

AD-A238 767



AAMRL-SR-90-502

SCD
L.I.C
ELECTE
JUL 16 1991

MICROCOMPUTER ENHANCEMENT OF THE ARTICULATED TOTAL BODY (ATB)
BIODYNAMIC MODELING SYSTEM

Andras Spiegel
Robert L. Hyslop
Ketron, Inc.
600 Louis Drive
Warminster, PA 18974

January 1990

Final Report for December 1988 to January 1990

Approved for public release; distribution is unlimited.

HARRY G. ARMSTRONG AEROSPACE MEDICAL RESEARCH LABORATORY
HUMAN SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-6573

91-05097



91 7 15 089

NOTICES

When US Government drawings, specifications, or other data are used for any purpose other than a definitely related Government procurement operation, the Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise, as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use or sell any patented invention that may in any way be related thereto.

Please do not request copies of this report from the Harry G. Armstrong Aerospace Medical Research Laboratory. Additional copies may be purchased from:

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22314

TECHNICAL REVIEW AND APPROVAL
AAMRL-SR-90-

This report has been reviewed by the Office of Public Affairs (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

JAMES W. BRINKLEY
Director
Biodynamics and Bioengineering Division
Harry G. Armstrong Aerospace Medical Research Laboratory

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION Unclassified		1b. RESTRICTIVE MARKINGS -None-	
2a. SECURITY CLASSIFICATION AUTHORITY		3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited.	
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A		5. MONITORING ORGANIZATION REPORT NUMBER(S) AAMRL-SR-90-	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) KTR 5095-02		7a. NAME OF MONITORING ORGANIZATION Harry G. Armstrong Aerospace Medical Research Laboratory	
6a. NAME OF PERFORMING ORGANIZATION KETRON, INC.	6b. OFFICE SYMBOL (if applicable)	7b. ADDRESS (City, State, and ZIP Code) Wright-Patterson AFB OH 45433-6573	
6c. ADDRESS (City, State, and ZIP Code) 600 Louis Drive Warminster PA 18974		9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F33615-88-C-0543	
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Same as 7a	8b. OFFICE SYMBOL (if applicable) AAMRL/BBM	10. SOURCE OF FUNDING NUMBERS	
8c. ADDRESS (City, State, and ZIP Code) Same as 7b		PROGRAM ELEMENT NO. 65505F	PROJECT NO. 7231
		TASK NO. 23	WORK UNIT ACCESSION NO. 14
11. TITLE (Include Security Classification) Microcomputer Enhancement of the Articulated Total Body (ATB) Biodynamic Modeling System			
12. PERSONAL AUTHOR(S) Spiegel, Andras and Hyslop, Robert L.			
13a. TYPE OF REPORT Final	13b. TIME COVERED FROM 9 Dec 88 TO 31 Jan 90	14. DATE OF REPORT (Year, Month, Day) 31 Jan 1990	15. PAGE COUNT 706
16. SUPPLEMENTARY NOTATION Small Business Innovative Research (SBIR) Phase I contract			
17. COSATI CODES		18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)	
FIELD	GROUP	Biodynamic Modeling, Simulation, Microcomputers, ATB, GEBOD, VIEW, Acceleration, Ejection	
20	11		
12	05		
19. ABSTRACT (Continue on reverse if necessary and identify by block number) Presented on other side.			
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT <input type="checkbox"/> DTIC USERS		21. ABSTRACT SECURITY CLASSIFICATION Unclassified	
22a. NAME OF RESPONSIBLE INDIVIDUAL LOUISE OBERGEFELL		22b. TELEPHONE (Include Area Code) (513) 255-3665	22c. OFFICE SYMBOL AAMRL/BBM

ACKNOWLEDGEMENTS

KETRON wishes to acknowledge the professional support provided us by Dr. Ints Kaleps, Ms. Louise A. Obergefell, and Lt. Deborah Determan. This support included documentation, sample problems, comparison data, as well as their time and expertise during invaluable technical discussions.

Accession For	
	<input checked="" type="checkbox"/>
	<input type="checkbox"/>
	<input type="checkbox"/>
	<input type="checkbox"/>
By	
Distribution/	
Availability Codes	
Dist	Special
A-1	



TABLE of CONTENTS

Section	Title	Page
1.	INTRODUCTION.....	1
1.1	Background.....	1
1.2	Objective.....	2
1.3	Report Overview.....	3
2.	SUMMARY.....	4
2.1	Results.....	4
2.2	Conclusions.....	6
2.3	Recommendations.....	7
3.	DISCUSSION.....	9
3.1	Introduction.....	9
3.2	Design criteria.....	9
3.2.1	Hardware.....	9
3.2.2	Operating System.....	9
3.2.3	Operating/developmental environment.....	9
3.2.4	Performance times.....	10
3.3	Hardware configuration.....	10
3.3.1	Selection Criteria.....	10
3.3.1.1	Processor, Memory, Clock.....	10
3.3.1.2	Display.....	11
3.3.1.3	Disk drives.....	11
3.3.1.4	Printer/plotter.....	11
3.3.2	Recommended hardware.....	12
3.4	Operating system.....	12
3.4.1	Selection Criteria.....	12
3.4.2	Recommended operating system.....	13
3.5	Support software.....	13
3.5.1	Selection Criteria.....	14
3.5.1.1	The UNIX Environment.....	14
3.5.1.2	The DOS Environment.....	15
3.5.2	Recommended support software.....	15
4.	ATBMS IMPLEMENTATION.....	16
4.1	Introduction.....	16
4.2	SYSTEM CONFIGURATION.....	16
4.2.1	Hardware Configuration.....	16
4.2.2	Support Software Configuration.....	16
4.2.3	Hard Disk Organization.....	16
4.2.3.1	The boot directory.....	16
4.2.3.2	The working drive.....	17
4.2.3.3	The compiler directory.....	17
4.2.3.4	The DOS extender directory.....	17

TABLE of CONTENTS (cont'd)

<u>Section</u>	<u>Title</u>	<u>Page</u>
4.2.3.5	The graphics directory.....	17
4.2.3.6	The working directory.....	17
4.2.4	Compiling and Linking FORTRAN.....	18
4.2.5	ATBMS DIRECTORY Files.....	19
4.2.6	Executing Programs.....	20
4.3	PROGRAM IMPLEMENTATION.....	20
4.3.1	The GEBOD Program.....	20
4.3.1.1	Source code modifications.....	20
4.3.1.2	The GEBODII.DIR file.....	21
4.3.1.3	Sample outputs.....	21
4.3.2	The ATB Program.....	22
4.3.2.1	Source code modifications.....	22
4.3.2.2	The ATBIV.DIR file.....	23
4.3.2.3	Sample outputs.....	24
4.3.3	The VIEW Program.....	24
4.3.3.1	Source code modifications.....	24
4.3.3.2	The VIEWPE.DIR file.....	25
4.3.3.3	Sample outputs.....	25
5.	REFERENCES.....	26

LIST of TABLES

<u>Number</u>	<u>Title</u>	<u>Page</u>
2-1	ATBMS Microcomputer System Comparisons.....	5
2-2	ATB Execution Time Comparisons.....	5
2-3	ATBMS Compilation and Link Time Comparisons.....	6
3-1	Recommended ATBMS Hardware Configuration.....	12

APPENDICES

<u>Letter</u>	<u>Title</u>
A	GEBOD Body Description Tabular Output (GEBOD.TAB)
B	GEBOD Body Description File Output (GEBOD.AIN)
C1	Example 1, ATB Basic Sled Test Simulation -- Input File (EX1.AIN)
C2	Example 1, ATB Basic Sled Test Simulation -- 386 Directory File (ATBIV1.DIR)

TABLE of CONTENTS (cont'd)

<u>Letter</u>	<u>Title</u>
G3	Example 1, ATB Basic Sled Test Simulation -- 386 Output File (EXPl.AOU)
G4	Example 1, ATB Basic Sled Test Simulation -- Perkin-Elmer Output File
D1	Example 2, ATB Dynamic Joint Test -- Input File (EX2.AIN)
D2	Example 2, ATB Dynamic Joint Test -- 386 Directory File (ATBIV2.DIR)
D3	Example 2, ATB Dynamic Joint Test -- 386 Output File (EXP2.AOU)
D4	Example 2, ATB Dynamic Joint Test -- Perkin-Elmer Output File
E1	Example 2, ATB Dynamic Joint Test -- Input File with Plotting Options (EX2PLOT.AIN)
E2	Example 2, ATB Dynamic Joint Test -- X-Y Plot
E3	Example 2, ATB Dynamic Joint Test -- Printer Plot
F1	Example 1, VIEW -- Input File (EXPl.VIN)
F2	Example 1, VIEW -- 386 Output Listing (EXPl.VOU)
F3	Example 1, VIEW -- 386 Output Graphics
G1	Example 2, VIEW -- Input File (EXP2.VIW)
G2	Example 2, VIEW -- 386 Output Listing (EXP2.VOU)
G3	Example 2, VIEW -- 386 Output Graphics
H	386-GEBOD Program Listing
I	386-ATB Program Listing
J	386-VIEW Program Listing

1. INTRODUCTION

1.1 BACKGROUND

The *Articulated Total Body* (ATB) model was derived from the *Three Dimensional Crash Victim Simulation* (3DCVS) which was originally conceived as a research tool to enable investigation of driver or occupant and motor vehicle interaction under various restraint configurations and other initial conditions during automobile crashes. Researchers at the U.S. Air Force Harry G. Armstrong Aerospace Medical Research Laboratory (AAMRL) adopted this model for the study of aircrew members during ejections from aircraft, with the view of eliminating or reducing injuries in a high acceleration environment. The use of modeling in this area of study is particularly appropriate and attractive since it reduces the cost of testing (whether with manikins or human subjects) and eliminates the risks associated with human testing. The modeling approach also supports system evaluations under operating conditions that would otherwise be infeasible to test with human subjects, and where the biofidelity of manikins is inadequate.

The *Three Dimensional Crash Victim Simulation* was developed by J. A. Bartz at the Cornell Aeronautical Laboratory for the U.S. Department of Transportation (DoT). The original model is a system of over 180 simultaneous ordinary differential equations programmed for solution by numerical integration, on an IBM mainframe. The 3DCVS model was improved by Fleck, Butler, and Vogel under further DoT sponsorship. Air Force motivated enhancements include the modeling of significant headblast forces during high speed ejections, and the modeling of complex, interacting-element restraint systems. Additional enhancements have included the development of significant graphic output capabilities to facilitate the analysis of results.

To facilitate the use of the model, additional application software has been written. These programs are GEBOD, VIEW, and CABS, which, together with ATB, comprise the ATB Modeling System (ATBMS). GEBOD prepares the body description for input to ATB using interactive inputs from the analyst. The VIEW program plots projected views of body position from the ATB simulation at constant time steps. CABS is used to compare tabular time history data files from

two ATB simulations.

The simulation, the numerical solution of the model, which was originally implemented on IBM mainframes, was adapted first to the CDC 6600 (another mainframe), and subsequently at AAMRL to the Perkin-Elmer 3250XP mini-computer.

1.2 OBJECTIVE

The ATB Modeling System is needed to address a variety of new technologies and challenges which are changing the operating environment of aircrews. These changes include increasing aircraft performance, which increase the need for expanding the safe ejection envelope. The increasing complexity of the weapon systems, cockpit, and missions have also stimulated the development of equipment to aid the aircrews -- night vision goggles, helmet mounted sights, and helmet mounted displays. Each of these devices has the potential for impacting the safe ejection envelope and increasing the injury potential of an ejection (as well as increasing the fatigue of normal flying). While microelectronics, fiber optic technology, and miniaturization can keep the equipment weight small, even these weights at 15 G ejection acceleration may pose a substantial injury hazard.

In view of the hazard potential of the technologies being introduced into the cockpit, exploration of their biodynamic effects is necessary. The use of the ATB Modeling System currently requires either costly mainframe computers, or a substantial minicomputer laboratory, which restricts its use at a time when computing technology is rapidly evolving, and is becoming available at the desktop level at an affordable cost. It is appropriate to reduce or eliminate the ATBMS analysis restrictions imposed by the present costly computer requirements in view of the needs of investigators in the DoD, industry, and the academic community to assess the consequences of new flightmission hardware developments.

Accordingly, this effort was undertaken with the objective of demonstrating the feasibility of using the Articulated Total Body Modeling System on readily available microcomputer systems.

1.3

REPORT OVERVIEW

Section 2 summarizes the results of this effort, with conclusions and recommendations. The system design criteria, hardware, operating system, and support software selections are described in Section 3. Section 4 discusses the software conversion and program execution. Supporting data are presented in the Appendices.

2. SUMMARY

2.1 RESULTS

The Articulated Total Body Modeling System was converted from the Perkin-Elmer 3250XP minicomputer to an Intel 80386/80387 microprocessor based personal computer under a Disk Operating System (DOS) environment. The design criteria, discussed in Section 3, were satisfied without compromises. The resulting hardware and software configuration suitable for using, or conducting further developmental work with, the ATBMS is also documented in Section 3. The results obtained on two sample cases using the microcomputer compare favorably with the results obtained from the Perkin-Elmer in terms of computational reproducibility. The results for comparison are presented in Appendices C3, C4, D3, and D4, respectively. The graphics capability presently in the ATBMS is also successfully replicated, as shown in Appendices E2, E3, F3, and G3.

The relevant hardware and performance characteristics of the computer systems used in this implementation are presented in Table 2-1. The significant differences between systems B and C are the presence of cache memory in system C and the clock frequencies of 20 vs 33 MHz. These two differences result in a three to one computational performance ratio, as indicated by both the Whetstone and Dhrystone measures. Two sample cases were used in comparing alternative hardware systems -- a basic sled test simulation (marked as example 1) and a dynamic joint test (example 2).

The execution timing results, summarized in Table 2-2, show that the 80386 system, operating with a 20 MHz clock, (80386/20, system B), is somewhat slower than the Perkin-Elmer. (System A is omitted from Table 2-2 due to insufficient memory for ATB execution.) However, the 80386/33 system (C) with cache memory is more than twice (2.17) the speed of the Perkin-Elmer. The comparison of two 80386 systems, at 20 and 33 MHz respectively, shows a performance ratio of slightly under three -- 2.79 and 2.95 for the two sample cases. These results are consistent with the systems' computational characteristics (3:1), which are slightly degraded by the I/O requirements. In view of the relatively high computation to I/O ratio of the ATB simulation the disk performance differences have only a slight impact.

Table 2-1. ATBMS Microcomputer System Comparisons

	<u>SYSTEM</u>		
	<u>A</u>	<u>B</u>	<u>C</u>
<u>HARDWARE CHARACTERISTICS</u>			
Clock frequency - MHz	16	20	33.35
Memory, main, kb	640	640	640
Memory, extended, kb	1024	3456	7552
Memory, cache, kb	0	0	64
<u>PROCESSOR PERFORMANCE'</u>			
Dhrystones	3501	3793	11379
Whetstones, k	850.2	945.0	2933.8
<u>PROCESSOR PERFORMANCE</u> <u>RELATIVE to 20 MHz SYS.</u>			
Dhrystones	92	100	300
Whetstones	90	100	310
<u>DISK PERFORMANCE'</u>			
capacity, mb	70	348	150
track to track seek, ms	3.42	5.32	5.36
average seek, ms	25.3	13.0	20.4
transfer rate, kb/sec	33.6	182.0	35.2

1: processor and disk performance were measured using *CheckIt*, Ver. 1.10, TouchStone Software Corporation, Seal Beach, CA 90740.

Table 2-2. ATB Execution Time Comparisons

<u>ATB</u> <u>PROBLEM</u>	<u>Perkin-Elmer</u> <u>3250XP</u>	<u>Intel 80386/80387</u>	
		<u>20 MHz</u> <u>(B)</u> <u>seconds</u>	<u>33 MHz</u> <u>(C)</u>
1	291.45	395.79	134.29
2	30.91	38.67	13.84
<u>percent, relative to 3250XP</u>			
1	100	136	46
2	100	125	45
<u>percent, relative to 80386/20</u>			
1	74	100	34
2	80	100	36

For program compilation and linking Table 2-3 shows the results for a 80386/7 at three different clock frequencies -- 16, 20, and 33 MHz. The data indicate that the 33 MHz system is approximately 1.7 times faster than the 20 MHz processor at this task. This speed ratio is much lower than the value (~2.9) for execution. The difference in compilation and linking performance is probably due to the relatively large volume of I/O which is slower with the disk installed on the 33 MHz system. This hypothesis is supported by the performance differences between compilation and linking, with linking being slower than compilation, due to more disk accesses to the library to resolve references.

Table 2-3. ATBMS Compilation and Link Time Comparisons

PROGRAM	16 MHz		20 MHz		33 MHz	
	(A)		(B)		(C)	
			<u>seconds</u>			
GEBOD	43	19	39	11	22	7
ATB	382	108	331	59	183	41
VIEW	41	54	35	32	19	22
			<u>percent, relative to 80382/20</u>			
GEBOD	110	173	100	100	56	64
ATB	115	183	100	100	55	69
VIEW	117	154	100	100	54	69
			<u>compilation + linking percent, relative to 80386/20</u>			
GEBOD	124		100		58	
ATB	126		100		57	
VIEW	142		100		61	

2.2 CONCLUSIONS

The results presented in Section 2.1 unquestionably demonstrate the capability of highly economical microcomputers for conducting both analytical and further developmental work with the ATBMS. The ATBMS conversion and execution results show that:

- (a) production hardware, commercially available disk operating systems and support software comprise a suitable host for the ATBMS work,
- (b) the demonstrated microcomputer host system performs without compromising the numerical accuracy or graphics capability of the ATBMS,
- (c) the high performance microcomputer (Intel 80386/80387 based, with 33 MHz clock rate and cache memory) offers execution times markedly superior

to that provided by the Perkin-Elmer 3250XP minicomputer, and

(d) based on the hardware and software requirements established in the course of this study, the microcomputer-based ATBMS use is considerably more economical than the minicomputer alternative.

2.3 RECOMMENDATIONS

In view of the demonstrated results, which show that the microcomputer based ATBMS is both technically and economically superior to its earlier-technology minicomputer implementation, further enhancement of the 386 ATBMS is recommended. Some specific enhancement areas are described below, while a fully developed set of recommended changes will be documented separately.

The principal areas for ATBMS enhancements are user interface improvements (UII) and performance upgrades (PU). UIIs include the extension of the program-files interface, which was implemented for this phase with the introduction of DIRECTORY files, described in Section 4.2.5. The extension will retain the DIRECTORY file concept, but expand it to take advantage of the full DOS path support, which is supported by the compiler and linker. This extension should also enable simplification of both file naming/moving rules and the ATBMS directory structure.

Since some simulations may require considerable processing time, when a large number of segments and joints are simulated for a long interval, a display of the simulation progress would be useful. The in-progress display might show the present time interval, the segment or joint number, indicators of integration convergence history, percent completed, as well as lower and upper bounds of the time required to complete execution.

An enhancement area which bridges both UII and PU is the interactive construction and testing of simulation input files. This capability, which is partially implemented by the GEBOD program, should reduce the time and effort required to prepare and test ATB inputs. The significant features of such an interactive program would naturally include accurate file construction in terms of data field and record placement, but more importantly would test data in terms of units of measure selected, perform conversions, and display options in menu

format in English and translate the responses into appropriate switch/flag settings. Upon completion the program would test the resulting file for completeness to ensure that all data related to explicitly or implicitly selected options were present. Discrepancies would be flagged, with an opportunity to correct faults on-line, until a satisfactory input file was ready for simulation.

The two significant performance upgrades recommended at this time include graphics capability enhancements and the development of a post-simulation analysis package. The graphics upgrade would take advantage of relatively recent hardware and software developments to provide the analyst with high quality graphs, plots, and pictures, optimized to control processing time.

Postsimulation data extraction, reduction, analysis, and display are necessary in view of the large quantity of data produced by a single simulation, and by the need to correlate and compare -- statistically and graphically -- the results of several simulations, or simulation and test results. Such data handling software's principal features include the ability to review large quantities of data rapidly, select data from various (similar or dissimilar) sources for comparison, with a choice of filters and display formats selected interactively by the user. References [7, 8] document data analysis packages with such functional characteristics. These are presently used on CDC Cyber and HP computers respectively.

3. DISCUSSION

3.1 INTRODUCTION

This section of the report summarizes the technical issues relevant to the adaptation of the ATBMS to the microcomputer environment. Specific topics discussed include the design criteria, hardware configuration, operating systems, support software, and changes to the application software.

It should be noted that in the era of rapidly evolving microcomputer hardware and software, as well as peripherals (displays, bulk storage devices, hard copy devices), subjective judgements were made -- and revised -- which "date" this report as it is being written. Nevertheless, the *method* used in development will stand the test of time, while the specific recommendations may change with time.

3.2 DESIGN CRITERIA

3.2.1 Hardware

In view of the overall objective -- to make the ATBMS readily available to investigators having limited resources -- it was a design criterion to select a hardware configuration that would provide acceptable ATBMS performance at reasonable cost, using standard, commonly and readily available hardware components. Hardware selection was further oriented toward systems whose continued support could reasonably be expected.

3.2.2 Operating System

An efficient, well known, commonly used operating system is desired that interacts with readily available support software, at an acceptable cost.

3.2.3 Operating/developmental Environment

It was a design criterion that the combination of hardware, operating system, support software, and application software should jointly constitute an environment appropriate to investigators in biodynamics, rather than microcomputer specialists.

3.2.4 Performance Times

In view of the computational intensity of the ATBMS, and particularly of the ATB simulation itself, an execution time not exceeding twice that of the Perkin-Elmer was desired.

3.3 HARDWARE CONFIGURATION

3.3.1 Selection Criteria

3.3.1.1 Processor, Memory, Clock

At the commencement of this project there were five classes of candidate microprocessor based systems potentially suited to the ATBMS application. These were the Intel 8000 based systems (IBM personal computers and their clones), the Motorola 68000 based Apple and Macintosh lines, microcomputers made by the Digital Equipment Corporation and by Hewlett-Packard (both using proprietary processors), and the various makes of "workstations".

The DEC and HP systems are not particularly common in general applications. The workstations, while offering the highest performance in the group, were significantly more expensive than other equipment. The Motorola 68000 based systems are not as common as the Intel systems and offer a smaller choice of support software.

Thus, by the process of elimination early in the project, the principal focus was on the Intel 8000 based systems. They appeared to be the most suitable due to their widespread use, which stimulates the commercial software vendors to develop and enhance the necessary support products. In view of the broad market base, continued hardware and software support can be reasonably expected. The Intel family of microprocessors also offers a range of performance capabilities through a choice between the 8086, 80286, and 80386 devices that are available at various clock frequencies, with their respective mathematical coprocessors. In view of the computational load posed by the ATB simulation, the Intel 80386/80387 processors are most suited to this application.

The ATBMS memory requirements are dictated by the ATB simulation, which is its largest component. Either sufficient memory has to be available to keep the program and data in memory during the entire execution, or a software

overlay approach has to be adopted. The in-memory alternative is preferred in view of the large volume of disk I/O needed for the overlay approach. Even with the substantial data transfer rates of high capacity disks, the system performance would suffer if overlays were used. The ATB simulation requires 4 Mb memory. The recommended processor supports up to 16 Mb of memory, which satisfies the preferred memory alternative.

The Intel 80386 processor is available in 16, 20, 25, and 33 MHz variants. It is clear from data presented above that increasing the clock frequency can significantly contribute to reducing execution times.

3.3.1.2 Display

While a very high resolution CAD type display is highly effective for viewing graphic output, a good quality EGA display is satisfactory for the ATB application.

3.3.1.3 Disk drives

Since most of the processing in ATBMS is computationally intensive, with a relatively low level of input/output activity, disk access time is not particularly critical. To accommodate the large output files, however, a disk of at least 70 Mb is recommended. Disks of that size typically are made with 28 ms access time, or less. Larger capacity disks would be recommended for high intensity analysis activity, particularly if frequent disk backups need to be avoided.

As software distribution and data handling media, floppy disks are the most common. User convenience would be well served if both common formats (5.25" and 3.5") were available.

3.3.1.4 Printer/plotter

The ATBMS can produce a large quantity of tabular output, as well as graphs. While a wide variety of printers and plotters are available, a single device -- a laser type printer -- can serve both as a high volume printer and a plotter. A printer operating at or above eight pages per minute is recommended.

3.3.2 Recommended Hardware

The hardware configuration recommended for ATBMS is summarized in Table 3-1.

Table 3-1. Recommended ATBMS Hardware Configuration

<u>Item</u>	<u>Description</u>
Processor	Intel 80386
Coprocessor	Intel 80387
Clock	33 MHz desirable
Memory quantity	4 Mb required
Cache memory	64 Kb
Memory speed	80 ns or less
Display	EGA or VGA
Disk drive capacity	70 Mb
Disk drive access	28 ms or less
Floppy disks	1.2 Mb, 5.25" 1.4 Mb, 3.5"
Printer/plotter	Laser Jet type 8 pages/minute, minimum

If computer-to-computer data transfer is desired, a modem (internal or external) can be added to the system. In view of the volume of data that is often generated by ATBMS, the analyst may find a tape-based backup system quite useful for archiving, particularly with a small hard disk and high analysis or developmental activity level.

3.4 OPERATING SYSTEM

3.4.1 Selection Criteria

Two operating systems are readily available for Intel microprocessor based systems -- the Disk Operating System, DOS, and variants of Bell Laboratories' UNIX system.

UNIX on a microcomputer offers two attractive features. First, all installed memory in the computer is directly available to the executing application. Second, microcomputer UNIX is available in a multiuser configuration.

This second feature is not of significant utility for the ATBMS implementation. In view of the computational load imposed by the simulation it is unlikely that the computer would be employed in a multiuser mode.

Since UNIX is not nearly as widely used as DOS, the quantity of available developmental software is somewhat limited, and is markedly more expensive than DOS software. Microcomputer users generally feel that UNIX based systems require more expertise and experience than DOS systems. Other attributes of the UNIX system will be discussed in Section 3.5, in conjunction with support software.

DOS presents two disadvantages in comparison with UNIX -- in conventional operations available application memory is limited to 640 Kb (regardless of amount installed), and the number of simultaneously open files within an application is limited to 15. These disadvantages, however, are alleviated by commercially available software packages which overcome DOS's built-in limits. Significant advantages to DOS are that it is well supported by a broad spectrum of commercially developed tools, which are typically less expensive than their UNIX counterparts -- when those counterparts exist. Furthermore DOS is well known not only to software developers, but also to application oriented computer users.

3.4.2 Recommended Operating System

Based on the above discussion, and observations presented in Section 3.5, DOS is the operating system of choice for the ATBMS implementation.

3.5 SUPPORT SOFTWARE

The term support software is used to cover compilers, linkers, graphics packages, and DOS memory extenders. These are all commercially available products necessary to complete the software portion of the hardware-software environment for ATBMS operation.

3.5.1 Selection Criteria

The support software selection was undertaken with the requirement to provide an efficient developmental/operating environment for the ATBMS. The

term efficient includes well documented, easy to use software, without the need for "work arounds" to solve potentially intractable problems. The operating environment has to be suitable to the problem-solving oriented application program user, rather than the dedicated, expert software developer.

It was also desired to minimize application software revisions, and to avoid new application development in order to fit the ATBMS to an operating environment.

3.5.1.1 The UNIX environment

When using a UNIX family operating system, reference [1], we found that UNIX based compiler/linker would not accept standard FORTRAN labelled COMMON specifications unless they were of identical length in each subroutine and function throughout the entire program. (Most compilers and linkers will process code -- with a warning -- as long as the longest COMMON is declared on its first occurrence. Shorter subsequent declarations are accepted without error.) This problem can be eliminated by suitably padding each COMMON declaration.

The second challenge posed by the UNIX operating system was the unavailability of a graphics support package capable of handling the CalComp graphics calls. CalComp (California Computer Products) is an old, well established X-Y plotter, which was available to the scientific applications community through FORTRAN CALL statements, which were supported by a dedicated load-time library. Their use became sufficiently widespread that "after-market" support libraries have been written for several devices, such as the TEKTRONIX terminals, and more recently for microcomputers. Such a library was not available for a microcomputer operating under UNIX.

Thus the available choices were either to modify the ATBMS to run under UNIX outputting data to a file, which would be post-processed under UNIX or DOS. The UNIX option would have dictated the development of special application programs to handle ATBMS graphics. The DOS option would also have required some custom software development, and the use of two operating systems to run the ATBMS, with some data moving between the UNIX and DOS file systems.

Based on the above, it is clear that using the UNIX operating system would have violated the established selection criteria.

3.5.1.2 The DOS environment

ATBMS implementation under DOS requires the simultaneously successful operation of the operating system and three support software elements. These are the compiler, the plotting library, and the DOS memory extender. The memory extender can be viewed as a memory driver that facilitates access to up to 4 Gb of installed memory, not only the first 640 Kb.

One support software collection included the *Phar Lap 386|ASM/LINK* memory extender, the *NDP Plot* plotter library, and the *NDP FORTRAN-386* compiler. The memory extender performed satisfactorily, while the plotting library failed to support its own demonstration example as documented. The significant deficiencies in the compiler included incorrect handling of real constants and incorrect handling of reserved keywords. Labelled COMMON is not checked for length, issues no diagnostic warnings, but misaligns COMMONs during execution. This software collection was deemed unacceptable for the ATBMS implementation. The software which satisfactorily supported the conversion effort is identified below.

3.5.2 Recommended Support Software

The *Lahey* compiler-linker, with the *AI Architects'* DOS extender, and *Plotworks* library, reference [2], were found to provide the necessary ATBMS implementation environment.

The DOS limitation on the number of simultaneously open files is eliminated by the *Lahey DOS Interface Library*, reference [3].

4. ATBMS IMPLEMENTATION

4.1 INTRODUCTION

This section provides the details of the ATBMS implementation in terms of support software organization requirements, compilation and linking procedures, and specific program modifications. The procedures for executing the individual programs are given. Examples of outputs and modified source code listings are provided in appendices.

It is assumed that the reader of this section is proficient in using Intel microprocessor based personal computers under the disk operating system.

4.2 SYSTEM CONFIGURATION

4.2.1 Hardware Configuration

The hardware configuration specified in Table 3-1 is assumed.

4.2.2 Support Software Configuration

The work presented in this report was carried out under MS-DOS versions 3.3 or 4.01.

The support software used -- compiler, plotting library, and DOS extender -- are identified in Section 5, reference [2].

4.2.3 Hard Disk Organization

The disk organization presented in this section was used to compile, link, and execute the ATBMS programs. While this is not the only possible organization, it follows a set of conventions that was used throughout the project. The conventions may be changed, but need to be kept consistent.

4.2.3.1 The boot directory

The boot drive root directory, C:\, contains the user's AUTOEXEC.BAT and CONFIG.SYS files. The PATH established in the AUTOEXEC.BAT file must contain at least the following elements:

```
PATH=C:\;C:\DOS;D:\;D:\F77L3;D:\OS386;D:\PLOT88;D:\ATBM
```

4.2.3.2 The working drive

The root directory of the working drive, D:\, must contain the AI DOS extender configuration file, CONFIG.AIA.

4.2.3.3 The compiler directory

The Lahey FORTRAN-77 compiler is installed to directory D:\F77L3.

4.2.3.4 The DOS extender directory

The DOS extender directory, D:\OS386, contains the files installed for A.I. Architects' OS/386. The DOS extender is a subset of the A.I. Developers Kit OS/386, and it may be packaged with the Lahey compiler. The complete Developers Kit is not required.

4.2.3.5 The graphics directory

The plotting library is placed in directory D:\PLOT88.

4.2.3.6 The working directory

The working directory employs the following naming conventions:

- .BAT are batch files, used for compilation and linking
- .FTN are FORTRAN source files
- .EXP are executable files generated by the compiler
- .DIR are directory files used for file handling by GEBOD, ATB, and VIEW discussed in Section 4.2.5

Also included in the working directory are:

- input files required for execution by GEBOD, ATB, and VIEW, specified in the corresponding .DIR file
- output files generated during program execution
- RPCSLAVE.EXE supplied with PLOT88, required for execution.

The object code, map, and listing files may be deleted or kept, at the discretion of the user.

4.2.4 Compiling and Linking FORTRAN

The executable program, .EXP, is generated from the source code by

the compiler and linker programs. Two batch files have been prepared, their use depending on the presence or absence of graphics calls in the source code. In either case, compilation listings are generated with the same name as the source code file, using a .C extension. Following successful compilation, the batch files proceed by linking object modules, using the graphics libraries, as appropriate.

The GEBOD program, which does not utilize graphics, is compiled and linked by CL.BAT:

```
F77L3 %1.FTN %2 /7/nS > %1.C
UP L32 %1
```

The compilation/linking is invoked as:

```
CL GEBODII
```

The ATB and VIEW programs, both utilizing graphics, are compiled by CLL.BAT:

```
F77L3 %1.FTN %2 /7/nS > %1.C
UP L32 %1,,,\PLOT88\PLOT88+\PLOT88\DRIVE88+\F77L3\F77L3
```

The compilation/linking is invoked as:

```
CLL ATBIV or CLL VIEWPE
```

Additional compiler options, such as debug, cross reference listings, etc. may be added as a second element (without embedded blanks) following the program name for either batch file.

The ATB program source code, due to its large size, is usually transferred on floppy disks as four files. This also facilitates code modifications, since most conventional text editors will not process very large text files. The compilation and linking procedure (CLL.BAT) is, however, for a single source file. The user has the option of either developing multiple file compilation and linking procedures, or combining the four ATB source code files prior to compilation with the ACOPY.BAT procedure:

```
COPY ATBIV.FT1+ATBIV.FT2+ATBIV.FT3+ATBIV.FT4 ATBIV.FTN
```

Similarly, the large data file which is input to GEBOD, is stored in two segments on floppy disks. The segments are combined by the GCOPY.BAT procedure:

COPY GEBOD.DT1+GEBOD.DT2 GEBOD.DAT

4.2.5 ATBMS DIRECTORY Files

The purpose of the DIRECTORY files in the 386-ATBMS implementation is to identify the input files, provide names for the output files, and initialize hardware devices -- displays, printers, and plotters. The alternative of having fixed file names and hardware devices embedded in the FORTRAN source code would have been unacceptably inflexible.

The file name records in the directory file follow the format:

- field 1: characters 1-12, left justified, valid DOS filename, required
- field 2: characters 15-21, FORTRAN logical unit designator, optional, recommended
- field 3: characters 25-80, file use description or comments, optional, recommended

The device initialization records in the directory file follow the following format:

- field 1: characters 1-4, right justified, numeric, PLOT88 PLOTS parameter value, required for device parameter initialization
- field 2: characters 7-12, alphanumeric, PLOTS parameter name, optional, recommended
- field 3: characters 15-21, alphanumeric, FORTRAN logical unit designator (when used), optional, recommended
- field 4: characters 25-80, alphanumeric, comments, optional, recommended

The following are the general rules for all DIRECTORY files:

- (1) The names of the DIRECTORY files may not be changed, they are GEBODII.DIR, ATBIV.DIR, and VIEWPE.DIR, respectively.
- (2) The SEQUENCE of records in the DIRECTORY files is critical. Records MUST NOT be deleted! Records MUST NOT be interchanged!
- (3) All output files are conservatively opened with STATUS=NEW. This prevents these files from being overwritten. To obtain a second or subsequent execution, the analyst has the following options, one of which MUST be exercised:

- (a) delete the output files,
- (b) rename the output files,
- (c) move the output files to a different DOS directory,
- (d) copy the output files to a different DOS directory, then delete the originals, or
- (e) edit the DIRECTORY file, assigning new names to the output files.

The DIRECTORY files specific to each program are described in Section 4.3.

4.2.6 Executing Programs

An executable file (.EXP) is produced by the compilation and link steps for each ATBMS program. Each program has its corresponding DIRECTORY (.DIR) file which identifies input/output file names and hardware device parameters. The .DIR file is prepared or modified by using any ASCII text editor. With the executable (.EXP), directory (.DIR), and input files in the current working directory, the programs are executed by the command

UP GEBODII, UP ATBIV, or UP VIEWPE

4.3 PROGRAM IMPLEMENTATION

The specific changes incorporated into each program, their DIRECTORY files, and sample outputs for each are described in the balance of this section. All source code modifications are identified by the string 80386 starting in character position 73.

4.3.1 The GEBOD Program

The modified GEBOD program (reference [4]) source code is provided in Appendix H.

4.3.1.1 Source code modifications

The following source code changes were made for the 386-GEBOD implementation.

- (1) Changes to COMMONs, to achieve equal length across all subprograms
 - (a) /DIMS/ in MAIN, CONTAC, SEGMAS, TORSO, FEET, RESULTS, CNVERT

- (b) /FLOAT/ in MAIN, CONTAC, SGINER, TORSO, FEET
- (c) /NAMES/ in BLOCK DATA, PFILE, RESULTS, RESULTZ
- (d) /JNTS/ in CONTAC, SEGMAS, CNVRT
- (e) /SGMNTS/ in SGINER

(2) Accessed the directory file GEBODII.DIR and opened the appropriate files in MAIN and DIALOG.

(3) Reduced real constant exponent from E+74 to E+38 in subprogram NDTRI.

(4) Modified EXTERNAL calls to ELLIP in subprogram SEGMAS,

(5) Modified EXTERNAL calls to ELLPMI in subprogram SGINER, and

(6) Modified ABEVAL calls in subprograms ELLIP and ELLPMI. Changes 4 through 6 were necessary since the present version of the compiler does not support the EXTERNAL statement.

(7) Removed case sensitivity from user responses in subprograms DIALOG, RESULTS, and RESULTZ.

4.3.1.2 The GEBODII.DIR file

The general rules discussed in Section 4.2.5 apply to GEBODII.DIR.

A sample if this directory file is shown below.

GEBODII.DIR

GEBOD.DAT	UNIT 2	GEBOD INPUT
GEBOD.AIN	UNIT 3	GEBOD BODY DESCRIPTION OUTPUT, INPUT TO ATB
GEBOD.TAB	UNIT 9	GEBOD BODY DESCRIPTION TABULAR OUTPUT
GEBODIN.USR	UNIT 1	USER-SUPPLIED GEBOD INPUT, 4(8F10.3) RECORDS

Comments:

(a) The body description tabular output file (unit 9) contains carriage control characters and may be printed directly.

(b) The user supplied input file (unit 1) is opened only if the appropriate option is selected during program execution. If this file is not used, then the corresponding field may be left blank, but the record must be present.

4.3.1.3 Sample outputs

The GEBOD program outputs, which is the body description in tabular form (GEBOD.TAB) and in ATB input form (GEBOD.AIN), are shown in Appendices A and B, respectively.

4.3.2 The ATB Program

The modified ATB program (reference [5]) source code is provided in Appendix I.

4.3.2.1 Source code modifications

The following source code changes were made for the 386-ATB implementation.

(1) Changes to COMMONs, to achieve equal length across all subprograms

(a) /CDINT/ in BLOCK DATA, ADJUST, CMPUTE, DINT, HICCSI, POSTPR, RSTART, TRIGFS.

(b) /TEMPVS/ in

BLOCK DATA,	AIRBAG,	AIRBGG,	AIRBG1,	AIRBG3,
BELTG,	BELTRT,	BGG,	BINPUT,	BLKDTA,
CHAIN,	CINPUT,	DAUX,	DAUX11,	DAUX12,
DAUX22,	DAUX31,	DAUX32,	DAUX33,	DAUX44,
DAUX55,	DRIFT,	EJOINT,	EQUILB,	FDINIT,
FINPUT,	FLXSEG,	HBELT,	HBPLAY,	HEDING,
HINPUT,	HPTURB,	HSETC,	HYEST,	HYLPX,
HYNTR,	IMPLS2,	INITAL,	KINPUT,	OUTPUT,
PDAUX,	PLEDG,	PLELP,	PLSEGF,	PLTXYZ,
POSTPR,	PRINT,	PRIPLT,	ROTATE,	SEGSEG,
SETUP1,	SETUP2,	SINPUT,	SPDAMP,	UNIT1,
VINPUT,	VISPR,	WINDY.		

(c) /TITLES/ in KINPUT.

(d) /CNSNTS/ in POSTPR.

(2) Accessed the directory file ATBIV.DIR and opened the appropriate files in MAIN.

(3) Removed Perkin-Elmer CARCON command in subprograms MAIN, HEDING, and OUTPUT.

(4) Modified system clock calls in subprogram ELTIME.

(5) Modified PLOTS call in subprogram POSTPR, and

(6) In subprogram SLPLOT modified the declaration of COMMON /TEMPVS/ variables passed as arguments from subprogram POSTPR. Changes 5 and 6 were necessary for compatibility with the PLOT88 package.

(7) Modified SYMBOL call in subprogram SLPLOT.

4.3.2.2 The ATBIV.DIR file

The general rules discussed in Section 4.2.5 apply to GEBODII.DIR.

A sample of this directory file is shown below.

ATBIV.DIR

ATB.TP1	UNIT 1	ATB OUTPUT, VIEW BODY ELEMENT AND CONTACT PLANE INPUT
ATB.PRP	UNIT 2	ATB PRINTER PLOTS
ATB.ROU	UNIT 3	ATB RESTART RUN OUTPUT FILE
ATB.RIN	UNIT 4	ATB RESTART RUN INPUT FILE
ATB.AIN	UNIT 5	ATB INPUT, GEBOD BODY DESCRIPTION OUTPUT
ATB.AOU	UNIT 6	ATB PRIMARY OUTPUT FILE
ATB.TP8	UNIT 8	ATB SUBROUTINE POSTPR OUTPUT FILE
0 IDEF	UNIT 10	DRAWING OPTION SEE PLOT88, 3.1
91 IOPORT	UNIT 10	HARDWARE INTERFACE 91/ 1 SEE PLOT88, 3.1
91 MODEL	UNIT 10	OUTPUT DEVICE 91/64 SEE PLOT88, 3.1
ATB.T24	21 - 85	LAST ATB TABULAR TIME HISTORY OUTPUT FILE (21-85)

Comments:

(a) The ATB printer plots, primary output file, and tabular time history files (units 2, 6, and 21-85) contain carriage control characters and may be printed directly.

(b) Optional files are not OPENed during execution unless the corresponding features have been selected in the input file. When optional files are not used, their name fields in the .DIR file may be left blank. However, the record position in the file must be retained! The following files are optional during ATB execution:

<u>UNIT</u>	<u>APPLICATION</u>
1	VIEW input
2	printer plots
3	RESTART run output
4	RESTART run input
8	POSTPR output
10	X-Y plots
21-85	tabular time history output

(c) When tabular time history output files (units 21-85) are selected, the last ATB directory entry must contain a valid DOS file name, with a three character extension. The second and third characters of the extension are used to specify the last logical output unit (21-85) used for tabular time history data. As an example, if tabular time history data are to be output on logical units 21 through 46, then the last entry should read ccccccc.c46, where

"c" represents any legal DOS file name character.

4.3.2.3 Sample outputs

Two sample cases, provided by AAMRL/BBM were used to test the 386-ATB implementation, and to develop the performance statistics presented in Table 2-1. Appendix C presents the inputs and outputs for the Basic Sled Test Simulation, with C1 and C2 listing the input and directory files, and Appendix C3 showing the 386 output. To facilitate comparison between the 386-ATB results and those obtained on the Perkin-Elmer, Appendix C4, showing the Perkin-Elmer results is provided. Similar appendices (D1-D4) are used for presenting example 2, the Dynamic Joint Test.

The input file for example 2 was modified to demonstrate both the X-Y plot and the printer plot capability of the 386-ATB. Appendix E1 shows the modified input file, while samples of the resulting plots are in appendices E2 and E3, respectively. The output is identical to the one shown in Appendix D3, with an increase in execution time to 43 seconds, due to the plotting options.

4.3.3 The VIEW Program

The modified VIEW program (reference [6]) source code is provided in Appendix J.

4.3.3.1 Source code modifications

The following source code changes were made for the 386-VIEW implementation.

- (1) Changes to COMMONs, to achieve equal length across all subprograms
 - (a) /PLTT/ in EXTEND, PLPLN, PNTPLT.
 - (b) /DEBUG/ in INPUT.
- (2) Modified hexadecimal constant assignments in subprogram NFRAME.
- (3) Accessed the directory file VIEWPE.DIR and opened the appropriate files in MAIN.
- (4) Moved code blocks following the statement numbers 200 and 400 within the DO-100 loop in subprogram PNTPLT to avoid external transfer into the DO-loop range.
- (5) Changed NEWPEN calls to COLOR calls in subprograms MAIN, PLPLN, and

TITLE for compatibility with the PLOT88 package.

(6) Modified the MAIN program to ignore device flag, DEVFLG, specified on record 1 of the control data input file. The device flag is now set to 4, for a multicolor graphics terminal.

(7) Added a scale factor in the MAIN program to allow display of the 20-character title on the graphics screen.

(8) Modified calls to SYMBOL and NUMBER in program MAIN to display frame titles on the graphics screen.

(9) Inserted a test on variable TEMP in subprogram ELIPSN. This test is needed due to differences in precision between computers.

(10) Replaced PLOTS call with PLOT call in subprogram NFRAME for compatibility with the PLOT88 package.

4.3.3.2 The VIEWPE.DIR file

The general rules discussed in Section 4.2.5 apply to VIEWPE.DIR. The output listing file (unit 6) contains carriage control characters and may be directly printed. A sample of this directory file is shown below.

VIEWPE.DIR

VIEW.TP1	UNIT 1	VIEW BODY ELEMENT AND CONTACT PLANE INPUT, ATB OUTPUT
VIEW.VIN	UNIT 5	VIEW CONTROL DATA INPUT
VIEW.VOU	UNIT 6	VIEW LISTING OUTPUT
0 IDEF		DRAWING OPTION SEE PLOT88, 3.1
91 IOPORT		HARDWARE INTERFACE 91/ 1 SEE PLOT88, 3.1
91 MODEL		OUTPUT DEVICE 91/64 SEE PLOT88, 3.1

4.3.3.3 Sample outputs

The two samples used for testing the 386-VIEW program implementation are ATB course examples 1 and 2. The input files, the output listings, and the resulting graphics are shown in Appendices F and G, respectively.

5. REFERENCES

1. Microport System V/386 Runtime System, Version 3.0e
LPI-FORTRAN for Intel 80386 UNIX System V, Version 3.02
2. Lahey compiler-linker F77L-EM/32
 Lahey Computer Systems, Inc., Incline Village, NV
AI Architects' DOS extender OS/386
 Lahey Computer Systems, Inc., Incline Village, NV
Plotworks PLOT88 library
 Plotworks Inc., Ramona, CA
3. Lahey Spindrift Utility Library (for F77L-EM/32)
4. Baughman, L.D., *Development of an Interactive Computer Program to Produce Body Description Data*, Technical Report AFAMRL-TR-83-058, WPAFB, OH, July 1983.
5. Obergefell, L.A., Fleck, J.T., Kaleps, I., and Gardner, T.R., *Articulated Total Body Model Enhancements*, Three volume Technical Report AAMRL-TR-88-007, -009, -043, WPAFB, OH, January, 1988.
6. Leetch, B.D. and Bowman, W.L., *Articulated Total Body (ATB) "VIEW" Program Software Report, Parts I and II*, Technical Report AFAMRL-TR-81-111, WPAFB, OH, June 1983.
7. D'Aulerio, L.A., Frisch, G.D., and Fender, D.A., *Ejection Test Data Analysis Package (ETDAP) User's Guide*, Report No. NADC-84133-60, Warminster, PA, October 1984.
8. Hyslop, R.L., *Horizontal Accelerator Data Acquisition System Enhancements*, KETRON Technical Report 4658-25, Warminster, PA, March 1989.

APPENDIX A

GEBOD Body Description Tabular Output (GEBOD.TAB)

01/10/90
 ADULT HUMAN MALE
 SELECTED BODY DIMENSIONS

COMPUTED BODY DIMENSIONS		75.00	X-TILE
0	WEIGHT	188.0	LB.
1	STANDING HEIGHT	70.66	IN.
2	SHOULDER HEIGHT	58.01	IN.
3	ARMPIT HEIGHT	51.78	IN.
4	WAIST HEIGHT	42.44	IN.
5	SEATED HEIGHT	37.07	IN.
6	HEAD LENGTH	7.869	IN.
7	HEAD BREADTH	6.186	IN.
8	HEAD TO CHIN HEIGHT	9.011	IN.
9	NECK CIRCUMFERENCE	15.44	IN.
10	SHOULDER BREADTH	16.27	IN.
11	CHEST DEPTH	10.04	IN.
12	CHEST BREADTH	13.33	IN.
13	WAIST DEPTH	9.216	IN.
14	WAIST BREADTH	12.73	IN.
15	BUTTOCK DEPTH	9.879	IN.
16	HIP BREADTH, STANDING	14.29	IN.
17	SHOULDER TO ELBOW LENGTH	14.33	IN.
18	FOREARM-HAND LENGTH	19.68	IN.
19	BICEPS CIRCUMFERENCE	12.89	IN.
20	ELBOW CIRCUMFERENCE	12.57	IN.
21	FOREARM CIRCUMFERENCE	11.39	IN.
22	WRIST CIRCUMFERENCE	7.067	IN.
23	KNEE HEIGHT, SEATED	22.31	IN.
24	THIGH CIRCUMFERENCE	24.16	IN.
25	UPPER LEG CIRCUMFERENCE	15.69	IN.
26	KNEE CIRCUMFERENCE	15.95	IN.
27	CALF CIRCUMFERENCE	15.11	IN.
28	ANKLE CIRCUMFERENCE	9.054	IN.
29	ANKLE HEIGHT, OUTSIDE	5.462	IN.
30	FOOT BREADTH	3.904	IN.
31	FOOT LENGTH	10.79	IN.

WEIGHT CORRECTION FACTOR = 1.099

01/10/90
CRASH VICTIM PARAMETERS (3-D)

SEGMENT NO. SYM PLOT	WEIGHT (LB.)	SEGMENT MOMENT OF INERTIA (LB-SEC**2-IN)			SEGMENT CONTACT ELLIPSOID SEMIAXIS (IN)			CENTER (IN)			
		X	Y	Z	X	Y	Z	X	Y	Z	
1	LT 1	31.18	1.34822	0.83419	1.47922	4.94	7.14	3.65	0.00	0.00	-0.07
2	CT 2	11.63	0.33664	0.18749	0.47497	4.61	6.36	4.01	0.00	0.00	-0.01
3	UT 3	48.11	2.84603	2.29042	1.94681	5.02	6.66	7.01	0.00	0.00	-0.83
4	N 4	3.06	0.02435	0.07435	0.01918	2.46	2.46	3.05	0.00	0.00	0.00
5	H 5	11.61	0.25522	0.79078	0.15060	3.93	3.09	5.73	0.00	0.00	0.00
6	RUL 6	20.12	1.61197	1.61197	0.20966	3.17	3.17	12.03	0.00	0.00	0.00
7	RLL 7	8.80	0.40723	6.0723	0.05271	2.41	2.41	9.14	0.00	0.00	0.00
8	RF 8	1.89	0.03583	0.03461	0.00470	2.73	1.95	5.40	0.00	0.00	1.40
9	LUL 9	20.12	1.61197	1.61197	0.20966	3.17	3.17	12.03	0.00	0.00	0.00
10	LLL A	8.80	0.40723	0.40723	0.05271	2.41	2.41	9.14	0.00	0.00	0.00
11	LF B	1.89	0.03583	0.03461	0.00470	2.73	1.95	5.40	0.00	0.00	1.40
12	RUA C	5.01	0.14425	0.14425	0.02185	2.05	2.05	7.17	0.00	0.00	0.00
13	RLA D	5.38	0.27886	0.27886	0.01830	1.81	1.81	9.84	0.00	0.00	0.00
14	LUA E	5.01	0.14425	0.14425	0.02185	2.05	2.05	7.17	0.00	0.00	0.00
15	LLA F	5.38	0.27886	0.27886	0.01830	1.81	1.81	9.84	0.00	0.00	0.00

JOINT J SYM PLOT JMT PIN	LOCATION(IN) - SEG(JMT)			LOCATION(IN) - SEG(J+1)			PRIN. AXIS(DEG) - SEG(JMT)			PRIN. AXIS(DEG) - SEG(J+1)		
	X	Y	Z	X	Y	Z	YAW	PITCH	ROLL	YAW	PITCH	ROLL
1	P M 1 0	0.00	0.00	-3.72	0.00	0.00	1.55	0.00	0.00	0.00	0.00	0.00
2	W N 2 0	0.00	0.00	-1.57	0.00	0.00	6.18	0.00	0.00	0.00	0.00	0.00
3	MP O 3 0	0.00	0.00	-7.83	0.00	0.00	0.59	0.00	0.00	0.00	20.00	0.00
4	HP P 4 0	0.00	0.00	-0.59	0.00	0.00	5.73	0.00	0.00	0.00	0.00	0.00
5	RH Q 1 0	0.00	3.30	-0.27	0.00	0.00	-8.18	0.00	0.00	8.10	-44.40	14.50
6	RK R 6 1	0.00	0.00	9.49	0.00	0.00	-6.60	0.00	0.00	0.00	65.60	-3.30
7	RA S 7 0	0.00	0.00	7.70	2.73	0.00	-2.55	0.00	0.00	-8.10	-80.00	0.00
8	LH T 1 0	0.00	-3.30	-0.27	0.00	0.00	-8.18	0.00	0.00	-8.10	-44.40	-14.50
9	LK U 9 1	0.00	0.00	9.49	0.00	0.00	-6.60	0.00	0.00	0.00	65.60	3.30
10	LA V 10 0	0.00	0.00	7.70	2.73	0.00	-2.55	0.00	0.00	8.10	-80.00	0.00
11	RS W 3 0	0.00	6.08	-5.00	0.00	0.00	-5.11	0.00	0.00	28.80	-39.60	63.30
12	RE X 12 1	0.00	0.00	5.16	0.00	0.00	-7.84	0.00	0.00	80.00	-70.00	0.00
13	LS Y 3 0	0.00	-6.08	-5.00	0.00	0.00	-5.11	0.00	0.00	-28.80	-39.60	-63.30
14	LE Z 14 1	0.00	0.00	5.16	0.00	0.00	-7.84	0.00	0.00	-80.00	-70.00	0.00

JOINTS RA, LA, RE, & LE ARE ORDERED PITCH, YAW, ROLL

APPENDIX B

GEBOD Body Description File Output (GEBOD.AIN)

15 14 01/10/90

LT 1	31.181	.34820	.83421	.4792	4.940	7.144	3.645	0.000	0.000	0.000	0.072	CARD B.1
CT 2	11.630	.33660	.18750	.4750	4.608	6.363	4.015	0.000	0.000	0.000	0.010	CARD B.2
UT 3	48.112	.84602	.29041	.9468	5.020	6.663	7.006	0.000	0.000	0.000	0.826	CARD B.2
M 4	3.060	.02430	.02430	.0192	2.458	2.458	3.049	0.000	0.000	0.000	0.000	CARD B.2
H 5	11.619	.25520	.29080	.1506	3.935	3.093	5.734	0.000	0.000	0.000	0.000	CARD B.2
RUL 6	20.121	.61201	.61200	.2097	3.172	3.172	2.026	0.000	0.000	0.000	0.000	CARD B.2
RL 7	8.800	.40720	.40720	.0527	2.405	2.405	9.143	0.000	0.000	0.000	0.000	CARD B.2
RF 8	1.890	.03580	.03460	.0047	2.731	1.952	5.395	0.000	0.000	0.000	1.401	CARD B.2
LUL 9	20.121	.61201	.61200	.2097	3.172	3.172	2.026	0.000	0.000	0.000	0.000	CARD B.2
LLL 8	8.800	.40720	.40720	.0527	2.405	2.405	9.143	0.000	0.000	0.000	0.000	CARD B.2
LF 9	1.890	.03580	.03460	.0047	2.731	1.952	5.395	0.000	0.000	0.000	1.401	CARD B.2
RUA 10	5.010	.14420	.14420	.0218	2.051	2.051	7.166	0.000	0.000	0.000	0.000	CARD B.2
RLA 11	5.380	.27890	.27890	.0183	1.813	1.813	9.841	0.000	0.000	0.000	0.000	CARD B.2
LUA 12	5.010	.14420	.14420	.0218	2.051	2.051	7.166	0.000	0.000	0.000	0.000	CARD B.2
LLA 13	5.380	.27890	.27890	.0183	1.813	1.813	9.841	0.000	0.000	0.000	0.000	CARD B.2
P M 14	0	0.00	0.00	-3.72	0.00	0.00	1.55					CARD B.3
		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00		0 0 0
W M 15	2	0	0.00	0.00	-1.57	0.00	0.00	6.18				CARD B.3
		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00		0 0 0
NP O 16	3	0	0.00	0.00	-7.83	0.00	0.00	0.59				CARD B.3
		0.00	0.00	0.00	0.00	0.00	20.00	0.00				0 0 0
HP P 17	4	0	0.00	0.00	-0.59	0.00	0.00	5.73				CARD B.3
		0.00	0.00	0.00	0.00	0.00	0.00	0.00				0 0 0
RH Q 18	1	0	0.00	3.30	-0.27	0.00	0.00	-8.18				CARD B.3
		0.00	0.00	0.00	0.00	8.10	-44.40	14.50				0 0 0
RK R 19	6	1	0.00	0.00	9.49	0.00	0.00	-6.60				CARD B.3
		0.00	0.00	0.00	0.00	0.00	65.60	-3.30				0 0 0
RA S 20	7	0	0.00	0.00	7.70	2.73	0.00	-2.55				CARD B.3
		0.00	0.00	0.00	0.00	-8.10	-80.00	0.00				2 3 1
LH T 21	1	0	0.00	-3.30	-0.27	0.00	0.00	-8.18				CARD B.3
		0.00	0.00	0.00	0.00	-8.10	-44.40	-14.50				0 0 0
LK U 22	9	1	0.00	0.00	9.49	0.00	0.00	-6.60				CARD B.3
		0.00	0.00	0.00	0.00	65.60	3.30					0 0 0
LA V 23	10	0	0.00	0.00	7.70	2.73	0.00	-2.55				CARD B.3
		0.00	0.00	0.00	0.00	8.10	-80.00	0.00				2 3 1
RS W 24	3	0	0.00	6.08	-5.00	0.00	0.00	-5.11				CARD B.3
		0.00	0.00	0.00	28.80	-39.60	63.30					0 0 0
RE X 25	12	1	0.00	0.00	5.16	0.00	0.00	-7.84				CARD B.3
		0.00	0.00	0.00	80.00	-70.00	0.00					2 3 1
LS Y 26	3	0	0.00	-6.08	-5.00	0.00	0.00	-5.11				CARD B.3
		0.00	0.00	0.00	-28.80	-39.60	-63.30					0 0 0
LE Z 27	14	1	0.00	0.00	5.16	0.00	0.00	-7.84				CARD B.3
		0.00	0.00	0.00	80.00	-70.00	0.00					2 3 1

APPENDIX C1

Example 1, ATB Basic Sled Test Simulation
Input File (EX1.AIN)

2 SEPT 1988 0 0 0.0

CARD A1A

EXAMPLE 1: BASIC SLED TEST SIMULATION
TWO BELT HARNESS WITH HYPERELLIPTOID FOR DASH BOARD

IN.	LB.	SEC.	0.0	0.0	386.088	0.0	0.0								CARD
4	40	0.002	0.0005	0.001	0.000625										A3
1	040	2 0											A4		
	15	14	95TH PERCENTILE MALE									A5			
LT	134.7721	.62801	.01011	.7954	5.168	7.436	3.778	0.000	0.000	-0.072	0	CARD B2A			
CT	213.0990	.41600	.23010	.5857	4.809	6.673	4.145	0.000	0.000	-0.011	0	CARD B2A			
UT	353.6733	.45252	.78322	.3509	5.220	6.906	7.314	0.000	0.000	-0.872	0	CARD B2A			
N	4	3.2890	.02780	.02780	.0216	2.520	2.520	3.156	0.000	0.000	0.000	0	CARD B2A		
H	511.9270	.27080	.30850	.1584	3.984	3.125	5.836	0.000	0.000	0.000	0	CARD B2A			
RUL	622.7252	.01302	.01300	.2576	3.308	3.308	12.652	0.000	0.000	0.000	0	CARD B2A			
RLL	7	9.7830	.49890	.49890	.0626	2.487	2.487	9.616	0.000	0.000	0.000	0	CARD B2A		
RF	8	2.0790	.04260	.04120	.0055	2.866	2.016	5.617	0.000	0.000	1.462	0	CARD B2A		
LUL	922.7252	.01302	.01300	.2576	3.308	3.308	12.652	0.000	0.000	0.000	0	CARD B2A			
LLL	A	9.7630	.49890	.49890	.0626	2.487	2.487	9.616	0.000	0.000	0.000	0	CARD B2A		
LF	B	2.0790	.04260	.04120	.0055	2.866	2.016	5.617	0.000	0.000	1.462	0	CARD B2A		
RUA	C	5.5420	.17430	.17430	.0259	2.122	2.122	7.497	0.000	0.000	0.000	0	CARD B2A		
RLA	D	5.9010	.33310	.33310	.0214	1.871	1.871	11.269	0.000	0.000	0.000	0	CARD B2A		
LUA	E	5.5420	.17430	.17430	.0259	2.122	2.122	7.497	0.000	0.000	0.000	0	CARD B2A		
LLA	F	5.9010	.33310	.33310	.0214	1.871	1.871	11.269	0.000	0.000	0.000	0	CARD B2A		
P	M	1	0	0.00	0.00	-3.85	0.00	0.00	1.61				CARD B3A		
				0.00	0.00	0.00	0.00	5.00	0.00						
W	N	2	0	0.00	0.00	-1.64	0.00	0.00	6.44				CARD B3A		
				0.00	0.00	0.00	0.00	5.00	0.00						
NP	O	3	0	0.00	0.00	-8.19	0.00	0.00	0.64				CARD B3A		
				0.00	0.00	0.00	0.00	10.00	0.00						
HP	P	4	0	0.00	0.00	-0.64	0.00	0.00	5.84				CARD B3A		
				0.00	0.00	0.00	0.00	10.00	0.00						
RH	Q	1	0	0.00	3.42	-0.31	0.00	0.00	-8.64				CARD B3A		
				0.00	0.00	0.00	0.00	-45.00	0.00						
RK	R	6	1	0.00	0.00	10.00	0.00	0.00	-6.97				CARD B3A		
				0.00	0.00	0.00	0.00	60.00	0.00						
RA	S	7	0	0.00	0.00	8.12	2.87	0.00	-2.66				CARD B3A		
				0.00	90.00	0.00	0.00	10.00	0.00						
LH	T	1	0	0.00	-3.42	-0.31	0.00	0.00	-8.64				CARD B3A		
				0.00	0.00	0.00	0.00	-45.00	0.00						
LK	U	9	1	0.00	0.00	10.00	0.00	0.00	-6.97				CARD B3A		
				0.00	0.00	0.00	0.00	60.00	0.00						
LA	V	10	0	0.00	0.00	8.12	2.87	0.00	-2.66				CARD B3A		
				0.00	90.00	0.00	0.00	10.00	0.00						
RS	W	3	0	0.00	6.24	-5.22	0.00	0.00	-5.37				CARD B3A		
				0.00	0.00	0.00	0.00	-4.10	0.00						
RE	X	12	1	0.00	0.00	5.42	0.00	0.00	-8.20				CARD B3A		
				0.00	0.00	0.00	0.00	-70.00	0.00						
LS	Y	3	C	0.00	-6.24	-5.22	0.00	0.00	-5.37				CARD B3A		
				0.00	0.00	0.00	0.00	-4.10	0.00						
LE	Z	14	1	0.00	0.00	5.42	0.00	0.00	-8.20				CARD B3A		
				0.00	0.00	0.00	0.00	-70.00	0.00						
.000	10.00	.00	0.70		20.000	.006	10.00	.00	0.70		5.000	CARD B4			
.000	10.00	.00	0.70		20.000	.000	10.00	.00	0.70		35.000	CARD B4			
.000	5.00	.00	0.70		25.000	.000	10.00	.00	0.70		35.00	CARD B4			
.000	5.00	.00	0.70		25.000	.000	10.00	.00	0.70		35.00	CARD B4			
.000	10.00	.00	0.70		70.000	.000	.800	.00	0.70		40.00	CARD B4			
.000	1.80	.00	0.70		60.000	.000	.000	.00	.00		.00	CARD B4			
.000	7.00	.00	0.70		35.000	.000	10.00	.00	0.70		26.00	CARD B4			
.000	10.00	.00	0.70		70.000	.000	.800	.00	0.70		40.000	CARD B4			
.000	1.80	.00	0.70		60.000	.000	.000	.00	.00		.00	CARD B4			
.000	7.00	.00	0.70		35.000	.000	10.00	.00	0.70		26.00	CARD B4			
.000	10.00	.00	0.70		122.500	.000	10.00	.00	0.70		65.00	CARD B4			
.000	1.80	.00	0.70		70.000	.000	.00	.00	.00		.00	CARD B4			
.000	10.00	.00	0.70		122.500	.000	10.00	.00	0.70		65.00	CARD B4			
.000	1.80	.00	0.70		70.000	.000	.00	.00	.00		.00	CARD B4			
0.1	0.0	30.0	0.0	0.0				0.0	0.0			CARD B5			
0.1	0.0	30.0	0.0	0.0				0.0	0.0			CARD B5			

APPENDIX C2

Example 1, ATB Basic Sled Test Simulation
386 Directory File (ATBIV1.DIR)

EXP1.TP1	UNIT 1	ATB OUTPUT, VIEW BODY ELEMENT AND CONTACT PLANE INPUT
EXP1.PRP	UNIT 2	ATB PRINTER PLOTS
ATB.ROU	UNIT 3	ATB RESTART RUN OUTPUT FILE
ATB.RIN	UNIT 4	ATB RESTART RUN INPUT FILE
EX1.AIN	UNIT 5	ATB INPUT, GEBOO BODY DESCRIPTION OUTPUT
EXP1.AOU	UNIT 6	ATB PRIMARY OUTPUT FILE
EXP1.TP8	UNIT 8	ATB SUBROUTINE POSTPR OUTPUT FILE
0 IDEF	UNIT 10	DRAWING OPTION SEE PLOTWORKS, PLOT88, 3.1
91 IOPORT	UNIT 10	HARDWARE INTERFACE 91/ 1 SEE PLOTWORKS, PLOT88, 3.1
91 MODEL	UNIT 10	OUTPUT DEVICE 91/64 SEE PLOTWORKS, PLOT88, 3.1
EXP1.T24	21 - 85	LAST ATB TABULAR TIME HISTORY OUTPUT FILE (21-85)

APPENDIX C3

Example 1, ATB Basic Sled Test Simulation
386 Output File (EXPl.AOU)

CARD 8.1

CARDS 8.2

SEGMENT I	SYM	PLOT	WEIGHT (LB.)	PRINCIPAL MOMENTS OF INERTIA (LB.-SEC.**2- IN.)			SEGMENT CONTACT ELLIPSOID SEMIAXES (IN.)			CENTER (IN.)			PRINCIPAL AXES (DEG)		
				X	Y	Z	X	Y	Z	X	Y	Z	YAW	PITCH	ROLL
1	LT	1	34.772	1.6280	1.0101	1.7954	5.168	7.436	3.778	0.000	0.000	-0.072	0.00	0.00	0.00
2	CT	2	13.099	0.4160	0.2301	0.5857	4.809	6.673	4.145	0.000	0.000	-0.011	0.00	0.00	0.00
3	UT	3	53.673	3.4525	2.7832	2.3509	5.220	6.906	7.314	0.000	0.000	-0.872	0.00	0.00	0.00
4	N	4	3.289	0.0278	0.0278	0.0216	2.520	2.520	3.156	0.000	0.000	0.000	0.00	0.00	0.00
5	H	5	11.927	0.2708	0.3085	0.1584	3.984	3.125	5.836	0.000	0.000	0.000	0.00	0.00	0.00
6	RUL	6	22.725	2.0130	2.0130	0.2576	3.308	3.308	12.652	0.000	0.000	0.000	0.00	0.00	0.00
7	RLL	7	9.763	0.4989	0.4989	0.0626	2.487	2.487	9.616	0.000	0.000	0.000	0.00	0.00	0.00
8	RF	8	2.079	0.0426	0.0412	0.0055	2.866	2.016	5.617	0.000	0.000	1.462	0.00	0.00	0.00
9	LUL	9	22.725	2.0130	2.0130	0.2576	3.308	3.308	12.652	0.000	0.000	0.000	0.00	0.00	0.00
10	LLL	A	9.763	0.4989	0.4989	0.0626	2.487	2.487	9.616	0.000	0.000	0.000	0.00	0.00	0.00
11	LF	B	2.079	0.0426	0.0412	0.0055	2.866	2.016	5.617	0.000	0.000	1.462	0.00	0.00	0.00
12	RUA	C	5.542	0.1743	0.1743	0.0259	2.122	2.122	7.497	0.000	0.000	0.000	0.00	0.00	0.00
13	RLA	D	5.901	0.3331	0.3331	0.0214	1.871	1.871	10.269	0.000	0.000	0.000	0.00	0.00	0.00
14	LUA	E	5.542	0.1743	0.1743	0.0259	2.122	2.122	7.497	0.000	0.000	0.000	0.00	0.00	0.00
15	LLA	F	5.901	0.3331	0.3331	0.0214	1.871	1.871	10.269	0.000	0.000	0.000	0.00	0.00	0.00

CARDS 8.3

JOINT J	SYM	PLOT	JNT	PIN	LOCATION(IN.) - SEG(JNT)			LOCATION(IN.) - SEG(J+1)			PRIN. AXIS(DEG) - SEG(JNT)			PRIN. AXIS(DEG) - SEG(J+1)		
					X	Y	Z	X	Y	Z	YAW	PITCH	ROLL	YAW	PITCH	ROLL
1	P	M	1	0	0.000	0.000	-3.850	0.000	0.000	1.610	0.00	0.00	0.00	0.00	5.00	0.00
2	W	N	2	0	0.000	0.000	-1.640	0.000	0.000	6.440	0.00	0.00	0.00	0.00	5.00	0.00
3	NP	O	3	0	0.000	0.000	-8.190	0.000	0.000	0.640	0.00	0.00	0.00	0.00	10.00	0.00
4	HP	P	4	0	0.000	0.000	-0.640	0.000	0.000	5.840	0.00	0.00	0.00	0.00	10.00	0.00
5	RH	Q	1	0	0.000	3.420	-0.310	0.000	0.000	-8.640	0.00	0.00	0.00	0.00	-45.00	0.00
6	RK	R	6	1	0.000	0.000	10.000	0.000	0.000	-6.970	0.00	0.00	0.00	0.00	60.00	0.00
7	RA	S	7	0	0.000	0.000	8.120	2.870	0.000	-2.660	0.00	90.00	0.00	0.00	10.00	0.00
8	LH	T	1	0	0.000	-3.420	-0.310	0.000	0.000	-8.640	0.00	0.00	0.00	0.00	-45.00	0.00
9	LK	U	9	1	0.000	0.000	10.000	0.000	0.000	-6.970	0.00	0.00	0.00	0.00	60.00	0.00
10	LA	V	10	0	0.000	0.000	8.120	2.870	0.000	-2.660	0.00	90.00	0.00	0.00	10.00	0.00
11	RS	W	3	0	0.000	6.240	-5.220	0.000	0.000	-5.370	0.00	0.00	0.00	0.00	-4.10	0.00
12	RE	X	12	1	0.000	0.000	5.420	0.000	0.000	-8.200	0.00	0.00	0.00	0.00	-70.00	0.00
13	LS	Y	3	0	0.000	-6.240	-5.220	0.000	0.000	-5.370	0.00	0.00	0.00	0.00	-4.10	0.00
14	LE	Z	14	1	0.000	0.000	5.420	0.000	0.000	-8.200	0.00	0.00	0.00	0.00	-70.00	0.00

FLEXURAL SPRING CHARACTERISTICS

TORSIONAL SPRING CHARACTERISTICS

JOINT	SPRING COEF. (IN. LB./DEG**J)			ENERGY DISSIPATION COEF.	JOINT STOP (DEG)	SPRING COEF. (IN. LB./DEG**J)			ENERGY DISSIPATION COEF.	JOINT STOP (DEG)
	LINEAR (J=1)	QUADRATIC (J=2)	CUBIC (J=3)			LINEAR (J=1)	QUADRATIC (J=2)	CUBIC (J=3)		
1 P	0.000	10.000	0.000	0.700	20.000	0.000	10.000	0.000	0.700	5.000
2 W	0.000	10.000	0.000	0.700	20.000	0.000	10.000	0.000	0.700	35.000
3 NP	0.000	5.000	0.000	0.700	25.000	0.000	10.000	0.000	0.700	35.000
4 HP	0.000	5.000	0.000	0.700	25.000	0.000	10.000	0.000	0.700	35.000
5 RH	0.000	10.000	0.000	0.700	70.000	0.000	0.800	0.000	0.700	40.000
6 RK	0.000	1.800	0.000	0.700	60.000	0.000	0.000	0.000	0.000	0.000
7 RA	0.000	7.000	0.000	0.700	35.000	0.000	10.000	0.000	0.700	26.000
8 LH	0.000	10.000	0.000	0.700	70.000	0.000	0.800	0.000	0.700	40.000
9 LK	0.000	1.800	0.000	0.700	60.000	0.000	0.000	0.000	0.000	0.000
10 LA	0.000	7.000	0.000	0.700	35.000	0.000	10.000	0.000	0.700	26.000
11 RS	0.000	10.000	0.000	0.700	122.500	0.000	10.000	0.000	0.700	65.000
12 RE	0.000	1.800	0.000	0.700	70.000	0.000	0.000	0.000	0.000	0.000
13 LS	0.000	10.000	0.000	0.700	122.500	0.000	10.000	0.000	0.700	65.000
14 LE	0.000	1.800	0.000	0.700	70.000	0.000	0.000	0.000	0.000	0.000

CARDS 8.5

JOINT VISCOUS CHARACTERISTICS AND LOCK-UNLOCK CONDITIONS

JOINT	VISCOUS COEFFICIENT (IN. LB.SEC./DEG)	COULOMB FRICTION COEF. (IN. LB.)	FULL FRICTION ANGLE (DEG)	MAX TORQUE FOR A LOCKED JOINT (IN. LB.)	MIN TORQUE FOR UNLOCKED JOINT (IN. LB.)	MIN. ANG. VELOCITY FOR UNLOCKED JOINT (RAD/SEC.)	IMPULSE RESTITUTION COEFFICIENT
1 P	0.100	0.00	30.00	0.00	0.00	0.00	0.000
2 W	0.100	0.00	30.00	0.00	0.00	0.00	0.000
3 NP	0.100	0.00	30.00	0.00	0.00	0.00	0.000
4 HP	0.100	0.00	30.00	0.00	0.00	0.00	0.000
5 RH	0.100	0.00	30.00	0.00	0.00	0.00	0.000
6 RK	0.100	0.00	30.00	0.00	0.00	0.00	0.000
7 RA	0.100	0.00	30.00	0.00	0.00	0.00	0.000
8 LH	0.100	0.00	30.00	0.00	0.00	0.00	0.000
9 LK	0.100	0.00	30.00	0.00	0.00	0.00	0.000
10 LA	0.100	0.00	30.00	0.00	0.00	0.00	0.000
11 RS	0.100	0.00	30.00	0.00	0.00	0.00	0.000
12 RE	0.100	0.00	30.00	0.00	0.00	0.00	0.000
13 LS	0.100	0.00	30.00	0.00	0.00	0.00	0.000
14 LE	0.100	0.00	30.00	0.00	0.00	0.00	0.000

SEGMENT INTEGRATION CONVERGENCE TEST INPUT

SEGMENT NO. SYM	ANGULAR VELOCITIES (RAD/SEC.)			LINEAR VELOCITIES (IN./SEC.)			ANGULAR ACCELERATIONS (RAD/SEC.**2)			LINEAR ACCELERATIONS (IN./SEC.**2)		
	MAG. TEST	ABS. ERROR	REL. ERROR	MAG. TEST	ABS. ERROR	REL. ERROR	MAG. TEST	ABS. ERROR	REL. ERROR	MAG. TEST	ABS. ERROR	REL. ERROR
1 LT	0.010	0.010	0.0100	0.010	0.010	0.0100	0.100	0.100	0.1000	0.100	0.100	0.0100
2 CT	0.010	0.010	0.0100	0.000	0.000	0.0000	0.100	0.100	0.1000	0.000	0.000	0.0000
3 UT	0.010	0.010	0.0100	0.000	0.000	0.0000	0.100	0.100	0.1000	0.000	0.000	0.0000
4 N	0.010	0.010	0.0100	0.000	0.000	0.0000	0.100	0.100	0.1000	0.000	0.000	0.0000
5 H	0.010	0.010	0.0100	0.000	0.000	0.0000	0.100	0.100	0.1000	0.000	0.000	0.0000
6 RUL	0.010	0.010	0.0100	0.000	0.000	0.0000	0.100	0.100	0.1000	0.000	0.000	0.0000
7 RLL	0.010	0.010	0.0100	0.000	0.000	0.0000	0.100	0.100	0.1000	0.000	0.000	0.0000
8 RF	0.010	0.010	0.0100	0.000	0.000	0.0000	0.100	0.100	0.1000	0.000	0.000	0.0000
9 LUL	0.010	0.010	0.0100	0.000	0.000	0.0000	0.100	0.100	0.1000	0.000	0.000	0.0000
10 LLL	0.010	0.010	0.0100	0.000	0.000	0.0000	0.100	0.100	0.1000	0.000	0.000	0.0000
11 LF	0.010	0.010	0.0100	0.000	0.000	0.0000	0.100	0.100	0.1000	0.000	0.000	0.0000
12 RUA	0.010	0.010	0.0100	0.000	0.000	0.0000	0.100	0.100	0.1000	0.000	0.000	0.0000
13 RLA	0.010	0.010	0.0100	0.000	0.000	0.0000	0.100	0.100	0.1000	0.000	0.000	0.0000
14 LUA	0.010	0.010	0.0100	0.000	0.000	0.0000	0.100	0.100	0.1000	0.000	0.000	0.0000
15 LLA	0.010	0.010	0.0100	0.000	0.000	0.0000	0.100	0.100	0.1000	0.000	0.000	0.0000

VEHICLE DECELERATION INPUTS

SLED ACCELERATION - 20G PEAK

YAW	PITCH	ROLL	VIPS	VTIME	XO(X)	XO(Y)	XO(Z)	NATAB	ATO	ADT	MSEG
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	15	0.000000	0.010000	0

UNIDIRECTIONAL VEHICLE POSITION TABLES

TIME (MSEC)	ACC (G)	VELOCITY (IN./SEC.)	POSITION (IN.)	TIME (MSEC)	ACC (G)	VELOCITY (IN./SEC.)	POSITION (IN.)
0.00000	0.00	0.0000	0.00000				
10.00000	5.00	-9.6522	-0.03217				
20.00000	10.00	-38.6088	-0.25739				
30.00000	15.00	-86.8698	-0.86870				
40.00000	20.00	-154.4352	-2.05914				
50.00000	15.00	-222.0006	-3.95740				
60.00000	10.00	-270.2616	-6.43480				
70.00000	5.00	-299.2182	-9.29829				
80.00000	0.00	-308.8704	-12.35482				
90.00000	0.00	-308.8704	-15.44352				
100.00000	0.00	-308.8704	-18.53222				
110.00000	0.00	-308.8704	-21.62093				
120.00000	0.00	-308.8704	-24.70963				
130.00000	0.00	-308.8704	-27.79834				
140.00000	0.00	-308.8704	-30.88704				

NPI NBLT NBAG NHP NQ NSD NHRSS NWINDF NJTFF NFORCE
12 0 0 3 0 0 1 0 0 0

PAGE 6
CARD D.1
CARDS D.2

PLANE INPUTS

PLANE NO. 1 SEAT. 6 DEGREE OFF H

	X	Y	Z
POINT 1	10.0000	8.0000	-10.0000
POINT 2	28.0100	8.0000	-11.8900
POINT 3	10.0000	-8.0000	-10.0000

PLANE NO. 2 BACK PANEL. 13 DEGR

	X	Y	Z
POINT 1	1.0000	9.0000	-48.9700
POINT 2	10.0000	9.0000	-10.0000
POINT 3	1.0000	-9.0000	-48.9700

PLANE NO. 3 FLOOR.

