPLT  0810
      DO  43  J=1,NBLT                                                   PRIPLT  0820
      IF  (MNBLT(J).LE.0)  GO TO 43                                      PRIPLT  0830
      M1 = MBLT(1,1,J)                                                   PRIPLT  0840
      M2 = MBLT(2,1,J)                                                   PRIPLT  0850
      M3 = MBLT(3,1,J)                                                   PRIPLT  0860
      DO  41  I=1,3                                                      PRIPLT  0870
   41 TEMP1(I) = BELT(I+6,J) + BD(I+3,M3)                                PRIPLT  0880
      CALL DOT31 (D(1,1,M2),TEMP1,TEMP)                                  PRIPLT  0890
      CALL DOT31 (D(1,1,M1),BELT(1,J),TEMP1)                             PRIPLT  0900
      CALL DOT31 (D(1,1,M1),BELT(4,J),TEMP2)                             PRIPLT  0910
      DO  42  I=1,3                                                      PRIPLT  0920
      TEMP1(I) = TEMP1(I) + SEGLP(I,M1)                                  PRIPLT  0930
      TEMP2(I) = TEMP2(I) + SEGLP(I,M1)                                  PRIPLT  0940
   42 TEMP (I) = TEMP (I) + SEGLP(I,M2)                                  PRIPLT  0950
      CALL PLTXYZ (TEMP1    ,CHARS(1))                                   PRIPLT  0960
      CALL PLTXYZ (TEMP2    ,CHARS(1))                                   PRIPLT  0970
      CALL PLTXYZ (TEMP     ,CHARS(1))                                   PRIPLT  0980
      CALL PLTXYZ (TPTS(1,J),CHARS(1))                                   PRIPLT  0990
      CALL PLTXYZ (TPTS(4,J),CHARS(1))                                   PRIPLT  1000
```

```
   43 CONTINUE                                                          PRIPLT  1010
C                                                                       PRIPLT  1020
C     PLOT POINTS IN PLAY ON HARNESS-BELT SYSTEMS USING SYMBOL(.).      PRIPLT  1030
C                                                                       PRIPLT  1040
   50 IF  (NHRNSS.LE.0)  GO TO 60                                       PRIPLT  1050
      J1 = 1                                                            PRIPLT  1060
      K1 = 1                                                            PRIPLT  1070
      DO  54  NH=1,NHRNSS                                               PRIPLT  1080
      IF  (NBLTPH(NH).LE.0)  GO TO 54                                   PRIPLT  1090
      J2 = J1 + NBLTPH(NH) - 1                                          PRIPLT  1100
      DO  53  NB=J1,J2                                                  PRIPLT  1110
      IF  (NPTPLY(NB).LE.0)  GO TO 53                                   PRIPLT  1120
      K2 = K1 + NPTPLY(NB) - 1                                          PRIPLT  1130
      DO  52   K=K1,K2                                                  PRIPLT  1140
      KI = NL(1,K)                                                      PRIPLT  1150
      KS = IABS(IBAR(1,KI))                                             PRIPLT  1160
      IF  (KS.GT.100)  KS = MOD(KS,100)                                 PRIPLT  1170
      CALL DOT31 (D(1,1,KS),BAR(4,KI),TEMP1)                           PRIPLT  1180
      CALL DOT31 (D(1,1,KS),BAR(7,KI),TEMP2)                           PRIPLT  1190
      DO  51  I=1,3                                                     PRIPLT  1200
   51 TEMP(I) = SEGLP(I,KS) + TEMP1(I) + TEMP2(I)                       PRIPLT  1210
   52 CALL PLTXYZ (TEMP,CHARS(1))                                       PRIPLT  1220
      K1 = K2+1                                                         PRIPLT  1230
   53 CONTINUE                                                          PRIPLT  1240
      J1 = J2+1                                                         PRIPLT  1250
   54 CONTINUE                                                          PRIPLT  1260
C                                                                       PRIPLT  1270
C     PLOT CENTER AND END OF AXES OF ELLIPSOIDAL TARGET USING SYMBOLS   PRIPLT  1280
C     (@) FOR CENTER, (-) FOR ENDS OF Z AXIS,(I) FOR ENDS OF X,Y AXES.  PRIPLT  1290
C                                                                       PRIPLT  1300
   60 IF  (NBAG.EQ.0)  GO TO 80                                         PRIPLT  1310
      BSN(1) =  1.0                                                     PRIPLT  1320
      BSN(2) = -1.0                                                     PRIPLT  1330
      DO  68  J=1,NBAG                                                  PRIPLT  1340
      IF  (MNBAG(J).EQ.0)  GO TO 68                                     PRIPLT  1350
      JB = NVEH+J                                                       PRIPLT  1360
      BCHAR = CHARS(5)                                                  PRIPLT  1370
      L2 = 2                                                            PRIPLT  1380
      DO  67  I=1,4                                                     PRIPLT  1390
      IF  (I.EQ.3)  BCHAR = CHARS(4)                                    PRIPLT  1400
      IF  (I.EQ.4)  BCHAR = CHARS(3)                                    PRIPLT  1410
      IF  (I.EQ.4)  L2 = 1                                              PRIPLT  1420
      DO  67  L=1,L2                                                    PRIPLT  1430
      DO  64  K=1,3                                                     PRIPLT  1440
   64 TEMP1(K) = BD(K+3,JB)                                             PRIPLT  1450
      IF  (I.EQ.4)  GO TO 65                                            PRIPLT  1460
      TEMP1(I) = TEMP1(I) + BSN(L)*BD(I,JB)                             PRIPLT  1470
   65 CALL DOT31 (D(1,1,JB),TEMP1,TEMP2)                               PRIPLT  1480
      DO  66  K=1,3                                                     PRIPLT  1490
   66 TEMP2(K) = TEMP2(K) + SEGLP(K,JB)                                 PRIPLT  1500
```

```
   67 CALL PLTXYZ (TEMP2,BCHAR)                                                  PRIPLT  1510
   68 CONTINUE                                                                   PRIPLT  1520
C                                                                                PRIPLT  1530
C     PRINT Y-Z , X-Z AND X-Y PLANE VIEW PLOTS.                                  PRIPLT  1540
C                                                                                PRIPLT  1550
   80 TMSC = 1000.0*TIME                                                         PRIPLT  1560
      IF  (.NOT.NPRT5)  GO TO 83                                                 PRIPLT  1570
      WRITE  (2,81)  TMSC,SEGLP(2,NVEH),SEGLP(3,NVEH)                           PRIPLT  1580
   81 FORMAT ('1 T=',F10.3,'    Y0=',F10.5,'    Z0=',F10.5,'    Y-Z PLANE')    PRIPLT  1590
      WRITE  (2,82)  PLOTYZ                                                      PRIPLT  1600
   82 FORMAT (2X,96A1)                                                           PRIPLT  1610
   83 IF  (.NOT.NPRT6)  GO TO 85                                                 PRIPLT  1620
      WRITE  (2,84)  TMSC,SEGLP(1,NVEH),SEGLP(3,NVEH)                           PRIPLT  1630
   84 FORMAT ('1 T=',F10.3,'    X0=',F10.5,'    Z0=',F10.5,'    X-Z PLANE')    PRIPLT  1640
      WRITE  (2,82)  PLOTXZ                                                      PRIPLT  1650
   85 IF  (.NOT.NPRT7)  GO TO 87                                                 PRIPLT  1660
      WRITE  (2,86)  TMSC,SEGLP(1,NVEH),SEGLP(2,NVEH)                           PRIPLT  1670
   86 FORMAT ('1 T=',F10.3,'    X0=',F10.5,'    Y0=',F10.5,'    X-Y PLANE')    PRIPLT  1680
      WRITE  (2,82)  PLOTXY                                                      PRIPLT  1690
   87 CALL ELTIME(2, 4)                                                          PRIPLT  1700
   99 RETURN                                                                     PRIPLT  1710
      END                                                                        PRIPLT  1720
```

```
      SUBROUTINE QSET(F,Y,X,DER,N)                                          QSET     0010
C                                                        REV 16 03/24/76    QSET     0020
      IMPLICIT REAL*8(A-H,O-Z)                                              QSET     0030
      DIMENSION F(5,3,80),Y(5,3,80),X(3,80),DER(3,80)                       QSET     0040
      DIMENSION T1(3),T2(3),T3(3),T4(3)                                     QSET     0050
      DO  20  I=1,N                                                         QSET     0060
      E1=DSQRT(1.D0 -X(1,I)**2-X(2,I)**2-X(3,I)**2)                         QSET     0070
      E1D=-(X(1,I)*DER(1,I)+X(2,I)*DER(2,I)+X(3,I)*DER(3,I))/E1             QSET     0080
      E2=DSQRT(1.D0-Y(1,1,I)**2-Y(1,2,I)**2-Y(1,3,I)**2)                    QSET     0090
      E2D=-(Y(1,1,I)*Y(2,1,I)+Y(1,2,I)*Y(2,2,I)+Y(1,3,I)*Y(2,3,I))/E2       QSET     0100
      UHB=X(1,I)*F(3,1,I)+X(2,I)*F(3,2,I)+X(3,I)*F(3,3,I)                   QSET     0110
      UHC=X(1,I)*F(4,1,I)+X(2,I)*F(4,2,I)+X(3,I)*F(4,3,I)                   QSET     0120
      UDB=DER(1,I)*F(3,1,I)+DER(2,I)*F(3,2,I)+DER(3,I)*F(3,3,I)             QSET     0130
      UDD=DER(1,I)**2+DER(2,I)**2+DER(3,I)**2                               QSET     0140
      EB=(E1D**2+UDD+UHB)/E1                                                QSET     0150
      EC =  (1.5*(UDB-E1D*EB)+UHC+F(5,1,I)*(E1D**2+UDD))/E1                 QSET     0160
      T1(1)=X(2,I)*F(3,3,I)-X(3,I)*F(3,2,I)                                 QSET     0170
      T2(1)=X(2,I)*F(4,3,I)-X(3,I)*F(4,2,I)                                 QSET     0180
      T3(1)=X(2,I)*Y(1,3,I)-X(3,I)*Y(1,2,I)                                 QSET     0190
      T4(1)=X(2,I)*Y(2,3,I)-X(3,I)*Y(2,2,I)                                 QSET     0200
      T1(2)=X(3,I)*F(3,1,I)-X(1,I)*F(3,3,I)                                 QSET     0210
      T2(2)=X(3,I)*F(4,1,I)-X(1,I)*F(4,3,I)                                 QSET     0220
      T3(2)=X(3,I)*Y(1,1,I)-X(1,I)*Y(1,3,I)                                 QSET     0230
      T4(2)=X(3,I)*Y(2,1,I)-X(1,I)*Y(2,3,I)                                 QSET     0240
      T1(3)=X(1,I)*F(3,2,I)-X(2,I)*F(3,1,I)                                 QSET     0250
      T2(3)=X(1,I)*F(4,2,I)-X(2,I)*F(4,1,I)                                 QSET     0260
      T3(3)=X(1,I)*Y(1,2,I)-X(2,I)*Y(1,1,I)                                 QSET     0270
      T4(3)=X(1,I)*Y(2,2,I)-X(2,I)*Y(2,1,I)                                 QSET     0280
      DO 20 J=1,3                                                           QSET     0290
      F(3,J,I)=E1*F(3,J,I)+T1(J)+EB*X(J,I)                                  QSET     0300
      F(4,J,I)=E1*F(4,J,I)+T2(J)+EC*X(J,I)                                  QSET     0310
      Y(3,J,I)=E1*Y(1,J,I)+T3(J)-E2*X(J,I)                                  QSET     0320
   20 Y(4,J,I)=E1*Y(2,J,I)+T4(J)-E2D*X(J,I)                                 QSET     0330
      RETURN                                                                QSET     0340
      END                                                                   QSET     0350
```

```
      DOUBLE PRECISION FUNCTION RCRT(A,PL,Z,IP)                          RCRT    0010
C                                                      REV 03 07/19/73    RCRT    0020
C     COMPUTES THE RADIUS OF CURVATURE AT POINT Z OF ELLIPSOID A         RCRT    0030
C     IN THE PLANE PL(I,IP) WHERE                                        RCRT    0040
C                                                                        RCRT    0050
C          A: 3X3 MATRIX DEFINING ELLIPSOID.                             RCRT    0060
C         PL: 4X3 MATRIX CONTAINING THREE ORTHONORMAL VECTORS.           RCRT    0070
C          Z: 3 COORDINATES OF POINT ON THE ELLIPSOID                    RCRT    0080
C             AS MEASURED FROM CENTER OF ELLIPSOID                       RCRT    0090
C         IP: IDENTIFIES THE NORMAL VECTOR OF PLANE IN WHICH THE         RCRT    0100
C             RADIUS OF CURVATURE IS DESIRED.                            RCRT    0110
C                                                                        RCRT    0120
      IMPLICIT REAL*8 (A-H,O-Z)                                          RCRT    0130
      DIMENSION A(3,3),PL(4,3),Z(3),T(5)                                 RCRT    0140
      DO 10  I=1,5                                                       RCRT    0150
   10 T(I) = 0.0                                                         RCRT    0160
      M = IP+1                                                           RCRT    0170
      N = IP+2                                                           RCRT    0180
      IF(M.GT.3) M = M-3                                                 RCRT    0190
      IF(N.GT.3) N = N-3                                                 RCRT    0200
      DO 30  I=1,3                                                       RCRT    0210
      S1 = 0.                                                            RCRT    0220
      S2 = 0.                                                            RCRT    0230
      DO 20  J=1,3                                                       RCRT    0240
      S1 = S1+A(I,J)*PL(J,M)                                             RCRT    0250
   20 S2 = S2+A(I,J)*PL(J,N)                                             RCRT    0260
      T(1) = T(1)+S1*Z(I)                                                RCRT    0270
      T(2) = T(2)+S2*Z(I)                                                RCRT    0280
      T(3) = T(3)+S1*PL(I,M)                                             RCRT    0290
      T(4) = T(4)+S2*PL(I,N)                                             RCRT    0300
   30 T(5) = T(5)+S1*PL(I,N)                                             RCRT    0310
      W = DSQRT(T(1)**2+T(2)**2)                                         RCRT    0320
      T(1) = T(1)/W                                                      RCRT    0330
      T(2) = T(2)/W                                                      RCRT    0340
      RCRT = W/(T(3)*T(2)**2-2.0*T(1)*T(2)*T(5)+T(4)*T(1)**2)            RCRT    0350
      IF(RCRT.LT.0.0) RCRT = -RCRT                                       RCRT    0360
      RETURN                                                             RCRT    0370
      END                                                                RCRT    0380
```

```
      SUBROUTINE ROTATE                                                 ROTATE  0010
C                                                          REV 20 04/23/80  ROTATE  0020
C     THE PURPOSE OF THIS ROUTINE IS TO TRANSFORM THOSE VARIABLES THAT  ROTATE  0030
C     HAVE BEEN SUPPLIED IN LOCAL GEOMETRIC COORDINATES TO PRINCIPAL    ROTATE  0040
C     AXES COORDINATES AS INDICATED BY LPMI(I) # 0 FOR I = 1 TO NSEG.   ROTATE  0050
C                                                                       ROTATE  0060
      IMPLICIT REAL*8 (A-H,O-Z)                                         ROTATE  0070
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           ROTATE  0080
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)        ROTATE  0090
      COMMON/RSAVE/  XSG(3,20,3),DPMI(3,3,30),LPMI(30),NSG(7),MSG(20,7) ROTATE  0100
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(3,60),HA(3,60),HB(3,60), ROTATE  0110
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),     ROTATE  0120
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)    ROTATE  0130
      COMMON/CNTSRF/ PL(17,30),BELT(20,8),TPTS(6,8),BD(24,40)           ROTATE  0140
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30), ROTATE  0150
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       ROTATE  0160
      COMMON/JBARTZ/ MNPL(   30),MNBLT(    8),MNSEG(   30),MNBAG(    6), ROTATE  0170
     *               MPL(3,5,30),MBLT(3,5,8),MSEG(3,5,30),MBAG(3,10,6), ROTATE  0180
     *               NTPL( 5,30),NTBLT( 5,8),NTSEG( 5,30)               ROTATE  0190
      COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),   ROTATE  0200
     *               HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12), ROTATE  0210
     *               RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),              ROTATE  0220
     *               KQ1(12),KQ2(12),KQTYPE(12)                         ROTATE  0230
      COMMON/DAMPER/ APSDM(3,20),APSDN(3,20),ASD(5,20),MSDM(20),MSDN(20) ROTATE  0240
      COMMON/HRNESS/ BAR(15,100),BB(100),BBDOT(100),PLOSS(2,100),       ROTATE  0250
     *               XLONG(20),HTIME(2),IBAR(5,100),NL(2,100),          ROTATE  0260
     *               NPTSPB(20),NPTPLY(20),NTHRNS(20),NBLTPH(5)         ROTATE  0270
      COMMON/TEMPVS/ T1(3),T3(3,3),LBD(40)                              ROTATE  0280
C                                                                       ROTATE  0290
C     TRANSFORM DIRECTION COSINE MATRICEES D FROM INPUT CARDS G.3.       ROTATE  0300
C                                                                       ROTATE  0310
      LTEST = 0                                                         ROTATE  0320
      DO  20  J=1,30                                                    ROTATE  0330
      IF  (J.GT.NSEG)  LPMI(J) = 0                                      ROTATE  0340
      IF  (LPMI(J).EQ.0)  GO TO 20                                      ROTATE  0350
      LTEST = 1                                                         ROTATE  0360
      DO  12  I=1,3                                                     ROTATE  0370
      T1(I) = WMEG(I,J)                                                 ROTATE  0380
      DO  12  K=1,3                                                     ROTATE  0390
   12 T3(I,K) = D(I,K,J)                                                ROTATE  0400
      CALL MAT33 (DPMI(1,1,J),T3,D(1,1,J))                              ROTATE  0410
      CALL MAT31 (DPMI(1,1,J),T1,WMEG(1,J))                             ROTATE  0420
   20 CONTINUE                                                          ROTATE  0430
      IF  (LTEST.EQ.0)  GO TO 99                                        ROTATE  0440
C                                                                       ROTATE  0450
C     TRANSFORM SR,HT AND HB FROM INPUT CARDS B.3.                      ROTATE  0460
C                                                                       ROTATE  0470
      IF  (NJNT.LE.0)  GO TO 31                                         ROTATE  0480
      DO  30  J=1,NJNT                                                  ROTATE  0490
      I = IABS(JNT(J))                                                  ROTATE  0500
```

```
      DO  24  K=1,2                                                    ROTATE  0510
      IF  (I.EQ.0 .OR. LPMI(I).EQ.0)  GO TO 24                         ROTATE  0520
      IJ = 2*J-2+K                                                     ROTATE  0530
      DO  22  LI=1,3                                                   ROTATE  0540
      T1(LI) = SR(LI,IJ)                                               ROTATE  0550
      DO  22  LJ=1,3                                                   ROTATE  0560
   22 T3(LI,LJ) = HT(LI,LJ,IJ)                                         ROTATE  0570
      CALL MAT31 (DPMI(1,1,I),T1,SR(1,IJ))                             ROTATE  0580
      CALL MAT33 (DPMI(1,1,I),T3,HT(1,1,IJ))                           ROTATE  0590
      DO  23  LI=1,3                                                   ROTATE  0600
   23 HB(LI,IJ) = HT(LI,2,IJ)                                          ROTATE  0610
   24 I = J+1                                                          ROTATE  0620
   30 CONTINUE                                                         ROTATE  0630
C                                                                      ROTATE  0640
C     TRANSFORM RK1,RK2 FROM INPUT CARDS D.6.                          ROTATE  0650
C                                                                      ROTATE  0660
   31 IF  (NQ.LE.0)  GO TO 41                                          ROTATE  0670
      K5 = 0                                                           ROTATE  0680
      DO  40  K=1,NQ                                                   ROTATE  0690
      IF  (K5.EQ.1)  GO TO 39                                          ROTATE  0700
      KSEG = KQ1(K)                                                    ROTATE  0710
      IF  (LPMI(KSEG).EQ.0)  GO TO 36                                  ROTATE  0720
      DO  35  I=1,3                                                    ROTATE  0730
   35 T1(I) = RK1(I,K)                                                 ROTATE  0740
      CALL MAT31 (DPMI(1,1,KSEG),T1,RK1(1,K))                          ROTATE  0750
   36 KSEG = KQ2(K)                                                    ROTATE  0760
      IF  (LPMI(KSEG).EQ.0)  GO TO 40                                  ROTATE  0770
      DO  37  I=1,3                                                    ROTATE  0780
   37 T1(I) = RK2(I,K)                                                 ROTATE  0790
      CALL MAT31 (DPMI(1,1,KSEG),T1,RK2(1,K))                          ROTATE  0800
   39 IF  (KQTYPE(K).EQ.5) K5 = 1-K5                                   ROTATE  0810
   40 CONTINUE                                                         ROTATE  0820
C                                                                      ROTATE  0830
C     TRANSFORM APSDM,APSDN FROM INPUT CARDS D.8.                      ROTATE  0840
C                                                                      ROTATE  0850
   41 IF  (NSD.LE.0)  GO TO 51                                         ROTATE  0860
      DO  50  J=1,NSD                                                  ROTATE  0870
      KSEG = MSDM(J)                                                   ROTATE  0880
      IF  (LPMI(KSEG).EQ.0)  GO TO 44                                  ROTATE  0890
      DO  43  I=1,3                                                    ROTATE  0900
   43 T1(I) = APSDM(I,J)                                               ROTATE  0910
      CALL MAT31 (DPMI(1,1,KSEG),T1,APSDM(1,J))                        ROTATE  0920
   44 KSEG = MSDN(J)                                                   ROTATE  0930
      IF  (LPMI(KSEG).EQ.0)  GO TO 50                                  ROTATE  0940
      DO  45  I=1,3                                                    ROTATE  0950
   45 T1(I) = APSDN(I,J)                                               ROTATE  0960
      CALL MAT31 (DPMI(1,1,KSEG),T1,APSDN(1,J))                        ROTATE  0970
   50 CONTINUE                                                         ROTATE  0980
C                                                                      ROTATE  0990
C     CHECK PLANE AND ELLPISOID ASSIGNMENTS ON INPUT CARDS F.1.        ROTATE  1000
```

```
C     TRANSFORM PLANE ARRAYS SET UP FROM INPUT CARD D.1.                    ROTATE  1010
C                                                                           ROTATE  1020
   51 DO  52  J=1,40                                                        ROTATE  1030
      LBD(J) = 0                                                            ROTATE  1040
   52 IF  (J.LE.NSEG)  LBD(J) = J                                           ROTATE  1050
      IF  (NPL.LE.0)  GO TO 61                                              ROTATE  1060
      DO  60  J=1,NPL                                                       ROTATE  1070
      IF  (MNPL(J).EQ.0)  GO TO 60                                          ROTATE  1080
      LPL = 0                                                               ROTATE  1090
      KPL = MNPL(J)                                                         ROTATE  1100
      DO  56  I=1,KPL                                                       ROTATE  1110
      M1 = MPL (1,I,J)                                                      ROTATE  1120
      M2 = MPL (2,I,J)                                                      ROTATE  1130
      M3 = MPL (3,I,J)                                                      ROTATE  1140
      IF  (LPL.EQ.M1 .OR. LPL.EQ.0)  GO TO 54                               ROTATE  1150
      WRITE  (6,53) J,M1,LPL                                                ROTATE  1160
   53 FORMAT('0 INPUT ERROR HAS BEEN DETECTED IN SUBROUTINE ROTATE.'/       ROTATE  1170
     *       '  PLANE NO.',I3,' HAS BEEN ASSIGNED TO BOTH SEGMENTS NO.',    ROTATE  1180
     *       I3,' AND NO.',I3,'.'/'  PROGRAM IS BEING TERMINATED.')         ROTATE  1190
      STOP 43                                                               ROTATE  1200
   54 LPL = M1                                                              ROTATE  1210
      IF  (LBD(M3).EQ.M2 .OR. LBD(M3).EQ.0)  GO TO 55                       ROTATE  1220
      WRITE  (6,68)  M3,M2,LBD(M3)                                          ROTATE  1230
      STOP 44                                                               ROTATE  1240
   55 LBD(M3) = M2                                                          ROTATE  1250
   56 CONTINUE                                                              ROTATE  1260
      IF  (LPMI(LPL).EQ.0)  GO TO 60                                        ROTATE  1270
      DO  59  K=1,3                                                         ROTATE  1280
      KK = K*(10-K)-9                                                       ROTATE  1290
      DO  58  I=1,3                                                         ROTATE  1300
      IK = KK+I                                                             ROTATE  1310
   58 T1(I) = PL(IK,J)                                                      ROTATE  1320
      CALL MAT31 (DPMI(1,1,LPL),T1,PL(KK+1,J))                              ROTATE  1330
   59 CONTINUE                                                              ROTATE  1340
   60 CONTINUE                                                              ROTATE  1350
C                                                                           ROTATE  1360
C     CHECK ELLIPSOID ASSIGNMENTS ON INPUT CARDS F.2.                       ROTATE  1370
C     TRANSFORM BELT(L,J) FOR L=1,9 FROM INPUT CARDS D.3.                   ROTATE  1380
C                                                                           ROTATE  1390
   61 IF  (NBLT.LE.0)  GO TO 66                                             ROTATE  1400
      DO  65  J=1,NBLT                                                      ROTATE  1410
      IF  (MNBLT(J).EQ.0)  GO TO 65                                         ROTATE  1420
      KBLT = MNBLT(J)                                                       ROTATE  1430
      DO  62  I=1,KBLT                                                      ROTATE  1440
      M1 = MBLT(1,I,J)                                                      ROTATE  1450
      M2 = MBLT(2,I,J)                                                      ROTATE  1460
      M3 = MBLT(3,I,J)                                                      ROTATE  1470
      IF  (LBD(M3).EQ.M2 .OR. LBD(M3).EQ.0)  GO TO 62                       ROTATE  1480
      WRITE  (6,68)  M3,M2,LBD(M3)                                          ROTATE  1490
      STOP 45                                                               ROTATE  1500
```

```
   62 LBD(M3) = M2                                                      ROTATE  1510
      IF  (LPMI(M1).EQ.0)  GO TO 63                                     ROTATE  1520
      DO  57  I=1,3                                                     ROTATE  1530
      T3(I,1) = BELT(I  ,J)                                             ROTATE  1540
   57 T3(I,2) = BELT(I+3,J)                                             ROTATE  1550
      CALL MAT31 (DPMI(1,1,M1),T3(1,1),BELT(1,J))                       ROTATE  1560
      CALL MAT31 (DPMI(1,1,M1),T3(1,2),BELT(4,J))                       ROTATE  1570
   63 IF  (LPMI(M2).EQ.0)  GO TO 65                                     ROTATE  1580
      DO  64  I=1,3                                                     ROTATE  1590
   64 T3(I,3) = BELT(I+6,J)                                             ROTATE  1600
      CALL MAT31 (DPMI(1,1,M2),T3(1,3),BELT(7,J))                       ROTATE  1610
   65 CONTINUE                                                          ROTATE  1620
C                                                                       ROTATE  1630
C     CHECK ELLIPSOID ASSIGNMENTS ON INPUT CARDS F.3.                   ROTATE  1640
C                                                                       ROTATE  1650
   66 DO  70  J=1,NSEG                                                  ROTATE  1660
      IF  (MNSEG(J).EQ.0)  GO TO 70                                     ROTATE  1670
      KSEG = MNSEG(J)                                                   ROTATE  1680
      DO  69  I=1,KSEG                                                  ROTATE  1690
      M1 = MSEG(1,I,J)                                                  ROTATE  1700
      M2 = MSEG(2,I,J)                                                  ROTATE  1710
      M3 = MSEG(3,I,J)                                                  ROTATE  1720
      IF  (LBD(M1).EQ.J .OR. LBD(M1).EQ.0)  GO TO 67                    ROTATE  1730
      WRITE  (6,68)  M1,J,LBD(M1)                                       ROTATE  1740
      STOP 46                                                           ROTATE  1750
   67 LBD(M1) = J                                                       ROTATE  1760
      IF  (LBD(M3).EQ.M2 .OR. LBD(M3).EQ.0)  GO TO 69                   ROTATE  1770
      WRITE  (6,68)  M3,M2,LBD(M3)                                      ROTATE  1780
   68 FORMAT('0 INPUT ERROR HAS BEEN DETECTED IN SUBROUTINE ROTATE.'/   ROTATE  1790
     *    '  ELLPISOID NO.',I3,' HAS BEEN ASSIGNED TO BOTH SEGMENTS NO.', ROTATE  1800
     *     I3,' AND NO.',I3,'.'/'  PROGRAM IS BEING TERMINATED.')       ROTATE  1810
      STOP 47                                                           ROTATE  1820
   69 LBD(M3) = M2                                                      ROTATE  1830
   70 CONTINUE                                                          ROTATE  1840
C                                                                       ROTATE  1850
C     CHECK ELLIPSOID ASSIGNMENTS ON INPUT CARDS F.6.                   ROTATE  1860
C                                                                       ROTATE  1870
      IF  (NBAG.EQ.0)  GO TO 74                                         ROTATE  1880
      DO  73  J=1,NBAG                                                  ROTATE  1890
      IF  (MNBAG(J).EQ.0)  GO TO 73                                     ROTATE  1900
      KBAG = MNBAG(J)                                                   ROTATE  1910
      DO  72  I=1,KBAG                                                  ROTATE  1920
      M2 = MBAG(2,I,J)                                                  ROTATE  1930
      M3 = MBAG(3,I,J)                                                  ROTATE  1940
      IF  (LBD(M3).EQ.M2 .OR. LBD(M3).EQ.0)  GO TO 72                   ROTATE  1950
      WRITE  (6,68)  M3,M2,LBD(M3)                                      ROTATE  1960
      STOP 50                                                           ROTATE  1970
   72 LBD(M3) = M2                                                      ROTATE  1980
   73 CONTINUE                                                          ROTATE  1990
C                                                                       ROTATE  2000
```

```
C     CHECK ELLIPSOID ASSIGNMENTS ON INPUT CARDS F.8.                    ROTATE  2010
C     TRANSFORM BAR(L,K) FOR L=4,12 FROM INPUT CARDS F.8.D.              ROTATE  2020
C                                                                        ROTATE  2030
   74 IF  (NHRNSS.EQ.0)  GO TO 81                                        ROTATE  2040
      J1 = 1                                                             ROTATE  2050
      K1 = 1                                                             ROTATE  2060
      DO  80  II=1,NHRNSS                                                ROTATE  2070
      IF  (NBLTPH(II).LE.0)  GO TO 80                                    ROTATE  2080
      J2 = J1 + NBLTPH(II) - 1                                           ROTATE  2090
      DO  79  JJ=J1,J2                                                   ROTATE  2100
      IF  (NPTSPB(JJ).LE.0)  GO TO 79                                    ROTATE  2110
      K2 = K1 + NPTSPB(JJ) - 1                                           ROTATE  2120
      DO  78  K=K1,K2                                                    ROTATE  2130
      M2 = MOD(IBAR(1,K),100)                                            ROTATE  2140
      M3 = IBAR(2,K)                                                     ROTATE  2150
      IF  (LBD(M3).EQ.M2 .OR. LBD(M3).EQ.0)  GO TO 75                    ROTATE  2160
      WRITE  (6,68)  M3,M2,LBD(M3)                                       ROTATE  2170
      STOP 51                                                            ROTATE  2180
   75 LBD(M3) = M2                                                       ROTATE  2190
      IF  (LPMI(M2).EQ.0)  GO TO 78                                      ROTATE  2200
      DO  77  J=3,9,3                                                    ROTATE  2210
      DO  76  I=1,3                                                      ROTATE  2220
      IJ = I+J                                                           ROTATE  2230
   76 T1(I) = BAR(IJ,K)                                                  ROTATE  2240
   77 CALL MAT31 (DPMI(1,1,M2),T1,BAR(J+1,K))                            ROTATE  2250
   78 CONTINUE                                                           ROTATE  2260
      K1 = K2+1                                                          ROTATE  2270
   79 CONTINUE                                                           ROTATE  2280
      J1 = J2+1                                                          ROTATE  2290
   80 CONTINUE                                                           ROTATE  2300
C                                                                        ROTATE  2310
C     TRANSFORM DATA IN BD ARRAYS FOR ELLIPSOIDS THAT HAVE BEEN ASSIGNED ROTATE  2320
C                                                                        ROTATE  2330
   81 DO  90  J=1,40                                                     ROTATE  2340
      IF  (LBD(J).EQ.0)  GO TO 90                                        ROTATE  2350
      KSEG = LBD(J)                                                      ROTATE  2360
      IF  (LPMI(KSEG).EQ.0)  GO TO 90                                    ROTATE  2370
      DO  82  I=1,3                                                      ROTATE  2380
   82 T1(I) = BD(I+3,J)                                                  ROTATE  2390
      CALL MAT31 (DPMI(1,1,KSEG),T1,BD(4,J))                            ROTATE  2400
      CALL DOTT33 (BD( 7,J),DPMI(1,1,KSEG),T3)                          ROTATE  2410
      CALL MAT33  (DPMI(1,1,KSEG),T3,BD( 7,J))                          ROTATE  2420
      CALL DOTT33 (BD(16,J),DPMI(1,1,KSEG),T3)                          ROTATE  2430
      CALL MAT33  (DPMI(1,1,KSEG),T3,BD(16,J))                          ROTATE  2440
   90 CONTINUE                                                           ROTATE  2450
   99 RETURN                                                             ROTATE  2460
      END                                                                ROTATE  2470
```

```
      SUBROUTINE ROT (A,L,TH)                                                ROT      0010
C                                                          REV 19 08/05/78    ROT      0020
C     COMPUTES ROTATION MATRIX A FOR ANGLE TH                                 ROT      0030
C     ABOUT X,Y OR Z AXIS AS L = 1,2, OR 3.                                   ROT      0040
C                                                                             ROT      0050
C       ARGUMENTS:                                                            ROT      0060
C         A: 3X3 ROTATION MATRIX TO BE COMPUTED.                              ROT      0070
C         L: 1,2 OR 3 TO ROTATE ABOUT X,Y OR Z AXIS.                          ROT      0080
C         TH: ANGLE OF ROTATION IN RADIANS.                                   ROT      0090
C                                                                             ROT      0100
      IMPLICIT REAL*8 (A-H,O-Z)                                               ROT      0110
      DIMENSION  A(3,3)                                                       ROT      0120
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                               ROT      0130
     *               UNITL,UNITM,UNITT,GRAVTY(3)                            ROT      0140
       C=DCOS(TH)                                                             ROT      0150
       S=DSIN(TH)                                                             ROT      0160
       IF  (DABS(C).LT.EPS(12))  C= 0.0                                       ROT      0170
       IF  (DABS(S).LT.EPS(12))  S= 0.0                                       ROT      0180
       ONE = 1.0                                                              ROT      0190
       IF  (1.0-DABS(C).LT.EPS(12))  C = DSIGN(ONE,C)                         ROT      0200
       IF  (1.0-DABS(S).LT.EPS(12))  S = DSIGN(ONE,S)                         ROT      0210
      IF (L.EQ.2) S = -S                                                      ROT      0220
        DO 30 I=1,3                                                           ROT      0230
        IF(I.EQ.3)GO TO 20                                                    ROT      0240
        DO 10 J=I,2                                                           ROT      0250
        A(I,J+1)=0.0                                                          ROT      0260
        A(J+1,I)=0.0                                                          ROT      0270
        IF(I+J+L.NE.5)GO TO 10                                                ROT      0280
        A(I,J+1)=S                                                            ROT      0290
        A(J+1,I)=-S                                                           ROT      0300
   10   CONTINUE                                                              ROT      0310
   20   A(I,I)= C                                                             ROT      0320
        IF(I.EQ.L)A(I,I)=1.0                                                  ROT      0330
   30   CONTINUE                                                              ROT      0340
        RETURN                                                                ROT      0350
      END                                                                     ROT      0360
```

```
      SUBROUTINE RSTART(IF,IT)                                          RSTART  0010
C                                                        REV 20 05/23/80  RSTART  0020
C     THE FIVE FUNCTIONS OF SUBROUTINE RSTART ARE:                       RSTART  0030
C        1. READ INPUT & INITIALIZATION RECORD FROM OLD RESTART TAPE.    RSTART  0040
C        2. WRITE INPUT & INITIALIZATION RECORD ONTO NEW RESTART TAPE.   RSTART  0050
C        3. READ TIME POINT RECORD FROM OLD RESTART TAPE.                RSTART  0060
C        4. READ NEW INPUT DATA FROM INPUT STREAM FOR RESTART.           RSTART  0070
C        5. WRITE TIME POINT RECORD ONTO NEW RESTART TAPE.               RSTART  0080
C                                                                        RSTART  0090
      IMPLICIT REAL*8(A-H,O-Z)                                           RSTART  0100
C                                                                        RSTART  0110
C     ALL LABELED COMMON BLOCKS ARE INCLUDED HERE                        RSTART  0120
C     TO GIVE A COMPLETE SET FOR REFERENCE                               RSTART  0130
C                                                                        RSTART  0140
C  1
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,            RSTART  0150
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)         RSTART  0160
      DIMENSION IC1(50)                                                  RSTART  0170
      EQUIVALENCE (IC1(1),NSEG)                                          RSTART  0180
C  2                                                                     RSTART  0190
      COMMON/CNTSRF/ PL(17,30),BELT(20,8),TPTS(6,8),BD(24,40)            RSTART  0200
      DIMENSION RC2(1678)                                                RSTART  0210
      EQUIVALENCE (RC2(1),PL(1,1))                                       RSTART  0220
C  3                                                                     RSTART  0230
      COMMON/VPOSTN/ ZPLT(3),SPLT(3),AXV(3,6),VATAB(6,101,6),            RSTART  0240
     *               VTO(6),VDT(6),TIMEV(6),OMEGV(6),NVTAB(6),INDXV(6)   RSTART  0250
      DIMENSION RC3(3684),IC3(12)                                        RSTART  0260
      EQUIVALENCE (RC3(1),ZPLT(1)),(IC3(1),NVTAB(1))                     RSTART  0270
C  4                                                                     RSTART  0280
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30), RSTART  0290
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)        RSTART  0300
      DIMENSION RC4(900)                                                 RSTART  0310
      EQUIVALENCE (RC4(1),D(1,1,1))                                      RSTART  0320
C  5                                                                     RSTART  0330
      COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60), RSTART  0340
     *               F(3,30),TQ(3,30),WJ(30)                             RSTART  0350
      DIMENSION RC5A(1296),RC5B(210)                                     RSTART  0360
      EQUIVALENCE (RC5A(1),V1(1,1)),(RC5B(1),F(1,1))                     RSTART  0370
C  6                                                                     RSTART  0380
      COMMON/ABDATA/ ZDEP(3,5),DBR(3,3,5),DPVCTR(3,5),DEPLOY(3,5),       RSTART  0390
     *               AB(3,5),B(9,4,5),ZR(3,4,5),BFB(3,4,5),DRR(9,4,5),   RSTART  0400
     *               VBAGG(5),VSCS(5),SPRK(5),CK(5),CMASS(5),CYMIN(5),   RSTART  0410
     *               CYMOUT(5),BAGPV(5),PD(5),VBAG(5),VOLBP(5),          RSTART  0420
     *               PCYV(5),PCYMIN(5),PVBAG(5),TV1(3,4,5),TV2(3,10,5),  RSTART  0430
     *               SWITCH(5),PYMOUT(5),SCALE(5),PREVT,IFULL(6)         RSTART  0440
      DIMENSION RC6A(610),RC6B(271)                                      RSTART  0450
      EQUIVALENCE (RC6A(1),ZDEP(1,1)) , (RC6B(1),CYMIN(1))               RSTART  0460
C  7                                                                     RSTART  0470
      COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),            RSTART  0480
     *               BLTTTL(5,8),PLTTL(5,30),BAGTTL(5,6),SEG(30),        RSTART  0490
     *               JOINT(30),CGS(30),JS(30)                            RSTART  0500
```