	X	Y	Z
POINT 1	0.0000	12.0000	-1.3000
POINT 2	60.0000	12.0000	-1.3000
POINT 3	0.0000	-12.0000	-1.3000

PLANE NO. 4 HEAD PAD. 13 DEGR

	X	Y	Z
POINT 1	2.4800	7.5000	-47.2600
POINT 2	4.9600	7.5000	-36.5500
POINT 3	2.4800	-7.5000	-47.2600

PLANE NO. 5 SEAT FRONT PANEL.

	X	Y	Z
POINT 1	28.0100	8.0000	-11.8900
POINT 2	26.6600	8.0000	-4.4000
POINT 3	28.0100	-8.0000	-11.8900

PLANE NO. 6 BACK PANEL2. 13 DEGR

	X	Y	Z
POINT 1	1.0000	9.0000	-48.9700
POINT 2	10.0000	9.0000	-10.0000
POINT 3	1.0000	-9.0000	-48.9700

PLANE NO. 7 FIREWALL.

	X	Y	Z
POINT 1	60.0000	12.0000	-25.0000
POINT 2	60.0000	-12.0000	-25.0000
POINT 3	60.0000	12.0000	-0.7500

PLANE NO. 8 RIGHT SIDE SEAT/IN.

	X	Y	Z
POINT 1	8.4100	8.1000	-6.6600
POINT 2	8.7000	8.1000	-14.7300
POINT 3	30.5800	8.1000	-6.6400

PLANE NO. 9 LEFT SIDE SEAT/IN.

	X	Y	Z
POINT 1	8.4100	-8.1000	-6.6600
POINT 2	30.5800	-8.1000	-6.6400
POINT 3	8.7000	-8.1000	-14.7300

PLANE NO. 10 RUDDER PEDALS.

	X	Y	Z
POINT 1	49.9920	9.0000	-2.2392
POINT 2	52.9920	9.0000	-4.7565
POINT 3	49.9920	-9.0000	-2.2392

PLANE NO. 11 LEFT SIDE PANEL.

	X	Y	Z
POINT 1	1.0000	-9.0000	-48.9700
POINT 2	10.9000	-9.0000	-6.1000
POINT 3	-7.7700	-9.0000	-46.9500

PLANE NO. 12 RIGHT SIDE PANEL.

	X	Y	Z
POINT 1	1.0000	9.0000	-48.9700
POINT 2	-7.7700	9.0000	-46.9500
POINT 3	10.9000	9.0000	-6.1000

FUNCTION NO. 3 SEGMENT-SEGMENT FCN. NT1(3) = 1

D0	D1	D2	D3	D4
0.0000	-5.0000	0.0000	0.0000	1.0000

FIRST PART OF FUNCTION - 6 TABULAR POINTS

D	F(D)
0.000000	0.0000
1.000000	470.0000
2.000000	890.0000
3.000000	1220.0000
4.000000	1470.0000
5.000000	1580.0000

FUNCTION NO. 6 CONSTANT, F=0.0 NT1(6) = 19

CARDS E

D0	D1	D2	D3	D4
0.0000	0.0000	0.0000	0.0000	0.0000

FUNCTION IS CONSTANT 0.000000

FUNCTION NO. 7 R FACTOR.

NT1(7) = 24

D0	D1	D2	D3	D4
0.0000	0.0000	0.7000	0.0000	0.0000

FUNCTION IS CONSTANT 0.700000

FUNCTION NO. 13 STIFF SURFACES

NT1(13) = 29

CARDS E

D0	D1	D2	D3	D4
0.0000	-4.0000	0.0000	0.0000	1.0000

FIRST PART OF FUNCTION - 8 TABULAR POINTS

D	F(D)
0.000000	0.0000
0.100000	5.0000
0.200000	20.0000
0.300000	40.0000
0.400000	60.0000
1.000000	860.0000
2.000000	2400.0000
3.000000	4000.0000

FUNCTION NO. 14 FRICTION FUNC. NTI(14) = 51

D0	D1	D2	D3	D4
0.0000	0.0000	0.5000	0.0000	1.0000

FUNCTION IS CONSTANT 0.500000

FUNCTION NO. 19 CF=.25,CREST=.25 NTI(19) = 56

CARDS E

D0	D1	D2	D3	D4
0.0000	0.0000	0.2500	0.0000	0.0000

FUNCTION IS CONSTANT 0.250000

FUNCTION NO. 20 DAMPING COEFF. C=900 NTI(20) = 61

D0	D1	D2	D3	D4
0.0000	1.0000	0.0000	0.0000	1.0000

FIRST PART OF FUNCTION - 5TH DEGREE POLYNOMIAL

A0	A1	A2	A3	A4	A5
0.000000	900.000000	0.000000	0.000000	0.000000	0.000000

FUNCTION NO. 21 RATE OF DEFLEC. NTI(21) = 72

CARDS E

D0	D1	D2	D3	D4
-40.0000	-150.0000	0.0000	0.0000	1.0000

FIRST PART OF FUNCTION - 21 TABULAR POINTS

D	F(D)
-40.000000	0.0000
-30.000000	0.0000
-20.000000	0.0000
-10.000000	0.0000
0.000000	0.0000
5.000000	1.0000
10.000000	1.0000
20.000000	0.9900
30.000000	0.9650
40.000000	0.9280
50.000000	0.8600
60.000000	0.6900
70.000000	0.4750
80.000000	0.3400
90.000000	0.2600
100.000000	0.2000
110.000000	0.1800
120.000000	0.0900
130.000000	0.0600
140.000000	0.0250
150.000000	0.0000

FUNCTION NO. 22 DAMPING COEFF. C=35 NT1(22) = 120

D0	D1	D2	D3	D4
0.0000	1.0000	0.0000	0.0000	1.0000

FIRST PART OF FUNCTION - 5TH DEGREE POLYNOMIAL

A0	A1	A2	A3	A4	A5
0.000000	35.000000	0.000000	0.000000	0.000000	0.000000

FUNCTION NO. 24 DAMPING COEFF. C=0.8 NT1(24) = 131

CARDS E

D0	D1	D2	D3	D4
-1000.0000	-1000.0000	0.0000	0.0000	1.0000

FIRST PART OF FUNCTION - 4 TABULAR POINTS

D	F(D)
-1000.000000	0.6000
-1.000000	0.6000
0.000000	1.0000
1000.000000	1.0000

FUNCTION NO. 25 DAMPING COEFF C=1100 NTI(25) = 145

D0	D1	D2	D3	D4
0.0000	1.0000	0.0000	0.0000	1.0000

FIRST PART OF FUNCTION - 5TH DEGREE POLYNOMIAL

A0	A1	A2	A3	A4	A5
0.000000	1100.000000	0.000000	0.000000	0.000000	0.000000

FUNCTION NO. 26 STIFF SURFACES-LL NTI(26) = 156

CARDS E

D0	D1	D2	D3	D4
0.0000	-4.0000	0.0000	0.0000	1.0000

FIRST PART OF FUNCTION - 8 TABULAR POINTS

D	F(D)
0.000000	0.0000
0.100000	5.0000
0.200000	20.0000
0.300000	40.0000
0.400000	60.0000
2.000000	860.0000
3.000000	2400.0000
4.000000	4000.0000

FUNCTION NO. 31 HARNESS FDF

NT1(31) = 178

D0	D1	D2	D3	D4
0.0000	-4.0000	0.0000	0.0000	0.0000

FIRST PART OF FUNCTION - 8 TABULAR POINTS

D	F(D)
0.000000	0.0000
0.010000	150.0000
0.020000	300.0000
0.030000	450.0000
0.050000	850.0000
0.100000	3500.0000
1.000000	35000.0000
4.000000	140000.0000

FUNCTION NO. 32 HARNESS FRICTION

NT1(32) = 200

CARDS E

D0	D1	D2	D3	D4
0.0000	0.0000	0.2000	0.0000	0.2000

FUNCTION IS CONSTANT 0.200000

FUNCTION NO. 34 HARNESS FRICTION

NTI(34) = 205

D0	D1	D2	D3	D4
0.0000	0.0000	0.9000	0.0000	0.2000

FUNCTION IS CONSTANT 0.900000

ALLOWED CONTACTS AND ASSOCIATED FUNCTIONS

PLANE	SEGMENT	FORCE DEFLECTION	INERTIAL SPIKE	R FACTOR	G FACTOR	FRICTION COEF. OPT
1- 16	1- 1	13	-20	-21	0	14 1
SEAT. 6 DEGREE OFF H	LT	STIFF SURFACES	DAMPING COEFF. C=900 RATE OF DEFLEC.			FRICTION FUNC.
1- 16	6- 6	13	-25	-21	0	14 1
SEAT. 6 DEGREE OFF H	RUL	STIFF SURFACES	DAMPING COEFF C=1100 RATE OF DEFLEC.			FRICTION FUNC.
1- 16	9- 9	13	-25	-21	0	14 1
SEAT. 6 DEGREE OFF H	LUL	STIFF SURFACES	DAMPING COEFF C=1100 RATE OF DEFLEC.			FRICTION FUNC.
2- 16	1- 1	13	-20	-21	0	14 1
BACK PANEL. 13 DEGR	LT	STIFF SURFACES	DAMPING COEFF. C=900 RATE OF DEFLEC.			FRICTION FUNC.
2- 16	2- 2	13	-20	-21	0	14 1
BACK PANEL. 13 DEGR	CT	STIFF SURFACES	DAMPING COEFF. C=900 RATE OF DEFLEC.			FRICTION FUNC.
2- 16	3- 3	13	-20	-21	0	14 1
BACK PANEL. 13 DEGR	UT	STIFF SURFACES	DAMPING COEFF. C=900 RATE OF DEFLEC.			FRICTION FUNC.
3- 16	8- 8	13	-22	-21	0	14 -1
FLOOR.	RF	STIFF SURFACES	DAMPING COEFF. C=35 RATE OF DEFLEC.			FRICTION FUNC.
3- 16	11- 11	13	-22	-21	0	14 -1
FLOOR.	LF	STIFF SURFACES	DAMPING COEFF. C=35 RATE OF DEFLEC.			FRICTION FUNC.
4- 16	5- 5	13	-22	-21	0	14 1
HEAD PAD. 13 DEGR	H	STIFF SURFACES	DAMPING COEFF. C=35 RATE OF DEFLEC.			FRICTION FUNC.
10- 16	8- 8	13	-22	-21	0	14 1
RUDDER PEDALS.	RF	STIFF SURFACES	DAMPING COEFF. C=35 RATE OF DEFLEC.			FRICTION FUNC.
10- 16	11- 11	13	-22	-21	0	14 1
RUDDER PEDALS.	LF	STIFF SURFACES	DAMPING COEFF. C=35 RATE OF DEFLEC.			FRICTION FUNC.

SEGMENT	SEGMENT	FORCE DEFLECTION	INERTIAL SPIKE	R FACTOR	G FACTOR	FRICTION COEF. OPT
2- 2	13- 13	3	0	7	0	19 0
CT	RLA SEGMENT-SEGMENT FCN.			R FACTOR.		CF=.25,CREST=.25
2- 2	15- 15	3	0	7	0	19 0
CT	LLA SEGMENT-SEGMENT FCN.			R FACTOR.		CF=.25,CREST=.25
6- 6	13- 13	3	0	7	0	19 0
RUL	RLA SEGMENT-SEGMENT FCN.			R FACTOR.		CF=.25,CREST=.25
9- 9	15- 15	3	0	7	0	19 0
LUL	LLA SEGMENT-SEGMENT FCN.			R FACTOR.		CF=.25,CREST=.25
13- 13	16- 24	13	-22	-21	0	14 0
RLA	VEH	STIFF SURFACES	DAMPING COEFF. C=35 RATE OF DEFLEC.			FRICTION FUNC.
15- 15	16- 24	13	-22	-21	0	14 0
LLA	VEH	STIFF SURFACES	DAMPING COEFF. C=35 RATE OF DEFLEC.			FRICTION FUNC.

NO. OF HARNESSES = 1

NO. OF BELTS PER HARNES = 2

FOR HARNES NO. 1 NO. OF POINTS PER BELT = 12 15

HARNES NO. 1 BELT NO. 1 FUNCTION NOS. 31 0 0 0 0 REFERENCE SLACK = 0.000 IN.

K	KS	KE	NT	NPD	NDR	FUNCTION NOS.				
1	16	0	109	1	1	0	0	0	0	0
2	1	1	115	0	1	0	0	0	0	34
3	1	1	121	0	1	0	0	0	0	34
4	1	1	127	0	1	0	0	0	0	34
5	1	1	133	0	1	0	0	0	0	34
6	1	23	139	0	1	0	0	0	0	0
7	1	1	145	0	1	0	0	0	0	34
8	1	1	151	0	1	0	0	0	0	34
9	1	1	157	0	1	0	0	0	0	34
10	1	1	163	0	1	0	0	0	0	34
11	1	1	169	0	1	0	0	0	0	34
12	16	0	175	1	1	0	0	0	0	0

CARDS F.8.0

K	BASE REFERENCE (IN.)			ADJUSTED REFERENCE (IN.)			OFFSET (IN.)			PREFERRED DIRECTION (IN.)		
	X	Y	Z	X	Y	Z	X	Y	Z	X	Y	Z
1	13.000	8.000	-10.300	13.000	8.000	-10.300	0.000	0.000	0.000	2.400	22.000	-0.300
2	-1.178	7.075	-0.781	-1.178	7.075	-0.781	0.000	0.000	-0.072	0.000	0.000	0.000
3	-0.029	6.796	-1.534	-0.029	6.795	-1.534	0.000	0.000	-0.072	0.000	0.000	0.000
4	0.910	5.778	-2.283	0.910	5.778	-2.283	0.000	0.000	-0.072	0.000	0.000	0.000
5	2.228	2.355	-3.192	2.228	2.355	-3.192	0.000	0.000	-0.072	0.000	0.000	0.000
6	2.957	0.000	3.059	2.957	0.000	3.059	0.000	0.000	-7.000	0.000	0.000	0.000
7	3.070	-0.080	-4.410	2.344	-0.061	-3.367	0.000	0.000	-0.072	0.000	0.000	0.000
8	1.785	-2.325	-3.343	1.785	-2.325	-3.343	0.000	0.000	-0.072	0.000	0.000	0.000
9	0.011	-5.145	-2.728	0.011	-5.145	-2.728	0.000	0.000	-0.072	0.000	0.000	0.000
10	-0.880	-5.789	-2.282	-0.880	-5.789	-2.282	0.000	0.000	-0.072	0.000	0.000	0.000
11	-2.460	-6.099	-1.200	-2.460	-6.098	-1.200	0.000	0.000	-0.072	0.000	0.000	0.000
12	13.000	-8.000	-10.300	13.000	-8.000	-10.300	0.000	0.000	0.000	2.400	22.000	-0.300

HARNES NO. 1 BELT NO. 2 FUNCTION NOS. 31 0 0 0 0 REFERENCE SLACK = 0.000 IN.

K	KS	KE	NT	NPD	NDR	FUNCTION NOS.				
13	16	0	187	1	1	0	0	0	0	0
14	1	1	193	0	1	0	0	0	0	32
15	1	1	199	0	1	0	0	0	0	32
16	1	23	205	0	1	0	0	0	0	0
17	1	1	211	0	1	0	0	0	0	32
18	2	2	217	0	1	0	0	0	0	32
19	2	2	223	0	1	0	0	0	0	32
20	3	3	229	0	1	0	0	0	0	32
21	3	3	235	0	1	0	0	0	0	32
22	3	3	241	0	1	0	0	0	0	32
23	3	3	247	0	1	0	0	0	0	32
24	3	22	253	0	0	0	0	0	0	32
25	3	3	259	0	1	0	0	0	0	32
26	3	3	265	0	1	0	0	0	0	32
27	16	0	271	1	1	0	0	0	0	0

CARDS F.8.0

K	BASE REFERENCE (IN.)			ADJUSTED REFERENCE (IN.)			OFFSET (IN.)			PREFERRED DIRECTION (IN.)		
	X	Y	Z	X	Y	Z	X	Y	Z	X	Y	Z
13	13.000	-8.000	-10.300	13.000	-8.000	-10.300	0.000	0.000	0.000	0.700	17.500	-21.300
14	-2.445	-6.101	-1.213	-2.445	-6.101	-1.213	0.000	0.000	-0.072	0.000	0.000	0.000
15	-0.960	-6.000	-2.500	-0.906	-5.661	-2.359	0.000	0.000	-0.072	0.000	0.000	0.000
16	0.000	-5.700	3.800	0.000	-5.367	3.578	0.000	0.000	-7.000	0.000	0.000	0.000
17	0.010	-4.000	-4.500	0.008	-3.061	-3.443	0.000	0.000	-0.072	0.000	0.000	0.000
18	1.818	-5.439	-1.820	1.818	-5.439	-1.820	0.000	0.000	-0.011	0.000	0.000	0.000
19	2.500	-2.500	-1.500	3.397	-3.397	-2.038	0.000	0.000	-0.011	0.000	0.000	0.000

20	3.000	-1.500	6.500	2.777	-1.388	6.016	0.000	0.000	-0.872	0.000	0.000	0.000
21	4.487	-0.133	3.734	4.487	-0.133	3.734	0.000	0.000	-0.872	0.000	0.000	0.000
22	4.421	3.319	-1.662	4.421	3.319	-1.662	0.000	0.000	-0.872	0.000	0.000	0.000
23	0.879	4.202	-5.672	0.879	4.202	-5.672	0.000	0.000	-0.872	0.000	0.000	0.000
24	0.300	0.200	-2.800	0.320	0.213	-2.985	0.000	4.000	-3.500	0.000	0.000	0.000
25	-1.000	4.300	-6.000	-0.955	4.105	-5.728	0.000	0.000	-0.872	0.000	0.000	0.000
26	-2.500	4.300	-4.000	-2.612	4.492	-4.179	0.000	0.000	-0.872	0.000	0.000	0.000
27	0.000	5.000	-35.200	0.000	5.000	-35.200	0.000	0.000	0.000	0.700	17.500	-21.300

ZPLT(X) ZPLT(Y) ZPLT(Z) J1 J2 J3 SPLT(1) SPLT(2) SPLT(3)
0. 0. 0. 0 0 0 0 0 10.00 6.00 1.00

INITIAL POSITIONS (INERTIAL REFERENCE)

CARDS G.2

SEGMENT NO. SEG	LINEAR POSITION (IN.)			LINEAR VELOCITY (IN./SEC.)		
	X	Y	Z	X	Y	Z
1 LT	14.38100	0.00000	-13.75000	0.00000	0.00000	0.00000
2 CT	13.16068	0.00000	-19.07188	0.00000	0.00000	0.00000
3 UT	11.31383	0.00000	-26.93796	0.00000	0.00000	0.00000
4 N	9.28353	0.00000	-35.53137	0.00000	0.00000	0.00000
5 H	7.77521	0.00000	-41.83338	0.00000	0.00000	0.00000
6 RUL	22.94073	3.42000	-14.48930	0.00000	0.00000	0.00000
7 RLL	38.16022	3.42000	-10.39045	0.00000	0.00000	0.00000
8 RF	48.12719	3.42000	-4.45598	0.00000	0.00000	0.00000
9 LUL	22.94073	-3.42000	-14.48930	0.00000	0.00000	0.00000
10 LLL	38.16022	-3.42000	-10.39045	0.00000	0.00000	0.00000
11 LF	48.12719	-3.42000	-4.45598	0.00000	0.00000	0.00000
12 RUA	12.34164	6.24000	-27.13188	0.00000	0.00000	0.00000
13 RLA	22.75808	6.24000	-21.48521	0.00000	0.00000	0.00000
14 LUA	12.34164	-6.24000	-27.13188	0.00000	0.00000	0.00000
15 LLA	22.75808	-6.24000	-21.48521	0.00000	0.00000	0.00000

INITIAL ANGULAR ROTATION AND VELOCITY

CARDS G.3

SEGMENT NO. SEG	ANGULAR ROTATION (DEG)			ANGULAR VELOCITY (DEG/SEC.)			IYPR
	YAW	PITCH	ROLL	X	Y	Z	
1 LT	0.00000	12.90000	0.00000	0.00000	0.00000	0.00000	3 2 1 0
2 CT	0.00000	12.95000	0.00000	0.00000	0.00000	0.00000	3 2 1 0
3 UT	0.00000	13.28000	0.00000	0.00000	0.00000	0.00000	3 2 1 0
4 N	0.00000	13.46000	0.00000	0.00000	0.00000	0.00000	3 2 1 0
5 H	0.00000	13.46000	0.00000	0.00000	0.00000	0.00000	3 2 1 0
6 RUL	0.00000	92.90000	0.00000	0.00000	0.00000	0.00000	3 2 1 0
7 RLL	0.00000	48.65000	0.00000	0.00000	0.00000	0.00000	3 2 1 0
8 RF	0.00000	128.80000	0.00000	0.00000	0.00000	0.00000	3 2 1 0
9 LUL	0.00000	92.90000	0.00000	0.00000	0.00000	0.00000	3 2 1 0
10 LLL	0.00000	48.65000	0.00000	0.00000	0.00000	0.00000	3 2 1 0
11 LF	0.00000	128.80000	0.00000	0.00000	0.00000	0.00000	3 2 1 0
12 RUA	0.00000	24.50000	0.00000	0.00000	0.00000	0.00000	3 2 1 0
13 RLA	0.00000	85.00000	0.00000	0.00000	0.00000	0.00000	3 2 1 0
14 LUA	0.00000	24.50000	0.00000	0.00000	0.00000	0.00000	3 2 1 0
15 LLA	0.00000	85.00000	0.00000	0.00000	0.00000	0.00000	3 2 1 0

LINEAR AND ANGULAR VELOCITIES HAVE BEEN SET EQUAL TO THE INITIAL VEHICLE VELOCITIES.

HBPLAY TIME = 0.000 MSEC. NH,NB,NPTS NT= 1 1 11 103
 NL(1)= 1 2 3 4 5 6 8 9 10 11 12
 BB = 4.124 1.402 1.574 3.779 2.557 2.656 3.388 1.187 1.940 4.738
 HBPLAY TIME = 0.000 MSEC. NH,NB,NPTS NT= 1 2 7 181
 NL(1)= 13 14 18 21 22 23 27
 BB = 4.748 7.394 6.842 6.406 5.423 10.829

TABULAR TIME HISTORY CONTROL PARAMETERS

TYPE	KSG	SELECTED SEGMENTS OR JOINTS
H.1	3	3 -5 5
REF		0 1 16
H.2	3	3 5 5
REF		0 0 3
H.3	3	3 5 5
REF		0 0 3
H.4	3	3 5 5
REF		0 0 16
H.5	3	3 5 5
REF		0 0 5
H.6	3	3 5 5
REF		0 0 3
H.7	2	3 4
REF		0 0

H.8 0

REF

H.9 1 4

REF 5

H.10 15 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

REF 16

SEGMENT	(INERTIAL)			(LOCAL)			(LOCAL)		
	ANGULAR ROTATION (DEG)			ANGULAR VELOCITY (RAD/SEC.)			ANGULAR ACCELERATION (RAD/SEC.**2)		
	YAW	PITCH	ROLL	X	Y	Z	X	Y	Z
1 LT	0.0000	12.9000	0.0000	0.00000	0.00000	0.00000	0.000000	-26.834649	0.000000
2 CT	0.0000	12.9500	0.0000	0.00000	0.00000	0.00000	0.000000	40.304245	0.000000
3 UT	0.0000	13.2800	0.0000	0.00000	0.00000	0.00000	0.000000	-0.859459	0.000000
4 N	0.0000	13.4600	0.0000	0.00000	0.00000	0.00000	0.000000	13.637948	0.000000
5 H	0.0000	13.4600	0.0000	0.00000	0.00000	0.00000	0.000000	-1.165343	0.000000
6 RUL	0.0000	92.9000	0.0000	0.00000	0.00000	0.00000	0.000000	-2.405246	0.000000
7 RLL	0.0000	48.6500	0.0000	0.00000	0.00000	0.00000	0.000000	-12.215669	0.000000
8 RF	0.0000	128.8000	0.0000	0.00000	0.00000	0.00000	0.000000	56.318297	0.000000
9 LUL	0.0000	92.9000	0.0000	0.00000	0.00000	0.00000	0.000000	-2.405246	0.000000
10 LLL	0.0000	48.6500	0.0000	0.00000	0.00000	0.00000	0.000000	-12.215669	0.000000
11 LF	0.0000	128.8000	0.0000	0.00000	0.00000	0.00000	0.000000	56.318297	0.000000
12 RUA	0.0000	24.5000	0.0000	0.00000	0.00000	0.00000	0.000000	-6.917594	0.000000
13 RLA	0.0000	85.0000	0.0000	0.00000	0.00000	0.00000	0.000000	-34.722447	0.000000
14 LUA	0.0000	24.5000	0.0000	0.00000	0.00000	0.00000	0.000000	-6.917594	0.000000
15 LLA	0.0000	85.0000	0.0000	0.00000	0.00000	0.00000	0.000000	-34.722447	0.000000
16 VEH	0.0000	0.0000	0.0000	0.00000	0.00000	0.00000	0.000000	0.000000	0.000000

SEGMENT	(INERTIAL)			(INERTIAL)			(INERTIAL)		
	LINEAR POSITION (IN.)			LINEAR VELOCITY (IN./SEC.)			LINEAR ACCELERATIONS (G'S)		
	X	Y	Z	X	Y	Z	X	Y	Z
1 LT	14.3810	0.0000	-13.7500	0.00000	0.00000	0.00000	0.017822	0.000000	-0.088444
2 CT	13.1607	0.0000	-19.0719	0.00000	0.00000	0.00000	0.114863	0.000000	-0.110519
3 UT	11.3138	0.0000	-26.9380	0.00000	0.00000	0.00000	-0.038031	0.000000	-0.075446
4 N	9.2835	0.0000	-35.5314	0.00000	0.00000	0.00000	-0.042274	0.000000	-0.074372
5 H	7.7752	0.0000	-41.8334	0.00000	0.00000	0.00000	-0.047117	0.000000	-0.073212
6 RUL	22.9407	3.4200	-14.4893	0.00000	0.00000	0.00000	0.041548	0.000000	-0.039498
7 RLL	38.1602	3.4200	-10.3904	0.00000	0.00000	0.00000	-0.100994	0.000000	0.188268
8 RF	48.1272	3.4200	-4.4560	0.00000	0.00000	0.00000	-0.187590	0.000000	-0.183587
9 LUL	22.9407	-3.4200	-14.4893	0.00000	0.00000	0.00000	0.041548	0.000000	-0.039498
10 LLL	38.1602	-3.4200	-10.3904	0.00000	0.00000	0.00000	-0.100994	0.000000	0.188268
11 LF	48.1272	-3.4200	-4.4560	0.00000	0.00000	0.00000	-0.187590	0.000000	-0.183587
12 RUA	12.3416	6.2400	-27.1319	0.00000	0.00000	0.00000	-0.114274	0.000000	-0.038215
13 RLA	22.7581	6.2400	-21.4852	0.00000	0.00000	0.00000	-0.266915	0.000000	0.736709
14 LUA	12.3416	-6.2400	-27.1319	0.00000	0.00000	0.00000	-0.114274	0.000000	-0.038215
15 LLA	22.7581	-6.2400	-21.4852	0.00000	0.00000	0.00000	-0.266915	0.000000	0.736709
16 VEH	0.0000	0.0000	0.0000	0.00000	0.00000	0.00000	0.000000	0.000000	0.000000

SEGMENT	(INERTIAL)			(LOCAL)			KINETIC ENERGY		
	U1 ARRAY (IN./SEC.**2)			U2 ARRAY (RAD/SEC.**2)			(LB.- IN.)		
	X	Y	Z	X	Y	Z	LINEAR	ANGULAR	TOTAL
1 LT	-0.1289E+03	0.0000E+00	-0.1135E+04	-0.63845E-15	-0.48015E+02	-0.60266E-16	0.00000E+00	0.00000E+00	0.00000E+00
2 CT	0.0000E+00	0.0000E+00	0.3861E+03	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
3 UT	0.6940E+02	0.0000E+00	0.3701E+03	-0.43592E-20	-0.31861E+01	0.13303E-17	0.00000E+00	0.00000E+00	0.00000E+00
4 N	0.0000E+00	0.0000E+00	0.3861E+03	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
5 H	0.7822E+02	0.0000E+00	0.3680E+03	-0.18372E-18	-0.26721E+00	0.42601E-16	0.00000E+00	0.00000E+00	0.00000E+00
6 RUL	-0.5309E+02	0.0000E+00	-0.1198E+03	0.46503E-17	0.21827E+02	0.67300E-15	0.00000E+00	0.00000E+00	0.00000E+00
7 RLL	0.0000E+00	0.0000E+00	0.3861E+03	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
8 RF	-0.9511E+01	0.0000E+00	-0.5210E+03	0.62882E-20	-0.17610E+03	0.23251E-17	0.00000E+00	0.00000E+00	0.00000E+00
9 LUL	-0.5309E+02	0.0000E+00	-0.1198E+03	0.46503E-17	0.21827E+02	0.67300E-15	0.00000E+00	0.00000E+00	0.00000E+00
10 LLL	0.0000E+00	0.0000E+00	0.3861E+03	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
11 LF	-0.9511E+01	0.0000E+00	-0.5210E+03	0.62882E-20	-0.17610E+03	0.23251E-17	0.00000E+00	0.00000E+00	0.00000E+00
12 RUA	0.0000E+00	0.0000E+00	0.3861E+03	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
13 RLA	0.0000E+00	0.0000E+00	0.3861E+03	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
14 LUA	0.0000E+00	0.0000E+00	0.3861E+03	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
15 LLA	0.0000E+00	0.0000E+00	0.3861E+03	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
							TOTAL BODY KINETIC ENERGY		
							0.00000E+00	0.00000E+00	0.00000E+00

JOINT	IPIN	(INERTIAL)			(INERTIAL)			RELATIVE ANGULAR VELOCITY (RAD/SEC.)	
		JOINT FORCES (LB.)			JOINT TORQUES (IN. LB.)				
		X	Y	Z	X	Y	Z		
1	P	0	-0.177E+02	0.000E+00	-0.100E+03	0.0000E+00	0.0000E+00	0.0000E+00	0.000
2	W	0	-0.192E+02	0.000E+00	-0.859E+02	0.0000E+00	0.0000E+00	0.0000E+00	0.000
3	NP	0	-0.312E+01	0.000E+00	-0.158E+02	0.0000E+00	0.0000E+00	0.0000E+00	0.000
4	HP	0	-0.298E+01	0.000E+00	-0.122E+02	0.0000E+00	0.0000E+00	0.0000E+00	0.000
5	SH	0	0.274E+01	0.000E+00	0.653E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.000
6	RK	1	-0.132E+01	0.000E+00	-0.550E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.000
7	RA	0	-0.339E+00	0.000E+00	0.242E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.000
8	LH	0	0.274E+01	0.000E+00	0.653E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.000
9	LK	1	-0.132E+01	0.000E+00	-0.550E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.000
10	LA	0	-0.339E+00	0.000E+00	0.242E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.000
11	RS	0	-0.221E+01	0.000E+00	-0.731E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.000
12	RE	1	-0.158E+01	0.000E+00	-0.155E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.000
13	LS	0	-0.221E+01	0.000E+00	-0.731E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.000
14	LE	1	-0.158E+01	0.000E+00	-0.155E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.000

POINT NO.	POINT INDEX	SEGMENT NO.	LENGTH (IN.)	BELT STRAIN ENERGY LOSS (IN. LB.)	(LOCAL OR ELLIPSOID) REFERENCE POINT (IN.)			(INERTIAL) BELT FORCES (LB.)			PENETRATION ENERGY LOSS (IN. LB.)
					X	Y	Z	X	Y	Z	
BELT NO. 1 OF HARNES NO. 1											
1	1	16	0.000	0.000	13.000	8.000	-10.300	0.000	0.000	0.000	0.000
2	2	1	4.124	0.000	-1.178	7.075	-0.781	0.000	0.000	0.000	0.000
3	3	1	1.402	0.000	-0.029	6.795	-1.534	0.000	0.000	0.000	0.000
4	4	1	1.574	0.000	0.910	5.778	-2.283	0.000	0.000	0.000	0.000
5	5	1	3.779	0.000	2.228	2.355	-3.192	0.000	0.000	0.000	0.000
6	6	1	2.557	0.000	2.957	0.000	3.059	0.000	0.000	0.000	0.000
7	8	1	2.656	0.000	1.785	-2.325	-3.343	0.000	0.000	0.000	0.000
8	9	1	3.388	0.000	0.011	-5.145	-2.728	0.000	0.000	0.000	0.000
9	10	1	1.187	0.000	-0.880	-5.789	-2.282	0.000	0.000	0.000	0.000
10	11	1	1.940	0.000	-2.460	-6.098	-1.200	0.000	0.000	0.000	0.000
11	12	16	4.738	0.000	13.000	-8.000	-10.300	0.000	0.000	0.000	0.000
TOTAL BELT ENERGY LOSS				0.000							
BELT NO. 2 OF HARNES NO. 1											
12	13	16	0.000	0.000	13.000	-8.000	-10.300	0.000	0.000	0.000	0.000
13	14	1	4.748	0.000	-2.445	-6.101	-1.213	0.000	0.000	0.000	0.000
14	18	2	7.394	0.000	1.818	-5.439	-1.820	0.000	0.000	0.000	0.000
15	21	3	6.842	0.000	4.487	-0.133	3.734	0.000	0.000	0.000	0.000
16	22	3	6.406	0.000	4.421	3.319	-1.662	0.000	0.000	0.000	0.000
17	23	3	5.423	0.000	0.879	4.202	-5.672	0.000	0.000	0.000	0.000
18	27	16	10.829	0.000	0.000	5.000	-35.200	0.000	0.000	0.000	0.000
TOTAL BELT ENERGY LOSS				0.000							
TOTAL HARNES ENERGY LOSS				0.000							

HPTURB ITER = 10 AT TIME = 16.000 MSEC. DELMAX = 0.010537 SCALE = 1.000000
HPTURB ITER = 10 AT TIME = 18.000 MSEC. DELMAX = 0.009232 SCALE = 1.000000
HPTURB ITER = 10 AT TIME = 19.000 MSEC. DELMAX = 0.009627 SCALE = 1.000000
HPTURB ITER = 10 AT TIME = 20.000 MSEC. DELMAX = 0.010050 SCALE = 1.000000
HPTURB ITER = 10 AT TIME = 21.000 MSEC. DELMAX = 0.010116 SCALE = 1.000000
HBPLAY TIME = 22.000 MSEC. NH,NB,NPTS NT= 1 2 8 181
NL(1)= 13 14 18 21 22 23 24 27
BB = 4.748 7.394 6.842 6.406 5.423 0.491 10.338
HPTURB ITER = 10 AT TIME = 22.000 MSEC. DELMAX = 0.012849 SCALE = 1.000000
HPTURB ITER = 10 AT TIME = 23.000 MSEC. DELMAX = 0.011517 SCALE = 1.000000
HPTURB ITER = 10 AT TIME = 24.000 MSEC. DELMAX = 0.011234 SCALE = 1.000000
HPTURB ITER = 10 AT TIME = 25.000 MSEC. DELMAX = 0.011569 SCALE = 1.000000
HPTURB ITER = 10 AT TIME = 26.000 MSEC. DELMAX = 0.012320 SCALE = 1.000000
HPTURB ITER = 10 AT TIME = 28.000 MSEC. DELMAX = 0.010639 SCALE = 1.000000
HPTURB ITER = 10 AT TIME = 29.000 MSEC. DELMAX = 0.012461 SCALE = 1.000000
HPTURB ITER = 10 AT TIME = 30.000 MSEC. DELMAX = 0.014987 SCALE = 1.000000
HBPLAY TIME = 31.000 MSEC. NH,NB,NPTS NT= 1 2 7 181
NL(1)= 13 14 18 21 22 24 27
BB = 4.748 7.394 6.842 6.406 5.572 10.679
HPTURB ITER = 10 AT TIME = 31.000 MSEC. DELMAX = 0.021341 SCALE = 1.000000
HBPLAY TIME = 32.000 MSEC. NH,NB,NPTS NT= 1 2 8 181
NL(1)= 13 14 18 21 22 23 24 27
BB = 4.748 7.394 6.842 6.406 5.066 0.260 10.926
HPTURB ITER = 10 AT TIME = 32.000 MSEC. DELMAX = 0.016547 SCALE = 1.000000
HBPLAY TIME = 33.000 MSEC. NH,NB,NPTS NT= 1 2 7 181
NL(1)= 13 14 18 21 22 23 27
BB = 4.748 7.394 6.842 6.406 5.066 11.186
HPTURB ITER = 10 AT TIME = 33.000 MSEC. DELMAX = 0.036770 SCALE = 1.000000
HBPLAY TIME = 34.000 MSEC. NH,N,NPTS NT= 1 2 8 181
NL(1)= 13 14 18 21 22 23 24 27
BB = 4.748 7.394 6.842 6.406 5.066 0.289 10.896
HPTURB ITER = 10 AT TIME = 34.000 MSEC. DELMAX = 0.041013 SCALE = 1.000000
HBPLAY TIME = 35.000 MSEC. NH,NB,NPTS NT= 1 2 7 181
NL(1)= 13 14 18 21 22 24 27
BB = 4.748 7.394 6.842 6.406 5.216 11.035
HPTURB ITER = 10 AT TIME = 35.000 MSEC. DELMAX = 0.021221 SCALE = 1.000000

```

HBPLAY TIME = 36.000 MSEC. NH,NB,NPTS NT= 1 2 8 181
NL(1)= 13 14 18 21 22 23 24 27
BB = 4.748 7.394 6.842 6.406 4.847 0.195 11.210
HPTURB ITER = 10 AT TIME = 36.000 MSEC. DELMAX = 0.012648 SCALE = 1.000000
HPTURB ITER = 10 AT TIME = 37.000 MSEC. DELMAX = 0.015643 SCALE = 1.000000
HPTURB ITER = 10 AT TIME = 38.000 MSEC. DELMAX = 0.011152 SCALE = 1.000000
HPTURB ITER = 10 AT TIME = 39.000 MSEC. DELMAX = 0.014391 SCALE = 1.000000
HPTURB ITER = 10 AT TIME = 40.000 MSEC. DELMAX = 0.017690 SCALE = 1.000000
HPTURB ITER = 10 AT TIME = 41.000 MSEC. DELMAX = 0.015081 SCALE = 1.000000
HPTURB ITER = 10 AT TIME = 42.000 MSEC. DELMAX = 0.017005 SCALE = 1.000000
HBPLAY TIME = 43.000 MSEC. NH,NB,NPTS NT= 1 2 7 181
NL(1)= 13 14 18 21 22 24 27
BB = 4.748 7.394 6.842 6.406 4.910 11.342
HPTURB ITER = 10 AT TIME = 43.000 MSEC. DELMAX = 0.053120 SCALE = 1.000000
HBPLAY TIME = 44.000 MSEC. NH,NB,NPTS NT= 1 2 8 181
NL(1)= 13 14 18 21 22 23 24 27
BB = 4.748 7.394 6.842 6.406 4.496 0.075 11.681
HPTURB ITER = 10 AT TIME = 44.000 MSEC. DELMAX = 0.013932 SCALE = 1.000000
HPTURB ITER = 10 AT TIME = 45.000 MSEC. DELMAX = 0.012674 SCALE = 1.000000
HBPLAY TIME = 46.000 MSEC. NH,NB,NPTS NT= 1 2 7 181
NL(1)= 13 18 21 22 23 24 27
BB = 12.143 6.842 6.406 4.496 0.055 11.701
HPTURB ITER = 10 AT TIME = 46.000 MSEC. DELMAX = 0.014891 SCALE = 1.000000
HBPLAY TIME = 47.000 MSEC. NH,NB,NPTS NT= 1 1 10 103
NL(1)= 1 3 4 5 6 8 9 10 11 12
BB = 5.526 1.574 3.779 2.557 2.656 3.388 1.187 1.940 4.738
HPTURB ITER = 10 AT TIME = 47.000 MSEC. DELMAX = 0.018982 SCALE = 1.000000
HBPLAY TIME = 48.000 MSEC. NH,NB,NPTS NT= 1 2 6 181
NL(1)= 13 18 21 22 24 27
BB = 12.143 6.842 6.406 4.534 11.718
HPTURB ITER = 10 AT TIME = 48.000 MSEC. DELMAX = 0.064777 SCALE = 1.000000
HBPLAY TIME = 49.000 MSEC. NH,NB,NPTS NT= 1 2 7 181
NL(1)= 13 18 21 22 23 24 27
BB = 12.143 6.842 6.406 4.192 0.045 12.015
HPTURB ITER = 10 AT TIME = 49.000 MSEC. DELMAX = 0.010077 SCALE = 1.000000
HPTURB ITER = 10 AT TIME = 50.000 MSEC. DELMAX = 0.010283 SCALE = 1.000000
HPTURB ITER = 10 AT TIME = 51.000 MSEC. DELMAX = 0.012303 SCALE = 1.000000
HBPLAY TIME = 52.000 MSEC. NH,NB,NPTS NT= 1 2 6 181
NL(1)= 13 18 21 22 24 27
BB = 12.143 6.842 6.406 4.222 12.029
HPTURB ITER = 10 AT TIME = 52.000 MSEC. DELMAX = 0.055553 SCALE = 1.000000
HPTURB ITER = 10 AT TIME = 53.000 MSEC. DELMAX = 0.035482 SCALE = 1.000000
HBPLAY TIME = 54.000 MSEC. NH,NB,NPTS NT= 1 1 9 103
NL(1)= 1 3 4 5 6 8 9 10 12
BB = 5.526 1.574 3.779 2.557 2.656 3.388 1.187 6.678
HPTURB ITER = 10 AT TIME = 54.000 MSEC. DELMAX = 0.024287 SCALE = 1.000000
HPTURB ITER = 10 AT TIME = 55.000 MSEC. DELMAX = 0.015678 SCALE = 1.000000
HPTURB ITER = 10 AT TIME = 56.000 MSEC. DELMAX = 0.011486 SCALE = 1.000000
HBPLAY TIME = 57.000 MSEC. NH,NB,NPTS NT= 1 2 7 181
NL(1)= 13 18 19 21 22 24 27
BB = 12.143 1.887 4.955 6.406 3.880 12.372
HPTURB ITER = 10 AT TIME = 57.000 MSEC. DELMAX = 0.385309 SCALE = 0.259532
HBPLAY TIME = 58.000 MSEC. NH,NB,NPTS NT= 1 2 8 181
NL(1)= 13 18 19 20 21 22 24 27
BB = 12.143 1.887 1.817 3.139 6.406 3.880 12.372
HPTURB ITER = 10 AT TIME = 58.000 MSEC. DELMAX = 0.443658 SCALE = 0.225399
HBPLAY TIME = 59.000 MSEC. NH,NB,NPTS NT= 1 2 9 181
NL(1)= 13 16 18 19 20 21 22 24 27
BB = 7.889 4.253 1.887 1.817 3.139 6.406 3.974 12.277
HPTURB ITER = 10 AT TIME = 59.000 MSEC. DELMAX = 0.052001 SCALE = 1.000000
HBPLAY TIME = 60.000 MSEC. NH,NB,NPTS NT= 1 2 8 181
NL(1)= 13 18 19 20 21 22 24 27
BB = 12.143 1.887 1.817 3.139 6.406 4.086 12.165
HPTURB ITER = 10 AT TIME = 60.000 MSEC. DELMAX = 0.057808 SCALE = 1.000000
HBPLAY TIME = 61.000 MSEC. NH,NB,NPTS NT= 1 2 7 181
NL(1)= 13 18 20 21 22 24 27

```


BB = 12.143 3.703 3.139 6.406 4.156 12.096
HPTURB ITER = 10 AT TIME = 61.000 MSEC. DELMAX = 0.015872 SCALE = 1.000000
HPTURB ITER = 10 AT TIME = 62.000 MSEC. DELMAX = 0.022130 SCALE = 1.000000
HPTURB ITER = 10 AT TIME = 63.000 MSEC. DELMAX = 0.015337 SCALE = 1.000000
HBPLAY TIME = 64.000 MSEC. NH,NB,NPTS NT= 1 2 6 181
NL(1)= 13 18 20 21 22 27

BB = 12.143 3.703 3.139 6.406 16.252
HPTURB ITER = 10 AT TIME = 64.000 MSEC. DELMAX = 0.011067 SCALE = 1.000000
HBPLAY TIME = 66.000 MSEC. NH,NB,NPTS NT= 1 1 8 103
NL(1)= 1 3 4 5 6 8 9 12

BB = 5.526 1.574 3.779 2.557 2.656 3.388 7.865
HPTURB ITER = 10 AT TIME = 67.000 MSEC. DELMAX = 0.011853 SCALE = 1.000000
HPTURB ITER = 10 AT TIME = 69.000 MSEC. DELMAX = 0.013454 SCALE = 1.000000

DINT CONV. TEST	72.000	N	ANG VEL	22.55	0.2652E-02	0.1176E-03	0.1000E-03	0.1000E-03	0.1000E-03
TEST FAILED AT TIME =	0.072000	FOR H =	0.001000						
DINT CONV. TEST	73.000	N	ANG VEL	23.86	0.3624E-02	0.1518E-03	0.1000E-03	0.1000E-03	0.1000E-03
TEST FAILED AT TIME =	0.073000	FOR H =	0.001000						
DINT CONV. TEST	74.000	N	ANG VEL	26.11	0.3039E-02	0.1164E-03	0.1000E-03	0.1000E-03	0.1000E-03
TEST FAILED AT TIME =	0.074000	FOR H =	0.001000						

SEGMENT	(INERTIAL)			(LOCAL)			(LOCAL)		
	ANGULAR ROTATION (DEG)			ANGULAR VELOCITY (RAD/SEC.)			ANGULAR ACCELERATION (RAD/SEC.**2)		
	YAW	PITCH	ROLL	X	Y	Z	X	Y	Z
1 LT	10.9078	40.4616	-2.5237	-3.50101	6.77176	3.01702	-712.525840	-923.312215	-344.896206
2 CT	5.8648	8.3877	-35.0629	-28.78011	38.13662	26.32712	2669.823693	4863.705047	714.436127
3 UT	22.6603	16.1876	4.2073	-4.55898	-8.33212	11.28470	-833.189016	-740.536145	24.730106
4 N	9.4134	-28.3216	9.0212	3.10468	-11.10456	11.62142	-474.529280	-2359.202266	525.452345
5 H	-14.7037	-62.0894	29.6482	16.06835	-58.33883	6.84809	321.909906	-351.659928	252.225950
6 RUL	11.0166	100.4701	10.5736	-0.26043	-3.94123	-2.59116	-82.766870	-780.201793	66.972956
7 RLL	-2.0078	53.9069	-3.2449	-2.05452	9.15294	-1.60027	12.275725	-89.117047	79.212811
8 RF	-1.2125	146.0433	2.4774	3.41634	1.03537	-1.56581	-92.295830	747.087282	-23.919556
9 LUL	4.3688	101.7927	4.4680	-0.37199	-3.33541	-0.44435	-290.296221	-679.956632	528.292675
10 LLL	-0.9289	45.3497	-1.2981	-0.57596	5.97768	0.06395	278.790376	-4.508497	528.778813
11 LF	-0.5610	155.8698	1.6050	1.79239	0.13205	-0.59660	-177.520463	1209.576341	196.625307
12 RUA	-75.0950	70.5657	-80.2833	-6.23227	-4.14211	-1.63954	244.397026	-3261.631888	-168.345676
13 RLA	-21.0534	51.2884	-31.6271	-6.18955	1.21106	1.79408	113.635227	2487.522772	-303.100898
14 LUA	-17.8558	62.4794	-23.3643	-2.95595	-0.00123	5.46149	322.014085	-2838.181404	888.201239
15 LLA	-16.0460	60.3973	-21.7744	-2.73881	-15.25083	5.57354	441.700348	4146.746554	916.611637
16 VEH	0.0000	0.0000	0.0000	0.00000	0.00000	0.00000	0.000000	0.000000	0.000000

SEGMENT	(INERTIAL)			(INERTIAL)			(INERTIAL)		
	LINEAR POSITION (IN.)			LINEAR VELOCITY (IN./SEC.)			LINEAR ACCELERATIONS (G'S)		
	X	Y	Z	X	Y	Z	X	Y	Z
1 LT	6.0377	0.0702	-13.0395	-334.65671	-2.92122	-92.21272	1.118456	-3.419755	-24.241372
2 CT	3.5221	-1.5083	-17.2697	-403.89532	-63.36301	-39.55188	-7.411998	0.308363	-23.341359
3 UT	1.5893	-2.7194	-24.7657	-399.28403	-115.11128	-20.63141	-8.535068	3.408484	-20.077252
4 N	-0.4641	-2.8941	-33.1662	-319.46157	-121.58509	-38.64033	16.009901	-4.901599	-22.093312
5 H	4.8868	-1.0902	-36.0984	-178.47710	-69.83431	287.80272	-29.256214	-10.703850	8.134162
6 RUL	13.6013	3.3950	-14.9332	-345.17072	-4.32318	-59.80904	2.896528	-1.986709	-14.025632
7 RLL	29.0739	3.6383	-12.6201	-300.76311	12.64266	-73.16830	3.883535	0.620604	6.154392
8 RF	39.4986	3.6714	-8.4455	-258.02278	23.08008	-137.61724	0.173190	0.601745	-0.915789
9 LUL	15.0441	-3.3466	-14.9210	-314.88548	2.85212	-60.25035	6.374569	3.680291	-3.334002
10 LLL	29.7934	-3.3024	-12.0614	-278.81100	10.63011	-57.36381	9.156247	6.479980	13.521611
11 LF	39.2761	-3.3229	-7.6097	-244.93381	11.69052	-92.38042	4.608846	0.164379	1.561647
12 RUA	3.1574	3.4023	-29.0240	-426.99852	-109.61718	-32.04897	-2.643843	6.759738	6.506879
13 RLA	15.1710	5.9986	-24.3530	-429.54688	-25.03050	-46.34975	23.116917	3.958924	9.885922
14 LUA	7.1965	-8.0306	-27.9271	-282.38560	-78.35066	-29.72059	-9.173876	2.088389	27.959874
15 LLA	19.2594	-6.2444	-21.8664	-343.80786	-23.50199	71.96752	11.107168	-20.329519	-12.311189
16 VEH	-12.3548	0.0000	0.0000	-308.87040	0.00000	0.00000	0.000000	0.000000	0.000000

SEGMENT	(INERTIAL)			(LOCAL)			KINETIC ENERGY			
	U1 ARRAY (IN./SEC.**2)			U2 ARRAY (RAD/SEC.**2)			(LB.- IN.)			
	EXTERNAL LINEAR ACCELERATIONS			EXTERNAL ANGULAR ACCELERATIONS			LINEAR	ANGULAR	TOTAL	
	X	Y	Z	X	Y	Z				
1	LT	-0.3009E+03	0.1559E+02	-0.2698E+05	-0.60308E+03	-0.27927E+04	-0.22971E+03	0.54266E+04	0.41308E+02	0.54679E+04
2	CT	0.2569E+03	-0.5220E+03	-0.6204E+03	0.88429E+04	0.83818E+04	0.71444E+03	0.28620E+04	0.54259E+03	0.34046E+04
3	UT	-0.6894E+03	-0.2430E+03	0.4419E+03	-0.69029E+03	-0.15455E+03	0.16513E+02	0.12032E+05	0.28218E+03	0.12314E+05
4	N	0.0000E+00	0.0000E+00	0.3861E+03	-0.26984E+04	0.10411E+05	0.52545E+03	0.50402E+03	0.33066E+01	0.50733E+03
5	H	0.0000E+00	0.0000E+00	0.3861E+03	-0.35125E+03	0.92855E+03	0.25223E+03	0.18467E+04	0.56365E+03	0.24104E+04
6	RUL	0.1244E+03	-0.7761E+01	0.5426E+02	-0.23939E+01	0.52884E+02	-0.10354E+01	0.36122E+04	0.16567E+02	0.36287E+04
7	RLL	0.0000E+00	0.0000E+00	0.3861E+03	-0.82176E+01	-0.24413E+03	-0.15598E+03	0.12134E+04	0.22031E+02	0.12355E+04
8	RF	0.0000E+00	0.0000E+00	0.3861E+03	-0.18022E+03	0.11120E+04	-0.75866E+03	0.23167E+03	0.27743E+00	0.23195E+03
9	LUL	-0.1221E+04	0.2472E+04	0.2645E+04	-0.70853E+03	-0.12076E+04	-0.31901E+03	0.30251E+04	0.11362E+02	0.30365E+04
10	LLL	0.0000E+00	0.0000E+00	0.3861E+03	-0.66486E+01	-0.17362E+03	-0.14022E+03	0.10259E+04	0.89964E+01	0.10349E+04
11	LF	0.0000E+00	0.0000E+00	0.3861E+03	-0.20969E+03	0.81201E+03	-0.72214E+02	0.18487E+03	0.69768E-01	0.18494E+03
12	RUA	0.0000E+00	0.0000E+00	0.3861E+03	-0.44483E+02	-0.70077E+04	0.79493E+03	0.14022E+04	0.49151E+01	0.14071E+04
13	RLA	-0.8603E+02	-0.4004E+01	0.7761E+03	0.31299E+02	0.34441E+04	-0.57371E+02	0.14312E+04	0.66593E+01	0.14379E+04
14	LUA	0.0000E+00	0.0000E+00	0.3861E+03	-0.16398E+03	-0.90273E+03	0.56251E+02	0.62271E+03	0.11478E+01	0.62386E+03
15	LLA	-0.9690E+04	-0.7116E+04	-0.3749E+04	0.20275E+04	0.70096E+04	0.12476E+04	0.94712E+03	0.40319E+02	0.98744E+03
							TOTAL BODY KINETIC ENERGY			
							0.36368E+05	0.15454E+04	0.37913E+05	

JOINT	IPIN	(INERTIAL)			(INERTIAL)			RELATIVE ANGULAR VELOCITY (RAD/SEC.)	
		JOINT FORCES (LB.)			JOINT TORQUES (IN. LB.)				
		X	Y	Z	X	Y	Z		
1	P	0	-0.478E+03	0.156E+03	-0.120E+04	0.1437E+04	0.1730E+04	-0.4588E+03	47.900
2	W	0	-0.373E+03	0.134E+03	-0.919E+03	-0.2162E+04	-0.6568E+03	0.7903E+03	63.460
3	NP	0	-0.296E+03	-0.144E+03	0.914E+01	-0.2178E+03	0.5526E+03	0.5465E+02	4.653
4	HP	0	-0.349E+03	-0.128E+03	0.851E+02	-0.8035E+02	0.2872E+03	0.4001E+02	48.641
5	RH	0	0.968E+02	-0.374E+02	-0.276E+03	0.1549E+01	0.6042E+02	0.1998E+02	11.110
6	RK	1	0.383E+02	0.731E+01	0.463E+02	-0.9820E+01	-0.7377E+02	0.4208E+02	13.094
7	RA	0	0.360E+00	0.125E+01	-0.398E+01	0.6057E+01	0.4563E+02	0.6082E+01	8.104
8	LH	0	0.316E+03	0.174E+01	-0.108E+03	-0.1049E+02	0.5690E+02	0.2195E+02	10.800
9	LK	1	0.990E+02	0.636E+02	0.123E+03	-0.1739E+03	-0.4959E+02	0.2365E+03	9.313
10	LA	0	0.958E+01	0.342E+00	0.117E+01	0.8697E+01	0.3333E+02	0.3159E+01	6.037
11	RS	0	0.123E+03	0.609E+02	0.770E+02	0.2329E+02	-0.3963E+02	0.2492E+02	9.125
12	RE	1	0.138E+03	0.234E+02	0.465E+02	-0.5987E+02	0.1090E+04	-0.4321E+03	5.353
13	LS	0	0.163E+03	0.378E+00	0.134E+03	-0.7896E+01	-0.5782E+02	0.3755E+02	12.112
14	LE	1	0.214E+03	-0.112E+02	-0.153E+02	-0.1860E+03	0.1440E+03	0.2989E+03	15.250

POSTPROCESSOR CONTROL PARAMETERS

	HIC & HSI POINT	CSI POINT
H.11	2	1

RUN DESCRIPTION: EXAMPLE 1: BASIC SLED TEST SIMULATION

TWO BELT HARNESS WITH HYPERELLIPSOID FOR DASH BOARD

PAGE: 21.01

VEHICLE DECELERATION: SLED ACCELERATION - 20G PEAK

CRASH VICTIM: 95TH PERCENTILE MALE

POINT TOTAL ACCELERATION (G'S)

TIME (MSEC)	POINT (0.00, 0.00, 0.00) ON SEGMENT NO. 3 - UT				POINT (0.00, 0.00, 0.00) ON SEGMENT NO. -5 - H				POINT (0.00, 0.00, 0.00) ON SEGMENT NO. 5 - H			
	IN	UT	REFERENCE	RES	IN	UT	REFERENCE	RES	IN	VEH	REFERENCE	RES
0.000	-0.020	0.000	-0.082	0.084	-0.262	0.000	0.890	0.928	-0.047	0.000	-0.073	0.087
2.000	-0.014	0.000	-0.107	0.108	-0.260	0.000	0.865	0.903	-0.051	0.000	-0.098	0.111
4.000	-0.007	0.000	-0.179	0.179	-0.258	0.000	0.793	0.834	-0.067	0.000	-0.168	0.181
6.000	-0.006	-0.003	-0.247	0.247	-0.269	0.000	0.725	0.774	-0.093	0.000	-0.232	0.250
8.000	-0.019	-0.012	-0.314	0.315	-0.306	-0.002	0.658	0.725	-0.144	-0.002	-0.289	0.323
10.000	-0.061	-0.023	-0.386	0.391	-0.379	-0.005	0.586	0.698	-0.233	-0.005	-0.342	0.413
12.000	-0.091	-0.032	-0.464	0.474	-0.475	-0.008	0.507	0.695	-0.344	-0.008	-0.396	0.525
14.000	-0.099	-0.023	-0.547	0.556	-0.487	-0.006	0.425	0.646	-0.375	-0.006	-0.474	0.604
16.000	-0.132	-0.024	-0.623	0.638	-0.506	-0.007	0.348	0.614	-0.411	-0.007	-0.544	0.682
18.000	-0.111	-0.005	-0.719	0.727	-0.510	-0.003	0.253	0.570	-0.437	-0.003	-0.635	0.771
20.000	-0.133	-0.001	-0.791	0.802	-0.525	-0.002	0.181	0.555	-0.468	-0.002	-0.702	0.844
22.000	-0.197	-0.016	-0.854	0.876	-0.551	-0.005	0.119	0.564	-0.509	-0.006	-0.756	0.911
24.000	-0.382	-0.063	-1.317	1.373	-0.662	-0.179	-0.342	0.767	-0.724	-0.179	-1.179	1.395
26.000	-0.390	-0.112	-1.296	1.358	-0.685	-0.231	-0.314	0.788	-0.739	-0.231	-1.147	1.384
28.000	-0.357	0.556	-1.166	1.340	-0.682	-0.014	-0.181	0.705	-0.705	-0.014	-1.019	1.239
30.000	-0.438	0.816	-1.246	1.553	-0.702	0.035	-0.256	0.748	-0.742	0.035	-1.087	1.317
32.000	-2.733	-0.835	-0.461	2.895	-1.271	-0.218	0.576	1.412	-1.106	-0.220	-0.149	1.137
34.000	-2.186	-0.420	-0.403	2.263	-1.208	-0.216	0.667	1.397	-1.025	-0.219	-0.076	1.051
36.000	-3.145	-0.474	-0.313	3.196	-1.481	-0.353	0.849	1.743	-1.252	-0.357	0.159	1.312
38.000	-4.178	-0.580	-1.221	4.391	-1.843	-0.709	0.133	1.979	-1.769	-0.710	-0.467	1.963
40.000	-5.137	0.364	-1.250	5.299	-2.004	-0.530	0.280	2.092	-1.897	-0.533	-0.297	1.993
42.000	-6.838	1.397	-2.254	7.334	-2.388	-0.543	-0.390	2.479	-2.417	-0.540	-0.885	2.630
44.000	-14.419	-3.743	0.830	14.920	-4.097	-1.242	3.893	5.787	-3.227	-1.300	3.624	5.024
46.000	-16.621	-1.760	-0.518	16.722	-4.375	-1.075	3.474	5.689	-3.633	-1.143	3.226	4.991
48.000	-21.535	0.772	0.632	21.558	-4.411	-0.608	5.515	7.088	-3.372	-0.744	5.190	6.234
50.000	-41.704	-14.755	6.467	44.707	-8.393	-3.295	18.145	20.262	-5.470	-3.858	18.124	19.321
52.000	-46.097	-12.754	8.809	48.633	-8.850	-2.015	24.198	25.844	-5.684	-2.996	24.033	24.877
54.000	-59.506	-18.177	14.142	63.807	-11.972	-2.812	40.775	42.589	-8.224	-4.949	40.493	41.615
56.000	-60.433	-15.422	13.472	63.808	-11.763	-2.643	48.226	49.710	-9.706	-5.876	47.398	48.737
58.000	-73.471	-17.978	-2.410	75.677	-11.839	-4.818	45.094	46.871	-12.593	-8.610	43.319	45.927
60.000	-93.922	-15.685	14.909	96.382	-10.687	1.243	64.829	65.716	-16.338	-5.515	62.414	64.752
62.000	-39.398	-7.543	-3.263	40.247	-8.458	0.385	50.073	50.783	-16.781	-5.749	46.585	49.848
64.000	-13.416	-3.242	-10.799	17.525	-4.985	0.413	34.219	34.583	-13.561	-4.374	30.511	33.674
66.000	2.837	-2.587	-20.041	20.406	-4.396	-0.971	17.138	17.719	-9.978	-3.581	13.198	16.929
68.000	12.171	-3.027	-29.353	31.921	-6.574	-2.288	10.559	12.647	-10.367	-3.898	5.106	12.195
70.000	12.817	-4.431	-27.755	30.891	-6.991	-2.721	11.812	13.993	-11.904	-4.656	4.693	13.617
72.000	8.827	-2.877	-26.335	27.924	-5.142	-2.316	12.679	13.876	-11.519	-4.665	5.172	13.461
74.000	3.362	-1.720	-32.445	32.664	-5.688	-2.458	13.892	15.211	-13.307	-5.198	4.223	14.897
76.000	-2.984	1.218	-32.702	32.861	-6.355	-2.850	21.202	22.317	-19.656	-7.488	6.458	22.003
78.000	-4.104	3.348	-30.967	31.417	-5.758	-2.791	26.452	27.215	-24.295	-9.144	7.173	26.932
80.000	-0.706	4.868	-21.526	22.081	-3.903	-2.154	32.157	32.464	-29.256	-10.704	8.134	32.197

DATE: 2 SEPT 1988

RUN DESCRIPTION: EXAMPLE 1: BASIC SLED TEST SIMULATION

TWO BELT HARNESS WITH HYPERELLIPSOID FOR DASH BOARD

PAGE: 22.01

VEHICLE DECELERATION: SLED ACCELERATION - 20G PEAK

CRASH VICTIM: 95TH PERCENTILE MALE

POINT REL. VELOCITY (IN./SEC.)

TIME (MSEC)	POINT (0.00, 0.00, 0.00) ON SEGMENT NO. 3 - UT				POINT (0.00, 0.00, 0.00) ON SEGMENT NO. 5 - H				POINT (0.00, 0.00, 0.00) ON SEGMENT NO. 5 - H			
	IN VEH REFERENCE				IN VEH REFERENCE				IN UT REFERENCE			
	X	Y	Z	RES	X	Y	Z	RES	X	Y	Z	RES
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
2.000	0.357	0.000	-0.064	0.362	0.349	0.000	-0.062	0.355	-0.008	0.000	0.000	0.008
4.000	1.482	0.000	-0.168	1.491	1.463	0.000	-0.163	1.472	-0.020	0.000	0.000	0.020
6.000	3.368	-0.002	-0.329	3.384	3.332	0.000	-0.320	3.347	-0.037	0.002	0.000	0.037
8.000	6.010	-0.010	-0.534	6.034	5.944	-0.002	-0.519	5.967	-0.067	0.008	0.000	0.068
10.000	9.392	-0.024	-0.792	9.425	9.275	-0.005	-0.764	9.307	-0.120	0.019	0.000	0.121
12.000	13.502	-0.048	-1.096	13.546	13.296	-0.010	-1.048	13.338	-0.211	0.038	0.000	0.214
14.000	18.360	-0.073	-1.462	18.418	18.037	-0.016	-1.386	18.090	-0.332	0.057	0.000	0.336
16.000	23.966	-0.096	-1.885	24.040	23.526	-0.022	-1.782	23.593	-0.452	0.074	-0.001	0.458
18.000	30.318	-0.118	-2.369	30.411	29.761	-0.028	-2.239	29.845	-0.572	0.090	-0.001	0.579
20.000	37.426	-0.138	-2.913	37.539	36.747	-0.034	-2.754	36.851	-0.697	0.104	-0.001	0.704
22.000	45.254	-0.173	-3.495	45.389	44.477	-0.043	-3.312	44.601	-0.798	0.130	-0.001	0.808
24.000	53.624	-0.222	-4.417	53.806	52.814	-0.157	-4.225	52.983	-0.832	0.065	-0.001	0.835
26.000	62.779	-0.294	-5.297	63.002	61.919	-0.307	-5.089	62.128	-0.885	-0.012	0.003	0.885
28.000	72.709	-0.176	-6.137	72.968	71.793	-0.419	-5.907	72.037	-0.945	-0.242	0.010	0.975
30.000	83.403	0.352	-6.915	83.690	82.451	-0.398	-6.666	82.721	-0.986	-0.749	0.019	1.238
32.000	94.080	0.535	-7.240	94.360	93.727	-0.423	-7.110	93.997	-0.376	-0.957	0.041	1.029
34.000	104.889	0.071	-7.234	105.138	105.604	-0.633	-7.306	105.859	0.710	-0.706	0.097	1.006
36.000	116.510	-0.408	-7.277	116.737	118.233	-0.984	-7.495	118.474	1.722	-0.585	0.197	1.830
38.000	127.722	-1.024	-7.102	127.923	131.306	-1.474	-7.573	131.533	3.590	-0.474	0.404	3.643
40.000	139.233	-1.201	-6.715	139.400	145.087	-1.867	-7.440	145.290	5.849	-0.719	0.720	5.937
42.000	149.450	-0.660	-6.479	149.592	158.571	-2.237	-7.528	158.765	9.079	-1.685	1.225	9.315
44.000	155.812	-0.836	-4.740	155.887	170.774	-2.712	-6.337	170.913	14.855	-2.106	2.198	15.163
46.000	157.720	-3.088	-1.115	157.754	181.675	-3.707	-3.283	181.743	23.690	-1.112	4.061	24.061
48.000	156.096	-4.584	3.913	156.213	191.901	-4.378	0.957	191.953	35.308	-0.762	6.593	35.927
50.000	141.126	-13.942	16.832	142.808	199.739	-6.638	13.452	200.301	57.525	5.185	12.814	59.163
52.000	122.195	-24.412	32.266	128.719	206.670	-9.011	29.910	209.017	82.311	11.324	21.803	85.900
54.000	94.373	-38.752	54.224	115.535	211.079	-12.280	56.971	218.976	112.157	18.960	37.279	119.701
56.000	63.940	-52.830	78.311	114.069	213.707	-16.371	91.124	232.899	141.171	23.677	58.599	154.673
58.000	27.937	-68.681	99.258	123.894	214.085	-21.715	126.806	249.767	171.680	26.531	86.244	193.948
60.000	-48.785	-89.387	132.584	167.178	209.556	-25.937	169.668	270.875	238.362	28.911	120.368	268.591
62.000	-77.547	-101.219	145.352	193.355	203.606	-30.570	209.855	293.988	252.303	25.298	154.604	296.984
64.000	-88.016	-107.303	145.939	201.393	199.040	-34.244	238.403	312.451	249.362	19.440	183.655	310.304
66.000	-87.056	-109.586	135.436	194.758	196.297	-37.048	254.173	323.279	237.429	13.587	207.586	315.673
68.000	-81.208	-110.824	115.623	179.570	193.532	-39.896	261.153	327.486	220.628	9.256	230.063	318.891
70.000	-73.848	-112.242	90.482	161.984	189.212	-43.176	264.497	328.061	200.518	6.746	252.981	322.881
72.000	-68.106	-113.520	68.202	148.919	183.289	-46.872	268.296	328.290	181.488	4.619	273.349	328.145
74.000	-67.903	-114.780	45.146	140.795	177.090	-50.470	271.958	328.435	167.908	1.761	295.644	340.002
76.000	-74.122	-116.052	20.340	139.197	166.505	-55.141	276.103	327.105	156.313	-2.407	320.292	356.408
78.000	-83.506	-116.413	-2.649	143.291	150.797	-61.731	281.259	325.049	143.815	-8.988	343.115	372.145
80.000	-90.414	-115.111	-20.631	147.820	130.393	-69.834	287.803	323.589	126.450	-16.916	360.086	382.017

RUN DESCRIPTION: EXAMPLE 1: BASIC SLED TEST SIMULATION
TWO BELT HARNESS WITH HYPERELLIPSOID FOR DASH BOARD

PAGE: 23.01

VEHICLE DECELERATION: SLED ACCELERATION - 20G PEAK
CRASH VICTIM: 95TH PERCENTILE MALE

POINT REL. LINEAR DISPLACEMENT (IN.)

TIME (MSEC)	POINT (0.00, 0.00, 0.00) ON SEGMENT NO. 3 - UT				POINT (0.00, 0.00, 0.00) ON SEGMENT NO. 5 - H				POINT (0.00, 0.00, 0.00) ON SEGMENT NO. 5 - H			
	IN VEH X	REFERENCE Y	REFERENCE Z	RES	IN VEH X	REFERENCE Y	REFERENCE Z	RES	IN X	UT Y	REFERENCE Z	RES
0.000	11.314	0.000	-26.938	29.217	7.775	0.000	-41.833	42.550	-0.022	0.000	-15.310	15.310
2.000	11.314	0.000	-26.938	29.218	7.775	0.000	-41.833	42.550	-0.022	0.000	-15.310	15.310
4.000	11.315	0.000	-26.938	29.218	7.777	0.000	-41.834	42.550	-0.022	0.000	-15.310	15.310
6.000	11.320	0.000	-26.939	29.221	7.782	0.000	-41.834	42.552	-0.023	0.000	-15.310	15.310
8.000	11.330	0.000	-26.940	29.225	7.791	0.000	-41.835	42.554	-0.023	0.000	-15.310	15.310
10.000	11.345	0.000	-26.941	29.232	7.806	0.000	-41.836	42.558	-0.022	0.000	-15.310	15.310
12.000	11.368	0.000	-26.943	29.243	7.828	0.000	-41.838	42.564	-0.022	0.000	-15.310	15.310
14.000	11.400	0.000	-26.945	29.257	7.860	0.000	-41.840	42.572	-0.021	0.000	-15.310	15.310
16.000	11.442	0.000	-26.949	29.277	7.901	0.000	-41.844	42.583	-0.020	0.001	-15.310	15.310
18.000	11.496	-0.001	-26.953	29.302	7.954	0.000	-41.848	42.597	-0.017	0.001	-15.310	15.310
20.000	11.564	-0.001	-26.958	29.334	8.021	0.000	-41.853	42.614	-0.015	0.001	-15.310	15.310
22.000	11.646	-0.001	-26.965	29.372	8.102	0.000	-41.859	42.636	-0.011	0.002	-15.310	15.310
24.000	11.745	-0.002	-26.972	29.419	8.199	0.000	-41.866	42.661	-0.005	0.004	-15.310	15.310
26.000	11.861	-0.002	-26.982	29.474	8.314	-0.001	-41.876	42.693	0.003	0.008	-15.310	15.310
28.000	11.996	-0.003	-26.994	29.539	8.447	-0.002	-41.887	42.730	0.014	0.016	-15.310	15.310
30.000	12.152	-0.003	-27.007	29.615	8.601	-0.003	-41.899	42.773	0.027	0.024	-15.310	15.310
32.000	12.330	-0.001	-27.021	29.701	8.777	-0.003	-41.913	42.822	0.044	0.032	-15.310	15.310
34.000	12.529	-0.001	-27.036	29.798	8.977	-0.004	-41.928	42.878	0.069	0.042	-15.310	15.310
36.000	12.750	-0.001	-27.050	29.905	9.200	-0.006	-41.943	42.940	0.102	0.058	-15.309	15.309
38.000	12.995	-0.003	-27.065	30.023	9.450	-0.008	-41.958	43.009	0.147	0.081	-15.308	15.309
40.000	13.262	-0.005	-27.079	30.152	9.726	-0.012	-41.973	43.085	0.206	0.112	-15.306	15.308
42.000	13.551	-0.007	-27.092	30.292	10.030	-0.016	-41.988	43.169	0.279	0.149	-15.303	15.306
44.000	13.857	-0.007	-27.104	30.441	10.360	-0.021	-42.002	43.261	0.371	0.191	-15.298	15.303
46.000	14.171	-0.011	-27.110	30.590	10.712	-0.027	-42.012	43.356	0.499	0.245	-15.288	15.298
48.000	14.486	-0.019	-27.107	30.735	11.086	-0.035	-42.014	43.452	0.659	0.307	-15.273	15.290
50.000	14.784	-0.037	-27.087	30.859	11.478	-0.046	-42.001	43.541	0.868	0.380	-15.247	15.276
52.000	15.048	-0.075	-27.039	30.944	11.884	-0.062	-41.958	43.609	1.144	0.470	-15.201	15.252
54.000	15.266	-0.138	-26.953	30.976	12.302	-0.083	-41.873	43.643	1.497	0.561	-15.128	15.212
56.000	15.424	-0.230	-26.821	30.940	12.727	-0.111	-41.726	43.624	1.939	0.645	-15.010	15.148
58.000	15.520	-0.349	-26.641	30.834	13.156	-0.149	-41.507	43.543	2.449	0.714	-14.836	15.054
60.000	15.493	-0.507	-26.412	30.624	13.579	-0.197	-41.214	43.394	2.991	0.751	-14.606	14.928
62.000	15.361	-0.699	-26.130	30.319	13.993	-0.253	-40.832	43.164	3.548	0.697	-14.323	14.772
64.000	15.193	-0.909	-25.837	29.987	14.395	-0.318	-40.382	42.872	4.108	0.585	-13.976	14.579
66.000	15.017	-1.126	-25.554	29.661	14.790	-0.390	-39.887	42.543	4.633	0.439	-13.579	14.354
68.000	14.848	-1.347	-25.302	29.368	15.180	-0.466	-39.371	42.199	5.107	0.285	-13.141	14.101
70.000	14.693	-1.569	-25.096	29.123	15.563	-0.549	-38.845	41.851	5.528	0.131	-12.660	13.815
72.000	14.552	-1.795	-24.938	28.929	15.936	-0.639	-38.313	41.500	5.887	-0.018	-12.144	13.496
74.000	14.417	-2.024	-24.824	28.778	16.297	-0.737	-37.772	41.145	6.179	-0.156	-11.604	13.147
76.000	14.275	-2.255	-24.759	28.668	16.641	-0.842	-37.225	40.784	6.409	-0.281	-11.038	12.767
78.000	14.118	-2.487	-24.742	28.595	16.959	-0.959	-36.667	40.411	6.577	-0.392	-10.451	12.355
80.000	13.944	-2.719	-24.766	28.551	17.242	-1.090	-36.098	40.019	6.684	-0.491	-9.851	11.915

RUN DESCRIPTION: EXAMPLE 1: BASIC SLED TEST SIMULATION
TWO BELT HARNESS WITH HYPERELLIPSOID FOR DASH BOARD

PAGE: 24.01

VEHICLE DECELERATION: SLED ACCELERATION - 20G PEAK
CRASH VICTIM: 95TH PERCENTILE MALE

SEGMENT ANGULAR ACCELERATION (REV/SEC.**2)

TIME (MSEC)	SEGMENT NO. 3 - UT				SEGMENT NO. 5 - H				SEGMENT NO. 5 - H			
	IN	UT	REFERENCE	RES	IN	H	REFERENCE	RES	IN	VEH	REFERENCE	RES
	X	Y	Z		X	Y	Z		X	Y	Z	
0.000	0.000	-0.137	0.000	0.137	0.000	-0.185	0.000	0.185	0.000	-0.185	0.000	0.185
2.000	0.001	-0.042	0.002	0.043	-0.001	-0.052	0.000	0.052	-0.001	-0.052	0.000	0.052
4.000	0.006	0.040	0.015	0.043	-0.006	-0.067	0.000	0.067	-0.006	-0.067	0.002	0.067
6.000	-0.004	0.247	0.049	0.252	0.021	-0.177	-0.001	0.178	0.020	-0.177	-0.005	0.178
8.000	-0.039	0.665	0.129	0.678	0.107	-0.365	-0.002	0.380	0.104	-0.365	-0.027	0.380
10.000	-0.080	1.188	0.277	1.223	0.221	-0.482	0.002	0.530	0.215	-0.482	-0.050	0.530
12.000	-0.118	1.611	0.416	1.669	0.332	-0.264	0.016	0.424	0.326	-0.264	-0.062	0.424
14.000	-0.083	1.943	0.457	1.997	0.283	-0.730	0.029	0.784	0.282	-0.730	-0.038	0.784
16.000	-0.087	2.460	0.635	2.542	0.302	-1.430	0.043	1.462	0.303	-1.430	-0.029	1.462
18.000	0.008	2.589	0.664	2.673	0.141	-1.610	0.060	1.617	0.151	-1.610	0.026	1.617
20.000	0.037	2.973	0.948	3.120	0.102	-2.115	0.080	2.119	0.118	-2.115	0.054	2.119
22.000	-0.010	3.607	1.429	3.879	0.244	-3.094	0.108	3.106	0.262	-3.094	0.048	3.106
24.000	-8.600	7.157	4.790	12.171	7.768	-7.307	0.183	10.666	7.598	-7.307	-1.626	10.666
26.000	-9.914	7.450	4.950	13.353	10.014	-8.144	0.309	12.911	9.812	-8.143	-2.022	12.911
28.000	-1.602	6.837	5.335	8.819	0.731	-7.928	0.450	7.974	0.814	-7.928	0.260	7.974
30.000	-2.418	7.045	5.092	9.022	-1.971	-8.546	0.594	8.791	-1.785	-8.548	1.017	8.791
32.000	-5.518	18.566	11.214	22.381	8.491	-29.792	0.788	30.988	8.434	-29.793	-1.243	30.988
34.000	-5.789	14.342	10.997	18.977	8.985	-28.116	1.107	29.537	8.988	-28.117	-1.045	29.537
36.000	-10.236	20.115	12.679	25.887	15.101	-38.011	1.455	40.927	15.018	-38.013	-2.127	40.927
38.000	-27.100	26.891	17.134	41.846	30.178	-51.830	1.903	60.006	29.806	-51.828	-5.111	60.006
40.000	-22.699	24.663	16.092	37.182	22.068	-57.722	2.414	61.843	21.998	-57.727	-2.878	61.843
42.000	-30.498	28.252	24.219	48.113	21.827	-71.925	2.986	75.223	21.854	-71.937	-2.442	75.223
44.000	-32.187	56.105	49.690	81.565	50.733	-136.186	3.903	145.381	50.213	-136.179	-8.335	145.381
46.000	-28.042	37.572	48.859	67.714	44.955	-148.639	5.635	155.391	44.854	-148.665	-5.761	155.391
48.000	-20.702	-4.036	40.577	45.731	24.155	-149.938	7.474	152.055	24.678	-150.038	-0.423	152.055
50.000	-48.385	37.032	81.432	101.704	139.703	-302.279	11.206	333.190	138.788	-302.255	-19.874	333.190
52.000	20.899	8.412	77.489	80.698	86.870	-320.679	15.481	332.598	87.017	-320.899	-8.573	332.598
54.000	21.075	54.615	169.909	179.711	120.265	-438.481	21.646	455.190	120.241	-438.852	-12.223	455.190
56.000	15.799	22.276	164.884	167.130	113.182	-428.843	29.299	444.494	113.279	-429.795	-4.359	444.494
58.000	-70.850	-73.615	-213.885	237.035	205.076	-427.001	36.749	475.117	204.444	-428.864	3.808	475.117
60.000	213.357	-282.895	320.674	477.894	-54.947	-386.329	39.469	392.209	-56.394	-388.083	-6.237	392.209
62.000	88.426	-23.362	27.240	95.430	-20.082	-300.005	42.593	303.679	-23.448	-302.770	1.119	303.679
64.000	44.445	-34.608	-63.285	84.723	-20.168	-163.484	40.696	169.676	-26.390	-167.272	10.665	169.676
66.000	-14.123	-63.981	-141.182	155.645	35.355	-125.898	37.342	135.996	24.564	-131.014	26.957	135.996
68.000	-37.438	-24.558	-71.335	84.222	82.975	-179.967	36.180	201.450	67.795	-186.059	36.984	201.450
70.000	-38.031	-65.178	-38.947	84.920	90.968	-166.521	37.777	193.473	68.935	-174.339	47.809	193.473
72.000	-58.153	-115.779	-40.978	135.889	65.468	-76.947	39.363	108.427	34.892	-86.935	54.600	108.427
74.000	-73.956	-128.701	-33.634	152.200	71.790	-104.065	35.913	131.427	39.042	-114.114	52.218	131.427
76.000	-108.913	-153.734	-12.238	188.802	85.877	-134.273	35.927	163.386	46.823	-145.589	57.501	163.386
78.000	-133.289	-152.796	3.154	202.787	80.966	-120.334	37.667	149.849	34.375	-133.462	58.829	149.849
80.000	-132.606	-117.860	3.936	177.457	51.234	-55.968	40.143	85.842	-0.345	-70.727	48.645	85.842

DATE: 2 SEPT 1988

RUN DESCRIPTION: EXAMPLE 1: BASIC SLED TEST SIMULATION

TWO BELT HARNESS WITH HYPERELLIPSOID FOR DASH BOARD

VEHICLE DECELERATION: SLED ACCELERATION - 20G PEAK

CRASH VICTIM: 95TH PERCENTILE MALE

SEGMENT REL. ANGULAR VELOCITY (REV/SEC.)

TIME (MSEC)	SEGMENT NO. 3 - UT				SEGMENT NO. 5 - H				SEGMENT NO. 5 - H			
	IN VEH		REFERENCE	RES	IN VEH		REFERENCE	RES	IN H		REFERENCE	RES
X	Y	Z	X		Y	Z	X		Y	Z		
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
2.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
4.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
6.000	0.000	0.000	0.000	0.000	0.000	-0.001	0.000	0.001	0.000	-0.001	0.000	0.001
8.000	0.000	0.001	0.000	0.001	0.000	-0.001	0.000	0.001	0.000	-0.001	0.000	0.001
10.000	0.000	0.003	0.001	0.003	0.001	-0.002	0.000	0.002	0.001	-0.002	0.000	0.002
12.000	0.000	0.006	0.002	0.006	0.001	-0.003	0.000	0.003	0.001	-0.003	0.000	0.003
14.000	0.000	0.009	0.003	0.010	0.002	-0.004	0.000	0.004	0.002	-0.004	0.000	0.004
16.000	0.000	0.014	0.004	0.014	0.002	-0.006	0.000	0.006	0.002	-0.006	0.000	0.006
18.000	0.000	0.019	0.005	0.019	0.003	-0.009	0.000	0.009	0.003	-0.009	0.000	0.009
20.000	0.000	0.024	0.006	0.025	0.004	-0.013	0.000	0.013	0.004	-0.013	0.000	0.013
22.000	0.001	0.031	0.009	0.032	0.005	-0.018	-0.001	0.019	0.005	-0.018	0.001	0.019
24.000	-0.012	0.044	0.021	0.050	0.016	-0.031	-0.003	0.035	0.017	-0.031	0.001	0.035
26.000	-0.026	0.058	0.033	0.072	0.033	-0.046	-0.007	0.057	0.034	-0.046	0.001	0.057
28.000	-0.036	0.072	0.046	0.093	0.046	-0.062	-0.009	0.077	0.047	-0.062	0.002	0.077
30.000	-0.036	0.085	0.055	0.108	0.044	-0.078	-0.007	0.089	0.044	-0.078	0.003	0.089
32.000	-0.037	0.112	0.075	0.139	0.045	-0.114	-0.006	0.123	0.045	-0.114	0.004	0.123
34.000	-0.049	0.146	0.102	0.185	0.067	-0.174	-0.010	0.187	0.067	-0.174	0.006	0.187
36.000	-0.076	0.180	0.132	0.236	0.105	-0.236	-0.016	0.258	0.106	-0.236	0.009	0.258
38.000	-0.114	0.231	0.174	0.311	0.158	-0.331	-0.025	0.367	0.159	-0.331	0.012	0.367
40.000	-0.139	0.277	0.212	0.376	0.200	-0.436	-0.031	0.481	0.202	-0.436	0.017	0.481
42.000	-0.172	0.326	0.260	0.452	0.239	-0.565	-0.035	0.614	0.240	-0.565	0.022	0.614
44.000	-0.196	0.409	0.348	0.571	0.285	-0.766	-0.041	0.818	0.287	-0.766	0.029	0.818
46.000	-0.235	0.504	0.469	0.728	0.385	-1.056	-0.056	1.125	0.388	-1.056	0.038	1.125
48.000	-0.229	0.530	0.567	0.809	0.447	-1.355	-0.060	1.428	0.449	-1.355	0.051	1.428
50.000	-0.241	0.669	0.811	1.079	0.652	-1.933	-0.092	2.042	0.655	-1.933	0.069	2.042
52.000	-0.192	0.728	0.993	1.246	0.850	-2.558	-0.116	2.698	0.854	-2.558	0.096	2.698
54.000	-0.059	0.902	1.343	1.618	1.071	-3.411	-0.144	3.578	1.075	-3.411	0.132	3.578
56.000	0.073	1.021	1.706	1.989	1.296	-4.299	-0.161	4.493	1.300	-4.297	0.183	4.493
58.000	0.050	0.985	1.756	2.014	1.581	-5.149	-0.162	5.389	1.586	-5.144	0.249	5.389
60.000	0.638	0.350	2.155	2.274	1.585	-6.117	-0.179	6.321	1.591	-6.109	0.325	6.321
62.000	0.889	0.383	2.284	2.480	1.553	-6.864	-0.186	7.040	1.563	-6.852	0.408	7.040
64.000	0.999	0.342	2.217	2.456	1.491	-7.290	-0.171	7.443	1.512	-7.271	0.492	7.443
66.000	0.984	0.240	2.039	2.276	1.477	-7.552	-0.134	7.696	1.514	-7.524	0.570	7.696
68.000	0.901	0.147	1.885	2.095	1.563	-7.860	-0.072	8.014	1.626	-7.821	0.642	8.014
70.000	0.823	0.060	1.817	1.996	1.708	-8.236	0.012	8.412	1.807	-8.184	0.716	8.412
72.000	0.766	-0.148	1.772	1.936	1.824	-8.529	0.116	8.722	1.975	-8.458	0.794	8.722
74.000	0.692	-0.414	1.729	1.908	1.879	-8.677	0.222	8.881	2.095	-8.586	0.868	8.881
76.000	0.606	-0.733	1.717	1.963	1.971	-8.960	0.328	9.180	2.256	-8.849	0.939	9.180
78.000	0.491	-1.104	1.763	2.138	2.058	-9.256	0.445	9.492	2.428	-9.120	1.013	9.492
80.000	0.353	-1.429	1.829	2.347	2.089	-9.448	0.553	9.692	2.557	-9.285	1.090	9.692

RUN DESCRIPTION: EXAMPLE 1: BASIC SLED TEST SIMULATION

TWO BELT HARNESS WITH HYPERELLIPSOID FOR DASH BOARD

PAGE: 26.01

VEHICLE DECELERATION: SLED ACCELERATION - 20G PEAK

CRASH VICTIM: 95TH PERCENTILE MALE

SEGMENT REL. ANGULAR DISPLACEMENT (DEG)

TIME (MSEC)	SEGMENT NO. 3 - UT				SEGMENT NO. 5 - H				SEGMENT NO. 5 - H			
	IN VEH REFERENCE				IN VEH REFERENCE				IN UT REFERENCE			
	YAW	PITCH	ROLL	RES	YAW	PITCH	ROLL	RES	YAW	PITCH	ROLL	RES
0.000	0.000	13.280	0.000	13.280	0.000	13.460	0.000	13.460	0.000	0.180	0.000	0.180
2.000	0.000	13.280	0.000	13.280	0.000	13.460	0.000	13.460	0.000	0.180	0.000	0.180
4.000	0.000	13.280	0.000	13.280	0.000	13.460	0.000	13.460	0.000	0.180	0.000	0.180
6.000	0.000	13.280	0.000	13.280	0.000	13.459	0.000	13.459	0.000	0.180	0.000	0.180
8.000	0.000	13.280	0.000	13.280	0.000	13.459	0.000	13.459	0.000	0.179	0.000	0.179
10.000	0.001	13.281	0.000	13.281	0.000	13.458	0.000	13.458	-0.001	0.176	0.000	0.176
12.000	0.001	13.284	0.000	13.284	0.000	13.456	0.001	13.456	-0.001	0.171	0.001	0.172
14.000	0.003	13.290	0.000	13.290	0.000	13.454	0.002	13.454	-0.003	0.164	0.003	0.164
16.000	0.005	13.298	0.000	13.298	0.000	13.450	0.003	13.450	-0.005	0.152	0.005	0.152
18.000	0.008	13.310	0.000	13.310	0.000	13.445	0.005	13.445	-0.008	0.135	0.007	0.136
20.000	0.012	13.325	0.000	13.325	0.000	13.437	0.008	13.437	-0.011	0.112	0.010	0.113
22.000	0.018	13.345	0.001	13.345	0.001	13.426	0.011	13.426	-0.016	0.081	0.014	0.084
24.000	0.027	13.372	-0.003	13.372	0.001	13.409	0.018	13.409	-0.025	0.037	0.027	0.052
26.000	0.043	13.409	-0.017	13.409	0.002	13.381	0.036	13.382	-0.040	-0.027	0.063	0.079
28.000	0.067	13.456	-0.042	13.456	0.003	13.343	0.067	13.343	-0.062	-0.113	0.124	0.179
30.000	0.097	13.513	-0.069	13.513	0.005	13.293	0.101	13.293	-0.089	-0.220	0.191	0.305
32.000	0.136	13.582	-0.095	13.583	0.008	13.227	0.132	13.228	-0.126	-0.354	0.257	0.455
34.000	0.193	13.675	-0.127	13.677	0.011	13.123	0.173	13.124	-0.178	-0.552	0.343	0.673
36.000	0.265	13.791	-0.175	13.795	0.017	12.978	0.237	12.980	-0.245	-0.813	0.471	0.970
38.000	0.359	13.939	-0.243	13.946	0.023	12.775	0.333	12.779	-0.332	-1.162	0.657	1.374
40.000	0.476	14.123	-0.336	14.136	0.032	12.500	0.466	12.508	-0.443	-1.620	0.910	1.907
42.000	0.617	14.341	-0.449	14.354	0.042	12.140	0.628	12.156	-0.579	-2.196	1.220	2.572
44.000	0.797	14.601	-0.582	14.638	0.055	11.672	0.813	11.700	-0.757	-2.920	1.583	3.398
46.000	1.052	14.935	-0.739	14.997	0.068	11.017	1.061	11.067	-1.018	-3.903	2.055	4.511
48.000	1.383	15.315	-0.905	15.414	0.082	10.149	1.367	10.240	-1.367	-5.141	2.620	5.902
50.000	1.837	15.754	-1.063	15.912	0.093	8.975	1.755	9.144	-1.860	-6.738	3.301	7.683
52.000	2.445	16.263	-1.213	16.515	0.095	7.361	2.304	7.712	-2.548	-8.837	4.197	10.024
54.000	3.262	16.853	-1.279	17.247	0.077	5.221	2.994	6.017	-3.499	-11.531	5.240	12.994
56.000	4.376	17.548	-1.225	18.167	0.024	2.443	3.846	4.556	-4.830	-14.954	6.468	16.740
58.000	5.730	18.282	-1.076	19.232	-0.080	-0.956	4.867	4.960	-6.520	-19.021	7.925	21.188
60.000	7.198	18.725	-0.796	20.112	-0.263	-5.032	6.043	7.858	-8.457	-23.473	9.510	26.030
62.000	9.010	18.894	-0.167	20.927	-0.545	-9.713	7.216	12.079	-10.799	-28.287	10.903	31.183
64.000	10.887	19.035	0.596	21.863	-0.927	-14.811	8.405	16.984	-13.329	-33.526	12.297	36.700
66.000	12.684	19.097	1.389	22.806	-1.407	-20.134	9.632	22.235	-15.925	-38.944	13.770	42.339
68.000	14.326	19.070	2.120	23.692	-2.012	-25.641	10.994	27.755	-18.620	-44.470	15.523	48.056
70.000	15.868	18.985	2.772	24.554	-2.808	-31.383	12.610	33.588	-21.668	-50.163	17.827	53.960
72.000	17.346	18.800	3.344	25.369	-3.876	-37.343	14.579	39.712	-25.358	-55.941	20.942	60.003
74.000	18.753	18.447	3.806	26.080	-5.305	-43.404	16.970	46.004	-30.054	-61.584	25.198	66.009
76.000	20.091	17.906	4.127	26.679	-7.292	-49.559	19.979	52.473	-36.534	-66.989	31.360	71.966
78.000	21.384	17.148	4.271	27.175	-10.213	-55.825	23.994	59.178	-46.425	-71.980	41.030	77.911
80.000	22.660	16.188	4.207	27.614	-14.704	-62.089	29.648	66.073	-62.537	-76.081	56.973	83.815

DATE: 2 SEPT 1988

RUN DESCRIPTION: EXAMPLE 1: BASIC SLED TEST SIMULATION
TWO BELT HARNESS WITH HYPERELLIPSOID FOR DASH BOARD

PAGE: 27.01

VEHICLE DECELERATION: SLED ACCELERATION - 20G PEAK
CRASH VICTIM: 95TH PERCENTILE MALE

JOINT PARAMETERS

TIME (MSEC)	STATE	JOINT NO. 3 - NP			TOTAL TORQUE (IN. LB.)			STATE	JOINT NO. 4 - HP			TOTAL TORQUE (IN. LB.)		
		IPIN	FLEXURE	AN'LES (DEG) AZIMUTH TORSION	SPRING	VISCOUS	RES.		IPIN	FLEXURE	AN'LES (DEG) AZIMUTH TORSION	SPRING	VISCOUS	RES.
0.000	0.	10.180	0.000	0.000	0.000	0.000	0.000	0.	10.000	0.000	0.000	0.000	0.000	0.000
2.000	0.	10.181	0.000	0.000	0.000	0.118	0.118	0.	9.999	0.000	0.000	0.000	0.118	0.118
4.000	0.	10.184	0.000	0.000	0.000	0.190	0.190	0.	9.996	0.000	0.000	0.000	0.194	0.194
6.000	0.	10.189	0.000	0.000	0.000	0.241	0.241	0.	9.991	0.000	0.000	0.000	0.263	0.263
8.000	0.	10.194	-0.001	0.000	0.000	0.224	0.224	0.	9.985	0.001	0.000	0.000	0.303	0.303
10.000	0.	10.197	-0.005	0.000	0.000	0.115	0.115	0.	9.979	0.002	0.000	0.000	0.284	0.284
12.000	0.	10.197	-0.013	-0.001	0.000	0.166	0.166	0.	9.974	0.005	0.000	0.000	0.187	0.187
14.000	0.	10.192	-0.026	-0.002	0.000	0.401	0.401	0.	9.971	0.009	-0.001	0.000	0.112	0.112
16.000	0.	10.182	-0.042	-0.003	0.000	0.680	0.680	0.	9.970	0.012	-0.002	0.000	0.069	0.069
18.000	0.	10.166	-0.061	-0.004	0.000	0.977	0.977	0.	9.969	0.013	-0.003	0.000	0.072	0.072
20.000	0.	10.144	-0.081	-0.006	0.000	1.281	1.281	0.	9.968	0.012	-0.004	0.000	0.104	0.104
22.000	0.	10.115	-0.104	-0.009	0.000	1.728	1.728	0.	9.967	0.009	-0.006	0.000	0.122	0.122
24.000	0.	10.068	-0.222	-0.013	0.000	3.586	3.586	0.	9.970	0.046	-0.009	0.000	0.771	0.771
26.000	0.	9.997	-0.530	-0.019	0.000	5.256	5.256	0.	9.977	0.134	-0.016	0.000	0.981	0.981
28.000	0.	9.907	-0.984	-0.026	0.000	6.009	6.009	0.	9.981	0.211	-0.024	0.000	0.458	0.458
30.000	0.	9.805	-1.332	-0.039	0.000	6.012	6.012	0.	9.978	0.136	-0.034	0.000	1.331	1.331
32.000	0.	9.671	-1.592	-0.058	0.000	9.657	9.657	0.	9.978	-0.039	-0.045	0.000	1.704	1.704
34.000	0.	9.440	-2.129	-0.084	0.000	14.463	14.463	0.	10.014	-0.105	-0.062	0.000	2.267	2.267
36.000	0.	9.152	-3.083	-0.113	0.000	18.782	18.782	0.	10.046	-0.075	-0.088	0.000	2.371	2.371
38.000	0.	8.767	-4.576	-0.147	0.000	26.352	26.352	0.	10.095	-0.005	-0.122	0.000	3.606	3.606
40.000	0.	8.280	-6.692	-0.188	0.000	31.362	31.362	0.	10.149	0.007	-0.163	0.000	3.348	3.348
42.000	0.	7.699	-9.366	-0.240	0.000	37.757	37.757	0.	10.197	-0.171	-0.212	0.000	4.202	4.202
44.000	0.	6.972	-12.870	-0.317	0.000	52.652	52.652	0.	10.265	-0.573	-0.269	0.000	7.887	7.887
46.000	0.	5.966	-19.595	-0.430	0.000	71.325	71.325	0.	10.445	-0.726	-0.352	0.000	11.132	11.132
48.000	0.	4.869	-30.859	-0.581	0.000	80.360	80.360	0.	10.627	-1.060	-0.464	0.000	10.999	10.999
50.000	0.	3.822	-54.577	-0.804	0.000	125.388	125.388	0.	10.957	-1.213	-0.617	0.000	25.143	25.143
52.000	0.	4.001	-95.947	-1.088	0.000	150.487	150.487	0.	11.421	-0.891	-0.839	0.000	25.934	25.934
54.000	0.	6.124	-127.683	-1.502	0.000	193.882	193.882	0.	11.949	-0.784	-1.122	0.000	32.618	32.618
56.000	0.	9.668	-143.883	-2.084	0.000	220.529	220.529	0.	12.423	-0.721	-1.497	0.000	27.056	27.056
58.000	0.	13.878	-152.083	-2.748	0.000	235.094	235.094	0.	12.563	-0.690	-1.964	0.000	26.095	26.095
60.000	0.	18.379	-157.625	-3.482	0.000	234.042	234.042	0.	12.544	-1.090	-2.446	0.000	28.518	28.518
62.000	0.	22.709	-162.883	-4.505	0.000	238.449	238.449	0.	12.127	-2.354	-2.954	0.000	43.769	43.769
64.000	0.	26.899	-167.023	-5.484	18.036	216.552	233.247	0.	11.096	-3.358	-3.457	0.000	73.428	73.428
66.000	0.	30.558	-170.082	-6.297	154.456	176.484	324.673	0.	9.293	-4.188	-3.894	0.000	115.796	115.796
68.000	0.	33.331	-172.228	-6.882	347.062	133.565	462.542	0.	6.439	-6.114	-4.250	0.000	175.281	175.281
70.000	0.	35.027	-173.783	-7.296	502.714	67.615	559.233	0.	2.361	-19.607	-4.534	0.000	244.357	244.357
72.000	0.	35.552	-174.978	-7.592	493.636	30.690	494.729	0.	3.448	-162.475	-4.760	0.000	303.225	303.225
74.000	0.	35.540	-177.910	-7.807	466.062	24.286	466.652	0.	9.415	-172.198	-4.909	0.000	300.392	300.392
76.000	0.	35.544	-175.681	-7.960	539.761	20.915	540.627	0.	15.394	-174.585	-4.988	0.000	299.780	299.780
78.000	0.	35.568	-177.364	-8.079	558.450	19.211	561.920	0.	21.358	-175.662	-5.027	0.000	295.975	295.975
80.000	0.	35.764	-178.027	-8.198	579.271	26.658	596.493	0.	27.109	-176.275	-5.046	22.243	278.693	300.921

DATE: 2 SEPT 1988
 RUN DESCRIPTION: EXAMPLE 1: BASIC SLED TEST SIMULATION
 TWO BELT HARNESS WITH HYPERELLIPSOID FOR DASH BOARD
 VEHICLE DECELERATION: SLED ACCELERATION - 20G PEAK
 CRASH VICTIM: 95TH PERCENTILE MALE

PAGE: 28.01

HP JOINT FORCES & TORQUES ON H IN H REFERENCE

TIME (MSEC)	JOINT FORCE (LB. 10**2)			JOINT TORQUE (IN.- LB. 10**2)		
	X	Y	Z	X	Y	Z
0.000	0.000	0.000	-0.126	0.000E+00	0.000E+00	0.000E+00
2.000	0.000	0.000	-0.129	-0.119E-06	0.118E-02	0.232E-07
4.000	0.001	0.000	-0.136	-0.939E-07	0.194E-02	0.765E-06
6.000	0.002	0.000	-0.142	0.273E-04	0.263E-02	0.587E-05
8.000	0.002	0.000	-0.148	0.116E-03	0.303E-02	0.258E-04
10.000	0.001	-0.001	-0.158	0.238E-03	0.283E-02	0.725E-04
12.000	-0.001	-0.001	-0.171	0.385E-03	0.183E-02	0.160E-03
14.000	-0.003	-0.001	-0.181	0.383E-03	0.101E-02	0.284E-03
16.000	-0.005	-0.001	-0.190	0.297E-03	0.458E-03	0.427E-03
18.000	-0.005	0.000	-0.202	0.148E-03	0.370E-03	0.601E-03
20.000	-0.007	0.000	-0.210	0.503E-05	0.665E-03	0.800E-03
22.000	-0.010	-0.001	-0.218	0.670E-04	0.574E-03	0.107E-02
24.000	-0.024	-0.021	-0.273	0.675E-02	-0.325E-02	0.181E-02
26.000	-0.027	-0.028	-0.270	0.883E-02	-0.301E-02	0.305E-02
28.000	-0.026	-0.002	-0.254	0.106E-02	-0.444E-03	0.444E-02
30.000	-0.029	0.004	-0.263	-0.115E-01	0.330E-02	0.586E-02
32.000	-0.097	-0.027	-0.164	-0.105E-01	-0.109E-01	0.776E-02
34.000	-0.090	-0.026	-0.153	-0.180E-02	-0.198E-01	0.108E-01
36.000	-0.123	-0.043	-0.131	0.560E-02	-0.182E-01	0.141E-01
38.000	-0.167	-0.086	-0.217	0.121E-01	-0.287E-01	0.182E-01
40.000	-0.187	-0.065	-0.200	-0.435E-02	-0.242E-01	0.227E-01
42.000	-0.235	-0.067	-0.280	-0.207E-01	-0.239E-01	0.277E-01
44.000	-0.440	-0.151	0.231	-0.200E-01	-0.675E-01	0.356E-01
46.000	-0.476	-0.133	0.180	-0.666E-02	-0.992E-01	0.500E-01
48.000	-0.484	-0.078	0.423	-0.409E-01	-0.785E-01	0.653E-01
50.000	-0.964	-0.400	1.929	0.475E-01	-0.229E+00	0.927E-01
52.000	-1.025	-0.250	2.650	0.333E-01	-0.227E+00	0.122E+00
54.000	-1.406	-0.348	4.626	0.417E-01	-0.281E+00	0.161E+00
56.000	-1.393	-0.331	5.514	0.379E-01	-0.168E+00	0.208E+00
58.000	-1.416	-0.595	5.141	0.910E-01	0.106E-01	0.244E+00
60.000	-1.296	0.123	7.496	-0.979E-01	0.101E+00	0.248E+00
62.000	-1.049	0.016	5.739	-0.807E-01	0.339E+00	0.264E+00
64.000	-0.656	0.016	3.853	-0.406E-01	0.692E+00	0.241E+00
66.000	-0.606	-0.153	1.823	-0.400E-01	0.114E+01	0.202E+00
68.000	-0.887	-0.314	1.048	-0.123E+00	0.174E+01	0.171E+00
70.000	-0.958	-0.369	1.210	-0.259E+00	0.242E+01	0.156E+00
72.000	-0.758	-0.324	1.329	-0.380E+00	0.300E+01	0.143E+00
74.000	-0.842	-0.344	1.491	-0.344E+00	0.298E+01	0.897E-01
76.000	-0.939	-0.393	2.383	-0.340E+00	0.298E+01	0.605E-01
78.000	-0.884	-0.387	3.033	-0.337E+00	0.294E+01	0.453E-01
80.000	-0.676	-0.312	3.738	-0.352E+00	0.299E+01	0.461E-01

RUN DESCRIPTION: EXAMPLE 1: BASIC SLED TEST SIMULATION
TWO BELT HARNESS WITH HYPERELLIPSOID FOR DASH BOARD

PAGE: 29.01

VEHICLE DECELERATION: SLED ACCELERATION - 20G PEAK
CRASH VICTIM: 95TH PERCENTILE MALE

BODY PROPERTIES - REFERENCE SEGMENT NO. 16 (VEH)

INCLUDED SEGMENT NOS: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

TIME (MSEC)	CENTER OF GRAVITY (IN.)			LINEAR MOMENTUM (LB.-SEC.)			ANGULAR MOMENTUM (IN.- LB.-SEC.)			KINETIC ENERGY (LB.- IN.)		
	X	Y	Z	X	Y	Z	X	Y	Z	LINEAR	ANGULAR	TOTAL
0.000	18.183	0.000	-20.231	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
2.000	18.183	0.000	-20.231	-0.252E-01	0.287E-06	0.723E-04	-0.699E-04	0.197E-01	0.304E-03	0.382E-01	0.495E-02	0.431E-01
4.000	18.185	0.000	-20.231	-0.121E+00	-0.220E-04	-0.665E-02	-0.641E-03	0.993E+00	0.158E-02	0.505E+00	0.344E-01	0.539E+00
6.000	18.189	0.000	-20.231	-0.299E+00	-0.442E-03	-0.326E-01	-0.121E-01	0.341E+01	0.154E-02	0.242E+01	0.137E+00	0.256E+01
8.000	18.197	0.000	-20.231	-0.543E+00	-0.201E-02	-0.842E-01	-0.554E-01	0.730E+01	-0.472E-02	0.754E+01	0.391E+00	0.793E+01
10.000	18.210	0.000	-20.232	-0.841E+00	-0.497E-02	-0.167E+00	-0.138E+00	0.127E+02	-0.181E-01	0.184E+02	0.862E+00	0.192E+02
12.000	18.230	0.000	-20.233	-0.118E+01	-0.991E-02	-0.283E+00	-0.275E+00	0.194E+02	-0.448E-01	0.384E+02	0.155E+01	0.399E+02
14.000	18.257	0.000	-20.234	-0.155E+01	-0.150E-01	-0.435E+00	-0.416E+00	0.274E+02	-0.749E-01	0.717E+02	0.247E+01	0.742E+02
16.000	18.294	0.000	-20.236	-0.195E+01	-0.197E-01	-0.622E+00	-0.546E+00	0.366E+02	-0.995E-01	0.123E+03	0.365E+01	0.127E+03
18.000	18.342	0.000	-20.239	-0.239E+01	-0.243E-01	-0.844E+00	-0.671E+00	0.469E+02	-0.121E+00	0.200E+03	0.515E+01	0.205E+03
20.000	18.402	0.000	-20.242	-0.287E+01	-0.286E-01	-0.110E+01	-0.789E+00	0.584E+02	-0.134E+00	0.307E+03	0.698E+01	0.314E+03
22.000	18.476	0.000	-20.247	-0.340E+01	-0.356E-01	-0.140E+01	-0.982E+00	0.713E+02	-0.157E+00	0.451E+03	0.923E+01	0.461E+03
24.000	18.564	-0.001	-20.253	-0.401E+01	-0.350E-01	-0.183E+01	-0.164E+01	0.880E+02	0.204E+00	0.641E+03	0.118E+02	0.653E+03
26.000	18.669	-0.001	-20.260	-0.463E+01	-0.365E-01	-0.227E+01	-0.252E+01	0.105E+03	0.607E+00	0.886E+03	0.146E+02	0.901E+03
28.000	18.792	-0.001	-20.270	-0.528E+01	-0.204E-02	-0.270E+01	-0.226E+01	0.122E+03	0.141E+01	0.120E+04	0.179E+02	0.122E+04
30.000	18.933	-0.001	-20.280	-0.598E+01	0.973E-01	-0.311E+01	0.463E+00	0.139E+03	0.286E+01	0.158E+04	0.216E+02	0.160E+04
32.000	19.095	0.000	-20.292	-0.691E+01	0.132E+00	-0.341E+01	0.168E+01	0.160E+03	0.401E+01	0.204E+04	0.264E+02	0.206E+04
34.000	19.278	0.000	-20.306	-0.820E+01	0.583E-01	0.369E+01	-0.240E+00	0.189E+03	0.455E+01	0.256E+04	0.343E+02	0.259E+04
36.000	19.481	0.000	-20.320	-0.988E+01	0.265E-01	-0.410E+01	-0.155E+01	0.226E+03	0.628E+01	0.315E+04	0.483E+02	0.320E+04
38.000	19.704	0.001	-20.336	-0.122E+02	0.801E-01	-0.454E+01	-0.166E-01	0.277E+03	0.101E+02	0.379E+04	0.721E+02	0.386E+04
40.000	19.948	0.001	-20.354	-0.149E+02	0.324E+00	-0.494E+01	0.272E+01	0.333E+03	0.171E+02	0.449E+04	0.104E+03	0.460E+04
42.000	20.212	0.003	-20.373	-0.180E+02	0.767E+00	-0.541E+01	0.108E+02	0.400E+03	0.278E+02	0.520E+04	0.136E+03	0.533E+04
44.000	20.492	0.007	-20.393	-0.220E+02	0.103E+01	-0.537E+01	0.138E+02	0.483E+03	0.377E+02	0.576E+04	0.166E+03	0.593E+04
46.000	20.782	0.010	-20.412	-0.272E+02	0.825E+00	-0.462E+01	0.280E+01	0.583E+03	0.427E+02	0.611E+04	0.192E+03	0.631E+04
48.000	21.078	0.014	-20.426	-0.336E+02	0.103E+01	-0.313E+01	-0.118E+01	0.699E+03	0.510E+02	0.623E+04	0.206E+03	0.644E+04
50.000	21.369	0.015	-20.431	-0.431E+02	-0.477E+00	0.120E+01	-0.558E+02	0.860E+03	0.409E+02	0.588E+04	0.251E+03	0.614E+04
52.000	21.644	0.010	-20.416	-0.545E+02	-0.246E+01	0.687E+01	-0.122E+03	0.104E+04	0.198E+02	0.539E+04	0.363E+03	0.575E+04
54.000	21.894	-0.005	-20.377	-0.685E+02	-0.545E+01	0.145E+02	-0.211E+03	0.126E+04	-0.927E+01	0.491E+04	0.673E+03	0.558E+04
56.000	22.111	-0.030	-20.309	-0.834E+02	-0.852E+01	0.220E+02	-0.295E+03	0.151E+04	-0.346E+02	0.471E+04	0.104E+04	0.574E+04
58.000	22.290	-0.069	-20.216	-0.991E+02	-0.135E+02	0.279E+02	-0.411E+03	0.179E+04	-0.797E+02	0.476E+04	0.128E+04	0.604E+04
60.000	22.406	-0.126	-20.096	-0.125E+03	-0.182E+02	0.375E+02	-0.524E+03	0.227E+04	-0.120E+03	0.581E+04	0.201E+04	0.782E+04
62.000	22.463	-0.202	-19.952	-0.140E+03	-0.221E+02	0.392E+02	-0.608E+03	0.259E+04	-0.146E+03	0.655E+04	0.226E+04	0.881E+04
64.000	22.487	-0.287	-19.810	-0.150E+03	-0.237E+02	0.370E+02	-0.647E+03	0.284E+04	-0.148E+03	0.673E+04	0.223E+04	0.896E+04
66.000	22.490	-0.376	-19.683	-0.158E+03	-0.241E+02	0.311E+02	-0.659E+03	0.306E+04	-0.145E+03	0.640E+04	0.206E+04	0.846E+04
68.000	22.476	-0.465	-19.584	-0.166E+03	-0.239E+02	0.220E+02	-0.658E+03	0.329E+04	-0.144E+03	0.579E+04	0.189E+04	0.768E+04
70.000	22.446	-0.552	-19.523	-0.171E+03	-0.235E+02	0.114E+02	-0.651E+03	0.348E+04	-0.141E+03	0.524E+04	0.177E+04	0.700E+04
72.000	22.406	-0.640	-19.495	-0.175E+03	-0.237E+02	0.416E+01	-0.648E+03	0.360E+04	-0.142E+03	0.484E+04	0.164E+04	0.648E+04
74.000	22.358	-0.728	-19.490	-0.180E+03	-0.238E+02	-0.130E+01	-0.646E+03	0.369E+04	-0.147E+03	0.453E+04	0.150E+04	0.604E+04
76.000	22.298	-0.816	-19.505	-0.183E+03	-0.240E+02	-0.635E+01	-0.648E+03	0.377E+04	-0.156E+03	0.443E+04	0.143E+04	0.587E+04
78.000	22.233	-0.905	-19.537	-0.185E+03	-0.242E+02	-0.114E+02	-0.655E+03	0.383E+04	-0.166E+03	0.456E+04	0.146E+04	0.602E+04
80.000	22.165	-0.995	-19.588	-0.186E+03	-0.244E+02	-0.161E+02	-0.659E+03	0.386E+04	-0.170E+03	0.477E+04	0.155E+04	0.632E+04

DATE: 2 SEPT 1988

RUN DESCRIPTION: EXAMPLE 1: BASIC SLED TEST SIMULATION
TWO BELT HARNESS WITH HYPERELLIPSOID FOR DASH BOARD

PAGE: 30.01

VEHICLE DECELERATION: SLED ACCELERATION - 20G PEAK
CRASH VICTIM: 95TH PERCENTILE MALE
CONTACT FORCES - SEGMENT PANELS VS. SEGMENTS

TIME (MSEC)	PANEL 1 (SEAT. 6 DEGREE OFF H) VS. SEGMENT 1 (LT)						PANEL 1 (SEAT. 6 DEGREE OFF H) VS. SEGMENT 6 (RUL)							
	DEFL- ECTION (IN.)	NORMAL FORCE (LB.)	FRICTION FORCE (LB.)	RESULTANT FORCE (LB.)	CONTACT LOCATION (IN.) (VEH REFERENCE)			DEFL- ECTION (IN.)	NORMAL FORCE (LB.)	FRICTION FORCE (LB.)	RESULTANT FORCE (LB.)	CONTACT LOCATION (IN.) (VEH REFERENCE)		
					X	Y	Z					X	Y	Z
0.000	0.458	137.10	0.00	137.10	14.372	0.000	-10.459	0.250	29.94	0.00	29.94	24.725	3.420	-11.545
2.000	0.458	137.02	8.94	137.31	14.373	0.000	-10.459	0.250	31.22	4.99	31.62	24.725	3.420	-11.545
4.000	0.458	136.86	27.51	139.60	14.375	0.000	-10.459	0.250	36.69	15.37	39.79	24.726	3.420	-11.545
6.000	0.458	136.72	46.60	144.44	14.380	0.000	-10.460	0.250	44.22	22.11	49.44	24.727	3.420	-11.545
8.000	0.457	138.78	64.83	153.18	14.390	0.000	-10.461	0.251	51.70	25.85	57.80	24.729	3.420	-11.546
10.000	0.458	144.58	72.29	161.65	14.407	0.000	-10.462	0.251	58.35	29.18	65.24	24.732	3.420	-11.546
12.000	0.458	152.48	76.24	170.48	14.431	0.000	-10.465	0.252	63.83	31.91	71.36	24.737	3.420	-11.546
14.000	0.458	161.71	80.85	180.80	14.464	0.000	-10.468	0.253	68.37	34.18	76.44	24.744	3.420	-11.547
16.000	0.459	172.39	86.19	192.73	14.508	0.000	-10.473	0.254	72.13	36.06	80.64	24.755	3.420	-11.548
18.000	0.459	184.02	92.01	205.74	14.565	-0.001	-10.479	0.255	75.12	37.56	83.99	24.770	3.420	-11.550
20.000	0.460	196.74	98.37	219.96	14.635	-0.001	-10.486	0.256	76.24	38.12	85.24	24.790	3.420	-11.552
22.000	0.461	211.45	105.72	236.41	14.720	-0.001	-10.495	0.257	74.47	37.23	83.26	24.816	3.420	-11.555
24.000	0.463	200.50	100.25	224.17	14.822	-0.001	-10.506	0.257	66.45	33.22	74.29	24.849	3.420	-11.558
26.000	0.464	199.45	99.72	222.99	14.942	-0.001	-10.519	0.257	58.64	29.32	65.56	24.891	3.420	-11.563
28.000	0.465	207.89	103.94	232.42	15.081	0.000	-10.533	0.256	49.90	24.95	55.79	24.942	3.420	-11.568
30.000	0.466	228.27	114.13	255.21	15.242	0.001	-10.550	0.255	45.48	22.74	50.85	25.002	3.420	-11.574
32.000	0.468	279.53	139.76	312.52	15.424	0.006	-10.569	0.253	53.74	26.87	60.08	25.069	3.421	-11.581
34.000	0.472	356.68	178.34	398.78	15.630	0.016	-10.591	0.251	53.97	26.99	60.35	25.138	3.421	-11.589
36.000	0.478	448.62	224.31	501.57	15.859	0.038	-10.615	0.248	38.40	19.20	42.93	25.207	3.421	-11.596
38.000	0.485	560.72	280.36	626.90	16.111	0.079	-10.641	0.244	37.83	18.91	42.29	25.272	3.422	-11.603
40.000	0.496	634.27	317.14	709.14	16.385	0.146	-10.670	0.239	65.17	32.59	72.87	25.325	3.424	-11.608
42.000	0.510	664.96	332.48	743.44	16.679	0.237	-10.701	0.234	89.40	44.70	99.96	25.364	3.428	-11.612
44.000	0.527	703.41	351.70	786.43	16.985	0.346	-10.733	0.230	119.38	59.69	133.47	25.391	3.434	-11.615
46.000	0.551	756.42	378.21	845.71	17.293	0.460	-10.765	0.227	157.62	78.81	176.23	25.402	3.441	-11.616
48.000	0.585	828.62	414.31	926.43	17.592	0.567	-10.797	0.225	201.14	100.57	224.88	25.394	3.451	-11.615
50.000	0.634	922.91	461.45	1031.84	17.867	0.638	-10.826	0.225	230.71	115.35	257.94	25.366	3.464	-11.613
52.000	0.705	1043.08	521.54	1166.20	18.102	0.630	-10.850	0.225	229.76	114.88	256.88	25.326	3.480	-11.608
54.000	0.806	1090.42	545.21	1219.13	18.285	0.526	-10.869	0.227	275.31	137.65	307.80	25.267	3.497	-11.602
56.000	0.942	1085.51	542.75	1213.64	18.410	0.360	-10.883	0.237	287.36	143.68	321.28	25.149	3.514	-11.590
58.000	1.113	1255.01	627.50	1403.14	18.483	0.194	-10.890	0.263	319.11	159.55	356.77	24.927	3.526	-11.567
60.000	1.323	1450.74	725.37	1621.98	18.497	0.037	-10.892	0.312	369.55	184.78	413.17	24.589	3.526	-11.531
62.000	1.576	1827.33	913.67	2043.02	18.468	-0.132	-10.889	0.373	440.16	220.08	492.11	24.222	3.511	-11.493
64.000	1.825	2293.38	1146.69	2564.07	18.450	-0.237	-10.887	0.443	571.58	285.79	639.04	23.850	3.487	-11.453
66.000	2.046	2723.02	1361.51	3044.43	18.439	-0.287	-10.886	0.512	745.66	372.83	833.68	23.509	3.460	-11.418
68.000	2.217	3434.09	1717.05	3839.43	18.418	-0.298	-10.883	0.565	895.39	447.69	1001.07	23.230	3.435	-11.388
70.000	2.318	3801.87	1900.93	4250.62	18.370	-0.288	-10.878	0.584	304.97	152.49	340.97	23.046	3.419	-11.369
72.000	2.356	2969.04	1484.52	3319.48	18.300	-0.277	-10.871	0.565	279.61	139.80	312.61	22.958	3.408	-11.360
74.000	2.342	2947.61	1473.80	3295.52	18.218	-0.271	-10.862	0.515	213.84	106.92	239.08	22.945	3.400	-11.359
76.000	2.282	2851.26	210.71	2859.04	18.125	-0.274	-10.853	0.436	108.38	54.19	121.18	23.013	3.395	-11.366
78.000	2.180	2688.19	119.57	2690.85	18.036	-0.252	-10.843	0.328	45.26	22.78	50.93	23.161	3.390	-11.381
80.000	2.039	2461.74	235.32	2472.96	17.954	-0.332	-10.835	0.191	18.66	9.33	20.85	23.244	3.381	-11.390

DATE: 2 SEPT 1988

RUN DESCRIPTION: EXAMPLE 1: BASIC SLED TEST SIMULATION
TWO BELT HARNESS WITH HYPERELLIPSOID FOR DASH BOARD

PAGE: 31.01

VEHICLE DECELERATION: SLED ACCELERATION - 20G PEAK
CRASH VICTIM: 95TH PERCENTILE MALE
CONTACT FORCES - SEGMENT PANELS VS. SEGMENTS

TIME (MSEC)	PANEL 1 (SEAT. 6 DEGREE OFF H) VS. SEGMENT 9 (LUL)							PANEL 2 (BACK PANEL. 13 DEGR) VS. SEGMENT 1 (LT)						
	DEFL- ECTION (IN.)	NORMAL FORCE (LB.)	FRICITION FORCE (LB.)	RESULTANT FORCE (LB.)	CONTACT LOCATION (IN.) (VEH REFERENCE)			DEFL- ECTION (IN.)	NORMAL FORCE (LB.)	FRICITION FORCE (LB.)	RESULTANT FORCE (LB.)	CONTACT LOCATION (IN.) (VEH REFERENCE)		
					X	Y	Z					X	Y	Z
0.000	0.250	29.94	0.00	29.94	24.725	-3.420	-11.545	0.055	2.77	0.00	2.77	9.382	0.000	-12.674
2.000	0.250	31.23	5.00	31.62	24.725	-3.420	-11.545	0.055	2.76	0.40	2.79	9.382	0.000	-12.674
4.000	0.250	36.71	15.41	39.81	24.726	-3.420	-11.545	0.054	2.68	0.97	2.85	9.382	0.000	-12.675
6.000	0.250	44.24	22.12	49.46	24.727	-3.420	-11.545	0.050	2.51	1.25	2.80	9.382	0.000	-12.676
8.000	0.251	51.74	25.87	57.84	24.729	-3.420	-11.546	0.044	2.18	1.09	2.44	9.382	0.000	-12.677
10.000	0.251	58.41	29.21	65.30	24.732	-3.420	-11.546	0.033	1.65	0.83	1.85	9.381	0.000	-12.680
12.000	0.252	63.91	31.95	71.45	24.737	-3.420	-11.546	0.017	0.86	0.43	0.96	9.380	0.000	-12.684
14.000	0.253	68.45	34.23	76.53	24.744	-3.420	-11.547	-0.006	0.00	0.00	0.00	0.000	0.000	0.000
16.000	0.254	72.21	36.10	80.73	24.755	-3.420	-11.548	-0.036	0.00	0.00	0.00	0.000	0.000	0.000
18.000	0.255	75.19	37.60	84.07	24.770	-3.420	-11.550	-0.077	0.00	0.00	0.00	0.000	0.000	0.000
20.000	0.256	76.31	38.16	85.32	24.790	-3.420	-11.552	-0.128	0.00	0.00	0.00	0.000	0.000	0.000
22.000	0.257	74.63	37.31	83.43	24.815	-3.420	-11.555	-0.192	0.00	0.00	0.00	0.000	0.000	0.000
24.000	0.257	65.18	32.59	72.87	24.849	-3.420	-11.558	-0.269	0.00	0.00	0.00	0.000	0.000	0.000
26.000	0.257	56.40	28.20	63.06	24.891	-3.420	-11.563	-0.362	0.00	0.00	0.00	0.000	0.000	0.000
28.000	0.256	47.59	23.79	53.20	24.943	-3.420	-11.568	-0.472	0.00	0.00	0.00	0.000	0.000	0.000
30.000	0.254	34.68	17.34	38.78	25.005	-3.420	-11.575	-0.601	0.00	0.00	0.00	0.000	0.000	0.000
32.000	0.252	30.42	15.21	34.01	25.079	-3.419	-11.582	-0.749	0.00	0.00	0.00	0.000	0.000	0.000
34.000	0.248	29.66	14.83	33.16	25.168	-3.419	-11.592	-0.918	0.00	0.00	0.00	0.000	0.000	0.000
36.000	0.241	28.14	14.07	31.46	25.280	-3.419	-11.604	-1.198	0.00	0.00	0.00	0.000	0.000	0.000
38.000	0.227	25.44	12.72	28.44	25.426	-3.418	-11.619	-1.317	0.00	0.00	0.00	0.000	0.000	0.000
40.000	0.206	21.30	10.65	23.81	25.615	-3.415	-11.639	-1.546	0.00	0.00	0.00	0.000	0.000	0.000
42.000	0.179	16.89	8.45	18.89	25.850	-3.411	-11.663	-1.794	0.00	0.00	0.00	0.000	0.000	0.000
44.000	0.147	12.03	6.01	13.45	26.130	-3.405	-11.693	-2.056	0.00	0.00	0.00	0.000	0.000	0.000
46.000	0.112	6.87	3.43	7.68	26.435	-3.396	-11.725	-2.328	0.00	0.00	0.00	0.000	0.000	0.000
48.000	0.080	3.99	1.99	4.46	26.760	-3.386	-11.759	-2.601	0.00	0.00	0.00	0.000	0.000	0.000
50.000	0.056	2.78	1.39	3.11	27.008	-3.372	-11.785	-2.864	0.00	0.00	0.00	0.000	0.000	0.000
52.000	0.051	58.93	29.46	65.88	27.020	-3.355	-11.786	-3.104	0.00	0.00	0.00	0.000	0.000	0.000
54.000	0.076	84.08	42.04	94.01	26.648	-3.335	-11.747	-3.311	0.00	0.00	0.00	0.000	0.000	0.000
56.000	0.137	138.11	69.06	154.41	26.015	-3.313	-11.681	-3.479	0.00	0.00	0.00	0.000	0.000	0.000
58.000	0.228	165.60	82.80	185.14	25.312	-3.292	-11.607	-3.609	0.00	0.00	0.00	0.000	0.000	0.000
60.000	0.353	148.11	74.06	165.60	24.605	-3.280	-11.533	-3.695	0.00	0.00	0.00	0.000	0.000	0.000
62.000	0.522	357.14	178.57	399.30	23.920	-3.280	-11.461	-3.747	0.00	0.00	0.00	0.000	0.000	0.000
64.000	0.684	686.47	343.24	767.50	23.448	-3.290	-11.411	-3.784	0.00	0.00	0.00	0.000	0.000	0.000
66.000	0.819	1246.33	623.16	1393.44	23.155	-3.301	-11.380	-3.811	0.00	0.00	0.00	0.000	0.000	0.000
68.000	0.905	1699.87	849.94	1900.52	22.994	-3.313	-11.364	-3.828	0.00	0.00	0.00	0.000	0.000	0.000
70.000	0.924	758.55	379.27	848.08	22.932	-3.322	-11.357	-3.837	0.00	0.00	0.00	0.000	0.000	0.000
72.000	0.888	710.46	355.23	794.31	22.936	-3.329	-11.358	-3.838	0.00	0.00	0.00	0.000	0.000	0.000
74.000	0.814	611.98	305.99	684.21	22.992	-3.333	-11.363	-3.831	0.00	0.00	0.00	0.000	0.000	0.000
76.000	0.713	477.86	238.93	534.26	23.096	-3.334	-11.374	-3.811	0.00	0.00	0.00	0.000	0.000	0.000
78.000	0.591	314.21	157.11	351.30	23.266	-3.333	-11.392	-3.793	0.00	0.00	0.00	0.000	0.000	0.000
80.000	0.450	126.56	63.28	141.50	23.529	-3.333	-11.420	-3.762	0.00	0.00	0.00	0.000	0.000	0.000

RUN DESCRIPTION: EXAMPLE 1: BASIC SLED TEST SIMULATION
 TWO BELT HARNESS WITH HYPERELLIPSOID FOR DASH BOARD

VEHICLE DECELERATION: SLED ACCELERATION - 20G PEAK
 CRASH VICTIM: 95TH PERCENTILE MALE
 CONTACT FORCES - SEGMENT PANELS VS. SEGMENTS