```
      REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT          RSTART  0510
      LOGICAL*1 CGS,JS                                                      RSTART  0520
      REAL  RC7,RC7A,XDTE,XCMENT                                            RSTART  0530
      DIMENSION RC7(305),RC7A(348),XDTE(3),XCMENT(40)                       RSTART  0540
      EQUIVALENCE (RC7(1),VPSTTL(1)),(RC7A(1),DATE(1))                      RSTART  0550
C  8                                                                        RSTART  0560
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                             RSTART  0570
     *               UNITL,UNITM,UNITT,GRAVTY(3)                            RSTART  0580
      DIMENSION RC8(34)                                                     RSTART  0590
      EQUIVALENCE (RC8(1),PI)                                               RSTART  0600
C  9                                                                        RSTART  0610
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(3,60),HA(3,60),HB(3,60),     RSTART  0620
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),         RSTART  0630
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)        RSTART  0640
      DIMENSION RC9(2400),IC9(150)                                          RSTART  0650
      EQUIVALENCE (RC9(1),PHI(1,1)),(IC9(1),JNT(1))                         RSTART  0660
C 10                                                                        RSTART  0670
      COMMON/JBARTZ/ MNPL(   30),MNBLT(   8),MNSEG(   30),MNBAG(    6),    RSTART  0680
     *               MPL(3,5,30),MBLT(3,5,8),MSEG(3,5,30),MBAG(3,10,6),     RSTART  0690
     *               NTPL( 5,30),NTBLT( 5,8),NTSEG( 5,30)                   RSTART  0700
      DIMENSION IC10(1614)                                                  RSTART  0710
      EQUIVALENCE (IC10(1),MNPL(1))                                         RSTART  0720
C 11                                                                        RSTART  0730
      COMMON/FORCES/ PSF(7,30),BSF(4,20),SSF(10,20),BAGSF(3,20),            RSTART  0740
     *               PRJNT(7,30),NPANEL(5),NPSF,NBSF,NSSF,NBGSF             RSTART  0750
      DIMENSION RC11(760),IC11(9)                                           RSTART  0760
      EQUIVALENCE (RC11(1),PSF(1,1)),(IC11(1),NPANEL(1))                    RSTART  0770
C 12                                                                        RSTART  0780
      COMMON/INTEST/ SGTEST(3,4,30),XTEST(3,120),SEGT(120),REGT(120)        RSTART  0790
      REAL    SEGT                                                          RSTART  0800
      DIMENSION RC12(720)                                                   RSTART  0810
      EQUIVALENCE (RC12(1),SGTEST(1,1,1))                                   RSTART  0820
C 13                                                                        RSTART  0830
      COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),       RSTART  0840
     *               HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12),    RSTART  0850
     *               RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),                  RSTART  0860
     *               KQ1(12),KQ2(12),KQTYPE(12)                             RSTART  0870
      DIMENSION RC13(72),IC13(36),RC13A(1212),RC13H(348)                    RSTART  0880
      EQUIVALENCE (RC13(1),RK1(1,1)),(IC13(1),KQ1(1)),                      RSTART  0890
     *            (RC13A(1),A13(1,1,1)),(RC13H(1),HHT(1,1,1))               RSTART  0900
C 14                                                                        RSTART  0910
      COMMON/TABLES/ MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(500),TAB(2600)    RSTART  0920
      DIMENSION IC14(554)                                                   RSTART  0930
      EQUIVALENCE (IC14(1),MXNTI)                                           RSTART  0940
C 15                                                                        RSTART  0950
      COMMON/COMAIN/VAR(240),DER(240),DT,H0,HMAX,HMIN,RSTIME,               RSTART  0960
     *              ISTEP,NSTEPS,NDINT,NEQ,IRSIN,IRSOUT                     RSTART  0970
      DIMENSION RC15(485),IC15(6)                                           RSTART  0980
      EQUIVALENCE (RC15(1),VAR(1)),(IC15(1),ISTEP)                          RSTART  0990
C 16                                                                        RSTART  1000
```

```
      COMMON/CDINT/ UU(4),GH(3,4),                                          RSTART  1010
     *              E(3,240),FF(5,240),GG(5,240),Y(5,240),U(5,240),         RSTART  1020
     *              H,HPRINT,HS,TPRINT,TSTART,ICNT,IDBL,IFLAG               RSTART  1030
C       NOTE: FF REPLACES F FROM SUBROUTINE DINT.                           RSTART  1040
      DIMENSION RC16(5541),IC16(3)                                          RSTART  1050
      EQUIVALENCE (RC16(1),UU(1)),(IC16(1),ICNT)                            RSTART  1060
C 17                                                                        RSTART  1070
      COMMON/DAMPER/ APSDM(3,20),APSDN(3,20),ASD(5,20),MSDM(20),MSDN(20)    RSTART  1080
      DIMENSION RC17(220),IC17(40)                                          RSTART  1090
      EQUIVALENCE (RC17(1),APSDM(1,1)),(IC17(1),MSDM(1))                    RSTART  1100
C 18                                                                        RSTART  1110
      COMMON/CEULER/ IEULER(30),HIR(3,3,30),ANG(3,30),ANGD(3,30),           RSTART  1120
     *               FE(3,30),TQE(3,30),CONST(3,30)                         RSTART  1130
      DIMENSION RC18(720)                                                   RSTART  1140
      EQUIVALENCE (RC18(1),HIR(1,1,1))                                      RSTART  1150
C 19                                                                        RSTART  1160
      COMMON/TEMPVI/ CREST,TTI(3),R1I(3),R2I(3),JSTOP(4,2,30)               RSTART  1170
      DIMENSION RC19(10),IC19(180)                                          RSTART  1180
      EQUIVALENCE (RC19(1),CREST),(IC19(1),JSTOP(1,1,1))                    RSTART  1190
C 20                                                                        RSTART  1200
      COMMON/CYDATA/ CYTD(5),CYPA(5),CYSP(5),CYT0(5),CYV0(5),CYCD(5),       RSTART  1210
     *               CYK(5),CYR(5),CYAT(5),CYPV(5),CYCD0(5),CYA0(5),        RSTART  1220
     *               CYP0(5),CYSS(5),CYL0(5),CYC(5),CYRHO0(5),CYVMAX(5),    RSTART  1230
     *               CYORFC(5),CYRHO(5),CYT(5),CYP(5),CYV(5)                RSTART  1240
      DIMENSION RC20A(95),RC20B(20)                                         RSTART  1250
      EQUIVALENCE (RC20A(1),CYTD(1)) , (RC20B(1),CYRHO(1))                  RSTART  1260
C 21                                                                        RSTART  1270
      COMMON/RSAVE/  XSG(3,20,3),DPMI(3,3,30),LPMI(30),NSG(7),MSG(20,7)     RSTART  1280
      DIMENSION RC21(450),IC21(177)                                         RSTART  1290
      EQUIVALENCE (RC21(1),XSG(1,1,1)) , (IC21(1),LPMI(1))                  RSTART  1300
C 22                                                                        RSTART  1310
      COMMON/FLXBLE/ HF(4,12,8),B42(3,3,24),V4(3,8),NFLEX(3,8)              RSTART  1320
      DIMENSION RC22(624),IC22(24)                                          RSTART  1330
      EQUIVALENCE (RC22(1),HF(1,1,1)) , (IC22(1),NFLEX(1,1))                RSTART  1340
C 23                                                                        RSTART  1350
      COMMON/HRNESS/ BAR(15,100),BB(100),BBDOT(100),PLOSS(2,100),           RSTART  1360
     *               XLONG(20),HTIME(2),IBAR(5,100),NL(2,100),              RSTART  1370
     *               NPTSPB(20),NPTPLY(20),NTHRNS(20),NBLTPH(5)             RSTART  1380
      DIMENSION RC23(1922),IC23(765)                                        RSTART  1390
      EQUIVALENCE (RC23(1),BAR(1,1)),(IC23(1),IBAR(1,1))                    RSTART  1400
C 24                                                                        RSTART  1410
      COMMON/WINDFR/ WTIME(30),QFU(3,5),QFV(3,5),                           RSTART  1420
     *               IWIND(30),MWSEG(5,30),NFVSEG(6),NFVNT(5)               RSTART  1430
      DIMENSION RC24(60),IC24(191)                                          RSTART  1440
      EQUIVALENCE (RC24(1),WTIME(1)),(IC24(1),IWIND(1))                     RSTART  1450
C                                                                           RSTART  1460
      REAL AOLD4,AAOLD4                                                     RSTART  1470
      DIMENSION COMMON(24),INDEX(3)                                         RSTART  1480
      DATA COMMON /8HCONTRL  ,8HCNTSRF  ,8HVPOSTN  ,8HSGMNTS  ,             RSTART  1490
     *             8HCMATRX  ,8HABDATA  ,8HTITLES  ,8HCNSNTS  ,             RSTART  1500
```

```
     *             8HDESCRP  ,8HJBARTZ  ,8HFORCES  ,8HINTEST  ,          RSTART  1510
     *             8HCSTRNT  ,8HTABLES  ,8HCOMAIN  ,8HCDINT   ,          RSTART  1520
     *             8HDAMPER  ,8HCEULER  ,8HTEMPVI  ,8HCYDATA  ,          RSTART  1530
     *             8HRSAVE   ,8HFLXBLE  ,8HHRNESS  ,8HWINDFR  /          RSTART  1540
      DATA BLANK/8H       /                                             RSTART  1550
      CALL ELTIME(1,25)                                                 RSTART  1560
      GO TO (100,200,300,400,500),IF                                    RSTART  1570
C                                                                       RSTART  1580
C        1. READ INPUT & INITIALIZATION RECORD FROM OLD RESTART TAPE.   RSTART  1590
C                                                                       RSTART  1600
  100 READ  (IT) IC1, PL, RC3, IC3, NSYM, RC6A, IFULL, XDTE, XCMENT,    RSTART  1610
     *           RC7, CGS, JS, RC8, RC9, IC9, IC10, NPANEL, SGTEST,     RSTART  1620
     *           RC13, IC13, IC14, DT, H0, HMAX, HMIN, NSTEPS, NDINT,   RSTART  1630
     *           RC17, IC17, IEULER, IC19, RC20A, RC21, IC21, HF, IC22, RSTART  1640
     *           RC23, IC23, RC24, IC24                                 RSTART  1650
      WRITE (6,101) IT,XDTE,XCMENT                                      RSTART  1660
  101 FORMAT('0 INPUT DATA HAS BEEN READ IN FROM UNIT NO.',I4//         RSTART  1670
     *       10X,3A4//10X,20A4/10X,20A4)                                RSTART  1680
      GO TO 999                                                         RSTART  1690
C                                                                       RSTART  1700
C        2. WRITE INPUT & INITIALIZATION RECORD ONTO NEW RESTART TAPE.  RSTART  1710
C                                                                       RSTART  1720
  200 WRITE (IT) IC1, PL, RC3, IC3, NSYM, RC6A, IFULL, DATE, COMENT,    RSTART  1730
     *           RC7, CGS, JS, RC8, RC9, IC9, IC10, NPANEL, SGTEST,     RSTART  1740
     *           RC13, IC13, IC14, DT, H0, HMAX, HMIN, NSTEPS, NDINT,   RSTART  1750
     *           RC17, IC17, IEULER, IC19, RC20A, RC21, IC21, HF, IC22, RSTART  1760
     *           RC23, IC23, RC24, IC24                                 RSTART  1770
      GO TO 999                                                         RSTART  1780
C                                                                       RSTART  1790
C        3. READ TIME POINT RECORD FROM OLD RESTART TAPE.               RSTART  1800
C                                                                       RSTART  1810
  300 READ  (IT) TIME, BELT, TPTS, BD, RC4, RC5B, RC6B, IFULL, IPIN,    RSTART  1820
     *           RC11, IC11, XTEST, SEGT, REGT, RC13H, KQTYPE, TAB,     RSTART  1830
     *           VAR, DER, NEQ, RC16, IC16, IEULER, RC18, IC19, RC20B,  RSTART  1840
     *           RC21, IC21, V4, RC23, NL, NPTPLY, WTIME, IWIND         RSTART  1850
      CALL OUTPUT(1)                                                    RSTART  1860
      GO TO 999                                                         RSTART  1870
C                                                                       RSTART  1880
C        5. WRITE TIME POINT RECORD ONTO NEW RESTART TAPE.              RSTART  1890
C                                                                       RSTART  1900
  500 WRITE (IT) TIME, BELT, TPTS, BD, RC4, RC5B, RC6B, IFULL, IPIN,    RSTART  1910
     *           RC11, IC11, XTEST, SEGT, REGT, RC13H, KQTYPE, TAB,     RSTART  1920
     *           VAR, DER, NEQ, RC16, IC16, IEULER, RC18, IC19, RC20B,  RSTART  1930
     *           RC21, IC21, V4, RC23, NL, NPTPLY, WTIME, IWIND         RSTART  1940
      GO TO 999                                                         RSTART  1950
C                                                                       RSTART  1960
C        4. READ NEW INPUT DATA FROM INPUT STREAM FOR RESTART.          RSTART  1970
C                                                                       RSTART  1980
  400 READ (5,399) AVAR,INDEX,ITYPE,RR,II,AA,RROLD,IIOLD,AAOLD         RSTART  1990
  399 FORMAT(A8,4I4,2(F8.0,I8,A8))                                      RSTART  2000
```

```
      CALL SEARCH(AVAR,INDEX,NCOM,ITEM)                                  RSTART  2010
      IF (NCOM.LE.0)    GO TO 490                                        RSTART  2020
      IF (NCOM.GT.24)   GO TO 999                                        RSTART  2030
      IF (ITYPE.GT.3)   GO TO 490                                        RSTART  2040
      GO TO ( 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12,            RSTART  2050
     *       13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24) , NCOM      RSTART  2060
C     COMMON /CONTRL/                                                    RSTART  2070
    1 IF (ITEM.GT.1)    GO TO 401                                        RSTART  2080
      IF (ITYPE.NE.1)   GO TO 490                                        RSTART  2090
      ROLD = TIME                                                        RSTART  2100
      TIME = RR                                                          RSTART  2110
      GO TO 492                                                          RSTART  2120
  401 IF (ITEM.GT.51)   GO TO 490                                        RSTART  2130
      IF (ITYPE.NE.2)   GO TO 490                                        RSTART  2140
      IOLD = IC1(ITEM-1)                                                 RSTART  2150
      IC1(ITEM-1) = II                                                   RSTART  2160
      GO TO 494                                                          RSTART  2170
C     COMMON /CNTSRF/                                                    RSTART  2180
    2 IF (ITEM.GT.1678) GO TO 490                                        RSTART  2190
      IF (ITYPE.NE.1)   GO TO 490                                        RSTART  2200
      ROLD = RC2(ITEM)                                                   RSTART  2210
      RC2(ITEM) = RR                                                     RSTART  2220
      GO TO 492                                                          RSTART  2230
C     COMMON /VPOSTN/                                                    RSTART  2240
    3 IF (ITEM.GT.3684) GO TO 403                                        RSTART  2250
      IF (ITYPE.NE.1)   GO TO 490                                        RSTART  2260
      ROLD = RC3(ITEM)                                                   RSTART  2270
      RC3(ITEM) = RR                                                     RSTART  2280
      GO TO 492                                                          RSTART  2290
  403 IF (ITEM.GT.3696) GO TO 490                                        RSTART  2300
      IF (ITYPE.NE.2)   GO TO 490                                        RSTART  2310
      IOLD = IC3(ITEM-3684)                                              RSTART  2320
      IC3(ITEM-3684) = II                                                RSTART  2330
      GO TO 494                                                          RSTART  2340
C     COMMON /SGMNTS/                                                    RSTART  2350
    4 IF (ITEM.GT.900 ) GO TO 404                                        RSTART  2360
      IF (ITYPE.NE.1)   GO TO 490                                        RSTART  2370
      ROLD = RC4(ITEM)                                                   RSTART  2380
      RC4(ITEM) = RR                                                     RSTART  2390
      GO TO 492                                                          RSTART  2400
  404 IF (ITEM.GT.930 ) GO TO 490                                        RSTART  2410
      IF (ITYPE.NE.2)   GO TO 490                                        RSTART  2420
      IOLD = NSYM(ITEM-900)                                              RSTART  2430
      NSYM(ITEM-900) = II                                                RSTART  2440
      GO TO 494                                                          RSTART  2450
C     COMMON /CMATRX/                                                    RSTART  2460
    5 IF (ITEM.GT.1506) GO TO 490                                        RSTART  2470
      IF (ITYPE.NE.1)   GO TO 490                                        RSTART  2480
      ROLD = RC5A(ITEM)                                                  RSTART  2490
      RC5A(ITEM) = RR                                                    RSTART  2500
```

```
      GO TO 492                                                          RSTART  2510
C     COMMON /ABDATA/                                                    RSTART  2520
    6 IF (ITEM.GT.881 ) GO TO 406                                        RSTART  2530
      IF (ITYPE.NE.1)   GO TO 490                                        RSTART  2540
      ROLD = RC6A(ITEM)                                                  RSTART  2550
      RC6A(ITEM) = RR                                                    RSTART  2560
      GO TO 492                                                          RSTART  2570
  406 IF (ITEM.GT.887 ) GO TO 490                                        RSTART  2580
      IF (ITYPE.NE.2)   GO TO 490                                        RSTART  2590
      IOLD = IFULL(ITEM-881)                                             RSTART  2600
      IFULL(ITEM-881) = II                                               RSTART  2610
      GO TO 494                                                          RSTART  2620
C     COMMON /TITLES/ NOTE: NO PROVISION FOR CGS OR JS.                  RSTART  2630
    7 IF (ITEM.GT.348 ) GO TO 490                                        RSTART  2640
      IF (ITYPE.NE.3)   GO TO 490                                        RSTART  2650
      AOLD = RC7A(ITEM)                                                  RSTART  2660
      RC7A(ITEM) = AA                                                    RSTART  2670
      GO TO 496                                                          RSTART  2680
C     COMMON /CNSNTS/                                                    RSTART  2690
    8 IF (ITEM.GT.34  ) GO TO 490                                        RSTART  2700
      IF (ITEM.GT.31  ) GO TO 408                                        RSTART  2710
      IF (ITEM.LE.28  ) GO TO 408                                        RSTART  2720
      IF (ITYPE.NE.3)   GO TO 490                                        RSTART  2730
      AOLD = RC8(ITEM)                                                   RSTART  2740
      RC8(ITEM) = AA                                                     RSTART  2750
      GO TO 496                                                          RSTART  2760
  408 IF (ITYPE.NE.1)   GO TO 490                                        RSTART  2770
      ROLD = RC8(ITEM)                                                   RSTART  2780
      RC8(ITEM) = RR                                                     RSTART  2790
      GO TO 492                                                          RSTART  2800
C     COMMON /DESCRP/                                                    RSTART  2810
    9 IF (ITEM.GT.2400) GO TO 409                                        RSTART  2820
      IF (ITYPE.NE.1)   GO TO 490                                        RSTART  2830
      ROLD = RC9(ITEM)                                                   RSTART  2840
      RC9(ITEM) = RR                                                     RSTART  2850
      GO TO 492                                                          RSTART  2860
  409 IF (ITEM.GT.2550) GO TO 490                                        RSTART  2870
      IF (ITYPE.NE.2)   GO TO 490                                        RSTART  2880
      IOLD = IC9(ITEM-2400)                                              RSTART  2890
      IC9(ITEM-2400) = II                                                RSTART  2900
      GO TO 494                                                          RSTART  2910
C     COMMON /JBARTZ/                                                    RSTART  2920
   10 IF (ITEM.GT.1614) GO TO 490                                        RSTART  2930
      IF (ITYPE.NE.2)   GO TO 490                                        RSTART  2940
      IOLD = IC10(ITEM)                                                  RSTART  2950
      IC10(ITEM) = II                                                    RSTART  2960
      GO TO 494                                                          RSTART  2970
C     COMMON /FORCES/                                                    RSTART  2980
   11 IF (ITEM.GT.760)  GO TO 411                                        RSTART  2990
      IF (ITYPE.NE.1)   GO TO 490                                        RSTART  3000
```

```
      ROLD = RC11(ITEM)                                                 RSTART  3010
      RC11(ITEM) = RR                                                   RSTART  3020
      GO TO 492                                                         RSTART  3030
  411 IF (ITEM.GT.769)  GO TO 490                                       RSTART  3040
      IF (ITYPE.NE.2)   GO TO 490                                       RSTART  3050
      IOLD = IC11(ITEM-760)                                             RSTART  3060
      IC11(ITEM-760) = II                                               RSTART  3070
      GO TO 494                                                         RSTART  3080
C     COMMON /INTEST/                                                   RSTART  3090
   12 IF (ITEM.GT.720 ) GO TO 412                                       RSTART  3100
      IF (ITYPE.NE.1)   GO TO 490                                       RSTART  3110
      ROLD = RC12(ITEM)                                                 RSTART  3120
      RC12(ITEM) = RR                                                   RSTART  3130
      GO TO 492                                                         RSTART  3140
  412 IF (ITEM.GT.840 ) GO TO 512                                       RSTART  3150
      IF (ITYPE.NE.3)   GO TO 490                                       RSTART  3160
      AOLD = SEGT(ITEM-720)                                             RSTART  3170
      SEGT(ITEM-720) = AA                                               RSTART  3180
      GO TO 496                                                         RSTART  3190
  512 IF (ITEM.GT.960 ) GO TO 490                                       RSTART  3200
      IF (ITYPE.NE.3)   GO TO 490                                       RSTART  3210
      AOLD = REGT(ITEM-840)                                             RSTART  3220
      REGT(ITEM-840) = AA                                               RSTART  3230
      GO TO 496                                                         RSTART  3240
C     COMMON /CSTRNT/                                                   RSTART  3250
   13 IF (ITEM.GT.1212) GO TO 413                                       RSTART  3260
      IF (ITYPE.NE.1)   GO TO 490                                       RSTART  3270
      ROLD = RC13A(ITEM)                                                RSTART  3280
      RC13A(ITEM) = RR                                                  RSTART  3290
      GO TO 492                                                         RSTART  3300
  413 IF (ITEM.GT.1248) GO TO 490                                       RSTART  3310
      IF (ITYPE.NE.2)   GO TO 490                                       RSTART  3320
      IOLD = IC13(ITEM-1212)                                            RSTART  3330
      IC13(ITEM-1212) = II                                              RSTART  3340
      GO TO 494                                                         RSTART  3350
C     COMMON /TABLES/                                                   RSTART  3360
   14 IF (ITEM.GT.554 ) GO TO 414                                       RSTART  3370
      IF (ITYPE.NE.2)   GO TO 490                                       RSTART  3380
      IOLD = IC14(ITEM)                                                 RSTART  3390
      IC14(ITEM) = II                                                   RSTART  3400
      GO TO 494                                                         RSTART  3410
  414 IF (ITEM.GT.3154) GO TO 490                                       RSTART  3420
      IF (ITYPE.NE.1)   GO TO 490                                       RSTART  3430
      ROLD = TAB(ITEM-554)                                              RSTART  3440
      TAB(ITEM-554) = RR                                                RSTART  3450
      GO TO 492                                                         RSTART  3460
C     COMMON /COMAIN/                                                   RSTART  3470
   15 IF (ITEM.GT.485 ) GO TO 415                                       RSTART  3480
      IF (ITYPE.NE.1)   GO TO 490                                       RSTART  3490
      ROLD = RC15(ITEM)                                                 RSTART  3500
```

```
      RC15(ITEM) = RR                                                   RSTART  3510
      GO TO 492                                                         RSTART  3520
  415 IF (ITEM.GT.491 ) GO TO 490                                       RSTART  3530
      IF (ITYPE.NE.2)   GO TO 490                                       RSTART  3540
      IOLD = IC15(ITEM-485)                                             RSTART  3550
      IC15(ITEM-485) = II                                               RSTART  3560
      GO TO 494                                                         RSTART  3570
C     COMMON /CDINT /                                                   RSTART  3580
   16 IF (ITEM.GT.5541) GO TO 416                                       RSTART  3590
      IF (ITYPE.NE.1)   GO TO 490                                       RSTART  3600
      ROLD = RC16(ITEM)                                                 RSTART  3610
      RC16(ITEM) = RR                                                   RSTART  3620
      GO TO 492                                                         RSTART  3630
  416 IF (ITEM.GT.5544) GO TO 490                                       RSTART  3640
      IF (ITYPE.NE.2)   GO TO 490                                       RSTART  3650
      IOLD = IC16(ITEM-5541)                                            RSTART  3660
      IC16(ITEM-5541) = II                                              RSTART  3670
      GO TO 494                                                         RSTART  3680
C     COMMON /DAMPER/                                                   RSTART  3690
   17 IF (ITEM.GT.220 ) GO TO 417                                       RSTART  3700
      IF (ITYPE.NE.1)   GO TO 490                                       RSTART  3710
      ROLD = RC17(ITEM)                                                 RSTART  3720
      RC17(ITEM) = RR                                                   RSTART  3730
      GO TO 492                                                         RSTART  3740
  417 IF (ITEM.GT.260 ) GO TO 490                                       RSTART  3750
      IF (ITYPE.NE.2)   GO TO 490                                       RSTART  3760
      IOLD = IC17(ITEM-220)                                             RSTART  3770
      IC17(ITEM-220) = II                                               RSTART  3780
      GO TO 494                                                         RSTART  3790
C     COMMON /CEULER/                                                   RSTART  3800
   18 IF (ITEM.GT.30  ) GO TO 418                                       RSTART  3810
      IF (ITYPE.NE.2)   GO TO 490                                       RSTART  3820
      IOLD = IEULER(ITEM)                                               RSTART  3830
      IEULER(ITEM) = II                                                 RSTART  3840
      GO TO 494                                                         RSTART  3850
  418 IF (ITEM.GT.750 ) GO TO 490                                       RSTART  3860
      IF (ITYPE.NE.1)   GO TO 490                                       RSTART  3870
      ROLD = RC18(ITEM-30)                                              RSTART  3880
      RC18(ITEM-30) = RR                                                RSTART  3890
      GO TO 492                                                         RSTART  3900
C     COMMON /TEMPVI/                                                   RSTART  3910
   19 IF (ITEM.GT.10  ) GO TO 419                                       RSTART  3920
      IF (ITYPE.NE.1)   GO TO 490                                       RSTART  3930
      ROLD = RC19(ITEM)                                                 RSTART  3940
      RC19(ITEM) = RR                                                   RSTART  3950
      GO TO 492                                                         RSTART  3960
  419 IF (ITEM.GT.190 ) GO TO 490                                       RSTART  3970
      IF (ITYPE.NE.2)   GO TO 490                                       RSTART  3980
      IOLD = IC19(ITEM-10)                                              RSTART  3990
      IC19(ITEM-10) = II                                                RSTART  4000
```

```
      GO TO 494                                                         RSTART  4010
C     COMMON/CYDATA/                                                    RSTART  4020
   20 IF (ITEM.GT.115 ) GO TO 490                                       RSTART  4030
      IF (ITYPE.NE.1)   GO TO 490                                       RSTART  4040
      ROLD = RC20A(ITEM)                                                RSTART  4050
      RC20A(ITEM) = RR                                                  RSTART  4060
      GO TO 492                                                         RSTART  4070
C     COMMON /RSAVE/                                                    RSTART  4080
   21 IF (ITEM.GT.450 ) GO TO 421                                       RSTART  4090
      IF (ITYPE.NE.1  ) GO TO 490                                       RSTART  4100
      ROLD = RC21(ITEM)                                                 RSTART  4110
      RC21(ITEM) = RR                                                   RSTART  4120
      GO TO 492                                                         RSTART  4130
  421 IF (ITEM.GT.627 ) GO TO 490                                       RSTART  4140
      IF (ITYPE.NE.2  ) GO TO 490                                       RSTART  4150
      IOLD = IC21(ITEM-450)                                             RSTART  4160
      IC21(ITEM-450) = II                                               RSTART  4170
      GO TO 494                                                         RSTART  4180
C     COMMON /FLXBLE/                                                   RSTART  4190
   22 IF (ITEM.GT.624 ) GO TO 422                                       RSTART  4200
      IF (ITYPE.NE.1  ) GO TO 490                                       RSTART  4210
      ROLD = RC22(ITEM)                                                 RSTART  4220
      RC22(ITEM) = RR                                                   RSTART  4230
      GO TO 492                                                         RSTART  4240
  422 IF (ITEM.GT.648 ) GO TO 490                                       RSTART  4250
      IF (ITYPE.NE.2  ) GO TO 490                                       RSTART  4260
      IOLD = IC22(ITEM-624)                                             RSTART  4270
      IC22(ITEM-624) = II                                               RSTART  4280
      GO TO 494                                                         RSTART  4290
C     COMMON /HRNESS/                                                   RSTART  4300
   23 IF (ITEM.GT.1922) GO TO 423                                       RSTART  4310
      IF (ITYPE.NE.1)   GO TO 490                                       RSTART  4320
      ROLD = RC23(ITEM)                                                 RSTART  4330
      RC23(ITEM) = RR                                                   RSTART  4340
      GO TO 492                                                         RSTART  4350
  423 IF (ITEM.GT.2687) GO TO 490                                       RSTART  4360
      IF (ITYPE.NE.2)   GO TO 490                                       RSTART  4370
      IOLD = IC23(ITEM-1922)                                            RSTART  4380
      IC23(ITEM-1922) = II                                              RSTART  4390
      GO TO 494                                                         RSTART  4400
C     COMMON /WINDFR/                                                   RSTART  4410
   24 IF (ITEM.GT.60  ) GO TO 424                                       RSTART  4420
      IF (ITYPE.NE.1)   GO TO 490                                       RSTART  4430
      ROLD = RC24(ITEM)                                                 RSTART  4440
      RC24(ITEM) = RR                                                   RSTART  4450
      GO TO 492                                                         RSTART  4460
  424 IF (ITEM.GT.251 ) GO TO 490                                       RSTART  4470
      IF (ITYPE.NE.2)   GO TO 490                                       RSTART  4480
      IOLD = IC24(ITEM-60)                                              RSTART  4490
      IC24(ITEM-60) = II                                                RSTART  4500
```

```
      GO TO 494                                                          RSTART  4510
C                                                                        RSTART  4520
C     ERROR MESSAGE - TERMINATE PROGRAM.                                 RSTART  4530
C                                                                        RSTART  4540
  490 WRITE (6,491) AVAR,INDEX,NCOM,ITEM,ITYPE,RR,II,AA                  RSTART  4550
  491 FORMAT('0 SUBROUTINE RSTART INPUT ERROR'//                         RSTART  4560
     *       ' AVAR= ',A8,'INDEX=',3I6,' NCOM=',I6,' ITEM=',I6,          RSTART  4570
     * ' ITYPE=',I6,' RR=',G15.8,' II=',I8,' AA= ',A8//                  RSTART  4580
     * ' PROGRAM IS BEING TERMINATED.')                                  RSTART  4590
      STOP 2                                                             RSTART  4600
C                                                                        RSTART  4610
C     PRINT MESSAGE FOR REAL VARIABLES.                                  RSTART  4620
C                                                                        RSTART  4630
  492 WRITE (6,493) AVAR,INDEX,COMMON(NCOM),ROLD,RR                      RSTART  4640
  493 FORMAT('0',A6,'(',I4,',',I4,',',I4,') OF COMMON/',A6,'/',          RSTART  4650
     *       ' HAS BEEN CHANGED FROM ',G15.8,' TO ',G15.8)               RSTART  4660
      IF (RROLD.EQ.0.0) GO TO 400                                        RSTART  4670
      IF (DABS(RROLD-ROLD).LE.0.00001*RROLD) GO TO 400                   RSTART  4680
      WRITE (6,383) RROLD                                                RSTART  4690
  383 FORMAT(' INPUT VALUE FOR RROLD WAS ',G15.8//)                      RSTART  4700
      GO TO 490                                                          RSTART  4710
C                                                                        RSTART  4720
C     PRINT MESSAGE FOR INTEGER VARIABLES.                               RSTART  4730
C                                                                        RSTART  4740
  494 WRITE (6,495) AVAR,INDEX,COMMON(NCOM),IOLD,II                      RSTART  4750
  495 FORMAT('0',A6,'(',I4,',',I4,',',I4,') OF COMMON/',A6,'/',          RSTART  4760
     *       ' HAS BEEN CHANGED FROM ',   I8,' TO ',   I8)             RSTART  4770
      IF (IIOLD.EQ.0) GO TO 400                                          RSTART  4780
      IF (IOLD.EQ.IIOLD) GO TO 400                                       RSTART  4790
      WRITE (6,385) IIOLD                                                RSTART  4800
  385 FORMAT(' INPUT VALUE FOR IIOLD WAS ',I8//)                         RSTART  4810
      GO TO 490                                                          RSTART  4820
C                                                                        RSTART  4830
C     PRINT MESSAGE FOR ALPHANUMERIC VARIABLES.                          RSTART  4840
C                                                                        RSTART  4850
  496 WRITE (6,497) AVAR,INDEX,COMMON(NCOM),AOLD,AA                      RSTART  4860
  497 FORMAT('0',A6,'(',I4,',',I4,',',I4,') OF COMMON/',A6,'/',          RSTART  4870
     *       ' HAS BEEN CHANGED FROM ',   A8,' TO ',   A8)             RSTART  4880
      IF (AAOLD.EQ.BLANK) GO TO 400                                      RSTART  4890
      AAOLD4 = AAOLD                                                     RSTART  4900
      AOLD4  = AOLD                                                      RSTART  4910
      IF (AOLD4.EQ.AAOLD4) GO TO 400                                     RSTART  4920
      WRITE (6,387) AAOLD                                                RSTART  4930
  387 FORMAT(' INPUT VALUE FOR AAOLD WAS ',A8//)                         RSTART  4940
      GO TO 490                                                          RSTART  4950
  999 CALL ELTIME(2,25)                                                  RSTART  4960
      RETURN                                                             RSTART  4970
      END                                                                RSTART  4980
```

```
      SUBROUTINE SEARCH(AVAR,INDEX,NCOM,ITEM)                                SEARCH  0010
C                                                          REV 20 05/23/80    SEARCH  0020
C     CALLED BY SUBROUTINE RSTART TO COMPUTE NCOM & ITEM FROM AVAR &          SEARCH  0030
C     INDEX. RETURNS NCOM=0 FOR ERROR AND NCOM=50 FOR BLANK.                  SEARCH  0040
C                                                                            SEARCH  0050
      IMPLICIT REAL*8(A-H,O-Z)                                               SEARCH  0060
      DIMENSION BVAR(257),KOUNT(25),NDIM(3,257),NJ(3),NK(3),INDEX(3)         SEARCH  0070
      DIMENSION C1 (  16) , NC1 (  48)                                       SEARCH  0080
      DIMENSION C2 (   4) , NC2 (  12)                                       SEARCH  0090
      DIMENSION C3 (  10) , NC3 (  30)                                       SEARCH  0100
      DIMENSION C4 (   9) , NC4 (  27)                                       SEARCH  0110
      DIMENSION C5 (   8) , NC5 (  24)                                       SEARCH  0120
      DIMENSION C6 (  30) , NC6 (  90)                                       SEARCH  0130
      DIMENSION C7 (  11) , NC7 (  33)                                       SEARCH  0140
      DIMENSION C8 (   9) , NC8 (  27)                                       SEARCH  0150
      DIMENSION C9 (  15) , NC9 (  45)                                       SEARCH  0160
      DIMENSION C10(  11) , NC10(  33)                                       SEARCH  0170
      DIMENSION C11(  10) , NC11(  30)                                       SEARCH  0180
      DIMENSION C12(   4) , NC12(  12)                                       SEARCH  0190
      DIMENSION C13(  16) , NC13(  48)                                       SEARCH  0200
      DIMENSION C14(   7) , NC14(  21)                                       SEARCH  0210
      DIMENSION C15(  13) , NC15(  39)                                       SEARCH  0220
      DIMENSION C16(  15) , NC16(  45)                                       SEARCH  0230
      DIMENSION C17(   5) , NC17(  15)                                       SEARCH  0240
      DIMENSION C18(   7) , NC18(  21)                                       SEARCH  0250
      DIMENSION C19(   5) , NC19(  15)                                       SEARCH  0260
      DIMENSION C20(  23) , NC20(  69)                                       SEARCH  0270
      DIMENSION C21(   5) , NC21(  15)                                       SEARCH  0280
      DIMENSION C22(   4) , NC22(  12)                                       SEARCH  0290
      DIMENSION C23(  12) , NC23(  36)                                       SEARCH  0300
      DIMENSION C24(   7) , NC24(  21)                                       SEARCH  0310
C                                                                            SEARCH  0320
      EQUIVALENCE (C1 (1),BVAR(  1)) , (NC1 (1),NDIM(1,  1))                 SEARCH  0330
      EQUIVALENCE (C2 (1),BVAR( 17)) , (NC2 (1),NDIM(1, 17))                 SEARCH  0340
      EQUIVALENCE (C3 (1),BVAR( 21)) , (NC3 (1),NDIM(1, 21))                 SEARCH  0350
      EQUIVALENCE (C4 (1),BVAR( 31)) , (NC4 (1),NDIM(1, 31))                 SEARCH  0360
      EQUIVALENCE (C5 (1),BVAR( 40)) , (NC5 (1),NDIM(1, 40))                 SEARCH  0370
      EQUIVALENCE (C6 (1),BVAR( 48)) , (NC6 (1),NDIM(1, 48))                 SEARCH  0380
      EQUIVALENCE (C7 (1),BVAR( 78)) , (NC7 (1),NDIM(1, 78))                 SEARCH  0390
      EQUIVALENCE (C8 (1),BVAR( 89)) , (NC8 (1),NDIM(1, 89))                 SEARCH  0400
      EQUIVALENCE (C9 (1),BVAR( 98)) , (NC9 (1),NDIM(1, 98))                 SEARCH  0410
      EQUIVALENCE (C10(1),BVAR(113)) , (NC10(1),NDIM(1,113))                 SEARCH  0420
      EQUIVALENCE (C11(1),BVAR(124)) , (NC11(1),NDIM(1,124))                 SEARCH  0430
      EQUIVALENCE (C12(1),BVAR(134)) , (NC12(1),NDIM(1,134))                 SEARCH  0440
      EQUIVALENCE (C13(1),BVAR(138)) , (NC13(1),NDIM(1,138))                 SEARCH  0450
      EQUIVALENCE (C14(1),BVAR(154)) , (NC14(1),NDIM(1,154))                 SEARCH  0460
      EQUIVALENCE (C15(1),BVAR(161)) , (NC15(1),NDIM(1,161))                 SEARCH  0470
      EQUIVALENCE (C16(1),BVAR(174)) , (NC16(1),NDIM(1,174))                 SEARCH  0480
      EQUIVALENCE (C17(1),BVAR(189)) , (NC17(1),NDIM(1,189))                 SEARCH  0490
      EQUIVALENCE (C18(1),BVAR(194)) , (NC18(1),NDIM(1,194))                 SEARCH  0500
```

```
      EQUIVALENCE (C19(1),BVAR(201)) , (NC19(1),NDIM(1,201))                SEARCH  0510
      EQUIVALENCE (C20(1),BVAR(206)) , (NC20(1),NDIM(1,206))                SEARCH  0520
      EQUIVALENCE (C21(1),BVAR(229)) , (NC21(1),NDIM(1,229))                SEARCH  0530
      EQUIVALENCE (C22(1),BVAR(234)) , (NC22(1),NDIM(1,234))                SEARCH  0540
      EQUIVALENCE (C23(1),BVAR(238)) , (NC23(1),NDIM(1,238))                SEARCH  0550
      EQUIVALENCE (C24(1),BVAR(250)) , (NC24(1),NDIM(1,250))                SEARCH  0560
C                                                                           SEARCH  0570
      DATA  NVAR/257/ , KOM/24/ , BLANK/8H        /                        SEARCH  0580
      DATA KOUNT/1,17,21,31,40,48,78,89,98,113,124,134,138,154,             SEARCH  0590
     *           161,174,189,194,201,206,229,234,238,250,257/               SEARCH  0600
C                                                                           SEARCH  0610
C  1  COMMON/CONTRL/                                                        SEARCH  0620
C                                                                           SEARCH  0630
      DATA C1 / 8HTIME    ,8HNSEG    ,8HNJNT    ,8HNPL     ,8HNBLT    ,   SEARCH  0640
     *          8HNBAG    ,8HNVEH    ,8HNGRND   ,8HNS      ,8HNQ      ,   SEARCH  0650
     *          8HNSD     ,8HNFLX    ,8HNNRNSS  ,8HNWINDF  ,8HNJNTF   ,   SEARCH  0660
     *          8HNPRT    /                                               SEARCH  0670
      DATA NC1 /  0,0,0   ,  0,0,0   ,  0,0,0   ,  0,0,0   ,  0,0,0   ,   SEARCH  0680
     *            0,0,0   ,  0,0,0   ,  0,0,0   ,  0,0,0   ,  0,0,0   ,   SEARCH  0690
     *            0,0,0   ,  0,0,0   ,  0,0,0   ,  0,0,0   ,  0,0,0   ,   SEARCH  0700
     *            36,0,0  /                                               SEARCH  0710
C                                                                           SEARCH  0720
C  2  COMMON/CNTSRF/                                                        SEARCH  0730
C                                                                           SEARCH  0740
      DATA C2 / 8HPL      ,8HBELT    ,8HTPTS    ,8HBD      /              SEARCH  0750
      DATA NC2 /  17,30,0 ,  20,8,0  ,  6,8,0   ,  24,40,0 /              SEARCH  0760
C                                                                           SEARCH  0770
C  3  COMMON/VPOSTN/                                                        SEARCH  0780
C                                                                           SEARCH  0790
      DATA C3 / 8HZPLT    ,8HSPLT    ,8HAXV     ,8HVATAB   ,8HVTO     ,   SEARCH  0800
     *          8HVDT     ,8HTIMEV   ,8HOMEGV   ,8HNVTAB   ,8HINDXV   /   SEARCH  0810
      DATA NC3 /  3,0,0   ,  3,0,0   ,  3,6,0   ,  6,101,6 ,  6,0,0   ,   SEARCH  0820
     *            6,0,0   ,  6,0,0   ,  6,0,0   ,  6,0,0   ,  6,0,0   /   SEARCH  0830
C                                                                           SEARCH  0840
C  4  COMMON/SGMNTS/                                                        SEARCH  0850
C                                                                           SEARCH  0860
      DATA C4 / 8HD       ,8HWMEG    ,8HWMEGD   ,8HU1      ,8HU2      ,   SEARCH  0870
     *          8HSEGLP   ,8HSEGLV   ,8HSEGLA   ,8HNSYM    /              SEARCH  0880
      DATA NC4 /  3,3,30  ,  3,30,0  ,  3,30,0  ,  3,30,0  ,  3,30,0  ,   SEARCH  0890
     *            3,30,0  ,  3,30,0  ,  3,30,0  ,  30,0,0  /              SEARCH  0900
C                                                                           SEARCH  0910
C  5  COMMON/CMATRX/                                                        SEARCH  0920
C                                                                           SEARCH  0930
      DATA C5 / 8HV1      ,8HV2      ,8HV3      ,8HB12     ,8HA22     ,   SEARCH  0940
     *          8HF       ,8HTQ      ,8HWJ      /                         SEARCH  0950
      DATA NC5 /  3,30,0  ,  3,30,0  ,  3,12,0  ,  3,3,60  ,  3,3,60  ,   SEARCH  0960
     *            3,30,0  ,  3,30,0  ,  30,0,0  /                         SEARCH  0970
C                                                                           SEARCH  0980
C  6  COMMON/ABDATA/                                                        SEARCH  0990
C                                                                           SEARCH  1000
```

```
      DATA C6 / 8HZDEP    ,8HDBR     ,8HDPVCTR  ,8HDEPLOY  ,8HAB      ,     SEARCH  1010
     *          8HB       ,8HZR      ,8HBFB     ,8HDRR     ,8HVBAGG   ,     SEARCH  1020
     *          8HVSCS    ,8HSPRK    ,8HCK      ,8HCMASS   ,8HCYMIN   ,     SEARCH  1030
     *          8HCYMOUT  ,8HBAGPV   ,8HPD      ,8HVBAG    ,8HVOLBP   ,     SEARCH  1040
     *          8HPCYV    ,8HPCYMIN  ,8HPVBAG   ,8HTV1     ,8HTV2     ,     SEARCH  1050
     *          8HSWITCH  ,8HPYMOUT  ,8HSCALE   ,8HPREVT   ,8HIFULL   /     SEARCH  1060
      DATA NC6 /  3,5,0   ,  3,3,5   ,  3,5,0   ,  3,5,0   ,  3,5,0   ,     SEARCH  1070
     *            9,4,5   ,  3,4,5   ,  3,4,5   ,  9,4,5   ,  5,0,0   ,     SEARCH  1080
     *            5,0,0   ,  5,0,0   ,  5,0,0   ,  5,0,0   ,  5,0,0   ,     SEARCH  1090
     *            5,0,0   ,  5,0,0   ,  5,0,0   ,  5,0,0   ,  5,0,0   ,     SEARCH  1100
     *            5,0,0   ,  5,0,0   ,  5,0,0   ,  3,4,5   ,  3,10,5  ,     SEARCH  1110
     *            5,0,0   ,  5,0,0   ,  5,0,0   ,  0,0,0   ,  6,0,0   /     SEARCH  1120
C                                                                            SEARCH  1130
C  7  COMMON/TITLES/                                                         SEARCH  1140
C                                                                            SEARCH  1150
      DATA C7 / 8HDATE    ,8HCOMENT  ,8HVPSTTL  ,8HBDYTTL  ,8HBLTTTL  ,     SEARCH  1160
     *          8HPLTTL   ,8HBAGTTL  ,8HSEG     ,8HJOINT   ,8HCGS     ,     SEARCH  1170
     *          8HJS      /                                                  SEARCH  1180
      DATA NC7 /  3,0,0   ,  40,0,0  ,  20,0,0  ,  5,0,0   ,  5,8,0   ,     SEARCH  1190
     *            5,30,0  ,  5,6,0   ,  30,0,0  ,  30,0,0  ,  30,0,0  ,     SEARCH  1200
     *            30,0,0  /                                                  SEARCH  1210
C                                                                            SEARCH  1220
C  8  COMMON/CNSNTS/                                                         SEARCH  1230
C                                                                            SEARCH  1240
      DATA C8 / 8HPI      ,8HRADIAN  ,8HG       ,8HTHIRD   ,8HEPS     ,     SEARCH  1250
     *          8HUNITL   ,8HUNITM   ,8HUNITT   ,8HGRAVTY  /               SEARCH  1260
      DATA NC8 /  0,0,0   ,  0,0,0   ,  0,0,0   ,  0,0,0   ,  24,0,0  ,     SEARCH  1270
     *            0,0,0   ,  0,0,0   ,  0,0,0   ,  3,0,0   /               SEARCH  1280
C                                                                            SEARCH  1290
C  9  COMMON/DESCRP/                                                         SEARCH  1300
C                                                                            SEARCH  1310
      DATA C9 / 8HPHI     ,8HW       ,8HRW      ,8HSR      ,8HHA      ,     SEARCH  1320
     *          8HHB      ,8HRPHI    ,8HHT      ,8HSPRING  ,8HVISC    ,     SEARCH  1330
     *          8HJNT     ,8HJPIN    ,8HJSING   ,8HJGLOB   ,8HJOINTF  /     SEARCH  1340
      DATA NC9 /  3,30,0  ,  30,0,0  ,  30,0,0  ,  3,60,0  ,  3,60,0  ,     SEARCH  1350
     *            3,60,0  ,  3,30,0  ,  3,3,60  ,  5,90,0  ,  7,90,0  ,     SEARCH  1360
     *            30,0,0  ,  30,0,0  ,  30,0,0  ,  30,0,0  ,  30,0,0  /     SEARCH  1370
C                                                                            SEARCH  1380
C 10  COMMON/JBARTZ/                                                         SEARCH  1390
C                                                                            SEARCH  1400
      DATA C10/ 8HMNPL    ,8HMNBLT   ,8HMNSEG   ,8HMNBAG   ,8HMPL     ,     SEARCH  1410
     *          8HMBLT    ,8HMSEG    ,8HMBAG    ,8HNTPL    ,8HNTBLT   ,     SEARCH  1420
     *          8HNTSEG   /                                                  SEARCH  1430
      DATA NC10/  30,0,0  ,  8,0,0   ,  30,0,0  ,  6,0,0   ,  3,5,30  ,     SEARCH  1440
     *            3,5,8   ,  3,5,30  ,  3,10,6  ,  5,30,0  ,  5,8,0   ,     SEARCH  1450
     *            5,30,0  /                                                  SEARCH  1460
C                                                                            SEARCH  1470
C 11  COMMON/FORCES/                                                         SEARCH  1480
C                                                                            SEARCH  1490
      DATA C11/ 8HPSF     ,8HBSF     ,8HSSF     ,8HBAGSF   ,8HPRJNT   ,     SEARCH  1500
```

```
     *           8HNPANEL  ,8HNPSF    ,8HNBSF    ,8HNSSF    ,8HNBGSF   /    SEARCH  1510
      DATA NC11/  7,30,0   ,  4,20,0  ,  10,20,0 ,  3,20,0  ,  7,30,0  ,    SEARCH  1520
     *            5,0,0    ,  0,0,0   ,  0,0,0   ,  0,0,0   ,  0,0,0   /    SEARCH  1530
C                                                                             SEARCH  1540
C 12  COMMON/INTEST/                                                          SEARCH  1550
C                                                                             SEARCH  1560
      DATA C12/ 8HSGTEST ,8HXTEST   ,8HSEGT    ,8HREGT    /                  SEARCH  1570
      DATA NC12/  3,4,30 ,  3,120,0 ,  120,0,0 ,  120,0,0 /                  SEARCH  1580
C                                                                             SEARCH  1590
C 13  COMMON/CSTRNT/                                                          SEARCH  1600
C                                                                             SEARCH  1610
      DATA C13/ 8HA13     ,8HA23     ,8HB31     ,8HB32     ,8HHHT     ,    SEARCH  1620
     *          8HRK1     ,8HRK2     ,8HQQ      ,8HTQQ     ,8HRQQ     ,    SEARCH  1630
     *          8HHQQ     ,8HSQQ     ,8HCFQQ    ,8HKQ1     ,8HKQ2     ,    SEARCH  1640
     *          8HKQTYPE  /                                                 SEARCH  1650
      DATA NC13/  3,3,24  ,  3,3,24  ,  3,3,24  ,  3,3,24  ,  3,3,12  ,    SEARCH  1660
     *            3,12,0  ,  3,12,0  ,  3,12,0  ,  3,12,0  ,  3,12,0  ,    SEARCH  1670
     *            3,12,0  ,  12,0,0  ,  12,0,0  ,  12,0,0  ,  12,0,0  ,    SEARCH  1680
     *            12,0,0  /                                                 SEARCH  1690
C                                                                             SEARCH  1700
C 14  COMMON/TABLES/                                                          SEARCH  1710
C                                                                             SEARCH  1720
      DATA C14/ 8HMXNTI   ,8HMXNTB   ,8HMXTB1   ,8HMXTB2   ,8HNTI    ,    SEARCH  1730
     *          8HNTAB    ,8HTAB    /                                       SEARCH  1740
      DATA NC14/  0,0,0   ,  0,0,0   ,  0,0,0   ,  0,0,0   ,  50,0,0  ,    SEARCH  1750
     *            500,0,0 ,  2600,0,0/                                      SEARCH  1760
C                                                                             SEARCH  1770
C 15  COMMON/COMAIN/                                                          SEARCH  1780
C                                                                             SEARCH  1790
      DATA C15/ 8HVAR     ,8HDER     ,8HDT      ,8HH0      ,8HHMAX    ,    SEARCH  1800
     *          8HHMIN    ,8HRSTIME  ,8HISTEP   ,8HNSTEPS  ,8HNDINT   ,    SEARCH  1810
     *          8HNEQ     ,8HIRSIN   ,8HIRSOUT  /                         SEARCH  1820
      DATA NC15/  240,0,0 ,  240,0,0 ,  0,0,0   ,  0,0,0   ,  0,0,0   ,    SEARCH  1830
     *            0,0,0   ,  0,0,0   ,  0,0,0   ,  0,0,0   ,  0,0,0   ,    SEARCH  1840
     *            0,0,0   ,  0,0,0   ,  0,0,0   /                         SEARCH  1850
C                                                                             SEARCH  1860
C 16  COMMON/CDINT /                                                          SEARCH  1870
C                                                                             SEARCH  1880
      DATA C16/ 8HUU      ,8HGH      ,8HE       ,8HFF      ,8HGG      ,    SEARCH  1890
     *          8HY       ,8HU       ,8HH       ,8HHPRINT  ,8HHS      ,    SEARCH  1900
     *          8HTPRINT  ,8HTSTART  ,8HICNT    ,8HIDBL    ,8HIFLAG   /    SEARCH  1910
      DATA NC16/  4,0,0   ,  3,4,0   ,  3,240,0 ,  5,240,0 ,  5,240,0 ,    SEARCH  1920
     *            5,240,0 ,  5,240,0 ,  0,0,0   ,  0,0,0   ,  0,0,0   ,    SEARCH  1930
     *            0,0,0   ,  0,0,0   ,  0,0,0   ,  0,0,0   ,  0,0,0   /    SEARCH  1940
C                                                                             SEARCH  1950
C 17  COMMON/DAMPER/                                                          SEARCH  1960
C                                                                             SEARCH  1970
      DATA C17/ 8HAPSDM   ,8HAPSDN   ,8HASD     ,8HMSDM    ,8HMSDN    /    SEARCH  1980
      DATA NC17/  3,20,0  ,  3,20,0  ,  5,20,0  ,  20,0,0  ,  20,0,0  /    SEARCH  1990
C                                                                             SEARCH  2000
```

```
C 18  COMMON/CEULER/                                                    SEARCH  2010
C                                                                       SEARCH  2020
      DATA C18/ 8HIEULER  ,8HHIR     ,8HANG     ,8HANGD    ,8HFE      ,  SEARCH  2030
     *          8HTQE     ,8HCONST   /                                 SEARCH  2040
      DATA NC18/  30,0,0  ,  3,3,30  ,  3,30,0  ,  3,30,0  ,  3,30,0  ,  SEARCH  2050
     *            3,30,0  ,  3,30,0  /                                 SEARCH  2060
C                                                                       SEARCH  2070
C 19  COMMON/TEMPVI/                                                    SEARCH  2080
C                                                                       SEARCH  2090
      DATA C19/ 8HCREST   ,8HTTI     ,8HR1I     ,8HR2I     ,8HJSTOP   /  SEARCH  2100
      DATA NC19/  0,0,0   ,  3,0,0   ,  3,0,0   ,  3,0,0   ,  4,2,30  /  SEARCH  2110
C                                                                       SEARCH  2120
C 20  COMMON/CYDATA/                                                    SEARCH  2130
C                                                                       SEARCH  2140
      DATA C20/ 8HCYTD    ,8HCYPA    ,8HCYSP    ,8HCYTO    ,8HCYVO    ,  SEARCH  2150
     *          8HCYCD    ,8HCYK     ,8HCYR     ,8HCYAT    ,8HCYPV    ,  SEARCH  2160
     *          8HCYCDO   ,8HCYAO    ,8HCYPO    ,8HCYSS    ,8HCYLO    ,  SEARCH  2170
     *          8HCYC     ,8HCYRHOO  ,8HCYVMAX  ,8HCYORFC  ,8HCYRHO   ,  SEARCH  2180
     *          8HCYT     ,8HCYRHP   ,8HCYV     /                      SEARCH  2190
      DATA NC20/  5,0,0   ,  5,0,0   ,  5,0,0   ,  5,0,0   ,  5,0,0   ,  SEARCH  2200
     *            5,0,0   ,  5,0,0   ,  5,0,0   ,  5,0,0   ,  5,0,0   ,  SEARCH  2210
     *            5,0,0   ,  5,0,0   ,  5,0,0   ,  5,0,0   ,  5,0,0   ,  SEARCH  2220
     *            5,0,0   ,  5,0,0   ,  5,0,0   ,  5,0,0   ,  5,0,0   ,  SEARCH  2230
     *            5,0,0   ,  5,0,0   ,  5,0,0   /                      SEARCH  2240
C                                                                       SEARCH  2250
C 21  COMMON/RSAVE/                                                     SEARCH  2260
C                                                                       SEARCH  2270
      DATA C21/ 8HXSG     ,8HDPMI    ,8HLPMI    ,8HNSG     ,8HMSG     /  SEARCH  2280
      DATA NC21/  3,20,3  ,  3,3,30  ,  30,0,0  ,  7,0,0   ,  20,7,0  /  SEARCH  2290
C                                                                       SEARCH  2300
C 22  COMMON/FLXBLE/                                                    SEARCH  2310
C                                                                       SEARCH  2320
      DATA C22/ 8HHF      ,8HB42     ,8HV4      ,8HNFLEX   /           SEARCH  2330
      DATA NC22/  4,12,8  ,  3,3,24  ,  3,8,0   ,  3,8,0   /           SEARCH  2340
C                                                                       SEARCH  2350
C 23  COMMON/HRNESS/                                                    SEARCH  2360
C                                                                       SEARCH  2370
      DATA C23/ 8HBAR     ,8HBB      ,8HBBDOT   ,8HPLOSS   ,8HXLONG   ,  SEARCH  2380
     *          8HHTIME   ,8HIBAR    ,8HNL      ,8HNPTSPB  ,8HNPTPLY  ,  SEARCH  2390
     *          8HNTHRNS  ,8HNBLTPH  /                                 SEARCH  2400
      DATA NC23/  15,100,0,  100,0,0 ,  100,0,0 ,  2,100,0 ,  20,0,0  ,  SEARCH  2410
     *            2,0,0   ,  5,100,0 ,  2,100,0 ,  20,0,0  ,  20,0,0  ,  SEARCH  2420
     *            20,0,0  ,  5,0,0   /                                 SEARCH  2430
C                                                                       SEARCH  2440
C 24  COMMON/WINDFR/                                                    SEARCH  2450
      DATA C24/ 8HWTIME   ,8HQFU     ,8HQFV     ,8HIWIND   ,8HMWSEG   ,  SEARCH  2460
     *          8HNFVSEG  ,8HNFVNT   /                                 SEARCH  2470
      DATA NC24/  30,0,0  ,  3,5,0   ,  3,5,0   ,  30,0,0  ,  5,30,0  ,  SEARCH  2480
     *            6,0,0   ,  5,0,0   /                                 SEARCH  2490
      DATA NC24/  30,0,0  ,  30,0,0  ,  5,30,0  /                       SEARCH  2500
```

```
      NCOM = 50                                                          SEARCH  2510
      IF (AVAR.EQ.BLANK) GO TO 99                                        SEARCH  2520
C                                                                        SEARCH  2530
C     SEARCH FOR VARIABLE NO. IV.                                        SEARCH  2540
C                                                                        SEARCH  2550
      NCOM = 0                                                           SEARCH  2560
      DO 10 IV=1,NVAR                                                    SEARCH  2570
      IF (AVAR.EQ.BVAR(IV)) GO TO 12                                     SEARCH  2580
   10 CONTINUE                                                           SEARCH  2590
      GO TO 99                                                           SEARCH  2600
C                                                                        SEARCH  2610
C     SEARCH FOR COMMON NO. IC.                                          SEARCH  2620
C                                                                        SEARCH  2630
   12 DO 20 IC=1,KOM                                                     SEARCH  2640
      IF (IV.GE.KOUNT(IC).AND.IV.LT.KOUNT(IC+1)) GO TO 22                SEARCH  2650
   20 CONTINUE                                                           SEARCH  2660
      GO TO 99                                                           SEARCH  2670
C                                                                        SEARCH  2680
C     COMPUTE ITEM NO. FOR VARIABLE IV IN COMMON IC.                     SEARCH  2690
C                                                                        SEARCH  2700
   22 K1 = KOUNT(IC)                                                     SEARCH  2710
      K2 = IV-1                                                          SEARCH  2720
      ITEM = 1                                                           SEARCH  2730
      IF (K1.EQ.IV) GO TO 25                                             SEARCH  2740
      DO 24 K=K1,K2                                                      SEARCH  2750
      NI = 1                                                             SEARCH  2760
      DO 23 I=1,3                                                        SEARCH  2770
      IF (NDIM(I,K).NE.0) NI=NI*NDIM(I,K)                                SEARCH  2780
   23 CONTINUE                                                           SEARCH  2790
   24 ITEM = ITEM+NI                                                     SEARCH  2800
   25 DO 26 I=1,3                                                        SEARCH  2810
      IF (INDEX(I).EQ.0 .AND. NDIM(I,IV).NE.0) GO TO 99                  SEARCH  2820
      IF (NDIM(I,IV).EQ.0 .AND. INDEX(I).GT.1) GO TO 99                  SEARCH  2830
      NJ(I) = MAX0(INDEX(I)-1,0)                                         SEARCH  2840
      NK(I) = MAX0(NDIM(I,IV),1)                                         SEARCH  2850
      IF (NJ(I).GE.NK(I)) GO TO 99                                       SEARCH  2860
   26 CONTINUE                                                           SEARCH  2870
      ITEM = ITEM+NJ(1)+NJ(2)*NK(1)+NJ(3)*NK(2)*NK(1)                    SEARCH  2880
      NCOM = IC                                                          SEARCH  2890
   99 RETURN                                                             SEARCH  2900
      END                                                                SEARCH  2910
```

```
      SUBROUTINE SEGSEG(M,MM,N,NS,NT)                                    SEGSEG  0010
C                                                          REV 20 04/11/80 SEGSEG  0020
C     DETERMINES IF ELLIPSOID (MM) ATTACHED TO SEGMENT (M)                SEGSEG  0030
C     INTERSECTS    ELLIPSOID (NN) ATTACHED TO SEGMENT (N) AND            SEGSEG  0040
C     ADDS THE RESULTING FORCES AND TORQUES TO THE ELEMENTS OF THE U1     SEGSEG  0050
C     AND U2 ARRAYS FOR BOTH SEGMENTS. NOTE: NN = IABS(NS) AND IF NS < 0, SEGSEG  0060
C     ELLIPSOID (MM) IS ASSUMED TO BE INTERIOR TO ELLIPSOID (NN).         SEGSEG  0070
C                                                                         SEGSEG  0080
      IMPLICIT REAL*8(A-H,O-Z)                                            SEGSEG  0090
      COMMON/TABLES/ MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(500),TAB(2600)  SEGSEG  0100
      COMMON/CNTSRF/ PL(17,30),BELT(20,8),TPTS(6,8),BD(24,40)             SEGSEG  0110
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),  SEGSEG  0120
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)         SEGSEG  0130
      COMMON/FORCES/ PSF(7,30),BSF(4,20),SSF(10,20),BAGSF(3,20),          SEGSEG  0140
     *               PRJNT(7,30),NPANEL(5),NPSF,NBSF,NSSF,NBGSF           SEGSEG  0150
      COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),     SEGSEG  0160
     *               HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12),  SEGSEG  0170
     *               RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),                SEGSEG  0180
     *               KQ1(12),KQ2(12),KQTYPE(12)                           SEGSEG  0190
C     THIS COMMON/TEMPVS/ IS SHARED BY PLELP, PLSEGF AND SEGSEG.          SEGSEG  0200
      COMMON/TEMPVS/DMNT(3,3),TEMP(3,3),B(3,3),XMN(3),RLN(3),XMM(3),      SEGSEG  0210
     *              TM(3),R(3),RM(3),DMNWN(3),RLM(3),RN(3),VMN(3),VR(3),  SEGSEG  0220
     *              WMN(3),WCM(3),WCN(3),VREL(3),FFM(3),FR(3),TQM(3),     SEGSEG  0230
     *              TQN(3),TQNT(3),T(3),H(3),T1(3),T2(3),RMD(3),RND(3),   SEGSEG  0240
     *              TD(3),TT4(3,4),TT5(3,4),T3(3),T4(3),P,AMR,FM,CF,      SEGSEG  0250
     *              VRM,VRT,VRTS,VRTEST,TF,ELOSS,MCF,NCF                  SEGSEG  0260
      CALL ELTIME(1,23)                                                   SEGSEG  0270
C                                                                         SEGSEG  0280
C     COMPUTE ELLIPSOID MATRICES A AND B FOR SEGMENTS M AND N,            SEGSEG  0290
C     AND R, THE VECTOR FROM THE CENTER OF A TO THE CENTER OF B,          SEGSEG  0300
C     IN THE LOCAL REFERENCE OF SEGMENT M.                                SEGSEG  0310
C                                                                         SEGSEG  0320
      NN = IABS(NS)                                                       SEGSEG  0330
      CALL DOTT33(D(1,1,M),D(1,1,N),DMNT)                                 SEGSEG  0340
      CALL DOTT33(BD(7,NN),DMNT,TEMP)                                     SEGSEG  0350
      CALL MAT33(DMNT,TEMP,B)                                             SEGSEG  0360
      DO 10 I=1,3                                                         SEGSEG  0370
   10 XMN(I) = SEGLP(I,M)-SEGLP(I,N)                                      SEGSEG  0380
      CALL MAT31(D(1,1,M),XMN,XMM)                                        SEGSEG  0390
      CALL MAT31(DMNT,BD(4,NN),RLN)                                       SEGSEG  0400
      DO 11 I=1,3                                                         SEGSEG  0410
   11 R(I) = RLN(I) - XMM(I) - BD(I+3,MM)                                 SEGSEG  0420
C                                                                         SEGSEG  0430
C     TEST FOR INTERSECTION AND COMPUTE Y, THE POINT OF INTERSECTION.     SEGSEG  0440
C                                                                         SEGSEG  0450
      TB = 1.0                                                            SEGSEG  0460
      IF (NS.LT.0) TB = -1.0                                              SEGSEG  0470
      LT = NTAB(NT)                                                       SEGSEG  0480
      CALL INTERS(BD(7,MM),B,R,TB,RM,TAB(LT+22),TM)                       SEGSEG  0490
      MCF = NTAB(NT+1)                                                    SEGSEG  0500
```

```
      NCF = -MCF                                                        SEGSEG  0510
      IF (NCF.GT.0) CFQQ(NCF) = -999.                                   SEGSEG  0520
      IF (TB.GE.1.0) GO TO 99                                           SEGSEG  0530
      S1 = 0.0                                                          SEGSEG  0540
      S2 = 0.0                                                          SEGSEG  0550
      DO 13 I=1,3                                                       SEGSEG  0560
      RI = R(I)                                                         SEGSEG  0570
      IF (NS.LT.0) RI = RM(I) + TB*(RM(I)-R(I))                         SEGSEG  0580
      S1 = S1 + RI**2                                                   SEGSEG  0590
   13 S2 = S2 + TM(I)**2                                                SEGSEG  0600
      AMR = DSQRT(S2)                                                   SEGSEG  0610
      P = (1.0/TB-1.0)*DSQRT(S1)                                        SEGSEG  0620
      DO 14 I=1,3                                                       SEGSEG  0630
      TM(I) = -TM(I)/AMR                                                SEGSEG  0640
      T2(I) = RM(I) - R(I)                                              SEGSEG  0650
      RN(I) = T2(I) + RLN(I)                                            SEGSEG  0660
   14 RLM(I) = RM(I) + BD(I+3,MM)                                       SEGSEG  0670
      CALL DOT31(DMNT,RN,RLN)                                           SEGSEG  0680
      CALL PLSEGF(M,N,NT)                                               SEGSEG  0690
      IF (MCF.LT.0) GO TO 30                                            SEGSEG  0700
C                                                                       SEGSEG  0710
C     STORE DATA IN SSF ARRAY FOR OUTPUT                                SEGSEG  0720
C                                                                       SEGSEG  0730
      SSF(1,NSSF) = P                                                   SEGSEG  0740
      SSF(2,NSSF) = FM                                                  SEGSEG  0750
      TF2FM2 = TF**2 - FM**2                                            SEGSEG  0760
      IF (TF2FM2.LE.0.0) TF2FM2 = 0.0                                   SEGSEG  0770
      SSF(3,NSSF) = DSQRT(TF2FM2)                                       SEGSEG  0780
      SSF(4,NSSF) = TF                                                  SEGSEG  0790
      DO 24 I=1,3                                                       SEGSEG  0800
      SSF(I+4,NSSF) = RLM(I)                                            SEGSEG  0810
   24 SSF(I+7,NSSF) = RLN(I)                                            SEGSEG  0820
      GO TO 99                                                          SEGSEG  0830
   30 SSF(1,NSSF) = P                                                   SEGSEG  0840
      DO 31 I=1,3                                                       SEGSEG  0850
      SSF(I+1,NSSF) = T(I)                                              SEGSEG  0860
      SSF(I+4,NSSF) = RLM(I)                                            SEGSEG  0870
   31 SSF(I+7,NSSF) = RLN(I)                                            SEGSEG  0880
      ANR = XDY(TM,B,T2)                                                SEGSEG  0890
      CALL CROSS(TM,WMN,T2)                                             SEGSEG  0900
      CALL MAT31(BD(7,MM),VR,T1)                                        SEGSEG  0910
      DO 33 I=1,3                                                       SEGSEG  0920
      DO 32 J=1,3                                                       SEGSEG  0930
      K = I + 3*(J+1)                                                   SEGSEG  0940
      TT4(I,J) = BD(K,MM)/AMR + B(I,J)/ANR                              SEGSEG  0950
   32 TT5(I,J) = TT4(I,J)                                               SEGSEG  0960
      TT4(I,4) = T1(I)/AMR + T2(I)                                      SEGSEG  0970
   33 TT5(I,4) = TM(I)                                                  SEGSEG  0980
      CALL DSMSOL(TT4,3,3)                                              SEGSEG  0990
      CALL DSMSOL(TT5,3,3)                                              SEGSEG  1000
```

```
      ALP1 = TM(1)*TT4(1,4) + TM(2)*TT4(2,4) + TM(3)*TT4(3,4)                SEGSEG  1010
      ALP1 = ALP1 - AMR*VRM/(AMR+ANR)                                        SEGSEG  1020
      ALP2 = TM(1)*TT5(1,4) + TM(2)*TT5(2,4) + TM(3)*TT5(3,4)                SEGSEG  1030
      ALP  = ALP1/ALP2                                                       SEGSEG  1040
      DO 34 I=1,3                                                            SEGSEG  1050
      RND(I) = TT4(I,4) - ALP*TT5(I,4)                                       SEGSEG  1060
      RMD(I) = RND(I) - VR(I)                                                SEGSEG  1070
   34 T3(I) = VR(I) + VRM*TM(I)                                              SEGSEG  1080
      CALL CROSS(DMNWN,T3,T1)                                                SEGSEG  1090
      CALL CROSS (WMN,RMD,T3)                                                SEGSEG  1100
      CALL MAT31 (B,RND,T4)                                                  SEGSEG  1110
      SQQ(NCF) = 0.0                                                         SEGSEG  1120
      DO 36 I=1,3                                                            SEGSEG  1130
      SQQ(NCF) = SQQ(NCF) + TM(I)*(T3(I)+2.0*T1(I)) - VREL(I)*T4(I)/ANR      SEGSEG  1140
   36 T3(I) = T3(I) + T1(I) + T4(I)*VRM/ANR                                  SEGSEG  1150
      CALL DOT31(D(1,1,M),T3,RQQ(1,NCF))                                     SEGSEG  1160
   99 CALL ELTIME(2,23)                                                      SEGSEG  1170
      RETURN                                                                 SEGSEG  1180
      END                                                                    SEGSEG  1190
```

```
      SUBROUTINE SETUP1                                                      SETUP1  0010
C                                                          REV 20 04/26/80   SETUP1  0020
C     FOR KK=1 (BEFORE CONTACT ROUTINE IN DAUX)                              SETUP1  0030
C     SET UP INITIAL VALUES OF A2 AND B2 ARRAYS FOR THIS TIME POINT.         SETUP1  0040
C     SET UP INITIAL VALUES OF ARRAYS U1,U2 AND V1.                          SETUP1  0050
C                                                                            SETUP1  0060
      IMPLICIT REAL*8(A-H,O-Z)                                               SETUP1  0070
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,                SETUP1  0080
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)             SETUP1  0090
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),     SETUP1  0100
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)            SETUP1  0110
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(3,60),HA(3,60),HB(3,60),      SETUP1  0120
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),          SETUP1  0130
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)         SETUP1  0140
      COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60),     SETUP1  0150
     *               F(3,30),TQ(3,30),WJ(30)                                 SETUP1  0160
      COMMON/TEMPVS/T(3),S(3),T1(3),T2(3),T3(3),T4(3),T5(3),T6(3),           SETUP1  0170
     *              T7(3),T8(3),T9(3),T10(3),T11(3),T12(3),HH(3),            SETUP1  0180
     *              TT1(3,3),TT2(3,3),S1,SQS1,S2,S3,S4                       SETUP1  0190
C                                                                            SETUP1  0200
      CALL ELTIME(1,10)                                                      SETUP1  0210
      DO 20 I=1,NGRND                                                        SETUP1  0220
C                                                                            SETUP1  0230
C     SET EACH U1N = 0                                                       SETUP1  0240
C                                                                            SETUP1  0250
      U1(1,I) = 0.0                                                          SETUP1  0260
      U1(2,I) = 0.0                                                          SETUP1  0270
      U1(3,I) = 0.0                                                          SETUP1  0280
C                                                                            SETUP1  0290
C     SET EACH U2N = WNX(PHIN*WN)                                            SETUP1  0300
C                                                                            SETUP1  0310
      U2(1,I) = WMEG(2,I)*WMEG(3,I) * (PHI(2,I)-PHI(3,I))                    SETUP1  0320
      U2(2,I) = WMEG(1,I)*WMEG(3,I) * (PHI(3,I)-PHI(1,I))                    SETUP1  0330
   20 U2(3,I) = WMEG(1,I)*WMEG(2,I) * (PHI(1,I)-PHI(2,I))                    SETUP1  0340
      IF (NPRT(11).NE.0) WRITE (6,21) ((U2(I,J),I=1,3),J=1,NSEG)             SETUP1  0350
   21 FORMAT(' U2 ARRAY'/(1X,1P9D14.4))                                      SETUP1  0360
      IF (NJNT.LE.0) GO TO 98                                                SETUP1  0370
      DO 40 J=1,NJNT                                                         SETUP1  0380
      DO 31 K=1,3                                                            SETUP1  0390
   31 V1(K,J) = 0.0                                                          SETUP1  0400
      I = IABS(JNT(J))                                                       SETUP1  0410
      IF (I.LE.0) GO TO 40                                                   SETUP1  0420
C                                                                            SETUP1  0430
C     FOR EACH JOINT SET                                                     SETUP1  0440
C     B12(2J-1) = B12(J,I  ) = -D(I)' * SR(2J-1) X                           SETUP1  0450
C     B12(2J  ) = B12(J,J+1) =  D(J+1)' * SR(2J) X                           SETUP1  0460
C                                                                            SETUP1  0470
      B12(1,1,2*J-1) = D(3,1,I)*SR(2,2*J-1) - D(2,1,I)*SR(3,2*J-1)           SETUP1  0480
      B12(2,1,2*J-1) = D(3,2,I)*SR(2,2*J-1) - D(2,2,I)*SR(3,2*J-1)           SETUP1  0490
      B12(3,1,2*J-1) = D(3,3,I)*SR(2,2*J-1) - D(2,3,I)*SR(3,2*J-1)           SETUP1  0500
```

```
      B12(1,2,2*J-1) = D(1,1,I)*SR(3,2*J-1) - D(3,1,I)*SR(1,2*J-1)          SETUP1  0510
      B12(2,2,2*J-1) = D(1,2,I)*SR(3,2*J-1) - D(3,2,I)*SR(1,2*J-1)          SETUP1  0520
      B12(3,2,2*J-1) = D(1,3,I)*SR(3,2*J-1) - D(3,3,I)*SR(1,2*J-1)          SETUP1  0530
      B12(1,3,2*J-1) = D(2,1,I)*SR(1,2*J-1) - D(1,1,I)*SR(2,2*J-1)          SETUP1  0540
      B12(2,3,2*J-1) = D(2,2,I)*SR(1,2*J-1) - D(1,2,I)*SR(2,2*J-1)          SETUP1  0550
      B12(3,3,2*J-1) = D(2,3,I)*SR(1,2*J-1) - D(1,3,I)*SR(2,2*J-1)          SETUP1  0560
C                                                                           SETUP1  0570
      B12(1,1,2*J  ) = D(2,1,J+1)*SR(3,2*J) - D(3,1,J+1)*SR(2,2*J)          SETUP1  0580
      B12(2,1,2*J  ) = D(2,2,J+1)*SR(3,2*J) - D(3,2,J+1)*SR(2,2*J)          SETUP1  0590
      B12(3,1,2*J  ) = D(2,3,J+1)*SR(3,2*J) - D(3,3,J+1)*SR(2,2*J)          SETUP1  0600
      B12(1,2,2*J  ) = D(3,1,J+1)*SR(1,2*J) - D(1,1,J+1)*SR(3,2*J)          SETUP1  0610
      B12(2,2,2*J  ) = D(3,2,J+1)*SR(1,2*J) - D(1,2,J+1)*SR(3,2*J)          SETUP1  0620
      B12(3,2,2*J  ) = D(3,3,J+1)*SR(1,2*J) - D(1,3,J+1)*SR(3,2*J)          SETUP1  0630
      B12(1,3,2*J  ) = D(1,1,J+1)*SR(2,2*J) - D(2,1,J+1)*SR(1,2*J)          SETUP1  0640
      B12(2,3,2*J  ) = D(1,2,J+1)*SR(2,2*J) - D(2,2,J+1)*SR(1,2*J)          SETUP1  0650
      B12(3,3,2*J  ) = D(1,3,J+1)*SR(2,2*J) - D(2,3,J+1)*SR(1,2*J)          SETUP1  0660
C                                                                           SETUP1  0670
C     NOTE THAT FOR EACH JOINT                                              SETUP1  0680
C     A21(M,N) = B12(N,M)                                                   SETUP1  0690
C                                                                           SETUP1  0700
C     FOR EACH JOINT SET                                                    SETUP1  0710
C     V1(J) = -D(I)'*W(I)X( W(I)XSR(2J-1) )                                 SETUP1  0720
C             +D(J+1)'*W(J+1)X( W(J+1)XSR(2J) )                             SETUP1  0730
C                                                                           SETUP1  0740
      CALL CROSS(WMEG(1,I),SR(1,2*J-1),T)                                   SETUP1  0750
      CALL CROSS(WMEG(1,I),T,S)                                             SETUP1  0760
      CALL DOT31(D(1,1,I),S,V1(1,J))                                        SETUP1  0770
      CALL CROSS(WMEG(1,J+1),SR(1,2*J),T)                                   SETUP1  0780
      CALL CROSS(WMEG(1,J+1),T,S)                                           SETUP1  0790
      CALL DOT31(D(1,1,J+1),S,T)                                            SETUP1  0800
      DO 39 K=1,3                                                           SETUP1  0810
   39 V1(K,J) = T(K) - V1(K,J)                                              SETUP1  0820
   40 CONTINUE                                                              SETUP1  0830
      IF (NPRT(11).NE.0) WRITE (6,41) ((V1(I,J),I=1,3),J=1,NJNT)            SETUP1  0840
   41 FORMAT(' V1 ARRAY'/(1X,1P9D14.4))                                     SETUP1  0850
C                                                                           SETUP1  0860
C     IF IPIN(M)=1, SET V2(M)=(WN.HN-WM.HM)DN'WNXHN                         SETUP1  0870
C                                                                           SETUP1  0880
      DO 50 J=1,NJNT                                                        SETUP1  0890
      DO 43 K=1,3                                                           SETUP1  0900
   43 V2(K,J) = 0.0                                                         SETUP1  0910
      IF (IPIN(J).NE.1) GO TO 50                                            SETUP1  0920
      I = IABS(JNT(J))                                                      SETUP1  0930
      CALL CROSS (WMEG(1,I  ),HB(1,2*J-1),T)                                SETUP1  0940
      CALL DOT31 (D(1,1,I  ),T,T1)                                          SETUP1  0950
C     CALL CROSS (WMEG(1,J+1),HB(1,2*J  ),T)                                SETUP1  0960
C     CALL DOT31 (D(1,1,J+1),T,T2)                                          SETUP1  0970
      S1 = WMEG(1,I)*HB(1,2*J-1)                                            SETUP1  0980
     *   + WMEG(2,I)*HB(2,2*J-1)                                            SETUP1  0990
     *   + WMEG(3,I)*HB(3,2*J-1)                                            SETUP1  1000
      S2 = WMEG(1,J+1)*HB(1,2*J)                                            SETUP1  1010
     *   + WMEG(2,J+1)*HB(2,2*J)                                            SETUP1  1020
     *   + WMEG(3,J+1)*HB(3,2*J)                                            SETUP1  1030
      DO 44 K=1,3                                                           SETUP1  1040
C  44 V2(K,J) = S1*T1(K) - S2*T2(K)                                         SETUP1  1050
   44 V2(K,J) = (S1-S2)*T1(K)                                               SETUP1  1060
   50 CONTINUE                                                              SETUP1  1070
   98 CALL ELTIME(2,10)                                                     SETUP1  1080
      RETURN                                                                SETUP1  1090
      END                                                                   SETUP1  1100
```