TIME (MSEC)	PANEL 2 (BACK PANEL. 13 DEGR) VS. SEGMENT 2 (CT)							PANEL 2 (BACK PANEL. 13 DEGR) VS. SEGMENT 3 (UT)						
	DEFL- ECTION	NORMAL FORCE	FRICTION FORCE	RESULTANT FORCE	CONTACT LOCATION (IN.)			DEFL- ECTION	NORMAL FORCE	FRICTION FORCE	RESULTANT FORCE	CONTACT LOCATION (IN.)		
	(IN.)	(LB.)	(LB.)	(LB.)	X	Y	Z	(IN.)	(LB.)	(LB.)	(LB.)	X	Y	Z
0.000	-0.312	0.00	0.00	0.00	0.000	0.000	0.000	0.133	9.90	0.00	9.90	6.151	0.000	-26.665
2.000	-0.312	0.00	0.00	0.00	0.000	0.000	0.000	9.132	9.87	0.06	9.87	6.151	0.000	-26.665
4.000	-0.314	0.00	0.00	0.00	0.000	0.000	0.000	0.131	9.61	0.75	9.64	6.151	0.000	-26.665
6.000	-0.320	0.00	0.00	0.00	0.000	0.000	0.000	0.126	8.90	1.74	9.07	6.151	0.000	-26.664
8.000	-0.330	0.00	0.00	0.00	0.000	0.000	0.000	0.117	7.52	2.55	7.94	6.152	0.000	-26.663
10.000	-0.347	0.00	0.00	0.00	0.000	0.000	0.000	0.102	5.25	2.42	5.77	6.152	0.000	-26.661
12.000	-0.372	0.00	0.00	0.00	0.000	0.000	0.000	0.079	3.95	1.98	4.42	6.153	0.000	-26.658
14.000	-0.406	0.00	0.00	0.00	0.000	0.000	0.000	0.048	2.38	1.19	2.66	6.154	0.000	-26.654
16.000	-0.452	0.00	0.00	0.00	0.000	0.000	0.000	0.006	0.30	0.15	0.33	6.155	0.000	-26.649
18.000	-0.510	0.00	0.00	0.00	0.000	0.000	0.000	-0.048	0.00	0.00	0.00	0.000	0.000	0.000
20.000	-0.583	0.00	0.00	0.00	0.000	0.000	0.000	-0.114	0.00	0.00	0.00	0.000	0.000	0.000
22.000	-0.671	0.00	0.00	0.00	0.000	0.000	0.000	-0.196	0.00	0.00	0.00	0.000	0.000	0.000
24.000	-0.776	0.00	0.00	0.00	0.000	0.000	0.000	-0.294	0.00	0.00	0.00	0.000	0.000	0.000
26.000	-0.901	0.00	0.00	0.00	0.000	0.000	0.000	-0.409	0.00	0.00	0.00	0.000	0.000	0.000
28.000	-1.045	0.00	0.00	0.00	0.000	0.000	0.000	-0.542	0.00	0.00	0.00	0.000	0.000	0.000
30.000	-1.211	0.00	0.00	0.00	0.000	0.000	0.000	-0.696	0.00	0.00	0.00	0.000	0.000	0.000
32.000	-1.400	0.00	0.00	0.00	0.000	0.000	0.000	-0.872	0.00	0.00	0.00	0.000	0.000	0.000
34.000	-1.612	0.00	0.00	0.00	0.000	0.000	0.000	-1.067	0.00	0.00	0.00	0.000	0.000	0.000
36.000	-1.849	0.00	0.00	0.00	0.000	0.000	0.000	-1.284	0.00	0.00	0.00	0.000	0.000	0.000
38.000	-2.110	0.00	0.00	0.00	0.000	0.000	0.000	-1.523	0.00	0.00	0.00	0.000	0.000	0.000
40.000	-2.397	0.00	0.00	0.00	0.000	0.000	0.000	-1.783	0.00	0.00	0.00	0.000	0.000	0.000
42.000	-2.707	0.00	0.00	0.00	0.000	0.000	0.000	-2.064	0.00	0.00	0.00	0.000	0.000	0.000
44.000	-3.034	0.00	0.00	0.00	0.000	0.000	0.000	-2.361	0.00	0.00	0.00	0.000	0.000	0.000
46.000	-3.370	0.00	0.00	0.00	0.000	0.000	0.000	-2.662	0.00	0.00	0.00	0.000	0.000	0.000
48.000	-3.702	0.00	0.00	0.00	0.000	0.000	0.000	-2.960	0.00	0.00	0.00	0.000	0.000	0.000
50.000	-4.012	0.00	0.00	0.00	0.000	0.000	0.000	-3.238	0.00	0.00	0.00	0.000	0.000	0.000
52.000	-4.276	0.00	0.00	0.00	0.000	0.000	0.000	-3.473	0.00	0.00	0.00	0.000	0.000	0.000
54.000	-4.477	0.00	0.00	0.00	0.000	0.000	0.000	-3.652	0.00	0.00	0.00	0.000	0.000	0.000
56.000	-4.606	0.00	0.00	0.00	0.000	0.000	0.000	-3.757	0.00	0.00	0.00	0.000	0.000	0.000
58.000	-4.661	0.00	0.00	0.00	0.000	0.000	0.000	-3.787	0.00	0.00	0.00	0.000	0.000	0.000
60.000	-4.582	0.00	0.00	0.00	0.000	0.000	0.000	-3.689	0.00	0.00	0.00	0.000	0.000	0.000
62.000	-4.378	0.00	0.00	0.00	0.000	0.000	0.000	-3.478	0.00	0.00	0.00	0.000	0.000	0.000
64.000	-4.177	0.00	0.00	0.00	0.000	0.000	0.000	-3.225	0.00	0.00	0.00	0.000	0.000	0.000
66.000	-3.915	0.00	0.00	0.00	0.000	0.000	0.000	-2.965	0.00	0.00	0.00	0.000	0.000	0.000
68.000	-3.689	0.00	0.00	0.00	0.000	0.000	0.000	-2.720	0.00	0.00	0.00	0.000	0.000	0.000
70.000	-3.476	0.00	0.00	0.00	0.000	0.000	0.000	-2.499	0.00	0.00	0.00	0.000	0.000	0.000
72.000	-3.281	0.00	0.00	0.00	0.000	0.000	0.000	-2.303	0.00	0.00	0.00	0.000	0.000	0.000
74.000	-3.098	0.00	0.00	0.00	0.000	0.000	0.000	-2.127	0.00	0.00	0.00	0.000	0.000	0.000
76.000	-2.921	0.00	0.00	0.00	0.000	0.000	0.000	-1.960	0.00	0.00	0.00	0.000	0.000	0.000
78.000	-2.744	0.00	0.00	0.00	0.000	0.000	0.000	-1.791	0.00	0.00	0.00	0.000	0.000	0.000
80.000	-2.561	0.00	0.00	0.00	0.000	0.000	0.000	-1.618	0.00	0.00	0.00	0.000	0.000	0.000

DATE: 2 SEPT 1988

RUN DESCRIPTION: EXAMPLE 1: BASIC SLED TEST SIMULATION
TWO BELT HARNESS WITH HYPERELLIPSOID FOR DASH BOARD
VEHICLE DECELERATION: SLED ACCELERATION - 20G PEAK
CRASH VICTIM: 95TH PERCENTILE MALE
CONTACT FORCES - SEGMENT PANELS VS. SEGMENTS

PAGE: 33.01

TIME (MSEC)	PANEL 3 (FLOOR.) VS. SEGMENT 8 (RF)							PANEL 3 (FLOOR.) VS. SEGMENT 11 (LF)						
	DEFL- ECTION (IN.)	NORMAL FORCE (LB.)	FRICTION FORCE (LB.)	RESULTANT FORCE (LB.)	CONTACT (VEH REFERENCE)	LOCATION (IN.)		DEFL- ECTION (IN.)	NORMAL FORCE (LB.)	FRICTION FORCE (LB.)	RESULTANT FORCE (LB.)	CONTACT (VEH REFERENCE)	LOCATION (IN.)	
					X	Y	Z					X	Y	Z
0.000	0.096	4.82	0.00	4.82	46.596	3.420	-1.300	0.096	4.82	0.00	4.82	46.596	-3.420	-1.300
2.000	0.097	4.88	0.97	4.98	46.596	3.420	-1.300	0.097	4.88	0.97	4.98	46.596	-3.420	-1.300
4.000	0.097	4.97	2.11	5.40	46.597	3.420	-1.300	0.097	4.98	2.11	5.40	46.597	-3.420	-1.300
6.000	0.097	5.05	2.52	5.64	46.601	3.420	-1.300	0.097	5.05	2.52	5.64	46.601	-3.420	-1.300
8.000	0.098	5.02	2.51	5.62	46.609	3.420	-1.300	0.098	5.03	2.52	5.62	46.609	-3.420	-1.300
10.000	0.098	4.90	2.45	5.48	46.620	3.420	-1.300	0.098	4.91	2.46	5.49	46.620	-3.420	-1.300
12.000	0.098	4.88	2.44	5.46	46.638	3.420	-1.300	0.098	4.89	2.44	5.46	46.638	-3.420	-1.300
14.000	0.097	4.83	2.42	5.41	46.662	3.420	-1.300	0.097	4.84	2.42	5.41	46.662	-3.420	-1.300
16.000	0.095	4.75	2.37	5.31	46.693	3.420	-1.300	0.095	4.76	2.38	5.32	46.694	-3.420	-1.300
18.000	0.092	4.61	2.31	5.16	46.734	3.420	-1.300	0.093	4.63	2.31	5.17	46.734	-3.420	-1.300
20.000	0.088	4.42	2.21	4.94	46.784	3.420	-1.300	0.089	4.44	2.22	4.97	46.784	-3.420	-1.300
22.000	0.083	4.17	2.00	4.66	46.844	3.420	-1.300	0.084	4.19	2.10	4.69	46.845	-3.420	-1.300
24.000	0.077	3.85	1.92	4.30	46.915	3.420	-1.300	0.078	3.88	1.94	4.34	46.916	-3.420	-1.300
26.000	0.069	3.46	1.73	3.86	46.998	3.420	-1.300	0.070	3.50	1.75	3.91	46.999	-3.420	-1.300
28.000	0.060	2.99	1.50	3.35	47.092	3.420	-1.300	0.061	3.05	1.52	3.41	47.093	-3.420	-1.300
30.000	0.049	2.46	1.23	2.75	47.199	3.420	-1.300	0.051	2.54	1.27	2.83	47.201	-3.420	-1.300
32.000	0.037	1.85	0.93	2.07	47.318	3.420	-1.300	0.040	1.98	0.99	2.21	47.322	-3.420	-1.300
34.000	0.022	1.12	0.56	1.26	47.448	3.420	-1.300	0.027	1.36	0.68	1.52	47.456	-3.420	-1.300
36.000	0.004	0.19	0.10	0.21	47.583	3.420	-1.300	0.013	0.65	0.32	0.73	47.597	-3.420	-1.300
38.000	-0.021	0.00	0.00	0.00	0.000	0.000	0.000	-0.004	0.00	0.00	0.00	0.000	0.000	0.000
40.000	-0.052	0.00	0.00	0.00	0.000	0.000	0.000	-0.024	0.00	0.00	0.00	0.000	0.000	0.000
42.000	-0.092	0.00	0.00	0.00	0.000	0.000	0.000	-0.047	0.00	0.00	0.00	0.000	0.000	0.000
44.000	-0.140	0.00	0.00	0.00	0.000	0.000	0.000	-0.072	0.00	0.00	0.00	0.000	0.000	0.000
46.000	-0.199	0.00	0.00	0.00	0.000	0.000	0.000	-0.101	0.00	0.00	0.00	0.000	0.000	0.000
48.000	-0.268	0.00	0.00	0.00	0.000	0.000	0.000	-0.137	0.00	0.00	0.00	0.000	0.000	0.000
50.000	-0.348	0.00	0.00	0.00	0.000	0.000	0.000	-0.183	0.00	0.00	0.00	0.000	0.000	0.000
52.000	-0.40	0.00	0.00	0.00	0.000	0.000	0.000	-0.245	0.00	0.00	0.00	0.000	0.000	0.000
54.000	-0.543	0.00	0.00	0.00	0.000	0.000	0.000	-0.330	0.00	0.00	0.00	0.000	0.000	0.000
56.000	-0.661	0.00	0.00	0.00	0.000	0.000	0.000	-0.434	0.00	0.00	0.00	0.000	0.000	0.000
58.000	-0.798	0.00	0.00	0.00	0.000	0.000	0.000	-0.554	0.00	0.00	0.00	0.000	0.000	0.000
60.000	-0.960	0.00	0.00	0.00	0.000	0.000	0.000	-0.686	0.00	0.00	0.00	0.000	0.000	0.000
62.000	-1.145	0.00	0.00	0.00	0.000	0.000	0.000	-0.830	0.00	0.00	0.00	0.000	0.000	0.000
64.000	-1.351	0.00	0.00	0.00	0.000	0.000	0.000	-0.974	0.00	0.00	0.00	0.000	0.000	0.000
66.000	-1.576	0.00	0.00	0.00	0.000	0.000	0.000	-1.122	0.00	0.00	0.00	0.000	0.000	0.000
68.000	-1.817	0.00	0.00	0.00	0.000	0.000	0.000	-1.277	0.00	0.00	0.00	0.000	0.000	0.000
70.000	-2.071	0.00	0.00	0.00	0.000	0.000	0.000	-1.446	0.00	0.00	0.00	0.000	0.000	0.000
72.000	-2.335	0.00	0.00	0.00	0.000	0.000	0.000	-1.625	0.00	0.00	0.00	0.000	0.000	0.000
74.000	-2.606	0.00	0.00	0.00	0.000	0.000	0.000	-1.812	0.00	0.00	0.00	0.000	0.000	0.000
76.000	-2.883	0.00	0.00	0.00	0.000	0.000	0.000	-2.005	0.00	0.00	0.00	0.000	0.000	0.000
78.000	-3.160	0.00	0.00	0.00	0.000	0.000	0.000	-2.199	0.00	0.00	0.00	0.000	0.000	0.000
80.000	-3.434	0.00	0.00	0.00	0.000	0.000	0.000	-2.387	0.00	0.00	0.00	0.000	0.000	0.000

RUN DESCRIPTION: EXAMPLE 1: BASIC SLED TEST SIMULATION
 TWO BELT HARNESS WITH HYPERELLIPSOID FOR DASH BOARD

PAGE: 34.01

VEHICLE DECELERATION: SLED ACCELERATION - 20G PEAK
 CRASH VICTIM: 95TH PERCENTILE MALE
 CONTACT FORCES - SEGMENT PANELS VS. SEGMENTS

TIME (MSEC)	PANEL 4 (HEAD PAD. 13 DEGR) VS. SEGMENT 5 (H)						PANEL 10 (RUDDER PEDALS.) VS. SEGMENT 8 (RF)							
	DEFL- ECTION (IN.)	NORMAL FORCE (LB.)	FRICTION FORCE (LB.)	RESULTANT FORCE (LB.)	CONTACT LOCATION (IN.) (VEH REFERENCE)			DEFL- ECTION (IN.)	NORMAL FORCE (LB.)	FRICTION FORCE (LB.)	RESULTANT FORCE (LB.)	CONTACT LOCATION (IN.) (VEH REFERENCE)		
					X	Y	Z					X	Y	Z
0.000	0.050	2.48	0.00	2.48	3.935	0.000	-40.978	0.002	0.08	0.00	0.08	51.239	3.420	-3.286
2.000	0.049	2.47	0.02	2.47	3.935	0.000	-40.978	0.001	0.07	0.01	0.07	51.239	3.420	-3.286
4.000	0.048	2.38	0.19	2.39	3.935	0.000	-40.978	0.002	0.11	0.04	0.12	51.239	3.420	-3.285
6.000	0.043	2.15	0.41	2.19	3.935	0.000	-40.977	0.005	0.29	0.14	0.32	51.240	3.420	-3.287
8.000	0.034	1.70	0.55	1.78	3.935	0.000	-40.976	0.010	0.75	0.37	0.83	51.245	3.420	-3.291
10.000	0.019	0.95	0.41	1.03	3.936	0.000	-40.974	0.019	1.59	0.80	1.78	51.254	3.420	-3.298
12.000	-0.003	0.00	0.00	0.00	0.000	0.000	0.000	0.031	2.66	1.33	2.98	51.266	3.420	-3.309
14.000	-0.034	0.00	0.00	0.00	0.000	0.000	0.000	0.048	4.09	2.05	4.58	51.284	3.420	-3.323
16.000	-0.075	0.00	0.00	0.00	0.000	0.000	0.000	0.070	5.92	2.96	6.62	51.306	3.420	-3.342
18.000	-0.128	0.00	0.00	0.00	0.000	0.000	0.000	0.096	8.17	4.08	9.13	51.334	3.420	-3.365
20.000	-0.194	0.00	0.00	0.00	0.000	0.000	0.000	0.128	13.66	6.83	15.28	51.367	3.420	-3.393
22.000	-0.274	0.00	0.00	0.00	0.000	0.000	0.000	0.165	20.41	10.21	22.82	51.406	3.420	-3.426
24.000	-0.371	0.00	0.00	0.00	0.000	0.000	0.000	0.205	28.17	14.09	31.50	51.464	3.420	-3.474
26.000	-0.484	0.00	0.00	0.00	0.000	0.000	0.000	0.249	38.47	19.24	43.01	51.490	3.420	-3.496
28.000	-0.617	0.00	0.00	0.00	0.000	0.000	0.000	0.296	49.29	24.64	55.10	51.503	3.420	-3.507
30.000	-0.770	0.00	0.00	0.00	0.000	0.000	0.000	0.343	60.44	30.22	67.58	51.511	3.420	-3.514
32.000	-0.945	0.00	0.00	0.00	0.000	0.000	0.000	0.392	71.83	35.92	80.31	51.516	3.420	-3.518
34.000	-1.142	0.00	0.00	0.00	0.000	0.000	0.000	0.440	128.19	64.09	143.32	51.520	3.420	-3.521
36.000	-1.363	0.00	0.00	0.00	0.000	0.000	0.000	0.477	178.75	89.37	199.85	51.521	3.420	-3.522
38.000	-1.610	0.00	0.00	0.00	0.000	0.000	0.000	0.496	204.92	102.46	229.11	51.519	3.420	-3.521
40.000	-1.882	0.00	0.00	0.00	0.000	0.000	0.000	0.498	191.03	95.51	213.58	51.515	3.420	-3.517
42.000	-2.181	0.00	0.00	0.00	0.000	0.000	0.000	0.492	182.60	91.30	204.15	51.508	3.419	-3.511
44.000	-2.505	0.00	0.00	0.00	0.000	0.000	0.000	0.482	168.82	84.41	188.75	51.501	3.418	-3.505
46.000	-2.849	0.00	0.00	0.00	0.000	0.000	0.000	0.470	153.91	76.96	172.08	51.494	3.417	-3.499
48.000	-3.211	0.00	0.00	0.00	0.000	0.000	0.000	0.460	139.82	69.91	156.32	51.489	3.415	-3.495
50.000	-3.584	0.00	0.00	0.00	0.000	0.000	0.000	0.450	126.65	63.33	141.60	51.488	3.413	-3.495
52.000	-3.960	0.00	0.00	0.00	0.000	0.000	0.000	0.441	114.28	57.14	127.77	51.494	3.410	-3.499
54.000	-4.328	0.00	0.00	0.00	0.000	0.000	0.000	0.430	100.59	50.30	112.47	51.507	3.406	-3.510
56.000	-4.675	0.00	0.00	0.00	0.000	0.000	0.000	0.415	79.36	39.68	88.73	51.531	3.402	-3.531
58.000	-4.989	0.00	0.00	0.00	0.000	0.000	0.000	0.388	57.60	28.80	64.40	51.577	3.398	-3.569
60.000	-5.254	0.00	0.00	0.00	0.000	0.000	0.000	0.346	49.17	24.59	54.98	51.679	3.394	-3.654
62.000	-5.458	0.00	0.00	0.00	0.000	0.000	0.000	0.289	37.73	18.86	42.18	51.812	3.392	-3.767
64.000	-5.609	0.00	0.00	0.00	0.000	0.000	0.000	0.215	23.04	11.52	25.75	51.976	3.393	-3.904
66.000	-5.724	0.00	0.00	0.00	0.000	0.000	0.000	0.125	8.82	4.41	9.86	52.183	3.400	-4.077
68.000	-5.817	0.00	0.00	0.00	0.000	0.000	0.000	0.020	0.99	0.49	1.10	52.440	3.417	-4.294
70.000	-5.894	0.00	0.00	0.00	0.000	0.000	0.000	-0.101	0.00	0.00	0.00	0.000	0.000	0.000
72.000	-5.963	0.00	0.00	0.00	0.000	0.000	0.000	-0.231	0.00	0.00	0.00	0.000	0.000	0.000
74.000	-6.032	0.00	0.00	0.00	0.000	0.000	0.000	-0.367	0.00	0.00	0.00	0.000	0.000	0.000
76.000	-6.105	0.00	0.00	0.00	0.000	0.000	0.000	-0.510	0.00	0.00	0.00	0.000	0.000	0.000
78.000	-6.178	0.00	0.00	0.00	0.000	0.000	0.000	-0.655	0.00	0.00	0.00	0.000	0.000	0.000
80.000	-6.252	0.00	0.00	0.00	0.000	0.000	0.000	-0.800	0.00	0.00	0.00	0.000	0.000	0.000

DATE: 2 SEPT 1988

RUN DESCRIPTION: EXAMPLE 1: BASIC SLED TEST SIMULATION
TWO BELT HARNESS WITH HYPERELLIPSOID FOR DASH BOARD

PAGE: 35.01

VEHICLE DECELERATION: SLED ACCELERATION - 20G PEAK

CRASH VICTIM: 95TH PERCENTILE MALE

CONTACT FORCES - SEGMENT PANELS VS. SEGMENTS

PANEL 10 (RUDDER PEDALS.) VS. SEGMENT 11 (LF)

TIME (MSEC)	DEFL- ECTION (IN.)	NORMAL FORCE (LB.)	FRICTION FORCE (LB.)	RESULTANT FORCE (LB.)	CONTACT LOCATION (IN.) (VEH REFERENCE)		
					X	Y	Z
0.000	0.002	0.08	0.00	0.08	51.239	-3.420	-3.286
2.000	0.001	0.07	0.01	0.07	51.239	-3.420	-3.286
4.000	0.002	0.11	0.04	0.12	51.239	-3.420	-3.285
6.000	0.005	0.29	0.14	0.32	51.240	-3.420	-3.287
8.000	0.010	0.75	0.37	0.83	51.245	-3.420	-3.291
10.000	0.019	1.60	0.80	1.78	51.254	-3.420	-3.298
12.000	0.031	2.67	1.33	2.98	51.266	-3.420	-3.308
14.000	0.048	4.10	2.05	4.58	51.284	-3.420	-3.323
16.000	0.070	5.93	2.96	6.63	51.306	-3.420	-3.342
18.000	0.096	8.18	4.09	9.15	51.334	-3.420	-3.365
20.000	0.128	13.70	6.85	15.32	51.366	-3.420	-3.393
22.000	0.165	20.47	10.23	22.88	51.406	-3.420	-3.425
24.000	0.206	28.26	14.13	31.59	51.464	-3.420	-3.474
26.000	0.250	38.58	19.29	43.14	51.489	-3.420	-3.496
28.000	0.296	49.45	24.73	55.29	51.503	-3.420	-3.507
30.000	0.344	60.69	30.34	67.85	51.511	-3.420	-3.513
32.000	0.394	72.29	36.14	80.82	51.516	-3.420	-3.518
34.000	0.444	133.60	66.80	149.37	51.519	-3.420	-3.520
36.000	0.483	188.09	94.05	210.29	51.519	-3.420	-3.520
38.000	0.507	219.74	109.87	245.68	51.515	-3.420	-3.518
40.000	0.514	216.05	108.03	241.55	51.508	-3.420	-3.511
42.000	0.515	213.02	106.51	238.16	51.497	-3.421	-3.502
44.000	0.514	212.01	106.00	237.03	51.482	-3.422	-3.490
46.000	0.514	211.89	105.95	236.90	51.465	-3.423	-3.475
48.000	0.514	212.07	106.04	237.10	51.446	-3.425	-3.459
50.000	0.511	207.70	103.85	232.22	51.425	-3.427	-3.441
52.000	0.498	190.37	95.18	212.83	51.399	-3.430	-3.420
54.000	0.472	155.61	77.80	173.97	51.361	-3.432	-3.388
56.000	0.436	107.92	53.96	120.66	51.306	-3.435	-3.342
58.000	0.395	59.01	29.50	65.97	51.268	-3.437	-3.310
60.000	0.353	50.53	25.27	56.50	51.259	-3.438	-3.302
62.000	0.307	41.43	20.72	46.32	51.288	-3.438	-3.327
64.000	0.268	33.64	16.82	37.61	51.387	-3.436	-3.409
66.000	0.234	26.90	13.45	30.07	51.514	-3.432	-3.516
68.000	0.200	20.05	10.02	22.41	51.657	-3.425	-3.637
70.000	0.159	13.90	6.95	15.54	51.813	-3.413	-3.767
72.000	0.112	6.75	3.37	7.55	51.972	-3.393	-3.901
74.000	0.056	2.80	1.40	3.13	52.128	-3.366	-4.032
76.000	-0.008	0.00	0.00	0.00	0.000	0.000	0.000
78.000	-0.077	0.00	0.00	0.00	0.000	0.000	0.000
80.000	-0.141	0.00	0.00	0.00	0.000	0.000	0.000

DATE: 2 SEPT 1988

RUN DESCRIPTION: EXAMPLE 1: BASIC SLED TEST SIMULATION
TWO BELT HARNESS WITH HYPERELLIPSOID FOR DASH BOARD

PAGE: 36.01

VEHICLE DECELERATION: SLED ACCELERATION - 20G PEAK
CRASH VICTIM: 95TH PERCENTILE MALE

HARNESS SYSTEM BELT ENDPOINT FORCES

TIME (MSEC)	BELT NO. 1 OF HARNESS NO. 1				BELT NO. 2 OF HARNESS NO. 1			
	POINT NO. 1		POINT NO. 12		POINT NO. 13		POINT NO. 27	
	STRAIN (IN./ IN.)	FORCE (LB.)	STRAIN (IN./ IN.)	FORCE (LB.)	STRAIN (IN./ IN.)	FORCE (LB.)	STRAIN (IN./ IN.)	FORCE (LB.)
0.000	0.000000	0.00	0.000000	0.00	0.000000	0.00	0.000000	0.00
2.000	0.000025	0.37	0.000017	0.25	0.000014	0.21	0.000018	0.27
4.000	0.000119	1.79	0.000065	0.98	0.000057	0.85	0.000152	2.27
6.000	0.000281	4.22	0.000150	2.25	0.000124	1.86	0.000470	7.04
8.000	0.000523	7.85	0.000276	4.15	0.000219	3.29	0.001018	15.27
10.000	0.000854	12.81	0.000438	6.57	0.000336	5.04	0.001803	27.04
12.000	0.001023	15.34	0.000481	7.21	0.000344	5.16	0.002454	36.81
14.000	0.001320	19.80	0.000515	7.72	0.000433	6.50	0.003168	47.52
16.000	0.001672	25.08	0.000551	8.27	0.000496	7.44	0.003856	57.84
18.000	0.001420	21.30	0.000400	6.00	0.000404	6.06	0.003427	51.41
20.000	0.001831	27.40	0.000484	7.26	0.000498	7.46	0.004276	64.13
22.000	0.002363	35.45	0.000617	9.26	0.000643	9.65	0.005479	82.19
24.000	0.003085	46.27	0.000877	13.15	0.000921	13.81	0.019008	285.12
26.000	0.004040	60.61	0.001215	18.22	0.001284	19.26	0.021036	315.53
28.000	0.006003	90.05	0.001724	25.85	0.001768	26.52	0.027340	410.10
30.000	0.009472	142.08	0.002256	33.83	0.002429	36.44	0.035707	564.14
32.000	0.013979	209.69	0.003249	48.73	0.004372	65.58	0.023671	355.06
34.000	0.019790	296.85	0.004754	71.30	0.008401	126.02	0.020865	312.97
36.000	0.026146	392.20	0.007124	106.86	0.018334	275.01	0.025025	375.38
38.000	0.032200	494.00	0.010831	162.46	0.030199	453.97	0.039302	636.04
40.000	0.037809	606.17	0.021023	315.35	0.040241	654.82	0.056200	1178.59
42.000	0.043063	711.27	0.036025	570.49	0.050245	863.00	0.075150	2182.96
44.000	0.048343	816.85	0.053858	1054.46	0.060539	1408.57	0.064897	1639.53
46.000	0.053827	1052.84	0.071518	1990.44	0.072248	2029.17	0.084525	2679.84
48.000	0.055271	1129.35	0.087363	2830.23	0.056456	1192.15	0.103767	3631.84
50.000	0.058800	1316.42	0.098307	3410.27	0.065766	1685.60	0.093883	3175.81
52.000	0.064710	1629.61	0.098688	3430.48	0.068526	1831.86	0.107224	3752.85
54.000	0.072120	2022.35	0.087992	2863.56	0.074577	2152.58	0.092638	3109.84
56.000	0.076068	2231.59	0.071069	1966.67	0.075155	2183.20	0.089018	2917.94
58.000	0.074497	2148.32	0.049516	840.31	0.069789	1898.82	0.084438	2675.23
60.000	0.061993	1485.61	0.026359	395.38	0.082583	2576.88	0.093363	3148.22
62.000	0.047024	790.48	0.005156	77.34	0.032040	490.80	0.086107	2763.69
64.000	0.028177	422.66	0.000000	0.00	0.024499	367.48	0.065353	1663.72
66.000	0.007038	105.57	0.000000	0.00	0.006231	93.46	0.051787	944.69
68.000	0.000000	0.00	0.000000	0.00	0.000000	0.00	0.037962	609.25
70.000	0.000000	0.00	0.000000	0.00	0.000000	0.00	0.022501	337.51
72.000	0.000000	0.00	0.000000	0.00	0.000000	0.00	0.012711	190.67
74.000	0.000000	0.00	0.000000	0.00	0.000000	0.00	0.006159	92.39
76.000	0.000000	0.00	0.000000	0.00	0.000000	0.00	0.000000	0.00
78.000	0.000000	0.00	0.000000	0.00	0.000000	0.00	0.000000	0.00
80.000	0.000000	0.00	0.000000	0.00	0.000000	0.00	0.000000	0.00

DATE: 2 SEPT 1988

RUN DESCRIPTION: EXAMPLE 1: BASIC SLED TEST SIMULATION
TWO BELT HARNESS WITH HYPERELLIPSOID FOR DASH BOARD

PAGE: 40.01

VEHICLE DECELERATION: SLED ACCELERATION - 20G PEAK

CRASH VICTIM: 95TH PERCENTILE MALE

CONTACT FORCES - SEGMENT NO. 9 (L'L) VS. SEGMENT NO. 15 (LLA)

TIME (MSEC)	DEFL- ECTION (IN.)	NORMAL FORCE (LB.)	FRICTION FORCE (LB.)	RESULTANT FORCE (LB.)	CONTACT LOCATION (IN.)					
					SEG. 9 X	SEG. 9 Y	SEG. 9 Z	SEG. 15 X	SEG. 15 Y	SEG. 15 Z
0.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
2.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
4.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
6.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.050	0.000
8.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
10.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
12.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
14.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
16.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
18.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
20.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
22.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
24.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
26.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
28.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
30.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
32.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
34.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
36.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
38.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
40.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
42.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
44.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
46.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
48.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
50.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
52.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
54.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
56.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
58.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
60.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
62.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
64.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
66.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
68.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
70.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
72.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
74.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
76.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
78.000	0.282	132.75	33.19	136.84	1.229	-1.146	10.441	-1.121	0.658	6.935
80.000	0.685	321.84	80.46	331.75	1.096	-1.142	10.037	-1.019	0.683	6.725

DATE: 2 SEPT 1988
 RUN DESCRIPTION: EXAMPLE 1: BASIC SLED TEST SIMULATION
 TWO BELT HARNESS WITH HYPERELLIPSOID FOR DASH BOARD
 VEHICLE DECELERATION: SLED ACCELERATION - 20G PEAK
 CRASH VICTIM: 95TH PERCENTILE MALE
 CONTACT FORCES - SEGMENT NO. 13 (RLA) VS. SEGMENT NO. 16 (VEH)

PAGE: 41.01

TIME (MSEC)	DEFL- ECTION (IN.)	NORMAL FORCE (LB.)	FRICTION FORCE (LB.)	RESULTANT FORCE (LB.)	CONTACT LOCATION (IN.)					
					SEG. 13 LOCAL REFERENCE			SEG. 16 LOCAL REFERENCE		
					X	Y	Z	X	Y	Z
0.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
2.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
4.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
6.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
8.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
10.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
12.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
14.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
16.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
18.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
20.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
22.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
24.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
26.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
28.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
30.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
32.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
34.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
36.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
38.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
40.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
42.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
44.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
46.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
48.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
50.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
52.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
54.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
56.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
58.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
60.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
62.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
64.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
66.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
68.000	0.217	24.79	12.39	27.71	1.714	0.433	2.758	32.371	6.599	-23.212
70.000	0.496	193.03	96.51	215.81	1.651	0.426	3.110	32.482	6.706	-23.329
72.000	0.671	444.83	222.41	497.33	1.602	0.408	3.504	32.546	6.820	-23.401
74.000	0.627	362.71	181.36	405.53	1.587	0.388	3.859	32.512	6.921	-23.391
76.000	0.420	87.00	43.50	97.27	1.600	0.370	4.164	32.409	7.003	-23.320
78.000	0.219	23.90	11.95	26.72	1.612	0.354	4.449	32.311	7.076	-23.249
80.000	0.103	5.46	2.73	6.10	1.608	0.342	4.722	32.256	7.143	-23.206

DATE: 2 SEPT 1988

RUN DESCRIPTION: EXAMPLE 1: BASIC SLED TEST SIMULATION
TWO BELT HARNESS WITH HYPERELLIPSOID FOR DASH BOARD

PAGE: 42.01

VEHICLE DECELERATION: SLED ACCELERATION - 20G PEAK
CRASH VICTIM: 95TH PERCENTILE MALE

CONTACT FORCES - SEGMENT NO. 15 (LLA) VS. SEGMENT NO. 16 (VEH)

TIME (MSEC)	DEFL- ECTION (IN.)	NORMAL FORCE (LB.)	FRICTION FORCE (LB.)	RESULTANT FORCE (LB.)	CONTACT LOCATION (IN.)					
					SEG. 15 LOCAL REFERENCE			SEG. 16 LOCAL REFERENCE		
					X	Y	Z	X	Y	Z
0.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
2.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
4.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
6.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
8.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
10.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
12.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
14.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
16.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
18.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
20.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
22.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
24.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
26.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
28.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
30.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
32.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
34.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
36.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
38.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
40.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
42.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
44.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
46.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
48.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
50.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
52.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
54.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
56.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
58.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
60.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
62.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
64.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
66.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
68.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
70.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
72.000	0.524	226.73	113.37	253.49	1.755	0.307	0.651	32.615	-6.119	-23.311
74.000	0.804	610.21	305.10	682.23	1.712	0.298	0.537	32.732	-6.130	-23.436
76.000	0.907	736.13	368.06	823.02	1.699	0.281	0.427	32.764	-6.147	-23.486
78.000	0.798	590.20	295.10	659.86	1.720	0.263	0.294	32.694	-6.176	-23.450
80.000	0.583	303.74	151.87	339.60	1.758	0.244	0.157	32.574	-6.224	-23.368

HIC, HSI AND CSI RESULTS
HEAD INJURY CRITERION

HIC = 192.44 TIME DURATION = 52.000 TO 65.000 MSEC
 WITH HEAD RESULTANTS = 25.844 AND 23.464 G'S

 AVERAGE HEAD RESULTANT FOR TIME DURATION = 46.574 G'S
HEAD SEVERITY INDEX

HSI = 245.20

MAX HEAD RESULTANT = 65.716 G'S AT 60.000 MSEC
CHEST SEVERITY INDEX

CSI = 733.64

MAX CHEST RESULTANT = 146.473 G'S AT 59.000 MSEC

SUB	CALLS	TIME	%
MAIN3D	1	213	1.69
INPUT	1	220	1.74
CHAIN	484	306	2.42
EJOINT	484	17	0.13
DINT	41	496	3.92
PDAUX	568	434	3.43
DAUX	483	520	4.11
SETUP1	483	236	1.87
CONTCT	483	235	1.86
PLELP	5313	901	7.13
SEGSEG	2898	886	7.01
HBELT	1072	1434	11.35
VISPR	483	758	6.00
SETUP2	483	89	0.70
DAUX11	483	629	4.98
DAUX12	483	181	1.43
DAUX22	483	110	0.87
FSMSOL	1072	1407	11.13
OUTPUT	86	1148	9.08
UPDATE	85	0	0.00
HPTURB	85	1628	12.88
DZP	482	148	1.17
POSTPR	1	643	5.09
TOTAL		12639	100.00

APPENDIX C4

Example 1, ATB Basic Sled Test Simulation
Perkin-Elmer Output File

SEGMENT I SYM PLOT	WEIGHT (LB.)	PRINCIPAL MOMENTS OF INERTIA (LB.-SEC.**2- IN.)			SEGMENT CONTACT ELLIPSOID						PRINCIPAL AXES (DEG)		
		X	Y	Z	SEMIAXES (IN.)			CENTER (IN.)			YAW	PITCH	ROLL
					X	Y	Z	X	Y	Z			
1 LT 1	34.772	1.6280	1.0101	1.7954	5.168	7.436	3.778	0.000	0.000	-0.072	0.00	0.00	0.00
2 CT 2	13.099	0.4160	0.2301	0.5857	4.809	6.673	4.145	0.000	0.000	-0.011	0.00	0.00	0.00
3 UT 3	53.673	3.4525	2.7832	2.3509	5.220	6.906	7.314	0.000	0.000	-0.872	0.00	0.00	0.00
4 N 4	3.289	0.0278	0.0278	0.0216	2.520	2.520	3.156	0.000	0.000	0.000	0.00	0.00	0.00
5 H 5	11.927	0.2708	0.3085	0.1584	3.984	3.125	5.836	0.000	0.000	0.000	0.00	0.00	0.00
6 RUL 6	22.725	2.0130	2.0130	0.2576	3.308	3.308	12.652	0.000	0.000	0.000	0.00	0.00	0.00
7 RLL 7	9.763	0.4989	0.4989	0.0626	2.487	2.487	9.616	0.000	0.000	0.000	0.00	0.00	0.00
8 RF 8	2.079	0.0426	0.0412	0.0055	2.866	2.016	5.617	0.000	0.000	1.462	0.00	0.00	0.00
9 LUL 9	22.725	2.0130	2.0130	0.2576	3.308	3.308	12.652	0.000	0.000	0.000	0.00	0.00	0.00
10 LLL A	9.763	0.4989	0.4989	0.0626	2.487	2.487	9.616	0.000	0.000	0.000	0.00	0.00	0.00
11 LF B	2.079	0.0426	0.0412	0.0055	2.866	2.016	5.617	0.000	0.000	1.462	0.00	0.00	0.00
12 RUA C	5.542	0.1743	0.1743	0.0259	2.122	2.122	7.497	0.000	0.000	0.000	0.00	0.00	0.00
13 RLA D	5.901	0.3331	0.3331	0.0214	1.871	1.871	10.269	0.000	0.000	0.000	0.00	0.00	0.00
14 LUA E	5.542	0.1743	0.1743	0.0259	2.122	2.122	7.497	0.000	0.000	0.000	0.00	0.00	0.00
15 LLA F	5.901	0.3331	0.3331	0.0214	1.871	1.871	10.269	0.000	0.000	0.000	0.00	0.00	0.00

JOINT J SYM PLOT JNT PIN	LOCATION(IN.) - SEG(JNT)			LOCATION(IN.) - SEG(J+1)			PRIN. AXIS(DEG) - SEG(JNT)			PRIN. AXIS(DEG) - SEG(J+1)		
	X	Y	Z	X	Y	Z	YAW	PITCH	ROLL	YAW	PITCH	ROLL
1 P M 1 0	0.000	0.000	-3.850	0.000	0.000	1.610	0.00	0.00	0.00	0.00	5.00	0.00
2 W N 2 0	0.000	0.000	-1.640	0.000	0.000	6.440	0.00	0.00	0.00	0.00	5.00	0.00
3 NP O 3 0	0.000	0.000	-8.190	0.000	0.000	0.640	0.00	0.00	0.00	0.00	10.00	0.00
4 HP P 4 0	0.000	0.000	-0.640	0.000	0.000	5.840	0.00	0.00	0.00	0.00	10.00	0.00
5 RH Q 1 0	0.000	3.420	-0.310	0.000	0.000	-8.640	0.00	0.00	0.00	0.00	-45.00	0.00
6 RK R 6 1	0.000	0.000	10.000	0.000	0.000	-6.970	0.00	0.00	0.00	0.00	60.00	0.00
7 RA S 7 0	0.000	0.000	8.120	2.870	0.000	-2.660	0.00	90.00	0.00	0.00	10.00	0.00
8 LH T 1 0	0.000	-3.420	-0.310	0.000	0.000	-8.640	0.00	0.00	0.00	0.00	-45.00	0.00
9 LK U 9 1	0.000	0.900	10.000	0.000	0.000	-6.970	0.00	0.00	0.00	0.00	60.00	0.00
10 LA V 10 0	0.000	0.000	8.120	2.870	0.000	-2.660	0.00	90.00	0.00	0.00	10.00	0.00
11 RS W 3 0	0.000	6.240	-5.220	0.000	0.000	-5.370	0.00	0.00	0.00	0.00	-4.10	0.00
12 RE X 12 1	0.000	0.000	5.420	0.000	0.000	-8.200	0.00	0.00	0.00	0.00	-70.00	0.00
13 LS Y 3 0	0.000	-6.240	-5.220	0.000	0.000	-5.370	0.00	0.00	0.00	0.00	-4.10	0.00
14 LE Z 14 1	0.000	0.000	5.420	0.000	0.000	-8.200	0.00	0.00	0.00	0.00	-70.00	0.00

CARDS 8.3

FLEXURAL SPRING CHARACTERISTICS

TORSIONAL SPRING CHARACTERISTICS

JOINT	SPRING COEF. (IN. LB./DEG**J)			ENERGY DISSIPATION COEF.	JOINT STOP (DEG)	SPRING COEF. (IN. LB./DEG**J)			ENERGY DISSIPATION COEF.	JOINT STOP (DEG)
	LINEAR (J=1)	QUADRATIC (J=2)	CUBIC (J=3)			LINEAR (J=1)	QUADRATIC (J=2)	CUBIC (J=3)		
1 P	0.000	10.000	0.000	0.700	20.000	0.000	10.000	0.000	0.700	5.000
2 W	0.000	10.000	0.000	0.700	20.000	0.000	10.000	0.000	0.700	35.000
3 NP	0.000	5.000	0.000	0.700	25.000	0.000	10.000	0.000	0.700	35.000
4 HP	0.000	5.000	0.000	0.700	25.000	0.000	10.000	0.000	0.700	35.000
5 RH	0.000	10.000	0.000	0.700	70.000	0.000	0.800	0.000	0.700	40.000
6 RK	0.000	1.800	0.000	0.700	60.000	0.000	0.000	0.000	0.000	0.000
7 RA	0.000	7.000	0.000	0.700	35.000	0.000	10.000	0.000	0.700	26.000
8 LH	0.000	10.000	0.000	0.700	70.000	0.000	0.800	0.000	0.700	40.000
9 LK	0.000	1.800	0.000	0.700	60.000	0.000	0.000	0.000	0.000	0.000
10 LA	0.000	7.000	0.000	0.700	35.000	0.000	10.000	0.000	0.700	26.000
11 RS	0.000	10.000	0.000	0.700	122.500	0.000	10.000	0.000	0.700	65.000
12 RE	0.000	1.800	0.000	0.700	70.000	0.000	0.000	0.000	0.000	0.000
13 LS	0.000	10.000	0.000	0.700	122.500	0.000	10.000	0.000	0.700	65.000
14 LE	0.000	1.800	0.000	0.700	70.000	0.000	0.000	0.000	0.000	0.000

CARDS B.5

JOINT VISCOUS CHARACTERISTICS AND LOCK-UNLOCK CONDITIONS

JOINT	VISCOUS COEFFICIENT (IN. LB.SEC./DEG)	COULOMB FRICTION COEF. (IN. LB.)	FULL FRICTION ANGULAR VELOCITY (DEG/SEC.)	MAX TORQUE FOR A LOCKED JOINT (IN. LB.)	MIN TORQUE FOR UNLOCKED JOINT (IN. LB.)	MIN. ANG. VELOCITY FOR UNLOCKED JOINT (RAD/SEC.)	IMPULSE RESTITUTION COEFFICIENT
1 P	0.100	0.00	30.00	0.00	0.00	0.00	0.000
2 W	0.100	0.00	30.00	0.00	0.00	0.00	0.000
3 NP	0.100	0.00	30.00	0.00	0.00	0.00	0.000
4 HP	0.100	0.00	30.00	0.00	0.00	0.00	0.000
5 RH	0.100	0.00	30.00	0.00	0.00	0.00	0.000
6 RK	0.100	0.00	30.00	0.00	0.00	0.00	0.000
7 RA	0.100	0.00	30.00	0.00	0.00	0.00	0.000
8 LH	0.100	0.00	30.00	0.00	0.00	0.00	0.000
9 LK	0.100	0.00	30.00	0.00	0.00	0.00	0.000
10 LA	0.100	0.00	30.00	0.00	0.00	0.00	0.000
11 RS	0.100	0.00	30.00	0.00	0.00	0.00	0.000
12 RE	0.100	0.00	30.00	0.00	0.00	0.00	0.000
13 LS	0.100	0.00	30.00	0.00	0.00	0.00	0.000
14 LE	0.100	0.00	30.00	0.00	0.00	0.00	0.000

SEGMENT NO. SYM	ANGULAR VELOCITIES (RAD/SEC.)			LINEAR VELOCITIES (IN./SEC.)			ANGULAR ACCELERATIONS (RAD/SEC.**2)			LINEAR ACCELERATIONS (IN./SEC.**2)		
	MAG. TEST	ABS. ERROR	REL. ERROR	MAG. TEST	ABS. ERROR	REL. ERROR	MAG. TEST	ABS. ERROR	REL. ERROR	MAG. TEST	ABS. ERROR	REL. ERROR
1 LT	0.010	0.010	0.0100	0.010	0.010	0.0100	0.100	0.100	0.1000	0.100	0.100	0.0100
2 CT	0.010	0.010	0.0100	0.000	0.000	0.0000	0.100	0.100	0.1000	0.000	0.000	0.0000
3 UT	0.010	0.010	0.0100	0.000	0.000	0.0000	0.100	0.100	0.1000	0.000	0.000	0.0000
4 N	0.010	0.010	0.0100	0.000	0.000	0.0000	0.100	0.100	0.1000	0.000	0.000	0.0000
5 H	0.010	0.010	0.0100	0.000	0.000	0.0000	0.100	0.100	0.1000	0.000	0.000	0.0000
6 RUL	0.010	0.010	0.0100	0.000	0.000	0.0000	0.100	0.100	0.1000	0.000	0.000	0.0000
7 RLL	0.010	0.010	0.0100	0.000	0.000	0.0000	0.100	0.100	0.1000	0.000	0.000	0.0000
8 RF	0.010	0.010	0.0100	0.000	0.000	0.0000	0.100	0.100	0.1000	0.000	0.000	0.0000
9 LUL	0.010	0.010	0.0100	0.000	0.000	0.0000	0.100	0.100	0.1000	0.000	0.000	0.0000
10 LLL	0.010	0.010	0.0100	0.000	0.000	0.0000	0.100	0.100	0.1000	0.000	0.000	0.0000
11 LF	0.010	0.010	0.0100	0.000	0.000	0.0000	0.100	0.100	0.1000	0.000	0.000	0.0000
12 RUA	0.010	0.010	0.0100	0.000	0.000	0.0000	0.100	0.100	0.1000	0.000	0.000	0.0000
13 RLA	0.010	0.010	0.0100	0.000	0.000	0.0000	0.100	0.100	0.1000	0.000	0.000	0.0000
14 LUA	0.010	0.010	0.0100	0.000	0.000	0.0000	0.100	0.100	0.1000	0.000	0.000	0.0000
15 LLA	0.010	0.010	0.0100	0.000	0.000	0.0000	0.100	0.100	0.1000	0.000	0.000	0.0000

SLED ACCELERATION - 20G PEAK

YAW	PITCH	ROLL	VIPS	VTIME	XO(X)	XO(Y)	XO(Z)	NATAB	ATO	ADT	MSEG
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	15	0.000000	0.010000	0

UNIDIRECTIONAL VEHICLE POSITION TABLES

TIME (MSEC)	ACC (G)	VELOCITY (IN./SEC.)	POSITION (IN.)	TIME (MSEC)	ACC (G)	VELOCITY (IN./SEC.)	POSITION (IN.)
0.00000	0.00	0.0000	0.00000				
10.00000	5.00	-9.6522	-0.03217				
20.00000	10.00	-38.6088	-0.25739				
30.00000	15.00	-86.8698	-0.86870				
40.00000	20.00	-154.4352	-2.05914				
50.00000	15.00	-222.0006	-3.95740				
60.00000	10.00	-270.2616	-6.43480				
70.00000	5.00	-299.2182	-9.29829				
80.00000	0.00	-308.8704	-12.35482				
90.00000	0.00	-308.8704	-15.44352				
100.00000	0.00	-308.8704	-18.53222				
110.00000	0.00	-308.8704	-21.62093				
120.00000	0.00	-308.8704	-24.70963				
130.00000	0.00	-308.8704	-27.79834				
140.00000	0.00	-308.8704	-30.88704				

PLANE INPUTS

PLANE NO. 1 SEAT. 6 DEGREE OFF H

	X	Y	Z
POINT 1	10.0000	8.0000	-10.0000
POINT 2	28.0100	8.0000	-11.8900
POINT 3	10.0000	-8.0000	-10.0000

PLANE NO. 2 BACK PANEL. 13 DEGR

	X	Y	Z
POINT 1	1.0000	9.0000	-48.9700
POINT 2	10.0000	9.0000	-10.0000
POINT 3	1.0000	-9.0000	-48.9700

PLANE NO. 3 FLOOR.

	X	Y	Z
POINT 1	0.0000	12.0000	-1.3000
POINT 2	60.0000	12.0000	-1.3000
POINT 3	0.0000	-12.0000	-1.3000

PLANE NO. 4 HEAD PAD. 13 DEGR

	X	Y	Z
POINT 1	2.4800	7.5000	-47.2600
POINT 2	4.9600	7.5000	-36.5500
POINT 3	2.4800	-7.5000	-47.2600

PLANE NO. 5 SEAT FRONT PANEL.

	X	Y	Z
POINT 1	28.0100	8.0000	-11.8900
POINT 2	26.6600	8.0000	-4.4000
POINT 3	28.0100	-8.0000	-11.8900

PLANE NO. 6 BACK PANEL2. 13 DEGR

	X	Y	Z
POINT 1	1.0000	9.0000	-48.9700
POINT 2	10.0000	9.0000	-10.0000
POINT 3	1.0000	-9.0000	-48.9700

PLANE NO. 7 FIREWALL.

	X	Y	Z
POINT 1	60.0000	12.0000	-25.0000
POINT 2	60.0000	-12.0000	-25.0000
POINT 3	60.0000	12.0000	-0.7500

PLANE NO. 8 RIGHT SIDE SEAT/IN.

	X	Y	Z
POINT 1	8.4100	8.1000	-6.6600
POINT 2	8.7000	8.1000	-14.7300
POINT 3	30.5800	8.1000	-6.6400

PLANE NO. 9 LEFT SIDE SEAT/IN.

	X	Y	Z
POINT 1	8.4100	-8.1000	-6.6600
POINT 2	30.5800	-8.1000	-6.6400
POINT 3	8.7000	-8.1000	-14.7300

PLANE NO. 10 RUDDER PEDALS.

	X	Y	Z
POINT 1	49.9920	9.0000	-2.2392
POINT 2	52.9920	9.0000	-4.7565
POINT 3	49.9920	-9.0000	-2.2392

PLANE NO. 11 LEFT SIDE PANEL.

	X	Y	Z
POINT 1	1.0000	-9.0000	-48.9700
POINT 2	10.9000	-9.0000	-6.1000
POINT 3	-7.7700	-9.0000	-46.9500

PLANE NO. 12 RIGHT SIDE PANEL.

	X	Y	Z
POINT 1	1.0000	9.0000	-48.9700
POINT 2	-7.7700	9.0000	-46.9500
POINT 3	10.9000	9.0000	-6.1000

FUNCTION NO. 3 SEGMENT-SEGMENT FCN. NTI(3) = 1

D0	D1	D2	D3	D4
0.0000	-5.0000	0.0000	0.0000	1.0000

FIRST PART OF FUNCTION - 6 TABULAR POINTS

D	F(D)
0.000000	0.0000
1.000000	470.0000
2.000000	890.0000
3.000000	1220.0000
4.000000	1470.0000
5.000000	1580.0000

FUNCTION NO. 6 CONSTANT, F=0.0 NTI(6) = 19

CARDS E

D0	D1	D2	D3	D4
0.0000	0.0000	0.0000	0.0000	0.0000

FUNCTION IS CONSTANT 0.000000

FUNCTION NO. 7 R FACTOR.

NTI(7) = 24

D0	D1	D2	D3	D4
0.0000	0.0000	0.7000	0.0000	0.0000

FUNCTION IS CONSTANT 0.700000

FUNCTION NO. 13 STIFF SURFACES

NTI(13) = 29

CARDS E

D0	D1	D2	D3	D4
0.0000	-4.0000	0.0000	0.0000	1.0000

FIRST PART OF FUNCTION - 8 TABULAR POINTS

D	F(D)
0.000000	0.0000
0.100000	5.0000
0.200000	20.0000
0.300000	40.0000
0.400000	60.0000
1.000000	860.0000
2.000000	2400.0000
3.000000	4000.0000

FUNCTION NO. 14 FRICTION FUNC.

NTI(14) = 51

PAGE 11
CARDS E

D0	D1	D2	D3	D4
0.0000	0.0000	0.5000	0.0000	1.0000

FUNCTION IS CONSTANT 0.500000

FUNCTION NO. 19 CF=.25,CREST=.25

NTI(19) = 56

CARDS E

D0	D1	D2	D3	D4
0.0000	0.0000	0.2500	0.0000	0.0000

FUNCTION IS CONSTANT 0.250000

FUNCTION NO. 20 DAMPING COEFF. C=900 NT1(20) = 61

D0	D1	D2	D3	D4
0.0000	1.0000	0.0000	0.0000	1.0000

FIRST PART OF FUNCTION - 5TH DEGREE POLYNOMIAL

A0	A1	A2	A3	A4	A5
0.000000	900.000000	0.000000	0.000000	0.000000	0.000000

FUNCTION NO. 21 RATE OF DEFLEC. NT1(21) = 72

D0	D1	D2	D3	D4
-40.0000	-150.0000	0.0000	0.0000	1.0000

FIRST PART OF FUNCTION - 21 TABULAR POINTS

D	F(D)
-40.000000	0.0000
-30.000000	0.0000
-20.000000	0.0000
-10.000000	0.0000
0.000000	0.0000
5.000000	1.0000
10.000000	1.0000
20.000000	0.9900
30.000000	0.9650
40.000000	0.9280
50.000000	0.8600
60.000000	0.6900
70.000000	0.4750
80.000000	0.3400
90.000000	0.2600
100.000000	0.2000
110.000000	0.1800
120.000000	0.0900
130.000000	0.0600
140.000000	0.0250
150.000000	0.0000

FUNCTION NO. 22 DAMPING COEFF. C=35 NT1(22) = 120

D0	D1	D2	D3	D4
0.0000	1.0000	0.0000	0.0000	1.0000

FIRST PART OF FUNCTION - 5TH DEGREE POLYNOMIAL

A0	A1	A2	A3	A4	A5
0.000000	35.000000	0.000000	0.000000	0.000000	0.000000

FUNCTION NO. 24 DAMPING COEFF. C=0.8 NT1(24) = 131

CARDS E

D0	D1	D2	D3	D4
-1000.0000	-1000.0000	0.0000	0.0000	1.0000

FIRST PART OF FUNCTION - 4 TABULAR POINTS

D	F(D)
-1000.000000	0.6000
-1.000000	0.6000
0.000000	1.0000
1000.000000	1.0000

FUNCTION NO. 25 DAMPING COEFF C=1100 NT1(25) = 145

D0	D1	D2	D3	D4
0.0000	1.0000	0.0000	0.0000	1.0000

FIRST PART OF FUNCTION - 5TH DEGREE POLYNOMIAL

A0	A1	A2	A3	A4	A5
0.000000	1100.000000	0.000000	0.000000	0.000000	0.000000

FUNCTION NO. 26 STIFF SURFACES-LL NT1(26) = 156

CARDS E

D0	D1	D2	D3	D4
0.0000	-4.0000	0.0000	0.0000	1.0000

FIRST PART OF FUNCTION - 8 TABULAR POINTS

D	F(D)
0.000000	0.0000
0.100000	5.0000
0.200000	20.0000
0.300000	40.0000
0.400000	60.0000
2.000000	860.0000
3.000000	2400.0000
4.000000	4000.0000

FUNCTION NO. 31 HARNESS FDF NTI(31) = 178

D0	D1	D2	D3	D4
0.0000	-4.0000	0.0000	0.0000	0.0000

FIRST PART OF FUNCTION - 8 TABULAR POINTS

D	F(D)
0.000000	0.0000
0.010000	150.0000
0.020000	300.0000
0.030000	450.0000
0.050000	850.0000
0.100000	3500.0000
1.000000	35000.0000
4.000000	140000.0000

FUNCTION NO. 32 HARNESS FRICTION NTI(32) = 200

CARDS E

D0	D1	D2	D3	D4
0.0000	0.0000	0.2000	0.0000	0.2000

FUNCTION IS CONSTANT 0.200000

FUNCTION NO. 34

HARNES FRICTION

HTI(34) = 205

D0
0.0000

D1
0.0000

D2
0.9000

D3
0.0000

D4
0.2000

FUNCTION IS CONSTANT 0.900000

CARDS F.1

PLANE	SEGMENT	FORCE DEFLECTION	INERTIAL SPIKE	R FACTOR	G FACTOR	FRICTION COEF. OPT
1- 16	1- 1	13	-20	-21	0	14 1
SEAT. 6 DEGREE OFF H	LT	STIFF SURFACES	DAMPING COEFF. C=900 RATE OF DEFLEC.			FRICTION FUNC.
1- 16	6- 6	13	-25	-21	0	14 1
SEAT. 6 DEGREE OFF H	RUL	STIFF SURFACES	DAMPING COEFF C=1100 RATE OF DEFLEC.			FRICTION FUNC.
1- 16	9- 9	13	-25	-21	0	14 1
SEAT. 6 DEGREE OFF H	LUL	STIFF SURFACES	DAMPING COEFF C=1100 RATE OF DEFLEC.			FRICTION FUNC.
2- 16	1- 1	13	-20	-21	0	14 1
BACK PANEL. 13 DEGR	LT	STIFF SURFACES	DAMPING COEFF. C=900 RATE OF DEFLEC.			FRICTION FUNC.
2- 16	2- 2	13	-20	-21	0	14 1
BACK PANEL. 13 DEGR	CT	STIFF SURFACES	DAMPING COEFF. C=900 RATE OF DEFLEC.			FRICTION FUNC.
2- 16	3- 3	13	-20	-21	0	14 1
BACK PANEL. 13 DEGR	UT	STIFF SURFACES	DAMPING COEFF. C=900 RATE OF DEFLEC.			FRICTION FUNC.
3- 16	8- 8	13	-22	-21	0	14 -1
FLOOR.	RF	STIFF SURFACES	DAMPING COEFF. C=35 RATE OF DEFLEC.			FRICTION FUNC.
3- 16	11- 11	13	-22	-21	0	14 -1
FLOOR.	LF	STIFF SURFACES	DAMPING COEFF. C=35 RATE OF DEFLEC.			FRICTION FUNC.
4- 16	5- 5	13	-22	-21	0	14 1
HEAD PAD. 13 DEGR	H	STIFF SURFACES	DAMPING COEFF. C=35 RATE OF DEFLEC.			FRICTION FUNC.
10- 16	8- 8	13	-22	-21	0	14 1
RUDDER PEDALS.	RF	STIFF SURFACES	DAMPING COEFF. C=35 RATE OF DEFLEC.			FRICTION FUNC.
10- 16	11- 11	13	-22	-21	0	14 1
RUDDER PEDALS.	LF	STIFF SURFACES	DAMPING COEFF. C=35 RATE OF DEFLEC.			FRICTION FUNC.

CARDS F.3

SEGMENT	SEGMENT	FORCE DEFLECTION	INERTIAL SPIKE	R FACTOR	G FACTOR	FRICTION COEF. OPT
2- 2	13- 13	3	0	7	0	19 0
CT	RLA SEGMENT-SEGMENT FCN.			R FACTOR.		CF=.25,CREST= 25
2- 2	15- 15	3	0	7	0	19 0
CT	LLA SEGMENT-SEGMENT FCN.			R FACTOR.		CF=.25,CREST=.25
6- 6	13- 13	3	0	7	0	19 0
RUL	RLA SEGMENT-SEGMENT FCN.			R FACTOR.		CF=.25,CREST=.25
9- 9	15- 15	3	0	7	0	19 0
LUL	LLA SEGMENT-SEGMENT FCN.			R FACTOR.		CF=.25,CREST=.25
13- 13	16- 24	13	-22	-21	0	14 0
RLA	VEH	STIFF SURFACES	DAMPING COEFF. C=35 RATE OF DEFLEC.			FRICTION FUNC.
15- 15	16- 24	13	-22	-21	0	14 0
LLA	VEH	STIFF SURFACES	DAMPING COEFF. C=35 RATE OF DEFLEC.			FRICTION FUNC.

NO. OF HARNESSES = 1

NO. OF BELTS PER HARNESSES = 2

FOR HARNESSES NO. 1 NO. OF POINTS PER BELT = 12 15

HARNESSES NO. 1 BELT NO. 1 FUNCTION NOS. 31 0 0 0 0 REFERENCE SLACK = 0.000 IN.

K	KS	KE	NT	NPD	NDR	FUNCTION NOS.				
1	16	0	109	1	1	0	0	0	0	0
2	1	1	115	0	1	0	0	0	0	34
3	1	1	121	0	1	0	0	0	0	34
4	1	1	127	0	1	0	0	0	0	34
5	1	1	133	0	1	0	0	0	0	34
6	1	23	139	0	1	0	0	0	0	0
7	1	1	145	0	1	0	0	0	0	34
8	1	1	151	0	1	0	0	0	0	34
9	1	1	157	0	1	0	0	0	0	34
10	1	1	163	0	1	0	0	0	0	34
11	1	1	169	0	1	0	0	0	0	34
12	16	0	175	1	1	0	0	0	0	0

CARDS F.8.D

K	BASE REFERENCE (IN.)			ADJUSTED REFERENCE (IN.)			OFFSET (IN.)			PREFERRED DIRECTION (IN.)		
	X	Y	Z	X	Y	Z	X	Y	Z	X	Y	Z
1	13.000	8.000	-10.300	13.000	8.000	-10.300	0.000	0.000	0.000	2.400	22.000	-0.300
2	-1.178	7.075	-0.781	-1.178	7.075	-0.781	0.000	0.000	-0.072	0.000	0.000	0.000
3	-0.029	6.796	-1.534	-0.029	6.795	-1.534	0.000	0.000	-0.072	0.000	0.000	0.000
4	0.910	5.778	-2.283	0.910	5.778	-2.283	0.000	0.000	-0.072	0.000	0.000	0.000
5	2.228	2.355	-3.192	2.228	2.355	-3.192	0.000	0.000	-0.072	0.000	0.000	0.000
6	2.957	0.000	3.059	2.957	0.000	3.059	0.000	0.000	-7.000	0.000	0.000	0.000
7	3.070	-0.080	-4.410	2.344	-0.061	-3.367	0.000	0.000	-0.072	0.000	0.000	0.000
8	1.785	-2.325	-3.343	1.785	-2.325	-3.343	0.000	0.000	-0.072	0.000	0.000	0.000
9	0.011	-5.145	-2.728	0.011	-5.145	-2.728	0.000	0.000	-0.072	0.000	0.000	0.000
10	-0.880	-5.789	-2.282	-0.880	-5.789	-2.282	0.000	0.000	-0.072	0.000	0.000	0.000
11	-2.460	-6.099	-1.200	-2.460	-6.098	-1.200	0.000	0.000	-0.072	0.000	0.000	0.000
12	13.000	-8.000	-10.300	13.000	-8.000	-10.300	0.000	0.000	0.000	2.400	22.000	-0.300

HARNESSES NO. 1 BELT NO. 2 FUNCTION NOS. 31 0 0 0 0 REFERENCE SLACK = 0.000 IN.

K	KS	KE	NT	NPD	NDR	FUNCTION NOS.				
13	16	0	187	1	1	0	0	0	0	0
14	1	1	193	0	1	0	0	0	0	32
15	1	1	199	0	1	0	0	0	0	32
16	1	23	205	0	1	0	0	0	0	0
17	1	1	211	0	1	0	0	0	0	32
18	2	2	217	0	1	0	0	0	0	32
19	2	2	223	0	1	0	0	0	0	32
20	3	3	229	0	1	0	0	0	0	32
21	3	3	235	0	1	0	0	0	0	32
22	3	3	241	0	1	0	0	0	0	32
23	3	3	247	0	1	0	0	0	0	32
24	3	22	253	0	0	0	0	0	0	32
25	3	3	259	0	1	0	0	0	0	32
26	3	3	265	0	1	0	0	0	0	32
27	16	0	271	1	1	0	0	0	0	0

CARDS F.8.D

K	BASE REFERENCE (IN.)			ADJUSTED REFERENCE (IN.)			OFFSET (IN.)			PREFERRED DIRECTION (IN.)		
	X	Y	Z	X	Y	Z	X	Y	Z	X	Y	Z
13	13.000	-8.000	-10.300	13.000	-8.000	-10.300	0.000	0.000	0.000	0.700	17.500	-21.300
14	-2.445	-6.101	-1.213	-2.445	-6.101	-1.213	0.000	0.000	-0.072	0.000	0.000	0.000
15	-0.960	-6.000	-2.500	-0.906	-5.661	-2.359	0.000	0.000	-0.072	0.000	0.000	0.000
16	0.000	-5.700	3.800	0.000	-5.367	3.578	0.000	0.000	-7.000	0.000	0.000	0.000
17	0.010	-4.000	-4.500	0.008	-3.061	-3.443	0.000	0.000	-0.072	0.000	0.000	0.000
18	1.818	-5.439	-1.820	1.818	-5.439	-1.820	0.000	0.000	-0.011	0.000	0.000	0.000
19	2.500	-2.500	-1.500	3.397	-3.397	-2.038	0.000	0.000	-0.011	0.000	0.000	0.000
20	3.000	-1.500	6.500	2.777	-1.388	6.016	0.000	0.000	-0.872	0.000	0.000	0.000
21	4.487	-0.133	3.734	4.487	-0.133	3.734	0.000	0.000	-0.872	0.000	0.000	0.000
22	4.421	3.319	-1.662	4.421	3.319	-1.662	0.000	0.000	-0.872	0.000	0.000	0.000
23	0.879	4.202	-5.672	0.879	4.202	-5.672	0.000	0.000	-0.872	0.000	0.000	0.000
24	0.300	0.200	-2.800	0.320	0.213	-2.985	0.000	4.000	-3.500	0.000	0.000	0.000
25	-1.000	4.300	-6.000	-0.955	4.105	-5.728	0.000	0.000	-0.872	0.000	0.000	0.000

27 0.000 5.000 -35.200 0.000 5.000 -35.200 0.000 0.000 0.000 0.700 17.500 -21.300

ZPLT(X) ZPLT(Y) ZPLT(Z) I1 J1 I2 J2 I3 SPLT(1) SPLT(2) SPLT(3)
0. 0. 0. 0 0 0 0 0 0 10.00 6.00 1.00

INITIAL POSITIONS (INERTIAL REFERENCE)

CARDS G.2

SEGMENT NO. SEG	LINEAR POSITION (IN.)			LINEAR VELOCITY (IN./SEC.)		
	X	Y	Z	X	Y	Z
1 LT	14.38100	0.00000	-13.75000	0.00000	0.00000	0.00000
2 CT	13.16068	0.00000	-19.07188	0.00000	0.00000	0.00000
3 UT	11.31383	0.00000	-26.93796	0.00000	0.00000	0.00000
4 N	9.28353	0.00000	-35.53137	0.00000	0.00000	0.00000
5 H	7.77521	0.00000	-41.83338	0.00000	0.00000	0.00000
6 RUL	22.94073	3.42000	-14.48930	0.00000	0.00000	0.00000
7 RLL	38.16022	3.42000	-10.39045	0.00000	0.00000	0.00000
8 RF	48.12719	3.42000	-4.45598	0.00000	0.00000	0.00000
9 LUL	22.94073	-3.42000	-14.48930	0.00000	0.00000	0.00000
10 LLL	38.16022	-3.42000	-10.39045	0.00000	0.00000	0.00000
11 LF	48.12719	-3.42000	-4.45598	0.00000	0.00000	0.00000
12 RUA	12.34164	6.24000	-27.13188	0.00000	0.00000	0.00000
13 RLA	22.75808	6.24000	-21.48521	0.00000	0.00000	0.00000
14 LUA	12.34164	-6.24000	-27.13188	0.00000	0.00000	0.00000
15 LLA	22.75808	-6.24000	-21.48521	0.00000	0.00000	0.00000

INITIAL ANGULAR ROTATION AND VELOCITY

CARDS G.3

SEGMENT NO. SEG	ANGULAR ROTATION (DEG)			ANGULAR VELOCITY (DEG/SEC.)			IYPR
	YAW	PITCH	ROLL	X	Y	Z	
1 LT	0.00000	12.90000	0.00000	0.00000	0.00000	0.00000	3 2 1 0
2 CT	0.00000	12.95000	0.00000	0.00000	0.00000	0.00000	3 2 1 0
3 UT	0.00000	13.28000	0.00000	0.00000	0.00000	0.00000	3 2 1 0
4 N	0.00000	13.46000	0.00000	0.00000	0.00000	0.00000	3 2 1 0
5 H	0.00000	13.46000	0.00000	0.00000	0.00000	0.00000	3 2 1 0
6 RUL	0.00000	92.90000	0.00000	0.00000	0.00000	0.00000	3 2 1 0
7 RLL	0.00000	48.65000	0.00000	0.00000	0.00000	0.00000	3 2 1 0
8 RF	0.00000	128.80000	0.00000	0.00000	0.00000	0.00000	3 2 1 0
9 LUL	0.00000	92.90000	0.00000	0.00000	0.00000	0.00000	3 2 1 0
10 LLL	0.00000	48.65000	0.00000	0.00000	0.00000	0.00000	3 2 1 0
11 LF	0.00000	128.80000	0.00000	0.00000	0.00000	0.00000	3 2 1 0
12 RUA	0.00000	24.50000	0.00000	0.00000	0.00000	0.00000	3 2 1 0
13 RLA	0.00000	85.00000	0.00000	0.00000	0.00000	0.00000	3 2 1 0
14 LUA	0.00000	24.50000	0.00000	0.00000	0.00000	0.00000	3 2 1 0
15 LLA	0.00000	85.00000	0.00000	0.00000	0.00000	0.00000	3 2 1 0

LINEAR AND ANGULAR VELOCITIES HAVE BEEN SET EQUAL TO THE INITIAL VEHICLE VELOCITIES.