```
      SUBROUTINE SETUP2                                                 SETUP2  0010
C                                                        REV 19 08/05/78  SETUP2  0020
C     CALLED BY DAUX AFTER CONTACT ROUTINES AND BY UPDATE PRIOR TO       SETUP2  0030
C     DAUX TO SET UP A2 ARRAY AND (FOR NQ#0) THE A13,A23 AND V3 ARRAYS.  SETUP2  0040
C                                                                        SETUP2  0050
      IMPLICIT REAL*8(A-H,O-Z)                                          SETUP2  0060
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           SETUP2  0070
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)        SETUP2  0080
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30), SETUP2  0090
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       SETUP2  0100
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(3,60),HA(3,60),HB(3,60), SETUP2  0110
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),     SETUP2  0120
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)    SETUP2  0130
      COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60), SETUP2  0140
     *               F(3,30),TQ(3,30),WJ(30)                            SETUP2  0150
      COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),   SETUP2  0160
     *               HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12),SETUP2  0170
     *               RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),              SETUP2  0180
     *               KQ1(12),KQ2(12),KQTYPE(12)                         SETUP2  0190
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         SETUP2  0200
     *               UNITL,UNITM,UNITT,GRAVTY(3)                        SETUP2  0210
      COMMON/TEMPVS/T(3),S(3),T1(3),T2(3),T3(3),T4(3),T5(3),T6(3),      SETUP2  0220
     *              T7(3),T8(3),T9(3),T10(3),T11(3),T12(3),HH(3),       SETUP2  0230
     *              TT1(3,3),TT2(3,3),S1,SQS1,S2,S3,S4                  SETUP2  0240
     *              ,WCRM(3),RM(3),WCM(3),WWCM(3),WWM(3),RBA (3),BA     SETUP2  0250
     *              ,WCRN(3),RN(3),WCN(3),WWCN(3),WWN(3),RBAD(3)        SETUP2  0260
C                                                                       SETUP2  0270
      CALL ELTIME(1,26)                                                 SETUP2  0280
C                                                                       SETUP2  0290
C     COMPUTE A22 ARRAY VIA DHHPIN FOR DAUX2 ROUTINES.                  SETUP2  0300
C                                                                       SETUP2  0310
      IF (NJNT.EQ.0) GO TO 50                                           SETUP2  0320
      DO 49 M=1,NJNT                                                    SETUP2  0330
      N = IABS(JNT(M))                                                  SETUP2  0340
      IF (N.EQ.0) GO TO 49                                              SETUP2  0350
      IF (IPIN(M).EQ.0 .OR. IPIN(M).GE.2) GO TO 49                      SETUP2  0360
      CALL DHHPIN(A22(1,1,2*M-1),T,N  ,M,2*M-1)                         SETUP2  0370
      CALL DHHPIN(A22(1,1,2*M  ),T,M+1,M,2*M  )                         SETUP2  0380
   49 CONTINUE                                                          SETUP2  0390
C                                                                       SETUP2  0400
C     SET UP A13,A23 AND V3 ARRAYS FOR DAUX33.                          SETUP2  0410
C                                                                       SETUP2  0420
   50 IF (NQ.EQ.0) GO TO 98                                             SETUP2  0430
      DO 70 K=1,NQ                                                      SETUP2  0440
      IF (KQTYPE(K).LT.0) GO TO 70                                      SETUP2  0450
      IF (KQTYPE(K).EQ.5) GO TO 70                                      SETUP2  0460
      M = KQ1(K)                                                        SETUP2  0470
      N = KQ2(K)                                                        SETUP2  0480
      IF (KQTYPE(K).EQ.2 .OR. KQTYPE(K).EQ.4) GO TO 53                  SETUP2  0490
C                                                                       SETUP2  0500
```

```
C     FOR KQTYPE = 1 OR 3, SET HHT = I                                  SETUP2  0510
C                                                                       SETUP2  0520
      DO 52 J=1,3                                                       SETUP2  0530
      DO 51 I=1,3                                                       SETUP2  0540
   51 HHT(I,J,K) = 0.0                                                  SETUP2  0550
   52 HHT(J,J,K) = 1.0                                                  SETUP2  0560
      IF (KQTYPE(K).NE.6) GO TO 61                                      SETUP2  0570
C                                                                       SETUP2  0580
C     FOR KQTYPE=6, SET HHT= I-TT'                                      SETUP2  0590
C                                                                       SETUP2  0600
      DO 60 J=1,3                                                       SETUP2  0610
      DO 60 I=1,3                                                       SETUP2  0620
   60 HHT(I,J,K) = HHT(I,J,K) - TQQ(I,K)*TQQ(J,K)                       SETUP2  0630
      GO TO 61                                                          SETUP2  0640
   53 IF (KQTYPE(K).NE.2) GO TO 56                                      SETUP2  0650
C                                                                       SETUP2  0660
C     FOR KQTYPE=2, COMPUTE HH AND HHT.                                 SETUP2  0670
C                                                                       SETUP2  0680
      CALL DOT31(D(1,1,M),RK1(1,K),T1)                                  SETUP2  0690
      CALL DOT31(D(1,1,N),RK2(1,K),T2)                                  SETUP2  0700
      S1 = 0.0                                                          SETUP2  0710
      DO 54 I=1,3                                                       SETUP2  0720
      HH(I) = SEGLP(I,M)+T1(I) - SEGLP(I,N)-T2(I)                       SETUP2  0730
   54 S1 = S1 + HH(I)**2                                                SETUP2  0740
      SQS1 = DSQRT(S1)                                                  SETUP2  0750
      DO 55 I=1,3                                                       SETUP2  0760
      HH(I) = HH(I)/SQS1                                                SETUP2  0770
   55 IF (DABS(HH(I)).LE.EPS(12)) HH(I) = 0.0                           SETUP2  0780
      CALL DOTT31(HH,HH,HHT(1,1,K))                                     SETUP2  0790
   56 IF (KQTYPE(K).NE.4) GO TO 61                                      SETUP2  0800
C                                                                       SETUP2  0810
C     FOR KQTYPE = 4, SET HHT = HHT                                     SETUP2  0820
C                                                                       SETUP2  0830
      CALL DOTT31(HQQ(1,K),HQQ(1,K),HHT(1,1,K))                         SETUP2  0840
C                                                                       SETUP2  0850
C     SET A13(2K-1) =  HHT                                              SETUP2  0860
C     AND A13(2K)   = -HHT                                              SETUP2  0870
C                                                                       SETUP2  0880
   61 DO 62 J=1,3                                                       SETUP2  0890
      DO 62 I=1,3                                                       SETUP2  0900
      A13(I,J,2*K-1) =  HHT(I,J,K)                                      SETUP2  0910
   62 A13(I,J,2*K  ) = -HHT(I,J,K)                                      SETUP2  0920
C                                                                       SETUP2  0930
C     SET A23(2K-1) = (R1X)(D1)A13(2K-1)                                SETUP2  0940
C     AND A23(2K)   = (R2X)(D2)A13(2K)                                  SETUP2  0950
C                                                                       SETUP2  0960
      CALL MAT33(D(1,1,M),A13(1,1,2*K-1),TT1)                           SETUP2  0970
      CALL MAT33(D(1,1,N),A13(1,1,2*K  ),TT2)                           SETUP2  0980
      DO 63 J=1,3                                                       SETUP2  0990
      CALL CROSS(RK1(1,K),TT1(1,J),A23(1,J,2*K-1) )                     SETUP2  1000
```

```
   63 CALL CROSS(RK2(1,K),TT2(1,J),A23(1,J,2*K  ) )                    SETUP2  1010
      IF (KQTYPE(K).EQ.4) GO TO 72                                     SETUP2  1020
C                                                                      SETUP2  1030
C     FOR KQTYPE = 1,2 OR 3, SET B31 = A13' AND B32 = A23'             SETUP2  1040
C                                                                      SETUP2  1050
      DO 71 I=1,3                                                      SETUP2  1060
      DO 71 J=1,3                                                      SETUP2  1070
      B31(I,J,2*K-1) = A13(J,I,2*K-1)                                  SETUP2  1080
      B31(I,J,2*K  ) = A13(J,I,2*K  )                                  SETUP2  1090
      B32(I,J,2*K-1) = A23(J,I,2*K-1)                                  SETUP2  1100
   71 B32(I,J,2*K  ) = A23(J,I,2*K  )                                  SETUP2  1110
      GO TO 76                                                         SETUP2  1120
C                                                                      SETUP2  1130
C     FOR KQTYPE = 4, SET B31(2K-1) =  HTT                             SETUP2  1140
C                         B31(2K  ) = -HTT                             SETUP2  1150
C                         B32       = (B31)(D')(RX)'                   SETUP2  1160
C                                                                      SETUP2  1170
   72 CALL DOTT31(HQQ(1,K),TQQ(1,K),B31(1,1,2*K-1))                    SETUP2  1180
      DO 73 I=1,3                                                      SETUP2  1190
      DO 73 J=1,3                                                      SETUP2  1200
   73 B3I(I,J,2*K) = -B31(I,J,2*K-1)                                   SETUP2  1210
      CALL DOTT33(D(1,1,M),B31(1,1,2*K-1),B32(1,1,2*K-1))              SETUP2  1220
      CALL DOTT33(D(1,1,N),B31(1,1,2*K  ),B32(1,1,2*K  ))              SETUP2  1230
      DO 74 J=1,3                                                      SETUP2  1240
      CALL CROSS(RK1(1,K),B32(1,J,2*K-1),TT1(1,J))                     SETUP2  1250
   74 CALL CROSS(RK2(1,K),B32(1,J,2*K  ),TT2(1,J))                     SETUP2  1260
      DO 75 I=1,3                                                      SETUP2  1270
      DO 75 J=1,3                                                      SETUP2  1280
      B32(I,J,2*K-1) = TT1(J,I)                                        SETUP2  1290
   75 B32(I,J,2*K  ) = TT2(J,I)                                        SETUP2  1300
C                                                                      SETUP2  1310
C     COMPUTE V3 = D2'(W2X(W2XR2)) - D1'(W1X(W1XR1))                   SETUP2  1320
C                                                                      SETUP2  1330
   76 CALL CROSS(WMEG(1,M),RK1(1,K),T3)                                SETUP2  1340
      CALL CROSS (WMEG(1,M),T3,T4)                                     SETUP2  1350
      CALL DOT31 (D(1,1,M),T4,T5)                                      SETUP2  1360
      CALL CROSS (WMEG(1,N),RK2(1,K),T6)                               SETUP2  1370
      CALL CROSS (WMEG(1,N),T6,T7)                                     SETUP2  1380
      CALL DOT31 (D(1,1,N),T7,T8)                                      SETUP2  1390
      DO 64 I=1,3                                                      SETUP2  1400
   64 V3(I,K) = T8(I) - T5(I)                                          SETUP2  1410
      IF (KQTYPE(K).NE.2) GO TO 67                                     SETUP2  1420
C                                                                      SETUP2  1430
C      RECOMPUTE V3 FOR KQTYPE=2.                                      SETUP2  1440
C                                                                      SETUP2  1450
      CALL DOT31 (D(1,1,M),T3,T9 )                                     SETUP2  1460
      CALL DOT31 (D(1,1,N),T6,T10)                                     SETUP2  1470
      S2 = 0.0                                                         SETUP2  1480
      DO 65 I=1,3                                                      SETUP2  1490
      T11(I) = SEGLV(I,M)+T9(I) - SEGLV(I,N)-T10(I)                    SETUP2  1500
```

```
   65 S2 = S2 + T11(I)**2                                                    SETUP2  1510
      S3 = HH(1)*V3(1,K) + HH(2)*V3(2,K) + HH(3)*V3(3,K)                     SETUP2  1520
      S4 = S3-S2/SQS1                                                        SETUP2  1530
      DO 66 I=1,3                                                            SETUP2  1540
   66 V3(I,K) = S4*HH(I)                                                     SETUP2  1550
   67 IF (KQTYPE(K).NE.3.AND.KQTYPE(K).NE.6) GO TO 77                        SETUP2  1560
C                                                                            SETUP2  1570
C     FOR KQTYPE=3 OR 6, ADD R DOT TERM FROM PLELP OR SEGSEG TO V3.          SETUP2  1580
C                                                                            SETUP2  1590
      DO 68 I=1,3                                                            SETUP2  1600
   68 V3(I,K) = V3(I,K) + RQQ(I,K)                                           SETUP2  1610
      IF (KQTYPE(K).NE.6)  GO TO 70                                          SETUP2  1620
C                                                                            SETUP2  1630
C     FOR KQTYPE=6, SET V3 = (I-TT')(V3+RQQ)                                 SETUP2  1640
C                                                                            SETUP2  1650
      VQQ = V3(1,K)*TQQ(1,K) + V3(2,K)*TQQ(2,K) + V3(3,K)*TQQ(3,K)           SETUP2  1660
      DO 69 I=1,3                                                            SETUP2  1670
   69 V3(I,K) = V3(I,K) - VQQ*TQQ(I,K)                                       SETUP2  1680
   77 IF (KQTYPE(K).NE.4) GO TO 70                                           SETUP2  1690
C                                                                            SETUP2  1700
C     FOR KQTYPE = 4, ADD R TERM FROM PLELP OR SEGSEG TO V3.                 SETUP2  1710
C                                                                            SETUP2  1720
      S3 = TQQ(1,K)*V3(1,K) + TQQ(2,K)*V3(2,K) + TQQ(3,K)*V3(3,K)            SETUP2  1730
      S4 = S3+SQQ(K)                                                         SETUP2  1740
      DO 78 I=1,3                                                            SETUP2  1750
   78 V3(I,K) = S4*HQQ(I,K)                                                  SETUP2  1760
   70 CONTINUE                                                               SETUP2  1770
C                                                                            SETUP2  1780
C     SPECIAL SETUP FOR TENSION ELEMENTS (KQTYPE = 5).                       SETUP2  1790
C                                                                            SETUP2  1800
      N = 0                                                                  SETUP2  1810
   79 N = N+1                                                                SETUP2  1820
      IF (N.GE.NQ) GO TO 98                                                  SETUP2  1830
      IF (KQTYPE(N).NE.5) GO TO 79                                           SETUP2  1840
      DO 81 I=1,3                                                            SETUP2  1850
      DO 80 J=1,3                                                            SETUP2  1860
      A13(I,J,2*N-1) = 0.0                                                   SETUP2  1870
      A13(I,J,2*N  ) = 0.0                                                   SETUP2  1880
      A23(I,J,2*N  ) = 0.0                                                   SETUP2  1890
      B31(I,J,2*N-1) = 0.0                                                   SETUP2  1900
      B31(I,J,2*N  ) = 0.0                                                   SETUP2  1910
      A13(I,J,2*N+1) = 0.0                                                   SETUP2  1920
      A13(I,J,2*N+2) = 0.0                                                   SETUP2  1930
      A23(I,J,2*N+1) = 0.0                                                   SETUP2  1940
      B31(I,J,2*N+1) = 0.0                                                   SETUP2  1950
      B31(I,J,2*N+2) = 0.0                                                   SETUP2  1960
      HHT(I,J,N    ) = 0.0                                                   SETUP2  1970
   80 HHT(I,J,N+1  ) = 0.0                                                   SETUP2  1980
      A13(I,I,2*N-1) = 1.0                                                   SETUP2  1990
      B31(I,I,2*N-1) = RK1(1,N+1)                                            SETUP2  2000
```

```
      B31(I,I,2*N   ) = RK1(3,N+1)                                          SETUP2  2010
      A13(I,I,2*N+2) = 1.0                                                  SETUP2  2020
      B31(I,I,2*N+1) = RK1(3,N+1)                                           SETUP2  2030
   81 B31(I,I,2*N+2) = RK1(2,N+1)                                           SETUP2  2040
      N1 = KQ1(N)                                                           SETUP2  2050
      N2 = KQ2(N)                                                           SETUP2  2060
      DO 82 K=1,3                                                           SETUP2  2070
      CALL CROSS(RK1(1,N),D(1,K,N1),A23(1,K,2*N-1))                         SETUP2  2080
   82 CALL CROSS(RK2(1,N),D(1,K,N2),A23(1,K,2*N+2))                         SETUP2  2090
      DO 83 I=1,3                                                           SETUP2  2100
      DO 83 J=1,3                                                           SETUP2  2110
      B32(I,J,2*N-1) = RK1(1,N+1)*A23(J,I,2*N-1)                            SETUP2  2120
      B32(I,J,2*N   ) = RK1(3,N+1)*A23(J,I,2*N+2)                           SETUP2  2130
      B32(I,J,2*N+1) = RK1(3,N+1)*A23(J,I,2*N-1)                            SETUP2  2140
   83 B32(I,J,2*N+2) = RK1(2,N+1)*A23(J,I,2*N+2)                            SETUP2  2150
      CALL CROSS(WMEG(1,N1),RK1(1,N),WCRM)                                  SETUP2  2160
      CALL CROSS(WMEG(1,N2),RK2(1,N),WCRN)                                  SETUP2  2170
      CALL DOT31(D(1,1,N1),RK1(1,N),RM)                                     SETUP2  2180
      CALL DOT31(D(1,1,N2),RK2(1,N),RN)                                     SETUP2  2190
      CALL DOT31(D(1,1,N1),WCRM,WCM)                                        SETUP2  2200
      CALL DOT31(D(1,1,N2),WCRN,WCN)                                        SETUP2  2210
      BA = 0.0                                                              SETUP2  2220
      DO 84 I=1,3                                                           SETUP2  2230
      RBA (I) = SEGLP(I,N2) + RN (I) - SEGLP(I,N1) - RM (I)                 SETUP2  2240
      RBAD(I) = SEGLV(I,N2) + WCN(I) - SEGLV(I,N1) - WCM(I)                 SETUP2  2250
   84 BA = BA + RBA(I)**2                                                   SETUP2  2260
      BA = DSQRT(BA)                                                        SETUP2  2270
      FORCE = 0.0                                                           SETUP2  2280
      IF (BA.GT.RK2(3,N+1)) FORCE = RK2(1,N+1)*(1.0-RK2(3,N+1)/BA)          SETUP2  2290
      DO 85 I=1,3                                                           SETUP2  2300
      V3(I,N) = RK2(2,N+1)*RBAD(I) + FORCE*RBA(I)                           SETUP2  2310
   85 V3(I,N+1) = -V3(I,N)                                                  SETUP2  2320
      CALL CROSS(WMEG(1,N1),WCRM,WWCM)                                      SETUP2  2330
      CALL CROSS(WMEG(1,N2),WCRN,WWCN)                                      SETUP2  2340
      CALL DOT31(D(1,1,N1),WWCM,WWM)                                        SETUP2  2350
      CALL DOT31(D(1,1,N2),WWCN,WWN)                                        SETUP2  2360
      DO 86 I=1,3                                                           SETUP2  2370
      V3(I,N   ) = V3(I,N   ) - RK1(1,N+1)*WWM(I) - RK1(3,N+1)*WWN(I)       SETUP2  2380
   86 V3(I,N+1) = V3(I,N+1) - RK1(3,N+1)*WWM(I) - RK1(2,N+1)*WWN(I)         SETUP2  2390
      N = N+1                                                               SETUP2  2400
      GO TO 79                                                              SETUP2  2410
   98 CALL ELTIME(2,26)                                                     SETUP2  2420
      RETURN                                                                SETUP2  2430
      END                                                                   SETUP2  2440
```

```
      SUBROUTINE SINPUT                                                 SINPUT  0010
C                                                          REV 20 05/09/80  SINPUT  0020
C     READS AND PRINTS THE INPUT CARDS THAT DESCRIBE THE PHYSICAL       SINPUT  0030
C     DIMENSIONS OF THE PLANES REPRESENTING THE VEHICLE PANELS AND OF    SINPUT  0040
C     THE RESTRAINT BELTS. ALSO PROCESSES THOSE DATA CARDS THAT DESCRIBE SINPUT  0050
C     ADDITIONAL CONTACT ELLIPSOIDS, CONSTRAINTS, BODY SEGMENT SYMMETRY SINPUT  0060
C     OPTIONS AND SPRING DAMPER FUNCTIONS.                              SINPUT  0070
C                                                                       SINPUT  0080
      IMPLICIT REAL*8 (A-H,O-Z)                                         SINPUT  0090
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           SINPUT  0100
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)        SINPUT  0110
      COMMON/CNTSRF/ PL(17,30),BELT(20,8),TPTS(6,8),BD(24,40)           SINPUT  0120
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30), SINPUT  0130
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       SINPUT  0140
      COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),   SINPUT  0150
     *               HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12), SINPUT  0160
     *               RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),              SINPUT  0170
     *               KQ1(12),KQ2(12),KQTYPE(12)                         SINPUT  0180
      COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),           SINPUT  0190
     *               BLTTTL(5,8),PLTTL(5,30),BAGTTL(5,6),SEG(30),       SINPUT  0200
     *               JOINT(30),CGS(30),JS(30)                           SINPUT  0210
      REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT      SINPUT  0220
      LOGICAL*1 CGS,JS                                                  SINPUT  0230
      COMMON/DAMPER/ APSDM(3,20),APSDN(3,20),ASD(5,20),MSDM(20),MSDN(20) SINPUT  0240
      COMMON/WINDFR/ WTIME(30),QFU(3,5),QFV(3,5),                       SINPUT  0250
     *               IWIND(30),MWSEG(5,30),NFVSEG(6),NFVNT(5)           SINPUT  0260
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         SINPUT  0270
     *               UNITL,UNITM,UNITT,GRAVTY(3)                        SINPUT  0280
      COMMON/TEMPVS/ P1(3),P2(3),P3(3),DE(3,3)                          SINPUT  0290
      DIMENSION IDYPR(3)                                                SINPUT  0300
      DATA IDYPR/3,2,1/                                                 SINPUT  0310
C                                                                       SINPUT  0320
C     INPUT CARD D.1                                                    SINPUT  0330
C                                                                       SINPUT  0340
      READ  (5,11)  NPL,NBLT,NBAG,NELP,NQ,NSD,NHRNSS,NWINDF,NJNTF,NFORCE SINPUT  0350
   11 FORMAT(12I6)                                                      SINPUT  0360
      WRITE (6,16)  NPL,NBLT,NBAG,NELP,NQ,NSD,NHRNSS,NWINDF,NJNTF,NFORCE SINPUT  0370
   16 FORMAT('1    NPL    NBLT    NBAG    NELP      NQ     NSD',       SINPUT  0380
     *       '  NHRNSS  NWINDF   NJNTF  NFORCE', 40X, 'CARD  D.1'/10I8) SINPUT  0390
      IF (NPL.EQ.0) GO TO 15                                            SINPUT  0400
      IPAGE = 0                                                         SINPUT  0410
      DO 20 J=1,NPL                                                     SINPUT  0420
C                                                                       SINPUT  0430
C     READ AND PRINT CARDS D.2.A,D.2.B AND D.2.C FOR THE JTH PLANE.     SINPUT  0440
C                                                                       SINPUT  0450
      READ (5,23) JJ,(PLTTL(I,J),I = 1,5),P1,P2,P3                      SINPUT  0460
   23 FORMAT (I4,4X,5A4/(3F12.0))                                       SINPUT  0470
      IF (JJ.NE.J) WRITE (6,24) JJ,J                                    SINPUT  0480
   24 FORMAT (' PLANE INDEX INPUT ERROR,',2I4)                          SINPUT  0490
      IF  (JJ.NE.J)  STOP 10                                            SINPUT  0500
```

```
      IF (MOD(J,7).EQ.1) WRITE (6,12) IPAGE                              SINPUT  0510
   12 FORMAT(I1,' PLANE INPUTS',106X,'CARDS D.2')                        SINPUT  0520
      IPAGE = 1                                                          SINPUT  0530
      WRITE (6,25) J, (PLTTL(I,J),I = 1,5),P1,P2,P3                      SINPUT  0540
   25 FORMAT('0 PLANE NO.',I4,4X,5A4//17X,'X',11X,'Y',11X,'Z'/           SINPUT  0550
     *       '  POINT 1 ' ,3F12.4/                                       SINPUT  0560
     *       '  POINT 2 ' ,3F12.4/                                       SINPUT  0570
     *       '  POINT 3 ' ,3F12.4)                                       SINPUT  0580
C                                                                        SINPUT  0590
C     PROGRAM NOW ASSUMES THE FINITE PLANE IS A PARALLELOGRAM IN SHAPE   SINPUT  0600
C     WHERE THE INPUT POINTS P1,P2,P3 ARE 3 OF THE CORNERS SUCH THAT     SINPUT  0610
C     EDGE P1-P2 IS LESS THAN 180 DEGREES CLOCKWISE (AS VIEWED BY THE    SINPUT  0620
C     OCCUPANT) FROM THE EDGE P1-P3.                                     SINPUT  0630
C                                                                        SINPUT  0640
C     SET UP PL ARRAY AS REQUIRED BY SUBROUTINE PLELP                    SINPUT  0650
C                                                                        SINPUT  0660
C         PL(1,J) = A0    NORMAL EQUATION OF JTH PLACE                 SINPUT  0670
C         PL(2,J) = B0      A0*X + B0*Y + C0*Z = D0                    SINPUT  0680
C         PL(3,J) = C0                                                 SINPUT  0690
C         PL(4,J) = D0                                                 SINPUT  0700
C                                                                      SINPUT  0710
C         PL(5,J)                                                      SINPUT  0720
C         PL(6,J)         NOT CURRENTLY USED BY PROGRAM                SINPUT  0730
C         PL(7,J)                                                      SINPUT  0740
C                                                                      SINPUT  0750
C         PL(8,J) =A1                                                  SINPUT  0760
C         PL(9,J) =B1     NORMAL EQUATION OF 1ST BOUNDARY PLANE        SINPUT  0770
C         PL(10,J)=C1       A1*X + B1*Y + C1*Z = D1                    SINPUT  0780
C         PL(11,J)=D1     AND E1 IS LENGTH OF PLANE FROM BOUNDARY.     SINPUT  0790
C         PL(12,J)=E1                                                  SINPUT  0800
C                                                                      SINPUT  0810
C         PL(13,J)=A2                                                  SINPUT  0820
C         PL(14,J)=B2     NORMAL EQUATION OF 2ND BOUNDARY PLANE        SINPUT  0830
C         PL(15,J)=C2       A2*X + B2*Y + C2*Z = D2                    SINPUT  0840
C         PL(16,J)=D2     AND E2 IS LENGTH OF PLANE FROM BOUNDARY.     SINPUT  0850
C         PL(17,J)=E2                                                  SINPUT  0860
C                                                                      SINPUT  0870
      S22 = 0.0                                                        SINPUT  0880
      S23 = 0.0                                                        SINPUT  0890
      S33 = 0.0                                                        SINPUT  0900
      DO 26 I =1,3                                                     SINPUT  0910
      P2(I) = P2(I)-P1(I)                                              SINPUT  0920
      P3(I) = P3(I)-P1(I)                                              SINPUT  0930
      S22 = S22 + P2(I)*P2(I)                                          SINPUT  0940
      S23 = S23 + P2(I)*P3(I)                                          SINPUT  0950
   26 S33 = S33 + P3(I)*P3(I)                                          SINPUT  0960
      S2  = DSQRT(S22)                                                 SINPUT  0970
      S3  = DSQRT(S33)                                                 SINPUT  0980
      CALL CROSS(P2,P3,PL(1,J))                                        SINPUT  0990
      S1 = 0.0                                                         SINPUT  1000
```

```
      DO 27 I=1,3                                                                SINPUT  1010
   27 S1 = S1 + PL(I,J)**2                                                       SINPUT  1020
      S1 = DSQRT(S1)                                                            SINPUT  1030
      DO 28 I=1,3                                                               SINPUT  1040
      PL(I,J) = PL(I,J)/S1                                                      SINPUT  1050
      PL(I+7 ,J) = (S33*P2(I) - S23*P3(I)) / (S1*S3)                            SINPUT  1060
   28 PL(I+12,J) = (S22*P3(I) - S23*P2(I)) / (S1*S2)                            SINPUT  1070
      PL( 4,J) = P1(1)*PL( 1,J) + P1(2)*PL( 2,J) + P1(3)*PL( 3,J)               SINPUT  1080
      PL(11,J) = P1(1)*PL( 8,J) + P1(2)*PL( 9,J) + P1(3)*PL(10,J)               SINPUT  1090
      PL(12,J) = P2(1)*PL( 8,J) + P2(2)*PL( 9,J) + P2(3)*PL(10,J)               SINPUT  1100
      PL(16,J) = P1(1)*PL(13,J) + P1(2)*PL(14,J) + P1(3)*PL(15,J)               SINPUT  1110
   20 PL(17,J) = P3(1)*PL(13,J) + P3(2)*PL(14,J) + P3(3)*PL(15,J)               SINPUT  1120
   15 IF (NBLT.EQ.0) GO TO 35                                                   SINPUT  1130
      DO 30 J=1,NBLT                                                            SINPUT  1140
C                                                                               SINPUT  1150
C     READ AND PRINT CARDS D.3.A, D.3.B AND D.3.C FOR THE JTH BELT.             SINPUT  1160
C                                                                               SINPUT  1170
      READ (5,13) (BLTTTL(I,J),I = 1,5),(BELT(I,J),I = 1,11)                    SINPUT  1180
   13 FORMAT (5A4/(6F12.0))                                                     SINPUT  1190
      IF (MOD(J,5).EQ.1) WRITE (6,21)                                           SINPUT  1200
   21 FORMAT('1 BELT INPUTS',107X,'CARDS D.3')                                  SINPUT  1210
   30 WRITE (6,14) J,(BLTTTL(I,J),I = 1,5),(BELT(I,J),I = 1,11)                 SINPUT  1220
   14 FORMAT('0 BELT NO.',I4,4X,5A4//                                           SINPUT  1230
     *    30X,'ANCHOR POINT A',46X,'ANCHOR POINT B'/                           SINPUT  1240
     *    2(16X,'X',19X,'Y',19X,'Z',3X)/6F20.3//                               SINPUT  1250
     *    26X,'FIXED POINT ON SEGMENT',45X,'SLACK(+)'/                         SINPUT  1260
     *    16X,'X',19X,'Y',19X,'Z',17X,'BLANK',13X,'LENGTH(-)'/5F20.3)          SINPUT  1270
C                                                                               SINPUT  1280
C     CALL AIRBG1 ROUTINE IF REQUIRED FOR AIRBAG INPUT                          SINPUT  1290
C                                                                               SINPUT  1300
   35 IF (NBAG.NE.0) CALL AIRBG1                                                SINPUT  1310
      IF (NELP.LE.0) GO TO 51                                                   SINPUT  1320
C                                                                               SINPUT  1330
C     READ AND PRINT CARDS D.5 FOR ELLIPSOID INPUT, IF ANY.                     SINPUT  1340
C       NOTE: NELP IS THE NO. OF ELLIPSOIDS TO BE SUPPLIED HERE, NOT THE        SINPUT  1350
C             NO. OF ELLIPSOIDS IN THE PROGRAM, SINCE THE FIRST NSEG            SINPUT  1360
C             ELLIPSOIDS WERE SUPPLIED ON CARDS B.2.A - B.2.I. HOWEVER          SINPUT  1370
C             THEY MAY BE REPLACED HERE IF DESIRED.                             SINPUT  1380
C                                                                               SINPUT  1390
      WRITE (6,41) UNITL,UNITL                                                  SINPUT  1400
   41 FORMAT('1 ADDITIONAL ELLIPSOID INPUT',92X,'CARDS D.5'//                   SINPUT  1410
     *  17X,'SEMIAXES (',A4,')',18X,'OFFSET (',A4,')',                         SINPUT  1420
     *  20X,'ROTATION (DEG)'/                                                  SINPUT  1430
     *  3X,'NO.',2(8X,'X',8X,'Y',8X,'Z',6X),7X,'YAW',7X,'PITCH',5X,            SINPUT  1440
     *  'ROLL'//)                                                              SINPUT  1450
      DO 50 MM=1,NELP                                                           SINPUT  1460
      READ (5,42) M,P1,P2,P3                                                    SINPUT  1470
   42 FORMAT(I6,9F6.0)                                                          SINPUT  1480
      WRITE (6,43) M,P1,P2,P3                                                   SINPUT  1490
   43 FORMAT(I6,3(3X,3F9.3,3X))                                                 SINPUT  1500
```

```
      CALL DRCYPR (DE,P3,IDYPR)                                          SINPUT  1510
      DO 45 I=1,3                                                        SINPUT  1520
      BD(I  ,M) = P1(I)                                                  SINPUT  1530
      BD(I+3,M) = P2(I)                                                  SINPUT  1540
      DO 45 J=1,3                                                        SINPUT  1550
      SUM1 = 0.0                                                         SINPUT  1560
      SUM2 = 0.0                                                         SINPUT  1570
      DO 44 L=1,3                                                        SINPUT  1580
      SUM1 = SUM1 + DE(L,I)/P1(L)**2*DE(L,J)                             SINPUT  1590
   44 SUM2 = SUM2 + DE(L,I)*P1(L)**2*DE(L,J)                             SINPUT  1600
      K = 3*I +J +3                                                      SINPUT  1610
      BD(K  ,M) = SUM1                                                   SINPUT  1620
   45 BD(K+9,M) = SUM2                                                   SINPUT  1630
   50 CONTINUE                                                           SINPUT  1640
C                                                                        SINPUT  1650
C     READ AND PRINT CARDS D.6 FOR CONSTRAINT INPUT, IF ANY.             SINPUT  1660
C                                                                        SINPUT  1670
   51 IF (NQ.LE.0) GO TO 70                                              SINPUT  1680
      DO 60 K=1,NQ                                                       SINPUT  1690
      READ (5,52) KQTYPE(K),KQ1(K),KQ2(K),(RK1(I,K),I=1,3)               SINPUT  1700
     *                                    ,(RK2(I,K),I=1,3)             SINPUT  1710
   52 FORMAT(3I6,6F6.0)                                                  SINPUT  1720
      IF (K.EQ.1) WRITE (6,53) UNITL,UNITL                               SINPUT  1730
   53 FORMAT('1 CONSTRAINT INPUT',102X,'CARDS D.6'//                     SINPUT  1740
     *       '   TYPE   SEGMENT  SEGMENT       POINT ON 1ST SEGMENT (',  SINPUT  1750
     *      A4,')','      POINT ON 2ND SEGMENT (',A4,')'/                SINPUT  1760
     *       '    NO.    NO. 1    NO. 2          X         Y         Z   SINPUT  1770
     *         X         Y         Z'//)                                 SINPUT  1780
      WRITE (6,54)  KQTYPE(K),KQ1(K),KQ2(K),(RK1(I,K),I=1,3)             SINPUT  1790
     *                                      ,(RK2(I,K),I=1,3)            SINPUT  1800
   54 FORMAT(I6,2I9,2(6X,3F9.3) )                                        SINPUT  1810
   60 CONTINUE                                                           SINPUT  1820
C                                                                        SINPUT  1830
C     CARD D.7  BODY SEGMENT SYMMETRY INPUT                              SINPUT  1840
C                                                                        SINPUT  1850
   70 READ (5,71) (NSYM(J),J=1,NSEG)                                     SINPUT  1860
   71 FORMAT(18I4)                                                       SINPUT  1870
      WRITE(6,72) (J,J=1,NSEG)                                           SINPUT  1880
      WRITE(6,73) (NSYM(J),J=1,NSEG)                                     SINPUT  1890
   72 FORMAT('0 BODY SEGMENT SYMMETRY INPUT',91X,'CARD D.7'//            SINPUT  1900
     *       '  SEG NO.',30I4)                                          SINPUT  1910
   73 FORMAT('0 NSYM(J)',30I4)                                           SINPUT  1920
      NSEG1 = NSEG+1                                                     SINPUT  1930
      DO 74 J=NSEG1,NGRND                                                SINPUT  1940
   74 NSYM(J) = 0                                                        SINPUT  1950
      IF  (NSD.LE.0)  GO TO 90                                           SINPUT  1960
C                                                                        SINPUT  1970
C     CARD D.8   SPRING DAMPERS FUNCTION INPUT.                          SINPUT  1980
C                                                                        SINPUT  1990
      DO 79 J=1,NSD                                                      SINPUT  2000
```

```
   79 READ (5,80) MSDM(J),MSDN(J),(APSDM(I,J),I=1,3),                    SINPUT  2010
     *            (APSDN(I,J),I=1,3),(ASD(I,J),I=1,5)                   SINPUT  2020
   80 FORMAT(2I3,11F6.0)                                                 SINPUT  2030
      WRITE (6,81) UNITL                                                 SINPUT  2040
   81 FORMAT('0',5X,'SPRING  DAMPERS  FUNCTION  INPUT',82X,'CARDS D.8'//  SINPUT  2050
     *   18X,'COORDINATES OF ATTACHMENT POINTS (',A4,')'/                 SINPUT  2060
     *   5X,'SEGMENT',9X,'SEGMENT M',16X,'SEGMENT N',15X,                 SINPUT  2070
     *   'SPRING FORCE FUNCTION'12X,'DAMPING FORCE FUNCTION'/             SINPUT  2080
     *   ' NO.  M   N',2(6X,'X',7X,'Y',7X,'Z',2X),7X,'D0',9X,'A1',11X,    SINPUT  2090
     *   'A2',13X,'B1',10X,'B2' // )                                      SINPUT  2100
      DO  82  J=1,NSD                                                    SINPUT  2110
   82 WRITE (6,83) J,MSDM(J),MSDN(J),(APSDM(I,J),I=1,3),                 SINPUT  2120
     *            (APSDN(I,J),I=1,3),(ASD(I,J),I=1,5)                   SINPUT  2130
   83 FORMAT(I3,2I4,2(1X,3F8.2),F11.2,2F12.3,F15.3,F12.3)                SINPUT  2140
C                                                                        SINPUT  2150
C     CARDS D.9  FORCE FUNCTIONS.                                        SINPUT  2160
C                                                                        SINPUT  2170
   90 NFVSEG(6) = NFORCE                                                 SINPUT  2180
      IF  (NFORCE.LE.0)  GO TO 99                                        SINPUT  2190
      WRITE  (6,91)                                                      SINPUT  2200
   91 FORMAT ('0',6X,'FORCE FUNCTIONS INPUT',92X,'CARDS D.9'//           SINPUT  2210
     *        5X,'NO.', 5X,'SEG', 5X,'FCN', 13X,'X', 9X,'Y', 9X,'Z',    SINPUT  2220
     *        13X,'YAW', 6X,'PITCH', 6X,'ROLL' //)                       SINPUT  2230
      DO  95  J=1,NFORCE                                                 SINPUT  2240
      READ  (5,92)  NFVSEG(J),NFVNT(J),P1,P2                             SINPUT  2250
   92 FORMAT (2I6,6F10.0)                                                SINPUT  2260
      WRITE  (6,93)  J,NFVSEG(J),NFVNT(J),P1,P2                          SINPUT  2270
   93 FORMAT (3I8,6X,3F10.3,6X,3F10.3)                                   SINPUT  2280
      CALL DRCYPR (DE,P2,IDYPR)                                          SINPUT  2290
      DO  94  I=1,3                                                      SINPUT  2300
   94 QFU(I,J) = DE(I,1)                                                 SINPUT  2310
   95 CALL CROSS (P1,QFU(1,J),QFV(1,J))                                  SINPUT  2320
   99 RETURN                                                             SINPUT  2330
      END                                                                SINPUT  2340
```

```
      SUBROUTINE  SLPLOT (X, NX, XO, XN, XL, XSIZE, XLAB, NXLB,          SLPLOT  0010
     *                    Y, NY, YO, YN, YL, YSIZE, YLAB, NYLB,          SLPLOT  0020
     *                    NPTS, NYY, NDY, PLAB1, NPLB1, PLAB2, NPLB2)    SLPLOT  0030
C                                                    REV 20 11/30/79    SLPLOT  0040
C                                                                       SLPLOT  0050
C     ARGUMENTS:                                                        SLPLOT  0060
C        X(NPTS)      - ARRAY OF NPTS ABSCISSAS TO BE PLOTTED.          SLPLOT  0070
C        Y(NDY,NYY)   - ARRAY OF NPTS*NYY ORDINATES TO BE PLOTTED.      SLPLOT  0080
C        NX,NY        - POSITIVE - NO. OF LINEAR SUBDIVISIONS.          SLPLOT  0090
C                       NEGATIVE - NO. OF LOGARITHMIC DECADES.          SLPLOT  0100
C        XO,YO        - AXES ORIGINS (POWER OF TEN IF NX,NY NEGATIVE).  SLPLOT  0110
C        XN,YN        - AXES END VALUES (REQUIRED IF NX,NY POSITIVE).   SLPLOT  0120
C        XL,YL        - LENGTH (INCHES) OF X,Y AXES.                    SLPLOT  0130
C        XSIZE,YSIZE  - PAPER SIZE (INCHES) IN X,Y DIRECTIONS.          SLPLOT  0140
C        XLAB,YLAB    - X,Y AXES LABELS (ALPHANUMERIC ARRAYS).          SLPLOT  0150
C        NXLB,NYLB    - NO. OF CHARACTERS IN X,Y LABELS.                SLPLOT  0160
C        NPTS         - NO. OF POINTS IN X ARRAY AND EACH Y ARRAY.      SLPLOT  0170
C        NYY          - NO. OF Y ARRAYS TO BE PLOTTED VS. X ARRAY.      SLPLOT  0180
C        NDY          - FIRST DIMENSION OF Y ARRAY IN CALLING ROUTINE.  SLPLOT  0190
C                       (NDY MUST BE .GE. NPTS)                         SLPLOT  0200
C        PLAB1,PLAB2  - 1ST & 2ND LINES OF PLOT ID LABELS (ALPHANUMERIC).SLPLOT  0210
C        NPLB1,NPLB2  - NO. OF CHARACTERS IN PLOT ID LABELS.            SLPLOT  0220
C                                                                       SLPLOT  0230
C     NOTE: PLOTS WILL BE TRUNCATED AS FOLLOWS:                         SLPLOT  0240
C      NX,NY POSITIVE - XO,YO .LE. X,Y .LE. XN,YN                       SLPLOT  0250
C      NX,NY NEGATIVE - XO,YO .LE. X,Y .LE. XN*10**(-NX),YO*10**(-NY)   SLPLOT  0260
C                                                                       SLPLOT  0270
      DIMENSION X(NPTS),Y(NDY,NYY),XLAB(1),YLAB(1),PLAB1(1),PLAB2(1)    SLPLOT  0280
C                                                                       SLPLOT  0290
C     NOTE: THIS ROUTINE HAS BEEN WRITTEN FOR THE PLOTTING FACILITIES   SLPLOT  0300
C     AT CALSPAN. THE FOLLOWING ITEMS ARE KNOWN TO BE CONTRARY TO THE   SLPLOT  0310
C     NORMAL CALCOMP PROCEDURES AND SHOULD BE EXAMINED BY USERS AT OTHER SLPLOT  0320
C     COMPUTER SYSTEMS AND CHANGES MADE ACCORDINGLY.                    SLPLOT  0330
C                                                                       SLPLOT  0340
C       1. AT CALSPAN THE PLOTTED CHARACTERS GENERATED BY SUBROUTINE    SLPLOT  0350
C          SYMBOL HAVE A WIDTH OF 6/7 TIMES THE HEIGHT. FOR THE CALCOMP SLPLOT  0360
C          ROUTINES THE WIDTH IS EQUAL TO THE HEIGHT. THE STATEMENT     SLPLOT  0370
C          'WIDTHF = 6.0/7.0' SHOULD BE CHANGED TO 'WIDTHF = 1.0'.      SLPLOT  0380
C                                                                       SLPLOT  0390
C       2. THE ONLY INITIALIZATION REQUIRED AT CALSPAN IS THE STATEMENT SLPLOT  0400
C          'CALL PLOT (0.0,0.0,0)' TO ESTABLISH A NEW PAGE, INCLUDING   SLPLOT  0410
C          THE FIRST PAGE. THIS IS FOLLOWED BY 'CALL PLOT (XO,YO,-3)' TO SLPLOT  0420
C          SET THE PLOT ORIGIN ON THE PAGE. PROPER PLOT INITIALIZATION  SLPLOT  0430
C          SHOULD BE DONE HERE AND IN SUBROUTINE POSTPR (AFTER STATEMENT SLPLOT  0440
C          NO. 30) AS REQUIRED BY THE USER'S PLOTTING FACILITY.         SLPLOT  0450
C                                                                       SLPLOT  0460
C       3. THE STATEMENT 'CALL NEWPEN(2)' SHOULD BE EXAMINED OR DELETED.SLPLOT  0470
C                                                                       SLPLOT  0480
C       4. THE STATEMENT 'CALL EFPLOT' AFTER STATEMENT NO. 50 IN POSTPR SLPLOT  0490
C          IS REQUIRED AT CALSPAN TO CLOSE OUT THE PLOT FILES. THIS     SLPLOT  0500
```

```
C           SHOULD BE CHANGED TO CONFORM TO THE REQUIREMENTS OF THE          SLPLOT  0510
C           USER'S PLOTTING FACILITIES.                                      SLPLOT  0520
C                                                                            SLPLOT  0530
C       5. THE NECESSARY JOB CONTROL LANGUAGE FOR PLOTTING IS NECESSARY.     SLPLOT  0540
C                                                                            SLPLOT  0550
C       6. THE ONLY CALCOMP ROUTINES NEEDED ARE SYMBOL, NUMBER AND PLOT.     SLPLOT  0560
C                                                                            SLPLOT  0570
      LOGICAL NXPOS,NXNEG,NYPOS,NYNEG                                        SLPLOT  0580
      DATA HN/0.07/, HL/0.105/                                               SLPLOT  0590
      WIDTHF = 6.0/7.0                                                       SLPLOT  0600
      WN = WIDTHF*HN                                                         SLPLOT  0610
      WL = WIDTHF*HL                                                         SLPLOT  0620
C         **  PLOT PAGE INITIALIZATION  **                                   SLPLOT  0630
CAL   CALL PLOT (0.00,0.00, 0)                                               SLPLOT  0640
      CALL PLOT (0.00,0.00, 999)                                             SLPLOT  0650
      XP = 0.5*(XSIZE-(XL-0.5))                                              SLPLOT  0660
      YP = 0.5*(YSIZE-(YL-1.0))                                              SLPLOT  0670
      CALL PLOT (XP,YP,-3)                                                   SLPLOT  0680
      CALL NEWPEN(2)                                                         SLPLOT  0690
      NXPOS = NX.GT.0                                                        SLPLOT  0700
      NXNEG = NX.LT.0                                                        SLPLOT  0710
      NYPOS = NY.GT.0                                                        SLPLOT  0720
      NYNEG = NY.LT.0                                                        SLPLOT  0730
C         **  PLOT AXES AND ID LABELS.  **                                   SLPLOT  0740
      XP = 0.0                                                               SLPLOT  0750
      YP = 0.0                                                               SLPLOT  0760
      IF  (.NOT.NXPOS)  GO TO 12                                             SLPLOT  0770
C         **        LINEAR X AXIS       **                                 SLPLOT  0780
      CALL LINAXS (XP, YP, 0.0, NX, XL)                                      SLPLOT  0790
      XB = XL/(XN-X0)                                                        SLPLOT  0800
C         **    LINEAR X AXIS NUMERICS  **                                   SLPLOT  0810
      DX = XL/FLOAT(NX)                                                      SLPLOT  0820
      EX = X0                                                                SLPLOT  0830
      DD = (XN-X0)/FLOAT(NX)                                                 SLPLOT  0840
      ND = 0.99 - ALOG10(ABS(DD))                                            SLPLOT  0850
      IF  (ND.LE.0)  ND = -1                                                 SLPLOT  0860
      IX = 0                                                                 SLPLOT  0870
      YC = YP - 2.0*HN                                                       SLPLOT  0880
   11 AX = ABS(EX)                                                           SLPLOT  0890
      NF = 0                                                                 SLPLOT  0900
      IF  (AX.GE.10.0)  NF = ALOG10(AX)                                      SLPLOT  0910
      NS = 0                                                                 SLPLOT  0920
      IF  (EX.LT.0.0)   NS = 1                                               SLPLOT  0930
      SP = NS+NF+2+ND                                                        SLPLOT  0940
      XC = XP - 0.5*SP*WN                                                    SLPLOT  0950
      CALL NUMBER (XC, YC, HN, EX, 0.0, ND)                                  SLPLOT  0960
      XP = XP + DX                                                           SLPLOT  0970
      EX = EX + DD                                                           SLPLOT  0980
      IX = IX + 1                                                            SLPLOT  0990
      IF  (ABS(EX).GT.ABS(0.1*DD))  GO TO 18                                 SLPLOT  1000
```

```
      IF  (IX.GT.NX)  GO TO 12                                                  SLPLOT  1010
      CALL PLOT (XP, YP+YL,3)                                                   SLPLOT  1020
      CALL PLOT (XP, YP   ,2)                                                   SLPLOT  1030
   18 IF  (IX.LE.NX)    GO TO 11                                                SLPLOT  1040
   12 IF  (.NOT.NXNEG)  GO TO 14                                                SLPLOT  1050
C         **          LOG X AXIS          **                                    SLPLOT  1060
      CALL LOGAXS (XP, YP, 0.0, -NX, XL)                                        SLPLOT  1070
      XB = XL/ALOG(10.0**(-NX))                                                 SLPLOT  1080
      XA = -XB*ALOG(X0)                                                         SLPLOT  1090
C         **     LOG X AXIS NUMERICS    **                                    SLPLOT  1100
      DX = XL/FLOAT(-NX)                                                        SLPLOT  1110
      EX = ALOG10(X0)                                                           SLPLOT  1120
      IX = 0                                                                    SLPLOT  1130
   13 CALL NUMBER (XP-1.0*WN, YP-2.5*HN, HN, 10.0, 0.0, -1)                     SLPLOT  1140
      CALL NUMBER (XP+1.0*WN, YP-2.0*HN, HN,   EX, 0.0, -1)                     SLPLOT  1150
      XP = XP + DX                                                              SLPLOT  1160
      EX = EX + 1.0                                                             SLPLOT  1170
      IX = IX - 1                                                               SLPLOT  1180
      IF  (IX.GE.NX)   GO TO 13                                                 SLPLOT  1190
   14 IF  (NXLB.LE.0)  GO TO 15                                                 SLPLOT  1200
C         **        X AXIS LABEL          **                                    SLPLOT  1210
      XPX = (XL-FLOAT(NXLB)*WL)/2.0                                             SLPLOT  1220
      YPX = YP-4.0*HN-HL                                                        SLPLOT  1230
      CALL SYMBOL(XPX, YPX, HL, XLAB,  0.0, NXLB)                               SLPLOT  1240
   15 IF  (NPLB1.LE.0)  GO TO 16                                                SLPLOT  1250
C         **   PLOT LABEL - 1ST LINE    **                                    SLPLOT  1260
      XP1 = (XL-FLOAT(NPLB1)*WL)/2.0                                            SLPLOT  1270
      YP1 = YP-4.0*HN-4.0*HL                                                    SLPLOT  1280
      CALL SYMBOL (XP1, YP1, HL, PLAB1,  0.0, NPLB1)                            SLPLOT  1290
   16 IF  (NPLB2.LE.0)  GO TO 20                                                SLPLOT  1300
C         **   PLOT LABEL - 2ND LINE    **                                    SLPLOT  1310
      XP2 = (XL-FLOAT(NPLB2)*WL)/2.0                                            SLPLOT  1320
      YP2 = YP-4.0*HN-6.0*HL                                                    SLPLOT  1330
      CALL SYMBOL (XP2, YP2, HL, PLAB2,  0.0, NPLB2)                            SLPLOT  1340
   20 XP = 0.0                                                                  SLPLOT  1350
C         **     COMPLETE AXIS GRID     **                                    SLPLOT  1360
      IF  (NYPOS)  CALL LINAXS (XL, YP,  90.0,  NY,  YL)                        SLPLOT  1370
      IF  (NYNEG)  CALL LOGAXS (XL, YP,  90.0, -NY,  YL)                        SLPLOT  1380
      IF  (NXPOS)  CALL LINAXS (XL ,YL, 180.0,  NX,  XL)                        SLPLOT  1390
      IF  (NXNEG)  CALL LOGAXS (XL ,YL, 180.0, -NX, -XL)                        SLPLOT  1400
      IF  (.NOT.NYPOS)  GO TO 22                                                SLPLOT  1410
C         **        LINEAR Y AXIS         **                                    SLPLOT  1420
      CALL LINAXS (XP, YL, -90.0,  NY,  YL)                                     SLPLOT  1430
      YB = YL/(YN-Y0)                                                           SLPLOT  1440
C         **    LINEAR Y AXIS NUMERICS  **                                    SLPLOT  1450
      DY = YL/FLOAT(NY)                                                         SLPLOT  1460
      EY = Y0                                                                   SLPLOT  1470
      DD = (YN-Y0)/FLOAT(NY)                                                    SLPLOT  1480
      ND = 0.99 - ALOG10(ABS(DD))                                               SLPLOT  1490
      IF  (ND.LE.0)  ND = -1                                                    SLPLOT  1500
```

```
      IY = 0                                                              SLPLOT  1510
      XC = XP - 1.0*HN                                                    SLPLOT  1520
   21 AY = ABS(EY)                                                        SLPLOT  1530
      NF = 0                                                              SLPLOT  1540
      IF  (AY.GE.10.0)  NF = ALOG10(AY)                                   SLPLOT  1550
      NS = 0                                                              SLPLOT  1560
      IF  (EY.LT.0.0)   NS = 1                                            SLPLOT  1570
      SP = NS+NF+2+ND                                                     SLPLOT  1580
      YC = YP - 0.5*SP*WN                                                 SLPLOT  1590
      CALL NUMBER (XC, YC, HN, EY, 90.0, ND)                              SLPLOT  1600
      YP = YP + DY                                                        SLPLOT  1610
      EY = EY + DD                                                        SLPLOT  1620
      IY = IY + 1                                                         SLPLOT  1630
      IF  (ABS(EY).GT.ABS(0.1*DD))  GO TO 19                              SLPLOT  1640
      IF  (IY.GT.NY)  GO TO 22                                            SLPLOT  1650
      CALL PLOT (XP+XL, YP, 3)                                            SLPLOT  1660
      CALL PLOT (XP   , YP, 2)                                            SLPLOT  1670
   19 IF  (IY.LE.NY)    GO TO 21                                         SLPLOT  1680
   22 IF  (.NOT.NYNEG)  GO TO 24                                         SLPLOT  1690
C        **         LOG Y AXIS          **                               SLPLOT  1700
      CALL LOGAXS (XP, YL, -90.0, -NY, -YL)                               SLPLOT  1710
      YB = YL/ALOG(10.0**(-NY))                                           SLPLOT  1720
      YA = -YB*ALOG(Y0)                                                   SLPLOT  1730
C        **     LOG Y AXIS NUMERICS    **                                SLPLOT  1740
      DY = YL/FLOAT(-NY)                                                  SLPLOT  1750
      EY = ALOG10(Y0)                                                     SLPLOT  1760
      IY = 0                                                              SLPLOT  1770
   23 CALL NUMBER (XP-1.0*HN, YP-1.0*WN, HN, 10.0, 90.0, -1)              SLPLOT  1780
      CALL NUMBER (XP-1.5*HN, YP+1.0*WN, HN,   EY, 90.0, -1)              SLPLOT  1790
      YP = YP + DY                                                        SLPLOT  1800
      EY = EY + 1.0                                                       SLPLOT  1810
      IY = IY - 1                                                         SLPLOT  1820
      IF  (IY.GE.NY)   GO TO 23                                           SLPLOT  1830
   24 IF  (NYLB.LE.0)  GO TO 25                                           SLPLOT  1840
C        **       Y AXIS LABEL        **                                 SLPLOT  1850
      XPY = XP-4.0*HN                                                     SLPLOT  1860
      YPY = (YL-FLOAT(NYLB)*WL)/2.0                                       SLPLOT  1870
      CALL SYMBOL(XPY, YPY, HL, YLAB, 90.0, NYLB)                         SLPLOT  1880
   25 CONTINUE                                                            SLPLOT  1890
C        **      PLOT DATA ARRAYS      **                                SLPLOT  1900
      NSYM = 24                                                           SLPLOT  1910
      IS = NPTS/NSYM                                                      SLPLOT  1920
      XOMIN = X0/1000.0                                                   SLPLOT  1930
      YOMIN = Y0/1000.0                                                   SLPLOT  1940
      DO  40  J=1,NYY                                                     SLPLOT  1950
      IPEN = 3                                                            SLPLOT  1960
      DO  39  I=1,NPTS                                                    SLPLOT  1970
      X1 = X2                                                             SLPLOT  1980
      Y1 = Y2                                                             SLPLOT  1990
      IF  (NXPOS)  X2 = XB*(X(I)  -X0)                                    SLPLOT  2000
```

```
      IF  (NYPOS)  Y2 = YB*(Y(I,J)-Y0)                                          SLPLOT  2010
      IF  (NXNEG)  X2 = XA + XB*ALOG(AMAX1(X(I)  ,X0MIN))                       SLPLOT  2020
      IF  (NYNEG)  Y2 = YA + YB*ALOG(AMAX1(Y(I,J),Y0MIN))                       SLPLOT  2030
      IF  (Y2.LT.0.0 .OR. Y2.GT.YL)  GO TO 33                                   SLPLOT  2040
      IF  (X2.LT.0.0 .OR. X2.GT.XL)  GO TO 33                                   SLPLOT  2050
      IF  (IPEN.EQ.3)  GO TO 33                                                 SLPLOT  2060
      CALL PLOT (X2,Y2,IPEN)                                                    SLPLOT  2070
C          **     PLOT NYSM SYMBOLS      **                                     SLPLOT  2080
      IF  (NYY.EQ.1 .OR. MOD(I,IS).NE.0)  GO TO 39                              SLPLOT  2090
      IF  (MOD((I/IS)-1,NYY)+1.EQ.J)  CALL SYMBOL (X2,Y2,0.14,J,0.0,-2)         SLPLOT  2100
      GO TO 39                                                                  SLPLOT  2110
   33 IF  (I.EQ.1)     GO TO 39                                                 SLPLOT  2120
      DX  = X2 - X1                                                             SLPLOT  2130
      IF  (DX.NE.0.0)  GO TO 34                                                 SLPLOT  2140
      AX0 = 1.0                                                                 SLPLOT  2150
      AXL = 0.0                                                                 SLPLOT  2160
      IF  (X1.GE.0.0)  AX0 = 0.0                                                SLPLOT  2170
      IF  (X1.LE.XL )  AXL = 1.0                                                SLPLOT  2180
      GO TO 35                                                                  SLPLOT  2190
   34 AX0 =    -X1 /DX                                                          SLPLOT  2200
      AXL = (XL-X1)/DX                                                          SLPLOT  2210
   35 AX1 = AMIN1(AX0,AXL)                                                      SLPLOT  2220
      AX2 = AMAX1(AX0,AXL)                                                      SLPLOT  2230
      DY  = Y2 - Y1                                                             SLPLOT  2240
      IF  (DY.NE.0.0)  GO TO 36                                                 SLPLOT  2250
      AY0 = 1.0                                                                 SLPLOT  2260
      AYL = 0.0                                                                 SLPLOT  2270
      IF  (Y1.GE.0.0)  AY0 = 0.0                                                SLPLOT  2280
      IF  (Y1.LE.YL )  AYL = 1.0                                                SLPLOT  2290
      GO TO 37                                                                  SLPLOT  2300
   36 AY0 =    -Y1 /DY                                                          SLPLOT  2310
      AYL = (YL-Y1)/DY                                                          SLPLOT  2320
   37 AY1 = AMIN1(AY0,AYL)                                                      SLPLOT  2330
      AY2 = AMAX1(AY0,AYL)                                                      SLPLOT  2340
      A1  = AMAX1(AX1,AY1,0.0)                                                  SLPLOT  2350
      A2  = AMIN1(AX2,AY2,1.0)                                                  SLPLOT  2360
      IF  (A1.GE.A2 )  GO TO 39                                                 SLPLOT  2370
      XP = X1 + A1*DX                                                           SLPLOT  2380
      YP = Y1 + A1*DY                                                           SLPLOT  2390
      CALL PLOT(XP,YP,IPEN)                                                     SLPLOT  2400
      IPEN = 2                                                                  SLPLOT  2410
      XP = X1 + A2*DX                                                           SLPLOT  2420
      YP = Y1 + A2*DY                                                           SLPLOT  2430
      CALL PLOT(XP,YP,IPEN)                                                     SLPLOT  2440
      IF  (A2.NE.1.0)  IPEN = 3                                                 SLPLOT  2450
   39 CONTINUE                                                                  SLPLOT  2460
   40 CONTINUE                                                                  SLPLOT  2470
      RETURN                                                                    SLPLOT  2480
      END                                                                       SLPLOT  2490
```

```
      SUBROUTINE SPDAMP                                                 SPDAMP  0010
C                                                          REV 20 05/19/80 SPDAMP  0020
C     COMPUTES THE SPRING AND VISCOUS FORCE OF A SPRING DAMPER BETWEEN  SPDAMP  0030
C     SPECIFIED POINTS ON SELECTED SEGMENTS AND ADDS THE RESULTING      SPDAMP  0040
C     FORCE AND TORQUE TO THE U1 AND U2 ARRAYS.                         SPDAMP  0050
C                                                                       SPDAMP  0060
      IMPLICIT REAL*8(A-H,O-Z)                                          SPDAMP  0070
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           SPDAMP  0080
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)        SPDAMP  0090
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30), SPDAMP  0100
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       SPDAMP  0110
      COMMON/DAMPER/ APSDM(3,20),APSDN(3,20),ASD(5,20),MSDM(20),MSDN(20) SPDAMP  0120
      COMMON/TABLES/ MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(500),TAB(2600) SPDAMP  0130
      COMMON/FORCES/ PSF(7,30),BSF(4,20),SSF(10,20),BAGSF(3,20),        SPDAMP  0140
     *               PRJNT(7,30),NPANEL(5),NPSF,NBSF,NSSF,NBGSF         SPDAMP  0150
      COMMON/TEMPVS/DELM(3),DELN(3),DD(3),DEL,T1(3),T2(3),T3(3),T4(3),  SPDAMP  0160
     *              DUNIT(3),DV(3),DMV,DD0,FS,FD,TOTF(3),               SPDAMP  0170
     *              T5(3),T6(3),T7(3),T8(3)                             SPDAMP  0180
      CALL ELTIME(1,32)                                                 SPDAMP  0190
      NBSF0 = NBSF                                                      SPDAMP  0200
      DO 90 I=1,NSD                                                     SPDAMP  0210
      M = MSDM(I)                                                       SPDAMP  0220
      N = MSDN(I)                                                       SPDAMP  0230
C                                                                       SPDAMP  0240
C     COMPUTE VECTOR AND ITS MAGNITUDE BETWEEN THE SPECIFIED POINTS.    SPDAMP  0250
C                                                                       SPDAMP  0260
      CALL DOT31 (D(1,1,M),APSDM(1,I),DELM)                             SPDAMP  0270
      CALL DOT31 (D(1,1,N),APSDN(1,I),DELN)                             SPDAMP  0280
      DEL = 0.0                                                         SPDAMP  0290
      DO 10 K=1,3                                                       SPDAMP  0300
      DD(K) = SEGLP(K,M)+DELM(K)-SEGLP(K,N)-DELN(K)                     SPDAMP  0310
   10 DEL = DEL+DD(K)**2                                                SPDAMP  0320
      IF  (DEL.LE.0.0)  GO TO 90                                        SPDAMP  0330
      DEL = DSQRT(DEL)                                                  SPDAMP  0340
C                                                                       SPDAMP  0350
C     COMPUTE RELATIVE VELOCITY AND ITS COMPONENT ON VECTOR LINE.       SPDAMP  0360
C                                                                       SPDAMP  0370
      CALL CROSS(WMEG(1,M),APSDM(1,I),T1)                               SPDAMP  0380
      CALL CROSS(WMEG(1,N),APSDN(1,I),T2)                               SPDAMP  0390
      CALL DOT31 (D(1,1,M),T1,T3)                                       SPDAMP  0400
      CALL DOT31 (D(1,1,N),T2,T4)                                       SPDAMP  0410
      DO 20 K=1,3                                                       SPDAMP  0420
      DUNIT(K) = DD(K)/DEL                                              SPDAMP  0430
   20 DV(K) = SEGLV(K,M)+T3(K)-SEGLV(K,N)-T4(K)                         SPDAMP  0440
      DMV = DUNIT(1)*DV(1)+DUNIT(2)*DV(2)+DUNIT(3)*DV(3)                SPDAMP  0450
C                                                                       SPDAMP  0460
C     COMPUTE SPRING AND VISCOUS FORCE AND THE COMPONENTS               SPDAMP  0470
C     ALONG THE UNIT VECTOR                                             SPDAMP  0480
C                                                                       SPDAMP  0490
      FS = 0.0                                                          SPDAMP  0500
```

```
      FD = 0.0                                                          SPDAMP  0510
      IF  (ASD(1,I).LE.0.0)  GO TO 21                                   SPDAMP  0520
      DD0 = DEL-ASD(1,I)                                                SPDAMP  0530
      IF  (DD0.LE.0.0 .AND. ASD(2,I).LE.0.0)  GO TO 41                  SPDAMP  0540
      FS = DD0*(DABS(ASD(2,I)) + DABS(DD0)*ASD(3,I))                    SPDAMP  0550
      FD = DMV*(ASD(4,I)+DABS(DMV)*ASD(5,I))                            SPDAMP  0560
      GO TO 29                                                          SPDAMP  0570
   21 DD0 = DEL+ASD(1,I)                                                SPDAMP  0580
      JF1 = ASD(2,I)                                                    SPDAMP  0590
      IF  (JF1.EQ.0)  GO TO 22                                          SPDAMP  0600
      JF2 = NTI(JF1)                                                    SPDAMP  0610
      IF  (DD0.GT.0.0 .OR. ASD(3,I).EQ.0.0)  FS = EVALFD(DD0,JF2,1)     SPDAMP  0620
   22 JF3 = ASD(4,I)                                                    SPDAMP  0630
      IF  (JF3.EQ.0)  GO TO 29                                          SPDAMP  0640
      JF4 = NTI(JF3)                                                    SPDAMP  0650
      IF  (DD0.GT.0.0 .OR. ASD(3,I).EQ.0.0)  FD = EVALFD(DD0,JF4,1)     SPDAMP  0660
   29 DO 30 K=1,3                                                       SPDAMP  0670
   30 TOTF(K) = (FS+FD)*DUNIT(K)                                        SPDAMP  0680
C                                                                       SPDAMP  0690
C     AND ADD THE RESULTING FORCE AND TORQUE TO THE U1 AND U2 ARRAYS.   SPDAMP  0700
C                                                                       SPDAMP  0710
      CALL MAT31(D(1,1,M),TOTF,T5)                                      SPDAMP  0720
      CALL MAT31(D(1,1,N),TOTF,T6)                                      SPDAMP  0730
      CALL CROSS(APSDM(1,I),T5,T7)                                      SPDAMP  0740
      CALL CROSS(APSDN(1,I),T6,T8)                                      SPDAMP  0750
      DO 40 K=1,3                                                       SPDAMP  0760
      U1(K,M) = U1(K,M) - TOTF(K)                                       SPDAMP  0770
      U1(K,N) = U1(K,N) + TOTF(K)                                       SPDAMP  0780
      U2(K,M) = U2(K,M) - T7(K)                                         SPDAMP  0790
   40 U2(K,N) = U2(K,N) + T8(K)                                         SPDAMP  0800
   41 IBSF = 3-2*MOD(I,2)                                               SPDAMP  0810
      NBSF = NBSF0 + (I+1)/2                                            SPDAMP  0820
      BSF(IBSF  ,NBSF) = DEL                                            SPDAMP  0830
      BSF(IBSF+1,NBSF) = FD + FS                                        SPDAMP  0840
   90 CONTINUE                                                          SPDAMP  0850
      CALL ELTIME(2,32)                                                 SPDAMP  0860
      RETURN                                                            SPDAMP  0870
      END                                                               SPDAMP  0880
```

```
      SUBROUTINE SPLINE (X,Y,F,N,L)                                      SPLINE  0010
C                                                        REV 19 05/14/79  SPLINE  0020
C                                                                         SPLINE  0030
C     ROUTINE TO FIT A SET OF POLYNOMIALS OF DEGREE L                     SPLINE  0040
C     TO A SET OF GIVEN DATA POINTS (X(I),Y(I),I=1,N)                     SPLINE  0050
C                                                                         SPLINE  0060
C     FUNCTION IS OF FORM:                                                SPLINE  0070
C                                                                         SPLINE  0080
C       Y = F(2,K) + F(3,K)*DX + F(4,K)*DX**2 + F(5,K)*DX**3              SPLINE  0090
C                                                                         SPLINE  0100
C       WHERE: DX = XX - F(1,K)                                           SPLINE  0110
C              F(1,K) .LE. XX .LT. F(1,K+1)  ;  (SETS K)                  SPLINE  0120
C              IF  (XX.GT.F(1,N))  ;  USE K=N, CONSTANT FIT TO Y(N)       SPLINE  0130
C              IF  (XX.LT.F(1,1))  ;  EXTRAPOLATED FIT FOR K=1            SPLINE  0140
C                                                                         SPLINE  0150
C                 F(1,I) = X(I) ,                  I=1,N                  SPLINE  0160
C                 F(2,I) = Y(I) ,                  I=1,N                  SPLINE  0170
C                                                                         SPLINE  0180
C      DEGREE L                                                 CONTINUITY SPLINE  0190
C              0   F(3,I) = F(4,I) = F(5,I) = 0 , I=1,N          NONE     SPLINE  0200
C              1   F(4,I) = F(5,I) = 0  ,        I=1,N          Y        SPLINE  0210
C              2   F(5,I) = 0 ,                  I=1,N          Y,Y'     SPLINE  0220
C              3   CUBIC SPLINE                                 Y,Y',Y'' SPLINE  0230
C                                                                         SPLINE  0240
C                 F(K,N)=0 FOR K=3,5 IN ALL CASES                         SPLINE  0250
C                                                                         SPLINE  0260
C     FOR L=2 AND L=3 THE CHANGES IN THE L'TH DERIVATIVES ARE MINIMIZED   SPLINE  0270
C                                                                         SPLINE  0280
C     SPECIAL CASES:                                                      SPLINE  0290
C         N=1 ;     TREATED AS L=0                                      SPLINE  0300
C         N=2 ;     TREATED AS L=MIN(L,1)                               SPLINE  0310
C         L<0 ;     TREATED AS L=0                                      SPLINE  0320
C         L>3 ;     TREATED AS L=3                                      SPLINE  0330
C                                                                         SPLINE  0340
C     STORAGE REQUIRED  X(N),Y(N),F(5,N); SET BY CALLING PROGRAM          SPLINE  0350
C                                                                         SPLINE  0360
C     USAGE:                                                              SPLINE  0370
C       ALL COMPUTATIONS AND REAL VARIABLES ARE DOUBLE PRECISION          SPLINE  0380
C       GIVEN:  L,N, (X(I),Y(I),I=1,N)                                    SPLINE  0390
C       CALL SPLINE (X,Y,F,N,L)          ; SETS F                         SPLINE  0400
C                                                                         SPLINE  0410
CC                                                                        SPLINE  0420
CC      TO EVALUATE FUNCTION AND DERIVATIVES AT POINT XX                  SPLINE  0430
CC                                                                        SPLINE  0440
C           DO  10  K=1,N                                                 SPLINE  0450
C           IF (K.EQ.N)  GO TO 11                                         SPLINE  0460
C           IF (XX.LT.F(1,K+1)) GO TO 11                                  SPLINE  0470
C       10  CONTINUE                                                      SPLINE  0480
C       11  DX = XX - F(1,K)                                              SPLINE  0490
C           YY = F(2,K) + DX*(F(3,K)+DX*(F(4,K)+DX*F(5,K)))               SPLINE  0500
```

```
C           YD = F(3,K) + DX*(2.0*F(4,K)+3.0*DX*F(5,K))                    SPLINE  0510
C           YDD = 2.0*F(4,K) + 6.0*DX*F(5,K)                               SPLINE  0520
C           YDDD = 6.0*F(5,K)                                              SPLINE  0530
C           YDDDD = 0.0                                                    SPLINE  0540
CC                                                                         SPLINE  0550
CC      FUNCTIONAL VALUE IN YY, DERIVATIVES IN YD'S                        SPLINE  0560
CC      REPEAT FOR NEXT VALUE OF XX                                        SPLINE  0570
C                                                                          SPLINE  0580
C     AUTHOR: DR. JOHN T. FLECK                                            SPLINE  0590
C                                                                          SPLINE  0600
      IMPLICIT REAL*8 (A-H,O-Z)                                            SPLINE  0610
      DIMENSION X(N),Y(N),F(5,N),C(2,3)                                    SPLINE  0620
      DO  20  I=1,N                                                        SPLINE  0630
      F(1,I) = X(I)                                                        SPLINE  0640
      DO  10  K=2,5                                                        SPLINE  0650
   10 F(K,I) = 0.0                                                         SPLINE  0660
      IF  (L.LT.3)  F(2,I) = Y(I)                                          SPLINE  0670
   20 IF  (L.GT.0 .AND. I.LT.N)  F(3,I) = (Y(I+1)-Y(I))/(X(I+1)-X(I))      SPLINE  0680
      IF  (L.LT.2 .OR.  N.LT.3)  GO TO 99                                  SPLINE  0690
      IF  (L.GE.3)  GO TO 50                                               SPLINE  0700
      D1 = X(2) - X(1)                                                     SPLINE  0710
      SS = 0.0                                                             SPLINE  0720
      DS = 0.0                                                             SPLINE  0730
      DO  30  I=3,N                                                        SPLINE  0740
      F(4,I-1) = F(3,I-1) - F(3,I-2) - F(4,I-2)                            SPLINE  0750
      DX1 = X(I) - X(I-1)                                                  SPLINE  0760
      DX2 = X(I-1) - X(I-2)                                                SPLINE  0770
      DD = D1/DX1 + D1/DX2                                                 SPLINE  0780
      SS = SS + DD*DD                                                      SPLINE  0790
      DS = DS + DD*(F(4,I-1)/DX1 - F(4,I-2)/DX2)                           SPLINE  0800
   30 D1 = -D1                                                             SPLINE  0810
      F(4,1) = DS/SS                                                       SPLINE  0820
      DX = (X(2)-X(1))*F(4,1)                                              SPLINE  0830
      F(3,1) = F(3,1) - DX                                                 SPLINE  0840
      DO  40  I=3,N                                                        SPLINE  0850
      XX = F(4,I-1) - DX                                                   SPLINE  0860
      F(3,I-1) = F(3,I-1) - XX                                             SPLINE  0870
      F(4,I-1) = XX/(X(I)-X(I-1))                                          SPLINE  0880
   40 DX = -DX                                                             SPLINE  0890
      GO TO 99                                                             SPLINE  0900
C                                                                          SPLINE  0910
C     CUBIC SPLINE                                                         SPLINE  0920
C                                                                          SPLINE  0930
   50 DO  51  I=2,N                                                        SPLINE  0940
      IF  (I.EQ.N)  GO TO 51                                               SPLINE  0950
      F(4,I) = 3.0*(F(3,I)-F(3,I-1))                                       SPLINE  0960
      F(5,I) = 2.0*(X(I+1)-X(I-1))                                         SPLINE  0970
   51 F(3,I-1) = 0.0                                                       SPLINE  0980
      F(2,N) = -1.0                                                        SPLINE  0990
      F(3,1) = -1.0                                                        SPLINE  1000
```

```
      DO  60  I=3,N                                                    SPLINE  1010
      DX = X(I-1) - X(I-2)                                             SPLINE  1020
      IF  (I.GT.3)  DX = DX/F(5,I-2)                                   SPLINE  1030
      DO  60  K=3,5                                                    SPLINE  1040
   60 F(K,I-1) = F(K,I-1)- F(K,I-2)*DX**((K-1)/2)                      SPLINE  1050
      DO  70  I=3,N                                                    SPLINE  1060
      NI = N-I                                                         SPLINE  1070
      DX = X(NI+3) - X(NI+2)                                           SPLINE  1080
      DO  70  K=2,4                                                    SPLINE  1090
   70 F(K,NI+2) = (F(K,NI+2) - DX*F(K,NI+3))/F(5,NI+2)                 SPLINE  1100
      DO  71  J=1,2                                                    SPLINE  1110
      DO  71  K=J,3                                                    SPLINE  1120
      C(J,K) = 0.0                                                     SPLINE  1130
      DO  71  I=3,N                                                    SPLINE  1140
      DX1 = X(I) - X(I-1)                                              SPLINE  1150
      DX2 = X(I-1) - X(I-2)                                            SPLINE  1160
   71 C(J,K) = C(J,K) + (  (F(J+1,I  )-F(J+1,I-1))/DX1                 SPLINE  1170
     *                   - (F(J+1,I-1)-F(J+1,I-2))/DX2)                SPLINE  1180
     *                 * (  (F(K+1,I  )-F(K+1,I-1))/DX1                SPLINE  1190
     *                   - (F(K+1,I-1)-F(K+1,I-2))/DX2)                SPLINE  1200
      DEN     =  C(1,1)*C(2,2) - C(1,2)*C(1,2)                         SPLINE  1210
      F(4,1) = (C(1,1)*C(2,3) - C(1,2)*C(1,3))/DEN                     SPLINE  1220
      F(4,N) = (C(2,2)*C(1,3) - C(1,2)*C(2,3))/DEN                     SPLINE  1230
      DO  72  I=3,N                                                    SPLINE  1240
   72 F(4,I-1) = F(4,I-1) - F(4,1)*F(3,I-1) - F(4,N)*F(2,I-1)          SPLINE  1250
      D1 = X(2) - X(1)                                                 SPLINE  1260
      F(3,1) = (Y(2)-Y(1))/D1 - (2.0*F(4,1)+F(4,2))*D1/3.0             SPLINE  1270
      F(2,1) = Y(1)                                                    SPLINE  1280
      DO  80  I=2,N                                                    SPLINE  1290
      F(2,I) = Y(I)                                                    SPLINE  1300
      DX = X(I) - X(I-1)                                               SPLINE  1310
      IF  (I.LT.N)  F(3,I) = F(3,I-1) + (F(4,I)+F(4,I-1))*DX           SPLINE  1320
   80 F(5,I-1) = (F(4,I)-F(4,I-1))/(3.0*DX)                            SPLINE  1330
      F(4,N) = 0.0                                                     SPLINE  1340
   99 RETURN                                                           SPLINE  1350
      END                                                              SPLINE  1360
```

```
      DOUBLE PRECISION FUNCTION SPRNGF(T,D,ZD,SPR,JSTOP)                    SPRNGF  0010
C                                                          REV 19 08/05/78  SPRNGF  0020
C     COMPUTES NONLINEAR SPRING TORQUE FOR JOINTS AS A FUNCTION OF ANGLE   SPRNGF  0030
C     ACTUALLY ROUTINE RETURNS TORQUE/ABS(SIN THETA)                       SPRNGF  0040
C                                                                          SPRNGF  0050
C     ARGUMENTS:                                                           SPRNGF  0060
C          T     : COS THETA WHERE THETA IS ANGLE OF JOINT (0<THETA<PI)     SPRNGF  0070
C          D     : ABS(SIN THETA)                                           SPRNGF  0080
C          ZD    : -THETA DOT * SIN THETA                                   SPRNGF  0090
C          SPR   : ARRAY OF 5 VALUES DESCRIBING FUNCTION EVALUATION         SPRNGF  0100
C          JSTOP : INDICATOR TO BE SET TO ONE IF JOINT IS IN STOP           SPRNGF  0110
C                                                                           SPRNGF  0120
      IMPLICIT REAL*8 (A-H,O-Z)                                            SPRNGF  0130
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                            SPRNGF  0140
     *               UNITL,UNITM,UNITT,GRAVTY(3)                           SPRNGF  0150
      DIMENSION SPR(5)                                                     SPRNGF  0160
C                                                                          SPRNGF  0170
C      RESET T=1 IF T>1 (HAD & HBD IN VISPR)                               SPRNGF  0180
C                                                                          SPRNGF  0190
      IF (T.GT.1.0) T = 1.0                                                SPRNGF  0200
      IF (T.LT.-1.0) T = -1.0                                              SPRNGF  0210
      Z = DARCOS(T)                                                        SPRNGF  0220
      U = EPS(1)*D                                                         SPRNGF  0230
      Q = 0.0                                                              SPRNGF  0240
      IF (D.NE.0.0) Q = -ZD/U                                              SPRNGF  0250
      IF (Q.GT.1.0) Q = 1.0                                                SPRNGF  0260
      IF (Q.LT.-1.0) Q = -1.0                                              SPRNGF  0270
      X = 0.5*(1.0+SPR(4) + Q*(1.0-SPR(4)) )                               SPRNGF  0280
      Y = 0.0                                                              SPRNGF  0290
      IF (D.NE.0.0) Y = Z/D                                                SPRNGF  0300
      Q = 1.0                                                              SPRNGF  0310
      IF (DABS(Z).LT.EPS(4)) Y = DSIGN(Q,Z)                                SPRNGF  0320
      SPRNGF = Y*SPR(1)                                                    SPRNGF  0330
      JSTOP = 0                                                            SPRNGF  0340
      IF (SPR(5).GT.0.0) GO TO 10                                          SPRNGF  0350
      SPRNGF = X*SPRNGF                                                    SPRNGF  0360
      GO TO 11                                                             SPRNGF  0370
   10 IF (Z.LT.SPR(5)) GO TO 11                                            SPRNGF  0380
      JSTOP = 1                                                            SPRNGF  0390
      Z = Z-SPR(5)                                                         SPRNGF  0400
      SPRNGF = SPRNGF + X/D*(SPR(2)+Z*SPR(3))*Z**2                         SPRNGF  0410
   11 CONTINUE                                                             SPRNGF  0420
      RETURN                                                               SPRNGF  0430
      END                                                                  SPRNGF  0440
```