HBPLAY TIME = 0.000 MSEC. NH,NB,NPTS NT= 1 1 11 103
 NL(1)= 1 2 3 4 5 6 8 9 10 11 12
 BB = 4.124 1.402 1.574 3.779 2.557 2.656 3.388 1.187 1.940 4.738

HBPLAY TIME = 0.000 MSEC. NH,NB,NPTS NT= 1 2 7 181
 NL(1)= 13 14 18 21 22 23 27
 BB = 4.748 7.394 6.842 6.406 5.423 10.829

TABULAR TIME HISTORY CONTROL PARAMETERS

TYPE	KSG	SELECTED SEGMENTS OR JOINTS
H.1	3	3 -5 5
REF		0 1 16
H.2	3	3 5 5
REF		0 0 3
H.3	3	3 5 5
REF		0 0 3
H.4	3	3 5 5
REF		0 0 16
H.5	3	3 5 5
REF		0 0 5
H.6	3	3 5 5
REF		0 0 3
H.7	2	3 4
REF		0 0
H.8	0	
REF		
H.9	1	4
REF		5
H.10	15	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
REF		16

SEGMENT	(INERTIAL)			(LOCAL)			(LOCAL)		
	ANGULAR ROTATION (DEG)			ANGULAR VELOCITY (RAD/SEC.)			ANGULAR ACCELERATION (RAD/SEC.**2)		
	YAW	PITCH	ROLL	X	Y	Z	X	Y	Z
1 LT	0.0000	12.9000	0.0000	0.00000	0.00000	0.00000	0.000000	-26.834649	0.000000
2 CT	0.0000	12.9500	0.0000	0.00000	0.00000	0.00000	0.000000	40.304245	0.000000
3 UT	0.0000	13.2800	0.0000	0.00000	0.00000	0.00000	0.000000	-0.859459	0.000000
4 N	0.0000	13.4600	0.0000	0.00000	0.00000	0.00000	0.000000	13.637948	0.000000
5 H	0.0000	13.4600	0.0000	0.00000	0.00000	0.00000	0.000000	-1.165343	0.000000
6 RUL	0.0000	92.9000	0.0000	0.00000	0.00000	0.00000	0.000000	-2.405246	0.000000
7 RLL	0.0000	48.6500	0.0000	0.00000	0.00000	0.00000	0.000000	-12.215669	0.000000
8 RF	0.0000	128.8000	0.0000	0.00000	0.00000	0.00000	0.000000	56.318297	0.000000
9 LUL	0.0000	92.9000	0.0000	0.00000	0.00000	0.00000	0.000000	-2.405246	0.000000
10 LLL	0.0000	48.6500	0.0000	0.00000	0.00000	0.00000	0.000000	-12.215669	0.000000
11 LF	0.0000	128.8000	0.0000	0.00000	0.00000	0.00000	0.000000	56.318297	0.000000
12 RUA	0.0000	24.5000	0.0000	0.00000	0.00000	0.00000	0.000000	-6.917594	0.000000
13 RLA	0.0000	85.0000	0.0000	0.00000	0.00000	0.00000	0.000000	-34.722447	0.000000
14 LUA	0.0000	24.5000	0.0000	0.00000	0.00000	0.00000	0.000000	-6.917594	0.000000
15 LLA	0.0000	85.0000	0.0000	0.00000	0.00000	0.00000	0.000000	-34.722447	0.000000
16 VEH	0.0000	0.0000	0.0000	0.00000	0.00000	0.00000	0.000000	0.000000	0.000000

SEGMENT	(INERTIAL)			(INERTIAL)			(INERTIAL)		
	LINEAR POSITION (IN.)			LINEAR VELOCITY (IN./SEC.)			LINEAR ACCELERATIONS (G'S)		
	X	Y	Z	X	Y	Z	X	Y	Z
1 LT	14.3810	0.0000	-13.7500	0.00000	0.00000	0.00000	0.017822	0.000000	-0.088444
2 CT	13.1607	0.0000	-19.0719	0.00000	0.00000	0.00000	0.114863	0.000000	-0.110519
3 UT	11.3138	0.0000	-26.9380	0.00000	0.00000	0.00000	-0.038031	0.000000	-0.075446
4 N	9.2835	0.0000	-35.5314	0.00000	0.00000	0.00000	-0.042274	0.000000	-0.074372
5 H	7.7752	0.0000	-41.8334	0.00000	0.00000	0.00000	-0.047117	0.000000	-0.073212
6 RUL	22.9407	3.4200	-14.4893	0.00000	0.00000	0.00000	0.041548	0.000000	-0.039498
7 RLL	38.1602	3.4200	-10.3904	0.00000	0.00000	0.00000	-0.100994	0.000000	0.188268
8 RF	48.1272	3.4200	-4.4560	0.00000	0.00000	0.00000	-0.187590	0.000000	-0.183587
9 LUL	22.9407	-3.4200	-14.4893	0.00000	0.00000	0.00000	0.041548	0.000000	-0.039498
10 LLL	38.1602	-3.4200	-10.3904	0.00000	0.00000	0.00000	-0.100994	0.000000	0.188268
11 LF	48.1272	-3.4200	-4.4560	0.00000	0.00000	0.00000	-0.187590	0.000000	-0.183587
12 RUA	12.3416	6.2400	-27.1319	0.00000	0.00000	0.00000	-0.114274	0.000000	-0.038215
13 RLA	22.7581	6.2400	-21.4852	0.00000	0.00000	0.00000	-0.266915	0.000000	0.736709
14 LUA	12.3416	-6.2400	-27.1319	0.00000	0.00000	0.00000	-0.114274	0.000000	-0.038215
15 LLA	22.7581	-6.2400	-21.4852	0.00000	0.00000	0.00000	-0.266915	0.000000	0.736709
16 VEH	0.0000	0.0000	0.0000	0.00000	0.00000	0.00000	0.000000	0.000000	0.000000

SEGMENT	(INERTIAL) U1 ARRAY (IN./SEC.**2)			(LOCAL) U2 ARRAY (RAD/SEC.**2)			KINETIC ENERGY (LB.- IN.)				
	EXTERNAL LINEAR ACCELERATIONS			EXTERNAL ANGULAR ACCELERATIONS			LINEAR	ANGULAR	TOTAL		
	X	Y	Z	X	Y	Z					
1	LT	-0.12890+03	0.00000+00	-0.11350+04	0.000000+00	-0.480150+02	0.000000+00	0.000000+00	0.000000+00	0.000000+00	
2	CT	0.00000+00	0.00000+00	0.38610+03	0.000000+00	0.000000+00	0.000000+00	0.000000+00	0.000000+00	0.000000+00	
3	UT	0.69400+02	0.00000+00	0.37010+03	0.000000+00	-0.318610+01	0.000000+00	0.000000+00	0.000000+00	0.000000+00	
4	N	0.00000+00	0.00000+00	0.38610+03	0.000000+00	0.000000+00	0.000000+00	0.000000+00	0.000000+00	0.000000+00	
5	H	0.78220+02	0.00000+00	0.36800+03	-0.133040-18	-0.267210+00	0.308510-16	0.000000+00	0.000000+00	0.000000+00	
6	RUL	-0.53090+02	0.00000+00	-0.11980+03	0.854930-17	0.218270+02	0.123730-14	0.000000+00	0.000000+00	0.000000+00	
7	RLL	0.00000+00	0.00000+00	0.38610+03	0.000000+00	0.000000+00	0.000000+00	0.000000+00	0.000000+00	0.000000+00	
8	RF	-0.95110+01	0.00000+00	-0.52100+03	0.000000+00	-0.176100+03	0.000000+00	0.000000+00	0.000000+00	0.000000+00	
9	LUL	-0.53090+02	0.00000+00	-0.11980+03	0.854930-17	0.218270+02	0.123730-14	0.000000+00	0.000000+00	0.000000+00	
10	LLL	0.00000+00	0.00000+00	0.38610+03	0.000000+00	0.000000+00	0.000000+00	0.000000+00	0.000000+00	0.000000+00	
11	LF	-0.95110+01	0.00000+00	-0.52100+03	0.000000+00	-0.176100+03	0.000000+00	0.000000+00	0.000000+00	0.000000+00	
12	RUA	0.00000+00	0.00000+00	0.38610+03	0.000000+00	0.000000+00	0.000000+00	0.000000+00	0.000000+00	0.000000+00	
13	RLA	0.00000+00	0.00000+00	0.38610+03	0.000000+00	0.000000+00	0.000000+00	0.000000+00	0.000000+00	0.000000+00	
14	LUA	0.00000+00	0.00000+00	0.38610+03	0.000000+00	0.000000+00	0.000000+00	0.000000+00	0.000000+00	0.000000+00	
15	LLA	0.00000+00	0.00000+00	0.38610+03	0.000000+00	0.000000+00	0.000000+00	0.000000+00	0.000000+00	0.000000+00	
							TOTAL BODY KINETIC ENERGY				
							0.000000+00	0.000000+00	0.000000+00		

JOINT	IPIN	(INERTIAL) JOINT FORCES (LB.)			(INERTIAL) JOINT TORQUES (IN. LB.)			RELATIVE ANGULAR VELOCITY (RAD/SEC.)	
		X	Y	Z	X	Y	Z		
1	P	0	-0.1770+02	0.0000+00	-0.1000+03	0.00000+00	0.00000+00	0.00000+00	0.000
2	W	0	-0.1920+02	0.0000+00	-0.8590+02	0.00000+00	0.00000+00	0.00000+00	0.000
3	NP	0	-0.3120+01	0.0000+00	-0.1580+02	0.00000+00	0.00000+00	0.00000+00	0.000
4	HP	0	-0.2980+01	0.0000+00	-0.1220+02	0.00000+00	0.00000+00	0.00000+00	0.000
5	RH	0	0.2740+01	0.0000+00	0.6530+00	0.00000+00	0.00000+00	0.00000+00	0.000
6	RK	1	-0.1320+01	0.0000+00	-0.5500+01	0.00000+00	0.00000+00	0.00000+00	0.000
7	RA	0	-0.3390+00	0.0000+00	0.2420+01	0.00000+00	0.00000+00	0.00000+00	0.000
8	LH	0	0.2740+01	0.0000+00	0.6530+00	0.00000+00	0.00000+00	0.00000+00	0.000
9	LK	1	-0.1320+01	0.0000+00	-0.5500+01	0.00000+00	0.00000+00	0.00000+00	0.000
10	LA	0	-0.3390+00	0.0000+00	0.2420+01	0.00000+00	0.00000+00	0.00000+00	0.000
11	RS	0	-0.2210+01	0.0000+00	-0.7310+01	0.00000+00	0.00000+00	0.00000+00	0.000
12	RE	1	-0.1580+01	0.0000+00	-0.1550+01	0.00000+00	0.00000+00	0.00000+00	0.000
13	LS	0	-0.2210+01	0.0000+00	-0.7310+01	0.00000+00	0.00000+00	0.00000+00	0.000
14	LE	1	-0.1580+01	0.0000+00	-0.1550+01	0.00000+00	0.00000+00	0.00000+00	0.000

POINT NO.	POINT INDEX	SEGMENT NO.	LENGTH (IN.)	BELT STRAIN ENERGY LOSS (IN. LB.)	(LOCAL OR ELLIPSOID) REFERENCE POINT (IN.)			(INERTIAL) BELT FORCES (LB.)			PENETRATION ENERGY LOSS (IN. LB.)
					X	Y	Z	X	Y	Z	
BELT NO.	1 OF HARNES NO.			1							
1	1	16	0.000	0.000	13.000	8.000	-10.300	0.000	0.000	0.000	0.000
2	2	1	4.124	0.000	-1.178	7.075	-0.781	0.000	0.000	0.000	0.000
3	3	1	1.402	0.000	-0.029	6.795	-1.534	0.000	0.000	0.000	0.000
4	4	1	1.574	0.000	0.910	5.778	-2.283	0.000	0.000	0.000	0.000
5	5	1	3.779	0.000	2.228	2.355	-3.192	0.000	0.000	0.000	0.000
6	6	1	2.557	0.000	2.957	0.000	3.059	0.000	0.000	0.000	0.000
7	8	1	2.656	0.000	1.785	-2.325	-3.343	0.000	0.000	0.000	0.000
8	9	1	3.388	0.000	0.011	-5.145	-2.728	0.000	0.000	0.000	0.000
9	10	1	1.187	0.000	-0.880	-5.789	-2.282	0.000	0.000	0.000	0.000
10	11	1	1.940	0.000	-2.460	-6.098	-1.200	0.000	0.000	0.000	0.000
11	12	16	4.738	0.000	13.000	-8.000	-10.300	0.000	0.000	0.000	0.000

TOTAL BELT ENERGY LOSS 0.000 0.000

BELT NO.	2 OF HARNES NO.			1							
12	13	16	0.000	0.000	13.000	-8.000	-10.300	0.000	0.000	0.000	0.000
13	14	1	4.748	0.000	-2.445	-6.101	-1.213	0.000	0.000	0.000	0.000
14	18	2	7.394	0.000	1.818	-5.439	-1.820	0.000	0.000	0.000	0.000
15	21	3	6.842	0.000	4.487	-0.133	3.734	0.000	0.000	0.000	0.000
16	22	3	6.406	0.000	4.421	3.319	-1.662	0.000	0.000	0.000	0.000
17	23	3	5.423	0.000	0.879	4.202	-5.672	0.000	0.000	0.000	0.000
18	27	16	10.829	0.000	0.000	5.000	-35.200	0.000	0.000	0.000	0.000

TOTAL BELT ENERGY LOSS 0.000 0.000

TOTAL HARNES ENERGY LOSS 0.000 0.000

HPTURB ITER = 10 AT TIME = 16.000 MSEC. DELMAX = 0.010537 SCALE = 1.000000

HPTURB ITER = 10 AT TIME = 18.000 MSEC. DELMAX = 0.009232 SCALE = 1.000000

HPTURB ITER = 10 AT TIME = 19.000 MSEC. DELMAX = 0.009627 SCALE = 1.000000

HPTURB ITER = 10 AT TIME = 20.000 MSEC. DELMAX = 0.010050 SCALE = 1.000000

HPTURB ITER = 10 AT TIME = 21.000 MSEC. DELMAX = 0.010116 SCALE = 1.000000

HBPLAY TIME = 22.000 MSEC. NH,NB,NPTS NT= 1 2 8 181
 NL(1)= 13 14 18 21 22 23 24 27
 BB = 4.748 7.394 6.842 6.406 5.423 0.491 10.338

HPTURB ITER = 10 AT TIME = 22.000 MSEC. DELMAX = 0.012349 SCALE = 1.000000

HPTURB ITER = 10 AT TIME = 23.000 MSEC. DELMAX = 0.011517 SCALE = 1.000000

HPTURB ITER = 10 AT TIME = 24.000 MSEC. DELMAX = 0.011234 SCALE = 1.000000

HPTURB ITER = 10 AT TIME = 25.000 MSEC. DELMAX = 0.011569 SCALE = 1.000000

HPTURB ITER = 10 AT TIME = 26.000 MSEC. DELMAX = 0.012320 SCALE = 1.000000

HPTURB ITER = 10 AT TIME = 28.000 MSEC. DELMAX = 0.010639 SCALE = 1.000000

HPTURB ITER = 10 AT TIME = 29.000 MSEC. DELMAX = 0.012461 SCALE = 1.000000

HPTURB ITER = 10 AT TIME = 30.000 MSEC. DELMAX = 0.014987 SCALE = 1.000000

HBPLAY TIME = 31.000 MSEC. NH,NB,NPTS NT= 1 2 7 181
 NL(1)= 13 14 18 21 22 24 27
 BB = 4.748 7.394 6.842 6.406 5.572 10.679

HPTURB ITER = 10 AT TIME = 31.000 MSEC. DELMAX = 0.021341 SCALE = 1.000000

HBPLAY TIME = 32.000 MSEC. NH,NB,NPTS NT= 1 2 8 181
 NL(1)= 13 14 18 21 22 23 24 27
 BB = 4.748 7.394 6.842 6.406 5.066 0.260 10.926

HPTURB ITER = 10 AT TIME = 32.000 MSEC. DELMAX = 0.016547 SCALE = 1.000000

HBPLAY TIME = 33.000 MSEC. NH,NB,NPTS NT= 1 2 7 181

BB = 4.748 7.394 6.842 6.406 5.066 11.186
 HPTURB ITER = 10 AT TIME = 33.000 MSEC. DELMAX = 0.036770 SCALE = 1.000000
 HBPLAY TIME = 34.000 MSEC. NH,NB,NPTS NT= 1 2 8 181
 NL(1)= 13 14 18 21 22 23 24 27
 BB = 4.748 7.394 6.842 6.406 5.066 0.289 10.896
 HPTURB ITER = 10 AT TIME = 34.000 MSEC. DELMAX = 0.041013 SCALE = 1.000000
 HBPLAY TIME = 35.000 MSEC. NH,NB,NPTS NT= 1 2 7 181
 NL(1)= 13 14 18 21 22 24 27
 BB = 4.748 7.394 6.842 6.406 5.216 11.035
 HPTURB ITER = 10 AT TIME = 35.000 MSEC. DELMAX = 0.021221 SCALE = 1.000000
 HBPLAY TIME = 36.000 MSEC. NH,NB,NPTS NT= 1 2 8 181
 NL(1)= 13 14 18 21 22 23 24 27
 BB = 4.748 7.394 6.842 6.406 4.847 0.195 11.210
 HPTURB ITER = 10 AT TIME = 36.000 MSEC. DELMAX = 0.012648 SCALE = 1.000000
 HPTURB ITER = 10 AT TIME = 37.000 MSEC. DELMAX = 0.015643 SCALE = 1.000000
 HPTURB ITER = 10 AT TIME = 38.000 MSEC. DELMAX = 0.011152 SCALE = 1.000000
 HPTURB ITER = 10 AT TIME = 39.000 MSEC. DELMAX = 0.014391 SCALE = 1.000000
 HPTURB ITER = 10 AT TIME = 40.000 MSEC. DELMAX = 0.017690 SCALE = 1.000000
 HPTURB ITER = 10 AT TIME = 41.000 MSEC. DELMAX = 0.015081 SCALE = 1.000000
 HPTURB ITER = 10 AT TIME = 42.000 MSEC. DELMAX = 0.017005 SCALE = 1.000000
 HBPLAY TIME = 43.000 MSEC. NH,NB,NPTS NT= 1 2 7 181
 NL(1)= 13 14 18 21 22 24 27
 BB = 4.748 7.394 6.842 6.406 4.910 11.342
 HPTURB ITER = 10 AT TIME = 43.000 MSEC. DELMAX = 0.053120 SCALE = 1.000000
 HBPLAY TIME = 44.000 MSEC. NH,NB,NPTS NT= 1 2 8 181
 NL(1)= 13 14 18 21 22 23 24 27
 BB = 4.748 7.394 6.842 6.406 4.496 0.075 11.681
 HPTURB ITER = 10 AT TIME = 44.000 MSEC. DELMAX = 0.013932 SCALE = 1.000000
 HPTURB ITER = 10 AT TIME = 45.000 MSEC. DELMAX = 0.012674 SCALE = 1.000000
 HBPLAY TIME = 46.000 MSEC. NH,NB,NPTS NT= 1 2 7 181
 NL(1)= 13 18 21 22 23 24 27
 BB = 12.143 6.842 6.406 4.496 0.055 11.701
 HPTURB ITER = 10 AT TIME = 46.000 MSEC. DELMAX = 0.014891 SCALE = 1.000000
 HBPLAY TIME = 47.000 MSEC. NH,NB,NPTS NT= 1 1 10 103
 NL(1)= 1 3 4 5 6 8 9 10 11 12
 BB = 5.526 1.574 3.779 2.557 2.656 3.388 1.187 1.940 4.738
 HPTURB ITER = 10 AT TIME = 47.000 MSEC. DELMAX = 0.018982 SCALE = 1.000000
 HBPLAY TIME = 48.000 MSEC. NH,NB,NPTS NT= 1 2 6 181
 NL(1)= 13 18 21 22 24 27
 BB = 12.143 6.842 6.406 4.534 11.718
 HPTURB ITER = 10 AT TIME = 48.000 MSEC. DELMAX = 0.064777 SCALE = 1.000000
 HBPLAY TIME = 49.000 MSEC. NH,NB,NPTS NT= 1 2 7 181
 NL(1)= 13 18 21 22 23 24 27
 BB = 12.143 6.842 6.406 4.192 0.045 12.015
 HPTURB ITER = 10 AT TIME = 49.000 MSEC. DELMAX = 0.010077 SCALE = 1.000000
 HPTURB ITER = 10 AT TIME = 50.000 MSEC. DELMAX = 0.010283 SCALE = 1.000000
 HPTURB ITER = 10 AT TIME = 51.000 MSEC. DELMAX = 0.012303 SCALE = 1.000000
 HBPLAY TIME = 52.000 MSEC. NH,NB,NPTS NT= 1 2 6 181
 NL(1)= 13 18 21 22 24 27
 BB = 12.143 6.842 6.406 4.222 12.029

HPTURB ITER = 10 AT TIME = 52.000 MSEC. DELMAX = 0.055553 SCALE = 1.000000

HPTURB ITER = 10 AT TIME = 53.000 MSEC. DELMAX = 0.035482 SCALE = 1.000000

HBPLAY TIME = 54.000 MSEC. NH,NB,NPTS NT= 1 1 9 103
NL(1)= 1 3 4 5 6 8 9 10 12
BB = 5.526 1.574 3.779 2.557 2.656 3.388 1.187 6.678

HPTURB ITER = 10 AT TIME = 54.000 MSEC. DELMAX = 0.024287 SCALE = 1.000000

HPTURB ITER = 10 AT TIME = 55.000 MSEC. DELMAX = 0.015678 SCALE = 1.000000

HPTURB ITER = 10 AT TIME = 56.000 MSEC. DELMAX = 0.011486 SCALE = 1.000000

HBPLAY TIME = 57.000 MSEC. NH,NB,NPTS NT= 1 2 7 181
NL(1)= 13 18 19 21 22 24 27
BB = 12.143 1.887 4.955 6.406 3.880 12.372

HPTURB ITER = 10 AT TIME = 57.000 MSEC. DELMAX = 0.385309 SCALE = 0.259532

HBPLAY TIME = 58.000 MSEC. NH,NB,NPTS NT= 1 2 8 181
NL(1)= 13 18 19 20 21 22 24 27
BB = 12.143 1.887 1.817 3.139 6.406 3.880 12.372

HPTURB ITER = 10 AT TIME = 58.000 MSEC. DELMAX = 0.443658 SCALE = 0.225399

HBPLAY TIME = 59.000 MSEC. NH,NB,NPTS NT= 1 2 9 181
NL(1)= 13 16 18 19 20 21 22 24 27
BB = 7.889 4.253 1.887 1.817 3.139 6.406 3.974 12.277

HPTURB ITER = 10 AT TIME = 59.000 MSEC. DELMAX = 0.052001 SCALE = 1.000000

HBPLAY TIME = 60.000 MSEC. NH,NB,NPTS NT= 1 2 8 181
NL(1)= 13 18 19 20 21 22 24 27
BB = 12.143 1.887 1.817 3.139 6.406 4.086 12.165

HPTURB ITER = 10 AT TIME = 60.000 MSEC. DELMAX = 0.057808 SCALE = 1.000000

HBPLAY TIME = 61.000 MSEC. NH,NB,NPTS NT= 1 2 7 181
NL(1)= 13 18 20 21 22 24 27
BB = 12.143 3.703 3.139 6.406 4.156 12.096

HPTURB ITER = 10 AT TIME = 61.000 MSEC. DELMAX = 0.015872 SCALE = 1.000000

HPTURB ITER = 10 AT TIME = 62.000 MSEC. DELMAX = 0.022130 SCALE = 1.000000

HPTURB ITER = 10 AT TIME = 63.000 MSEC. DELMAX = 0.015337 SCALE = 1.000000

HBPLAY TIME = 64.000 MSEC. NH,NB,NPTS NT= 1 2 6 181
NL(1)= 13 18 20 21 22 27
BB = 12.143 3.703 3.139 6.406 16.252

HPTURB ITER = 10 AT TIME = 64.000 MSEC. DELMAX = 0.011067 SCALE = 1.000000

HBPLAY TIME = 66.000 MSEC. NH,NB,NPTS NT= 1 1 8 103
NL(1)= 1 3 4 5 6 8 9 12
BB = 5.526 1.574 3.779 2.557 2.656 3.388 7.865

HPTURB ITER = 10 AT TIME = 67.000 MSEC. DELMAX = 0.011853 SCALE = 1.000000

HPTURB ITER = 10 AT TIME = 69.000 MSEC. DELMAX = 0.013454 SCALE = 1.000000

DINT CONV. TEST 72.000 N ANG VEL 22.55 0.2652E-02 0.1176E-03 0.1000E-03 0.1000E-03 0.1000E-03

TEST FAILED AT TIME = 0.072000 FOR H = 0.001000

DINT CONV. TEST 73.000 N ANG VEL 23.86 0.3624E-02 0.1518E-03 0.1000E-03 0.1000E-03 0.1000E-03

TEST FAILED AT TIME = 0.073000 FOR H = 0.001000

DINT CONV. TEST 74.000 N ANG VEL 26.11 0.3039E-02 0.1164E-03 0.1000E-03 0.1000E-03 0.1000E-03

TEST FAILED AT TIME = 0.074000 FOR H = 0.001000

SEGMENT	(INERTIAL) ANGULAR ROTATION (DEG)			(LOCAL) ANGULAR VELOCITY (RAD/SEC.)			(LOCAL) ANGULAR ACCELERATION (RAD/SEC.**2)		
	YAW	PITCH	ROLL	X	Y	Z	X	Y	Z
	1 LT	10.9078	40.4616	-2.5237	-3.50101	6.77176	3.01702	-712.525841	-923.312215
2 CT	5.8648	8.3877	-35.0629	-28.78011	38.13662	26.32712	2669.823703	4863.705052	714.436129
3 UT	22.6603	16.1876	4.2073	-4.55898	-8.33212	11.28470	-833.189019	-740.536145	24.730107
4 N	9.4134	-28.3216	9.0212	3.10468	-11.10456	11.62142	-474.529280	-2359.202265	525.452347
5 H	-14.7037	-62.0894	29.6482	16.06835	-58.33883	6.84809	321.909908	-351.659929	252.225950
6 RUL	11.0166	100.4701	10.5736	-0.26043	-3.94123	-2.59116	-82.766870	-780.201793	66.972957
7 RLL	-2.0078	53.9069	-3.2449	-2.05452	9.15294	-1.60027	12.275725	-89.117047	79.212811
8 RF	-1.2125	146.0433	2.4774	3.41634	1.03537	-1.56581	-92.295830	747.087281	-23.919556
9 LUL	4.3688	101.7927	4.4680	-0.37199	-3.33541	-0.44435	-290.296221	-679.956631	528.292676
10 LLL	-0.9289	45.3497	-1.2981	-0.57596	5.97768	0.06395	278.790377	-4.508498	528.778814
11 LF	-0.5610	155.8698	1.6050	1.79239	0.13205	-0.59660	-177.520464	1209.576339	196.625307
12 RUA	-75.0950	70.5657	-80.2833	-6.23227	-4.14211	-1.63954	244.397026	-3261.631891	-168.345676
13 RLA	-21.0534	51.2884	-31.6271	-6.18955	1.21106	1.79408	113.635226	2487.522774	-303.100899
14 LUA	-17.8558	62.4794	-23.3643	-2.95595	-0.00123	5.46149	322.014086	-2838.181399	888.201217
15 LLA	-16.0460	60.3973	-21.7744	-2.73881	-15.25083	5.57354	441.700349	4146.746559	916.611615
16 VEH	0.0000	0.0000	0.0000	0.00000	0.00000	0.00000	0.000000	0.000000	0.000000

SEGMENT	(INERTIAL) LINEAR POSITION (IN.)			(INERTIAL) LINEAR VELOCITY (IN./SEC.)			(INERTIAL) LINEAR ACCELERATIONS (G'S)		
	X	Y	Z	X	Y	Z	X	Y	Z
1 LT	6.0377	0.0702	-13.0395	-334.65671	-2.92122	-92.21272	1.118456	-3.419755	-24.241372
2 CT	3.5221	-1.5083	-17.2697	-403.89532	-63.36301	-39.55188	-7.411998	0.308363	-23.341359
3 UT	1.5893	-2.7194	-24.7657	-399.28403	-115.11128	-20.63141	-8.535068	3.408484	-20.077252
4 N	-0.4611	-2.8941	-33.1662	-319.46157	-121.58509	-38.64033	16.009901	-4.901599	-22.093312
5 H	4.8568	-1.0902	-36.0984	-178.47710	-69.83431	287.80272	-29.256214	-10.703850	8.134162
6 RUL	13.6013	3.3950	-14.9332	-345.17072	-4.32318	-59.80904	2.896528	-1.986709	-14.025632
7 RLL	29.0739	3.6383	-12.6201	-300.76311	12.64266	-73.16830	3.883535	0.620604	6.154392
8 RF	39.4986	3.6714	-8.4455	-258.02278	23.08008	-137.61724	0.173190	0.601745	-0.915789
9 LUL	15.0441	-3.3466	-14.9210	-314.88548	2.85212	-60.25035	6.374569	3.680291	-3.334002
10 LLL	29.7934	-3.3024	-12.0614	-278.81100	10.63011	-57.36381	9.156247	6.479980	13.521611
11 LF	39.2761	-3.3229	-7.6097	-244.93381	11.69052	-92.38042	4.608846	0.164379	1.561647
12 RUA	3.1574	3.4023	-29.0240	-426.99852	-109.61718	-32.04897	-2.643843	6.759738	6.506879
13 RLA	15.1710	5.9986	-24.3530	-429.54688	-25.03050	-46.34975	23.116917	3.958924	9.885922
14 LUA	7.1965	-8.0306	-27.9271	-282.38560	-78.35066	-29.72059	-9.173876	2.088389	27.959874
15 LLA	19.2594	-6.2444	-21.8664	-343.80786	-23.50199	71.96752	11.107169	-20.329519	-12.311190
16 VEH	-12.3548	0.0000	0.0000	-308.87040	0.00000	0.00000	0.000000	0.000000	0.000000

SEGMENT	(INERTIAL)			(LOCAL)			KINETIC ENERGY			
	U1 ARRAY (IN./SEC.**2)			U2 ARRAY (RAD/SEC.**2)			(LB.- IN.)			
	EXTERNAL	LINEAR	ACCELERATIONS	EXTERNAL	ANGULAR	ACCELERATIONS	LINEAR	ANGULAR	TOTAL	
	X	Y	Z	X	Y	Z				
1	LT	-0.3009D+03	0.1559D+02	-0.2698D+05	-0.60308D+03	-0.27927D+04	-0.22971D+03	0.54266D+04	0.41308D+02	0.54679D+04
2	CT	0.2569D+03	-0.5220D+03	-0.6204D+03	0.88429D+04	0.83818D+04	0.71444D+03	0.28620D+04	0.54259D+03	0.34046D+04
3	UT	-0.6894D+03	-0.2430D+03	0.4419D+03	-0.69029D+03	-0.15455D+03	0.16513D+02	0.12032D+05	0.28218D+03	0.12314D+05
4	N	0.0000D+00	0.0000D+00	0.3861D+03	-0.26984D+04	0.10411D+05	0.52545D+03	0.50402D+03	0.33066D+01	0.50733D+03
5	H	0.0000D+00	0.0000D+00	0.3861D+03	-0.35125D+03	0.92855D+03	0.25223D+03	0.18467D+04	0.56365D+03	0.24104D+04
6	RUL	0.1244D+03	-0.7761D+01	0.5426D+02	-0.23939D+01	0.52884D+02	-0.10354D+01	9.36122D+04	0.16567D+02	0.36287D+04
7	RLL	0.0000D+00	0.0000D+00	0.3861D+03	-0.82176D+01	-0.24413D+03	-0.15598D+03	0.12134D+04	0.22031D+02	0.12355D+04
8	RF	0.0000D+00	0.0000D+00	0.3861D+03	-0.18022D+03	0.11120D+04	-0.75866D+03	0.23167D+03	0.27743D+00	0.23195D+03
9	LUL	-0.1221D+04	0.2472D+04	0.2645D+04	-0.70853D+03	-0.12076D+04	-0.31901D+03	0.30251D+04	0.11362D+02	0.30365D+04
10	LLL	0.0000D+00	0.0000D+00	0.3861D+03	-0.66486D+01	-0.17362D+03	-0.14022D+03	0.10259D+04	0.89964D+01	0.10349D+04
11	LF	0.0000D+00	0.0000D+00	0.3861D+03	-0.20969D+03	0.81201D+03	-0.72214D+02	0.18487D+03	0.69768D-01	0.18494D+03
12	RUA	0.0000D+00	0.0000D+00	0.3861D+03	-0.44483D+02	-0.70077D+04	0.79493D+03	0.14022D+04	0.49151D+01	0.14071D+04
13	RLA	-0.8603D+02	-0.4004D+01	0.7761D+03	0.31299D+02	0.34441D+04	-0.57371D+02	0.14312D+04	0.66593D+01	0.14379D+04
14	LUA	0.0000D+00	0.0000D+00	0.3861D+03	-0.16398D+03	-0.90273D+03	0.56251D+02	0.62271D+03	0.11478D+01	0.62386D+03
15	LLA	-0.9690D+04	-0.7116D+04	-0.3749D+04	0.20275D+04	0.70096D+04	0.12476D+04	0.94712D+03	0.40319D+02	0.98744D+03
							TOTAL BODY KINETIC ENERGY			
							0.36368D+05	0.15454D+04	0.37913D+05	

JOINT	IPIN	(INERTIAL)			(INERTIAL)			RELATIVE ANGULAR VELOCITY (RAD/SEC.)	
		JOINT FORCES (LB.)			JOINT TORQUES (IN. LB.)				
		X	Y	Z	X	Y	Z		
1	P	0	-0.478D+03	0.156D+03	-0.120D+04	0.1437D+04	0.1730D+04	-0.4588D+03	47.900
2	W	0	-0.373D+03	0.134D+03	-0.919D+03	-0.2162D+04	-0.6568D+03	0.7903D+03	63.460
3	NP	0	-0.296D+03	-0.144D+03	0.914D+01	-0.2178D+03	0.5526D+03	0.5465D+02	4.653
4	HP	0	-0.349D+03	-0.128D+03	0.851D+02	-0.8035D+02	0.2872D+03	0.4001D+02	48.641
5	RH	0	0.968D+02	-0.374D+02	-0.276D+03	0.1549D+01	0.6042D+02	0.1998D+02	11.110
6	RK	1	0.383D+02	0.731D+01	0.463D+02	-0.9820D+01	-0.7377D+02	0.4208D+02	13.094
7	RA	0	0.360D+00	0.125D+01	-0.398D+01	0.6057D+01	0.4563D+02	0.6082D+01	8.104
8	LH	0	0.316D+03	0.174D+01	-0.108D+03	-0.1049D+02	0.5690D+02	0.2195D+02	10.800
9	LK	1	0.990D+02	0.636D+02	0.123D+03	-0.1739D+03	-0.4959D+02	0.2365D+03	9.313
10	LA	0	0.958D+01	0.342D+00	0.117D+01	0.8697D+01	0.3333D+02	0.3159D+01	6.037
11	RS	0	0.123D+03	0.609D+02	0.770D+02	0.2329D+02	-0.3963D+02	0.2492D+02	9.125
12	RE	1	0.138D+03	0.234D+02	0.465D+02	-0.5987D+02	0.1090D+04	-0.4321D+03	5.353
13	LS	0	0.163D+03	0.378D+00	0.134D+03	-0.7896D+01	-0.5782D+02	0.3755D+02	12.112
14	LE	1	0.214D+03	-0.112D+02	-0.153D+02	-0.1860D+03	0.1440D+03	0.2989D+03	15.250

POSTPROCESSOR CONTROL PARAMETERS

N.11	HIC & HSI POINT 2	CSI POINT 1
------	----------------------	----------------

RUN DESCRIPTION: EXAMPLE 1: BASIC SLED TEST SIMULATION

PAGE: 21.01

VEHICLE DECELERATION: SLED ACCELERATION - 20G PEAK

CRASH VICTIM: 95TH PERCENTILE MALE

POINT TOTAL ACCELERATION (G'S)

TIME (MSEC)	POINT (0.00, 0.00, 0.00) ON SEGMENT NO. 3 - UT				POINT (0.00, 0.00, 0.00) ON SEGMENT NO. -5 - H				POINT (0.00, 0.00, 0.00) ON SEGMENT NO. 5 - H			
	IN UT REFERENCE				ACCELEROMETER (1G)				IN VEH REFERENCE			
	X	Y	Z	RES	X	Y	Z	RES	X	Y	Z	RES
0.000	-0.020	0.000	-0.082	0.084	-0.262	0.000	0.890	0.928	-0.047	0.000	-0.073	0.087
2.000	-0.014	0.000	-0.107	0.108	-0.260	0.000	0.865	0.903	-0.051	0.000	-0.098	0.111
4.000	-0.007	0.000	-0.179	0.179	-0.258	0.000	0.793	0.834	-0.067	0.000	-0.168	0.181
6.000	-0.006	-0.003	-0.247	0.247	-0.269	0.000	0.725	0.774	-0.093	0.000	-0.232	0.250
8.000	-0.019	-0.012	-0.314	0.315	-0.306	-0.002	0.658	0.725	-0.144	-0.002	-0.289	0.323
10.000	-0.061	-0.023	-0.386	0.391	-0.379	-0.005	0.586	0.698	-0.233	-0.005	-0.342	0.413
12.000	-0.091	-0.032	-0.464	0.474	-0.475	-0.008	0.507	0.695	-0.344	-0.008	-0.396	0.525
14.000	-0.099	-0.023	-0.547	0.556	-0.487	-0.006	0.425	0.646	-0.375	-0.006	-0.474	0.604
16.000	-0.132	-0.024	-0.623	0.638	-0.506	-0.007	0.348	0.614	-0.411	-0.007	-0.544	0.682
18.000	-0.111	-0.005	-0.719	0.727	-0.510	-0.003	0.253	0.570	-0.437	-0.003	-0.635	0.771
20.000	-0.133	-0.001	-0.791	0.802	-0.525	-0.002	0.181	0.555	-0.468	-0.002	-0.702	0.844
22.000	-0.197	-0.016	-0.854	0.876	-0.551	-0.005	0.119	0.564	-0.509	-0.006	-0.756	0.911
24.000	-0.382	-0.063	-1.317	1.373	-0.662	-0.179	-0.342	0.767	-0.724	-0.179	-1.179	1.395
26.000	-0.390	-0.112	-1.296	1.358	-0.685	-0.231	-0.314	0.788	-0.739	-0.231	-1.147	1.384
28.000	-0.357	0.556	-1.166	1.340	-0.682	-0.014	-0.181	0.705	-0.705	-0.014	-1.019	1.239
30.000	-0.438	0.816	-1.246	1.553	-0.702	0.035	-0.256	0.748	-0.742	0.035	-1.087	1.317
32.000	-2.733	-0.835	-0.461	2.895	-1.271	-0.218	0.576	1.412	-1.106	-0.220	-0.149	1.137
34.000	-2.186	-0.420	-0.403	2.263	-1.208	-0.216	0.667	1.397	-1.025	-0.219	-0.076	1.051
36.000	-3.145	-0.474	-0.313	3.196	-1.481	-0.353	0.849	1.743	-1.252	-0.357	0.159	1.312
38.000	-4.178	-0.580	-1.221	4.391	-1.843	-0.709	0.133	1.979	-1.769	-0.710	-0.467	1.963
40.000	-5.137	0.364	-1.250	5.299	-2.004	-0.530	0.280	2.092	-1.897	-0.533	-0.297	1.993
42.000	-6.838	1.397	-2.254	7.334	-2.388	-0.543	-0.390	2.479	-2.417	-0.540	-0.885	2.630
44.000	-14.419	-3.743	0.830	14.920	-4.097	-1.242	3.893	5.787	-3.227	-1.300	3.624	5.024
46.000	-16.621	-1.760	-0.518	16.722	-4.375	-1.075	3.474	5.689	-3.633	-1.143	3.226	4.991
48.000	-21.535	0.772	0.632	21.558	-4.411	-0.608	5.515	7.088	-3.372	-0.744	5.190	6.234
50.000	-41.704	-14.755	6.467	44.707	-8.393	-3.295	18.145	20.262	-5.470	-3.858	18.124	19.321
52.000	-46.097	-12.754	8.809	48.633	-8.850	-2.015	24.198	25.844	-5.684	-2.996	24.033	24.877
54.000	-59.506	-18.177	14.142	63.807	-11.972	-2.812	40.775	42.589	-8.224	-4.949	40.493	41.615
56.000	-60.433	-15.422	13.472	63.808	-11.763	-2.643	48.226	49.710	-9.706	-5.876	47.398	48.737
58.000	-73.471	-17.978	-2.410	75.677	-11.839	-4.818	45.094	46.871	-12.593	-8.610	43.319	45.927
60.000	-93.922	-15.685	14.909	96.382	-10.687	1.243	64.829	65.716	-16.338	-5.515	62.414	64.752
62.000	-39.398	-7.543	-3.263	40.247	-8.458	0.385	50.073	50.783	-16.781	-5.749	46.585	49.848
64.000	-13.416	-3.242	-10.799	17.525	-4.985	0.413	34.219	34.583	-13.561	-4.374	30.511	33.674
66.000	2.837	-2.587	-20.041	20.406	-4.396	-0.971	17.138	17.719	-9.978	-3.581	13.198	16.929
68.000	12.171	-3.027	-29.353	31.921	-6.574	-2.288	10.559	12.647	-10.367	-3.898	5.106	12.195
70.000	12.817	-4.431	-27.755	30.891	-6.991	-2.721	11.812	13.993	-11.904	-4.656	4.693	13.617
72.000	8.827	-2.877	-26.335	27.924	-5.142	-2.316	12.679	13.876	-11.519	-4.665	5.172	13.461
74.000	3.362	-1.720	-32.445	32.664	-5.688	-2.458	13.892	15.211	-13.307	-5.198	4.223	14.897
76.000	-2.984	1.218	-32.702	32.861	-6.355	-2.850	21.202	22.317	-19.656	-7.488	6.458	22.003
78.000	-4.104	3.340	-30.967	31.417	-5.758	-2.791	26.452	27.215	-24.295	-9.144	7.173	26.932
80.000	-0.706	4.868	-21.526	22.081	-3.903	-2.154	32.157	32.464	-29.256	-10.704	8.134	32.197

RUN DESCRIPTION: EXAMPLE 1: BASIC SLED TEST SIMULATION
TWO BELT HARNESS WITH HYPERELLIPTOID FOR DASH BOARD

VEHICLE DECELERATION: SLED ACCELERATION - 20G PEAK
CRASH VICTIM: 95TH PERCENTILE MALE

POINT REL. VELOCITY (IN./SEC.)

TIME (MSEC)	POINT (0.00, 0.00, 0.00) ON SEGMENT NO. 3 - UT				POINT (0.00, 0.00, 0.00) ON SEGMENT NO. 5 - H				POINT (0.00, 0.00, 0.00) ON SEGMENT NO. 5 - H			
	IN VEH		REFERENCE	RES	IN VEH		REFERENCE	RES	IN VEH		REFERENCE	RES
	X	Y	Z		X	Y	Z		X	Y	Z	
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
2.000	0.357	0.000	-0.064	0.362	0.349	0.000	-0.062	0.355	-0.008	0.000	0.000	0.008
4.000	1.482	0.000	-0.168	1.491	1.463	0.000	-0.163	1.472	-0.020	0.000	0.000	0.020
6.000	3.368	-0.002	-0.329	3.384	3.332	0.000	-0.320	3.347	-0.037	0.002	0.000	0.037
8.000	6.010	-0.010	-0.534	6.034	5.944	-0.002	-0.519	5.967	-0.067	0.008	0.000	0.068
10.000	9.392	-0.024	-0.792	9.425	9.275	-0.005	-0.764	9.307	-0.120	0.019	0.000	0.121
12.000	13.502	-0.048	-1.096	13.546	13.296	-0.010	-1.048	13.338	-0.211	0.038	0.000	0.214
14.000	18.360	-0.073	-1.462	18.418	18.037	-0.016	-1.386	18.090	-0.332	0.057	0.000	0.336
16.000	23.966	-0.096	-1.885	24.040	23.526	-0.022	-1.782	23.593	-0.452	0.074	-0.001	0.458
18.000	30.318	-0.118	-2.369	30.411	29.761	-0.028	-2.239	29.845	-0.572	0.090	-0.001	0.579
20.000	37.426	-0.138	-2.913	37.539	36.747	-0.034	-2.754	36.851	-0.697	0.104	-0.001	0.704
22.000	45.254	-0.173	-3.495	45.389	44.477	-0.043	-3.312	44.601	-0.798	0.130	-0.001	0.808
24.000	53.624	-0.222	-4.417	53.806	52.814	-0.157	-4.225	52.983	-0.832	0.065	-0.001	0.835
26.000	62.779	-0.294	-5.297	63.002	61.919	-0.307	-5.089	62.128	-0.885	-0.012	0.003	0.885
28.000	72.709	-0.176	-6.137	72.968	71.793	-0.419	-5.907	72.037	-0.945	-0.242	0.010	0.975
30.000	83.404	0.352	-6.915	83.690	82.451	-0.398	-6.666	82.721	-0.986	-0.749	0.019	1.238
32.000	94.080	0.535	-7.240	94.360	93.727	-0.423	-7.110	93.997	-0.376	-0.957	0.041	1.029
34.000	104.889	0.071	-7.234	105.138	105.604	-0.633	-7.306	105.859	0.710	-0.706	0.097	1.006
36.000	116.510	-0.408	-7.277	116.737	118.233	-0.984	-7.495	118.474	1.722	-0.585	0.197	1.830
38.000	127.722	-1.024	-7.102	127.923	131.306	-1.474	-7.573	131.533	3.590	-0.474	0.404	3.643
40.000	139.233	-1.201	-6.715	139.400	145.087	-1.867	-7.440	145.290	5.849	-0.719	0.720	5.937
42.000	149.450	-0.660	-6.479	149.592	158.571	-2.237	-7.528	158.765	9.079	-1.685	1.225	9.315
44.000	155.812	-0.836	-4.740	155.887	170.774	-2.712	-6.337	170.913	14.855	-2.106	2.198	15.163
46.000	157.720	-3.088	-1.115	157.754	181.675	-3.707	-3.283	181.743	23.690	-1.112	4.061	24.061
48.000	156.096	-4.584	3.913	156.213	191.901	-4.378	0.957	191.953	35.308	-0.762	6.593	35.927
50.000	141.126	-13.942	16.832	142.808	199.739	-6.638	13.452	200.301	57.525	5.185	12.814	59.163
52.000	122.195	-24.412	32.266	128.719	206.670	-9.011	29.910	209.017	82.311	11.324	21.803	85.900
54.000	94.373	-38.752	54.224	115.535	211.079	-12.280	56.971	218.976	112.157	18.960	37.279	119.701
56.000	63.940	-52.830	78.311	114.069	213.707	-16.371	91.124	232.899	141.171	23.677	58.599	154.673
58.000	27.937	-68.681	99.258	123.894	214.085	-21.715	126.806	249.767	171.680	26.531	86.244	193.948
60.000	-48.785	-89.387	132.584	167.178	209.556	-25.937	169.668	270.875	238.362	28.911	120.368	268.591
62.000	-77.547	-101.219	145.352	193.355	203.606	-30.570	209.855	293.988	252.303	25.298	154.604	296.984
64.000	-88.016	-107.303	145.939	201.393	199.040	-34.244	238.403	312.451	249.362	19.440	183.655	310.304
66.000	-87.056	-109.586	135.436	194.758	196.297	-37.048	254.173	323.279	237.429	13.587	207.586	315.673
68.000	-81.208	-110.824	115.623	179.570	193.532	-39.896	261.153	327.486	220.628	9.256	230.063	318.891
70.000	-73.848	-112.242	90.482	161.984	189.212	-43.176	264.497	328.061	200.518	6.746	252.981	322.881
72.000	-68.106	-113.520	68.202	148.919	183.289	-46.872	268.296	328.290	181.488	4.619	273.349	328.145
74.000	-67.903	-114.780	45.146	140.795	177.090	-50.470	271.958	328.435	167.908	1.761	295.644	340.002
76.000	-74.122	-116.052	20.340	139.197	166.505	-55.141	276.104	327.105	156.313	-2.407	320.292	356.408
78.000	-83.506	-116.413	-2.649	143.291	150.797	-61.731	281.259	325.049	143.815	-8.988	343.115	372.145
80.000	-90.414	-115.111	-20.631	147.820	130.393	-69.834	287.803	323.589	126.450	-16.916	360.086	382.017

DATE: 2 SEPT 1988

RUN DESCRIPTION: EXAMPLE 1: BASIC SLED TEST SIMULATION
TWO BELT HARNESS WITH HYPERELLIPOID FOR DASH BOARD
VEHICLE DECELERATION: SLED ACCELERATION - 20G PEAK
CRASH VICTIM: 95TH PERCENTILE MALE

PAGE: 23.01

POINT REL. LINEAR DISPLACEMENT (IN.)

TIME (MSEC)	POINT (0.00, 0.00, 0.00) ON SEGMENT NO. 3 - UT				POINT (0.00, 0.00, 0.00) ON SEGMENT NO. 5 - H				POINT (0.00, 0.00, 0.00) ON SEGMENT NO. 5 - H			
	IN VEH		REFERENCE		IN VEH		REFERENCE		IN UT		REFERENCE	
	X	Y	Z	RES	X	Y	Z	RES	X	Y	Z	RES
0.000	11.314	0.000	-26.938	29.217	7.775	0.000	-41.833	42.550	-0.022	0.000	-15.310	15.310
2.000	11.314	0.000	-26.938	29.218	7.775	0.000	-41.833	42.550	-0.022	0.000	-15.310	15.310
4.000	11.316	0.000	-26.938	29.218	7.777	0.000	-41.834	42.550	-0.022	0.000	-15.310	15.310
6.000	11.320	0.000	-26.939	29.221	7.782	0.000	-41.834	42.552	-0.023	0.000	-15.310	15.310
8.000	11.330	0.000	-26.940	29.225	7.791	0.000	-41.835	42.554	-0.023	0.000	-15.310	15.310
10.000	11.345	0.000	-26.941	29.232	7.806	0.000	-41.836	42.558	-0.022	0.000	-15.310	15.310
12.000	11.368	0.000	-26.943	29.243	7.828	0.000	-41.838	42.564	-0.022	0.000	-15.310	15.310
14.000	11.400	0.000	-26.945	29.257	7.860	0.000	-41.840	42.572	-0.021	0.000	-15.310	15.310
16.000	11.442	0.000	-26.949	29.277	7.901	0.000	-41.844	42.583	-0.020	0.001	-15.310	15.310
18.000	11.496	-0.001	-26.953	29.302	7.954	0.000	-41.848	42.597	-0.017	0.001	-15.310	15.310
20.000	11.564	-0.001	-26.958	29.334	8.021	0.000	-41.853	42.614	-0.015	0.001	-15.310	15.310
22.000	11.646	-0.001	-26.965	29.372	8.102	0.000	-41.859	42.636	-0.011	0.002	-15.310	15.310
24.000	11.745	-0.002	-26.972	29.419	8.199	0.000	-41.866	42.661	-0.005	0.004	-15.310	15.310
26.000	11.861	-0.002	-26.982	29.474	8.314	-0.001	-41.876	42.693	0.003	0.008	-15.310	15.310
28.000	11.996	-0.003	-26.994	29.539	8.447	-0.002	-41.887	42.730	0.014	0.016	-15.310	15.310
30.000	12.152	-0.003	-27.007	29.615	8.601	-0.003	-41.899	42.773	0.027	0.024	-15.310	15.310
32.000	12.330	-0.001	-27.021	29.701	8.777	-0.003	-41.913	42.822	0.044	0.032	-15.310	15.310
34.000	12.529	-0.001	-27.036	29.798	8.977	-0.004	-41.928	42.878	0.069	0.042	-15.310	15.310
36.000	12.750	-0.001	-27.050	29.905	9.200	-0.006	-41.943	42.940	0.102	0.058	-15.309	15.309
38.000	12.995	-0.003	-27.065	30.023	9.450	-0.008	-41.958	43.009	0.147	0.081	-15.308	15.309
40.000	13.262	-0.005	-27.079	30.152	9.726	-0.012	-41.973	43.085	0.206	0.112	-15.306	15.308
42.000	13.551	-0.007	-27.092	30.292	10.030	-0.016	-41.988	43.169	0.279	0.149	-15.303	15.306
44.000	13.857	-0.007	-27.104	30.441	10.360	-0.021	-42.002	43.261	0.371	0.191	-15.298	15.303
46.000	14.171	-0.011	-27.110	30.590	10.712	-0.027	-42.012	43.356	0.499	0.245	-15.288	15.298
48.000	14.486	-0.019	-27.107	30.735	11.086	-0.035	-42.014	43.452	0.659	0.307	-15.273	15.290
50.000	14.784	-0.037	-27.087	30.859	11.478	-0.046	-42.001	43.541	0.868	0.380	-15.247	15.276
52.000	15.048	-0.075	-27.039	30.944	11.884	-0.062	-41.958	43.609	1.144	0.470	-15.201	15.252
54.000	15.266	-0.138	-26.953	30.976	12.302	-0.083	-41.873	43.643	1.497	0.561	-15.128	15.212
56.000	15.424	-0.230	-26.821	30.940	12.727	-0.111	-41.726	43.624	1.939	0.645	-15.010	15.148
58.000	15.520	-0.349	-26.641	30.834	13.156	-0.149	-41.507	43.543	2.449	0.714	-14.836	15.054
60.000	15.493	-0.507	-26.412	30.624	13.579	-0.197	-41.214	43.394	2.991	0.751	-14.606	14.928
62.000	15.361	-0.699	-26.130	30.319	13.993	-0.253	-40.832	43.164	3.548	0.697	-14.323	14.772
64.000	15.193	-0.909	-25.837	29.987	14.395	-0.318	-40.382	42.872	4.108	0.585	-13.976	14.579
66.000	15.017	-1.126	-25.554	29.661	14.790	-0.390	-39.887	42.543	4.633	0.439	-13.579	14.354
68.000	14.848	-1.347	-25.302	29.368	15.180	-0.466	-39.371	42.199	5.107	0.285	-13.141	14.101
70.000	14.693	-1.569	-25.096	29.123	15.563	-0.549	-38.845	41.851	5.528	0.131	-12.660	13.815
72.000	14.552	-1.795	-24.938	28.929	15.936	-0.639	-38.313	41.500	5.887	-0.018	-12.144	13.496
74.000	14.417	-2.024	-24.824	28.778	16.297	-0.737	-37.772	41.145	6.179	-0.156	-11.604	13.147
76.000	14.275	-2.255	-24.759	28.668	16.641	-0.842	-37.225	40.784	6.409	-0.281	-11.038	12.767
78.000	14.118	-2.487	-24.742	28.595	16.959	-0.959	-36.667	40.411	6.577	-0.392	-10.451	12.355
80.000	13.944	-2.719	-24.766	28.551	17.242	-1.090	-36.098	40.019	6.684	-0.491	-9.851	11.915

DATE: 2 SEPT 1988
 RUM DESCRIPTION: EXAMPLE 1: BASIC SLED TEST SIMULATION
 TWO BELT HARNESS WITH HYPERELLIPSOID FOR DASH BOARD
 VEHICLE DECELERATION: SLED ACCELERATION - 20G PEAK
 CRASH VICTIM: 95TH PERCENTILE MALE

SEGMENT ANGULAR ACCELERATION (REV/SEC.**2)

TIME (MSEC)	SEGMENT NO. 3 - UT				SEGMENT NO. 5 - H				SEGMENT NO. 5 - H			
	IN	UT	REFERENCE	RES	IN	H	REFERENCE	RES	IN	VEH	REFERENCE	RES
	X	Y	Z		X	Y	Z		X	Y	Z	
0.000	0.000	-0.137	0.000	0.137	0.000	-0.185	0.000	0.185	0.000	-0.185	0.000	0.185
2.000	0.001	-0.042	0.002	0.043	-0.001	-0.052	0.000	0.052	-0.001	-0.052	0.000	0.052
4.000	0.006	0.040	0.015	0.043	-0.006	-0.067	0.000	0.067	-0.006	-0.067	0.000	0.067
6.000	-0.004	0.247	0.049	0.252	0.021	-0.177	-0.001	0.178	0.020	-0.177	-0.005	0.178
8.000	-0.039	0.665	0.129	0.678	0.107	-0.365	-0.002	0.380	0.104	-0.365	-0.027	0.380
10.000	-0.080	1.188	0.277	1.223	0.221	-0.482	0.002	0.530	0.215	-0.482	-0.050	0.530
12.000	-0.118	1.611	0.416	1.669	0.332	-0.264	0.016	0.424	0.326	-0.264	-0.062	0.424
14.000	-0.083	1.943	0.457	1.997	0.283	-0.730	0.029	0.784	0.282	-0.730	-0.038	0.784
16.000	-0.087	2.460	0.635	2.542	0.302	-1.430	0.043	1.462	0.303	-1.430	-0.029	1.462
18.000	0.008	2.589	0.664	2.673	0.141	-1.610	0.060	1.617	0.151	-1.610	0.026	1.617
20.000	0.037	2.973	0.948	3.120	0.102	-2.115	0.080	2.119	0.118	-2.115	0.054	2.119
22.000	-0.010	3.607	1.429	3.879	0.244	-3.094	0.108	3.106	0.262	-3.094	0.048	3.106
24.000	-8.600	7.157	4.770	12.171	7.768	-7.307	0.183	10.666	7.598	-7.307	-1.626	10.666
26.000	-9.914	7.450	4.950	13.353	10.014	-8.144	0.309	12.911	9.812	-8.143	-2.022	12.911
28.000	-1.602	6.837	5.335	8.819	0.731	-7.928	0.450	7.974	0.814	-7.928	0.260	7.974
30.000	-2.418	7.045	5.092	9.022	-1.971	-8.546	0.594	8.791	-1.785	-8.548	1.017	8.791
32.000	-5.518	18.566	11.214	22.381	8.491	-29.792	0.788	30.988	8.434	-29.793	-1.243	30.988
34.000	-5.789	14.342	10.997	18.977	8.985	-28.116	1.107	29.537	8.988	-28.117	-1.045	29.537
36.000	-10.236	20.115	12.679	25.887	15.101	-38.011	1.455	40.927	15.018	-38.013	-2.127	40.927
38.000	-27.100	26.891	17.134	41.846	30.178	-51.830	1.903	60.006	29.806	-51.828	-5.111	60.006
40.000	-22.699	24.663	16.092	37.182	22.068	-57.722	2.414	61.843	21.998	-57.727	-2.878	61.843
42.000	-30.498	28.252	24.219	48.113	21.827	-71.925	2.986	75.223	21.854	-71.937	-2.442	75.223
44.000	-32.187	56.105	49.690	81.565	50.733	-136.186	3.903	145.381	50.213	-136.179	-8.335	145.381
46.000	-28.042	37.572	48.859	67.714	44.955	-148.639	5.635	155.391	44.854	-148.665	-5.761	155.391
48.000	-20.702	-4.036	40.577	45.731	24.155	-149.938	7.474	152.055	24.678	-150.038	-0.423	152.055
50.000	-48.385	37.032	81.432	101.704	139.703	-302.280	11.206	333.190	138.788	-302.255	-19.874	333.190
52.000	20.899	8.412	77.489	80.698	86.870	-320.679	15.481	332.598	87.017	-320.899	-8.573	332.598
54.000	21.075	54.615	169.909	179.711	120.265	-438.481	21.646	455.190	120.241	-438.852	-12.223	455.190
56.000	15.799	22.276	164.884	167.130	113.182	-428.843	29.299	444.494	113.279	-429.795	-4.359	444.494
58.000	-70.850	-73.615	-213.885	237.035	205.076	-427.000	36.749	475.117	204.444	-428.864	3.808	475.117
60.000	213.357	-282.895	320.674	477.894	-54.947	-386.330	39.469	392.208	-56.394	-388.083	-6.237	392.208
62.000	88.426	-23.362	27.240	95.430	-20.082	-300.005	42.593	303.679	-23.448	-302.770	1.119	303.679
64.000	44.445	-34.608	-63.285	84.723	-20.168	-163.484	40.696	169.676	-26.390	-167.272	10.665	169.676
66.000	-14.123	-63.981	-141.182	155.645	35.355	-125.898	37.342	135.996	24.564	-131.014	26.957	135.996
68.000	-37.438	-24.558	-71.335	84.222	82.975	-179.967	36.180	201.450	67.795	-186.059	36.984	201.450
70.000	-38.031	-65.178	-38.947	84.920	90.968	-166.521	37.777	193.473	68.935	-174.339	47.809	193.473
72.000	-58.153	-115.779	-40.978	135.889	65.468	-76.947	39.363	108.427	34.892	-86.935	54.600	108.427
74.000	-73.956	-128.701	-33.634	152.200	71.790	-104.065	35.913	131.427	39.042	-114.114	52.218	131.427
76.000	-108.913	-153.734	-12.238	188.802	85.877	-134.273	35.927	163.386	46.823	-145.589	57.501	163.386
78.000	-133.289	-152.796	3.154	202.787	80.966	-120.334	37.667	149.849	34.375	-133.462	58.829	149.849
80.000	-132.606	-117.860	3.936	177.457	51.234	-55.968	40.143	85.842	-0.345	-70.727	48.645	85.842

RUN DESCRIPTION: EXAMPLE 1: BASIC SLED TEST SIMULATION
TWO BELT HARNESS WITH HYPERELLIPTOID FOR DASH BOARD

PAGE: 25.01

VEHICLE DECELERATION: SLED ACCELERATION - 20G PEAK
CRASH VICTIM: 95TH PERCENTILE MALE

SEGMENT REL. ANGULAR VELOCITY (REV/SEC.)

TIME (MSEC)	SEGMENT NO. 3 - UT				SEGMENT NO. 5 - H				SEGMENT NO. 5 - H			
	IN VEH REFERENCE				IN VEH REFERENCE				IN H REFERENCE			
	X	Y	Z	RES	X	Y	Z	RES	X	Y	Z	RES
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
2.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
4.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
6.000	0.000	0.000	0.000	0.000	0.000	-0.001	0.000	0.001	0.000	-0.001	0.000	0.001
8.000	0.000	0.001	0.000	0.001	0.000	-0.001	0.000	0.001	0.000	-0.001	0.000	0.001
10.000	0.000	0.003	0.001	0.003	0.001	-0.002	0.000	0.002	0.001	-0.002	0.000	0.002
12.000	0.000	0.006	0.002	0.006	0.001	-0.003	0.000	0.003	0.001	-0.003	0.000	0.003
14.000	0.000	0.009	0.003	0.010	0.002	-0.004	0.000	0.004	0.002	-0.004	0.000	0.004
16.000	0.000	0.014	0.004	0.014	0.002	-0.006	0.000	0.006	0.002	-0.006	0.000	0.006
18.000	0.000	0.019	0.005	0.019	0.003	-0.009	0.000	0.009	0.003	-0.009	0.000	0.009
20.000	0.000	0.024	0.006	0.025	0.004	-0.013	0.000	0.013	0.004	-0.013	0.000	0.013
22.000	0.001	0.031	0.009	0.032	0.005	-0.018	-0.001	0.019	0.005	-0.018	0.001	0.019
24.000	-0.012	0.044	0.021	0.050	0.016	-0.031	-0.003	0.035	0.017	-0.031	0.001	0.035
26.000	-0.026	0.058	0.033	0.072	0.033	-0.046	-0.007	0.057	0.034	-0.046	0.001	0.057
28.000	-0.036	0.072	0.046	0.093	0.046	-0.062	-0.009	0.077	0.047	-0.062	0.002	0.077
30.000	-0.036	0.085	0.055	0.108	0.044	-0.078	-0.007	0.089	0.044	-0.078	0.003	0.089
32.000	-0.037	0.112	0.075	0.139	0.045	-0.114	-0.006	0.123	0.045	-0.114	0.004	0.123
34.000	-0.049	0.146	0.102	0.185	0.067	-0.174	-0.010	0.187	0.067	-0.174	0.006	0.187
36.000	-0.076	0.180	0.132	0.236	0.105	-0.236	-0.016	0.258	0.106	-0.236	0.009	0.258
38.000	-0.114	0.231	0.174	0.311	0.158	-0.331	-0.025	0.367	0.159	-0.331	0.012	0.367
40.000	-0.139	0.277	0.212	0.376	0.200	-0.436	-0.031	0.481	0.202	-0.436	0.017	0.481
42.000	-0.172	0.326	0.260	0.452	0.239	-0.565	-0.035	0.614	0.240	-0.565	0.022	0.614
44.000	-0.196	0.409	0.348	0.571	0.285	-0.766	-0.041	0.818	0.287	-0.766	0.029	0.818
46.000	-0.235	0.504	0.469	0.728	0.385	-1.056	-0.056	1.125	0.388	-1.056	0.038	1.125
48.000	-0.229	0.530	0.567	0.809	0.447	-1.355	-0.060	1.428	0.449	-1.355	0.051	1.428
50.000	-0.241	0.669	0.811	1.079	0.652	-1.933	-0.092	2.042	0.655	-1.933	0.069	2.042
52.000	-0.192	0.728	0.993	1.246	0.850	-2.558	-0.116	2.698	0.854	-2.558	0.096	2.698
54.000	-0.059	0.902	1.343	1.618	1.071	-3.411	-0.144	3.578	1.075	-3.411	0.132	3.578
56.000	0.073	1.021	1.706	1.989	1.296	-4.299	-0.161	4.493	1.300	-4.297	0.183	4.493
58.000	0.050	0.985	1.756	2.014	1.581	-5.149	-0.162	5.389	1.586	-5.144	0.249	5.389
60.000	0.638	0.350	2.155	2.274	1.585	-6.117	-0.179	6.321	1.591	-6.109	0.325	6.321
62.000	0.889	0.383	2.284	2.480	1.553	-6.864	-0.186	7.040	1.563	-6.852	0.408	7.040
64.000	0.999	0.342	2.217	2.456	1.491	-7.290	-0.171	7.443	1.512	-7.271	0.492	7.443
66.000	0.984	0.240	2.039	2.276	1.477	-7.552	-0.134	7.696	1.514	-7.524	0.570	7.696
68.000	0.901	0.147	1.885	2.095	1.563	-7.860	-0.072	8.014	1.626	-7.821	0.642	8.014
70.000	0.823	0.060	1.817	1.996	1.708	-8.236	0.012	8.412	1.807	-8.184	0.716	8.412
72.000	0.766	-0.148	1.772	1.936	1.824	-8.529	0.116	8.722	1.975	-8.458	0.794	8.722
74.000	0.692	-0.414	1.729	1.908	1.879	-8.677	0.222	8.881	2.095	-8.586	0.868	8.881
76.000	0.606	-0.733	1.717	1.963	1.971	-8.960	0.328	9.180	2.256	-8.849	0.939	9.180
78.000	0.491	-1.104	1.763	2.138	2.058	-9.256	0.445	9.492	2.428	-9.120	1.013	9.492
80.000	0.353	-1.429	1.829	2.347	2.089	-9.448	0.553	9.692	2.557	-9.285	1.090	9.692

DATE: 2 SEPT 1988
 RUN DESCRIPTION: EXAMPLE 1: BASIC SLED TEST SIMULATION
 TWO BELT HARNESS WITH HYPERELLIPSOID FOR DASH BOARD
 VEHICLE DECELERATION: SLED ACCELERATION - 20G PEAK
 CRASH VICTIM: 95TH PERCENTILE MALE

SEGMENT REL. ANGULAR DISPLACEMENT (DEG)

TIME (MSEC)	SEGMENT NO. 3 - UT				SEGMENT NO. 5 - H				SEGMENT NO. 5 - H			
	YAW	PITCH	ROLL	RES	YAW	PITCH	ROLL	RES	YAW	PITCH	ROLL	RES
0.000	0.000	13.280	0.000	13.280	0.000	13.460	0.000	13.460	0.000	0.180	0.000	0.180
2.000	0.000	13.280	0.000	13.280	0.000	13.460	0.000	13.460	0.000	0.180	0.000	0.180
4.000	0.000	13.280	0.000	13.280	0.000	13.460	0.000	13.460	0.000	0.180	0.000	0.180
6.000	0.000	13.280	0.000	13.280	0.000	13.459	0.000	13.459	0.000	0.180	0.000	0.180
8.000	0.000	13.280	0.000	13.280	0.000	13.459	0.000	13.459	0.000	0.179	0.000	0.179
10.000	0.001	13.281	0.000	13.281	0.000	13.458	0.000	13.458	-0.001	0.176	0.000	0.176
12.000	0.001	13.284	0.000	13.284	0.000	13.456	0.001	13.456	-0.001	0.171	0.001	0.172
14.000	0.003	13.290	0.000	13.290	0.000	13.454	0.002	13.454	-0.003	0.164	0.003	0.164
16.000	0.005	13.298	0.000	13.298	0.000	13.450	0.003	13.450	-0.005	0.152	0.005	0.152
18.000	0.008	13.310	0.000	13.310	0.000	13.445	0.005	13.445	-0.008	0.135	0.007	0.136
20.000	0.012	13.325	0.000	13.325	0.000	13.437	0.008	13.437	-0.011	0.112	0.010	0.113
22.000	0.018	13.345	0.001	13.345	0.001	13.426	0.011	13.426	-0.016	0.081	0.014	0.084
24.000	0.027	13.372	-0.003	13.372	0.001	13.409	0.018	13.409	-0.025	0.037	0.027	0.052
26.000	0.043	13.409	-0.017	13.409	0.002	13.381	0.036	13.382	-0.040	-0.027	0.063	0.079
28.000	0.067	13.456	-0.042	13.456	0.003	13.343	0.067	13.343	-0.062	-0.113	0.124	0.179
30.000	0.097	13.513	-0.069	13.513	0.005	13.293	0.101	13.293	-0.089	-0.220	0.191	0.305
32.000	0.136	13.582	-0.095	13.583	0.008	13.227	0.132	13.228	-0.126	-0.354	0.257	0.455
34.000	0.193	13.675	-0.127	13.677	0.011	13.123	0.173	13.124	-0.178	-0.552	0.343	0.673
36.000	0.265	13.791	-0.175	13.795	0.017	12.978	0.237	12.980	-0.245	-0.813	0.471	0.970
38.000	0.359	13.939	-0.243	13.946	0.023	12.775	0.333	12.779	-0.332	-1.162	0.657	1.374
40.000	0.476	14.123	-0.336	14.136	0.032	12.500	0.466	12.508	-0.443	-1.620	0.910	1.907
42.000	0.617	14.341	-0.449	14.364	0.042	12.140	0.628	12.156	-0.579	-2.196	1.220	2.572
44.000	0.797	14.601	-0.582	14.638	0.055	11.672	0.813	11.700	-0.757	-2.920	1.583	3.398
46.000	1.052	14.935	-0.739	14.997	0.068	11.017	1.061	11.067	-1.018	-3.903	2.055	4.511
48.000	1.383	15.315	-0.905	15.414	0.082	10.149	1.367	10.240	-1.367	-5.141	2.620	5.902
50.000	1.837	15.754	-1.063	15.912	0.093	8.975	1.755	9.144	-1.860	-6.738	3.301	7.683
52.000	2.445	16.263	-1.213	16.515	0.095	7.361	2.304	7.712	-2.540	-8.837	4.197	10.024
54.000	3.262	16.853	-1.279	17.247	0.077	5.221	2.994	6.017	-3.499	-11.531	5.240	12.994
56.000	4.376	17.548	-1.225	18.167	0.024	2.443	3.846	4.556	-4.830	-14.954	6.468	16.740
58.000	5.730	18.282	-1.076	19.232	-0.080	-0.956	4.867	4.960	-6.520	-19.021	7.925	21.188
60.000	7.198	18.725	-0.796	20.112	-0.263	-5.032	6.043	7.858	-8.457	-23.473	9.510	26.030
62.000	9.010	18.894	-0.167	20.927	-0.545	-9.713	7.216	12.079	-10.799	-28.287	10.903	31.183
64.000	10.887	19.035	0.596	21.863	-0.927	-14.811	8.405	16.984	-13.329	-33.526	12.297	36.700
66.000	12.684	19.097	1.389	22.806	-1.407	-20.134	9.632	22.235	-15.925	-38.944	13.770	42.339
68.000	14.326	19.070	2.120	23.692	-2.012	-25.641	10.994	27.755	-18.620	-44.470	15.523	48.056
70.000	15.868	18.985	2.772	24.554	-2.808	-31.383	12.610	33.588	-21.668	-50.163	17.827	53.960
72.000	17.346	18.800	3.344	25.369	-3.876	-37.343	14.579	39.712	-25.358	-55.941	20.942	60.003
74.000	18.753	18.447	3.806	26.080	-5.305	-43.404	16.970	46.004	-30.054	-61.584	25.198	66.009
76.000	20.091	17.906	4.127	26.679	-7.292	-49.559	19.979	52.473	-36.534	-66.989	31.360	71.966
78.000	21.384	17.148	4.271	27.175	-10.213	-55.825	23.994	59.178	-46.425	-71.980	41.030	77.911
80.000	22.660	16.188	4.207	27.614	-14.704	-62.089	29.648	66.073	-62.537	-76.081	56.973	83.815