```
      SUBROUTINE TRIGFS                                                 TRIGFS  0010
C                                                        REV 19 08/05/78  TRIGFS  0020
      IMPLICIT REAL*8 (A-H,O-Z)                                         TRIGFS  0030
      COMMON/CDINT/ UU(4),GH(3,4),                                      TRIGFS  0040
     *              E(3,240), F(5,240),GG(5,240),Y(5,240),U(5,240),     TRIGFS  0050
     *              H,HPRINT,HS,TPRINT,TSTART,ICNT,IDBL,IFLAG           TRIGFS  0060
      BETA = 0.0                                                        TRIGFS  0070
      IF (HS.NE.0.0) BETA = (H/HS)**2                                   TRIGFS  0080
      R1 = HS/H                                                         TRIGFS  0090
      R2 = 1.0+BETA*R1                                                  TRIGFS  0100
      GH(3,1) = 2.0/(H*R2)                                              TRIGFS  0110
      GH(2,1) = GH(3,1)*(BETA-1.0)                                      TRIGFS  0120
      GH(1,1) = GH(3,1)* BETA                                           TRIGFS  0130
      GH(1,2) = 4.0*BETA/(R2*H**2)                                      TRIGFS  0140
      GH(3,2) = GH(1,2)* R1                                             TRIGFS  0150
      GH(2,2) = GH(1,2)*(R1+1.0)                                        TRIGFS  0160
      GH(3,3) = 1.0/H                                                   TRIGFS  0170
      GH(2,3) = 4.0*GH(3,3)                                             TRIGFS  0180
      GH(1,3) = 3.0*GH(3,3)                                             TRIGFS  0190
      GH(3,4) = 2.0/H**2                                                TRIGFS  0200
      GH(2,4) = 2.0*GH(3,4)                                             TRIGFS  0210
      GH(1,4) =     GH(3,4)                                             TRIGFS  0220
      UU(1) = 2.0/H                                                     TRIGFS  0230
      UU(2) = 0.0                                                       TRIGFS  0240
      UU(3) = 0.5*H                                                     TRIGFS  0250
      UU(4) = 0.25*H**2                                                 TRIGFS  0260
      IF  (HS.EQ.0.0)  GO TO 99                                         TRIGFS  0270
      UU(1) = BETA*(4.25+2.25/R1)                                       TRIGFS  0280
      UU(2) = BETA*(2.25+1.25/R1)/R1                                    TRIGFS  0290
      UAU   = 1.0+UU(1)+UU(2)                                           TRIGFS  0300
      UU(1) = 2.0*UU(1)/(UAU*H)                                         TRIGFS  0310
      UU(2) = 4.0*UU(2)/(UAU*H**2)                                      TRIGFS  0320
   99 RETURN                                                            TRIGFS  0330
      END                                                               TRIGFS  0340
```

```
      SUBROUTINE UNIT1(IND)                                                  UNIT1   0010
C                                                          REV 20 01/22/80   UNIT1   0020
C     THIS SUBROUTINE REPLACES THE PROGRAM CODE THAT PREVIOUSLY WAS          UNIT1   0030
C     NEAR THE END OF THE MAIN PROGRAM TO WRITE ON UNIT 1 THAT DATA          UNIT1   0040
C     USED FOR VARIOUS PLOTTING PROGRAMS (E.G. BUBBLE MAN PLOT).             UNIT1   0050
C                                                                            UNIT1   0060
C     THIS SUBROUTINE IS WRITTEN TO GENERATE UNIT 1 IN SUCH A MANNER         UNIT1   0070
C     TO BE COMPATIBLE WITH THE INPUT REQUIREMENTS FOR THE AMRL VIEW         UNIT1   0080
C     PROGRAM THAT IS NOW BEING DISTRIBUTED ON THE CVS PROGRAM TAPES.        UNIT1   0090
C                                                                            UNIT1   0100
C     ARGUMENTS:                                                             UNIT1   0110
C          IND = 0: CALL IS FROM THE MAIN PROGRAM                            UNIT1   0120
C              # 0: CALL IS FROM SUBROUTINE EQUILB                          UNIT1   0130
C                                                                            UNIT1   0140
      IMPLICIT REAL*8 (A-H,O-Z)                                              UNIT1   0150
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,                UNIT1   0160
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)             UNIT1   0170
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),     UNIT1   0180
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)            UNIT1   0190
      COMMON/CNTSRF/ PL(17,30),BELT(20,8),TPTS(6,8),BD(24,40)                UNIT1   0200
      COMMON/JBARTZ/ MNPL(   30),MNBLT(    8),MNSEG(   30),MNBAG(    6),   UNIT1   0210
     *               MPL(3,5,30),MBLT(3,5,8),MSEG(3,5,30),MBAG(3,10,6),     UNIT1   0220
     *               NTPL( 5,30),NTBLT( 5,8),NTSEG( 5,30)                    UNIT1   0230
      COMMON/RSAVE/  XSG(3,20,3),DPMI(3,3,30),LPMI(30),NSG(7),MSG(20,7)      UNIT1   0240
      COMMON/TEMPVS/ XD(3,3,30),XSEGLP(3,30),XPL(17,30),XBD(24,40),          UNIT1   0250
     *               T1(3),T3(3,3)                                           UNIT1   0260
      REAL XTIME,XD,XSEGLP,XPL,XBD                                           UNIT1   0270
      DATA IFIRST/0/                                                         UNIT1   0280
      IF  (NPRT(1).EQ.0)  GO TO 99                                           UNIT1   0290
      IF  (IFIRST.NE.0)  GO TO 20                                            UNIT1   0300
      IFIRST = 1                                                             UNIT1   0310
C                                                                            UNIT1   0320
C     FIRST TIME IN ROUTINE, WRITE STATIC DATA ON OUTPUT UNIT 1.             UNIT1   0330
C     DATA MUST BE CONVERTED TO SINGLE PRECISION FOR VIEW PROGRAM.           UNIT1   0340
C                                                                            UNIT1   0350
      DO  11  J=1,30                                                         UNIT1   0360
      DO  11  I=1,17                                                         UNIT1   0370
   11 XPL(I,J) = PL(I,J)                                                     UNIT1   0380
      DO  12  J=1,40                                                         UNIT1   0390
      DO  12  I=1,24                                                         UNIT1   0400
   12 XBD(I,J) = BD(I,J)                                                     UNIT1   0410
      DO  15  J=1,NSEG                                                       UNIT1   0420
      IF  (LPMI(J).EQ.0)  GO TO 15                                           UNIT1   0430
      CALL DOT31 (DPMI(1,1,J),BD(4,J),T1)                                    UNIT1   0440
      DO  14  I=1,3                                                          UNIT1   0450
   14 XBD(I+3,J) = T1(I)                                                     UNIT1   0460
   15 CONTINUE                                                               UNIT1   0470
      WRITE (1) NSEG,NPL,XPL,XBD,MPL                                         UNIT1   0480
C                                                                            UNIT1   0490
C     WRITE TIME POINT DATA ON OUTPUT UNIT 1.                                UNIT1   0500
```

```
C     DATA MUST BE CONVERTED TO SINGLE PRECISION FOR VIEW PROGRAM.          UNIT1   0510
C                                                                           UNIT1   0520
  20  XTIME = TIME                                                          UNIT1   0530
      DO  22  K=1,30                                                        UNIT1   0540
      DO  22  J=1,3                                                         UNIT1   0550
      DO  21  I=1,3                                                         UNIT1   0560
  21  XD(I,J,K) = D(I,J,K)                                                  UNIT1   0570
  22  XSEGLP(J,K) = SEGLP(J,K)                                              UNIT1   0580
      DO  25  K=1,NSEG                                                      UNIT1   0590
      IF  (LPMI(K).EQ.0)  GO TO 25                                          UNIT1   0600
      CALL DOT33 (DPMI(1,1,K),D(1,1,K),T3)                                  UNIT1   0610
      DO  24  I=1,3                                                         UNIT1   0620
      DO  24  J=1,3                                                         UNIT1   0630
  24  XD(I,J,K) = T3(I,J)                                                   UNIT1   0640
  25  CONTINUE                                                              UNIT1   0650
      WRITE (1) XTIME,XSEGLP,XD                                             UNIT1   0660
  99  RETURN                                                                UNIT1   0670
      END                                                                   UNIT1   0680
```

```
      SUBROUTINE UPDATE(I)                                                  UPDATE  0010
C                                                          REV 20 04/11/80  UPDATE  0020
C     CALLED BY SUBROUTINE DINT                                            UPDATE  0030
C                                                                          UPDATE  0040
C       (I=1)  AT THE START OF A NEW STEP TO SETUP ANY NEW CONDITIONS      UPDATE  0050
C              TO BE VALID FOR ENTIRE INTEGRATION STEP                     UPDATE  0060
C                A. UPDATE FORCE DEFLECTION FUNCTIONS(SUBROUTINE UPDFDC)   UPDATE  0070
C                B. TEST FOR LOCKED JOINTS                                 UPDATE  0080
C       NOTE: ARGUMENT I WILL BE SET TO -1 TO RESET INTEGRATOR.            UPDATE  0090
C                                                                          UPDATE  0100
C       (I=2)  AT THE END OF EACH SUCCESSFUL INTEGRATION STEP TO           UPDATE  0110
C              COMPLETE CALCULATIONS FOR OUTPUT (SUBROUTINE AIRBG3).       UPDATE  0120
C                                                                          UPDATE  0130
      IMPLICIT REAL*8(A-H,O-Z)                                             UPDATE  0140
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,              UPDATE  0150
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)           UPDATE  0160
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),   UPDATE  0170
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)          UPDATE  0180
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(3,60),HA(3,60),HB(3,60),    UPDATE  0190
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),        UPDATE  0200
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)       UPDATE  0210
      COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60),   UPDATE  0220
     *               F(3,30),TQ(3,30),WJ(30)                               UPDATE  0230
      COMMON/JBARTZ/ MNPL(   30),MNBLT(    8),MNSEG(   30),MNBAG(    6),  UPDATE  0240
     *               MPL(3,5,30),MBLT(3,5,8),MSEG(3,5,30),MBAG(3,10,6),    UPDATE  0250
     *               NTPL( 5,30),NTBLT( 5,8),NTSEG( 5,30)                  UPDATE  0260
      COMMON/TABLES/ MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(500),TAB(2600)   UPDATE  0270
      COMMON/FORCES/ PSF(7,30),BSF(4,20),SSF(10,20),BAGSF(3,20),           UPDATE  0280
     *               PRJNT(7,30),NPANEL(5),NPSF,NBSF,NSSF,NBGSF            UPDATE  0290
      COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),      UPDATE  0300
     *               HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12),   UPDATE  0310
     *               RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),                 UPDATE  0320
     *               KQ1(12),KQ2(12),KQTYPE(12)                            UPDATE  0330
      COMMON/TEMPVI/ CREST,TTI(3),R1I(3),R2I(3),JSTOP(4,2,30)              UPDATE  0340
      COMMON/CEULER/ IEULER(30),HIR(3,3,30),ANG(3,30),ANGD(3,30),          UPDATE  0350
     *               FE(3,30),TQE(3,30),CONST(3,30)                        UPDATE  0360
      COMMON/HRNESS/ BAR(15,100),BB(100),BBDOT(100),PLOSS(2,100),          UPDATE  0370
     *               XLONG(20),HTIME(2),IBAR(5,100),NL(2,100),             UPDATE  0380
     *               NPTSPB(20),NPTPLY(20),NTHRNS(20),NBLTPH(5)            UPDATE  0390
      DIMENSION TQTEST(3),LOCK(8,3) ,T(3)                                  UPDATE  0400
      DATA LOCK/-8, 6, 5, 7,-3,-2,-4, 1,                                   UPDATE  0410
     *           6,-8, 4,-3, 7,-1,-5, 2,                                   UPDATE  0420
     *           5, 4,-8,-2,-1, 7,-6, 3/                                   UPDATE  0430
C                                                                          UPDATE  0440
C     CALL AIRBG3 FOR AIRBAG, IF ANY.                                      UPDATE  0450
C                                                                          UPDATE  0460
      IF  (NBAG.NE.0)  CALL AIRBG3(I)                                      UPDATE  0470
      IF  (I.EQ.2)  GO TO 42                                               UPDATE  0480
      CALL ELTIME (1,7)                                                    UPDATE  0490
      IF  (NPL.LE.0)  GO TO 13                                             UPDATE  0500
```

```
C                                                                       UPDATE  0510
C     CALL UPDFDC FOR EACH ALLOWED PLANE-SEGMENT CONTACT.               UPDATE  0520
C                                                                       UPDATE  0530
      NPSF = 0                                                          UPDATE  0540
      DO 12 J=1,NPL                                                     UPDATE  0550
      NK = MNPL(J)                                                      UPDATE  0560
      IF (NK.LE.0) GO TO 12                                             UPDATE  0570
      DO 11 K = 1, NK                                                   UPDATE  0580
      NPSF = NPSF+1                                                     UPDATE  0590
      NT = NTPL(K,J)                                                    UPDATE  0600
      NF = NTAB(NT+5)                                                   UPDATE  0610
      CALL UPDFDC(NT)                                                   UPDATE  0620
      IF (NT.GT.0.OR.TAB(NF+3).EQ.0.0) GO TO 11                         UPDATE  0630
      CALL IMPULS(1,K,J)                                                UPDATE  0640
      I = -1                                                            UPDATE  0650
   11 CONTINUE                                                          UPDATE  0660
   12 CONTINUE                                                          UPDATE  0670
   13 IF (NBLT.LE.0) GO TO 16                                           UPDATE  0680
C                                                                       UPDATE  0690
C     CALL UPDFDC FOR EACH ALLOWED BELT-SEGMENT CONTACT.                UPDATE  0700
C                                                                       UPDATE  0710
      DO 15 J=1,NBLT                                                    UPDATE  0720
      NK = MNBLT(J)                                                     UPDATE  0730
      IF (NK.LE.0) GO TO 15                                             UPDATE  0740
      DO 14 K = 1,NK                                                    UPDATE  0750
      NT = NTBLT(K,J)                                                   UPDATE  0760
      NF = NTAB(NT+5)                                                   UPDATE  0770
      NT6 = NT+6                                                        UPDATE  0780
      CALL UPDFDC(NT)                                                   UPDATE  0790
C                                                                       UPDATE  0800
C        AND FOR 2ND FUNCTION, IF FULL BELT FRICTION.                   UPDATE  0810
C                                                                       UPDATE  0820
   14 IF (NF.NE.0) CALL UPDFDC(NT6)                                     UPDATE  0830
   15 CONTINUE                                                          UPDATE  0840
C                                                                       UPDATE  0850
C     CALL UPDFDC FOR EACH ALLOWED SEGMENT-SEGMENT CONTACT.             UPDATE  0860
C                                                                       UPDATE  0870
   16 NSSF = 0                                                          UPDATE  0880
      DO 18 J=1,NSEG                                                    UPDATE  0890
      NK = MNSEG(J)                                                     UPDATE  0900
      IF (NK.LE.0) GO TO 18                                             UPDATE  0910
      DO 17 K = 1,NK                                                    UPDATE  0920
      NSSF = NSSF+1                                                     UPDATE  0930
      NT = NTSEG(K,J)                                                   UPDATE  0940
      NF = NTAB(NT+5)                                                   UPDATE  0950
      CALL UPDFDC(NT)                                                   UPDATE  0960
      IF (NT.GT.0.OR.TAB(NF+3).EQ.0.0) GO TO 17                         UPDATE  0970
      CALL IMPULS(3,K,J)                                                UPDATE  0980
      I = -1                                                            UPDATE  0990
   17 CONTINUE                                                          UPDATE  1000
```

```
   18 CONTINUE                                                          UPDATE  1010
      IF (NHRNSS.LE.0) GO TO 71                                         UPDATE  1020
C                                                                       UPDATE  1030
C     CALL UPDFDC FOR EACH BELT OF HARNESS-BELT SYSTEMS.                UPDATE  1040
C                                                                       UPDATE  1050
      CALL HPTURB                                                       UPDATE  1060
      J1 = 1                                                            UPDATE  1070
      K1 = 1                                                            UPDATE  1080
      DO 70 II=1,NHRNSS                                                 UPDATE  1090
      IF (NBLTPH(II).LE.0) GO TO 70                                     UPDATE  1100
      J2 = J1 + NBLTPH(II) - 1                                          UPDATE  1110
      DO 69 J=J1,J2                                                     UPDATE  1120
      IF  (NPTPLY(J).LE.0)  GO TO 69                                    UPDATE  1130
      NT = NTHRNS(J)                                                    UPDATE  1140
      CALL UPDFDC(NT)                                                   UPDATE  1150
      K2 = K1 + NPTPLY(J) - 1                                           UPDATE  1160
      DO  68  K=K1,K2                                                   UPDATE  1170
      KI = NL(1,K)                                                      UPDATE  1180
      NT = IBAR(3,KI)                                                   UPDATE  1190
      CALL UPDFDC(NT)                                                   UPDATE  1200
   68 CONTINUE                                                          UPDATE  1210
      K1 = K2+1                                                         UPDATE  1220
   69 CONTINUE                                                          UPDATE  1230
      J1 = J2+1                                                         UPDATE  1240
   70 CONTINUE                                                          UPDATE  1250
   71 IF (NJNT.LE.0) GO TO 37                                           UPDATE  1260
C                                                                       UPDATE  1270
C           CHECK FOR IMPULSE ON JOINT STOPS                            UPDATE  1280
C     TO BE CALLED IF IN JOINT STOP (JSTOP(1)=1) THIS TIME STEP         UPDATE  1290
C     BUT NOT IN IN JOINT STOP (JSTOP(2)=0) AT PREVIOUS TIME.           UPDATE  1300
C                                                                       UPDATE  1310
      DO 21 K=1,NJNT                                                    UPDATE  1320
      IF (JNT(K).EQ.0) GO TO 21                                         UPDATE  1330
      IF (IABS(IPIN(K)).NE.4 .AND. VISC(7,3*K-2).EQ.0.0) GO TO 20       UPDATE  1340
      DO 19 J=1,3                                                       UPDATE  1350
      K3J = 3*K-3+J                                                     UPDATE  1360
      IF (IABS(IPIN(K)).NE.4) K3J=3*K-2                                 UPDATE  1370
      IF (IABS(IPIN(K)).EQ.4 .AND. VISC(7,K3J).EQ.0.0) GO TO 19         UPDATE  1380
      IF (JSTOP(J,1,K).NE.1.OR.JSTOP(J,2,K).NE.0) GO TO 19              UPDATE  1390
      CALL IMPULS(4,J,K)                                                UPDATE  1400
      I = -1                                                            UPDATE  1410
   19 JSTOP(J,2,K) = JSTOP(J,1,K)                                       UPDATE  1420
   20 IF (IGLOB(K).EQ.0) GO TO 21                                       UPDATE  1430
      NT = IGLOB(K)                                                     UPDATE  1440
      MT = NTAB(NT+5)                                                   UPDATE  1450
      NT1 = NTAB(NT+2)                                                  UPDATE  1460
      NTAB(NT+2) = 0                                                    UPDATE  1470
      CALL UPDFDC(NT)                                                   UPDATE  1480
      NT = IABS(NT)                                                     UPDATE  1490
      NTAB(NT+2) = NT1                                                  UPDATE  1500
```

```
      IF (TAB(MT+3).EQ.0.0) GO TO 21                                    UPDATE  1510
      IF (JSTOP(4,1,K).NE.1.OR.JSTOP(4,2,K).NE.0) GO TO 21              UPDATE  1520
      CALL IMPULS(4,4,K)                                                UPDATE  1530
      I = -1                                                            UPDATE  1540
   21 JSTOP(4,2,K) = JSTOP(4,1,K)                                       UPDATE  1550
C                                                                       UPDATE  1560
C     TEST TO LOCK OR UNLOCK JOINTS                                     UPDATE  1570
C                                                                       UPDATE  1580
C                                                                       UPDATE  1590
C     CONDITIONS TO CHANGE SIGN OF IPIN(J)                              UPDATE  1600
C                                                                       UPDATE  1610
C                    PINNED                UNPINNED                     UPDATE  1620
C         LOCKED (-1) |H.TQ| > T1     (-2) |TQ| > T1                   UPDATE  1630
C                                                                       UPDATE  1640
C       UNLOCKED  (+1) |H.TQ| < T2     (+2) |TQ| < T2                  UPDATE  1650
C                          OR                    OR                     UPDATE  1660
C                       WJ  < T3              WJ  < T3                  UPDATE  1670
C                                                                       UPDATE  1680
      DO 28 J=1,NJNT                                                    UPDATE  1690
      IF (IABS(IPIN(J)).EQ.4) GO TO 28                                  UPDATE  1700
      IF (IPIN(J)) 22,28,23                                             UPDATE  1710
   22 T1 = VISC(4,3*J-2)                                                UPDATE  1720
      IF (T1.EQ.0.0) GO TO 28                                           UPDATE  1730
      IF  (IPIN(J).NE.-1)  GO TO 51                                     UPDATE  1740
      TQM = XDY(HB(1,2*J),D(1,1,J+1),TQ(1,J))                           UPDATE  1750
      ABSTQM = DABS(TQM)                                                UPDATE  1760
      IF  (ABSTQM.GT.T1)  HA(2,2*J-1) = TQM                             UPDATE  1770
      TQM = ABSTQM                                                      UPDATE  1780
      GO TO 52                                                          UPDATE  1790
   51 TQM = DSQRT(TQ(1,J)**2 + TQ(2,J)**2 + TQ(3,J)**2)                 UPDATE  1800
      IF  (TQM.GT.T1)  CALL DOT31(HIR(1,1,J),TQ(1,J),HA(1,2*J-1))       UPDATE  1810
   52 IF  (TQM-T1)  28,28,26                                            UPDATE  1820
   23 T2 = VISC(5,3*J-2)                                                UPDATE  1830
      IF  (HA(2,2*J).NE.0.0)  GO TO 54                                  UPDATE  1840
      DO  53  K=1,3                                                     UPDATE  1850
   53 HA(K,2*J-1) = 0.0                                                 UPDATE  1860
   54 IF  (T2.EQ.0.0)  GO TO 24                                         UPDATE  1870
      IF (IPIN(J).GE.2)  TQM = DSQRT(TQ(1,J)**2+TQ(2,J)**2+TQ(3,J)**2)  UPDATE  1880
      IF (IPIN(J).EQ. 1) TQM = DABS(XDY(HB(1,2*J),D(1,1,J+1),TQ(1,J)))  UPDATE  1890
      IF (TQM-T2) 25,28,28                                              UPDATE  1900
   24 T3 = VISC(6,3*J-2)                                                UPDATE  1910
      IF (T3.EQ.0.0) GO TO 28                                           UPDATE  1920
      IF (WJ(J)-T3) 25,28,28                                            UPDATE  1930
   25 CALL IMPLS2(0,J,HB(1,2*J))                                        UPDATE  1940
      I = -1                                                            UPDATE  1950
   26 IPIN(J) = -IPIN(J)                                                UPDATE  1960
      TMSEC = 1000.0*TIME                                               UPDATE  1970
      IPINJ = -IPIN(J)                                                  UPDATE  1980
      WRITE (6,27) TMSEC,J,IPINJ,IPIN(J)                                UPDATE  1990
   27 FORMAT('0 AT TIME =',F9.3,' MSEC,   IPIN(',I2,') HAS BEEN CHANGED  UPDATE  2000
```

```
     *FROM',I3,' TO',I3)                                                    UPDATE  2010
   28 CONTINUE                                                              UPDATE  2020
C                                                                           UPDATE  2030
C     TEST TO LOCK OR UNLOCK EULER JOINTS AXES.                             UPDATE  2040
C     USE SAME TEST AS ABOVE BUT ON EACH AXIS SERARATELY.                   UPDATE  2050
C                                                                           UPDATE  2060
C      IF LOCK(IEULER,K) IS NEGATIVE, AXIS K IS LOCKED;                     UPDATE  2070
C         TO UNLOCK AXIS SET IEULER TO -LOCK(IEULER,K).                     UPDATE  2080
C                                                                           UPDATE  2090
C     IF LOCK(IEULER,K) IS POSITIVE, AXIS K IS UNLOCKED;                    UPDATE  2100
C       TO LOCK AXIS SET IEULER TO LOCK(IEULER,K).                          UPDATE  2110
C                                                                           UPDATE  2120
      DO 36 J=1,NJNT                                                        UPDATE  2130
      IF (IABS(IPIN(J)).NE.4) GO TO 36                                      UPDATE  2140
      JEULER = IEULER(J)                                                    UPDATE  2150
      CALL DOT31(HIR(1,1,J),TQ(1,J),TQTEST)                                 UPDATE  2160
      DO 31 K=1,3                                                           UPDATE  2170
      K3J = 3*J-3+K                                                         UPDATE  2180
      NLOCK = LOCK(JEULER,K)                                                UPDATE  2190
      IF (NLOCK.GT.0) GO TO 29                                              UPDATE  2200
      IF (VISC(4,K3J).EQ.0.0) GO TO 31                                      UPDATE  2210
      IF (DABS(TQTEST(K)).LE.VISC(4,K3J)),GO TO 31                          UPDATE  2220
      JEULER = -NLOCK                                                       UPDATE  2230
      HA(K,2*J-1) = TQTEST(K)                                               UPDATE  2240
      GO TO 31                                                              UPDATE  2250
   29 IF  (HA(K,2*J).EQ.0.0)  HA(K,2*J-1) = 0.0                             UPDATE  2260
      IF (VISC(5,K3J).EQ.0.0) GO TO 30                                      UPDATE  2270
      IF (DABS(TQTEST(K)).LT.VISC(5,K3J)) JEULER =  NLOCK                   UPDATE  2280
      GO TO 31                                                              UPDATE  2290
   30 IF (VISC(6,K3J).EQ.0.0) GO TO 31                                      UPDATE  2300
      IF (DABS(ANGD(K,J)).LT.VISC(6,K3J)) JEULER =  NLOCK                   UPDATE  2310
   31 CONTINUE                                                              UPDATE  2320
      IF (JEULER.EQ.IEULER(J)) GO TO 36                                     UPDATE  2330
      TMSEC = 1000.0*TIME                                                   UPDATE  2340
      WRITE (6,32) TMSEC,J,IEULER(J),JEULER                                 UPDATE  2350
   32 FORMAT('0 AT TIME =',F9.3,' MSEC, IEULER(',I2,') HAS BEEN CHANGED     UPDATE  2360
     *FROM',I3,' TO',I3)                                                    UPDATE  2370
      IF (JEULER.EQ.8)          GO TO 35                                    UPDATE  2380
      IF (IEULER(J).EQ.7) GO TO 35                                          UPDATE  2390
      IF (IEULER(J).EQ.6 .AND. (JEULER.EQ.2.OR.JEULER.EQ.1)) GO TO 35       UPDATE  2400
      IF (IEULER(J).EQ.5 .AND. (JEULER.EQ.3.OR.JEULER.EQ.1)) GO TO 35       UPDATE  2410
      IF (IEULER(J).EQ.4 .AND. (JEULER.EQ.3.OR.JEULER.EQ.2)) GO TO 35       UPDATE  2420
      MODE = -1                                                             UPDATE  2430
      K = JEULER                                                            UPDATE  2440
      IF (K.GT.3) GO TO 33                                                  UPDATE  2450
      IF (K.EQ.2) GO TO 34                                                  UPDATE  2460
      K4 = 4-K                                                              UPDATE  2470
      CALL CROSS (HIR(1,K4,J),HIR(1,2,J),T)                                 UPDATE  2480
      IEULER(J) = 8                                                         UPDATE  2490
      IPIN(J)   = 4                                                         UPDATE  2500
```

```
      CALL IMPLS2(MODE,J,T)                                                  UPDATE  2510
      I = -1                                                                 UPDATE  2520
      GO TO 35                                                               UPDATE  2530
   33 MODE =  1                                                              UPDATE  2540
      K = K-3                                                                UPDATE  2550
      IF (K.GT.3) MODE=0                                                    .UPDATE  2560
   34 IEULER(J) = 8                                                          UPDATE  2570
      IPIN(J)   = 4                                                          UPDATE  2580
      CALL IMPLS2(MODE,J,HIR(1,K,J))                                         UPDATE  2590
      I = -1                                                                 UPDATE  2600
   35 IEULER(J) = JEULER                                                     UPDATE  2610
      IPIN(J) = 4                                                            UPDATE  2620
      IF (IEULER(J).NE.8) IPIN(J) = -4                                       UPDATE  2630
   36 CONTINUE                                                               UPDATE  2640
C                                                                            UPDATE  2650
   37 IF  (NQ.LE.0)  GO TO 41                                                UPDATE  2660
      DO 40 K=1,NQ                                                           UPDATE  2670
      IF (KQTYPE(K).LT.3) GO TO 40                                           UPDATE  2680
      IF (KQTYPE(K).GT.4) GO TO 40                                           UPDATE  2690
      IF (CFQQ(K).LT.0.0) KQTYPE(K) = -KQTYPE(K)                             UPDATE  2700
      IF (CFQQ(K).LT.0.0) GO TO 39                                           UPDATE  2710
C                                                                            UPDATE  2720
C     TEST IF ROLLING CONSTRAINT SHOULD BE SLIDING AND VICE VERSA.           UPDATE  2730
C                                                                            UPDATE  2740
      QN = -XDY(TQQ(1,K),HHT(1,1,K),QQ(1,K))                                 UPDATE  2750
      IF (NPRT(24).NE.0) WRITE (6,38) KQTYPE(K),KQ1(K),KQ2(K),               UPDATE  2760
     *                   (RK1(II,K),II=1,3),(RK2(II,K),II=1,3),              UPDATE  2770
     *                   ((HHT(II,J,K),J=1,3),II=1,3),                       UPDATE  2780
     *      (QQ(II,K),II=1,3),(TQQ(II,K),II=1,3),(RQQ(II,K),II=1,3),         UPDATE  2790
     *             (HQQ(II,K),II=1,3),SQQ(K),CFQQ(K),QN                      UPDATE  2800
   38 FORMAT('0 UPDATE ROLL-SLIDE TEST'/(2X,9G14.6))                         UPDATE  2810
      IF (QN.LT.0.0) KQTYPE(K) = -4                                          UPDATE  2820
      IF (QN.LT.0.0) GO TO 39                                                UPDATE  2830
      QDOTQ = QQ(1,K)**2 + QQ(2,K)**2 + QQ(3,K)**2                           UPDATE  2840
      QT = DSQRT(QDOTQ-QN**2)                                                UPDATE  2850
      IF (KQTYPE(K).EQ.3 .AND. QT.LE.CFQQ(K)*QN) GO TO 40                    UPDATE  2860
      IF (KQTYPE(K).EQ.4 .AND. QT.GE.0.9*CFQQ(K)*QN) GO TO 40                UPDATE  2870
      KQTYPE(K) = 7-KQTYPE(K)                                                UPDATE  2880
   39 CALL OUTPUT(0)                                                         UPDATE  2890
      CALL SETUP2                                                            UPDATE  2900
      CALL DAUX(K)                                                           UPDATE  2910
      IF  (NPRT(24).NE.0)  CALL OUTPUT(1)                                    UPDATE  2920
      IF  (NPRT( 3).NE.0)  CALL PRINT (6HUPDATE)                             UPDATE  2930
      I = -1                                                                 UPDATE  2940
   40 CONTINUE                                                               UPDATE  2950
   41 CALL ELTIME(2,7)                                                       UPDATE  2960
   42 RETURN                                                                 UPDATE  2970
      END                                                                    UPDATE  2980
```

```
      SUBROUTINE  UPDFDC  (M)                                                UPDFDC  0010
C                                                                REV 19 10/19/79   UPDFDC  0020
C     UPDATE FORCE DEFLECTION CURVE DEFINITION THAT IS DEFINED                   UPDFDC  0030
C     IN LOCATION M OF NTAB ARRAY.  SUBROUTINE ASSUMES THAT                      UPDFDC  0040
C     A SUCCESSFUL INTEGRATION STEP HAS JUST BEEN COMPLETED AND                  UPDFDC  0050
C     WILL COMPUTE ENTIRE CURVE DEFINITION TO BE VALID FOR NEXT STEP.            UPDFDC  0060
C                                                                                UPDFDC  0070
      IMPLICIT REAL*8(A-H,O-Z)                                                   UPDFDC  0080
      COMMON/TABLES/ MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(500),TAB(2600)         UPDFDC  0090
       L      = NTAB(M)                                                         UPDFDC  0100
      IF (L.EQ.0)  GO TO 99                                                      UPDFDC  0110
      D       = TAB(L)                                                         UPDFDC  0120
      IF (D.LT.0.0) D = 0.0                                                      UPDFDC  0130
      DLAST   = TAB(L+1)                                                       UPDFDC  0140
      IF (D.EQ.DLAST)  GO TO 99                                                UPDFDC  0150
      DCUBIC = TAB(L+6)                                                        UPDFDC  0160
      IF (D.EQ.DCUBIC) GO TO 98                                                UPDFDC  0170
      AREA    = TAB(L+2)                                                       UPDFDC  0180
      RLAST   = TAB(L+3)                                                       UPDFDC  0190
      GLAST   = TAB(L+4)                                                       UPDFDC  0200
      DG      = TAB(L+5)                                                       UPDFDC  0210
      DG0     = DG                                                             UPDFDC  0220
      DREF    = TAB(L+7)                                                       UPDFDC  0230
      DMAX    = TAB(L+8)                                                       UPDFDC  0240
      DINER   = TAB(L+9)                                                       UPDFDC  0250
      FDMAX   = TAB(L+10)                                                      UPDFDC  0260
      DC0     = TAB(L+18)                                                      UPDFDC  0270
      LQ      = L+11                                                           UPDFDC  0280
      LC      = L+14                                                           UPDFDC  0290
      IF (NTAB(M+1).LT.0) GO TO 98                                             UPDFDC  0300
      IF (D-DCUBIC) 10,98,20                                                   UPDFDC  0310
C                                                                              UPDFDC  0320
C     D < DCUBIC, DEFINE NEW CUBIC                                             UPDFDC  0330
C     Y(X) = A0 + A1*(X-X1) + A2*(X-X1)**2 + A3*(X-X1)**3                      UPDFDC  0340
C     WHOSE DERIVATIVE IS                                                      UPDFDC  0350
C     Y'(X) = A1 + 2*A2*(X-X1) + 3*A3+(X-X1)**2                                UPDFDC  0360
C                                                                              UPDFDC  0370
   10 X1 = DMAX1 (D,DG)                                                        UPDFDC  0380
      X2 = DREF                                                                UPDFDC  0390
C                                                                              UPDFDC  0400
C     IF INERTIAL SPIKE EXISTS AND IF DIMAX < DREF , DROP INERTIAL SPIKE       UPDFDC  0410
      NI = NTAB(M+2)                                                           UPDFDC  0420
      IF (NI.GT.0.AND.TAB(NI+3).GT.0.0.AND.DREF.GT.TAB(NI+3))NTAB(M+2)=0       UPDFDC  0430
      DX = X2-X1                                                               UPDFDC  0440
      X = X1-DG                                                                UPDFDC  0450
      Y1 = TAB(LQ) +X *(TAB(LQ+1)+X *TAB(LQ+2))                                UPDFDC  0460
      Y1P = TAB(LQ+1)+2.0*X *TAB(LQ+2)                                         UPDFDC  0470
      X2DOT = 0.0                                                              UPDFDC  0480
      CALL FRCDFL (X2,X2DOT,M,0,Y2P,ELOSS)                                     UPDFDC  0490
      CALL FRCDFL (X2,X2DOT,M,1,Y2 ,ELOSS)                                     UPDFDC  0500
```

```
      DCUBIC = X1                                                          UPDFDC  0510
      DC0    = DCUBIC                                                      UPDFDC  0520
C                                                                          UPDFDC  0530
C     A0 = Y(X1)   (THE VALUE OF THE QUADRATIC AT X1)                      UPDFDC  0540
C     A1 = Y'(X1)  (THE DERIVATIVE OF THE QUADRATIC AT X1)                 UPDFDC  0550
C                                                                          UPDFDC  0560
      A0 = Y1                                                              UPDFDC  0570
      A1 = Y1P                                                             UPDFDC  0580
C                                                                          UPDFDC  0590
C     SOLVE SIMULTANEOUSLY FOR A2 AND A3                                   UPDFDC  0600
C        A2*(X2-X1)**2 +    A3*(X2-X1)**3 = Y(X2)-A0-A1*(X2-X1)          UPDFDC  0610
C      2*A2*(X2-X1)    +  3*A3*(X2-X1)**2 = Y'(X2)-A1                     UPDFDC  0620
C                                                                          UPDFDC  0630
      R13 = (Y2 - Y1 -Y1P*DX)/DX**2                                        UPDFDC  0640
      R23 = (Y2P - Y1P)/DX                                                 UPDFDC  0650
      A2  = 3.0*R13 - R23                                                  UPDFDC  0660
      A3  = (R23 - 2.0*R13)/DX                                             UPDFDC  0670
C                                                                          UPDFDC  0680
C     IF LOCAL MINIMUN OF CUBIC (ABSCISSA VALUE WHERE Y'(X) = 0)           UPDFDC  0690
C     LIES BETWEEN DCUBIC AND DREF AND IS NEGATIVE, THEN REPLACE           UPDFDC  0700
C     CUBIC DEFINITION WITH STRAIGHT LINE BETWEEN (X1,Y1) AND (X2,Y2).     UPDFDC  0710
C                                                                          UPDFDC  0720
      IF (A3.NE.0.0)  GO TO 14                                             UPDFDC  0730
      R2 = -0.5*A1/A2                                                      UPDFDC  0740
      GO TO 15                                                             UPDFDC  0750
   14 A33 = 3.0*A3                                                         UPDFDC  0760
      DISC = A2**2-A1*A33                                                  UPDFDC  0770
      IF (DISC.LT.0.0)  GO TO 13                                           UPDFDC  0780
      SQDISC = DSQRT(DISC)                                                 UPDFDC  0790
      R1 = (-A2+SQDISC)/A33                                                UPDFDC  0800
      IF (R1.LE.0.0.OR.R1.GE.DX)  GO TO 11                                 UPDFDC  0810
      FR1 = A0+R1*(A1+R1*(A2+R1*A3))                                       UPDFDC  0820
      IF (FR1.LT.0.0) GO TO 12                                             UPDFDC  0830
   11 R2 = (-A2-SQDISC)/A33                                                UPDFDC  0840
   15 IF (R2.LE.0.0.OR.R2.GE.DX) GO TO 13                                  UPDFDC  0850
      FR2 = A0+R2*(A1+R2*(A2+R2*A3))                                       UPDFDC  0860
      IF (FR2.GE.0.0)  GO TO 13                                            UPDFDC  0870
   12 A0 = Y1                                                              UPDFDC  0880
      A1 = (Y2-Y1)/DX                                                      UPDFDC  0890
      A2 = 0.0                                                             UPDFDC  0900
      A3 = 0.0                                                             UPDFDC  0910
   13 TAB(LC) = A0                                                         UPDFDC  0920
      TAB(LC+1) = A1                                                       UPDFDC  0930
      TAB(LC+2) = A2                                                       UPDFDC  0940
      TAB(LC+3) = A3                                                       UPDFDC  0950
      TAB(L +6) = DCUBIC                                                   UPDFDC  0960
      TAB(L+18) = DC0                                                      UPDFDC  0970
      GO TO 98                                                             UPDFDC  0980
   20 IF (D-DREF) 21,21,30                                                 UPDFDC  0990
C                                                                          UPDFDC  1000
```

```
C     DCUBIC < D < DREF, DEFINE NEW QUADRATIC FROM CUBIC CURVE.                  UPDFDC  1010
C                                                                                UPDFDC  1020
   21 X = D-DC0                                                                  UPDFDC  1030
      Y2 = TAB(LC)+X*(TAB(LC+1)+X*(TAB(LC+2)+X*TAB(LC+3)))                       UPDFDC  1040
      X1 = DCUBIC - DG                                                           UPDFDC  1050
      AREA = X1*(TAB(LQ)+X1*(TAB(LQ+1)/2.0+X1*TAB(LQ+2)/3.0))                    UPDFDC  1060
     * + X*(TAB(LC)+X*(TAB(LC+1)/2.0+X*(TAB(LC+2)/3.0+X*TAB(LC+3)/4.0)))        UPDFDC  1070
      X = DCUBIC - DC0                                                           UPDFDC  1080
      IF (X.NE.0.0) AREA = AREA                                                  UPDFDC  1090
     * - X*(TAB(LC)+X*(TAB(LC+1)/2.0+X*(TAB(LC+2)/3.0+X*TAB(LC+3)/4.0)))        UPDFDC  1100
      GO TO 31                                                                   UPDFDC  1110
C                                                                                UPDFDC  1120
C     DREF < D, DEFINE NEW QUADRATIC FROM BASE CURVE.                            UPDFDC  1130
C                                                                                UPDFDC  1140
C     IF DINER < D , REMOVE INERTIAL SPIKE                                       UPDFDC  1150
C                                                                                UPDFDC  1160
   30 IF  (NTAB(M+2).GT.0 .AND. D.GE.DINER)  NTAB(M+2) = 0                       UPDFDC  1170
      NR = NTAB(M+3)                                                             UPDFDC  1180
      RLAST = 1.0                                                                UPDFDC  1190
      IF (NR.GT.0  ) RLAST = EVALFD(D,NR,1)                                      UPDFDC  1200
      IF (RLAST.NE.1.0) GO TO 39                                                 UPDFDC  1210
C                                                                                UPDFDC  1220
C     R = 1,  USE BASE CURVE FOR UNLOADING                                       UPDFDC  1230
C                                                                                UPDFDC  1240
      DG = 0.0                                                                   UPDFDC  1250
      DCUBIC = 0.0                                                               UPDFDC  1260
      DREF = 0.0                                                                 UPDFDC  1270
      A0 = 0.0                                                                   UPDFDC  1280
      A1 = 0.0                                                                   UPDFDC  1290
      A2 = 0.0                                                                   UPDFDC  1300
      GO TO 32                                                                   UPDFDC  1310
   39 NG = NTAB(M+4)                                                             UPDFDC  1320
      GLAST = 0.0                                                                UPDFDC  1330
      IF (NG.GT.0  ) GLAST = EVALFD(D,NG,1)                                      UPDFDC  1340
      NB =  NTAB(M+1)                                                            UPDFDC  1350
      D0 = TAB(NB)                                                               UPDFDC  1360
      DG = D0 + GLAST*(D-D0)                                                     UPDFDC  1370
      Y2 = EVALFD(D, NB,1)                                                       UPDFDC  1380
      NI = NTAB(M+2)                                                             UPDFDC  1390
      IF (NI.GT.0) Y2 = Y2+EVALFD(D,NI,1)                                        UPDFDC  1400
      AREA = EVALFD(D,NB,2)                                                      UPDFDC  1410
      DREF = D                                                                   UPDFDC  1420
   31 DCUBIC = D                                                                 UPDFDC  1430
      X1 = DG                                                                    UPDFDC  1440
      X2 = D                                                                     UPDFDC  1450
      DX = X2-X1                                                                 UPDFDC  1460
      Y1 = 0.0                                                                   UPDFDC  1470
      RAREA = RLAST*AREA                                                         UPDFDC  1480
C                                                                                UPDFDC  1490
C     COMPUTE UNLOADING QUADRATIC COEFFICIENTS SUCH THAT                         UPDFDC  1500
```

```
C     ENDPOINT DERIVATES ARE NON-NEGATIVE.                              UPDFDC  1510
C                                                                       UPDFDC  1520
      A0  = 0.0                                                         UPDFDC  1530
      A1  = 2.0/DX*(3.0*RAREA/DX-Y2)                                    UPDFDC  1540
      IF (A1.LT.0.0)  A1 = 0.0                                          UPDFDC  1550
      A2 = (Y2/DX-A1)/DX                                                UPDFDC  1560
      IF (A2.GE.0.0) GO TO 32                                           UPDFDC  1570
      A1 = Y2/DX                                                        UPDFDC  1580
      A2 = 0.0                                                          UPDFDC  1590
C                                                                       UPDFDC  1600
C     RESTORE TAB VALUES THAT MAY HAVE BEEN CHANGED                     UPDFDC  1610
C                                                                       UPDFDC  1620
   32 TAB(L+2) = AREA                                                   UPDFDC  1630
      TAB(L+3) = RLAST                                                  UPDFDC  1640
      TAB(L+4) = GLAST                                                  UPDFDC  1650
      TAB(L+5) = DG                                                     UPDFDC  1660
      TAB(L+6) = DCUBIC                                                 UPDFDC  1670
      TAB(L+7) = DREF                                                   UPDFDC  1680
      TAB(LQ)   = A0                                                    UPDFDC  1690
      TAB(LQ+1) = A1                                                    UPDFDC  1700
      TAB(LQ+2) = A2                                                    UPDFDC  1710
   98 TAB(L+1) = D                                                      UPDFDC  1720
      IF (D.GT.DG0 .AND. DLAST.LE.DG0) M=-M                             UPDFDC  1730
   99 RETURN                                                            UPDFDC  1740
      END                                                               UPDFDC  1750
```

```
      SUBROUTINE VEHPOS                                                 VEHPOS  0010
C                                                          REV 19 09/15/7B  VEHPOS  0020
C     COMPUTES COMPONENTS OF VEHICLE ACCELERATIONS ONLY AS A FUNCTION     VEHPOS  0030
C     OF TIME USING DATA AND TABLES PRODUCED BY SUBROUTINE VINPUT.        VEHPOS  0040
C                                                                         VEHPOS  0050
      IMPLICIT REAL*8 (A-H,O-Z)                                         VEHPOS  0060
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           VEHPOS  0070
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)        VEHPOS  0080
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30), VEHPOS  0090
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       VEHPOS  0100
      COMMON/VPOSTN/ ZPLT(3),SPLT(3),AXV(3,6),VATAB(6,101,6),           VEHPOS  0110
     *               VT0(6),VDT(6),TIMEV(6),OMEGV(6),NVTAB(6),INDXV(6)  VEHPOS  0120
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         VEHPOS  0130
     *               UNITL,UNITM,UNITT,GRAVTY(3)                        VEHPOS  0140
      DIMENSION AX(3)                                                   VEHPOS  0150
      T = TIME                                                          VEHPOS  0160
      M = 1                                                             VEHPOS  0170
   15 DO  16  I=1,3                                                     VEHPOS  0180
   16 AX(I) = AXV(I,M)                                                  VEHPOS  0190
      AT0 = VT0(M)                                                      VEHPOS  0200
      ADT = VDT(M)                                                      VEHPOS  0210
      VTIME = TIMEV(M)                                                  VEHPOS  0220
      OMEG  = OMEGV(M)                                                  VEHPOS  0230
      NATAB = NVTAB(M)                                                  VEHPOS  0240
      K = INDXV(M)                                                      VEHPOS  0250
      IF(NATAB.NE.0) GO TO 20                                           VEHPOS  0260
C                                                                       VEHPOS  0270
C     HALF-SINE WAVE DECELERATION                                       VEHPOS  0280
C                                                                       VEHPOS  0290
      IF(T.GT.VTIME) T=VTIME                                            VEHPOS  0300
      WT = OMEG*T                                                       VEHPOS  0310
      SWT = DSIN(WT)                                                    VEHPOS  0320
      DO 10 I=1,3                                                       VEHPOS  0330
      AW = AX(I)*OMEG                                                   VEHPOS  0340
      SEGLA(I,K) = -AW*OMEG*SWT                                         VEHPOS  0350
   10 WMEGD(I,K) = 0.0                                                  VEHPOS  0360
      GO TO 99                                                          VEHPOS  0370
   20 IF (NATAB.LT.0) GO TO 30                                          VEHPOS  0380
C                                                                       VEHPOS  0390
C      UNIDIRECTIONAL DECELERATION                                      VEHPOS  0400
C                                                                       VEHPOS  0410
      IF (T.LT.VTIME) GO TO 21                                          VEHPOS  0420
C                                                                       VEHPOS  0430
C     TIME POINT EXCEEDS TABLE, USE LAST VALUES OF ACCELERATION.        VEHPOS  0440
C                                                                       VEHPOS  0450
      AC0 = VATAB(1,NATAB,M)                                            VEHPOS  0460
      GO TO 25                                                          VEHPOS  0470
C                                                                       VEHPOS  0480
C         USE QUADRATIC INTERPOLATION FROM TABLES FOR CURRENT VALUE OF  VEHPOS  0490
C         TIME TO BE CONSISTENT WITH SIMPSON INTEGRATION OF TABLES.     VEHPOS  0500
```

```
C                                                                       VEHPOS  0510
   21 J= 0.5*(T-AT0)/ADT +1.0                                           VEHPOS  0520
      XK = T/ADT -DFLOAT(2*J-1)                                         VEHPOS  0530
      X1 = XK+1.0                                                       VEHPOS  0540
      X3 = XK-1.0                                                       VEHPOS  0550
      AC0 = 0.5*XK*X3*VATAB(1,2*J-1,M)                                  VEHPOS  0560
     *    -    X3*X1*VATAB(1,2*J  ,M)                                  VEHPOS  0570
     *    + 0.5*XK*X1*VATAB(1,2*J+1,M)                                  VEHPOS  0580
C                                                                       VEHPOS  0590
C        COMPONENTS OF VEHICLE ACCELERATION.                            VEHPOS  0600
C                                                                       VEHPOS  0610
   25 DO 29 I=1,3                                                       VEHPOS  0620
      SEGLA(I,K) = -G*AX(I)*AC0                                         VEHPOS  0630
   29 WMEGD(I,K) = 0.0                                                  VEHPOS  0640
      GO TO 99                                                          VEHPOS  0650
C                                                                       VEHPOS  0660
C     OMNIDIRECTIONAL DECELERATION                                      VEHPOS  0670
C                                                                       VEHPOS  0680
   30 J = (TIME-AT0)/ADT + 1.0                                          VEHPOS  0690
      IF (J.GE.-NATAB) GO TO 32                                         VEHPOS  0700
C                                                                       VEHPOS  0710
C        INTERPOLATION FROM VINPUT TABLES OF COMPONENTS OF VEHICLE      VEHPOS  0720
C        LINEAR AND ANGULAR ACCELERATION.                               VEHPOS  0730
C                                                                       VEHPOS  0740
      TJ       = AT0 + DFLOAT(J-1)*ADT                                  VEHPOS  0750
      DLT      = TIME-TJ                                                VEHPOS  0760
      R1 = DLT/ADT                                                      VEHPOS  0770
      R2 = 1.0-R1                                                       VEHPOS  0780
      DO 31 I=1,3                                                       VEHPOS  0790
      SEGLA(I,K) =     -G*(VATAB(I  ,J+1,M)*R1 + VATAB(I  ,J,M)*R2)    VEHPOS  0800
   31 WMEGD(I,K) = RADIAN*(VATAB(I+3,J+1,M)*R1 + VATAB(I+3,J,M)*R2)     VEHPOS  0810
      GO TO 99                                                          VEHPOS  0820
C                                                                       VEHPOS  0830
C     TIME POINT EXCEEDS TABLE, USE LAST VALUES OF ACCELERATION.        VEHPOS  0840
C                                                                       VEHPOS  0850
   32 J = - NATAB                                                       VEHPOS  0860
      DO 33 I=1,3                                                       VEHPOS  0870
      SEGLA(I,K) =     -G*VATAB(I  ,J,M)                                VEHPOS  0880
   33 WMEGD(I,K) = RADIAN*VATAB(I+3,J,M)                                VEHPOS  0890
   99 M = M+1                                                           VEHPOS  0900
      IF  (M.LE.6 .AND. INDXV(M).NE.0)  GO TO 15                        VEHPOS  0910
      RETURN                                                            VEHPOS  0920
      END                                                               VEHPOS  0930
```

```
      SUBROUTINE VINPUT                                                 VINPUT  0010
C                                                          REV 20 05/07/80  VINPUT  0020
C     PERFORMS CARD INPUT AND COMPUTES DATA AND TABLES REQUIRED BY      VINPUT  0030
C     SUBROUTINE VEHPOS TO INTEGRATE THE CRASH VEHICLE MOTION FOR ONE OF VINPUT  0040
C     THREE PERMISSABLE OPTIONS:                                        VINPUT  0050
C       (1) HALF SINE-WAVE LINEAR DECELERATION IMPULSE                  VINPUT  0060
C       (2) UNIDIRECTIONAL LINEAR DECELERATION TABULAR INPUT            VINPUT  0070
C       (3) OMNIDIRECTIONAL LINEAR AND ANGULAR ACCELERATION TABULAR     VINPUT  0080
C           INPUT (6 DEGREES OF FREEDOM VEHICLE MOTION)                 VINPUT  0090
C                                                                       VINPUT  0100
      IMPLICIT REAL*8 (A-H,O-Z)                                         VINPUT  0110
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           VINPUT  0120
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)        VINPUT  0130
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30), VINPUT  0140
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       VINPUT  0150
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(3,60),HA(3,60),HB(3,60), VINPUT  0160
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),     VINPUT  0170
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)    VINPUT  0180
      COMMON/VPOSTN/ ZPLT(3),SPLT(3),AXV(3,6),VATAB(6,101,6),           VINPUT  0190
     *               VT0(6),VDT(6),TIMEV(6),OMEGV(6),NVTAB(6),INDXV(6)  VINPUT  0200
      COMMON/TEMPVS/ X0(3),XDOT0(3),XCOMP(3),XVCOMP(3),ANGLE(3),        VINPUT  0210
     *               ATAB(15,100),DVEH(3,3),VMEG(3),VMEGD(3),           VINPUT  0220
     *              XACOMP(3),THET(3),AX(3),F(5,100),XYZ(6,102),TT(102), VINPUT  0230
     *               VIPS,VMPH,AT0,ADT,VTIME,OMEG,NATAB                 VINPUT  0240
      COMMON/INTEST/ SGTEST(3,4,30),XTEST(3,120),SEGT(120),REGT(120)    VINPUT  0250
      REAL    SEGT                                                      VINPUT  0260
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         VINPUT  0270
     *               UNITL,UNITM,UNITT,GRAVTY(3)                        VINPUT  0280
      COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),           VINPUT  0290
     *               BLTTTL(5,8),PLTTL(5,30),BAGTTL(5,6),SEG(30),       VINPUT  0300
     *               JOINT(30),CGS(30),JS(30)                           VINPUT  0310
      REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT      VINPUT  0320
      LOGICAL*1 CGS,JS                                                  VINPUT  0330
      DIMENSION IDYPR(3)                                                VINPUT  0340
      REAL VEH(6),GRND                                                  VINPUT  0350
      DATA VEH/4HVEH1,4HVEH2,4HVEH3,4HVEH4,4HVEH5,4HVEH /,GRND/4HGRND/  VINPUT  0360
      DATA IDYPR/3,2,1/                                                 VINPUT  0370
C                                                                       VINPUT  0380
C     READ AND PRINT CONTENTS OF CARDS C.1 AND C.2                      VINPUT  0390
C                                                                       VINPUT  0400
      NVEH = NSEG                                                       VINPUT  0410
      NVH  = 0                                                          VINPUT  0420
      DO   11  I=1,6                                                    VINPUT  0430
   11 INDXV(I) = 0                                                      VINPUT  0440
   12 READ (5,13) VPSTTL                                                VINPUT  0450
   13 FORMAT (20A4)                                                     VINPUT  0460
      READ(5,14) ANGLE,VIPS,VTIME,X0,NATAB,AT0,ADT,MSEG                 VINPUT  0470
   14 FORMAT(8F6.0,I6,2F6.0,I6)                                         VINPUT  0480
      WRITE (6,15) VPSTTL,ANGLE,VIPS,VTIME,X0,NATAB,AT0,ADT,MSEG        VINPUT  0490
   15 FORMAT('1 VEHICLE DECELERATION INPUTS',91X,'CARDS C'//3X,20A4//)   VINPUT  0500
```

```
     *    7X,'YAW',9X,'PITCH',7X,'ROLL',8X,'VIPS',8X,'VTIME',7X,'XO(X)',    VINPUT  0510
     *    7X,'XO(Y)',7X,'XO(Z)',2X,'NATAB',6X,'ATO',9X,'ADT',4X,'MSEG'/     VINPUT  0520
     *    8F12.3,I5,2X,2F12.6,I5)                                           VINPUT  0530
      DA1 = ANGLE(1)*RADIAN                                                 VINPUT  0540
      DA2 = ANGLE(2)*RADIAN                                                 VINPUT  0550
      AX(3) = DCOS(DA2)                                                     VINPUT  0560
      AX(1) = DCOS(DA1)*AX(3)                                               VINPUT  0570
      AX(2) = DSIN(DA1)*AX(3)                                               VINPUT  0580
      AX(3) = DSIN(DA2)                                                     VINPUT  0590
      IF(NATAB.NE.0) GO TO 18                                               VINPUT  0600
C                                                                           VINPUT  0610
C     HALF-SINE WAVE DECELERATION                                           VINPUT  0620
C                                                                           VINPUT  0630
      OMEG = PI/VTIME                                                       VINPUT  0640
      AT = 0.5*VIPS/OMEG                                                    VINPUT  0650
      IF (VIPS.LT.0.0) VIPS = 0.0                                           VINPUT  0660
      DO 16 I=1,3                                                           VINPUT  0670
      XACOMP(I) = 0.0                                                       VINPUT  0680
      XDOTO(I) = VIPS*AX(I)                                                 VINPUT  0690
   16 AX(I) = AT*AX(I)                                                      VINPUT  0700
      WRITE (6,17) VIPS,UNITL,UNITT,ANGLE,VTIME,UNITT                       VINPUT  0710
   17 FORMAT('0 PASSENGER COMPARTMENT DISPLACEMENT HISTORY'/                 VINPUT  0720
     *       '  ANALYTICAL HALF-SINE WAVE DECELERATION'/                    VINPUT  0730
     *       '  VO=',F8.3,1X,A4,'/',A4,', OBLIQUE ANGLES =',3F7.2,          VINPUT  0740
     *       ' DEGREES, TIME DURATION =',F7.3,1X,A4//)                      VINPUT  0750
      GO TO 28                                                              VINPUT  0760
   18 IF (NATAB.LT.0) GO TO 31                                              VINPUT  0770
C                                                                           VINPUT  0780
C     FOR UNIDIRECTIONAL VEHICLE MOTION                                     VINPUT  0790
C     READ LINEAR DECELERATION TABLES FROM CARDS C.3                        VINPUT  0800
C                                                                           VINPUT  0810
      READ (5,19) (ATAB(1,I),I=1,NATAB)                                     VINPUT  0820
   19 FORMAT (12F6.0)                                                       VINPUT  0830
C                                                                           VINPUT  0840
C     EXTEND TABLE IF NECESSARY SUCH THAT NATAB IS ODD AND                  VINPUT  0850
C     LAST ENTRY NEED NOT BE ZERO. IF TABLE SIZE IS EXCEEDED ON TIME,       VINPUT  0860
C     VALUE OF LAST ENTRY WILL BE USED.                                     VINPUT  0870
C                                                                           VINPUT  0880
      IF (MOD(NATAB,2).EQ.1) GO TO 20                                       VINPUT  0890
      ATAB(1,NATAB+1) = ATAB(1,NATAB)                                       VINPUT  0900
      NATAB = NATAB+1                                                       VINPUT  0910
   20 VTIME = ADT * DFLOAT(NATAB-1)                                         VINPUT  0920
C                                                                           VINPUT  0930
C     USING SIMPSON'S INTEGRATION, COMPUTE VELOCITY AND DISPLACEMENT        VINPUT  0940
C     TABLE FOR NATAB EQUALLY SPACED (ADT) TIME POINTS.                     VINPUT  0950
C     FOR I=1,NATAB                                                         VINPUT  0960
C       ATAB(1,I) = LINEAR DECELERATION (G'S)                               VINPUT  0970
C       ATAB(2,I) = LINEAR VELOCITY (L UNITS/T UNITS)                       VINPUT  0980
C       ATAB(3,I) = LINEAR DISPLACEMENT (L UNITS)                           VINPUT  0990
C                                                                           VINPUT  1000
```

```
      ATAB(2,1) = VIPS                                                          VINPUT  1010
      ATAB(3,1) = 0.0                                                           VINPUT  1020
      DA1 = ADT/3.0                                                             VINPUT  1030
      DA2 = ADT/12.0                                                            VINPUT  1040
      UNITS = -G                                                                VINPUT  1050
      DO 22 J=2,3                                                               VINPUT  1060
      DO 21 I=2,NATAB,2                                                         VINPUT  1070
      F1 = ATAB(J-1,I-1) * UNITS                                                VINPUT  1080
      F2 = ATAB(J-1,I  ) * UNITS                                                VINPUT  1090
      F3 = ATAB(J-1,I+1) * UNITS                                                VINPUT  1100
      ATAB(J,I  ) = ATAB(J,I-1) + DA2*(5.0*F1+8.0*F2-F3)                        VINPUT  1110
   21 ATAB(J,I+1) = ATAB(J,I-1) + DA1*(    F1+4.0*F2+F3)                        VINPUT  1120
   22 UNITS = 1.0                                                               VINPUT  1130
C                                                                               VINPUT  1140
C     PRINT TABLES                                                              VINPUT  1150
C                                                                               VINPUT  1160
      WRITE (6,23) (UNITL,UNITT,UNITL,I=1,2)                                    VINPUT  1170
   23 FORMAT('0 UNIDIRECTIONAL VEHICLE POSITION TABLES'//                       VINPUT  1180
     *     2('      TIME       ACC       VELOCITY     POSITION    ')/          VINPUT  1190
     *   2('     (MSEC)      (G)    (',A4,'/',A4,')',5X,'(',A4,')',4X)/)        VINPUT  1200
      DO 26 J=1,50                                                              VINPUT  1210
      IF (J.GT.NATAB) GO TO 26                                                  VINPUT  1220
      T1 = (AT0 + DFLOAT(J-1)*ADT)*1000.0                                       VINPUT  1230
      IF (J+50.LE.NATAB) GO TO 25                                               VINPUT  1240
      WRITE (6,24) T1,(ATAB(I,J),I=1,3)                                         VINPUT  1250
   24 FORMAT(2(F11.5,F10.2,F13.4,F13.5,3X))                                     VINPUT  1260
      GO TO 26                                                                  VINPUT  1270
   25 T2 = (AT0 + DFLOAT(J+49)*ADT)*1000.0                                      VINPUT  1280
      WRITE (6,24) T1,(ATAB(I,J),I=1,3),T2,(ATAB(I,J+50),I=1,3)                 VINPUT  1290
   26 CONTINUE                                                                  VINPUT  1300
C                                                                               VINPUT  1310
C     INITIALIZATION                                                            VINPUT  1320
C                                                                               VINPUT  1330
      DO 27 I=1,3                                                               VINPUT  1340
      XACOMP(I) = -G*AX(I)*ATAB(1,1)                                            VINPUT  1350
   27 XDOT0(I)= VIPS*AX(I)                                                      VINPUT  1360
   28 DO 30 I=1,3                                                               VINPUT  1370
      DO 29 J=1,3                                                               VINPUT  1380
   29 DVEH(I,J) = 0.0                                                           VINPUT  1390
      DVEH(I,I) = 1.0                                                           VINPUT  1400
      VMEGD(I) = 0.0                                                            VINPUT  1410
   30 VMEG(I)   = 0.0                                                           VINPUT  1420
      GO TO 64                                                                  VINPUT  1430
C                                                                               VINPUT  1440
C     FOR OMNIDIRECTIONAL (6 DEGREES OF FREEDOM) VEHICLE MOTION                 VINPUT  1450
C     READ LINEAR DECELERATION AND ANGULAR ACCELERATION TABLES                  VINPUT  1460
C     FROM CARDS C.4.                                                           VINPUT  1470
C                                                                               VINPUT  1480
   31 MATAB = -NATAB                                                            VINPUT  1490
      READ  (5,32)  LTYPE,LFIT,NPTS,(VMEG(I),I=1,3)                             VINPUT  1500
```

```
32 FORMAT (3I6,22X,3F10.0)                                                    VINPUT  1510
   IF  (LTYPE.GT.0)  GO TO 34                                                 VINPUT  1520
   READ (5,33) ((ATAB(I,J),I=1,3),(ATAB(I,J),I=10,12),J=1,MATAB)              VINPUT  1530
33 FORMAT (10X,6F10.0)                                                        VINPUT  1540
   ISKIP = 0                                                                  VINPUT  1550
   GO TO 46                                                                   VINPUT  1560
34 LPTS = LTYPE-1 + NPTS                                                      VINPUT  1570
   READ  (5,35)  (TT(I),(XYZ(I,J),J=1,6),I=1,LPTS)                            VINPUT  1580
35 FORMAT (7F10.0)                                                            VINPUT  1590
   WRITE  (6,36)  LTYPE,LFIT,NPTS,                                            VINPUT  1600
  *               (TT(I),(XYZ(I,J),J=1,6),I=1,LPTS)                           VINPUT  1610
36 FORMAT ('0 SPLINE FIT TABULAR INPUT'//                                     VINPUT  1620
  *        3X,'LTYPE =',I6,'  LFIT =',I6,'  NPTS =',I6//                      VINPUT  1630
  *        (F15.6,3X,3F12.3,3X,3F12.3))                                       VINPUT  1640
   DO  37  I=1,3                                                              VINPUT  1650
   I4 = 4-I                                                                   VINPUT  1660
   X0(I) = XYZ(1,I)                                                           VINPUT  1670
   IF  (LTYPE.EQ.1)  GO TO 37                                                 VINPUT  1680
   XDOT0(I) = XYZ(2,I)                                                        VINPUT  1690
   VMEG(I) = XYZ(2,I+3)                                                       VINPUT  1700
37 ANGLE(I4) = XYZ(1,I+3)                                                     VINPUT  1710
   DO  45  II=1,6                                                             VINPUT  1720
   CALL SPLINE (TT(LTYPE),XYZ(LTYPE,II),F,NPTS,LFIT)                          VINPUT  1730
   I = II                                                                     VINPUT  1740
   IF  (II.GT.3)  I = II + 6                                                  VINPUT  1750
   IF  (LTYPE.NE.1)  GO TO 38                                                 VINPUT  1760
   IF  (II.LE.3)  XDOT0(I) = F(3,1)                                           VINPUT  1770
   IF  (II.GT.3)  VMEG(II-3) = F(3,1)                                         VINPUT  1780
   IF  (II.GT.3)  I = 16-II                                                   VINPUT  1790
38 UNITS = 1.0                                                                VINPUT  1800
   IF  (LTYPE.LT.3 .AND. II.LE.3)  UNITS = -1.0/G                             VINPUT  1810
   K1 = 1                                                                     VINPUT  1820
   DO  45  J=1,MATAB                                                          VINPUT  1830
   TTT = AT0 + DFLOAT(J-1)*ADT                                                VINPUT  1840
   DO  39  K=K1,NPTS                                                          VINPUT  1850
   IF  (K.EQ.NPTS)  GO TO 40                                                  VINPUT  1860
   IF  (DABS(TTT-F(1,K+1)).LT.EPS(8))  TTT = F(1,K+1)                         VINPUT  1870
   IF  (TTT.LT.F(1,K+1))  GO TO 40                                            VINPUT  1880
39 CONTINUE                                                                   VINPUT  1890
40 K1 = K                                                                     VINPUT  1900
   DX = TTT - F(1,K)                                                          VINPUT  1910
   GO TO (41,42,43),LTYPE                                                     VINPUT  1920
41 ACC = 2.0*F(4,K) + 6.0*DX*F(5,K)                                           VINPUT  1930
   GO TO 44                                                                   VINPUT  1940
42 ACC = F(3,K) + DX*(2.0*F(4,K)+3.0*DX*F(5,K))                               VINPUT  1950
   GO TO 44                                                                   VINPUT  1960
43 ACC = F(2,K) + DX*(F(3,K)+DX*(F(4,K)+DX*F(5,K)))                           VINPUT  1970
44 ATAB(I,J) = ACC*UNITS                                                      VINPUT  1980
45 CONTINUE                                                                   VINPUT  1990
   ISKIP = 1                                                                  VINPUT  2000
```

```
   46 DO  55  J=1,MATAB                                                    VINPUT  2010
      IF (MOD(J,45).NE.1) GO TO 49                                         VINPUT  2020
C                                                                          VINPUT  2030
C     PRINT PAGE HEADING AT START OF EACH 45 TIME POINTS.                  VINPUT  2040
C                                                                          VINPUT  2050
      IPAGE = (J-1)/45 + 1                                                 VINPUT  2060
      WRITE (6,48) ISKIP,VPSTTL,IPAGE,UNITL,UNITT,UNITL                    VINPUT  2070
   48 FORMAT(I1,'  VEHICLE LINEAR TIME HISTORY',3X,20A4,3X,                VINPUT  2080
     *            'PAGE NO.',I3//                                         VINPUT  2090
     *       4X,'TIME',12X,'LINEAR DECELERATIONS (G''S)',                 VINPUT  2100
     *                 11X,'LINEAR VELOCITIES (',A4,'/',A4,')',           VINPUT  2110
     *                 11X,'LINEAR DISPLACEMENTS (',A4,')'     /          VINPUT  2120
     *        3X,'(MSEC)',3(11X,'X',11X,'Y',11X,'Z',3X)     /  )           VINPUT  2130
      ISKIP = 1                                                            VINPUT  2140
   49 IF (J.GT.1) GO TO 52                                                 VINPUT  2150
C                                                                          VINPUT  2160
C     INTEGRATION INITIALIZATION FOR TIME = 0.                             VINPUT  2170
C                                                                          VINPUT  2180
      DO 50 I=1,3                                                          VINPUT  2190
      ATAB(I+6,J) = X0(I)                                                  VINPUT  2200
      ATAB(I+12,J) = VMEG(I)                                               VINPUT  2210
   50 THET(I) = ANGLE(I)*RADIAN                                            VINPUT  2220
      CALL DRCYPR (DVEH,ANGLE,IDYPR)                                       VINPUT  2230
      DO 51 I=1,3                                                          VINPUT  2240
      IF (LTYPE.EQ.0)  XDOT0(I) = VIPS*DVEH(1,I)                           VINPUT  2250
   51 ATAB(I+3,J) = XDOT0(I)                                               VINPUT  2260
      GO TO 54                                                             VINPUT  2270
   52 DO 53 I=1,3                                                          VINPUT  2280
                                                                           VINPUT  2290
C     INTEGRATE LINEAR VELOCITY AND DISPLACEMENT.                          VINPUT  2300
C                                                                          VINPUT  2310
      ATAB(I+3,J) = ATAB(I+3,J-1)-G*ADT/2.0*(ATAB(I,J-1)+ATAB(I,J))        VINPUT  2320
   53 ATAB(I+6,J) = ATAB(I+6,J-1)                                          VINPUT  2330
     *    +ADT*(ATAB(I+3,J-1)-G*ADT/6.0*(2.0*ATAB(I,J-1)+ATAB(I,J)))      VINPUT  2340
   54 T1 = (AT0 + DFLOAT(J-1)*ADT)*1000.0                                  VINPUT  2350
   55 WRITE(6,56) T1,(ATAB(I,J),I=1,9)                                     VINPUT  2360
   56 FORMAT(F9.3,3(3X,3F12.3))                                            VINPUT  2370
      DO 61 J=1,MATAB                                                      VINPUT  2380
      IF (MOD(J,45).NE.1) GO TO 58                                         VINPUT  2390
C                                                                          VINPUT  2400
C     PRINT PAGE HEADING AT START OF EACH 45 TIME POINTS.                  VINPUT  2410
C                                                                          VINPUT  2420
      IPAGE = (J-1)/45 + 1                                                 VINPUT  2430
      WRITE (6,57) VPSTTL,IPAGE,UNITT,UNITT                                VINPUT  2440
   57 FORMAT('1 VEHICLE ANGULAR TIME HISTORY',3X,20A4,3X,'PAGE NO.',I3//   VINPUT  2450
     *        4X,'TIME', 7X,'ANGULAR ACCELERATIONS (DEG/',A4,'**2)',       VINPUT  2460
     *                   7X,'ANGULAR VELOCITIES (DEG/',A4,')',             VINPUT  2470
     *                  11X,'ANGULAR DISPLACEMENTS (DEG)' /                VINPUT  2480
     *       3X,'(MSEC)',2(11X,'X',11X,'Y',11X,'Z',3X),                    VINPUT  2490
     *       10X,'YAW',8X,'PITCH',8X,'ROLL' /)                             VINPUT  2500
```

```
   58 IF(J.EQ.1) GO TO 60                                                    VINPUT  2510
C                                                                            VINPUT  2520
C     INTEGRATE ANGULAR VELOCITY AND DISPLACEMENT.                           VINPUT  2530
C                                                                            VINPUT  2540
      DO 59 I=1,3                                                            VINPUT  2550
      ATAB(I+12,J) = ATAB(I+12,J-1)+(ATAB(I+9,J-1)+ATAB(I+9,J))*ADT/2.0      VINPUT  2560
   59 THET(I) = ADT*(ATAB(I+12,J-1)+(2.0*ATAB(I+9,J-1)+ATAB(I+9,J))*ADT      VINPUT  2570
     */6.0)*RADIAN                                                           VINPUT  2580
      CALL DSETD(DVEH,THET,THT)                                              VINPUT  2590
   60 CALL YPRDEG(DVEH,THET)                                                 VINPUT  2600
      T1 = (AT0 + DFLOAT(J-1)*ADT)*1000.0                                    VINPUT  2610
   61 WRITE (6,56)  T1,(ATAB(I,J),I=10,15),THET                              VINPUT  2620
C                                                                            VINPUT  2630
C     PROGRAM INITIALIZATION FOR TIME = 0.                                   VINPUT  2640
C                                                                            VINPUT  2650
      CALL DRCYPR (DVEH,ANGLE,IDYPR)                                         VINPUT  2660
      DO 63 I=1,3                                                            VINPUT  2670
      XACOMP(I) = -G*ATAB(I,1)                                               VINPUT  2680
      VMEG(I)  = ATAB(I+12,1)*RADIAN                                         VINPUT  2690
   63 VMEGD(I) = ATAB(I+9 ,1)*RADIAN                                         VINPUT  2700
   64 J = MSEG                                                               VINPUT  2710
      IF  (MSEG.EQ.0)  GO TO 65                                              VINPUT  2720
      IF  (MSEG.LE.NSEG)  GO TO 66                                           VINPUT  2730
      IF  (MSEG.NE.NVEH+1)  STOP 6                                           VINPUT  2740
   65 NVEH = NVEH+1                                                          VINPUT  2750
      J = NVEH                                                               VINPUT  2760
C                                                                            VINPUT  2770
C     SETUP FOR ALL PRESCRIBED SEGMENT MOTION.                               VINPUT  2780
C                                                                            VINPUT  2790
   66 NVH = NVH+1                                                            VINPUT  2800
      ISING(J) = -1                                                          VINPUT  2810
      IF  (MSEG.GT.NSEG)  SEG(J) = VEH(NVH)                                  VINPUT  2820
      RW(J) = 0.0                                                            VINPUT  2830
      DO  67  I=1,3                                                          VINPUT  2840
      RPHI (I,J) = 0.0                                                       VINPUT  2850
      SEGLA(I,J) = VMEGD(I)                                                  VINPUT  2860
      WMEGD(I,J) = XACOMP(I)                                                 VINPUT  2870
   67 AXV(I,NVH) = AX(I)                                                     VINPUT  2880
      VT0(NVH) = AT0                                                         VINPUT  2890
      VDT(NVH) = ADT                                                         VINPUT  2900
      OMEGV(NVH) = OMEG                                                      VINPUT  2910
      TIMEV(NVH) = VTIME                                                     VINPUT  2920
      NVTAB(NVH) = NATAB                                                     VINPUT  2930
      INDXV(NVH) = J                                                         VINPUT  2940
      NJ = IABS(NATAB)                                                       VINPUT  2950
      IF  (NJ.LE.0)  GO TO 69                                                VINPUT  2960
      DO  68  K=1,NJ                                                         VINPUT  2970
      DO  68  I=1,3                                                          VINPUT  2980
      VATAB(I  ,K,NVH) = ATAB(I,K)                                           VINPUT  2990
   68 VATAB(I+3,K,NVH) = ATAB(I+9,K)                                         VINPUT  3000
   69 IF  (J.LE.NSEG)  GO TO 72                                              VINPUT  3010
C                                                                            VINPUT  3020
C     SETUP FOR NEW VEHICLE (SEGMENT) MOTION.                                VINPUT  3030
                                                                             VINPUT  3040
      W(J)  = 0.0                                                            VINPUT  3050
      WD(J) = 0.0                                                            VINPUT  3060
      DO  71  I=1,3                                                          VINPUT  3070
```

```
      D(I,K,J) = DVEH(I,K)                                                  VINPUT  3090
   70 SGTEST(I,K,J) = 0.0                                                   VINPUT  3100
      SGTEST(I,4,J) = 0.0                                                   VINPUT  3110
      SEGLP(I,J) = X0(I)                                                    VINPUT  3120
      SEGLV(I,J) = XDOT0(I)                                                 VINPUT  3130
      WMEG (I,J) = VMEG(I)                                                  VINPUT  3140
      PHI  (I,J) = 0.0                                                      VINPUT  3150
   71 RPHI (I,J) = 0.0                                                      VINPUT  3160
   72 IF  (MSEG.NE.0)  GO TO 12                                             VINPUT  3170
      SEG(NVEH) = VEH(6)                                                    VINPUT  3180
C                                                                           VINPUT  3190
C     SET UP SEGMENT DATA FOR GROUND                                        VINPUT  3200
C                                                                           VINPUT  3210
      NGRND = NVEH+1                                                        VINPUT  3220
      IF  (NGRND.GT.30 .OR. NVH.GT.6)  STOP 7                               VINPUT  3230
      SEG(NGRND) = GRND                                                     VINPUT  3240
      J = NGRND                                                             VINPUT  3250
      ISING(J) = -1                                                         VINPUT  3260
      W(J)  = 0.0                                                           VINPUT  3270
      RW(J) = 0.0                                                           VINPUT  3280
      DO  74  I=1,3                                                         VINPUT  3290
      DO  73  K=1,3                                                         VINPUT  3300
      D(I,K,J) = 0.0                                                        VINPUT  3310
   73 SGTEST(I,K,J) = 0.0                                                   VINPUT  3320
      D(I,I,J) = 1.0                                                        VINPUT  3330
      SGTEST(I,4,J) = 0.0                                                   VINPUT  3340
      SEGLP(I,J) = 0.0                                                      VINPUT  3350
      SEGLV(I,J) = 0.0                                                      VINPUT  3360
      SEGLA(I,J) = 0.0                                                      VINPUT  3370
      WMEG (I,J) = 0.0                                                      VINPUT  3380
      WMEGD(I,J) = 0.0                                                      VINPUT  3390
      PHI  (I,J) = 0.0                                                      VINPUT  3400
   74 RPHI (I,J) = 0.0
```

```
      DOUBLE PRECISION FUNCTION VISCOS(ZD,VISC,HA)                      VISCOS  0010
C                                                    REV 19 10/23/78    VISCOS  0020
C     COMPUTES SUM OF COULOMB AND VISCOUS TORQUES                       VISCOS  0030
C     AT JOINTS AS A FUNCTION OF THETA DOT.                             VISCOS  0040
C     ACTUALLY ROUTINE RETURNS SUM/ZD.                                  VISCOS  0050
C                                                                       VISCOS  0060
C         ARGUMENTS:                                                    VISCOS  0070
C          ZD   : ITHETA DOTI WHERE THETA IS THE ANGLE OF THE JOINT.    VISCOS  0080
C          VISC: ARRAY OF 5 VALUES DESCRIBING FUNCTION EVALUATION.      VISCOS  0090
C                                                                       VISCOS  0100
```