DATE: 2 SEPT 1988
 RUN DESCRIPTION: EXAMPLE 1: BASIC SLED TEST SIMULATION
 TWO BELT HARNESS WITH HYPERELLIPSOID FOR DASH BOARD
 VEHICLE DECELERATION: SLED ACCELERATION - 20G PEAK
 CRASH VICTIM: 95TH PERCENTILE MALE

PAGE: 27.01

JOINT PARAMETERS

TIME (MSEC)	STATE	JOINT NO. 3 - NP				TOTAL TORQUE (IN. LB.)			STATE	JOINT NO. 4 - HP				TOTAL TORQUE (IN. LB.)		
		IPIN	FLEXURE	AZIMUTH	TORSION	SPRING	VISCOUS	RES.		IPIN	FLEXURE	AZIMUTH	TORSION	SPRING	VISCOUS	RES.
0.000	0.	10.180	0.000	0.000	0.000	0.000	0.000	0.	10.000	0.000	0.000	0.000	0.000	0.000	0.000	
2.000	0.	10.181	0.000	0.000	0.000	0.118	0.118	0.	9.999	0.000	0.000	0.000	0.118	0.118	0.118	
4.000	0.	10.184	0.000	0.000	0.000	0.190	0.190	0.	9.996	0.000	0.000	0.000	0.194	0.194	0.194	
6.000	0.	10.189	0.000	0.000	0.000	0.241	0.241	0.	9.991	0.000	0.000	0.000	0.263	0.263	0.263	
8.000	0.	10.194	-0.001	0.000	0.000	0.224	0.224	0.	9.985	0.001	0.000	0.000	0.303	0.303	0.303	
10.000	0.	10.197	-0.005	0.000	0.000	0.115	0.115	0.	9.979	0.002	0.000	0.000	0.284	0.284	0.284	
12.000	0.	10.197	-0.013	-0.001	0.000	0.166	0.166	0.	9.974	0.005	0.000	0.000	0.187	0.187	0.187	
14.000	0.	10.192	-0.026	-0.002	0.000	0.401	0.401	0.	9.971	0.009	-0.001	0.000	0.112	0.112	0.112	
16.000	0.	10.182	-0.042	-0.003	0.000	0.680	0.680	0.	9.970	0.012	-0.002	0.000	0.069	0.069	0.069	
18.000	0.	10.166	-0.061	-0.004	0.000	0.977	0.977	0.	9.969	0.013	-0.003	0.000	0.072	0.072	0.072	
20.000	0.	10.144	-0.081	-0.006	0.000	1.281	1.281	0.	9.968	0.012	-0.004	0.000	0.104	0.104	0.104	
22.000	0.	10.115	-0.104	-0.009	0.000	1.728	1.728	0.	9.967	0.009	-0.006	0.000	0.122	0.122	0.122	
24.000	0.	10.068	-0.222	-0.013	0.000	3.586	3.586	0.	9.970	0.046	-0.009	0.000	0.771	0.771	0.771	
26.000	0.	9.997	-0.530	-0.019	0.000	5.256	5.256	0.	9.977	0.134	-0.016	0.000	0.981	0.981	0.981	
28.000	0.	9.907	-0.984	-0.026	0.000	6.009	6.009	0.	9.981	0.211	-0.024	0.000	0.458	0.458	0.458	
30.000	0.	9.805	-1.332	-0.039	0.000	6.012	6.012	0.	9.978	0.136	-0.034	0.000	1.331	1.331	1.331	
32.000	0.	9.671	-1.592	-0.058	0.000	9.657	9.657	0.	9.978	-0.039	-0.045	0.000	1.704	1.704	1.704	
34.000	0.	9.440	-2.129	-0.084	0.000	14.463	14.463	0.	10.014	-0.105	-0.062	0.000	2.267	2.267	2.267	
36.000	0.	9.152	-3.083	-0.113	0.000	18.782	18.782	0.	10.046	-0.075	-0.088	0.000	2.371	2.371	2.371	
38.000	0.	8.767	-4.576	-0.147	0.000	26.352	26.352	0.	10.095	-0.005	-0.122	0.000	3.606	3.606	3.606	
40.000	0.	8.280	-6.692	-0.188	0.000	31.362	31.362	0.	10.149	0.007	-0.163	0.000	3.348	3.348	3.348	
42.000	0.	7.699	-9.366	-0.240	0.000	37.757	37.757	0.	10.197	-0.171	-0.212	0.000	4.202	4.202	4.202	
44.000	0.	6.972	-12.870	-0.317	0.000	52.652	52.652	0.	10.265	-0.573	-0.269	0.000	7.887	7.887	7.887	
46.000	0.	5.966	-19.595	-0.430	0.000	71.325	71.325	0.	10.445	-0.726	-0.352	0.000	11.132	11.132	11.132	
48.000	0.	4.869	-30.859	-0.581	0.000	80.360	80.360	0.	10.627	-1.060	-0.464	0.000	10.999	10.999	10.999	
50.000	0.	3.822	-54.577	-0.804	0.000	125.388	125.388	0.	10.957	-1.213	-0.617	0.000	25.143	25.143	25.143	
52.000	0.	4.001	-95.947	-1.088	0.000	150.487	150.487	0.	11.421	-0.891	-0.839	0.000	25.934	25.934	25.934	
54.000	0.	6.124	-127.683	-1.502	0.000	193.882	193.882	0.	11.949	-0.784	-1.122	0.000	32.618	32.618	32.618	
56.000	0.	9.668	-143.883	-2.084	0.000	220.529	220.529	0.	12.423	-0.721	-1.497	0.000	27.056	27.056	27.056	
58.000	0.	13.878	-152.083	-2.748	0.000	235.094	235.094	0.	12.563	-0.690	-1.964	0.000	26.095	26.095	26.095	
60.000	0.	18.379	-157.625	-3.482	0.000	234.042	234.042	0.	12.544	-1.090	-2.446	0.000	28.518	28.518	28.518	
62.000	0.	22.709	-162.883	-4.505	0.000	238.449	238.449	0.	12.127	-2.354	-2.954	0.000	43.769	43.769	43.769	
64.000	0.	26.899	-167.023	-5.484	18.036	216.552	233.247	0.	11.096	-3.358	-3.457	0.000	73.428	73.428	73.428	
66.000	0.	30.558	-170.082	-6.297	154.456	176.484	324.673	0.	9.293	-4.188	-3.894	0.000	115.796	115.796	115.796	
68.000	0.	33.331	-172.228	-6.882	347.062	123.565	462.542	0.	6.439	-6.114	-4.250	0.000	175.281	175.281	175.281	
70.000	0.	35.027	-173.783	-7.296	502.714	67.615	559.233	0.	2.361	-19.607	-4.534	0.000	244.357	244.357	244.357	
72.000	0.	35.552	-174.978	-7.592	493.636	30.690	494.729	0.	3.448	-162.475	-4.760	0.000	303.225	303.225	303.225	
74.000	0.	35.540	-175.910	-7.807	466.062	24.286	466.652	0.	9.415	-172.198	-4.909	0.000	300.392	300.392	300.392	
76.000	0.	35.544	-176.681	-7.960	539.760	20.915	540.627	0.	15.394	-174.585	-4.988	0.000	299.780	299.780	299.780	
78.000	0.	35.568	-177.364	-8.079	558.450	19.211	561.920	0.	21.358	-175.662	-5.027	0.000	295.975	295.975	295.975	
80.000	0.	35.764	-178.027	-8.198	579.271	26.658	596.493	0.	27.109	-176.275	-5.046	22.243	278.693	300.921	300.921	

DATE: 2 SEPT 1988
 RUN DESCRIPTION: EXAMPLE 1: BASIC SLED TEST SIMULATION
 TWO BELT HARNESS WITH HYPERELLIPOID FOR DASH BOARD
 VEHICLE DECELERATION: SLED ACCELERATION - 20G PEAK
 CRASH VICTIM: 95TH PERCENTILE MALE

PAGE: 28.01

HP JOINT FORCES & TORQUES ON H IN H REFERENCE

TIME (MSEC)	JOINT FORCE (LB. 10**2)			JOINT TORQUE (IN.- LB. 10**2)		
	X	Y	Z	X	Y	Z
0.000	0.000	0.000	-0.126	0.0000+00	0.0000+00	0.0000+00
2.000	0.000	0.000	-0.129	-0.1190-06	0.1180-02	0.2320-07
4.000	0.001	0.000	-0.136	-0.9390-07	0.1940-02	0.7650-06
6.000	0.002	0.000	-0.142	0.2730-04	0.2630-02	0.5870-05
8.000	0.002	0.000	-0.148	0.1160-03	0.3030-02	0.2580-04
10.000	0.001	-0.001	-0.158	0.2380-03	0.2830-02	0.7250-04
12.000	-0.001	-0.001	-0.171	0.3850-03	0.1830-02	0.1600-03
14.000	-0.003	-0.001	-0.181	0.3830-03	0.1010-02	0.2840-03
16.000	-0.005	-0.001	-0.190	0.2970-03	0.4580-03	0.4270-03
18.000	-0.005	0.000	-0.202	0.1480-03	0.3700-03	0.6010-03
20.000	-0.007	0.000	-0.210	0.5030-05	0.6650-03	0.8000-03
22.000	-0.010	-0.001	-0.218	0.6700-04	0.5740-03	0.1070-02
24.000	-0.024	-0.021	-0.273	0.6750-02	-0.3250-02	0.1810-02
26.000	-0.027	-0.028	-0.270	0.8830-02	-0.3010-02	0.3050-02
28.000	-0.026	-0.002	-0.254	0.1060-02	-0.4440-03	0.4440-02
30.000	-0.029	0.004	-0.263	-0.1150-01	0.3300-02	0.5860-02
32.000	-0.097	-0.027	-0.164	-0.1050-01	-0.1090-01	0.7760-02
34.000	-0.090	-0.026	-0.153	-0.1800-02	-0.1980-01	0.1080-01
36.000	-0.123	-0.043	-0.131	0.5600-02	-0.1820-01	0.1410-01
38.000	-0.167	-0.086	-0.217	0.1210-01	-0.2870-01	0.1820-01
40.000	-0.187	-0.065	-0.200	-0.4350-02	-0.2420-01	0.2270-01
42.000	-0.235	-0.067	-0.280	-0.2070-01	-0.2390-01	0.2770-01
44.000	0.440	-0.151	0.231	-0.2000-01	-0.6750-01	0.3560-01
46.000	-0.476	-0.133	0.180	-0.6660-02	-0.9920-01	0.5000-01
48.000	-0.484	-0.078	0.423	-0.4090-01	-0.7850-01	0.6530-01
50.000	-0.964	-0.400	1.929	0.4750-01	-0.2290+00	0.9270-01
52.000	-1.025	-0.250	2.650	0.3330-01	-0.2270+00	0.1220+00
54.000	-1.406	-0.348	4.626	0.4170-01	-0.2810+00	0.1610+00
56.000	-1.393	-0.331	5.514	0.3790-01	-0.1680+00	0.2080+00
58.000	-1.416	-0.595	5.141	0.9100-01	0.1060-01	0.2440+00
60.000	-1.296	0.123	7.496	-0.9790-01	0.1010+00	0.2480+00
62.000	-1.049	0.016	5.739	-0.8070-01	0.3390+00	0.2640+00
64.000	-0.656	0.016	3.853	-0.4060-01	0.6920+00	0.2410+00
66.000	-0.606	-0.153	1.823	-0.4000-01	0.1140+01	0.2020+00
68.000	-0.887	-0.314	1.048	-0.1230+00	0.1740+01	0.1710+00
70.000	-0.958	-0.369	1.210	-0.2590+00	0.2420+01	0.1560+00
72.000	-0.758	-0.324	1.329	-0.3800+00	0.3000+01	0.1430+00
74.000	-0.842	-0.344	1.491	-0.3440+00	0.2980+01	0.8970-01
76.000	-0.939	-0.393	2.383	-0.3400+00	0.2980+01	0.6050-01
78.000	-0.884	-0.387	3.033	-0.3370+00	0.2940+01	0.4530-01
80.000	-0.676	-0.312	3.738	-0.3520+00	0.2990+01	0.4610-01

DATE: 2 SEPT 1988
 RUN DESCRIPTION: EXAMPLE 1: BASIC SLED TEST SIMULATION
 TWO BELT HARNESS WITH HYPERELLIPTOID FOR DASH BOARD
 VEHICLE DECELERATION: SLED ACCELERATION - 20G PEAK
 CRASH VICTIM: 95TH PERCENTILE MALE

INCLUDED SEGMENT NOS: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
 BODY PROPERTIES - REFERENCE SEGMENT NO. 16 (VEH)

TIME (MSEC)	CENTER OF GRAVITY (IN.)			LINEAR MOMENTUM (LB.-SEC.)			ANGULAR MOMENTUM (IN.- LB.-SEC.)			KINETIC ENERGY (LB.- IN.)		
	X	Y	Z	X	Y	Z	X	Y	Z	LINEAR	ANGULAR	TOTAL
0.000	18.183	0.000	-20.231	0.0000+00	0.0000+00	0.0000+00	0.0000+00	0.0000+00	0.0000+00	0.0000+00	0.0000+00	0.0000+00
2.000	18.183	0.000	-20.231	-0.2520-01	0.2870-06	0.7230-04	-0.6990-04	0.1970-01	0.3040-03	0.3820-01	0.4950-02	0.4310-01
4.000	18.185	0.000	-20.231	-0.1210+00	-0.2200-04	-0.6650-02	-0.6410-03	0.9990+00	0.1580-02	0.5050+00	0.3440-01	0.5390+00
6.000	18.189	0.000	-20.231	-0.2990+00	-0.4420-03	-0.3260-01	-0.1210-01	0.3410+01	0.1540-02	0.2420+01	0.1370+00	0.2560+01
8.000	18.197	0.000	-20.231	-0.5430+00	-0.2010-02	-0.8420-01	-0.5540-01	0.7300+01	-0.4720-02	0.7540+01	0.3910+00	0.7930+01
10.000	18.210	0.000	-20.232	-0.8410+00	-0.4970-02	-0.1670+00	-0.1380+00	0.1270+02	-0.1810-01	0.1840+02	0.8620+00	0.1920+02
12.000	18.230	0.000	-20.233	-0.1180+01	-0.9910-02	-0.2830+00	-0.2750+00	0.1940+02	-0.4480-01	0.3840+02	0.1550+01	0.3990+02
14.000	18.257	0.000	-20.234	-0.1550+01	-0.1500-01	-0.4350+00	-0.4160+00	0.2740+02	-0.7490-01	0.7170+02	0.2470+01	0.7420+02
16.000	18.294	0.000	-20.236	-0.1950+01	-0.1970-01	-0.6220+00	-0.5460+00	0.3660+02	-0.9950-01	0.1230+03	0.3650+01	0.1270+03
18.000	18.342	0.000	-20.239	-0.2390+01	-0.2430-01	-0.8440+00	-0.6710+00	0.4690+02	-0.1210+00	0.2000+03	0.5150-01	0.2050+03
20.000	18.402	0.000	-20.242	-0.2870+01	-0.2860-01	-0.1100+01	-0.7890+00	0.5840+02	-0.1340+00	0.3070+03	0.6980+01	0.3140+03
22.000	18.476	0.000	-20.247	-0.3400+01	-0.3560-01	-0.1400+01	-0.9820+00	0.7130+02	-0.1570+00	0.4510+03	0.9230+01	0.4610+03
24.000	18.564	-0.001	-20.253	-0.4010+01	-0.3500-01	-0.1830+01	-0.1640+01	0.8800+02	0.2040+00	0.6410+03	0.1180+02	0.6530+03
26.000	18.669	-0.001	-20.260	-0.4630+01	-0.3650-01	-0.2270+01	-0.2520+01	0.1050+03	0.6070+00	0.8860+03	0.1460+02	0.9010+03
28.000	18.792	-0.001	-20.270	-0.5280+01	-0.2040-02	-0.2700+01	-0.2260+01	0.1220+03	0.1410+01	0.1200+04	0.1790+02	0.1220+04
30.000	18.933	-0.001	-20.280	-0.5980+01	0.9730-01	-0.3110+01	0.4630+00	0.1390+03	0.2860+01	0.1580+04	0.2160+02	0.1600+04
32.000	19.095	0.000	-20.292	-0.6910+01	0.1320+00	-0.3410+01	0.1680+01	0.1600+03	0.4010+01	0.2040+04	0.2640+02	0.2060+04
34.000	19.278	0.000	-20.306	-0.8200+01	0.5830-01	-0.3690+01	-0.2400+00	0.1890+03	0.4550+01	0.2560+04	0.3430+02	0.2590+04
36.000	19.481	0.000	-20.320	-0.9880+01	0.2650-01	-0.4100+01	-0.1550+01	0.2260+03	0.6280+01	0.3150+04	0.4830+02	0.3200+04
38.000	19.704	0.001	-20.336	-0.1220+02	0.8010-01	-0.4540+01	-0.1660+01	0.2770+03	0.1010+02	0.3790+04	0.7210+02	0.3860+04
40.000	19.948	0.001	-20.354	-0.1490+02	0.3240+00	-0.4940+01	0.2720+01	0.3330+03	0.1710+02	0.4490+04	0.1040+03	0.4600+04
42.000	20.212	0.003	-20.373	-0.1800+02	0.7670+00	-0.5410+01	0.1080+02	0.4000+03	0.2780+02	0.5200+04	0.1360+03	0.5330+04
44.000	20.492	0.007	-20.393	-0.2200+02	0.1030+01	-0.5370+01	0.1380+02	0.4830+03	0.3770+02	0.5760+04	0.1660+03	0.5930+04
46.000	20.782	0.010	-20.412	-0.2720+02	0.8250+00	-0.4620+01	0.2800+01	0.5830+03	0.4270+02	0.6110+04	0.1920+03	0.6310+04
48.000	21.078	0.014	-20.426	-0.3360+02	0.1030+01	-0.3130+01	-0.1180+01	0.6990+03	0.5100+02	0.6230+04	0.2060+03	0.6440+04
50.000	21.369	0.015	-20.431	-0.4310+02	-0.4770+00	0.1200+01	-0.5580+02	0.8600+03	0.4090+02	0.5880+04	0.2510+03	0.6140+04
52.000	21.644	0.010	-20.416	-0.5450+02	-0.2460+01	0.6870+01	-0.1220+03	0.1040+04	0.1980+02	0.5390+04	0.3630+03	0.5750+04
54.000	21.894	-0.005	-20.377	-0.6850+02	-0.5450+01	0.1450+02	-0.2110+03	0.1260+04	-0.9270+01	0.4910+04	0.6730+03	0.5580+04
56.000	22.111	-0.030	-20.309	-0.8340+02	-0.8520+01	0.2200+02	-0.2950+03	0.1510+04	-0.3460+02	0.4710+04	0.1040+04	0.5740+04
58.000	22.290	-0.069	-20.216	-0.9910+02	-0.1350+02	0.2790+02	-0.4110+03	0.1790+04	-0.7970+02	0.4760+04	0.1280+04	0.6040+04
60.000	22.406	-0.126	-20.096	-0.1250+03	-0.1820+02	0.3750+02	-0.5240+03	0.2270+04	-0.1200+03	0.5810+04	0.2010+04	0.7820+04
62.000	22.463	-0.202	-19.952	-0.1400+03	-0.2210+02	0.3920+02	-0.6080+03	0.2590+04	-0.1460+03	0.6550+04	0.2260+04	0.8810+04
64.000	22.487	-0.287	-19.810	-0.1500+03	-0.2370+02	0.3700+02	-0.6470+03	0.2840+04	-0.1480+03	0.6730+04	0.2230+04	0.8960+04
66.000	22.490	-0.376	-19.683	-0.1580+03	-0.2410+02	0.3110+02	-0.6590+03	0.3060+04	-0.1450+03	0.6400+04	0.2060+04	0.8460+04
68.000	22.476	-0.465	-19.584	-0.1660+03	-0.2390+02	0.2200+02	-0.6580+03	0.3290+04	-0.1440+03	0.5790+04	0.1890+04	0.7680+04
70.000	22.446	-0.552	-19.523	-0.1710+03	-0.2350+02	0.1140+02	-0.6510+03	0.3480+04	-0.1410+03	0.5240+04	0.1770+04	0.7000+04
72.000	22.406	-0.640	-19.495	-0.1750+03	-0.2370+02	0.4160+01	-0.6480+03	0.3600+04	-0.1420+03	0.4840+04	0.1640+04	0.6480+04
74.000	22.358	-0.728	-19.490	-0.1800+03	-0.2380+02	-0.1300+01	-0.6460+03	0.3690+04	-0.1470+03	0.4530+04	0.1500+04	0.6040+04
76.000	22.298	-0.816	-19.505	-0.1830+03	-0.2400+02	-0.6350+01	-0.6480+03	0.3770+04	-0.1560+03	0.4430+04	0.1430+04	0.5870+04
78.000	22.233	-0.905	-19.537	-0.1850+03	-0.2420+02	-0.1140+02	-0.6550+03	0.3830+04	-0.1660+03	0.4560+04	0.1460+04	0.6020+04
80.000	22.165	-0.995	-19.588	-0.1860+03	-0.2440+02	-0.1610+02	-0.6590+03	0.3860+04	-0.1700+03	0.4770+04	0.1550+04	0.6320+04

RUN DESCRIPTION: EXAMPLE 1: BASIC SLED TEST SIMULATION
TWO BELT HARNESS WITH HYPERELLIPSOID FOR DASH BOARD

PAGE: 30.01

VEHICLE DECELERATION: SLED ACCELERATION - 20G PEAK
CRASH VICTIM: 95TH PERCENTILE MALE
CONTACT FORCES - SEGMENT PANELS VS. SEGMENTS

TIME (MSEC)	PANEL 1 (SEAT. 6 DEGREE OFF H) VS. SEGMENT 1 (LT)						PANEL 1 (SEAT. 6 DEGREE OFF H) VS. SEGMENT 6 (RUL)							
	DEFL- ECTION (IN.)	NORMAL FORCE (LB.)	FRICTION FORCE (LB.)	RESULTANT FORCE (LB.)	CONTACT LOCATION (IN.) (VEH REFERENCE)			DEFL- ECTION (IN.)	NORMAL FORCE (LB.)	FRICTION FORCE (LB.)	RESULTANT FORCE (LB.)	CONTACT LOCATION (IN.) (VEH REFERENCE)		
					X	Y	Z					X	Y	Z
0.000	0.458	137.10	0.00	137.10	14.372	0.000	-10.459	0.250	29.94	0.00	29.94	24.725	3.420	-11.545
2.000	0.458	137.02	8.94	137.31	14.373	0.000	-10.459	0.250	31.22	4.99	31.62	24.725	3.420	-11.545
4.000	0.458	136.86	27.51	139.60	14.375	0.000	-10.459	0.250	36.69	15.37	39.79	24.726	3.420	-11.545
6.000	0.458	136.72	46.60	144.44	14.380	0.000	-10.460	0.250	44.22	22.11	49.44	24.727	3.420	-11.545
8.000	0.457	138.78	64.83	153.18	14.390	0.000	-10.461	0.251	51.70	25.85	57.80	24.729	3.420	-11.546
10.000	0.458	144.58	72.29	161.65	14.407	0.000	-10.462	0.251	58.35	29.18	65.24	24.732	3.420	-11.546
12.000	0.458	152.48	76.24	170.48	14.431	0.000	-10.465	0.252	63.83	31.91	71.36	24.737	3.420	-11.546
14.000	0.458	161.71	80.85	180.80	14.464	0.000	-10.468	0.253	68.37	34.18	76.44	24.744	3.420	-11.547
16.000	0.459	172.39	86.19	192.73	14.508	0.000	-10.473	0.254	72.13	36.06	80.64	24.755	3.420	-11.548
18.000	0.459	184.02	92.01	205.74	14.565	-0.001	-10.479	0.255	75.12	37.56	83.99	24.770	3.420	-11.550
20.000	0.460	196.74	98.37	219.96	14.635	-0.001	-10.486	0.256	76.24	38.12	85.24	24.790	3.420	-11.552
22.000	0.461	211.45	105.72	236.41	14.720	-0.001	-10.495	0.257	74.47	37.23	83.26	24.816	3.420	-11.555
24.000	0.463	200.50	100.25	224.17	14.822	-0.001	-10.506	0.257	66.45	33.22	74.29	24.849	3.420	-11.558
26.000	0.464	199.45	99.72	222.99	14.942	-0.001	-10.519	0.257	58.64	29.32	65.56	24.891	3.420	-11.563
28.000	0.465	207.89	103.94	232.42	15.081	0.000	-10.533	0.256	49.90	24.95	55.79	24.942	3.420	-11.568
30.000	0.466	228.27	114.13	255.21	15.242	0.001	-10.550	0.255	45.48	22.74	50.85	25.002	3.420	-11.574
32.000	0.468	279.53	139.76	312.52	15.424	0.006	-10.569	0.253	53.74	26.87	60.08	25.069	3.421	-11.581
34.000	0.472	356.68	178.34	398.78	15.630	0.016	-10.591	0.251	53.97	26.99	60.35	25.138	3.421	-11.589
36.000	0.478	448.62	224.31	501.57	15.859	0.038	-10.615	0.248	38.40	19.20	42.93	25.207	3.421	-11.596
38.000	0.485	560.72	280.36	626.90	16.111	0.079	-10.641	0.244	37.83	18.91	42.29	25.272	3.422	-11.603
40.000	0.496	634.27	317.14	709.14	16.385	0.146	-10.670	0.239	65.17	32.59	72.87	25.324	3.424	-11.608
42.000	0.510	664.96	332.48	743.44	16.679	0.237	-10.701	0.234	89.40	44.70	99.96	25.364	3.428	-11.612
44.000	0.527	703.41	351.70	786.43	16.985	0.346	-10.733	0.230	119.38	59.69	133.47	25.391	3.434	-11.615
46.000	0.551	756.42	378.21	845.71	17.293	0.460	-10.765	0.227	157.62	78.81	176.23	25.402	3.441	-11.616
48.000	0.585	828.62	414.31	926.43	17.592	0.567	-10.797	0.225	201.14	100.57	224.88	25.394	3.451	-11.615
50.000	0.634	922.91	461.45	1031.84	17.867	0.638	-10.826	0.225	230.71	115.35	257.94	25.366	3.464	-11.613
52.000	0.705	1043.08	521.54	1166.20	18.102	0.630	-10.850	0.225	229.76	114.88	256.88	25.326	3.480	-11.608
54.000	0.806	1090.42	545.21	1219.13	18.285	0.526	-10.869	0.227	275.31	137.65	307.80	25.267	3.497	-11.602
56.000	0.942	1085.51	542.75	1213.64	18.410	0.360	-10.883	0.237	287.36	143.68	321.28	25.149	3.514	-11.590
58.000	1.113	1255.01	627.50	1403.14	18.483	0.194	-10.890	0.263	319.10	159.55	356.77	24.927	3.526	-11.567
60.000	1.323	1450.74	725.37	1621.98	18.497	0.037	-10.892	0.312	369.55	184.78	413.17	24.589	3.526	-11.531
62.000	1.576	1827.33	913.67	2043.02	18.468	-0.132	-10.889	0.373	440.16	220.08	492.11	24.222	3.511	-11.493
64.000	1.825	2293.38	1146.69	2564.07	18.450	-0.237	-10.887	0.443	571.58	285.79	639.04	23.850	3.487	-11.453
66.000	2.046	2723.02	1361.51	3044.43	18.439	-0.287	-10.886	0.512	745.66	372.83	833.68	23.509	3.460	-11.418
68.000	2.217	3434.09	1717.05	3839.43	18.418	-0.298	-10.883	0.565	895.39	447.69	1001.07	23.230	3.435	-11.388
70.000	2.318	3801.87	1900.93	4250.62	18.370	-0.288	-10.878	0.584	304.97	152.49	340.97	23.046	3.419	-11.369
72.000	2.356	2969.04	1484.52	3319.48	18.300	-0.277	-10.871	0.565	279.61	139.80	312.61	22.958	3.408	-11.360
74.000	2.342	2947.61	1473.80	3295.52	18.218	-0.271	-10.862	0.515	213.84	106.92	239.08	22.945	3.400	-11.359
76.000	2.282	2851.26	210.71	2859.04	18.125	-0.274	-10.853	0.436	108.38	54.19	121.18	23.013	3.395	-11.366
78.000	2.180	2688.19	119.57	2690.85	18.036	-0.292	-10.843	0.328	45.56	22.78	50.93	23.161	3.390	-11.381
80.000	2.039	2461.74	235.32	2472.96	17.954	-0.332	-10.835	0.191	18.66	9.33	20.86	23.244	3.381	-11.390

DATE: 2 SEPT 1988
 RUN DESCRIPTION: EXAMPLE 1: BASIC SLED TEST SIMULATION
 TWO BELT HARNESS WITH HYPERELLIPTOID FOR DASH BOARD
 VEHICLE DECELERATION: SLED ACCELERATION - 20G PEAK
 CRASH VICTIM: 95TH PERCENTILE MALE
 CONTACT FORCES - SEGMENT PANELS VS. SEGMENTS

PAGE: 31.01

TIME (MSEC)	PANEL 1 (SEAT. 6 DEGREE OFF H) VS. SEGMENT 9 (LUL)						PANEL 2 (BACK PANEL. 13 DEGR) VS. SEGMENT 1 (LT)							
	DEFL- ECTION	NORMAL FORCE	FRICTION FORCE	RESULTANT FORCE	CONTACT LOCATION (IN.)			DEFL- ECTION	NORMAL FORCE	FRICTION FORCE	RESULTANT FORCE	CONTACT LOCATION (IN.)		
	(IN.)	(LB.)	(LB.)	(LB.)	X	Y	Z	(IN.)	(LB.)	(LB.)	(LB.)	X	Y	Z
0.000	0.250	29.94	0.00	29.94	24.725	-3.420	-11.545	0.055	2.77	0.00	2.77	9.382	0.000	-12.674
2.000	0.250	31.23	5.00	31.62	24.725	-3.420	-11.545	0.055	2.76	0.40	2.79	9.382	0.000	-12.674
4.000	0.250	36.71	15.41	39.81	24.726	-3.420	-11.545	0.054	2.68	0.97	2.85	9.382	0.000	-12.675
6.000	0.250	44.24	22.12	49.46	24.727	-3.420	-11.545	0.050	2.51	1.25	2.80	9.382	0.000	-12.676
8.000	0.251	51.74	25.87	57.84	24.729	-3.420	-11.546	0.044	2.18	1.09	2.44	9.382	0.000	-12.677
10.000	0.251	58.41	29.21	65.30	24.732	-3.420	-11.546	0.033	1.65	0.83	1.85	9.381	0.000	-12.680
12.000	0.252	63.91	31.95	71.45	24.737	-3.420	-11.546	0.017	0.86	0.43	0.96	9.380	0.000	-12.684
14.000	0.253	68.45	34.23	76.53	24.744	-3.420	-11.547	-0.006	0.00	0.00	0.00	0.000	0.000	0.000
16.000	0.254	72.21	36.10	80.73	24.755	-3.420	-11.548	-0.036	0.00	0.00	0.00	0.000	0.000	0.000
18.000	0.255	75.19	37.60	84.07	24.770	-3.420	-11.550	-0.077	0.00	0.00	0.00	0.000	0.000	0.000
20.000	0.256	76.31	38.16	85.32	24.790	-3.420	-11.552	-0.128	0.00	0.00	0.00	0.000	0.000	0.000
22.000	0.257	74.63	37.31	83.43	24.815	-3.420	-11.555	-0.192	0.00	0.00	0.00	0.000	0.000	0.000
24.000	0.257	65.18	32.59	72.87	24.849	-3.420	-11.558	-0.269	0.00	0.00	0.00	0.000	0.000	0.000
26.000	0.257	56.40	28.20	63.06	24.891	-3.420	-11.563	-0.362	0.00	0.00	0.00	0.000	0.000	0.000
28.000	0.256	47.59	23.79	53.20	24.943	-3.420	-11.568	-0.472	0.00	0.00	0.00	0.000	0.000	0.000
30.000	0.254	34.68	17.34	38.78	25.005	-3.420	-11.575	-0.601	0.00	0.00	0.00	0.000	0.000	0.000
32.000	0.252	30.42	15.21	34.01	25.079	-3.419	-11.582	-0.749	0.00	0.00	0.00	0.000	0.000	0.000
34.000	0.248	29.66	14.83	33.16	25.168	-3.419	-11.592	-0.918	0.00	0.00	0.00	0.000	0.000	0.000
36.000	0.241	28.14	14.07	31.46	25.280	-3.419	-11.604	-1.108	0.00	0.00	0.00	0.000	0.000	0.000
38.000	0.227	25.44	12.72	28.44	25.426	-3.418	-11.619	-1.317	0.00	0.00	0.00	0.000	0.000	0.000
40.000	0.206	21.30	10.65	23.81	25.615	-3.415	-11.639	-1.546	0.00	0.00	0.00	0.000	0.000	0.000
42.000	0.179	16.89	8.45	18.89	25.850	-3.411	-11.663	-1.794	0.00	0.00	0.00	0.000	0.000	0.000
44.000	0.147	12.03	6.01	13.45	26.130	-3.405	-11.693	-2.056	0.00	0.00	0.00	0.000	0.000	0.000
46.000	0.112	6.87	3.43	7.68	26.435	-3.396	-11.725	-2.328	0.00	0.00	0.00	0.000	0.000	0.000
48.000	0.080	3.99	1.99	4.46	26.760	-3.386	-11.759	-2.601	0.00	0.00	0.00	0.000	0.000	0.000
50.000	0.056	2.78	1.39	3.11	27.008	-3.372	-11.785	-2.864	0.00	0.00	0.00	0.000	0.000	0.000
52.000	0.051	58.93	29.46	65.88	27.020	-3.355	-11.786	-3.104	0.00	0.00	0.00	0.000	0.000	0.000
54.000	0.076	84.08	42.04	94.01	26.648	-3.335	-11.747	-3.311	0.00	0.00	0.00	0.000	0.000	0.000
56.000	0.137	138.11	69.06	154.41	26.015	-3.313	-11.681	-3.479	0.00	0.00	0.00	0.000	0.000	0.000
58.000	0.228	165.60	82.80	185.14	25.312	-3.292	-11.607	-3.609	0.00	0.00	0.00	0.000	0.000	0.000
60.000	0.353	148.11	74.06	165.60	24.605	-3.280	-11.533	-3.695	0.00	0.00	0.00	0.000	0.000	0.000
62.000	0.522	357.14	178.57	399.30	23.920	-3.280	-11.461	-3.747	0.00	0.00	0.00	0.000	0.000	0.000
64.000	0.684	686.47	343.24	767.50	23.448	-3.290	-11.411	-3.784	0.00	0.00	0.00	0.000	0.000	0.000
66.000	0.819	1246.33	623.16	1393.44	23.155	-3.301	-11.380	-3.811	0.00	0.00	0.00	0.000	0.000	0.000
68.000	0.905	1699.87	849.94	1900.52	22.994	-3.313	-11.364	-3.828	0.00	0.00	0.00	0.000	0.000	0.000
70.000	0.924	758.55	379.27	848.08	22.932	-3.322	-11.357	-3.837	0.00	0.00	0.00	0.000	0.000	0.000
72.000	0.888	710.46	355.23	794.31	22.936	-3.329	-11.358	-3.838	0.00	0.00	0.00	0.000	0.000	0.000
74.000	0.814	611.98	305.99	684.21	22.992	-3.333	-11.363	-3.831	0.00	0.00	0.00	0.000	0.000	0.000
76.000	0.713	477.86	238.93	534.26	23.096	-3.334	-11.374	-3.811	0.00	0.00	0.00	0.000	0.000	0.000
78.000	0.591	314.21	157.11	351.30	23.266	-3.333	-11.392	-3.793	0.00	0.00	0.00	0.000	0.000	0.000
80.000	0.450	126.56	63.28	141.50	23.529	-3.333	-11.420	-3.782	0.00	0.00	0.00	0.000	0.000	0.000

DATE: 2 SEPT 1988

RUN DESCRIPTION: EXAMPLE 1: BASIC SLED TEST SIMULATION
TWO BELT HARNESS WITH HYPERELLIPSOID FOR DASH BOARD

PAGE: 32.01

VEHICLE DECELERATION: SLED ACCELERATION - 20G PEAK

CRASH VICTIM: 95TH PERCENTILE MALE

CONTACT FORCES - SEGMENT PANELS VS. SEGMENTS

TIME (MSEC)	PANEL 2 (BACK PANEL. 13 DEGR) VS. SEGMENT 2 (CT)				PANEL 2 (BACK PANEL. 13 DEGR) VS. SEGMENT 3 (UT)									
	DEFL- ECTION (IN.)	NORMAL FORCE (LB.)	FRICTION FORCE (LB.)	RESULTANT FORCE (LB.)	CONTACT LOCATION (IN.) (VEH REFERENCE)			DEFL- ECTION (IN.)	NORMAL FORCE (LB.)	FRICTION FORCE (LB.)	RESULTANT FORCE (LB.)	CONTACT LOCATION (IN.) (VEH REFERENCE)		
					X	Y	Z					X	Y	Z
0.000	-0.312	0.00	0.00	0.00	0.000	0.000	0.000	0.133	9.90	0.00	9.90	6.151	0.000	-26.665
2.000	-0.312	0.00	0.00	0.00	0.000	0.000	0.000	0.132	9.87	0.06	9.87	6.151	0.000	-26.665
4.000	-0.314	0.00	0.00	0.00	0.000	0.000	0.000	0.131	9.61	0.75	9.64	6.151	0.000	-26.665
6.000	-0.320	0.00	0.00	0.00	0.000	0.000	0.000	0.126	8.90	1.74	9.07	6.151	0.000	-26.664
8.000	-0.330	0.00	0.00	0.00	0.000	0.000	0.000	0.117	7.52	2.55	7.94	6.152	0.000	-26.663
10.000	-0.347	0.00	0.00	0.00	0.000	0.000	0.000	0.102	5.25	2.42	5.77	6.152	0.000	-26.661
12.000	-0.372	0.00	0.00	0.00	0.000	0.000	0.000	0.079	3.95	1.98	4.42	6.153	0.000	-26.658
14.000	-0.406	0.00	0.00	0.00	0.000	0.000	0.000	0.048	2.38	1.19	2.66	6.154	0.000	-26.654
16.000	-0.452	0.00	0.00	0.00	0.000	0.000	0.000	0.006	0.30	0.15	0.33	6.155	0.000	-26.649
18.000	-0.510	0.00	0.00	0.00	0.000	0.000	0.000	-0.048	0.00	0.00	0.00	0.000	0.000	0.000
20.000	-0.583	0.00	0.00	0.00	0.000	0.000	0.000	-0.114	0.00	0.00	0.00	0.000	0.000	0.000
22.000	-0.671	0.00	0.00	0.00	0.000	0.000	0.000	-0.196	0.00	0.00	0.00	0.000	0.000	0.000
24.000	-0.776	0.00	0.00	0.00	0.000	0.000	0.000	-0.294	0.00	0.00	0.00	0.000	0.000	0.000
26.000	-0.901	0.00	0.00	0.00	0.000	0.000	0.000	-0.409	0.00	0.00	0.00	0.000	0.000	0.000
28.000	-1.045	0.00	0.00	0.00	0.000	0.000	0.000	-0.542	0.00	0.00	0.00	0.000	0.000	0.000
30.000	-1.211	0.00	0.00	0.00	0.000	0.000	0.000	-0.696	0.00	0.00	0.00	0.000	0.000	0.000
32.000	-1.400	0.00	0.00	0.00	0.000	0.000	0.000	-0.872	0.00	0.00	0.00	0.000	0.000	0.000
34.000	-1.612	0.00	0.00	0.00	0.000	0.000	0.000	-1.067	0.00	0.00	0.00	0.000	0.000	0.000
36.000	-1.849	0.00	0.00	0.00	0.000	0.000	0.000	-1.284	0.00	0.00	0.00	0.000	0.000	0.000
38.000	-2.110	0.00	0.00	0.00	0.000	0.000	0.000	-1.523	0.00	0.00	0.00	0.000	0.000	0.000
40.000	-2.397	0.00	0.00	0.00	0.000	0.000	0.000	-1.783	0.00	0.00	0.00	0.000	0.000	0.000
42.000	-2.707	0.00	0.00	0.00	0.000	0.000	0.000	-2.064	0.00	0.00	0.00	0.000	0.000	0.000
44.000	-3.034	0.00	0.00	0.00	0.000	0.000	0.000	-2.361	0.00	0.00	0.00	0.000	0.000	0.000
46.000	-3.370	0.00	0.00	0.00	0.000	0.000	0.000	-2.662	0.00	0.00	0.00	0.000	0.000	0.000
48.000	-3.702	0.00	0.00	0.00	0.000	0.000	0.000	-2.960	0.00	0.00	0.00	0.000	0.000	0.000
50.000	-4.012	0.00	0.00	0.00	0.000	0.000	0.000	-3.238	0.00	0.00	0.00	0.000	0.000	0.000
52.000	-4.276	0.00	0.00	0.00	0.000	0.000	0.000	-3.473	0.00	0.00	0.00	0.000	0.000	0.000
54.000	-4.477	0.00	0.00	0.00	0.000	0.000	0.000	-3.652	0.00	0.00	0.00	0.000	0.000	0.000
56.000	-4.606	0.00	0.00	0.00	0.000	0.000	0.000	-3.757	0.00	0.00	0.00	0.000	0.000	0.000
58.000	-4.661	0.00	0.00	0.00	0.000	0.000	0.000	-3.787	0.00	0.00	0.00	0.000	0.000	0.000
60.000	-4.582	0.00	0.00	0.00	0.000	0.000	0.000	-3.689	0.00	0.00	0.00	0.000	0.000	0.000
62.000	-4.378	0.00	0.00	0.00	0.000	0.000	0.000	-3.478	0.00	0.00	0.00	0.000	0.000	0.000
64.000	-4.149	0.00	0.00	0.00	0.000	0.000	0.000	-3.225	0.00	0.00	0.00	0.000	0.000	0.000
66.000	-3.915	0.00	0.00	0.00	0.000	0.000	0.000	-2.965	0.00	0.00	0.00	0.000	0.000	0.000
68.000	-3.689	0.00	0.00	0.00	0.000	0.000	0.000	-2.720	0.00	0.00	0.00	0.000	0.000	0.000
70.000	-3.476	0.00	0.00	0.00	0.000	0.000	0.000	-2.499	0.00	0.00	0.00	0.000	0.000	0.000
72.000	-3.281	0.00	0.00	0.00	0.000	0.000	0.000	-2.303	0.00	0.00	0.00	0.000	0.000	0.000
74.000	-3.098	0.00	0.00	0.00	0.000	0.000	0.000	-2.127	0.00	0.00	0.00	0.000	0.000	0.000
76.000	-2.921	0.00	0.00	0.00	0.000	0.000	0.000	-1.960	0.00	0.00	0.00	0.000	0.000	0.000
78.000	-2.744	0.00	0.00	0.00	0.000	0.000	0.000	-1.791	0.00	0.00	0.00	0.000	0.000	0.000
80.000	-2.561	0.00	0.00	0.00	0.000	0.000	0.000	-1.618	0.00	0.00	0.00	0.000	0.000	0.000

RUN DESCRIPTION: EXAMPLE 1: BASIC SLED TEST SIMULATION
TWO BELT HARNESS WITH HYPERELLIPTOID FOR DASH BOARD

VEHICLE DECELERATION: SLED ACCELERATION - 20G PEAK

CRASH VICTIM: 95TH PERCENTILE MALE

CONTACT FORCES - SEGMENT PANELS VS. SEGMENTS

TIME (MSEC)	PANEL 3 (FLOOR.) VS. SEGMENT 8 (RF)				PANEL 3 (FLOOR.) VS. SEGMENT 11 (LF)									
	DEFL- ECTION (IN.)	NORMAL FORCE (LB.)	FRICTION FORCE (LB.)	RESULTANT FORCE (LB.)	CONTACT LOCATION (IN.)			DEFL- ECTION (IN.)	NORMAL FORCE (LB.)	FRICTION FORCE (LB.)	RESULTANT FORCE (LB.)	CONTACT LOCATION (IN.)		
					(VEH REFERENCE) X	Y	Z					(VEH REFERENCE) X	Y	Z
0.000	0.096	4.82	0.00	4.82	46.596	3.420	-1.300	0.096	4.82	0.00	4.82	46.596	-3.420	-1.300
2.000	0.097	4.88	0.97	4.98	46.596	3.420	-1.300	0.097	4.88	0.97	4.98	46.596	-3.420	-1.300
4.000	0.097	4.97	2.11	5.40	46.597	3.420	-1.300	0.097	4.98	2.11	5.40	46.597	-3.420	-1.300
6.000	0.097	5.05	2.52	5.64	46.601	3.420	-1.300	0.097	5.05	2.52	5.64	46.601	-3.420	-1.300
8.000	0.098	5.02	2.51	5.62	46.609	3.420	-1.300	0.098	5.03	2.52	5.62	46.609	-3.420	-1.300
10.000	0.098	4.90	2.45	5.48	46.620	3.420	-1.300	0.098	4.91	2.46	5.49	46.620	-3.420	-1.300
12.000	0.098	4.88	2.44	5.46	46.638	3.420	-1.300	0.098	4.89	2.44	5.46	46.638	-3.420	-1.300
14.000	0.097	4.83	2.42	5.41	46.662	3.420	-1.300	0.097	4.84	2.42	5.41	46.662	-3.420	-1.300
16.000	0.095	4.75	2.37	5.31	46.693	3.420	-1.300	0.095	4.76	2.38	5.32	46.694	-3.420	-1.300
18.000	0.092	4.61	2.31	5.16	46.734	3.420	-1.300	0.093	4.63	2.31	5.17	46.734	-3.420	-1.300
20.000	0.088	4.42	2.21	4.94	46.784	3.420	-1.300	0.089	4.44	2.22	4.97	46.784	-3.420	-1.300
22.000	0.083	4.17	2.08	4.66	46.844	3.420	-1.300	0.084	4.19	2.10	4.69	46.845	-3.420	-1.300
24.000	0.077	3.85	1.92	4.30	46.915	3.420	-1.300	0.078	3.88	1.94	4.34	46.916	-3.420	-1.300
26.000	0.069	3.46	1.73	3.86	46.998	3.420	-1.300	0.070	3.50	1.75	3.91	46.999	-3.420	-1.300
28.000	0.060	2.99	1.50	3.35	47.092	3.420	-1.300	0.061	3.05	1.52	3.41	47.093	-3.420	-1.300
30.000	0.049	2.46	1.23	2.75	47.199	3.420	-1.300	0.051	2.54	1.27	2.83	47.201	-3.420	-1.300
32.000	0.037	1.85	0.93	2.07	47.318	3.420	-1.300	0.040	1.98	0.99	2.21	47.322	-3.420	-1.300
34.000	0.022	1.12	0.56	1.26	47.448	3.420	-1.300	0.027	1.36	0.68	1.52	47.456	-3.420	-1.300
36.000	0.004	0.19	0.10	0.21	47.583	3.420	-1.300	0.013	0.65	0.32	0.73	47.597	-3.420	-1.300
38.000	-0.021	0.00	0.00	0.00	0.000	0.000	0.000	-0.004	0.00	0.00	0.00	0.000	0.000	0.000
40.000	-0.052	0.00	0.00	0.00	0.000	0.000	0.000	-0.024	0.00	0.00	0.00	0.000	0.000	0.000
42.000	-0.092	0.00	0.00	0.00	0.000	0.000	0.000	-0.047	0.00	0.00	0.00	0.000	0.000	0.000
44.000	-0.140	0.00	0.00	0.00	0.000	0.000	0.000	-0.072	0.00	0.00	0.00	0.000	0.000	0.000
46.000	-0.199	0.00	0.00	0.00	0.000	0.000	0.000	-0.101	0.00	0.00	0.00	0.000	0.000	0.000
48.000	-0.268	0.00	0.00	0.00	0.000	0.000	0.000	-0.137	0.00	0.00	0.00	0.000	0.000	0.000
50.000	-0.348	0.00	0.00	0.00	0.000	0.000	0.000	-0.183	0.00	0.00	0.00	0.000	0.000	0.000
52.000	-0.440	0.00	0.00	0.00	0.000	0.000	0.000	-0.245	0.00	0.00	0.00	0.000	0.000	0.000
54.000	-0.543	0.00	0.00	0.00	0.000	0.000	0.000	-0.330	0.00	0.00	0.00	0.000	0.000	0.000
56.000	-0.661	0.00	0.00	0.00	0.000	0.000	0.000	-0.434	0.00	0.00	0.00	0.000	0.000	0.000
58.000	-0.798	0.00	0.00	0.00	0.000	0.000	0.000	-0.554	0.00	0.00	0.00	0.000	0.000	0.000
60.000	-0.960	0.00	0.00	0.00	0.000	0.000	0.000	-0.686	0.00	0.00	0.00	0.000	0.000	0.000
62.000	-1.145	0.00	0.00	0.00	0.000	0.000	0.000	-0.830	0.00	0.00	0.00	0.000	0.000	0.000
64.000	-1.351	0.00	0.00	0.00	0.000	0.000	0.000	-0.974	0.00	0.00	0.00	0.000	0.000	0.000
66.000	-1.576	0.00	0.00	0.00	0.000	0.000	0.000	-1.122	0.00	0.00	0.00	0.000	0.000	0.000
68.000	-1.817	0.00	0.00	0.00	0.000	0.000	0.000	-1.277	0.00	0.00	0.00	0.000	0.000	0.000
70.000	-2.071	0.00	0.00	0.00	0.000	0.000	0.000	-1.446	0.00	0.00	0.00	0.000	0.000	0.000
72.000	-2.335	0.00	0.00	0.00	0.000	0.000	0.000	-1.625	0.00	0.00	0.00	0.000	0.000	0.000
74.000	-2.606	0.00	0.00	0.00	0.000	0.000	0.000	-1.812	0.00	0.00	0.00	0.000	0.000	0.000
76.000	-2.883	0.00	0.00	0.00	0.000	0.000	0.000	-2.005	0.00	0.00	0.00	0.000	0.000	0.000
78.000	-3.160	0.00	0.00	0.00	0.000	0.000	0.000	-2.199	0.00	0.00	0.00	0.000	0.000	0.000
80.000	-3.434	0.00	0.00	0.00	0.000	0.000	0.000	-2.387	0.00	0.00	0.00	0.000	0.000	0.000

DATE: 2 SEPT 1988
 RUN DESCRIPTION: EXAMPLE 1: BASIC SLED TEST SIMULATION
 TWO BELT HARNESS WITH HYPERELLIPSOID FOR DASH BOARD
 VEHICLE DECELERATION: SLED ACCELERATION - 20G PEAK
 CRASH VICTIM: 95TH PERCENTILE MALE
 CONTACT FORCES - SEGMENT PANELS VS. SEGMENTS

TIME (MSEC)	PANEL 4 (HEAD PAD. 13 DEGR) VS. SEGMENT 5 (H)								PANEL 10 (RUDDER PEDALS.) VS. SEGMENT 8 (RF)							
	DEFL- ECTION (IN.)	NORMAL FORCE (LB.)	FRICTION FORCE (LB.)	RESULTANT FORCE (LB.)	CONTACT VEH	LOCATION (IN.) REFERENCE			DEFL- ECTION (IN.)	NORMAL FORCE (LB.)	FRICTION FORCE (LB.)	RESULTANT FORCE (LB.)	CONTACT VEH	LOCATION (IN.) REFERENCE		
					X	Y	Z						X	Y	Z	
0.000	0.050	2.48	0.00	2.48	3.935	0.000	-40.978	0.002	0.08	0.00	0.08	51.239	3.420	-3.286		
2.000	0.049	2.47	0.02	2.47	3.935	0.000	-40.978	0.001	0.07	0.01	0.07	51.239	3.420	-3.286		
4.000	0.048	2.38	0.19	2.39	3.935	0.000	-40.978	0.002	0.11	0.04	0.12	51.239	3.420	-3.285		
6.000	0.043	2.15	0.41	2.19	3.935	0.000	-40.977	0.005	0.29	0.14	0.32	51.240	3.420	-3.287		
8.000	0.034	1.70	0.55	1.78	3.935	0.000	-40.976	0.010	0.75	0.37	0.83	51.245	3.420	-3.291		
10.000	0.019	0.95	0.41	1.03	3.936	0.000	-40.974	0.019	1.59	0.80	1.78	51.254	3.420	-3.298		
12.000	-0.003	0.00	0.00	0.00	0.000	0.000	0.000	0.031	2.66	1.33	2.98	51.266	3.420	-3.309		
14.000	-0.034	0.00	0.00	0.00	0.000	0.000	0.000	0.048	4.09	2.05	4.58	51.284	3.420	-3.323		
16.000	-0.075	0.00	0.00	0.00	0.000	0.000	0.000	0.070	5.92	2.96	6.62	51.306	3.420	-3.342		
18.000	-0.128	0.00	0.00	0.00	0.000	0.000	0.000	0.096	8.17	4.08	9.13	51.334	3.420	-3.365		
20.000	-0.194	0.00	0.00	0.00	0.000	0.000	0.000	0.128	13.66	6.83	15.28	51.367	3.420	-3.393		
22.000	-0.274	0.00	0.00	0.00	0.000	0.000	0.000	0.165	20.41	10.21	22.82	51.406	3.420	-3.426		
24.000	-0.371	0.00	0.00	0.00	0.000	0.000	0.000	0.205	28.17	14.09	31.50	51.464	3.420	-3.474		
26.000	-0.484	0.00	0.00	0.00	0.000	0.000	0.000	0.249	38.47	19.24	43.01	51.490	3.420	-3.496		
28.000	-0.617	0.00	0.00	0.00	0.000	0.000	0.000	0.296	49.29	24.64	55.10	51.503	3.420	-3.507		
30.000	-0.770	0.00	0.00	0.00	0.000	0.000	0.000	0.343	60.44	30.22	67.58	51.511	3.420	-3.514		
32.000	-0.945	0.00	0.00	0.00	0.000	0.000	0.000	0.392	71.83	35.92	80.31	51.516	3.420	-3.518		
34.000	-1.142	0.00	0.00	0.00	0.000	0.000	0.000	0.440	128.19	64.09	143.32	51.520	3.420	-3.521		
36.000	-1.363	0.00	0.00	0.00	0.000	0.000	0.000	0.477	178.75	89.37	199.85	51.521	3.420	-3.522		
38.000	-1.610	0.00	0.00	0.00	0.000	0.000	0.000	0.496	204.92	102.46	229.11	51.519	3.420	-3.521		
40.000	-1.882	0.00	0.00	0.00	0.000	0.000	0.000	0.498	191.03	95.51	213.58	51.515	3.420	-3.517		
42.000	-2.181	0.00	0.00	0.00	0.000	0.000	0.000	0.492	182.60	91.30	204.15	51.508	3.419	-3.511		
44.000	-2.505	0.00	0.00	0.00	0.000	0.000	0.000	0.482	168.82	84.41	188.75	51.501	3.418	-3.505		
46.000	-2.849	0.00	0.00	0.00	0.000	0.000	0.000	0.470	153.91	76.96	172.08	51.494	3.417	-3.499		
48.000	-3.211	0.00	0.00	0.00	0.000	0.000	0.000	0.460	139.82	69.91	156.32	51.489	3.415	-3.495		
50.000	-3.584	0.00	0.00	0.00	0.000	0.000	0.000	0.450	126.65	63.33	141.60	51.488	3.413	-3.495		
52.000	-3.960	0.00	0.00	0.00	0.000	0.000	0.000	0.441	114.28	57.14	127.77	51.494	3.410	-3.499		
54.000	-4.328	0.00	0.00	0.00	0.000	0.000	0.000	0.430	100.59	50.30	112.47	51.507	3.406	-3.510		
56.000	-4.675	0.00	0.00	0.00	0.000	0.000	0.000	0.415	79.36	39.68	88.73	51.531	3.402	-3.531		
58.000	-4.989	0.00	0.00	0.00	0.000	0.000	0.000	0.388	57.60	28.80	64.40	51.577	3.398	-3.569		
60.000	-5.254	0.00	0.00	0.00	0.000	0.000	0.000	0.346	49.17	24.59	54.98	51.679	3.394	-3.654		
62.000	-5.458	0.00	0.00	0.00	0.000	0.000	0.000	0.289	37.73	18.86	42.18	51.812	3.392	-3.767		
64.000	-5.609	0.00	0.00	0.00	0.000	0.000	0.000	0.215	23.04	11.52	25.75	51.976	3.393	-3.904		
66.000	-5.724	0.00	0.00	0.00	0.000	0.000	0.000	0.125	8.82	4.41	9.86	52.183	3.400	-4.077		
68.000	-5.817	0.00	0.00	0.00	0.000	0.000	0.000	0.020	0.99	0.49	1.10	52.440	3.417	-4.294		
70.000	-5.894	0.00	0.00	0.00	0.000	0.000	0.000	-0.101	0.00	0.00	0.00	0.000	0.000	0.000		
72.000	-5.963	0.00	0.00	0.00	0.000	0.000	0.000	-0.231	0.00	0.00	0.00	0.000	0.000	0.000		
74.000	-6.032	0.00	0.00	0.00	0.000	0.000	0.000	-0.367	0.00	0.00	0.00	0.000	0.000	0.000		
76.000	-6.105	0.00	0.00	0.00	0.000	0.000	0.000	-0.510	0.00	0.00	0.00	0.000	0.000	0.000		
78.000	-6.178	0.00	0.00	0.00	0.000	0.000	0.000	-0.655	0.00	0.00	0.00	0.000	0.000	0.000		
80.000	-6.252	0.00	0.00	0.00	0.000	0.000	0.000	-0.800	0.00	0.00	0.00	0.000	0.000	0.000		

DATE: 2 SEPT 1988
 RUN DESCRIPTION: EXAMPLE 1: BASIC SLED TEST SIMULATION
 TWO BELT HARNESS WITH HYPERELLIPSOID FOR DASH BOARD
 VEHICLE DECELERATION: SLED ACCELERATION - 20G PEAK
 CRASH VICTIM: 95TH PERCENTILE MALE
 CONTACT FORCES - SEGMENT PANELS VS. SEGMENTS

PAGE: 35.01

TIME (MSEC)	PANEL 10 (RUDDER PEDALS.) VS. SEGMENT 11 (LF)				CONTACT LOCATION (IN.)		
	DEFL- ECTION (IN.)	NORMAL FORCE (LB.)	FRICTION FORCE (LB.)	RESULTANT FORCE (LB.)	(VEH X	Y	Z
0.000	0.002	0.08	0.00	0.08	51.239	-3.420	-3.286
2.000	0.001	0.07	0.01	0.07	51.239	-3.420	-3.286
4.000	0.002	0.11	0.04	0.12	51.239	-3.420	-3.285
6.000	0.005	0.29	0.14	0.32	51.240	-3.420	-3.287
8.000	0.010	0.75	0.37	0.83	51.245	-3.420	-3.291
10.000	0.019	1.60	0.80	1.78	51.254	-3.420	-3.298
12.000	0.031	2.67	1.33	2.98	51.266	-3.420	-3.308
14.000	0.048	4.10	2.05	4.58	51.284	-3.420	-3.323
16.000	0.070	5.93	2.96	6.63	51.306	-3.420	-3.342
18.000	0.096	8.18	4.09	9.15	51.334	-3.420	-3.365
20.000	0.128	13.70	6.85	15.32	51.366	-3.420	-3.393
22.000	0.165	20.47	10.23	22.88	51.406	-3.420	-3.425
24.000	0.206	28.26	14.13	31.59	51.464	-3.420	-3.474
26.000	0.250	38.58	19.29	43.14	51.489	-3.420	-3.496
28.000	0.296	49.45	24.73	55.29	51.503	-3.420	-3.507
30.000	0.344	60.69	30.34	67.85	51.511	-3.420	-3.513
32.000	0.394	72.29	36.14	80.82	51.516	-3.420	-3.518
34.000	0.444	133.60	66.80	149.37	51.519	-3.420	-3.520
36.000	0.483	188.09	94.05	210.29	51.519	-3.420	-3.520
38.000	0.507	219.74	109.87	245.68	51.515	-3.420	-3.518
40.000	0.514	216.05	108.03	241.55	51.508	-3.420	-3.511
42.000	0.515	213.02	106.51	238.16	51.497	-3.421	-3.502
44.000	0.514	212.01	106.00	237.03	51.482	-3.422	-3.490
46.000	0.514	211.89	105.95	236.90	51.465	-3.423	-3.475
48.000	0.514	212.07	106.04	237.10	51.446	-3.425	-3.459
50.000	0.511	207.70	103.85	232.22	51.425	-3.427	-3.441
52.000	0.498	190.37	95.18	212.83	51.399	-3.430	-3.420
54.000	0.472	155.61	77.80	173.97	51.361	-3.432	-3.388
56.000	0.436	107.92	53.96	120.66	51.306	-3.435	-3.342
58.000	0.395	59.01	29.50	65.97	51.268	-3.437	-3.310
60.000	0.353	50.53	25.27	56.50	51.259	-3.438	-3.302
62.000	0.307	41.43	20.72	46.32	51.288	-3.438	-3.327
64.000	0.268	33.64	16.82	37.61	51.387	-3.436	-3.409
66.000	0.234	26.90	13.45	30.07	51.514	-3.432	-3.516
68.000	0.200	20.05	10.02	22.41	51.657	-3.425	-3.637
70.000	0.159	13.90	6.95	15.54	51.813	-3.413	-3.767
72.000	0.112	6.75	3.37	7.55	51.972	-3.393	-3.901
74.000	0.056	2.80	1.40	3.13	52.128	-3.366	-4.032
76.000	-0.008	0.00	0.00	0.00	0.000	0.000	0.000
78.000	-0.077	0.00	0.00	0.00	0.000	0.000	0.000
80.000	-0.141	0.00	0.00	0.00	0.000	0.000	0.000

RUN DESCRIPTION: EXAMPLE 1: BASIC SLED TEST SIMULATION
 TWO BELT HARNESS WITH HYPERELLIPTOID FOR DASH BOARD
 VEHICLE DECELERATION: SLED ACCELERATION - 20G PEAK
 CRASH VICTIM: 95TH PERCENTILE MALE

HARNESS SYSTEM BELT ENDPOINT FORCES

TIME (MSEC)	BELT NO. 1 OF HARNESS NO. 1		BELT NO. 12 OF HARNESS NO. 1		BELT NO. 13 OF HARNESS NO. 1		BELT NO. 27 OF HARNESS NO. 1	
	POINT NO. 1 STRAIN (IN./ IN.)	FORCE (LB.)	POINT NO. 12 STRAIN (IN./ IN.)	FORCE (LB.)	POINT NO. 13 STRAIN (IN./ IN.)	FORCE (LB.)	POINT NO. 27 STRAIN (IN./ IN.)	FORCE (LB.)
0.000	0.000000	0.00	0.000000	0.00	0.000000	0.00	0.000000	0.00
2.000	0.000025	0.37	0.000017	0.25	0.000014	0.21	0.000018	0.27
4.000	0.000119	1.79	0.000065	0.98	0.000057	0.85	0.000152	2.27
6.000	0.000281	4.22	0.000150	2.25	0.000124	1.86	0.000470	7.04
8.000	0.000523	7.85	0.000276	4.15	0.000219	3.29	0.001018	15.27
10.000	0.000854	12.81	0.000438	6.57	0.000336	5.04	0.001803	27.04
12.000	0.001023	15.34	0.000481	7.21	0.000344	5.16	0.002454	36.81
14.000	0.001320	19.80	0.000515	7.72	0.000433	6.50	0.003168	47.52
16.000	0.001672	25.08	0.000551	8.27	0.000496	7.44	0.003856	57.84
18.000	0.001420	21.30	0.000400	6.00	0.000404	6.06	0.003427	51.41
20.000	0.001831	27.46	0.000484	7.26	0.000498	7.46	0.004276	64.13
22.000	0.002363	35.45	0.000617	9.26	0.000643	9.65	0.005479	82.19
24.000	0.003085	46.27	0.000877	13.15	0.000921	13.81	0.019008	285.12
26.000	0.004040	60.61	0.001215	18.22	0.001284	19.26	0.021036	315.53
28.000	0.006003	90.05	0.001724	25.85	0.001768	26.52	0.027340	410.10
30.000	0.009472	142.08	0.002256	33.83	0.002429	36.44	0.035707	564.14
32.000	0.013979	209.69	0.003249	48.73	0.004372	65.58	0.023671	355.06
34.000	0.019790	296.85	0.004754	71.30	0.008401	126.02	0.020865	312.97
36.000	0.026146	392.20	0.007124	106.86	0.018334	275.01	0.025025	375.38
38.000	0.032200	494.00	0.010831	162.46	0.030199	453.97	0.039302	636.04
40.000	0.037809	606.17	0.021023	315.35	0.040241	654.82	0.056200	1170.59
42.000	0.043063	711.27	0.036025	570.49	0.050245	863.00	0.075150	2182.56
44.000	0.048343	816.85	0.053858	1054.46	0.060539	1408.57	0.064897	1639.53
46.000	0.053827	1052.84	0.071518	1990.44	0.072248	2029.17	0.084525	2679.84
48.000	0.055271	1129.35	0.087363	2830.23	0.056456	1192.15	0.103767	3631.84
50.000	0.058800	1316.42	0.098307	3410.27	0.065766	1685.60	0.093883	3175.81
52.000	0.064710	1629.61	0.098688	3430.48	0.068526	1831.86	0.107224	3752.85
54.000	0.072120	2022.35	0.087992	2863.56	0.074577	2152.58	0.092638	3109.84
56.000	0.076068	2231.59	0.071069	1966.67	0.075155	2183.20	0.089018	2917.94
58.000	0.074497	2148.32	0.049516	840.31	0.069789	1898.82	0.084438	2675.23
60.000	0.061993	1485.61	0.026359	395.38	0.082583	2576.88	0.093363	3148.22
62.000	0.047024	790.48	0.005156	77.34	0.032040	490.80	0.086107	2763.69
64.000	0.028177	422.66	0.000000	0.00	0.024499	367.48	0.065353	1663.72
66.000	0.007038	105.57	0.000000	0.00	0.006231	93.46	0.051787	944.69
68.000	0.000000	0.00	0.000000	0.00	0.000000	0.00	0.037962	609.25
70.000	0.000000	0.00	0.000000	0.00	0.000000	0.00	0.022501	337.51
72.000	0.000000	0.00	0.000000	0.00	0.000000	0.00	0.012711	190.67
74.000	0.000000	0.00	0.000000	0.00	0.000000	0.00	0.006159	92.39
76.000	0.000000	0.00	0.000000	0.00	0.000000	0.00	0.000000	0.00
78.000	0.000000	0.00	0.000000	0.00	0.000000	0.00	0.000000	0.00
80.000	0.000000	0.00	0.000000	0.00	0.000000	0.00	0.000000	0.00

DATE: 2 SEPT 1988
 RUN DESCRIPTION: EXAMPLE 1: BASIC SLED TEST SIMULATION
 TWO BELT HARNESS WITH HYPERELLIPSOID FOR DASH BOARD
 VEHICLE DECELERATION: SLED ACCELERATION - 20G PEAK
 CRASH VICTIM: 95TH PERCENTILE MALE

PAGE: 40.01

CONTACT FORCES - SEGMENT NO. 9 (LUL) VS. SEGMENT NO. 15 (LLA)

TIME (MSEC)	DEFL- ECTION (IN.)	NORMAL FORCE (LB.)	FRICTION FORCE (LB.)	RESULTANT FORCE (LB.)	CONTACT LOCATION (IN.)					
					SEG. 9 LOCAL REFERENCE			SEG. 15 LOCAL REFERENCE		
					X	Y	Z	X	Y	Z
0.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
2.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
4.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
6.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
8.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
10.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
12.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
14.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
16.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
18.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
20.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
22.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
24.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
26.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
28.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
30.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
32.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
34.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
36.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
38.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
40.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
42.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
44.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
46.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
48.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
50.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
52.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
54.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
56.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
58.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
60.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
62.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
64.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
66.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
68.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
70.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
72.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
74.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
76.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
78.000	0.282	132.75	33.19	136.84	1.229	-1.146	10.441	-1.121	0.658	6.935
80.000	0.685	321.84	80.46	331.75	1.096	-1.142	10.037	-1.019	0.683	6.725

DATE: 2 SEPT 1988
 RUN DESCRIPTION: EXAMPLE 1: BASIC SLED TEST SIMULATION
 TWO BELT HARNESS WITH HYPERELLIPSOID FOR DASH BOARD
 VEHICLE DECELERATION: SLED ACCELERATION - 20G PEAK
 CRASH VICTIM: 95TH PERCENTILE MALE

PAGE: 41.01

CONTACT FORCES - SEGMENT NO. 13 (RLA) VS. SEGMENT NO. 16 (VEH)

TIME (MSEC)	DEFL- ECTION (IN.)	NORMAL FORCE (LB.)	FRICTION FORCE (LB.)	RESULTANT FORCE (LB.)	CONTACT LOCATION (IN.)					
					SEG. 13 X	LOCAL Y	REFERENCE Z	SEG. 16 X	LOCAL Y	REFERENCE Z
0.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
2.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
4.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
6.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
8.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
10.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
12.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
14.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
16.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
18.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
20.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
22.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
24.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
26.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
28.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
30.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
32.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
34.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
36.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
38.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
40.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
42.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
44.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
46.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
48.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
50.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
52.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
54.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
56.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
58.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
60.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
62.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
64.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
66.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
68.000	0.217	24.79	12.39	27.71	1.714	0.433	2.758	32.371	6.599	-23.212
70.000	0.496	193.03	96.51	215.81	1.651	0.426	3.110	32.482	6.706	-23.329
72.000	0.671	444.83	222.41	497.33	1.602	0.408	3.504	32.546	6.820	-23.401
74.000	0.627	362.71	181.36	405.53	1.587	0.388	3.859	32.512	6.921	-23.391
76.000	0.420	87.00	43.50	97.27	1.600	0.370	4.164	32.409	7.003	-23.320
78.000	0.219	23.90	11.95	26.72	1.612	0.354	4.449	32.311	7.076	-23.249
80.000	0.103	5.46	2.73	6.10	1.608	0.342	4.722	32.256	7.143	-23.206

DATE: 2 SEPT 1988
 RUN DESCRIPTION: EXAMPLE 1: BASIC SLED TEST SIMULATION
 TWO BELT HARNESS WITH HYPERELLIPSOID FOR DASH BOARD
 VEHICLE DECELERATION: SLED ACCELERATION - 20G PEAK
 CRASH VICTIM: 95TH PERCENTILE MALE

PAGE: 42.01

CONTACT FORCES - SEGMENT NO. 15 (LLA) VS. SEGMENT NO. 16 (VEH)

TIME (MSEC)	DEFL- ECTION (IN.)	NORMAL FORCE (LB.)	FRICTION FORCE (LB.)	RESULTANT FORCE (LB.)	CONTACT LOCATION (IN.)					
					SEG. 15 LOCAL REFERENCE			SEG. 16 LOCAL REFERENCE		
					X	Y	Z	X	Y	Z
0.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
2.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
4.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
6.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
8.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
10.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
12.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
14.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
16.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
18.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
20.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
22.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
24.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
26.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
28.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
30.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
32.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
34.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
36.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
38.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
40.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
42.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
44.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
46.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
48.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
50.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
52.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
54.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
56.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
58.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
60.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
62.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
64.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
66.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
68.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
70.000	0.000	0.00	0.00	0.00	0.000	0.000	0.000	0.000	0.000	0.000
72.000	0.524	226.73	113.37	253.49	1.755	0.307	0.651	32.615	-6.119	-23.311
74.000	0.804	610.21	305.10	682.23	1.712	0.298	0.537	32.732	-6.130	-23.436
76.000	0.907	736.13	368.06	823.02	1.699	0.281	0.427	32.764	-6.147	-23.486
78.000	0.798	590.20	295.10	659.86	1.720	0.263	0.294	32.694	-6.176	-23.450
80.000	0.583	303.74	151.87	339.60	1.758	0.244	0.157	32.574	-6.224	-23.368

HEAD INJURY CRITERION

HIC = 192.44 TIME DURATION = 52.000 TO 65.000 MSEC
WITH HEAD RESULTANTS = 25.844 AND 23.464 G'S

AVERAGE HEAD RESULTANT FOR TIME DURATION = 46.574 G'S

HEAD SEVERITY INDEX

HSI = 245.20

MAX HEAD RESULTANT = 65.716 G'S AT 60.000 MSEC

CHEST SEVERITY INDEX

CSI = 733.64

MAX CHEST RESULTANT = 146.473 G'S AT 59.000 MSEC

SUB	CALLS	TIME	%
MAIN3D	1	155	0.53
INPUT	1	305	1.05
CHAIN	484	832	2.85
EJOINT	484	29	0.10
DINT	41	885	3.04
PDAUX	568	1112	3.82
DAUX	483	1056	3.62
SETUP1	483	766	2.63
CONTC	483	444	1.52
PLELP	5313	2244	7.70
SEGSEG	2898	2029	6.96
HBELT	1072	4385	15.05
VISPR	483	1922	6.59
SETUP2	483	157	0.54
DAUX11	483	1562	5.36
DAUX12	483	678	2.33
DAUX22	483	370	1.27
FMSOL	1072	3988	13.68
OUTPUT	86	707	2.43
UPDATE	85	53	0.18
HPTURB	85	3862	13.25
DZP	482	455	1.56
POSTPR	1	1149	3.94
TOTAL		29145	100.00

APPENDIX D1

Example 2, ATB Dynamic Joint Test
Input File (EX2.AIN)

APPENDIX D2

Example 2, ATB Dynamic Joint Test
386 Directory File (ATBIV2.DIR)

EXP2.TP1	UNIT 1	ATB OUTPUT, VIEW BODY ELEMENT AND CONTACT PLANE INPUT
EXP2.PRP	UNIT 2	ATB PRINTER PLOTS
ATB.ROU	UNIT 3	ATB RESTART RUN OUTPUT FILE
ATB.RIN	UNIT 4	ATB RESTART RUN INPUT FILE
EX2.AIN	UNIT 5	ATB INPUT, GEBOD BODY DESCRIPTION OUTPUT
EXP2.AOU	UNIT 6	ATB PRIMARY OUTPUT FILE
EXP2.TP8	UNIT 8	ATB SUBROUTINE POSTPR OUTPUT FILE
0 IDEF	UNIT 10	DRAWING OPTION SEE PLOTWORKS, PLOT88, 3.1
91 IOPORT	UNIT 10	HARDWARE INTERFACE 91/ 1 SEE PLOTWORKS, PLOT88, 3.1
91 MODEL	UNIT 10	OUTPUT DEVICE 91/64 SEE PLOTWORKS, PLOT88, 3.1
EXP2.T24	21 - 85	LAST ATB TABULAR TIME HISTORY OUTPUT FILE (21-85)

APPENDIX D3

Example 2, ATB Dynamic Joint Test
386 Output File (EXP2.AOU)

DEVELOPED BY CALSPAN CORP., P.O. BOX 400, BUFFALO NY 14225
 AND BY J&J TECHNOLOGIES INC., ORCHARD PARK, NY 14127

FOR THE AIR FORCE ARMSTRONG AEROSPACE MEDICAL RESEARCH
 LABORATORY, WRIGHT PATTERSON AIR FORCE BASE
 UNDER CONTRACTS F33615-75C-5002, -78C-0516 AND -80C-05117

AND FOR THE NATIONAL HIGHWAY TRAFFIC SAFETY ADMINISTRATION,
 U.S. DEPARTMENT OF TRANSPORTATION, UNDER CONTRACTS
 FH-11-7592, HS-053-2-485, HS-6-01300 AND HS-6-01410.

PROGRAM DOCUMENTATION: NHTSA REPORT NOS. DOT-HS-801-507
 THROUGH 510 (FORMERLY CALSPAN REPORT NO. 20-5180-L-1),
 AVAILABLE FROM NTIS (ACCESSION NOS. PB-241692,3,4 AND 5),
 APPENDIXES A-J TO THE ABOVE (AVAILABLE FROM CALSPAN),
 AND REPORT NOS. AMRL-TR-75-14 (NTIS NO. AD-A014 816),
 AFAMRL-TR-80-14 (NTIS NO. AD-A088 029), AND
 AFAMRL-TR-83-073 (NTIS NO. AD-B079 184).

PROGRAM ATB-IV 80386 IMPLEMENTATION COMPLETED BY KETRON,
 OCTOBER 31, 1989, UNDER CONTRACT F33615-88-C-0543

FEB. 4, 1988 IRSIN= 0 IRSOUT= 0 RSTIME = 0.0000

CARDS A

EXAMPLE 2: DYNAMIC JOINT TEST
 SLIP JOINT / 600 KNOT WIND

UNITL = IN. UNITH = LB. UNITT = SEC. GRAVITY VECTOR = (0.0000, 0.0000, 386.0880) G = 386.0880

NDINT = 4 NSTEPS = 20 DT = 0.002000 HO = 0.000500 HMAX = 0.001000 HMIN = 0.000063

NPRT ARRAY

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
1	0	20	2	0	0	0	0	0	0	0	0	0	0	0	0	0	16	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	1

SEGMENT	WEIGHT (LB.)	PRINCIPAL MOMENTS OF INERTIA (LB.-SEC.**2- IN.)			SEGMENT CONTACT ELLIPSOID						PRINCIPAL AXES (DEG)		
		(LB.-SEC.**2- IN.)			SEMIAXES (IN.)			CENTER (IN.)			PRINCIPAL AXES (DEG)		
		X	Y	Z	X	Y	Z	X	Y	Z	YAW	PITCH	ROLL
1 LARM L	5.901	0.3331	0.3331	0.0214	1.871	1.871	10.269	0.000	0.000	0.000	0.00	0.00	0.00
2 UARM U	5.542	0.1743	0.1743	0.0259	2.122	2.122	7.497	0.000	0.000	0.000	0.00	0.00	0.00

CARDS 8.3

JOINT	J SYM	PLOT	JNT	PIN	LOCATION(IN.) - SEG(JNT)			LOCATION(IN.) - SEG(J+1)			PRIN. AXIS(DEG) - SEG(JNT)			PRIN. AXIS(DEG) - SEG(J+1)		
					X	Y	Z	X	Y	Z	YAW	PITCH	ROLL	YAW	PITCH	ROLL
1	ELBW	E	1	5	0.000	0.000	8.200	0.000	0.000	-5.420	0.00	0.00	0.00	0.00	-67.19	0.00

UNLOCK CONDITIONS FOR SLIP JOINTS

JOINT	TENSION (LB.)	COMPRESSION (LB.)
1	0.000	0.000

FLEXURAL SPRING CHARACTERISTICS

TORSIONAL SPRING CHARACTERISTICS

JOINT	SPRING COEF. (IN. LB./DEG**J)			ENERGY DISSIPATION COEF.	JOINT STOP (DEG)	SPRING COEF. (IN. LB./DEG**J)			ENERGY DISSIPATION COEF.	JOINT STOP (DEG)
	LINEAR (J=1)	QUADRATIC (J=2)	CUBIC (J=3)			LINEAR (J=1)	QUADRATIC (J=2)	CUBIC (J=3)		
1 ELBW	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

CARDS B.5

JOINT VISCOUS CHARACTERISTICS AND LOCK-UNLOCK CONDITIONS

JOINT	VISCOUS COEFFICIENT (IN. LB.SEC./DEG)	COULOMB FRICTION COEF. (IN. LB.)	FULL FRICTION ANGULAR VELOCITY (DEG/SEC.)	MAX TORQUE FOR A LOCKED JOINT (IN. LB.)	MIN TORQUE FOR UNLOCKED JOINT (IN. LB.)	MIN. ANG. VELOCITY FOR UNLOCKED JOINT (RAD/SEC.)	IMPULSE RESTITUTION COEFFICIENT
1 ELBW	0.000	0.00	30.00	0.00	0.00	0.00	0.000

SEGMENT INTEGRATION CONVERGENCE TEST INPUT

SEGMENT NO. SYM	ANGULAR VELOCITIES (RAD/SEC.)			LINEAR VELOCITIES (IN./SEC.)			ANGULAR ACCELERATIONS (RAD/SEC.**2)			LINEAR ACCELERATIONS (IN./SEC.**2)		
	MAG. TEST	ABS. ERROR	REL. ERROR	MAG. TEST	ABS. ERROR	REL. ERROR	MAG. TEST	ABS. ERROR	REL. ERROR	MAG. TEST	ABS. ERROR	REL. ERROR
1 LARM	0.000	0.000	0.0000	0.000	0.000	0.0000	0.010	0.010	0.0010	0.001	0.001	0.0010
2 UARM	0.000	0.000	0.0000	0.000	0.000	0.0000	0.010	0.010	0.0010	0.000	0.000	0.0000

CONSTANT WIND VELOCITY OF 600 KNOTS

YAW	PITCH	ROLL	VIPS	VTIME	XO(X)	XO(Y)	XO(Z)	NATAB	ATO	ADT	MSEG
0.000	0.000	0.000	12161.000	0.000	0.000	0.000	0.000	-3	0.000000	2.300000	0

SPLINE FIT TABULAR INPUT

LTYPE = 2 LFIT = 1 NPTS = 3

INITIAL LINEAR POSITION (IN.)				INITIAL ANGULAR POSITION (DEG)									
TIME(SEC.)=	0.0000	X=	0.000	Y=	0.000	Z=	0.000	X=	0.000	Y=	0.000	Z=	0.000

TIME(SEC.)	LINEAR VELOCITY (IN./SEC.)			ANGULAR VELOCITY (DEG/SEC.)		
	X	Y	Z	X	Y	Z
0.00000	0.000	0.000	12161.000	0.000	0.000	0.000
2.30000	0.000	0.000	12161.000	0.000	0.000	0.000
4.60000	0.000	0.000	12161.000	0.000	0.000	0.000

VEHICLE LINEAR TIME HISTORY CONSTANT WIND VELOCITY OF 600 KNOTS

PAGE NO. 1

TIME (MSEC)	LINEAR DECELERATIONS (G'S)			LINEAR VELOCITIES (IN./SEC.)			LINEAR DISPLACEMENTS (IN.)		
	X	Y	Z	X	Y	Z	X	Y	Z
0.000	0.000	0.000	0.000	0.000	0.000	12161.000	0.000	0.000	0.000
2300.000	0.000	0.000	0.000	0.000	0.000	12161.000	0.000	0.000	27970.300
4600.000	0.000	0.000	0.000	0.000	0.000	12161.000	0.000	0.000	55940.600

NPL NBLT NBAG NBLP NQ NSD NHRSS NWINDF NJNTF NFORCE
1 0 0 0 0 0 0 1 0 0

PAGE 8
CARD D.1
CARDS D.2

PLANE INPUTS

PLANE NO. 1 WIND PLANE

	X	Y	Z
POINT 1	10.0000	10.0000	5.0000
POINT 2	-10.0000	10.0000	5.0000
POINT 3	10.0000	-10.0000	5.0000

BODY SEGMENT SYMMETRY INPUT

CARD D.7

SEG NO. 1 2

NSYM(J) 0 0

WIND FORCE FUNCTION NO. 1 WIND FORCE

NT1(1) = 1

PAGE 9
CARDS E.6

SPEC. HEAT RATIO	SONIC VEL.	ABS. PRESS.	SEGMENT	REF. SEGMENT
1.4000	13404.0000	14.6960	1LARM	3VEH

SEGMENT-ELLIPSOID	SEGMENT-PLANE	WIND FORCE FUNCTION	DRAG COEFFICIENT FUNCTION	BLOCKING SEGMENTS-ELLIPSOID
-1 - 1	3 - 1	1	0	
LARM	VEH -WIND PLANE	WIND FORCE		2- 2
-2 - 2	3 - 1	1	0	
UARM	VEH -WIND PLANE	WIND FORCE		1- 1

ZPLT(X)	ZPLT(Y)	ZPLT(Z)	I1	J1	I2	J2	I3	SPLT(1)	SPLT(2)	SPLT(3)
0.	0.	0.	0	0	0	0	1	10.00	6.00	1.00

INITIAL POSITIONS (INERTIAL REFERENCE)

CARDS G.2

SEGMENT NO. SEG	LINEAR POSITION (IN.)			LINEAR VELOCITY (IN./SEC.)		
	X	Y	Z	X	Y	Z
1 LARM	0.66000	0.00000	-18.98000	0.00000	0.00000	0.00000
2 UARM	0.18467	0.00000	-5.36830	0.00000	0.00000	0.00000

INITIAL ANGULAR ROTATION AND VELOCITY

CARDS G.3

SEGMENT NO. SEG	ANGULAR ROTATION (DEG)			ANGULAR VELOCITY (DEG/SEC.)						
	YAW	PITCH	ROLL	X	Y	Z	IYPR			
1 LARM	0.00000	-2.00000	0.00000	0.00000	0.00000	0.00000	3	2	1	0
2 UARM	0.00000	-2.00000	0.00000	0.00000	0.00000	0.00000	3	2	1	0

TABULAR TIME HISTORY CONTROL PARAMETERS

TYPE	KSG	SELECTED SEGMENTS OR JOINTS
H.1	0	
	REF	
H.2	0	
	REF	
H.3	3	1 2 2
	REF	2 1 4
H.4	0	
	REF	
H.5	0	
	REF	
H.6	0	
	REF	
H.7	1	1
	REF	0
H.8	2	1 2
	REF	0 0
H.9	1	1
	REF	1

SEGMENT	(INERTIAL) ANGULAR ROTATION (DEG)			(LOCAL) ANGULAR VELOCITY (RAD/SEC.)			(LOCAL) ANGULAR ACCELERATION (RAD/SEC.**2)		
	YAW	PITCH	ROLL	X	Y	Z	X	Y	Z
1 LARM	0.0000	-2.0000	0.0000	0.00000	0.00000	0.00000	0.000516	7.241005	-0.000467
2 UARM	0.0000	-2.0000	0.0000	0.00000	0.00000	0.00000	-0.000433	22.756386	0.000000
3 VEH	0.0000	0.0000	0.0000	0.00000	0.00000	0.00000	0.000000	0.000000	0.000000

SEGMENT	(INERTIAL) LINEAR POSITION (IN.)			(INERTIAL) LINEAR VELOCITY (IN./SEC.)			(INERTIAL) LINEAR ACCELERATIONS (G'S)		
	X	Y	Z	X	Y	Z	X	Y	Z
1 LARM	0.6600	0.0000	-18.9800	0.00000	0.00000	0.00000	0.123940	0.000002	1.323803
2 UARM	0.1847	0.0000	-5.3683	0.00000	0.00000	0.00000	-0.131968	-0.000003	22.212406
3 VEH	0.0000	0.0000	0.0000	0.00000	0.00000	12161.00000	0.000000	0.000000	0.000000

SEGMENT	(INERTIAL) U1 ARRAY (IN./SEC.**2) EXTERNAL LINEAR ACCELERATIONS			(LOCAL) U2 ARRAY (RAD/SEC.**2) EXTERNAL ANGULAR ACCELERATIONS			KINETIC ENERGY (LB.- IN.)		
	X	Y	Z	X	Y	Z	LINEAR	ANGULAR	TOTAL
1 LARM	0.0000E+00	0.0000E+00	0.5094E+03	0.85920E-03	-0.10774E+02	-0.46703E-03	0.00000E+00	0.00000E+00	0.00000E+00
2 UARM	0.0000E+00	0.0000E+00	0.8578E+04	-0.30574E-13	-0.46180E-14	-0.21433E-14	0.00000E+00	0.00000E+00	0.00000E+00
TOTAL BODY KINETIC ENERGY							0.00000E+00	0.00000E+00	0.00000E+00

JOINT	IPIN	(INERTIAL) JOINT FORCES (LB.)			(INERTIAL) JOINT TORQUES (IN. LB.)			RELATIVE ANGULAR VELOCITY (RAD/SEC.)
		X	Y	Z	X	Y	Z	
1 ELBW	5	-0.731E+00	-0.139E-04	-0.255E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.000

SEGMENT	(INERTIAL) ANGULAR ROTATION (DEG)			(LOCAL) ANGULAR VELOCITY (RAD/SEC.)			(LOCAL) ANGULAR ACCELERATION (RAD/SEC.**2)		
	YAW	PITCH	ROLL	X	Y	Z	X	Y	Z
	1 LARM	0.0000	-4.6324	0.0000	0.00002	-2.40848	0.00000	0.000697	-69.675443
2 UARM	0.0000	1.9906	0.0000	-0.00004	2.43012	0.00000	-0.000080	-49.821585	-0.000001
3 VEH	0.0000	0.0000	0.0000	0.00000	0.00000	0.00000	0.000000	0.000000	0.000000

SEGMENT	(INERTIAL) LINEAR POSITION (IN.)			(INERTIAL) LINEAR VELOCITY (IN./SEC.)			(INERTIAL) LINEAR ACCELERATIONS (G'S)		
	X	Y	Z	X	Y	Z	X	Y	Z
1 LARM	0.5275	0.0000	-13.8865	-7.63751	-0.00008	255.94533	-0.811909	-0.000008	17.016782
2 UARM	0.3258	0.0000	-3.6572	8.13242	0.00008	79.32710	0.864511	0.000009	3.974612
3 VEH	0.0000	0.0000	486.4400	0.00000	0.00000	12161.00000	0.000000	0.000000	0.000000

SEGMENT	(INERTIAL) U1 ARRAY (IN./SEC.**2) EXTERNAL LINEAR ACCELERATIONS			(LOCAL) U2 ARRAY (RAD/SEC.**2) EXTERNAL ANGULAR ACCELERATIONS			KINETIC ENERGY (LB.- IN.)		
	X	Y	Z	X	Y	Z	LINEAR	ANGULAR	TOTAL
1 LARM	0.2556E-02	0.2735E-12	0.6595E+04	0.87740E-07	0.83063E-12	-0.18158E-13	0.50106E+03	0.96612E+00	0.50203E+03
2 UARM	0.4615E-03	0.4939E-13	0.1508E+04	-0.15599E-02	0.98653E+02	-0.90086E-06	0.45639E+02	0.51466E+00	0.46153E+02
							TOTAL BODY KINETIC ENERGY		
							0.54670E+03	0.14808E+01	0.54818E+03

JOINT	IPIN	(INERTIAL) JOINT FORCES (LB.)			(INERTIAL) JOINT TORQUES (IN. LB.)			RELATIVE ANGULAR VELOCITY (RAD/SEC.)	
		X	Y	Z	X	Y	Z		
1	ELBW	5	0.479E+01	0.481E-04	0.388E+00	0.0000E+00	0.0000E+00	0.0000E+00	4.839

	HIC & HSI POINT	CSI POINT
H.11	0	0

RUN DESCRIPTION: EXAMPLE 2: DYNAMIC JOINT TEST

SLIP JOINT / 600 KNOT WIND

PAGE: 21.01

VEHICLE DECELERATION: CONSTANT WIND VELOCITY OF 600 KNOTS

CRASH VICTIM: 95TH PERCENTILE MALE

POINT REL. LINEAR DISPLACEMENT (IN.)

TIME (MSEC)	POINT (0.00, 0.00, 8.20) ON SEGMENT NO. 1 - LARM IN UARM REFERENCE				POINT (0.00, 0.00, -5.42) ON SEGMENT NO. 2 - UARM IN LARM REFERENCE				POINT (0.00, 0.00, 0.00) ON SEGMENT NO. 2 - UARM IN GRND REFERENCE			
	X	Y	Z	RES	X	Y	Z	RES	X	Y	Z	RES
0.000	0.000	0.000	-5.420	5.420	0.000	0.000	8.200	8.200	0.185	0.000	-5.368	5.371
0.500	0.000	0.000	-5.420	5.420	0.000	0.000	8.200	8.200	0.185	0.000	-5.368	5.371
1.000	0.000	0.000	-5.419	5.419	0.000	0.000	8.199	8.199	0.185	0.000	-5.366	5.370
1.500	0.000	0.000	-5.417	5.417	0.000	0.000	8.197	8.197	0.185	0.000	-5.365	5.368
2.000	0.000	0.000	-5.414	5.414	0.000	0.000	8.194	8.194	0.185	0.000	-5.362	5.365
3.000	0.000	0.000	-5.405	5.405	0.000	0.000	8.185	8.185	0.185	0.000	-5.356	5.359
4.000	0.000	0.000	-5.392	5.392	0.000	0.000	8.172	8.172	0.186	0.000	-5.347	5.350
5.000	0.000	0.000	-5.375	5.375	0.000	0.000	8.155	8.155	0.187	0.000	-5.336	5.339
6.000	0.000	0.000	-5.354	5.354	0.000	0.000	8.134	8.134	0.187	0.000	-5.322	5.325
7.000	0.000	0.000	-5.329	5.329	0.000	0.000	8.109	8.109	0.188	0.000	-5.306	5.309
8.000	-0.001	0.000	-5.299	5.299	0.000	0.000	8.079	8.079	0.190	0.000	-5.288	5.291
9.000	-0.001	0.000	-5.266	5.266	0.000	0.000	8.046	8.046	0.191	0.000	-5.267	5.271
10.000	-0.002	0.000	-5.228	5.228	0.000	0.000	8.008	8.008	0.192	0.000	-5.245	5.249
11.000	-0.002	0.000	-5.185	5.185	0.000	0.000	7.965	7.965	0.194	0.000	-5.221	5.224
12.000	-0.003	0.000	-5.139	5.139	0.000	0.000	7.919	7.919	0.196	0.000	-5.194	5.198
13.000	-0.005	0.000	-5.088	5.088	0.000	0.000	7.868	7.868	0.198	0.000	-5.166	5.170
14.000	-0.006	0.000	-5.032	5.032	0.000	0.000	7.812	7.812	0.200	0.000	-5.135	5.139
15.000	-0.008	0.000	-4.973	4.973	0.000	0.000	7.753	7.753	0.202	0.000	-5.103	5.107
16.000	-0.011	0.000	-4.909	4.909	0.000	0.000	7.689	7.689	0.204	0.000	-5.068	5.073
17.000	-0.013	0.000	-4.840	4.841	0.000	0.000	7.620	7.620	0.207	0.000	-5.032	5.036
18.000	-0.017	0.000	-4.768	4.768	0.000	0.000	7.548	7.548	0.210	0.000	-4.993	4.998
19.000	-0.021	0.000	-4.691	4.691	0.000	0.000	7.471	7.471	0.213	0.000	-4.953	4.957
20.000	-0.026	0.000	-4.610	4.610	0.000	0.000	7.390	7.390	0.216	0.000	-4.910	4.915
21.000	-0.031	0.000	-4.525	4.525	0.000	0.000	7.304	7.304	0.219	0.000	-4.865	4.870
22.000	-0.038	0.000	-4.435	4.435	0.000	0.000	7.214	7.214	0.223	0.000	-4.819	4.824
23.000	-0.045	0.000	-4.341	4.341	0.000	0.000	7.120	7.120	0.227	0.000	-4.770	4.775
24.000	-0.053	0.000	-4.243	4.243	0.000	0.000	7.021	7.021	0.231	0.000	-4.719	4.725
25.000	-0.063	0.000	-4.140	4.141	0.000	0.000	6.919	6.919	0.235	0.000	-4.667	4.672
26.000	-0.073	0.000	-4.033	4.034	0.000	0.000	6.811	6.811	0.239	0.000	-4.612	4.618
27.000	-0.085	0.000	-3.922	3.923	0.000	0.000	6.700	6.700	0.244	0.000	-4.555	4.562
28.000	-0.098	0.000	-3.807	3.808	0.000	0.000	6.584	6.584	0.248	0.000	-4.496	4.503
29.000	-0.113	0.000	-3.687	3.689	0.000	0.000	6.464	6.464	0.254	0.000	-4.436	4.443
30.000	-0.129	0.000	-3.563	3.566	0.000	0.000	6.339	6.339	0.259	0.000	-4.373	4.381
31.000	-0.146	0.000	-3.435	3.438	0.000	0.000	6.209	6.209	0.264	0.000	-4.309	4.317
32.000	-0.165	0.000	-3.302	3.306	0.000	0.000	6.075	6.075	0.270	0.000	-4.243	4.252
33.000	-0.187	0.000	-3.164	3.170	0.000	0.000	5.937	5.937	0.276	0.000	-4.175	4.184
34.000	-0.209	0.000	-3.022	3.030	0.000	0.000	5.793	5.793	0.282	0.000	-4.106	4.115
35.000	-0.234	0.000	-2.876	2.885	0.000	0.000	5.645	5.645	0.289	0.000	-4.035	4.045
36.000	-0.261	0.000	-2.724	2.737	0.000	0.000	5.491	5.491	0.296	0.000	-3.962	3.973
37.000	-0.290	0.000	-2.568	2.584	0.000	0.000	5.333	5.333	0.303	0.000	-3.888	3.900
38.000	-0.320	0.000	-2.407	2.428	0.000	0.000	5.170	5.170	0.310	0.000	-3.813	3.825
39.000	-0.354	0.000	-2.241	2.269	0.000	0.000	5.002	5.002	0.318	0.000	-3.736	3.749
40.000	-0.389	0.000	-2.071	2.107	0.000	0.000	4.828	4.828	0.326	0.000	-3.657	3.672

DATE: FEB. 4, 1988
 RUN DESCRIPTION: EXAMPLE 2: DYNAMIC JOINT TEST
 SLIP JOINT / 600 KNOT WIND
 VEHICLE DECELERATION: CONSTANT WIND VELOCITY OF 600 KNOTS
 CRASH VICTIM: 95TH PERCENTILE MALE

PAGE: 22.01

JOINT PARAMETERS

JOINT NO. 1 - ELBW

TIME (MSEC)	STATE IPIN	JOINT ANGLES (DEG)			TOTAL TORQUE (IN. LB.)		
		FLEXURE	AZIMUTH	TORSION	SPRING	VISCOUS	RES.
0.000	5.	67.190	180.000	0.000	0.000	0.000	0.000
0.500	5.	67.189	180.000	0.000	0.000	0.000	0.000
1.000	5.	67.186	180.000	0.000	0.000	0.000	0.000
1.500	5.	67.181	180.000	0.000	0.000	0.000	0.000
2.000	5.	67.172	180.000	0.000	0.000	0.000	0.000
3.000	5.	67.149	180.000	0.000	0.000	0.000	0.000
4.000	5.	67.116	180.000	0.000	0.000	0.000	0.000
5.000	5.	67.073	180.000	0.000	0.000	0.000	0.000
6.000	5.	67.021	180.000	0.000	0.000	0.000	0.000
7.000	5.	66.959	180.000	0.000	0.000	0.000	0.000
8.000	5.	66.888	180.000	0.000	0.000	0.000	0.000
9.000	5.	66.808	180.000	0.000	0.000	0.000	0.000
10.000	5.	66.720	180.000	0.000	0.000	0.000	0.000
11.000	5.	66.623	180.000	0.000	0.000	0.000	0.000
12.000	5.	66.517	180.000	0.000	0.000	0.000	0.000
13.000	5.	66.403	180.000	0.000	0.000	0.000	0.000
14.000	5.	66.280	180.000	0.000	0.000	0.000	0.000
15.000	5.	66.149	180.000	0.000	0.000	0.000	0.000
16.000	5.	66.010	180.000	0.000	0.000	0.000	0.000
17.000	5.	65.862	180.000	0.000	0.000	0.000	0.000
18.000	5.	65.706	180.000	0.000	0.000	0.000	0.000
19.000	5.	65.542	180.000	0.000	0.000	0.000	0.000
20.000	5.	65.369	180.000	0.000	0.000	0.000	0.000
21.000	5.	65.189	180.000	0.000	0.000	0.000	0.000
22.000	5.	65.001	180.000	0.000	0.000	0.000	0.000
23.000	5.	64.805	180.000	0.000	0.000	0.000	0.000
24.000	5.	64.601	180.000	0.000	0.000	0.000	0.000
25.000	5.	64.390	-180.000	0.000	0.000	0.000	0.000
26.000	5.	64.171	-180.000	0.000	0.000	0.000	0.000
27.000	5.	63.945	-180.000	0.000	0.000	0.000	0.000
28.000	5.	63.712	180.000	0.000	0.000	0.000	0.000
29.000	5.	63.473	180.000	0.000	0.000	0.000	0.000
30.000	5.	63.229	180.000	0.000	0.000	0.000	0.000
31.000	5.	62.978	180.000	0.000	0.000	0.000	0.000
32.000	5.	62.723	180.000	0.000	0.000	0.000	0.000
33.000	5.	62.462	180.000	0.000	0.000	0.000	0.000
34.000	5.	62.199	180.000	0.000	0.000	0.000	0.000
35.000	5.	61.932	180.000	0.000	0.000	0.000	0.000
36.000	5.	61.663	180.000	0.000	0.000	0.000	0.000
37.000	5.	61.392	180.000	0.000	0.000	0.000	0.000
38.000	5.	61.119	180.000	0.000	0.000	0.000	0.000
39.000	5.	60.844	180.000	0.000	0.000	0.000	0.000
40.000	5.	60.567	180.000	0.000	0.000	0.000	0.000

RUN DESCRIPTION: EXAMPLE 2: DYNAMIC JOINT TEST

SLIP JOINT / 600 KNOT WIND

PAGE: 23.01

VEHICLE DECELERATION: CONSTANT WIND VELOCITY OF 600 KNOTS

CRASH VICTIM: 95TH PERCENTILE MALE

SEGMENT WIND FORCE (LB.)

TIME (MSEC)	SEGMENT NO. 1 - LARM				SEGMENT NO. 2 - UARM			
	IN GRND REFERENCE				IN GRND REFERENCE			
	X	Y	Z	RES	X	Y	Z	RES
0.000	0.000	0.000	1.885	1.885	0.000	0.000	117.585	117.585
0.500	0.000	0.000	92.335	92.335	0.000	0.000	28.783	28.783
1.000	0.000	0.000	92.288	92.288	0.000	0.000	28.768	28.768
1.500	0.000	0.000	92.242	92.242	0.000	0.000	28.752	28.752
2.000	0.000	0.000	92.198	92.198	0.000	0.000	28.735	28.735
3.000	0.000	0.000	92.112	92.112	0.000	0.000	28.702	28.702
4.000	0.000	0.000	92.032	92.032	0.000	0.000	28.667	28.667
5.000	0.000	0.000	91.958	91.958	0.000	0.000	28.632	28.632
6.000	0.000	0.000	91.888	91.888	0.000	0.000	28.595	28.595
7.000	0.000	0.000	91.824	91.824	0.000	0.000	23.798	23.798
8.000	0.000	0.000	91.766	91.766	0.000	0.000	23.765	23.765
9.000	0.000	0.000	91.713	91.713	0.000	0.000	23.733	23.733
10.000	0.000	0.000	91.666	91.666	0.000	0.000	23.699	23.699
11.000	0.000	0.000	91.624	91.624	0.000	0.000	23.666	23.666
12.000	0.000	0.000	91.589	91.589	0.000	0.000	23.632	23.632
13.000	0.000	0.000	91.560	91.560	0.000	0.000	23.597	23.597
14.000	0.000	0.000	91.538	91.538	0.000	0.000	23.563	23.563
15.000	0.000	0.000	91.523	91.523	0.000	0.000	23.529	23.529
16.000	0.000	0.000	91.515	91.515	0.000	0.000	23.494	23.494
17.000	0.000	0.000	91.515	91.515	0.000	0.000	23.460	23.460
18.000	0.000	0.000	91.524	91.524	0.000	0.000	23.427	23.427
19.000	0.000	0.000	91.541	91.541	0.000	0.000	23.394	23.394
20.000	0.000	0.000	91.567	91.567	0.000	0.000	23.361	23.361
21.000	0.000	0.000	91.602	91.602	0.000	0.000	23.330	23.330
22.000	0.000	0.000	91.648	91.648	0.000	0.000	23.299	23.299
23.000	0.000	0.000	91.704	91.704	0.000	0.000	23.269	23.269
24.000	0.000	0.000	91.772	91.772	0.000	0.000	23.240	23.240
25.000	0.000	0.000	91.852	91.852	0.000	0.000	23.213	23.213
26.000	0.000	0.000	91.944	91.944	0.000	0.000	23.187	23.187
27.000	0.000	0.000	92.050	92.050	0.000	0.000	23.162	23.162
28.000	0.000	0.000	92.169	92.169	0.000	0.000	20.825	20.825
29.000	0.000	0.000	92.303	92.303	0.000	0.000	20.806	20.806
30.000	0.000	0.000	92.452	92.452	0.000	0.000	20.789	20.789
31.000	0.000	0.000	92.616	92.616	0.000	0.000	20.773	20.773
32.000	0.000	0.000	92.798	92.798	0.000	0.000	20.759	20.759
33.000	0.000	0.000	92.996	92.996	0.000	0.000	16.136	16.136
34.000	0.000	0.000	93.211	93.211	0.000	0.000	16.127	16.127
35.000	0.000	0.000	93.444	93.444	0.000	0.000	16.120	16.120
36.000	0.000	0.000	93.695	93.695	0.000	0.000	16.113	16.113
37.000	0.000	0.000	93.966	93.966	0.000	0.000	16.108	16.108
38.000	0.000	0.000	94.257	94.257	0.000	0.000	16.103	16.103
39.000	0.000	0.000	94.569	94.569	0.000	0.000	16.100	16.100
40.000	0.000	0.000	94.903	94.903	0.000	0.000	16.097	16.097

DATE: FEB. 4, 1988

RUN DESCRIPTION: EXAMPLE 2: DYNAMIC JOINT TEST

SLIP JOINT / 600 KNOT WIND

VEHICLE DECELERATION: CONSTANT WIND VELOCITY OF 600 KNOTS

CRASH VICTIM: 95TH PERCENTILE MALE

ELBW JOINT FORCES & TORQUES ON UARM IN LARM REFERENCE

TIME (MSEC)	JOINT FORCE (LB. 10**2)			JOINT TORQUE (IN.- LB. 10**2)		
	X	Y	Z	X	Y	Z
0.000	-0.007	0.000	0.000	0.000E+00	0.000E+00	0.000E+00
0.500	0.022	0.000	0.000	0.000E+00	0.000E+00	0.000E+00
1.000	0.022	0.000	0.000	0.000E+00	0.000E+00	0.000E+00
1.500	0.022	0.000	0.000	0.000E+00	0.000E+00	0.000E+00
2.000	0.023	0.000	0.000	0.000E+00	0.000E+00	0.000E+00
3.000	0.023	0.000	0.000	0.000E+00	0.000E+00	0.000E+00
4.000	0.023	0.000	0.000	0.000E+00	0.000E+00	0.000E+00
5.000	0.023	0.000	0.000	0.000E+00	0.000E+00	0.000E+00
6.000	0.023	0.000	0.000	0.000E+00	0.000E+00	0.000E+00
7.000	0.022	0.000	0.000	0.000E+00	0.000E+00	0.000E+00
8.000	0.022	0.000	0.000	0.000E+00	0.000E+00	0.000E+00
9.000	0.022	0.000	0.000	0.000E+00	0.000E+00	0.000E+00
10.000	0.022	0.000	0.000	0.000E+00	0.000E+00	0.000E+00
11.000	0.023	0.000	0.000	0.000E+00	0.000E+00	0.000E+00
12.000	0.023	0.000	0.000	0.000E+00	0.000E+00	0.000E+00
13.000	0.023	0.000	0.000	0.000E+00	0.000E+00	0.000E+00
14.000	0.024	0.000	0.000	0.000E+00	0.000E+00	0.000E+00
15.000	0.024	0.000	0.000	0.000E+00	0.000E+00	0.000E+00
16.000	0.025	0.000	0.000	0.000E+00	0.000E+00	0.000E+00
17.000	0.025	0.000	0.000	0.000E+00	0.000E+00	0.000E+00
18.000	0.026	0.000	0.000	0.000E+00	0.000E+00	0.000E+00
19.000	0.027	0.000	0.000	0.000E+00	0.000E+00	0.000E+00
20.000	0.027	0.000	0.000	0.000E+00	0.000E+00	0.000E+00
21.000	0.028	0.000	0.000	0.000E+00	0.000E+00	0.000E+00
22.000	0.029	0.000	0.000	0.000E+00	0.000E+00	0.000E+00
23.000	0.030	0.000	0.000	0.000E+00	0.000E+00	0.000E+00
24.000	0.030	0.000	0.000	0.000E+00	0.000E+00	0.000E+00
25.000	0.031	0.000	0.000	0.000E+00	0.000E+00	0.000E+00
26.000	0.033	0.000	0.000	0.000E+00	0.000E+00	0.000E+00
27.000	0.034	0.000	0.000	0.000E+00	0.000E+00	0.000E+00
28.000	0.032	0.000	0.000	0.000E+00	0.000E+00	0.000E+00
29.000	0.034	0.000	0.000	0.000E+00	0.000E+00	0.000E+00
30.000	0.035	0.000	0.000	0.000E+00	0.000E+00	0.000E+00
31.000	0.036	0.000	0.000	0.000E+00	0.000E+00	0.000E+00
32.000	0.038	0.000	0.000	0.000E+00	0.000E+00	0.000E+00
33.000	0.034	0.000	0.000	0.000E+00	0.000E+00	0.000E+00
34.000	0.036	0.000	0.000	0.000E+00	0.000E+00	0.000E+00
35.000	0.038	0.000	0.000	0.000E+00	0.000E+00	0.000E+00
36.000	0.040	0.000	0.000	0.000E+00	0.000E+00	0.000E+00
37.000	0.042	0.000	0.000	0.000E+00	0.000E+00	0.000E+00
38.000	0.044	0.000	0.000	0.000E+00	0.000E+00	0.000E+00
39.000	0.046	0.000	0.000	0.000E+00	0.000E+00	0.000E+00
40.000	0.048	0.000	0.000	0.000E+00	0.000E+00	0.000E+00

ELAPSED CPU TIME = 13.51 SECONDS

PAGE 20

SUB	CALLS	TIME	%
MAIN3D	1	97	7.18
INPUT	1	49	3.63
CHAIN	170	18	1.33
EJOINT	170	0	0.00
DINT	21	77	5.70
PDAUX	211	71	5.26
DAUX	169	47	3.48
SETUP1	169	17	1.26
CONTC	169	17	1.26
WINDY	338	684	50.63
VISPR	169	22	1.63
SETUP2	169	0	0.00
DAUX11	169	12	0.89
DAUX12	169	6	0.44
DAUX22	169	0	0.00
FSMSOL	169	11	0.81
OUTPUT	43	90	6.66
UPDATE	42	0	0.00
DZP	168	28	2.07
POSTPR	1	105	7.77
TOTAL		1351	100.00

APPENDIX D4

Example 2, ATB Dynamic Joint Test
Perkin-Elmer Output File

DEVELOPED BY CALSPAN CORP., P.O. BOX 400, BUFFALO NY 14225
AND BY J&J TECHNOLOGIES INC., ORCHARD PARK, NY 14127

FOR THE AIR FORCE ARMSTRONG AEROSPACE MEDICAL RESEARCH
LABORATORY, WRIGHT PATTERSON AIR FORCE BASE
UNDER CONTRACTS F33615-75C-5002, -78C-0516 AND -80C-05117

AND FOR THE NATIONAL HIGHWAY TRAFFIC SAFETY ADMINISTRATION,
U.S. DEPARTMENT OF TRANSPORTATION, UNDER CONTRACTS
FH-11-7592, HS-053-2-485, HS-6-01300 AND "S-6-01410.

PROGRAM DOCUMENTATION: NHTSA REPORT NOS. DOT-HS-801-507
THROUGH 510 (FORMERLY CALSPAN REPORT NO. 20-5700-L-1),
AVAILABLE FROM NTIS (ACCESSION NOS. PB-241692, 3, 4 AND 5),
APPENDIXES A-J TO THE ABOVE (AVAILABLE FROM CALSPAN),
AND REPORT NOS. AMRL-TR-75-14 (NTIS NO. AD-A014 816),
AFAMRL-TR-80-14 (NTIS NO. AD-A088 029), AND
AFAMRL-TR-83-073 (NTIS NO. AD-B079 184).

PROGRAM ATB-IV, EXECUTED ON THE AAMRL/BB CONCURRENT
3250 COMPUTER, WRIGHT-PATTERSON AFB, OHIO

FEB. 4, 1988 IRSIN= 0 IRSOUT= 0 RSTIME = 0.0000

CARDS A

EXAMPLE 2: DYNAMIC JOINT TEST
SLIP JOINT / 600 KNOT WIND

UNITL = IN. UNITM = LB. UNITT = SEC. GRAVITY VECTOR = (0.0000, 0.0000, 386.0880) G = 386.0880
NDINT = 4 NSTEPS = 20 DT = 0.002000 HO = 0.000500 HMAX = 0.001000 HMIN = 0.000063

NPRT ARRAY

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
1	0	20	2	0	0	0	0	0	0	0	0	0	0	0	0	0	16	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	1

CARD B.1

CARDS B.2

SEGMENT I SYM PLOT	WEIGHT (LB.)	PRINCIPAL MOMENTS OF INERTIA (LB.-SEC.**2- IN.)			SEGMENT CONTACT ELLIPSOID						PRINCIPAL AXES (DEG)		
		X	Y	Z	SEMIAXES (IN.)			CENTER (IN.)			YAW	PITCH	ROLL
					X	Y	Z	X	Y	Z			
1 LARM L	5.901	0.3331	0.3331	0.0214	1.871	1.871	10.269	0.000	0.000	0.000	0.00	0.00	0.00
2 UARM U	5.542	0.1743	0.1743	0.0259	2.122	2.122	7.497	0.000	0.000	0.000	0.00	0.00	0.00

CARDS B.3

JOINT J SYM PLOT	JNT PIN	LOCATION(IN.) - SEG(JNT)			LOCATION(IN.) - SEG(J+1)			PRIN. AXIS(DEG) - SEG(JNT)			PRIN. AXIS(DEG) - SEG(J+1)		
		X	Y	Z	X	Y	Z	YAW	PITCH	ROLL	YAW	PITCH	ROLL
1 ELBW E	1 5	0.000	0.000	8.200	0.000	0.000	-5.420	0.00	0.00	0.00	0.00	-67.19	0.00

UNLOCK CONDITIONS FOR SLIP JOINTS

JOINT	TENSION (LB.)	COMPRESSION (LB.)
1	0.000	0.000

FLEXURAL SPRING CHARACTERISTICS

TORSIONAL SPRING CHARACTERISTICS

JOINT	SPRING COEF. (IN. LB./DEG**J)			ENERGY DISSIPATION COEF.	JOINT STOP (DEG)	SPRING COEF. (IN. LB./DEG**J)			ENERGY DISSIPATION COEF.	JOINT STOP (DEG)
	LINEAR (J=1)	QUADRATIC (J=2)	CUBIC (J=3)			LINEAR (J=1)	QUADRATIC (J=2)	CUBIC (J=3)		
1 ELBW	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

CARDS B.5

JOINT VISCOUS CHARACTERISTICS AND LOCK-UNLOCK CONDITIONS

JOINT	VISCOUS COEFFICIENT (IN. LB.SEC./DEG)	COULOMB FRICTION COEF. (IN. LB.)	FULL FRICTION ANGULAR VELOCITY (DEG/SEC.)	MAX TORQUE FOR A LOCKED JOINT (IN. LB.)	MIN TORQUE FOR UNLOCKED JOINT (IN. LB.)	MIN. ANG. VELOCITY FOR UNLOCKED JOINT (RAD/SEC.)	IMPULSE RESTITUTION COEFFICIENT
1 ELBW	0.000	0.00	30.00	0.00	0.00	0.00	0.000

SEGMENT INTEGRATION CONVERGENCE TEST INPUT

PAGE 4
CARDS B.6

SEGMENT NO. SYM	ANGULAR VELOCITIES (RAD/SEC.)			LINEAR VELOCITIES (IN./SEC.)			ANGULAR ACCELERATIONS (RAD/SEC.**2)			LINEAR ACCELERATIONS (IN./SEC.**2)		
	MAG. TEST	ABS. ERROR	REL. ERROR	MAG. TEST	ABS. ERROR	REL. ERROR	MAG. TEST	ABS. ERROR	REL. ERROR	MAG. TEST	ABS. ERROR	REL. ERROR
1 LARM	0.000	0.000	0.0000	0.000	0.000	0.0000	0.010	0.010	0.0010	0.001	0.001	0.0010
2 UARM	0.000	0.000	0.0000	0.000	0.000	0.0000	0.010	0.010	0.0010	0.000	0.000	0.0000

CONSTANT WIND VELOCITY OF 600 KNOTS

YAW	PITCH	ROLL	VIPS	VTIME	XO(X)	XO(Y)	XO(Z)	NATAB	ATO	ADT	MSEG
0.000	0.000	0.000	12161.000	0.000	0.000	0.000	0.000	-3	0.000000	2.300000	0

SPLINE FIT TABULAR INPUT

LTYPE = 2 LFIT = 1 NPTS = 3

TIME(SEC.)=	INITIAL LINEAR POSITION (IN.)			INITIAL ANGULAR POSITION (DEG)		
	X=	Y=	Z=	X=	Y=	Z=
0.0000	0.000	0.000	12161.000	0.000	0.000	0.000

TIME(SEC.)	LINEAR VELOCITY (IN./SEC.)			ANGULAR VELOCITY (DEG/SEC.)		
	X	Y	Z	X	Y	Z
0.00000	0.000	0.000	12161.000	0.000	0.000	0.000
2.30000	0.000	0.000	12161.000	0.000	0.000	0.000
4.60000	0.000	0.000	12161.000	0.000	0.000	0.000

VEHICLE LINEAR TIME HISTORY CONSTANT WIND VELOCITY OF 600 KNOTS

PAGE NO. 1

TIME (MSEC)	LINEAR DECELERATIONS (G'S)			LINEAR VELOCITIES (IN./SEC.)			LINEAR DISPLACEMENTS (IN.)		
	X	Y	Z	X	Y	Z	X	Y	Z
0.000	0.000	0.000	0.000	0.000	0.000	12161.000	0.000	0.000	0.000
2300.000	0.000	0.000	0.000	0.000	0.000	12161.000	0.000	0.000	27970.300
4600.000	0.000	0.000	0.000	0.000	0.000	12161.000	0.000	0.000	55940.600

NPL NBLT NBAG NBLP NQ NSD NHRNSS NWINDF NJNTF NFORCE
1 0 0 0 0 0 0 1 0 0

PAGE 8
CARD D.1

PLANE INPUTS

CARDS D.2

PLANE NO. 1 WIND PLANE

	X	Y	Z
POINT 1	10.0000	10.0000	5.0000
POINT 2	-10.0000	10.0000	5.0000
POINT 3	10.0000	-10.0000	5.0000

BODY SEGMENT SYMMETRY INPUT

CARD D.7

SEG NO. 1 2

NSYM(J) 0 0

WIND FORCE FUNCTION NO. 1 WIND FORCE

NTI(1) = 1

PAGE 9
CARDS E.6

SPEC. HEAT RATIO	SONIC VEL.	ABS. PRESS.	SECMENT	REF. SEGMENT
1.4000	13404.0000	14.6960	1LARM	3VEH

SEGMENT-ELLIPSOID	SEGMENT-PLANE	WIND FORCE FUNCTION	DRAG COEFFICIENT FUNCTION	BLOCKING SEGMENTS-ELLIPSOID
-1 - 1 LARM	3 - 1 VEH -WIND PLANE	1 WIND FORCE	0	2- 2
-2 - 2 UARM	3 - 1 VEH -WIND PLANE	1 WIND FORCE	0	1- 1

ZPLT(X)	ZPLT(Y)	ZPLT(Z)	I1	J1	I2	J2	I3	SPLT(1)	SPLT(2)	SPLT(3)
0.	0.	0.	0	0	0	0	1	10.00	6.00	1.00

INITIAL POSITIONS (INERTIAL REFERENCE)

CARDS G.2

SEGMENT NO. SEG	LINEAR POSITION (IN.)			LINEAR VELOCITY (IN./SEC.)		
	X	Y	Z	X	Y	Z
1 LARM	0.66000	0.00000	-18.98000	0.00000	0.00000	0.00000
2 UARM	0.18467	0.00000	-5.36830	0.00000	0.00000	0.00000

INITIAL ANGULAR ROTATION AND VELOCITY

CARDS G.3

SEGMENT NO. SEG	ANGULAR ROTATION (DEG)			ANGULAR VELOCITY (DEG/SEC.)						
	YAW	PITCH	ROLL	X	Y	Z	IYPR			
1 LARM	0.00000	-2.00000	0.00000	0.00000	0.00000	0.00000	3	2	1	0
2 UARM	0.00000	-2.00000	0.00000	0.00000	0.00000	0.00000	3	2	1	0

TABULAR TIME HISTORY CONTROL PARAMETERS

TYPE KSG SELECTED SEGMENTS OR JOINTS

H.1	0	
REF		
H.2	0	
REF		
H.3	3	1 2 2
REF		2 1 4
H.4	0	
REF		
H.5	0	
REF		
H.6	0	
REF		
H.7	1	1
REF		0
H.8	2	1 2
REF		0 0
H.9	1	1
REF		1

SEGMENT	(INERTIAL) ANGULAR ROTATION (DEG)			(LOCAL) ANGULAR VELOCITY (RAD/SEC.)			(LOCAL) ANGULAR ACCELERATION (RAD/SEC.**2)		
	YAW	PITCH	ROLL	X	Y	Z	X	Y	Z
	1 LARM	0.0000	-2.0000	0.0000	0.00000	0.00000	0.00000	0.000348	7.240952
2 UARM	0.0000	-2.0000	0.0000	0.00000	0.00000	0.00000	-0.000292	22.756331	0.000000
3 VEH	0.0000	0.0000	0.0000	0.00000	0.00000	0.00000	0.000000	0.000000	0.000000

SEGMENT	(INERTIAL) LINEAR POSITION (IN.)			(INERTIAL) LINEAR VELOCITY (IN./SEC.)			(INERTIAL) LINEAR ACCELERATIONS (G'S)		
	X	Y	Z	X	Y	Z	X	Y	Z
	1 LARM	0.6600	0.0000	-18.9800	0.00000	0.00000	0.00000	0.123940	0.000002
2 UARM	0.1847	0.0000	-5.3683	0.00000	0.00000	0.00000	-0.131958	-0.000002	22.212334
3 VEH	0.0000	0.0000	0.0000	0.00000	0.00000	12161.00000	0.000000	0.000000	0.000000

SEGMENT	(INERTIAL) U1 ARRAY (IN./SEC.**2) EXTERNAL LINEAR ACCELERATIONS			(LOCAL) U2 ARRAY (RAD/SEC.**2) EXTERNAL ANGULAR ACCELERATIONS			KINETIC ENERGY (LB.- IN.)		
	X	Y	Z	X	Y	Z	LINEAR	ANGULAR	TOTAL
	1 LARM	0.00000+00	0.00000+00	0.50940+03	0.57981D-03	-0.10774D+02	-0.31516D-03	0.000000+00	0.000000+00
2 UARM	0.00000+00	0.00000+00	0.85780+04	-0.11465D-13	-0.38367D-14	-0.32149D-14	0.000000+00	0.000000+00	0.000000+00
TOTAL BODY KINETIC ENERGY							0.000000+00	0.000000+00	0.000000+00

JOINT	IPIN	(INERTIAL) JOINT FORCES (LB.)			(INERTIAL) JOINT TORQUES (IN. LB.)			RELATIVE ANGULAR VELOCITY (RAD/SEC.)		
		X	Y	Z	X	Y	Z			
		1 ELBW	5	-0.731D+00	-0.941D-05	-0.255D-01	0.00000+00	0.00000+00	0.00000+00	0.000
DINT CONV. TEST		1.000	LARM	ANG ACC	3064.	0.4514E-02	0.1473E-05	0.1000E-03	0.1000E-03	0.1000E-05

TEST FAILED AT TIME = 0.001000 FOR H = 0.000500

SEGMENT	(INERTIAL) ANGULAR ROTATION (DEG)			(LOCAL) ANGULAR VELOCITY (RAD/SEC.)			(LOCAL) ANGULAR ACCELERATION (RAD/SEC.**2)		
	YAW	PITCH	ROLL	X	Y	Z	X	Y	Z
	1 LARM	0.0000	-4.6312	-0.0023	-0.00364	-2.40686	0.00000	-0.105497	-69.644370
2 UARM	0.0000	1.9884	0.0050	0.00666	2.42783	-0.00004	0.012320	-49.752471	-0.000001
3 VEK	0.0000	0.0000	0.0000	0.00000	0.00000	0.00000	0.000000	0.000000	0.000000

SEGMENT	(INERTIAL) LINEAR POSITION (IN.)			(INERTIAL) LINEAR VELOCITY (IN./SEC.)			(INERTIAL) LINEAR ACCELERATIONS (G'S)		
	X	Y	Z	X	Y	Z	X	Y	Z
1 LARM	0.5275	0.0001	-13.8864	-7.63222	0.01244	255.94443	-0.811510	0.001280	17.016222
2 UARM	0.3257	-0.0001	-3.6569	8.12663	-0.01319	79.35888	0.864086	-0.001363	3.974519
3 VEK	0.0000	0.0000	486.4400	0.00000	0.00000	12161.00000	0.000000	0.000000	0.000000

SEGMENT	(INERTIAL) U1 ARRAY (IN./SEC.**2) EXTERNAL LINEAR ACCELERATIONS			(LOCAL) U2 ARRAY (RAD/SEC.**2) EXTERNAL ANGULAR ACCELERATIONS			KINETIC ENERGY (LB.- IN.)		
	X	Y	Z	X	Y	Z	LINEAR	ANGULAR	TOTAL
1 LARM	0.25520-02	-0.67750-08	0.65950+04	0.591660-07	-0.894470-10	0.110240-13	0.501060+03	0.964830+00	0.502020+03
2 UARM	0.46090-03	-0.12240-08	0.15070+04	0.245770+00	0.986500+02	-0.883560-06	0.456740+02	0.513700+00	0.461880+02
							TOTAL BODY KINETIC ENERGY		
							0.546730+03	0.147850+01	0.548210+03

JOINT	IPIN	(INERTIAL) JOINT FORCES (LB.)			(INERTIAL) JOINT TORQUES (IN. LB.)			RELATIVE ANGULAR VELOCITY (RAD/SEC.)	
		X	Y	Z	X	Y	Z		
1	ELBW	5	0.4790+01	-0.7550-02	0.3880+00	0.00000+00	0.00000+00	0.00000+00	4.835

POSTPROCESSOR CONTROL PARAMETERS

	NIC & NSI POINT	CSI POINT
N.11	0	0

DATE: FEB. 4, 1988
 RUN DESCRIPTION: EXAMPLE 2: DYNAMIC JOINT TEST
 SLIP JOINT / 600 KNOT WIND
 VEHICLE DECELERATION: CONSTANT WIND VELOCITY OF 600 KNOTS
 CRASH VICTIM: 95TH PERCENTILE MALE

PAGE: 21.01

POINT REL. LINEAR DISPLACEMENT (IN.)

TIME (MSEC)	POINT (0.00, 0.00, 8.20) ON SEGMENT NO. 1 - LARM IN UARM REFERENCE				POINT (0.00, 0.00, -5.42) ON SEGMENT NO. 2 - UARM IN LARM REFERENCE				POINT (0.00, 0.00, 0.00) ON SEGMENT NO. 2 - UARM IN GRND REFERENCE			
	X	Y	Z	RES	X	Y	Z	RES	X	Y	Z	RES
0.000	0.000	0.000	-5.420	5.420	0.000	0.000	8.200	8.200	0.185	0.000	-5.368	5.371
0.500	0.000	0.000	-5.420	5.420	0.000	0.000	8.200	8.200	0.185	0.000	-5.368	5.371
0.750	0.000	0.000	-5.420	5.420	0.000	0.000	8.200	8.200	0.185	0.000	-5.367	5.370
1.000	0.000	0.000	-5.419	5.419	0.000	0.000	8.199	8.199	0.185	0.000	-5.366	5.370
1.500	0.000	0.000	-5.417	5.417	0.000	0.000	8.197	8.197	0.185	0.000	-5.365	5.368
2.000	0.000	0.000	-5.414	5.414	0.000	0.000	8.194	8.194	0.185	0.000	-5.362	5.366
3.000	0.000	0.000	-5.405	5.405	0.000	0.000	8.185	8.185	0.185	0.000	-5.356	5.359
4.000	0.000	0.000	-5.392	5.392	0.000	0.000	8.172	8.172	0.186	0.000	-5.347	5.350
5.000	0.000	0.000	-5.375	5.375	0.000	0.000	8.155	8.155	0.187	0.000	-5.336	5.339
6.000	0.000	0.000	-5.354	5.354	0.000	0.000	8.134	8.134	0.187	0.000	-5.322	5.325
7.000	0.000	0.000	-5.329	5.329	0.000	0.200	8.109	8.109	0.188	0.000	-5.306	5.309
8.000	-0.001	0.000	-5.299	5.299	0.000	0.000	8.079	8.079	0.190	0.000	-5.288	5.291
9.000	-0.001	0.000	-5.266	5.266	0.000	0.000	8.046	8.046	0.191	0.000	-5.268	5.271
10.000	-0.002	0.000	-5.228	5.228	0.000	0.000	8.008	8.008	0.192	0.000	-5.245	5.249
11.000	-0.002	0.000	-5.185	5.185	0.000	0.000	7.965	7.965	0.194	0.000	-5.221	5.224
12.000	-0.003	0.000	-5.138	5.138	0.000	0.000	7.918	7.918	0.196	0.000	-5.194	5.198
13.000	-0.005	0.000	-5.087	5.087	0.000	0.000	7.867	7.867	0.198	0.000	-5.166	5.170
14.000	-0.006	0.000	-5.032	5.032	0.000	0.000	7.812	7.812	0.200	0.000	-5.136	5.139
15.000	-0.008	0.000	-4.973	4.973	0.000	0.000	7.752	7.752	0.202	0.000	-5.103	5.107
16.000	-0.011	0.000	-4.909	4.909	0.000	0.000	7.688	7.688	0.204	0.000	-5.069	5.073
17.000	-0.013	0.000	-4.840	4.840	0.000	0.000	7.620	7.620	0.207	0.000	-5.032	5.036
18.000	-0.017	0.000	-4.768	4.768	0.000	0.000	7.548	7.548	0.210	0.000	-4.993	4.998
19.000	-0.021	0.000	-4.691	4.691	0.000	0.000	7.471	7.471	0.213	0.000	-4.953	4.957
20.000	-0.026	0.000	-4.610	4.610	0.000	0.000	7.389	7.389	0.216	0.000	-4.910	4.915
21.000	-0.031	0.000	-4.524	4.524	0.000	0.000	7.304	7.304	0.219	0.000	-4.865	4.870
22.000	-0.038	0.000	-4.435	4.435	0.000	0.000	7.214	7.214	0.223	0.000	-4.819	4.824
23.000	-0.045	0.000	-4.341	4.341	0.000	0.000	7.120	7.120	0.227	0.000	-4.770	4.775
24.000	-0.053	0.000	-4.242	4.243	0.000	0.000	7.021	7.021	0.231	0.000	-4.719	4.725
25.000	-0.063	0.000	-4.140	4.140	0.000	0.000	6.918	6.918	0.235	0.000	-4.667	4.673
26.000	-0.073	0.000	-4.033	4.034	0.000	0.000	6.811	6.811	0.239	0.000	-4.612	4.618
27.000	-0.085	0.000	-3.922	3.923	0.000	0.000	6.700	6.700	0.244	0.000	-4.555	4.562
28.000	-0.098	0.000	-3.807	3.808	0.000	0.000	6.584	6.584	0.248	0.000	-4.496	4.503
29.000	-0.113	0.000	-3.687	3.689	0.000	0.000	6.464	6.464	0.254	0.000	-4.436	4.443
30.000	-0.129	0.000	-3.563	3.565	0.000	0.000	6.339	6.339	0.259	0.000	-4.373	4.381
31.000	-0.146	0.000	-3.435	3.438	0.000	0.000	6.209	6.209	0.264	0.000	-4.309	4.317
32.000	-0.165	0.000	-3.302	3.306	0.000	0.000	6.075	6.075	0.270	0.000	-4.243	4.251
33.000	-0.186	0.000	-3.164	3.170	0.000	0.000	5.937	5.937	0.276	0.000	-4.175	4.184
34.000	-0.209	0.000	-3.022	3.030	0.000	0.000	5.793	5.793	0.282	0.000	-4.105	4.115
35.000	-0.234	0.000	-2.876	2.885	0.000	0.000	5.645	5.645	0.289	0.000	-4.035	4.045
36.000	-0.261	0.000	-2.724	2.737	0.000	0.000	5.492	5.492	0.296	0.000	-3.962	3.973
37.000	-0.289	0.000	-2.568	2.584	0.000	0.000	5.333	5.333	0.303	0.000	-3.888	3.900
38.000	-0.320	0.000	-2.407	2.428	0.000	0.000	5.170	5.170	0.310	0.000	-3.813	3.825
39.000	-0.353	0.000	-2.241	2.269	0.000	0.000	5.002	5.002	0.318	0.000	-3.735	3.749
40.000	-0.389	0.000	-2.071	2.107	0.000	0.000	4.829	4.829	0.326	0.000	-3.657	3.671

DATE: FEB. 4, 1968
 RUN DESCRIPTION: EXAMPLE 2: DYNAMIC JOINT TEST
 SLIP JOINT / 600 KNOT WIND
 VEHICLE DECELERATION: CONSTANT WIND VELOCITY OF 600 KNOTS
 CRASH VICTIM: 95TH PERCENTILE MALE

PAGE: 22.01

JOINT PARAMETERS

JOINT NO. 1 - ELB							
TIME (MSEC)	STATE	JOINT ANGLES (DEG)			TOTAL TORQUE (IN. LB.)		
		FLEXURE	AZIMUTH	TORSION	SPRING	VISCOUS	RES.
0.000	5.	67.190	180.000	0.000	0.000	0.000	0.000
0.500	5.	67.189	180.000	0.000	0.000	0.000	0.000
0.750	5.	67.188	-180.000	0.000	0.000	0.000	0.000
1.000	5.	67.186	-180.000	0.000	0.000	0.000	0.000
1.500	5.	67.180	-180.000	0.000	0.000	0.000	0.000
2.000	5.	67.172	-180.000	0.000	0.000	0.000	0.000
3.000	5.	67.149	-180.000	0.000	0.000	0.000	0.000
4.000	5.	67.116	-180.000	0.000	0.000	0.000	0.000
5.000	5.	67.073	-180.000	0.000	0.000	0.000	0.000
6.000	5.	67.021	-180.000	0.000	0.000	0.000	0.000
7.000	5.	66.959	-180.000	0.000	0.000	0.000	0.000
8.000	5.	66.888	-180.000	0.000	0.000	0.000	0.000
9.000	5.	66.808	-180.000	0.000	0.000	0.000	0.000
10.000	5.	66.720	-180.000	0.000	0.000	0.000	0.000
11.000	5.	66.623	-180.000	0.000	0.000	0.000	0.000
12.000	5.	66.517	-180.000	0.000	0.000	0.000	0.000
13.000	5.	66.403	-180.000	0.000	0.000	0.000	0.000
14.000	5.	66.280	-180.000	0.000	0.000	0.000	0.000
15.000	5.	66.149	-180.000	0.000	0.000	0.000	0.000
16.000	5.	66.010	-180.000	0.000	0.000	0.000	0.000
17.000	5.	65.862	-180.000	0.000	0.000	0.000	0.000
18.000	5.	65.706	-180.000	0.000	0.000	0.000	0.000
19.000	5.	65.542	-180.000	0.000	0.000	0.000	0.000
20.000	5.	65.369	-180.000	0.000	0.000	0.000	0.000
21.000	5.	65.189	-180.000	0.000	0.000	0.000	0.000
22.000	5.	65.001	-180.000	0.000	0.000	0.000	0.000
23.000	5.	64.805	-180.000	0.000	0.000	0.000	0.000
24.000	5.	64.601	-180.000	0.000	0.000	0.000	0.000
25.000	5.	64.390	-180.000	0.000	0.000	0.000	0.000
26.000	5.	64.171	-180.000	0.000	0.000	0.000	0.000
27.000	5.	63.945	-180.000	0.000	0.000	0.000	0.000
28.000	5.	63.713	-180.000	-0.001	0.000	0.000	0.000
29.000	5.	63.474	-179.999	-0.001	0.000	0.000	0.000
30.000	5.	63.230	-179.999	-0.001	0.000	0.000	0.000
31.000	5.	62.980	-179.999	-0.002	0.000	0.000	0.000
32.000	5.	62.724	-179.999	-0.002	0.000	0.000	0.000
33.000	5.	62.464	-179.998	-0.002	0.000	0.000	0.000
34.000	5.	62.200	-179.998	-0.003	0.000	0.000	0.000
35.000	5.	61.934	-179.998	-0.003	0.000	0.000	0.000
36.000	5.	61.666	-179.998	-0.003	0.000	0.000	0.000
37.000	5.	61.395	-179.997	-0.004	0.000	0.000	0.000
38.000	5.	61.122	-179.997	-0.004	0.000	0.000	0.000
39.000	5.	60.847	-179.997	-0.004	0.000	0.000	0.000
40.000	5.	60.570	-179.996	-0.005	0.000	0.000	0.000

DATE: FEB. 4, 1988
 RUN DESCRIPTION: EXAMPLE 2: DYNAMIC JOINT TEST
 SLIP JOINT / 600 KNOT WIND
 VEHICLE DECELERATION: CONSTANT WIND VELOCITY OF 600 KNOTS
 CRASH VICTIM: 95TH PERCENTILE MALE

PAGE: 23.01

SEGMENT WIND FORCE (LB.)