```
   46 DO  55  J=1,MATAB                                                    VINPUT  2010
      IF (MOD(J,45).NE.1) GO TO 49                                         VINPUT  2020
C                                                                          VINPUT  2030
C     PRINT PAGE HEADING AT START OF EACH 45 TIME POINTS.                  VINPUT  2040
C                                                                          VINPUT  2050
      IPAGE = (J-1)/45 + 1                                                 VINPUT  2060
      WRITE (6,48) ISKIP,VPSTTL,IPAGE,UNITL,UNITT,UNITL                    VINPUT  2070
   48 FORMAT(I1,'  VEHICLE LINEAR TIME HISTORY',3X,20A4,3X,                VINPUT  2080
     *           'PAGE NO.',I3//                                           VINPUT  2090
     *       4X,'TIME',12X,'LINEAR DECELERATIONS (G''S)',                  VINPUT  2100
     *                 11X,'LINEAR VELOCITIES (',A4,'/',A4,')',            VINPUT  2110
     *                 11X,'LINEAR DISPLACEMENTS (',A4,')'    /            VINPUT  2120
     *        3X,'(MSEC)',3(11X,'X',11X,'Y',11X,'Z',3X)     /  )           VINPUT  2130
      ISKIP = 1                                                            VINPUT  2140
   49 IF (J.GT.1) GO TO 52                                                 VINPUT  2150
C                                                                          VINPUT  2160
C     INTEGRATION INITIALIZATION FOR TIME = 0.                             VINPUT  2170
C                                                                          VINPUT  2180
      DO 50 I=1,3                                                          VINPUT  2190
      ATAB(I+6,J) = X0(I)                                                  VINPUT  2200
      ATAB(I+12,J) = VMEG(I)                                               VINPUT  2210
   50 THET(I) = ANGLE(I)*RADIAN                                            VINPUT  2220
      CALL DRCYPR (DVEH,ANGLE,IDYPR)                                       VINPUT  2230
      DO 51 I=1,3                                                          VINPUT  2240
      IF (LTYPE.EQ.0)  XDOT0(I) = VIPS*DVEH(1,I)                           VINPUT  2250
   51 ATAB(I+3,J) = XDOT0(I)                                               VINPUT  2260
      GO TO 54                                                             VINPUT  2270
   52 DO 53 I=1,3                                                          VINPUT  2280
C                                                                          VINPUT  2290
C     INTEGRATE LINEAR VELOCITY AND DISPLACEMENT.                          VINPUT  2300
C                                                                          VINPUT  2310
      ATAB(I+3,J) = ATAB(I+3,J-1)-G*ADT/2.0*(ATAB(I,J-1)+ATAB(I,J))        VINPUT  2320
   53 ATAB(I+6,J) = ATAB(I+6,J-1)                                          VINPUT  2330
     *    +ADT*(ATAB(I+3,J-1)-G*ADT/6.0*(2.0*ATAB(I,J-1)+ATAB(I,J)))      VINPUT  2340
   54 T1 = (AT0 + DFLOAT(J-1)*ADT)*1000.0                                  VINPUT  2350
   55 WRITE(6,56) T1,(ATAB(I,J),I=1,9)                                     VINPUT  2360
   56 FORMAT(F9.3,3(3X,3F12.3))                                            VINPUT  2370
      DO 61 J=1,MATAB                                                      VINPUT  2380
      IF (MOD(J,45).NE.1) GO TO 58                                         VINPUT  2390
C                                                                          VINPUT  2400
C     PRINT PAGE HEADING AT START OF EACH 45 TIME POINTS.                  VINPUT  2410
C                                                                          VINPUT  2420
      IPAGE = (J-1)/45 + 1                                                 VINPUT  2430
      WRITE (6,57) VPSTTL,IPAGE,UNITT,UNITT                                VINPUT  2440
   57 FORMAT('1 VEHICLE ANGULAR TIME HISTORY',3X,20A4,3X,'PAGE NO.',I3//   VINPUT  2450
     *        4X,'TIME',  7X,'ANGULAR ACCELERATIONS (DEG/',A4,'**2)',      VINPUT  2460
     *                    7X,'ANGULAR VELOCITIES (DEG/',A4,')',            VINPUT  2470
     *                   11X,'ANGULAR DISPLACEMENTS (DEG)' /               VINPUT  2480
     *       3X,'(MSEC)',2(11X,'X',11X,'Y',11X,'Z',3X),                    VINPUT  2490
     *       10X,'YAW',8X,'PITCH',8X,'ROLL' /)                             VINPUT  2500
```

```
   58 IF(J.EQ.1) GO TO 60                                                    VINPUT  2510
C                                                                            VINPUT  2520
C     INTEGRATE ANGULAR VELOCITY AND DISPLACEMENT.                           VINPUT  2530
C                                                                            VINPUT  2540
      DO 59 I=1,3                                                            VINPUT  2550
      ATAB(I+12,J) = ATAB(I+12,J-1)+(ATAB(I+9,J-1)+ATAB(I+9,J))*ADT/2.0      VINPUT  2560
   59 THET(I) = ADT*(ATAB(I+12,J-1)+(2.0*ATAB(I+9,J-1)+ATAB(I+9,J))*ADT      VINPUT  2570
     */6.0)*RADIAN                                                           VINPUT  2580
      CALL DSETD(DVEH,THET,THT)                                              VINPUT  2590
   60 CALL YPRDEG(DVEH,THET)                                                 VINPUT  2600
      T1 = (AT0 + DFLOAT(J-1)*ADT)*1000.0                                    VINPUT  2610
   61 WRITE (6,56)  T1,(ATAB(I,J),I=10,15),THET                              VINPUT  2620
C                                                                            VINPUT  2630
C     PROGRAM INITIALIZATION FOR TIME = 0.                                   VINPUT  2640
C                                                                            VINPUT  2650
      CALL DRCYPR (DVEH,ANGLE,IDYPR)                                         VINPUT  2660
      DO 63 I=1,3                                                            VINPUT  2670
      XACOMP(I) = -G*ATAB(I,1)                                               VINPUT  2680
      VMEG(I)  = ATAB(I+12,1)*RADIAN                                         VINPUT  2690
   63 VMEGD(I) = ATAB(I+9 ,1)*RADIAN                                         VINPUT  2700
   64 J = MSEG                                                               VINPUT  2710
      IF  (MSEG.EQ.0)  GO TO 65                                              VINPUT  2720
      IF  (MSEG.LE.NSEG)  GO TO 66                                           VINPUT  2730
      IF  (MSEG.NE.NVEH+1)  STOP 6                                           VINPUT  2740
   65 NVEH = NVEH+1                                                          VINPUT  2750
      J = NVEH                                                               VINPUT  2760
C                                                                            VINPUT  2770
C     SETUP FOR ALL PRESCRIBED SEGMENT MOTION.                               VINPUT  2780
C                                                                            VINPUT  2790
   66 NVH = NVH+1                                                            VINPUT  2800
      ISING(J) = -1                                                          VINPUT  2810
      IF  (MSEG.GT.NSEG)  SEG(J) = VEH(NVH)                                  VINPUT  2820
      RW(J) = 0.0                                                            VINPUT  2830
      DO  67  I=1,3                                                          VINPUT  2840
      RPHI (I,J) = 0.0                                                       VINPUT  2850
      SEGLA(I,J) = VMEGD(I)                                                  VINPUT  2860
      WMEGD(I,J) = XACOMP(I)                                                 VINPUT  2870
   67 AXV(I,NVH) = AX(I)                                                     VINPUT  2880
      VT0(NVH) = AT0                                                         VINPUT  2890
      VDT(NVH) = ADT                                                         VINPUT  2900
      OMEGV(NVH) = OMEG                                                      VINPUT  2910
      TIMEV(NVH) = VTIME                                                     VINPUT  2920
      NVTAB(NVH) = NATAB                                                     VINPUT  2930
      INDXV(NVH) = J                                                         VINPUT  2940
      NJ = IABS(NATAB)                                                       VINPUT  2950
      IF  (NJ.LE.0)  GO TO 69                                                VINPUT  2960
      DO  68  K=1,NJ                                                         VINPUT  2970
      DO  68  I=1,3                                                          VINPUT  2980
      VATAB(I  ,K,NVH) = ATAB(I,K)                                           VINPUT  2990
   68 VATAB(I+3,K,NVH) = ATAB(I+9,K)                                         VINPUT  3000
```

```
   69 IF  (J.LE.NSEG)  GO TO 72                                          VINPUT  3010
C                                                                        VINPUT  3020
C     SETUP FOR NEW VEHICLE (SEGMENT) MOTION                             VINPUT  3030
C                                                                        VINPUT  3040
      W(J)  = 0.0                                                        VINPUT  3050
      RW(J) = 0.0                                                        VINPUT  3060
      DO  71  I=1,3                                                      VINPUT  3070
      DO  70  K=1,3                                                      VINPUT  3080
      D(I,K,J) = DVEH(I,K)                                               VINPUT  3090
   70 SGTEST(I,K,J) = 0.0                                                VINPUT  3100
      SGTEST(I,4,J) = 0.0                                                VINPUT  3110
      SEGLP(I,J) = XO(I)                                                 VINPUT  3120
      SEGLV(I,J) = XDOTO(I)                                              VINPUT  3130
      WMEG (I,J) = VMEG(I)                                               VINPUT  3140
      PHI  (I,J) = 0.0                                                   VINPUT  3150
   71 RPHI (I,J) = 0.0                                                   VINPUT  3160
   72 IF  (MSEG.NE.0)  GO TO 12                                          VINPUT  3170
      SEG(NVEH) = VEH(6)                                                 VINPUT  3180
C                                                                        VINPUT  3190
C     SET UP SEGMENT DATA FOR GROUND                                     VINPUT  3200
C                                                                        VINPUT  3210
      NGRND = NVEH+1                                                     VINPUT  3220
      IF  (NGRND.GT.30 .OR. NVH.GT.6)  STOP 7                            VINPUT  3230
      SEG(NGRND) = GRND                                                  VINPUT  3240
      J = NGRND                                                          VINPUT  3250
      ISING(J) = -1                                                      VINPUT  3260
      W(J)  = 0.0                                                        VINPUT  3270
      RW(J) = 0.0                                                        VINPUT  3280
      DO  74  I=1,3                                                      VINPUT  3290
      DO  73  K=1,3                                                      VINPUT  3300
      D(I,K,J) = 0.0                                                     VINPUT  3310
   73 SGTEST(I,K,J) = 0.0                                                VINPUT  3320
      D(I,I,J) = 1.0                                                     VINPUT  3330
      SGTEST(I,4,J) = 0.0                                                VINPUT  3340
      SEGLP(I,J) = 0.0                                                   VINPUT  3350
      SEGLV(I,J) = 0.0                                                   VINPUT  3360
      SEGLA(I,J) = 0.0                                                   VINPUT  3370
      WMEG (I,J) = 0.0                                                   VINPUT  3380
      WMEGD(I,J) = 0.0                                                   VINPUT  3390
      PHI  (I,J) = 0.0                                                   VINPUT  3400
   74 RPHI (I,J) = 0.0                                                   VINPUT  3410
      RETURN                                                             VINPUT  3420
      END                                                                VINPUT  3430
```

```
      DOUBLE PRECISION FUNCTION VISCOS(ZD,VISC,HA)                      VISCOS  0010
C                                                    REV 19 10/23/78    VISCOS  0020
C     COMPUTES SUM OF COULOMB AND VISCOUS TORQUES                       VISCOS  0030
C     AT JOINTS AS A FUNCTION OF THETA DOT.                             VISCOS  0040
C     ACTUALLY ROUTINE RETURNS SUM/ZD.                                  VISCOS  0050
C                                                                       VISCOS  0060
C         ARGUMENTS:                                                    VISCOS  0070
C          ZD   : ITHETA DOTI WHERE THETA IS THE ANGLE OF THE JOINT.    VISCOS  0080
C          VISC: ARRAY OF 5 VALUES DESCRIBING FUNCTION EVALUATION.      VISCOS  0090
C                                                                       VISCOS  0100
      IMPLICIT REAL*8 (A-H,O-Z)                                         VISCOS  0110
      DIMENSION VISC(5)                                                 VISCOS  0120
      Z = ZD                                                            VISCOS  0130
      IF (ZD.LT.VISC(3)) Z = VISC(3)/(2.0-ZD/VISC(3))                   VISCOS  0140
      HA = (Z-ZD)/Z                                                     VISCOS  0150
      VISCOS = VISC(1)+VISC(2)/Z                                        VISCOS  0160
      RETURN                                                            VISCOS  0170
      END                                                               VISCOS  0180
```

```
      SUBROUTINE VISPR(IJ,NJ)                                                  VISPR   0010
C                                                          REV 20 06/05/80    VISPR   0020
C     COMPUTES VISCOS AND SPRING TORQUES AT THE JOINTS                         VISPR   0030
C     AND ADDS THEM TO THE U2 ARRAY.                                           VISPR   0040
C                                                                              VISPR   0050
C     ARGUMENTS:                                                               VISPR   0060
C        NJ = 0 - REGULAR COMPUTATION FOR ALL JOINTS                           VISPR   0070
C           # 0 - COMPUTE ONLY FOR JOINT NJ IMPULSE                           VISPR   0080
C                                                                              VISPR   0090
C        IJ = 1 IMPULSE FOR FLEXURE ONLY                                       VISPR   0100
C           = 2 IMPULSE FOR TORSION ONLY                                       VISPR   0110
C           = 4 IMPULSE FOR GLOBALGRAPHIC ONLY                                 VISPR   0120
C                                                                              VISPR   0130
      IMPLICIT REAL*8 (A-H,O-Z)                                                VISPR   0140
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,                  VISPR   0150
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)               VISPR   0160
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),       VISPR   0170
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)              VISPR   0180
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(3,60),HA(3,60),HB(3,60),        VISPR   0190
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),            VISPR   0200
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)           VISPR   0210
      COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60),       VISPR   0220
     *               F(3,30),TQ(3,30),WJ(30)                                   VISPR   0230
      COMMON/FORCES/ PSF(7,30),BSF(4,20),SSF(10,20),BAGSF(3,20),               VISPR   0240
     *               PRJNT(7,30),NPANEL(5),NPSF,NBSF,NSSF,NBGSF                VISPR   0250
      COMMON/CEULER/ IEULER(30),HIR(3,3,30),ANG(3,30),ANGD(3,30),              VISPR   0260
     *               FE(3,30),TQE(3,30),CONST(3,30)                            VISPR   0270
      COMMON/TEMPVI/ CREST,TTI(3),R1I(3),R2I(3),JSTOP(4,2,30)                  VISPR   0280
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                                VISPR   0290
     *               UNITL,UNITM,UNITT,GRAVTY(3)                               VISPR   0300
      COMMON/TEMPVS/ T3(3),T6(3),T7(3),T8(3),T9(3),                            VISPR   0310
     *               WIJ(3),ANGL(3),DH1(3,3),HD3(3,3),                         VISPR   0320
     *               HAD,HBD,WIJM,CV,CSA,CSB,TQC                               VISPR   0330
      IF (NJNT.LE.0) GO TO 99                                                  VISPR   0340
      CALL ELTIME(1,13)                                                        VISPR   0350
      IF (NPRT(12).NE.0)WRITE (6,11)  TIME                                     VISPR   0360
   11 FORMAT('1 VISPR COMPUTATIONS FOR TIME =',F12.6)                          VISPR   0370
      J1 = 1                                                                   VISPR   0380
      J2 = NJNT                                                                VISPR   0390
      IF (NJ.EQ.0) GO TO 13                                                    VISPR   0400
      J1 = NJ                                                                  VISPR   0410
      J2 = NJ                                                                  VISPR   0420
   13 DO 90 J=J1,J2                                                            VISPR   0430
      DO 12 L=1,3                                                              VISPR   0440
      T3(L) = 0.0                                                              VISPR   0450
      T6(L) = 0.0                                                              VISPR   0460
      ANGL(L) = 0.0                                                            VISPR   0470
   12 TQ(L,J) = 0.0                                                            VISPR   0480
      WJ(J) = 0.0                                                              VISPR   0490
C                                                                              VISPR   0500
```

```
C     DO NOT COMPUTE TORQUES FOR NULL, LOCKED OR EULER JOINTS.                  VISPR   0510
C                                                                               VISPR   0520
      I = IABS(JNT(J))                                                          VISPR   0530
      IF (I.LE.0) GO TO 90                                                      VISPR   0540
      CALL DOT33 (D(1,1,J+1),HT(1,1,2*J),HIR(1,1,J))                            VISPR   0550
      IF  (IABS(IPIN(J)).GE.4)  GO TO 90                                        VISPR   0560
C                                                                               VISPR   0570
C     ZERO T1-T9 ARRAYS AND HAD,HBD,WIJM,CV,CS4,CSB AND TQC.                    VISPR   0580
C                                                                               VISPR   0590
      WIJM = 0.0                                                                VISPR   0600
      CV   = 0.0                                                                VISPR   0610
      CSA  = 0.0                                                                VISPR   0620
      CSB  = 0.0                                                                VISPR   0630
      TQC  = 0.0                                                                VISPR   0640
      CALL DOT33 (D(1,1,I),HT(1,1,2*J-1),DH1)                                   VISPR   0650
      CALL DOT33 (DH1,HIR(1,1,J),HD3)                                           VISPR   0660
      HAD  = HD3(3,3)                                                           VISPR   0670
      IF  (HAD.GT. 1.0)  HAD =  1.0                                             VISPR   0680
      IF  (HAD.LT.-1.0)  HAD = -1.0                                             VISPR   0690
      ANGL(1) = DARCOS(HAD)                                                     VISPR   0700
      IF  (HD3(2,3).NE.0.0 .OR. HD3(1,3).NE.0.0)                                VISPR   0710
     *ANGL(2) = DATAN2(HD3(2,3),HD3(1,3))                                       VISPR   0720
      ANGL(3) = DATAN2(HD3(2,1)-HD3(1,2),HD3(1,1)+HD3(2,2))                     VISPR   0730
      IF  (IPIN(J).LT.0)  GO TO 41                                              VISPR   0740
      IF (NJ.NE.0.AND.IJ.EQ.4) GO TO 27                                         VISPR   0750
C                                                                               VISPR   0760
C     CONVERT TO INERTIAL REFERENCE SYSTEM                                      VISPR   0770
C        T1= D(I)'*HA(NJ)      T4=D(J+1)'*HA(MJ)                                VISPR   0780
C        T3= D(I)'*WMEG(I)     T6=D(J+1)'*WMEG(J+1)                             VISPR   0790
C                                                                               VISPR   0800
C     HAD = COS TA = T1.T4                                                      VISPR   0810
C     WIJ = T3-T6                                                               VISPR   0820
C     WJ  = |WIJ|                                                               VISPR   0830
C                                                                               VISPR   0840
      DO 20 L=1,3                                                               VISPR   0850
      DO 15 M=1,3                                                               VISPR   0860
      T3(L) = T3(L)+ D(M,L,I)* WMEG(M,I)                                        VISPR   0870
   15 T6(L) = T6(L)+ D(M,L,J+1)* WMEG(M,J+1)                                    VISPR   0880
      WIJ(L)= T3(L)-T6(L)                                                       VISPR   0890
   20 WIJM = WIJM + WIJ(L)**2                                                   VISPR   0900
      WIJM = DSQRT(WIJM)                                                        VISPR   0910
      WJ(J) = WIJM                                                              VISPR   0920
C                                                                               VISPR   0930
C     T7 = T1 X T4                                                              VISPR   0940
C     HAC = |T7|                                                                VISPR   0950
C                                                                               VISPR   0960
      CALL CROSS (DH1(1,3),HIR(1,3,J),T7)                                       VISPR   0970
      HACC = T7(1)**2 + T7(2)**2 + T7(3)**2                                     VISPR   0980
      HAC = DSQRT(HACC)                                                         VISPR   0990
C                                                                               VISPR   1000
```

```
C      COMPUTE CV, THE MAGNITUDE OF VISCOUS AND COULOMB TORQUE/WIJM          VISPR   1010
C              RA = +SGN TA DOT = -WIJ.T7                                    VISPR   1020
C         AND CSA, THE MAGNITUDE OF FLEXURE TORQUE/HAC                       VISPR   1030
C                                                                            VISPR   1040
       CV = VISCOS(WIJM,VISC(1,3*J-2),HA2)                                   VISPR   1050
       IF  (NJ.EQ.0)  HA(2,2*J) = HA2                                        VISPR   1060
       CREST = VISC(7,3*J-2)                                                 VISPR   1070
       RA = -(WIJ(1)*T7(1) + WIJ(2)*T7(2) + WIJ(3)*T7(3))                    VISPR   1080
       IF  (HAC.NE.0.0)  RA = RA/HAC                                         VISPR   1090
       JSTP = 0                                                              VISPR   1100
       IF (JOINTF(J).EQ.0) CSA = EFUNCT(ANGL(1),RA,SPRING(1,3*J-2),JSTP)     VISPR   1110
       IF (JOINTF(J).NE.0) CSA = FNTERP(ANGL(1),ANGL(2),JOINTF(J))           VISPR   1120
       IF  (HAC.NE.0.0)  CSA = CSA/HAC                                       VISPR   1130
       IF (NJ.EQ.0) JSTOP(1,1,J) = JSTP                                      VISPR   1140
       IF (IPIN(J).EQ.1) GO TO 34                                            VISPR   1150
C                                                                            VISPR   1160
C      RB = +SGN TB DOT = -WIJ.T8                                            VISPR   1170
C      COMPUTE CSB, THE MAGNITUDE OF TORSIONAL TORQUE/HBC                    VISPR   1180
C                                                                            VISPR   1190
       RB  = -(WIJ(1)*HIR(1,3,J) + WIJ(2)*HIR(2,3,J) + WIJ(3)*HIR(3,3,J))    VISPR   1200
       CSB = EFUNCT(ANGL(3),RB,SPRING(1,3*J-1),JSTP)                         VISPR   1210
       IF (NJ.EQ.0) JSTOP(2,1,J) = JSTP                                      VISPR   1220
       IF (NJ.GT.0) GO TO 34                                                 VISPR   1230
C                                                                            VISPR   1240
C      COMPUTE EFFECT OF GLOBALGRAPHIC JOINT STOP (IPIN=3)                   VISPR   1250
C                                                                            VISPR   1260
    27 IF (IPIN(J).NE.3) GO TO 34                                            VISPR   1270
       CALL GLOBAL (J,HD3(1,3),DH1,TQC,T9,ANGL)                              VISPR   1280
C                                                                            VISPR   1290
C      COMPUTE TOTAL TORQUE IN INERTIAL REFERENCE BY                         VISPR   1300
C      TQ = -CV*WIJ + CSA*T7 + CSB*T8 + TQC*T9                               VISPR   1310
C                                                                            VISPR   1320
    34 IF (NJ.EQ.0) GO TO 36                                                 VISPR   1330
       CV = 0.0                                                              VISPR   1340
       IF (IJ.NE.1) CSA = 0.0                                                VISPR   1350
       IF (IJ.NE.2) CSB = 0.0                                                VISPR   1360
       IF (IJ.NE.4) TQC = 0.0                                                VISPR   1370
       IF (HA(2,2*J).EQ.0.0) GO TO 36                                        VISPR   1380
       CALL MAT31 (HIR(1,1,J),HA(1,2*J-1),TQ(1,J))                           VISPR   1390
       DO 38 L=1,3                                                           VISPR   1400
    38 TQ(L,J) = HA(2,2*J)*TQ(L,J)                                           VISPR   1410
    36 DO 37 L=1,3                                                           VISPR   1420
       TQ(L,J) = TQ(L,J) -CV*WIJ(L) +CSA*T7(L) +CSB*HIR(L,3,J) +TQC*T9(L)    VISPR   1430
    37 TTI(L) = TQ(L,J)                                                      VISPR   1440
       IF (NPRT(12).NE.0)WRITE (6,39)                                        VISPR   1450
      *         J,CV,CSA,CSB,HAC,RA,RB,(TQ(L,J),L=1,3),                    VISPR   1460
      *         WIJ,T7,ANGL, DH1 , HD3,                                    VISPR   1470
      *         ((HIR(L,K,J),L=1,3),K=1,3)                                 VISPR   1480
    39 FORMAT (1H0,I3,3F14.3,6F14.6/(4X,9F14.6))                             VISPR   1490
C                                                                            VISPR   1500
```

```
C     ADD TORQUE CONVERTED TO LOCAL REFERENCE BY                          VISPR   1510
C      U2I = U2I + DI*TQ                                                  VISPR   1520
C      U2J = U2J - DJ*TQ                                                  VISPR   1530
C                                                                         VISPR   1540
      DO 40 L=1,3                                                         VISPR   1550
      DO 40 M=1,3                                                         VISPR   1560
      U2(L,I  ) = U2(L,I  ) + D(L,M,I  )*TQ(M,J)                          VISPR   1570
   40 U2(L,J+1) = U2(L,J+1) - D(L,M,J+1)*TQ(M,J)                          VISPR   1580
C                                                                         VISPR   1590
C     STORE DATA FOR OUTPUT ROUTINE INTO PRJNT ARRAY.                     VISPR   1600
C                                                                         VISPR   1610
   41 PRJNT(1,J) = IPIN(J)                                                VISPR   1620
      PRJNT(2,J) = ANGL(1)                                                VISPR   1630
      PRJNT(3,J) = ANGL(2)                                                VISPR   1640
      PRJNT(4,J) = ANGL(3)                                                VISPR   1650
      PRJNT(5,J) = (CSA*HAC)**2 + CSB**2                                  VISPR   1660
      PRJNT(6,J) = (CV*WIJM)**2                                           VISPR   1670
      PRJNT(7,J) = TQ(1,J)**2 + TQ(2,J)**2 + TQ(3,J)**2                   VISPR   1680
   90 CONTINUE                                                            VISPR   1690
      CALL ELTIME(2,13)                                                   VISPR   1700
   99 RETURN                                                              VISPR   1710
      END                                                                 VISPR   1720
```

```
      SUBROUTINE WINDY(M,MM,N,NN,NT)                                    WINDY   0010
C                                                        REV 20 05/09/80  WINDY   0020
C     COMPUTES FORCES AND TORQUES ADDING THEM TO THE U1 AND U2 ARRAYS    WINDY   0030
C     OF WIND BLAST FORCES DETERMINED BY FUNCTION STORED IN TAB(NT)      WINDY   0040
C     ON ELLIPSOID (MM) ATTACHED TO BODY SEGMENT (M) WHICH EXTENDS       WINDY   0050
C     THROUGH THE INTERSECTING PLANE (NN) ATTACHED TO SEGMENT (N).       WINDY   0060
C                                                                        WINDY   0070
       IMPLICIT REAL*8 (A-H,O-Z)                                         WINDY   0080
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           WINDY   0090
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)        WINDY   0100
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30), WINDY   0110
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       WINDY   0120
      COMMON/TABLES/ MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(500),TAB(2600) WINDY   0130
      COMMON/WINDFR/ WTIME(30),QFU(3,5),QFV(3,5),                       WINDY   0140
     *               IWIND(30),MWSEG(5,30),NFVSEG(6),NFVNT(5)           WINDY   0150
      COMMON/CNTSRF/ PL(17,30),BELT(20,8),TPTS(6,8),BD(24,40)           WINDY   0160
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         WINDY   0170
     *               UNITL,UNITM,UNITT,GRAVTY(3)                        WINDY   0180
      COMMON/TEMPVS/ DMNT(3,3),XMN(3),XMM(3),TM(3),BET,BTS,P,FT(3),     WINDY   0190
     *               FF(3),AF(3),FAF,TF,BREF,SCALE,TRACER,AREA,RLM(3),  WINDY   0200
     *               TQM(3),RM(3)                                       WINDY   0210
      CALL ELTIME(1,37)                                                 WINDY   0220
      IF  (M.LT.0)  GO TO 50                                            WINDY   0230
C                                                                       WINDY   0240
C     COMPUTE PENETRATION DISTANCE; IF NEGATIVE, RETURN.                WINDY   0250
C                                                                       WINDY   0260
      CALL DOTT33 (D(1,1,M),D(1,1,N),DMNT)                              WINDY   0270
      DO 10 I=1,3                                                       WINDY   0280
   10 XMN(I) = SEGLP(I,M) - SEGLP(I,N)                                  WINDY   0290
      CALL MAT31 (D(1,1,M),XMN,XMM)                                     WINDY   0300
      CALL MAT31 (DMNT,PL(1,NN),TM)                                     WINDY   0310
      BET = PL(4,NN)                                                    WINDY   0320
      DO 11 I=1,3                                                       WINDY   0330
   11 BET = BET - TM(I)*(BD(I+3,MM)+XMM(I))                             WINDY   0340
      CALL MAT31 (BD(16,MM),TM,RM)                                      WINDY   0350
      BTS = TM(1)*RM(1) + TM(2)*RM(2) + TM(3)*RM(3)                     WINDY   0360
      BTE = -DSQRT(BTS)                                                 WINDY   0370
      P   = BET - BTE                                                   WINDY   0380
      IF (P.LT.0.0) GO TO 99                                            WINDY   0390
C                                                                       WINDY   0400
C     FETCH OR STORE INITIAL PENETRATION TIME.                          WINDY   0410
C                                                                       WINDY   0420
      IWIND(M) = M                                                      WINDY   0430
      IF (TIME.LE.WTIME(M)) WTIME(M) = TIME                             WINDY   0440
      FTIME = TIME - WTIME(M)                                           WINDY   0450
C                                                                       WINDY   0460
C     GET FORCE VECTOR FT FROM TABLE NT FOR TIME = FTIME.               WINDY   0470
C                                                                       WINDY   0480
   22 KT = NTI(NT)                                                      WINDY   0490
      NENTRY = TAB(KT+5)                                                WINDY   0500
```

```
      K1 = KT+10                                                        WINDY   051
      K2 = 4*NENTRY + KT+2                                              WINDY   052
      IF (NENTRY.EQ.1) GO TO 31                                         WINDY   053
      DO 30 K=K1,K2,4                                                   WINDY   054
      IF (FTIME.GT.TAB(K)) GO TO 30                                     WINDY   055
      KK = K                                                            WINDY   0560
      R1 = (TAB(K)-FTIME)/(TAB(K)-TAB(K-4))                             WINDY   0570
      GO TO 32                                                          WINDY   0580
   30 CONTINUE                                                          WINDY   0590
   31 KK = K2                                                           WINDY   0600
      R1 = 0.0                                                          WINDY   0610
   32 R2 = 1.0 - R1                                                     WINDY   0620
      DO 33 I=1,3                                                       WINDY   0630
      K= KK+I                                                           WINDY   0640
   33 FT(I) = R2*TAB(K) + R1*TAB(K-4)                                   WINDY   0650
C                                                                       WINDY   0660
C     COMPUTE PRESENTED AREA TO WIND FORCE.                             WINDY   0670
C                                                                       WINDY   0680
      CALL MAT31 (D(1,1,M),FT,FF)                                       WINDY   0690
      CALL MAT31 (BD(7,MM),FF,AF)                                       WINDY   0700
      FAF = FF(1)*AF(1) + FF(2)*AF(2) + FF(3)*AF(3)                     WINDY   0710
      IF (FAF.LE.0.0) GO TO 99                                          WINDY   0720
      TF  = TM(1)*FF(1) + TM(2)*FF(2) + TM(3)*FF(3)                     WINDY   0730
      BREF = DSQRT(BTS-TF*TF/FAF)                                       WINDY   0740
      SCALE = (-BET+BREF)/(-BTE+BREF)                                   WINDY   0750
      IF (SCALE.GE.1.0) GO TO 99                                        WINDY   0760
      IF (SCALE.LT.0.0) SCALE = 0.0                                     WINDY   0770
      TRACER = (BD( 7,MM)-AF(1)**2/FAF)*(BD(11,MM)-AF(2)**2/FAF)        WINDY   0780
     *       + (BD( 7,MM)-AF(1)**2/FAF)*(BD(15,MM)-AF(3)**2/FAF)        WINDY   0790
     *       + (BD(11,MM)-AF(2)**2/FAF)*(BD(15,MM)-AF(3)**2/FAF)        WINDY   0800
     *       - (BD( 8,MM)-AF(1)*AF(2)/FAF)**2                           WINDY   0810
     *       - (BD( 9,MM)-AF(1)*AF(3)/FAF)**2                           WINDY   0820
     *       - (BD(12,MM)-AF(2)*AF(3)/FAF)**2                           WINDY   0830
      AREA = (1.0-SCALE**2) * PI / DSQRT(TRACER)                        WINDY   0840
C                                                                       WINDY   0850
C     ADD FORCE AND TORQUES TO U1 AND U2 ARRAYS FOR SEGMENT M.          WINDY   0860
C                                                                       WINDY   0870
      SCALE = SCALE/BTE                                                 WINDY   0880
      DO 36 I=1,3                                                       WINDY   0890
      RLM(I) = RM(I)*SCALE + BD(I+3,MM)                                 WINDY   0900
      FT (I) = FT(I)*AREA                                               WINDY   0910
   36 FF (I) = FF(I)*AREA                                               WINDY   0920
      CALL CROSS (RLM,FF,TQM)                                           WINDY   0930
      DO 39 I=1,3                                                       WINDY   0940
      U1(I,M) = U1(I,M) + FT(I)                                         WINDY   0950
   39 U2(I,M) = U2(I,M) + TQM(I)                                        WINDY   0960
      IF (NPRT(14).NE.0) WRITE (6,41) TIME,M,P,AREA,FT,TQM              WINDY   0970
   41 FORMAT(' WIND FORCE',F14.6,I6,2F10.3,3X,3F12.5,3X,3F12.5)         WINDY   0980
      GO TO 99                                                          WINDY   0990
C                                                                       WINDY   1000
```

```
C     M NEGATIVE: CALCULATE FORCE FUNCTIONS.                          WINDY   1010
C                                                                     WINDY   1020
   50 NFORCE = NFVSEG(6)                                              WINDY   1030
      DO  60  J=1,NFORCE                                              WINDY   1040
      NFS = IABS(NFVSEG(J))                                           WINDY   1050
      NFT = IABS(NFVNT (J))                                           WINDY   1060
      KFT = NTI(NFT)                                                  WINDY   1070
      FRCE = EVALFD (TIME,KFT,1)                                      WINDY   1080
      IF  (NFVSEG(J).GT.0)  GO TO 52                                  WINDY   1090
      DO  51  I=1,3                                                   WINDY   1100
   51 U2(I,NFS) = U2(I,NFS) + FRCE*QFU(I,J)                           WINDY   1110
      GO TO 60                                                        WINDY   1120
   52 CALL DOT31 (D(1,1,NFS),QFU(1,J),TM)                             WINDY   1130
      DO  53  I=1,3                                                   WINDY   1140
      U1(I,NFS) = U1(I,NFS) + FRCE*TM(I)                              WINDY   1150
   53 U2(I,NFS) = U2(I,NFS) + FRCE*QFV(I,J)                           WINDY   1160
   60 CONTINUE                                                        WINDY   1170
   99 CALL ELTIME (2,37)                                              WINDY   1180
      RETURN                                                          WINDY   1190
      END                                                             WINDY   1200
```

```
      DOUBLE PRECISION FUNCTION XDY(X,D,Y)                                  XDY      0010
C                                                    REV 07 01/31/74        XDY      0020
C     FUNCTION ROUTINE TO COMPUTE X.DY OR Y.D'X                              XDY      0030
C                                                                            XDY      0040
      IMPLICIT REAL*8(A-H,O-Z)                                               XDY      0050
      DIMENSION X(3),D(3,3),Y(3)                                             XDY      0060
      XDY = 0.0                                                              XDY      0070
      DO 10 I=1,3                                                            XDY      0080
      DO 10 J=1,3                                                            XDY      0090
   10 XDY = XDY + X(I)*D(I,J)*Y(J)                                           XDY      0100
      RETURN                                                                 XDY      0110
      END                                                                    XDY      0120
```

```
      SUBROUTINE YPRDEG(D,A)                                                YPRDEG  0010
C                                                            REV 19 08/05/78  YPRDEG  0020
C     COMPUTES YAW PITCH AND ROLL IN DEGREES AND PLACES THEM                 YPRDEG  0030
C        INTO THE A ARRAY FOR A GIVEN DIRECTION COSINE MATRIX D.             YPRDEG  0040
C                                                                            YPRDEG  0050
C        ASSUMES D = D(R)D(P)D(Y) , WHERE                                    YPRDEG  0060
C                                                                            YPRDEG  0070
C             1  0  0            CP 0 -SP             CY SY 0               YPRDEG  0080
C     D(R) = 0  CR SR  ,D(P) = 0  1  0  AND D(Y) = -SY CY 0                  YPRDEG  0090
C             0 -SR CR           SP 0  CP             0  0  1               YPRDEG  0100
C                                                                            YPRDEG  0110
      IMPLICIT REAL*8(A-H,O-Z)                                               YPRDEG  0120
      DIMENSION A(3),D(3,3)                                                  YPRDEG  0130
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                              YPRDEG  0140
     *               UNITL,UNITM,UNITT,GRAVTY(3)                             YPRDEG  0150
      IF (D(1,1).EQ.0.0.AND.D(1,2).EQ.0.0) GO TO 10                          YPRDEG  0160
      IF (D(2,3).EQ.0.0.AND.D(3,3).EQ.0.0) GO TO 10                          YPRDEG  0170
      YAW = DATAN2(D(1,2),D(1,1))                                            YPRDEG  0180
      ROLL = DATAN2(D(2,3),D(3,3))                                           YPRDEG  0190
      GO TO 11                                                               YPRDEG  0200
   10 YAW = DATAN2(-D(2,1),D(2,2))                                           YPRDEG  0210
      ROLL = 0.0                                                             YPRDEG  0220
   11 PITCH = -DARSIN(D(1,3))                                                YPRDEG  0230
      IF (DABS(ROLL).LE.0.5*PI) GO TO 20                                     YPRDEG  0240
      IF (DABS(YAW ).LE.0.5*PI) GO TO 20                                     YPRDEG  0250
      PITCH = DSIGN(PI-DABS(PITCH),PITCH)                                    YPRDEG  0260
      YAW  = DATAN2(-D(1,2),-D(1,1))                                         YPRDEG  0270
      ROLL = DATAN2(-D(2,3),-D(3,3))                                         YPRDEG  0280
   20 A(1) = YAW/RADIAN                                                      YPRDEG  0290
      A(2) = PITCH/RADIAN                                                    YPRDEG  0300
      A(3) = ROLL/RADIAN                                                     YPRDEG  0310
      RETURN                                                                 YPRDEG  0320
      END                                                                    YPRDEG  0330
```

9.0 REFERENCES

1. Bartz, John A., "A Three-Dimensional Computer Simulation of a Motor Vehicle Crash Victim, Phase I - Development of the Computer Program", Calspan Report No. VJ-2978-V-1, July, 1971.

2. Bartz, John A. and Butler, Frank E., "A Three-Dimensional Computer Simulation of a Motor Vehicle Crash Victim, Phase 2 - Validation Study of the Model", Calspan Report No. VJ-2978-V-2, December, 1972.

3. Bartz, John A., Butler, Frank E. and Ryan, Charles T., "A Three-Dimensional Computer Simulation of a Motor Vehicle Crash Victim, Further Development - Mutual Force-Deflection Characteristics and Comprehensive 'Debug' Facility", Calspan Report No. ZQ-5326-V-2, September, 1974.

4. Fleck, J. T., Butler, F. E., and Vogel, S. L., "An Improved Three-Dimensional Computer Simulation of Motor Vehicle Crash Victims", Volumes I-IV, Report Nos. DOT-HS-801507, -508, -509, -510, July, 1974.

5. Fleck, J. T. and Butler, F. E., "Development of an Improved Computer Model of the Human Body and Extremity Dynamics", Report No. AMRL-TR-75-14, July, 1975 (AD A-014816).

6. Butler, F. E. and Fleck, J. T., "Advanced Restraint System Modeling", Report No. AFAMRL-TR-80-14, May, 1980.

7. Federal Motor Vehicle Safety Standard 208, Part 572, "Anthropomorphic Test Dummy".

8. "Development of Approximating Solutions for CVS Program and of Dummy Design Information", Contract No. DOT-HS-6-01418 (in progress).

9. DeLeys, Norman J., "Data For Validation of Crash Victim Simulator", Calspan Report No. 6197-V-1, August, 1981.

10. Bowman, William L., "The Articulated Total Body (ATB) VIEW Package Software Report", Report No. AFAMRL-TR-81-111, June 1981.

11. Bowman, William L., "The Articulated Total Body (ATB) VIEW Package User Guide", Report No. AFAMRL-TR-81-124, August 1981.

12. Strieb, M., "Structural Modification of the 'Three-Dimensional Crash Victim Simulation' (CVS) Software", Technical Report No. 1194-A, Analytics Inc., Willow Grove, PA 19090, May 1976 (AD-A027726).