TIME (MSEC)	SEGMENT NO. 1 - LARM IN GRND REFERENCE				SEGMENT NO. 2 - UARM IN GRND REFERENCE			
	X	Y	Z	RES	X	Y	Z	RES
0.000	0.000	0.000	1.885	1.885	0.000	0.000	117.584	117.584
0.500	0.000	0.000	92.336	92.336	0.000	0.000	28.783	28.783
0.750	0.000	0.000	92.311	92.311	0.000	0.000	28.775	28.775
1.000	0.000	0.000	92.288	92.288	0.000	0.000	28.768	28.768
1.500	0.000	0.000	92.242	92.242	0.000	0.000	28.752	28.752
2.000	0.000	0.000	92.198	92.198	0.000	0.000	28.735	28.735
3.000	0.000	0.000	92.112	92.112	0.000	0.000	28.702	28.702
4.000	0.000	0.000	92.032	92.032	0.000	0.000	28.667	28.667
5.000	0.000	0.000	91.958	91.958	0.000	0.000	28.632	28.632
6.000	0.000	0.000	91.888	91.888	0.000	0.000	28.595	28.595
7.000	0.000	0.000	91.825	91.825	0.000	0.000	23.798	23.798
8.000	0.000	0.000	91.766	91.766	0.000	0.000	23.765	23.765
9.000	0.000	0.000	91.713	91.713	0.000	0.000	23.733	23.733
10.000	0.000	0.000	91.666	91.666	0.000	0.000	23.699	23.699
11.000	0.000	0.000	91.624	91.624	0.000	0.000	23.666	23.666
12.000	0.000	0.000	91.589	91.589	0.000	0.000	23.632	23.632
13.000	0.000	0.000	91.560	91.560	0.000	0.000	23.597	23.597
14.000	0.000	0.000	91.538	91.538	0.000	0.000	23.563	23.563
15.000	0.000	0.000	91.523	91.523	0.000	0.000	23.529	23.529
16.000	0.000	0.000	91.516	91.516	0.000	0.000	23.494	23.494
17.000	0.000	0.000	91.516	91.516	0.000	0.000	23.460	23.460
18.000	0.000	0.000	91.524	91.524	0.000	0.000	23.427	23.427
19.000	0.000	0.000	91.541	91.541	0.000	0.000	23.394	23.394
20.000	0.000	0.000	91.567	91.567	0.000	0.000	23.361	23.361
21.000	0.000	0.000	91.602	91.602	0.000	0.000	23.330	23.330
22.000	0.000	0.000	91.648	91.648	0.000	0.000	23.299	23.299
23.000	0.000	0.000	91.704	91.704	0.000	0.000	23.269	23.269
24.000	0.000	0.000	91.772	91.772	0.000	0.000	23.240	23.240
25.000	0.000	0.000	91.852	91.852	0.000	0.000	23.213	23.213
26.000	0.000	0.000	91.944	91.944	0.000	0.000	23.187	23.187
27.000	0.000	0.000	92.050	92.050	0.000	0.000	23.162	23.162
28.000	0.000	0.000	92.169	92.169	0.000	0.000	20.825	20.825
29.000	0.000	0.000	92.303	92.303	0.000	0.000	20.806	20.806
30.000	0.000	0.000	92.451	92.451	0.000	0.000	20.789	20.789
31.000	0.000	0.000	92.615	92.615	0.000	0.000	20.773	20.773
32.000	0.000	0.000	92.796	92.796	0.000	0.000	20.758	20.758
33.000	0.000	0.000	92.994	92.994	0.000	0.000	16.136	16.136
34.000	0.000	0.000	93.209	93.209	0.000	0.000	16.127	16.127
35.000	0.000	0.000	93.442	93.442	0.000	0.000	16.119	16.119
36.000	0.000	0.000	93.693	93.693	0.000	0.000	16.113	16.113
37.000	0.000	0.000	93.963	93.963	0.000	0.000	16.108	16.108
38.000	0.000	0.000	94.254	94.254	0.000	0.000	16.103	16.103
39.000	0.000	0.000	94.566	94.566	0.000	0.000	16.100	16.100
40.000	0.000	0.000	94.900	94.900	0.000	0.000	16.097	16.097

DATE: FEB. 4, 1988
 RUN DESCRIPTION: EXAMPLE 2: DYNAMIC JOINT TEST
 SLIP JOINT ' 600 KNOT WIND
 VEHICLE DECELERATION: CONSTANT WIND VELOCITY OF 600 KNOTS
 CRASH VICTIM: 95TH PERCENTILE MALE

PAGE: 24.01

ELBW JOINT FORCES & TORQUES ON UARM IN LARM REFERENCE

TIME (MSEC)	JOINT FORCE (LB. 10**2)			JOINT TORQUE (IN.- LB. 10**2)		
	X	Y	Z	X	Y	Z
0.000	-0.007	0.000	0.000	0.0000+00	0.0000+00	0.0000+00
0.500	0.022	0.000	0.000	0.0000+00	0.0000+00	0.0000+00
0.750	0.022	0.000	0.000	0.0000+00	0.0000+00	0.0000+00
1.000	0.022	0.000	0.000	0.0000+00	0.0000+00	0.0000+00
1.500	0.022	0.000	0.000	0.0000+00	0.0000+00	0.0000+00
2.000	0.023	0.000	0.000	0.0000+00	0.0000+00	0.0000+00
3.000	0.023	0.000	0.000	0.0000+00	0.0000+00	0.0000+00
4.000	0.023	0.000	0.000	0.0000+00	0.0000+00	0.0000+00
5.000	0.023	0.000	0.000	0.0000+00	0.0000+00	0.0000+00
6.000	0.023	0.000	0.000	0.0000+00	0.0000+00	0.0000+00
7.000	0.022	0.000	0.000	0.0000+00	0.0000+00	0.0000+00
8.000	0.022	0.000	0.000	0.0000+00	0.0000+00	0.0000+00
9.000	0.022	0.000	0.000	0.0000+00	0.0000+00	0.0000+00
10.000	0.022	0.000	0.000	0.0000+00	0.0000+00	0.0000+00
11.000	0.023	0.000	0.000	0.0000+00	0.0000+00	0.0000+00
12.000	0.023	0.000	0.000	0.0000+00	0.0000+00	0.0000+00
13.000	0.023	0.000	0.000	0.0000+00	0.0000+00	0.0000+00
14.000	0.024	0.000	0.000	0.0000+00	0.0000+00	0.0000+00
15.000	0.024	0.000	0.000	0.0000+00	0.0000+00	0.0000+00
16.000	0.025	0.000	0.000	0.0000+00	0.0000+00	0.0000+00
17.000	0.025	0.000	0.000	0.0000+00	0.0000+00	0.0000+00
18.000	0.026	0.000	0.000	0.0000+00	0.0000+00	0.0000+00
19.000	0.027	0.000	0.000	0.0000+00	0.0000+00	0.0000+00
20.000	0.027	0.000	0.000	0.0000+00	0.0000+00	0.0000+00
21.000	0.028	0.000	0.000	0.0000+00	0.0000+00	0.0000+00
22.000	0.029	0.000	0.000	0.0000+00	0.0000+00	0.0000+00
23.000	0.030	0.000	0.000	0.0000+00	0.0000+00	0.0000+00
24.000	0.030	0.000	0.000	0.0000+00	0.0000+00	0.0000+00
25.000	0.031	0.000	0.000	0.0000+00	0.0000+00	0.0000+00
26.000	0.032	-0.002	0.000	0.0000+00	0.0000+00	0.0000+00
27.000	0.034	0.000	0.000	0.0000+00	0.0000+00	0.0000+00
28.000	0.032	0.000	0.000	0.0000+00	0.0000+00	0.0000+00
29.000	0.034	0.000	0.000	0.0000+00	0.0000+00	0.0000+00
30.000	0.035	0.000	0.000	0.0000+00	0.0000+00	0.0000+00
31.000	0.036	0.000	0.000	0.0000+00	0.0000+00	0.0000+00
32.000	0.038	0.000	0.000	0.0000+00	0.0000+00	0.0000+00
33.000	0.034	0.000	0.000	0.0000+00	0.0000+00	0.0000+00
34.000	0.036	0.000	0.000	0.0000+00	0.0000+00	0.0000+00
35.000	0.038	0.000	0.000	0.0000+00	0.0000+00	0.0000+00
36.000	0.040	0.000	0.000	0.0000+00	0.0000+00	0.0000+00
37.000	0.041	0.000	0.000	0.0000+00	0.0000+00	0.0000+00
38.000	0.044	0.000	0.000	0.0000+00	0.0000+00	0.0000+00
39.000	0.046	0.000	0.000	0.0000+00	0.0000+00	0.0000+00
40.000	0.048	0.000	0.000	0.0000+00	0.0000+00	0.0000+00

SUB	CALLS	TIME	%
MAIN30	1	47	1.52
INPUT	1	44	1.42
CHAIN	192	37	1.20
EJOINT	192	3	0.10
DINT	21	143	4.63
PDAUX	234	123	3.98
DAUX	191	86	2.78
SETUP1	191	46	1.49
CONTCT	191	22	0.71
WINDY	382	2109	68.23
VISPR	191	72	2.33
SETUP2	191	6	0.19
DAUX11	191	31	1.00
DAUX12	191	1	0.03
DAUX22	191	9	0.29
FSMSOL	191	22	0.71
OUTPUT	44	79	2.56
UPDATE	43	3	0.10
D2P	190	42	1.36
POSTPR	1	166	5.37
TOTAL		3091	100.00

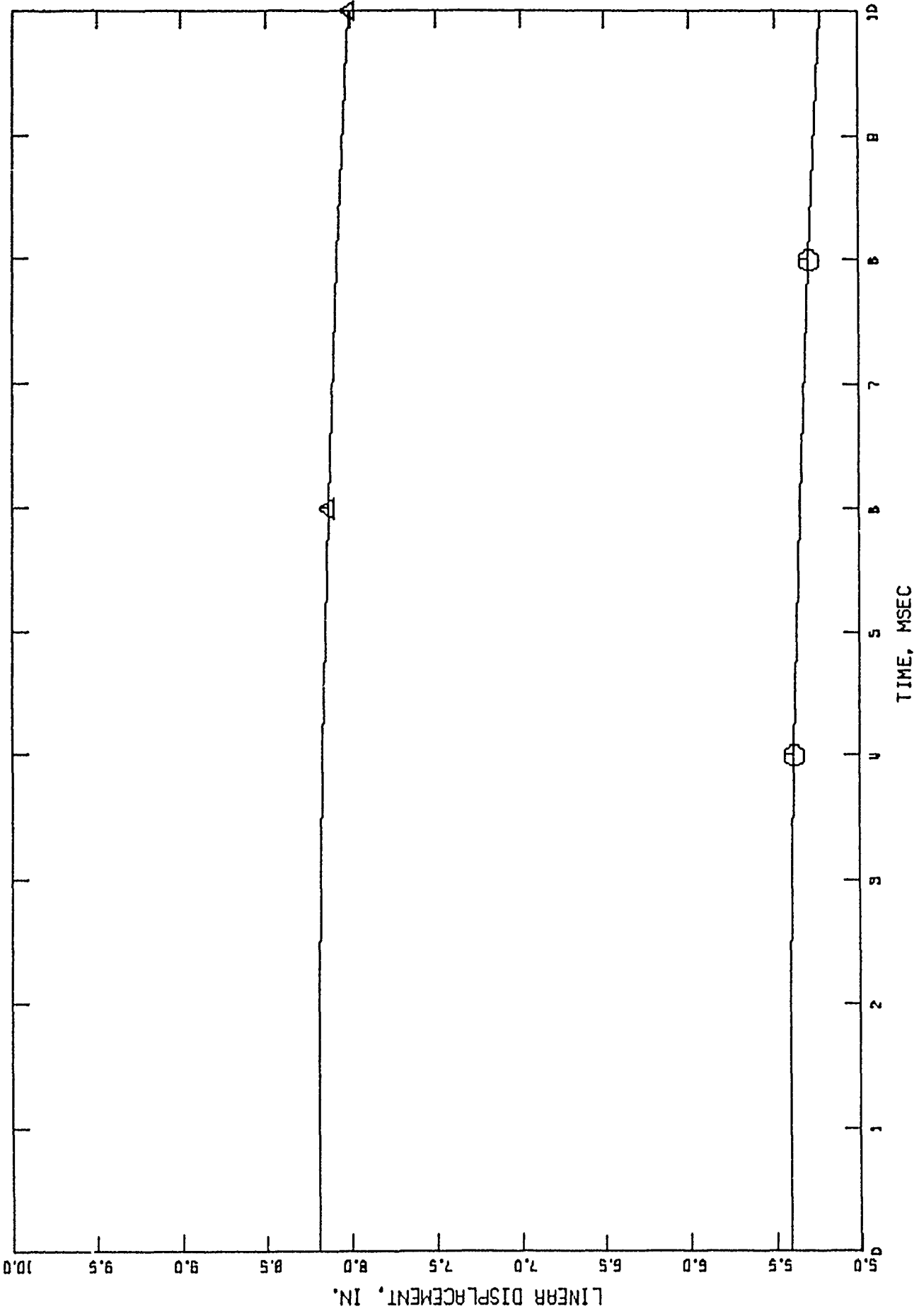
APPENDIX E1

Example 2, ATB Dynamic Joint Test
Input File with Plotting Options (EX2PLOT.AIN)

FEB. 4, 1988	0	0	0.0												CARD A1A																	
EXAMPLE 2: DYNAMIC JOINT TEST																																
SLIP JOINT / 600 KNOT WIND																																
IN. LB.SEC.	0.0		0.0	386.088		0.0									CARD A3																	
4	20	0.002	0.0005	0.001	.000063										CARD A4																	
1	020	3	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	CARD A5	
2	1			95TH PERCENTILE MALE												CARD B1																
LARM	L	5.901	.3331	.3331	.0214	1.871	1.871	10.269	0.000	0.000	0.000																				CARD B2A	
UARM	U	5.542	.1743	.1743	.0259	2.122	2.122	7.497	0.000	0.000	0.000																				CARD B2A	
ELBW	E	1	5	0.00	0.00	8.20	0.00	0.00	-5.42	1	0.00	0.00																			CARD B3A	
				0.00	0.00	0.00	0.00	-67.19	0.00																							
.00	.00	.00	.00			.00	.00	.00	.00	.00																						CARD B4
0.0	0.0	30.	0.0	0.0					0.0	0.0																					CARD B5	
.00	.00	.00	.00	.00	.00	.00	.01	.01	.001	.001	.001	.001	.001	.001	.001																CARD B6	
.00	.00	.00	.00	.00	.00	.01	.01	.001	.00	.00																					CARD B6	
CONSTANT WIND VELOCITY OF 600 KNOTS																																
0.0	0.0	0.012161.	0.00	0.00	0.00	0.00	0.00	-3	0.0	2.30																					CARD C2A	
2	1	3																													CARD C2B	
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000																					CARD C5	
0.000	0.000	0.000	12161.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000																					CARD C5	
2.300	0.000	0.000	12161.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000																					CARD C5	
4.600	0.000	0.000	12161.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000																					CARD C5	
1	0	0	0	0	0	0	0	1	0	0																					CARD D1	
1	WIND PLANE															CARD D2A																
	10.0		10.0		5.0																										CARD D2B	
	-10.0		10.0		5.0																										CARD D2C	
	10.0		-10.0		5.0																										CARD D2D	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0												CARD D7	
999																CARD E1																
1	WIND FORCE															CARD E6A																
	1.4000	13404.0000	14.6960			1		3																							CARD E6B	
0																CARD F1A																
0																CARD F3A																
0																CARD F4A																
1																CARD F7A																
-1	1	3	1	1	0	1	2	2																							CARD F7B	
-2	2	3	1	1	0	1	1	1																							CARD F7B	
	0.0		0.0		0.0	0	0	0	0	0	1																				CARD G1A	
0.66000	0.00000	-18.98000			0.0			0.0																							CARD G2A	
0.00	-2.00	0.00			0.00			0.00			0.00	3	2	1																	CARD G3A	
0.00	-2.00	0.00			0.00			0.00			0.00	3	2	1																	CARD G3A	
0																CARD H1A																
																CARD H1B																
0																CARD H2A																
																CARD H2B																
3	2	1			0.0			0.0			8.20																				CARD H3A	
	1	2			0.0			0.0			-5.42																				CARD H3B	
	4	2			0.0			0.0			0.0																				CARD H3B	
0																CARD H4																
0																CARD H5																
0																CARD H6																
1																CARD H7																
2	0	1	0	2												CARD H8																
1	1	1														CARD H9																
0																CARD H10																
																CARD H11																
1	2															CARD I1																
21	0	21	4	21	8											CARD I2																
10		0.0	.0.0	9.0	10.0											CARD I3																
10		5.0	10.0	6.5	7.5											CARD I4																
10	TIME, MSEC															CARD I5																
24	LINEAR DISPLACEMENT, V.															CARD I6																
23	UPPER ARM AND LOWER ARM															CARD I7																
16	SEGMENTS 1 AND 2															CARD I8																

APPENDIX E2

Example 2, ATB Dynamic Joint Test X-Y Plot



UPPER ARM AND LOWER ARM
SEGMENTS 1 AND 2

APPENDIX E3

Example 2, ATB Dynamic Joint Test Printer Plot

T= 0.000 Y0= 0.00000 Z0= 0.00000 Y-Z PLANE

.....

T= 0.000 X0= 0.00000 Z0= 0.00000 X-Z PLANE
L

T= 2.000 X0= 0.00000 Z0= 24.32200 X-Z PLANE

L

.....

T= 2.000 X0= 0.00000 Y0= 0.00000 X-Y PLANE

.....

APPENDIX F1

Example 1, VIEW -- Input File (EXP1.VIN)

2
1

ATB COURSE EXAMPLE 1

4
4
4
4
4
4
4
4
4
4
4
4
4
4
4
4
4
4
4
4
4

0.000 0.01000 .0800

0000

0 016

0

2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	1	1
1	1	3	1	1	3	3	3	1
1	3	3	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1

8 5 8 3 5 5 3 3 5 3 3 3 3 3 3 1010

20

4

110.00

2.0	3.0							
0.0	1000.0	-20.0	0.0	0.0	-20.0	16	2	
0.00	9.00							

APPENDIX F2

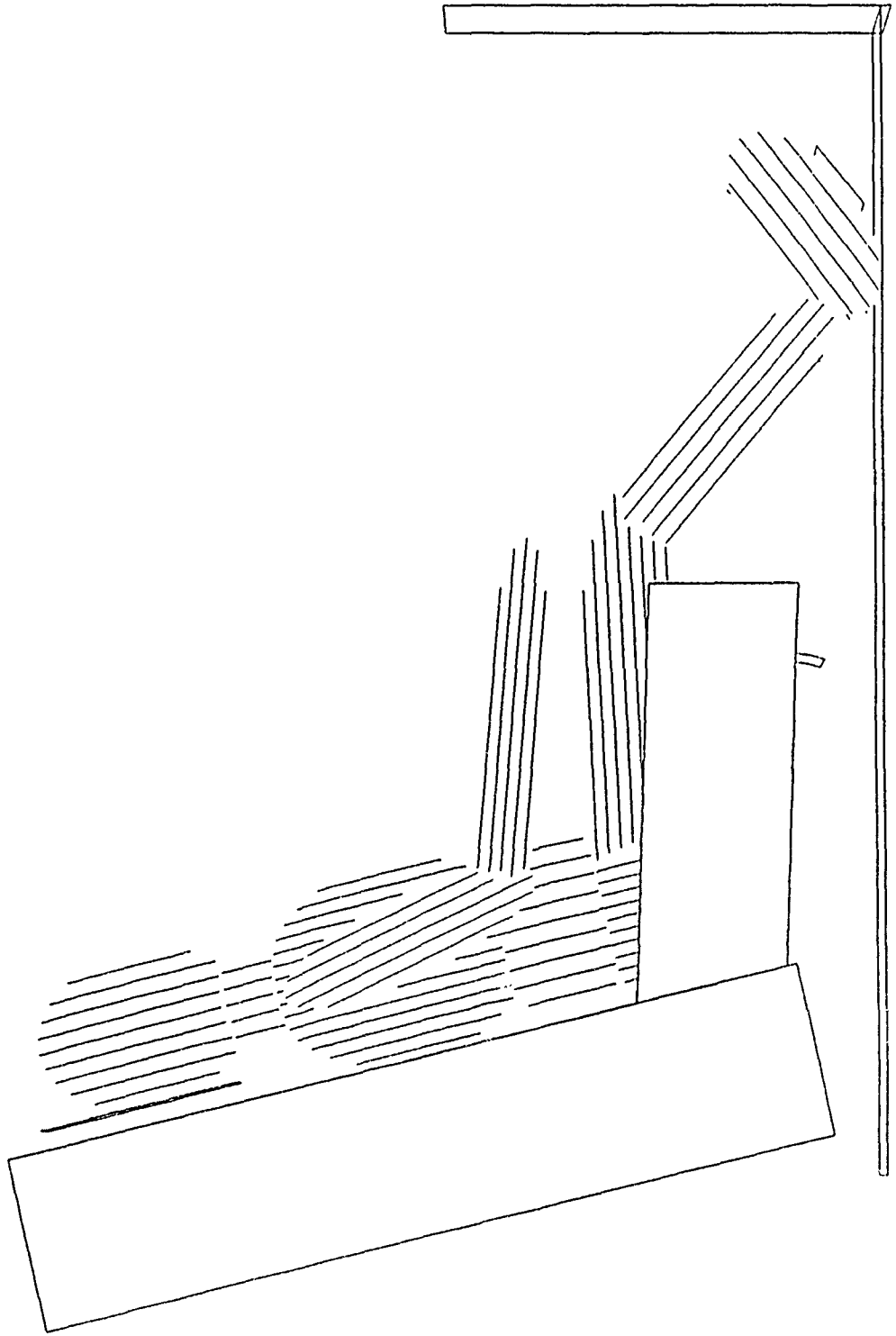
Example 1, VIEW -- 386 Output Listing (EXPL.VOU)

APPENDIX F3

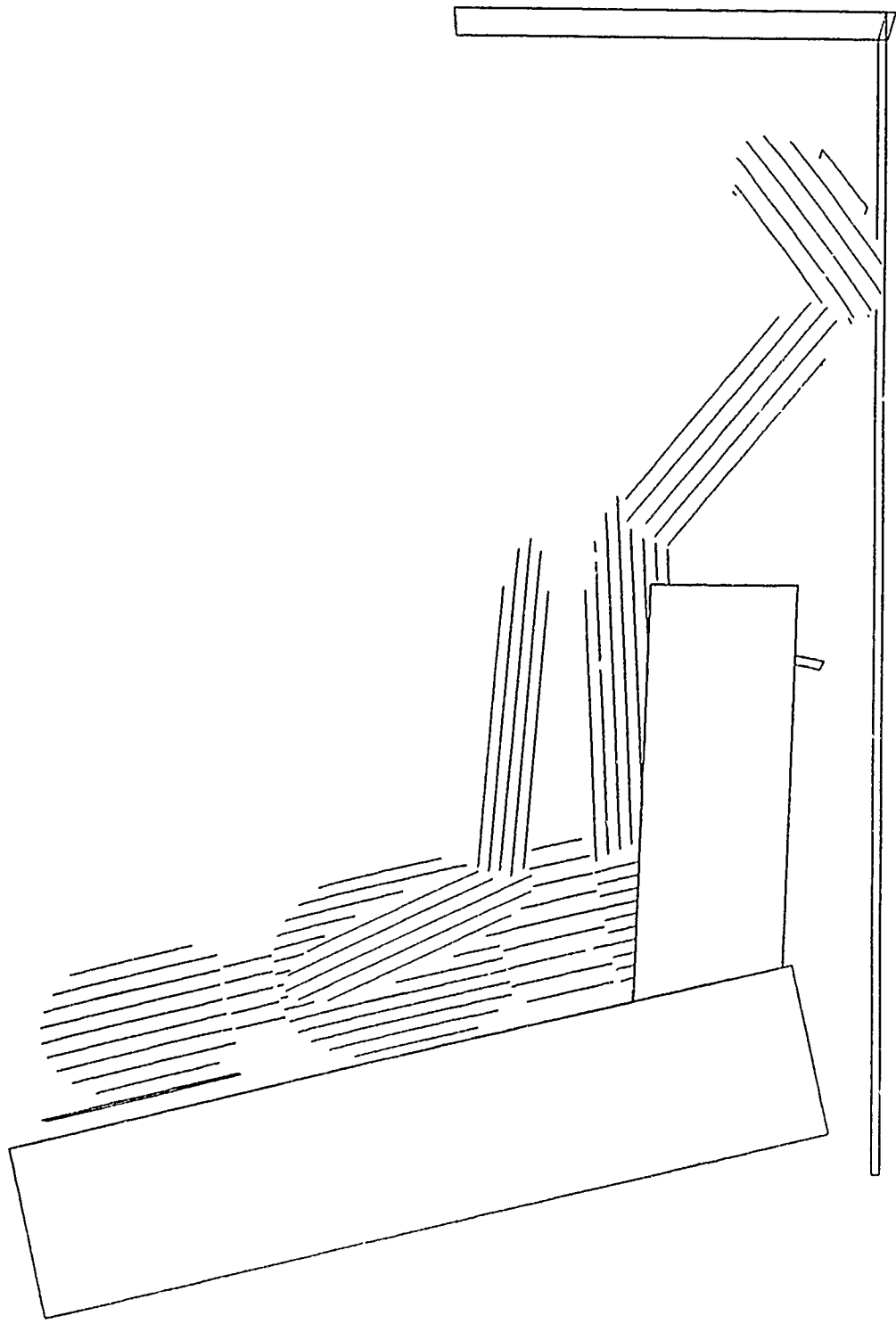
Example 1, VIEW -- 386 Output Graphics

ATB COURSE EXAMPLE 1

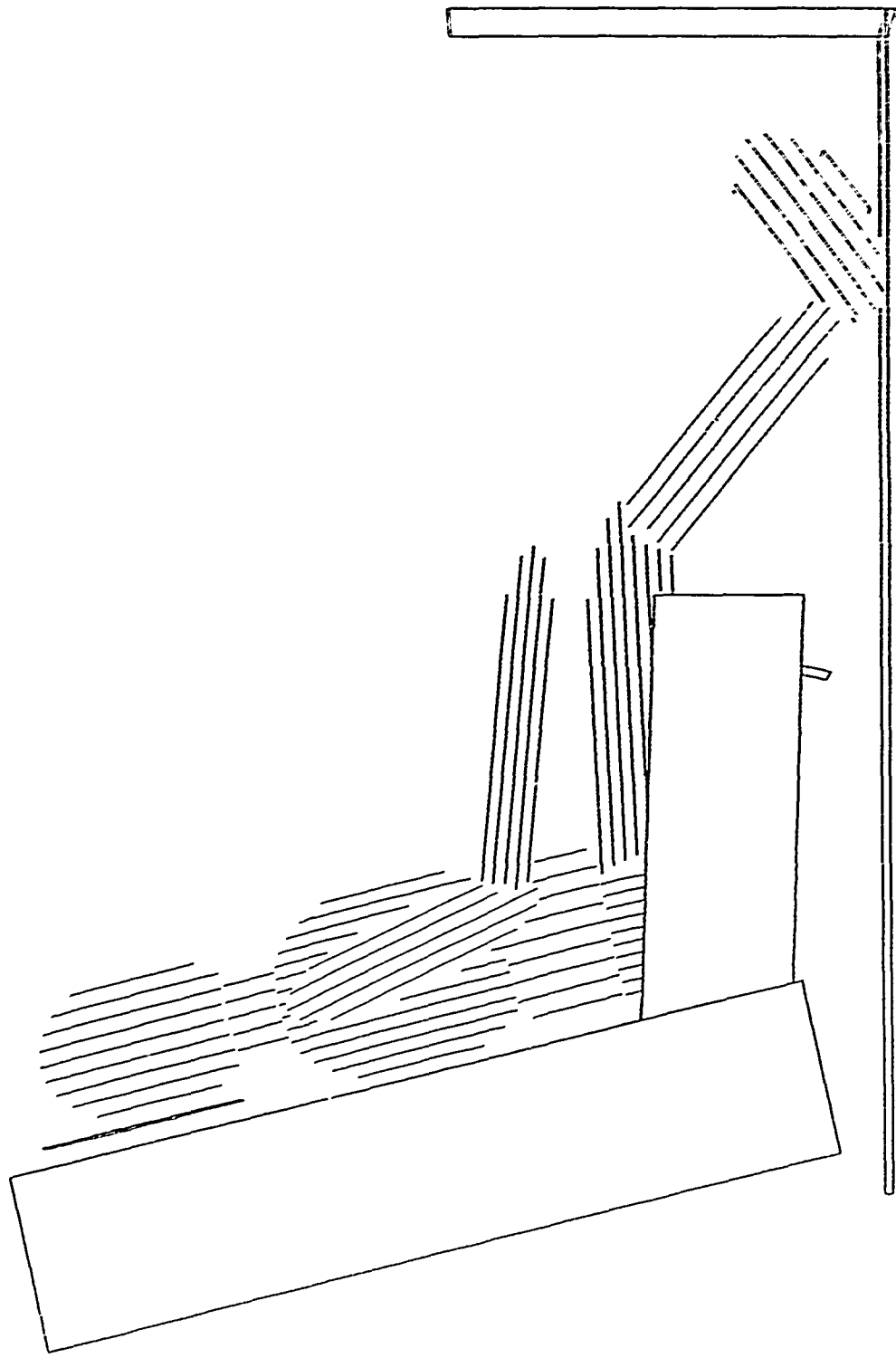
TIME (MSEC) 0



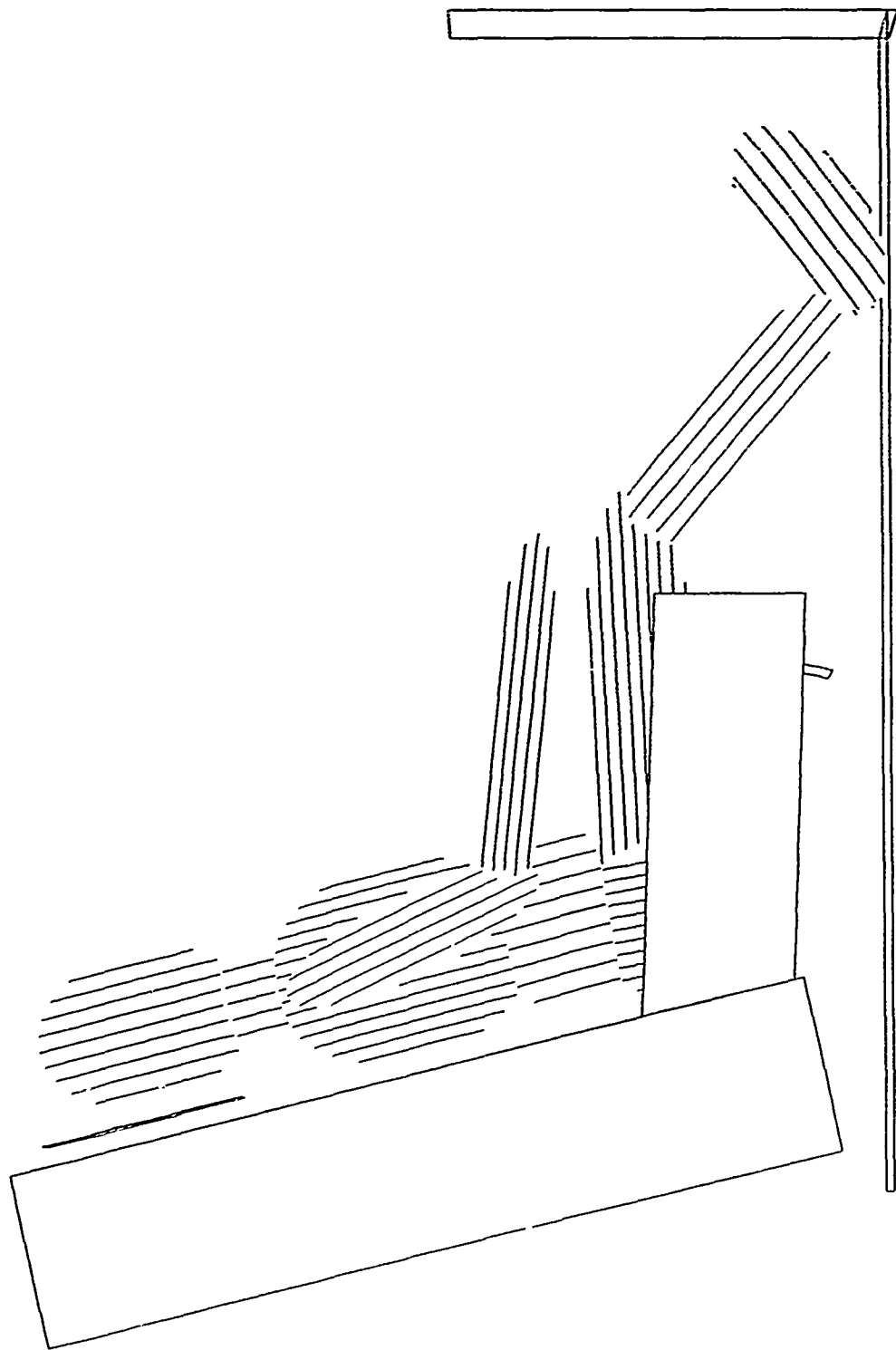
TIME (MSEC) 10



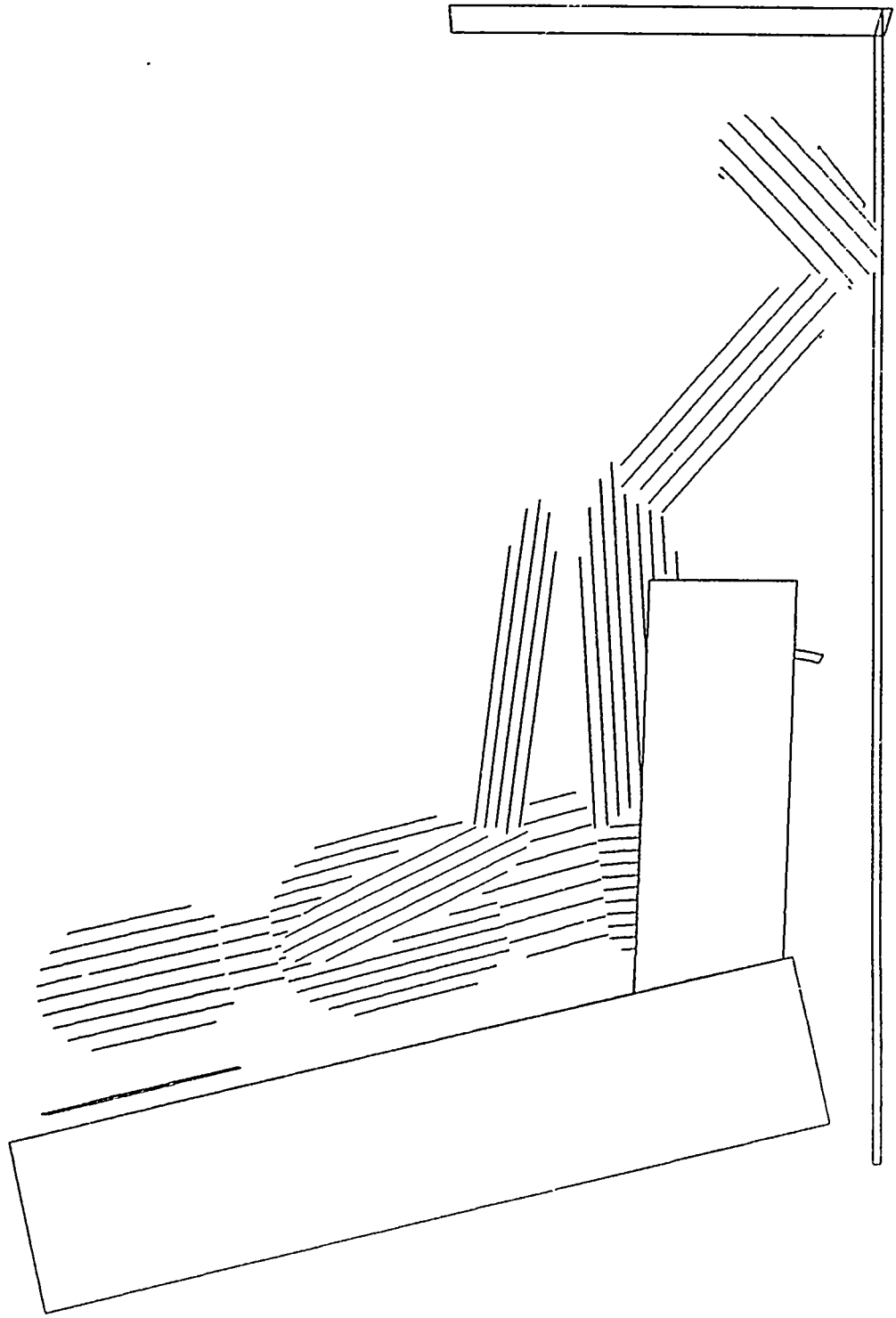
TIME (MSEC) 20



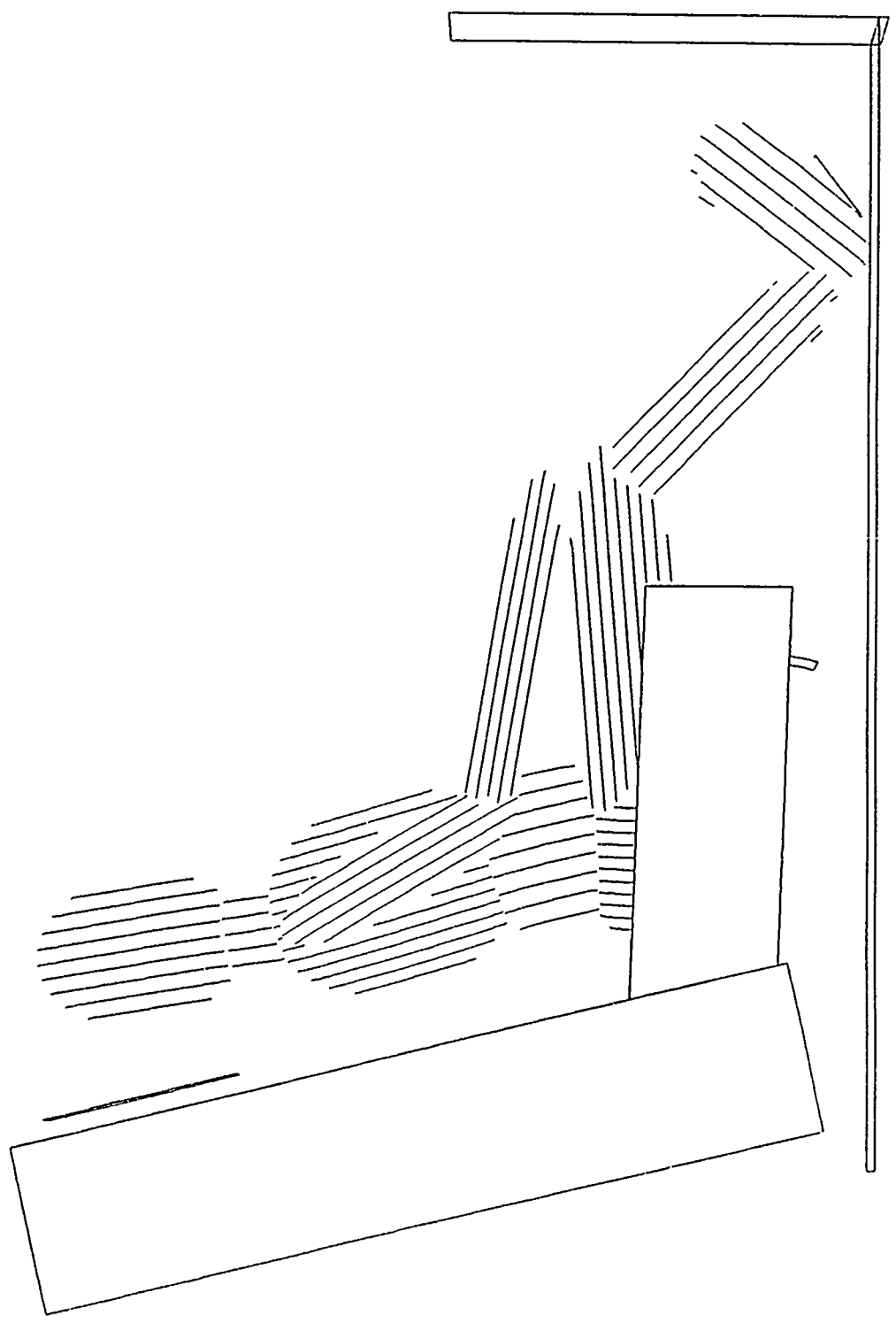
TIME (MSEC) 30



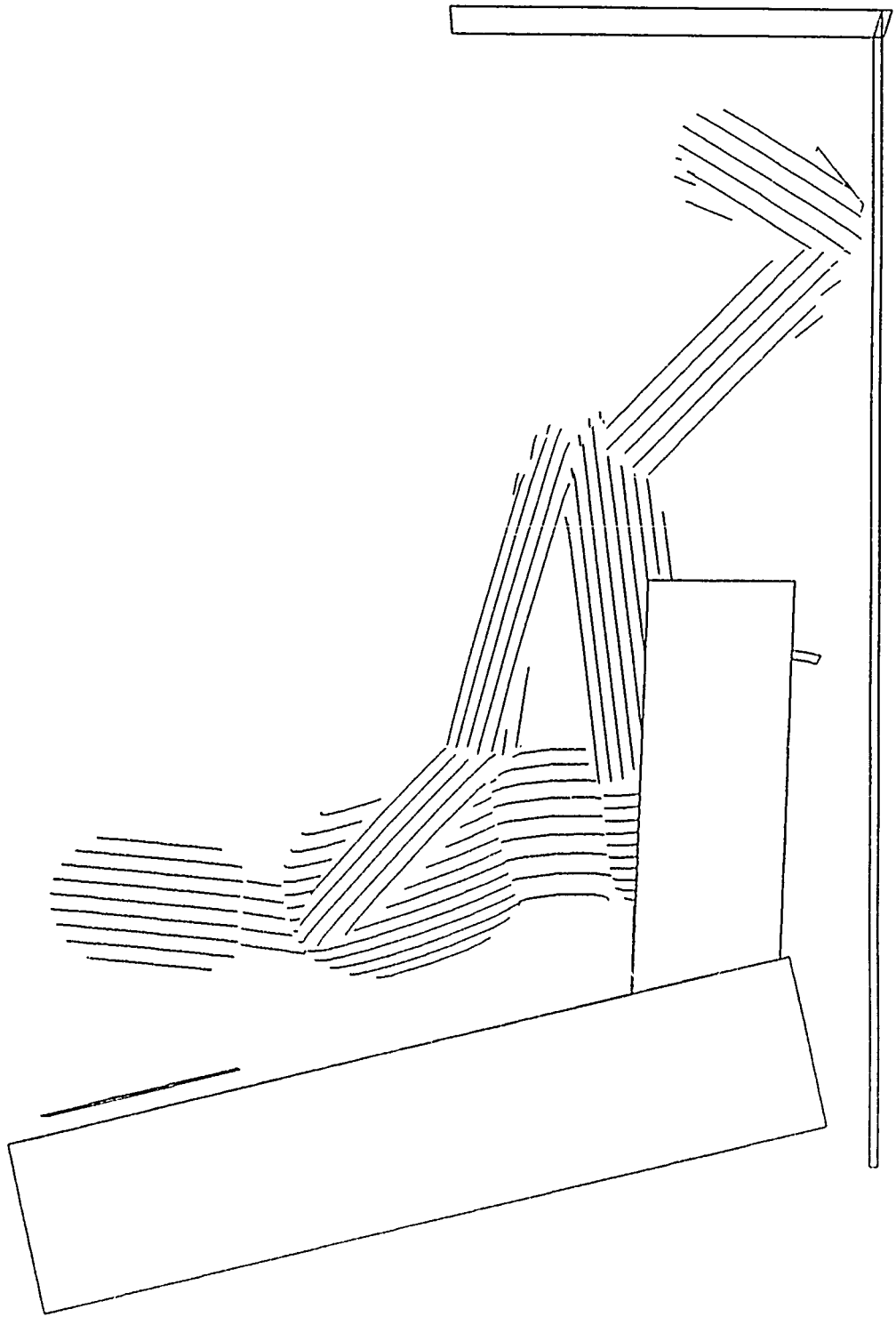
TIME (MSEC) 40



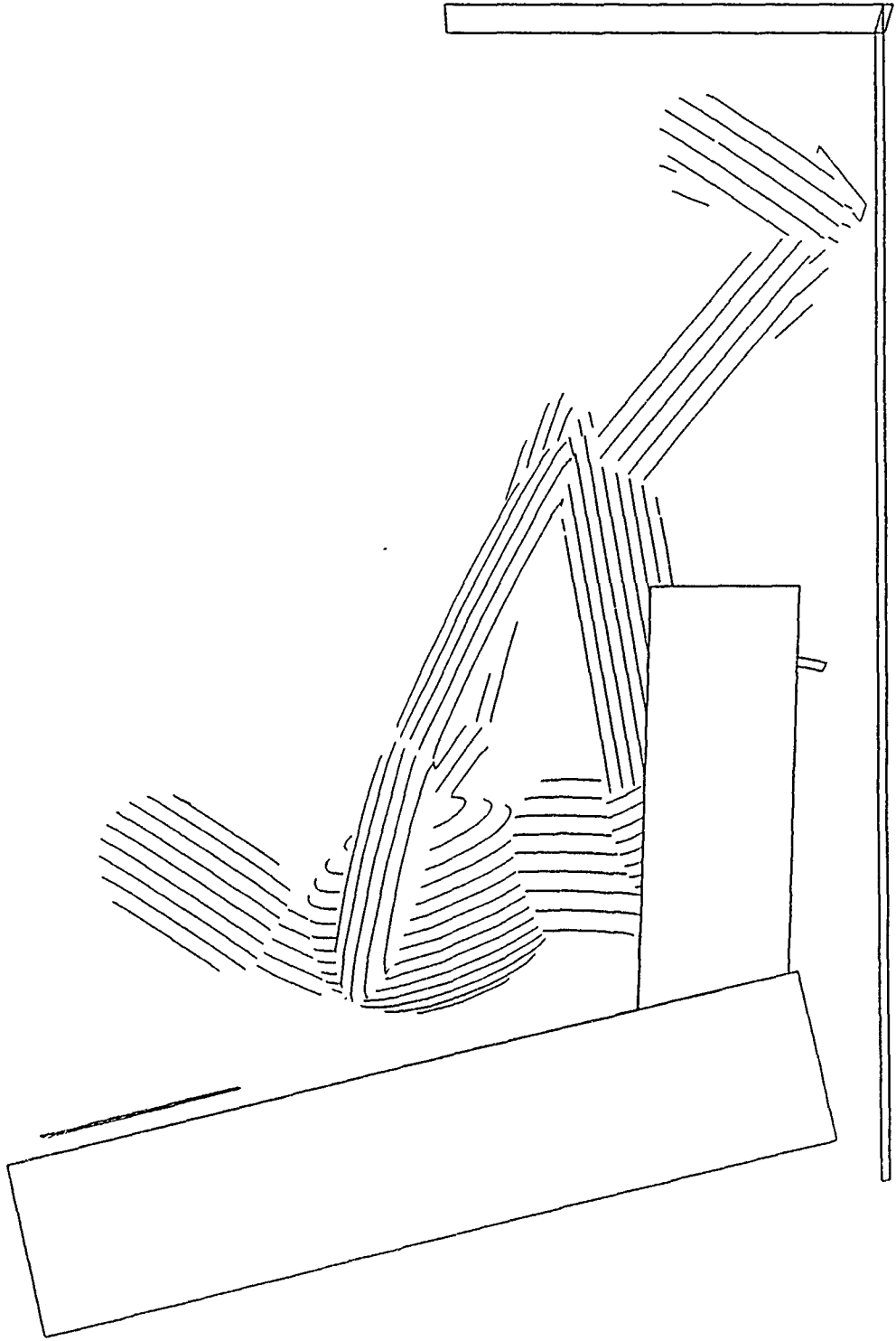
TIME (MSEC) 50



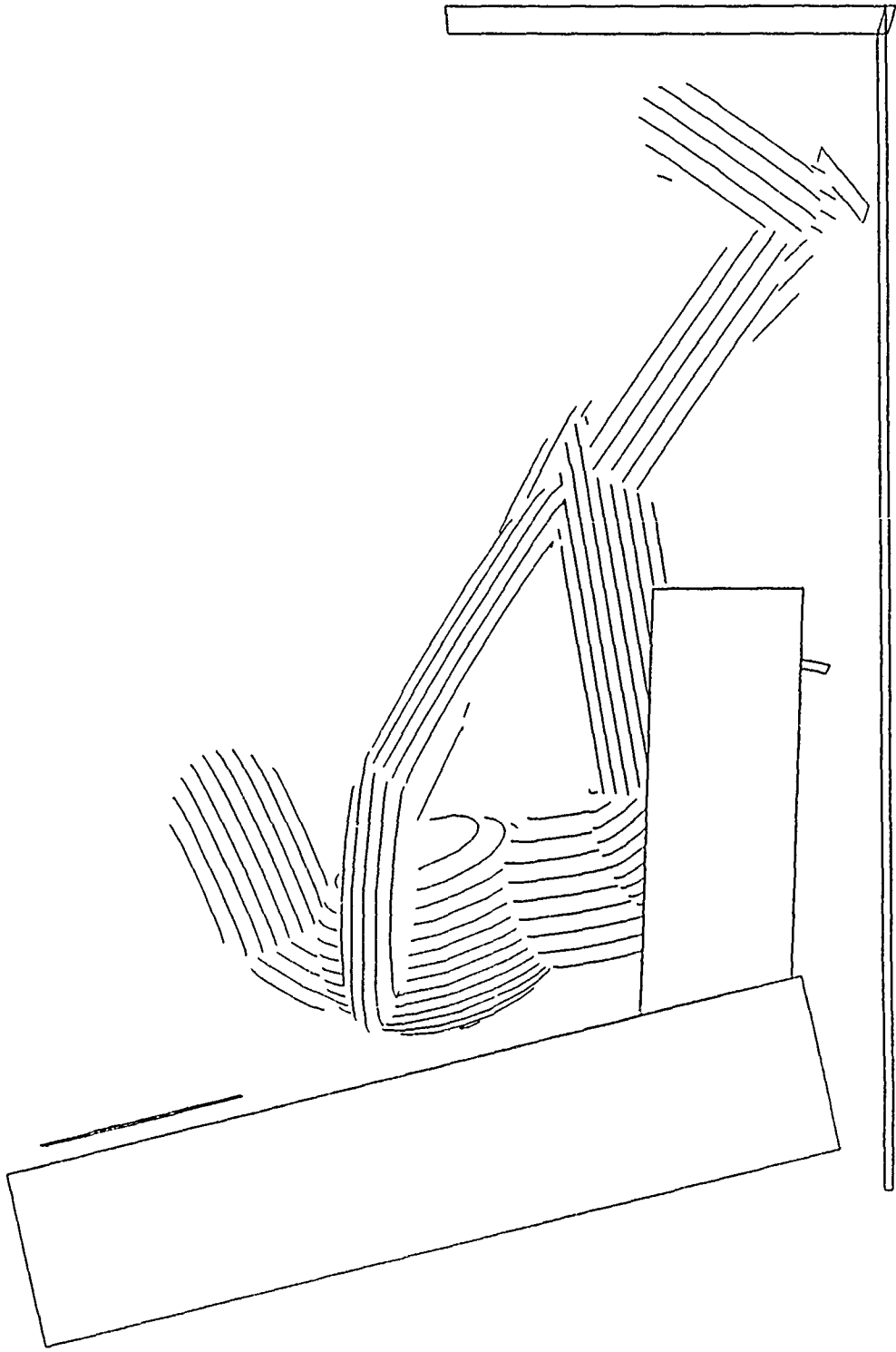
TIME (MSEC) 60



TIME (MSEC) 70



TIME (MSEC) 80



APPENDIX G1

Example 2, VIEW -- Input File (EXP2.VIW)

2
1

ATB COURSE EXAMPLE 2

4
4
4
4
4
4
4
4
4
4
4
4
4
4
4
4
4
4
4
4
4
4
4

0.000 0.01000 .0400

0000
0 0 4

0

2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	1	1
1	1	3	1	1	3	3	1	1
1	3	3	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
4								

8 5 8 3 5 5 3 3 5 3 3 3 3 3 3 1 0 1 0
20

4

	110.00							
4.0	4.0							
0.0	1000.0	-20.0	0.0	0.0	-20.0	4	2	
0.00	9.00							

APPENDIX G2

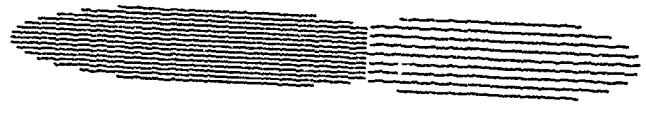
Example 2, VIEW -- 386 Output Listing (EXP2.VOU)

APPENDIX G3

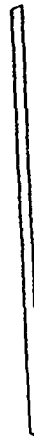
Example 2, VIEW -- 386 Output Graphics

ATB COURSE EXAMPLE 2

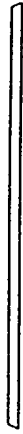
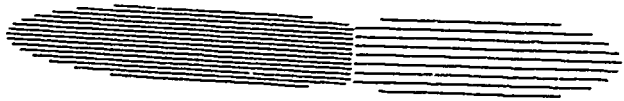
TIME (MSEC) 0



TIME (MSEC) 10



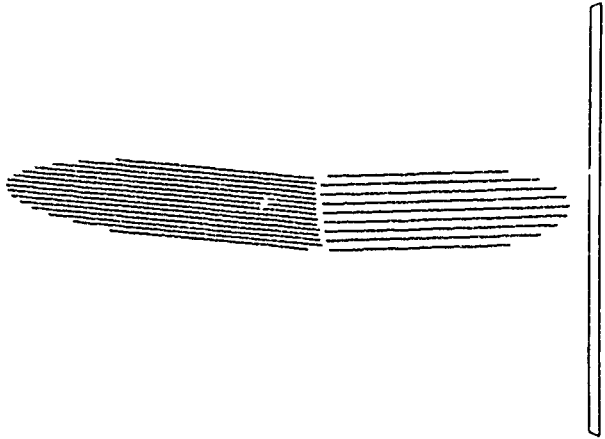
TIME (MSEC) 20



TIME (MSEC) 30



TIME (MSEC) 40



APPENDIX H

386-GEOD Program Listing

```

C--PROGRAM GEBOD------(PERKIN-ELMER  FTN VII)-----C
C   GEBOD GENERATES BODY DESCRIPTION DATA SETS FOR USE IN THE ARTICULATED  C
C   TOTAL BODY (ATB) MODEL.  USER HAS CHOICE OF PRODUCING A DATA SET  C
C   CORRESPONDING TO ADULT MALE, ADULT FEMALE, CHILD SUBJECT, OR SOME  C
C   DUMMY SUBJECTS.  THE BASIC GEOMETRY OF THE MODEL USED TO CALCULATE THE  C
C   BODY DESCRIPTION DATA FOR THE HUMAN DATASETS IS AN EXTENSIVELY  C
C   MODIFIED VERSION OF THE MODEL USED BY CALSPAN CORP.  THE MODIFICATIONS  C
C   TO THE MODEL AND THE RESULTING PROGRAM, WERE WRITTEN BY THE  C
C   UNIVERSITY OF DAYTON RESEARCH INSTITUTE UNDER CONTRACT F33615-81-C-0513.  C
C   THE MODIFICATIONS TO INCLUDE DUMMY DATASETS WERE WRITTEN BY BEECHER  C
C   RESEARCH COMPANY UNDER CONTRACT F33615-87-C-0530.  C
C   80386 IMPLEMENTATION BY KETRON, INC., OCTOBER 31, 1989, UNDER CONTRACT  C
C   F33615-88-C-0543.  C
C  C
C   FILES:  C
C   UNIT1--USER SUPPLIED DIMENSIONAL DATA.  C
C   UNIT2--GEBOD.DAT DATA FILE FOR I/O.  C
C   UNIT3--RESULTS READY FOR INSERTION INTO AN ATB MODEL INPUT DECK.  C
C   UNIT5 (CRT OUTPUT)--INPUT PROMPTS ARE WRITTEN TO THIS UNIT.  C
C   UNIT7 (CRT INPUT)--SUBJECT DESCRIPTION IS ENTERED TO THIS UNIT.  C
C   UNIT9 (PRINTER)--TABLE FORM OF RESULTS.  C
C  C
C   THE GEBOD.DAT FILE WHICH ACCOMPANIES THIS PROGRAM CONTAINS THE  C
C   FOLLOWING DATA:  C
C   1. THE NUMBER OF SUBJECT TYPES IN LIST-DIRECTED FORMAT  C
C   AVAILABLE TO THE USER.  C
C   2. THE TITLES FOR THE SUBJECT TYPES IN LIST-DIRECTED FORMAT.  C
C   3. THE AAMRL/BBM ATB INPUT DATASETS THAT ARE AVAILABLE TO SPECIFIC  C
C   USERS AND USED AS INPUT TO THIS PROGRAM.  EACH ATB INPUT DATASET  C
C   IS PRECEDED BY A HEADER IN LIST-DIRECTED FORMAT STATING THE  C
C   NUMBER OF RECORDS IN THE DATASET AND THE SUBJECT TYPE IT IS  C
C   ASSOCIATED WITH.  FOR EACH SUBJECT TYPE THERE ARE TWO DATASETS,  C
C   ONE IN ENGLISH UNITS AND THE OTHER IN METRIC UNITS.  NOTE  C
C   THAT NOT ALL SUBJECT TYPES HAVE DATA STORED IN THIS FILE AND  C
C   FOR SOME USERS THERE MAY NOT BE ANY ATB INPUT DATASETS STORED.  C
C  C
C   ROUTINES (IN ORDER OF APPEARANCE):  C
C   PROGRAM GEBOD  C
C   BLOCK DATA  C
C   SUBROUTINE DIALOG  C
C   SUBROUTINE ASKUN  C
C   SUBROUTINE PFILE  C
C   SUBROUTINE CONTAC  C
C   SUBROUTINE SEGMAS  C
C   SUBROUTINE SGINER  C
C   SUBROUTINE ELLIP  C
C   ENTRY ELLPMI  C
C   SUBROUTINE TORSO  C
C   SUBROUTINE FEET  C
C   SUBROUTINE RESULTS  C
C   SUBROUTINE CNVRT  C
C   SUBROUTINE RESULTZ  C

```

```

C          SUBROUTINE ATBOUT                      C
C          SUBROUTINE NOATBOUT                   C
C-----C

```

```

PROGRAM GEBOD
COMMON /DIMS/ DD(-1:31), REGEQ(24, -1:31), PRED(0:2)
&          , RANGE(2, -1:1, 3), CONV(-1:1)      80386
COMMON /FLOAT/ PI, PI2, PI4, DENS, GRAV        80386
CHARACTER*12 FILE2, FILE3, FILE9              80386

PI = ACOS(-1.)
PI2 = PI + PI
PI4 = PI2 + PI2

OPEN (UNIT=5, FILE='CON', CARRIAGE CONTROL='LIST') 80386
OPEN (UNIT=7, FILE='CON')                          80386
OPEN (UNIT=4, FILE='GEBODII.DIR', STATUS='OLD')    80386
READ (4, '(A12/A12/A12)') FILE2, FILE3, FILE9     80386
OPEN (UNIT=2, FILE=FILE2, STATUS='OLD')            80386
OPEN (UNIT=3, FILE=FILE3, STATUS='NEW')            80386
OPEN (UNIT=9, FILE=FILE9, STATUS='NEW', CARRIAGE CONTROL='FORTRAN') 80386
10 CALL DIALOG (IPTR, ICNT, ISTRT, ISUB)
   IF (ISUB.LE.4) THEN
     IF (IPTR.LT.25) THEN
       DO 100 I = ISTRT, 31
         DD(I) = REGEQ(IPTR+ICNT, I)
         DO 100 J = 0, ICNT - 1
100          DD(I) = DD(I) + PRED(J)*REGEQ(IPTR+J, I)
       ENDIF

       CALL CONTAC
       CALL SEGMAS
       CALL SGINER
       CALL RESULTS (ISTRT)
     ELSE
       CALL RESULTZ (ISUB)
     ENDIF
   REWIND (2)
   GOTO 10
END

```

```

C--BLOCK DATA-----C
C   DATA IN COMMON AREAS C
C   /SGMNTS/--SEGMENT DATA C
C       ABCN--X,Y, AND Z CONTACT ELLIPSOID SEMIAXES FOR EACH SEGMENT C
C       RMN--MASS OF EACH SEGMENT C
C       PHI--SEGMENT MOMENTS OF INERTIA C
C       XYZCG--LOCAL REFERENCE COORDINATES OF CENTER OF SEGMENT C
C           CONTACT ELLIPSOIDS C
C C C
C   /JNTS/--JOINT DATA C
C       RNJ--LOCAL REFERENCE SYSTEM COORDINATES OF EACH JOINT. THIRD C
C           SUBSCRIPT EQUAL TO 2 SPECIFIES LOCAL REFERENCE SYSTEM OF SEGMENT C
C           J+1. A VALUE OF 1 IN THIS POSITION SPECIFIES LOCAL REFERENCE C
C           SYSTEM OF SEGMENT JNT(J), WHERE IN BOTH CASES J IS THE VALUE OF C
C           THE FIRST SUBSCRIPT. C
C       JNT--POINTER ARRAY FOR ASSOCIATING JOINTS WITH SEGMENTS C
C       YPRLL--YAW, PITCH, AND ROLL ANGLES OF JOINT AXES RELATIVE TO LOCAL C
C           REFERENCE AXES C
C       IPIN--SPECIFIES JOINT TYPE C
C C C
C   /DIMS/--BODY DIMENSION DATA C
C       DD--BODY DIMENSIONS C
C       REGEQ--REGRESSION EQUATIONS USED TO COMPUTE BODY DIMENSIONS C
C       PRED--VALUES OF PREDICTING VARIABLES TO BE USED WITH REGEQ C
C       RANGE--ACCEPTIBLE RANGE FOR PREDICTING VARIABLES C
C       CONV--CONSTANTS FOR CONVERTING BODY DIMENSIONS FROM METRIC TO C
C           ENGLISH UNITS C
C C C
C   /NAMES/--CHARACTER STRINGS FOR VARIOUS LABLES C
C       SUBTYP--DESCRIPTION TO USER OF 4 SOURCES OF BODY DIMENSION DATA C
C       SEGLAB--SEGMENT LABLES C
C       JNTLAB--JOINT LABLES C
C       PLTSYM--PLOTING SYMBOL FOR SEGMENTS AND JOINTS C
C       DIMLAB--NAMES OF THE BODY DIMENSIONS C
C       TITLE--RUN TITLE SUPPLIED BY THE USER C
C       UNITS--UNITS FOR USER TO CHOOSE BETWEEN C
C C C
C   /FLOAT/--VARIOUS CONSTANTS C
C       PI--THE CONSTANT PI C
C       PI2--PI**2 C
C       PI4--PI**4 C
C       DENS--DENSITY USED FOR COMPUTATION OF SEGMENT INERTIAL PROPERTIES C
C       GRAV--ACCELERATION OF GRAVITY C
C C C
C   /CYL/--DATA USED IN RIGHT ELLIPTICAL CYLINDER MODELLING OF SEGMENTS C
C       FOR COMPUTATION OF INERTIAL PROPERTIES C
C       ZEES--CONTAINS Z CORDINATE GIVING VERTICAL EXTENT OF CYLINDERS C
C       ZH--HEIGHT OF EACH CYLINDER IN THE NUMERICAL INTEGRATION C
C       NSEG--NUMBER OF CYLINDERS IN THE NUMERICAL INTEGRATION C
C       CGZ--Z COORDINATE OF CENTER OF GRAVITY OF MODEL C
C       WTCOR--EIGHT COORECTION FACTOR USED TO ADJUST DENS C
C

```



```

C     ALL DATA WILL BE CALCULATED AND STORED IN ENGLISH UNITS
C     DD(-1)--MONTHS
C     DD(0)--POUNDS
C     DD(1)-DD(31)--INCHES
C     DENS--SLUG/IN**3
C     GRAV--FT/SEC**2
C     RMN--SLUGS
C     PHI--SLUG*IN**2
C     ALL REMAINING LINEAR MEASUREMENTS ARE IN INCHES
C     THESE UNITS ARE UTILIZED UNTIL JUST PRIOR TO WRITTING OUT OF RE-
C     SULTS.  VALUES ARE THEN CONVERTED, IN PLACE, TO APPROPRIATE UNITS.
C     THE VALUE STORED IN GRAV IS USED FOR CONVERSION FROM SLUGS TO
C     POUNDS.
C-----C

```

BLOCK DATA

```

COMMON /SGMNTS/ ABCN(15,3),RMN(15),PHI(3,15),XYZCG(15,3)
COMMON /JNTS/ RNJ(3,28),JNT(14),YPRLL(14,6),IPIN(14)
COMMON /DIMS/ DD(-1:31),REGEQ(24,-1:31),PRED(3),
& RANGE(2,-1:1,3),CONV(-1:1)
COMMON /NAMES/ SUBTYP(10),SEGLAB(15),JNTLAB(14),PLTSYM(29),
& DIMLAB(-1:31),TITLE,UNITS(3,-1:2)
COMMON /FLOAT/ PI,PI2,PI4,DENS,GRAV
COMMON /CYL/ ZEE(0:15),ZH(15),NSEG(15),CGZ(15),WTCOR
CHARACTER SUBTYP*35,SEGLAB*3,JNTLAB*2,PLTSYM*i,DIMLAB*24,TITLE*60 80386
CHARACTER UNITS*6

```

```

DATA SEGLAB/'LT ','CT ','UT ','N ','H ','RUL','RLL',
&'RF ','LUL','LLL','LF ','RUA','RLA','LUA','LLA'/
DATA JNTLAB/'P ','W ','NP','HP','RH','RK','RA',
&'LH','LK','LA','RS','RE','LS','LE '/
DATA PLTSYM/'1 ','2 ','3 ','4 ','5 ','6 ','7 ','8 ','9 ','A','B','C','D',
&'E','F','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z'/
DATA JNT/1,2,3,4,1,6,7,1,9,10,3,12,3,14/
DATA UNITS /'MONTHS','YEARS',' ','LB.','N.','%-TILE',
& 'IN.','M.','%-TILE','ENGLISH','METRIC',' '/
DATA CONV /12.,.2248,39.370/,DENS,GRAV/1.12287E-3,32.17405/

```

C DIMENSIONAL DATA LABELS

```

DATA DIMLAB /
& 'AGE ' , 'WEIGHT '
& 'STANDING HEIGHT ' , 'SHOULDER HEIGHT '
& 'ARMPIT HEIGHT ' , 'WAIST HEIGHT '
& 'SEATED HEIGHT ' , 'HEAD LENGTH '
& 'HEAD BREADTH ' , 'HEAD TO CHIN HEIGHT '
& 'NECK CIRCUMFERENCE ' , 'SHOULDER BREADTH '
& 'CHEST DEPTH ' , 'CHEST BREADTH '
& 'WAIST DEPTH ' , 'WAIST BREADTH '
& 'BUTTOCK DEPTH ' , 'HIP BREADTH,STANDING '
& 'SHOULDER TO ELBOW LENGTH' , 'FOREARM-HAND LENGTH '
& 'BICEPS CIRCUMFERENCE ' , 'ELBOW CIRCUMFERENCE '
& 'FOREARM CIRCUMFERENCE ' , 'WRIST CIRCUMFERENCE '

```

```

& 'KNEE HEIGHT, SEATED ' , 'THIGH CIRCUMFERENCE ' ,
& 'UPPER LEG CIRCUMFERENCE ' , 'KNEE CIRCUMFERENCE ' ,
& 'CALF CIRCUMFERENCE ' , 'ANKLE CIRCUMFERENCE ' ,
& 'ANKLE HEIGHT, OUTSIDE ' , 'FOOT BREADTH ' ,
& 'FOOT LENGTH ' /
DATA RANGE /
& 24.00 , 240.0 , 22.27 , 247.6 , 32.01 , 76.54 ,
& 18.50 , 56.50 , 85.00 , 200.0 , 56.93 , 72.05 ,
& 21.50 , 50.50 , 118.0 , 264.0 , 62.17 , 77.64 /
DATA (REGEQ(J, -1), J=1,24) /
& 1.000 , .4547E-12, 1.213 , 25.43 , 5.202 , -158.9 ,
& 1.000 , 0. , 0. , .4547E-12, 0. , 0. ,
& 0. , 0. , 0. , 0. , 0. , 0. ,
& 0. , 0. , 0. , 0. , 0. , 0. /
DATA (REGEQ(J, 0), J=1,24) /
& .6625 , -.7660 , 1.000 , .4547E-12, 3.743 , -122.2 ,
& 0. , 1.000 , 0. , .4547E-12, 1.000 , .4547E-12,
& 3.737 , -111.2 , 1.000 , 0. , .4547E-12, 1.000 ,
& .9095E-12, 4.530 , -142.7 , 1.000 , 0. , .9095E-12/
DATA (REGEQ(J, 1), J=1,24) /
& .1756 , 32.62 , .2314 , 35.58 , 1.000 , .2274E-12,
& 0. , 0. , 1.000 , .2274E-12, .7589E-01, 54.16 ,
& 1.000 , .2274E-12, 0. , 1.000 , .2274E-12, .5850E-01,
& 59.66 , 1.000 , .4547E-12, 2.0. , 1.000 , .4547E-12/
DATA (REGEQ(J, 2), J=1,24) /
& .1518 , 24.79 , .1994 , 27.37 , .8721 , -3.898 ,
& 0. , 0. , .8721 , -3.898 , .7182E-01, 42.77 ,
& .8751 , -3.936 , .7555E-02, .8469 , -3.096 , .5844E-01,
& 47.02 , .8911 , -5.049 , .8583E-02, .8522 , -3.824 /
DATA (REGEQ(J, 3), J=1,24) /
& .1467 , 22.53 , .1916 , 25.19 , .8315 , -4.469 ,
& 0. , -.1217E-01, .8770 , -5.954 , .5840E-01, 39.83 ,
& .8309 , -5.762 , -.6495E-02, .8551 , -6.484 , .4637E-01,
& 43.06 , .8398 , -7.529 , -.3761E-02, .8569 , -8.066 /
DATA (REGEQ(J, 4), J=1,24) /
& .1186 , 17.74 , .1530 , 19.97 , .6822 , -4.726 ,
& 0. , -.1855E-01, .7519 , -7.012 , .5302E-01, 32.73 ,
& .6845 , -4.206 , 0. , .6845 , -4.206 , .3642E-01,
& 35.59 , .7043 , -7.256 , -.6506E-02, .7338 , -8.185 /
DATA (REGEQ(J, 5), J=1,24) /
& .7804E-01, 19.17 , .1047 , 20.34 , .4412 , 4.849 ,
& .5848E-02, .1910E-01, .3393 , 8.114 , .3620E-01, 29.09 ,
& .4225 , 6.734 , .5773E-02, .4010 , 7.376 , .2664E-01,
& 32.06 , .4043 , 8.460 , .4060E-02, .3859 , 9.039 /
DATA (REGEQ(J, 6), J=1,24) /
& .3642E-02, 6.881 , .5571E-02, 6.881 , .2249E-01, 6.110 ,
& -.3975E-02, .2793E-02, .3264E-01, 5.825 , .4897E-02, 6.625 ,
& .3600E-01, 4.950 , .3021E-02, .2471E-01, 5.286 , .3209E-02,
& 7.266 , .2680E-01, 5.952 , .2233E-02, .1668E-01, 6.271 /
DATA (REGEQ(J, 7), J=1,24) /
& .3642E-02, 5.220 , .5337E-02, 5.237 , .2096E-01, 4.532 ,
& -.6471E-03, .3641E-02, .1071E-01, 4.873 , .4089E-02, 5.194 ,

```

```

& .1347E-01, 4.855 , .4089E-02,0. , 5.194 , .3060E-02,
& 5.611 , .1174E-01, 5.322 , .3060E-02,0. , 5.611 /
DATA (REGEQ(J, 8), J=1,24) /
& .8988E-02, 6.582 , .1266E-01, 6.666 , .5203E-01, 4.867 ,
&-.2157E-02, .4550E-02, .4621E-01, 5.080 , .7362E-02, 7.688 ,
& .6816E-01, 4.275 , .3057E-02, .5674E-01, 4.615 , .3143E-02,
& 8.420 , .4659E-01, 5.712 ,0. , .4659E-01, 5.712 /
DATA (REGEQ(J, 9), J=1,24) /
& .2281E-01, 8.421 , .3449E-01, 8.448 , .1301 , 4.177 ,
&-.2812E-02, .3292E-01, .2141E-01, 7.760 , .2316E-01, 10.34 ,
& .8900E-01, 7.607 , .2316E-01,0. , 10.34 , .2444E-01,
& 10.85 , .6527E-01, 10.54 , .2806E-01, -.6185E-01, 14.54 /
DATA (REGEQ(J,10), J=1,24) /
& .3595E-01, 7.530 , .4866E-01, 8.020 , .2040 , .8749 ,
& .4037E-02, .1483E-01, .1271 , 3.354 , .1927E-01, 11.66 ,
& .1246 , 6.160 , .1369E-01, .7344E-01, 7.682 , .1612E-01,
& 13.24 , .1181 , 7.790 , .1253E-01, .6133E-01, 9.578 /
DATA (REGEQ(J,11), J=1,24) /
& .2870E-01, 5.225 , .4249E-01, 5.333 , .1620 , -.2418E-01,
&0. , .3764E-01, .2091E-01, 4.589 , .3412E-01, 4.965 ,
& .7300E-01, 4.650 , .3989E-01, -.7607E-01, 9.086 , .2690E-01,
& 4.984 , .6114E-01, 5.386 , .3174E-01, -.8265E-01, 9.915 /
DATA (REGEQ(J,12), J=1,24) /
& .2699E-01, 5.195 , .3856E-01, 5.411 , .1504 , .3666 ,
& .4513E-02, .2798E-01, .2213E-01, 4.506 , .3185E-01, 6.968 ,
& .8875E-01, 5.357 , .3505E-01, -.4222E-01, 9.255 , .2911E-01,
& 7.854 , .8877E-01, 6.709 , .3254E-01, -.5866E-01, 11.35 /
DATA (REGEQ(J,13), J=1,24) /
& .1284E-01, 5.024 , .2279E-01, 4.763 , .7598E-01, 2.485 ,
&-.8176E-02, .3995E-01, -.3147E-01, 6.097 , .2922E-01, 2.979 ,
& .5358E-01, 3.279 , .3511E-01, -.7761E-01, 7.183 , .3026E-01,
& 3.528 , .5167E-01, 5.174 , .3706E-01, -.1162 , 10.46 /
DATA (REGEQ(J,14), J=1,24) /
& .2718E-01, 5.367 , .3977E-01, 5.508 , .1522 , .4650 ,
& .3925E-02, .3503E-01,0. , 5.406 , .3527E-01, 5.010 ,
& .1060 , 2.734 , .3800E-01, -.3598E-01, 6.959 , .3741E-01,
& 5.694 , .1105 , 4.473 , .4211E-01, -.8025E-01, 10.48 /
DATA (REGEQ(J,15), J=1,24) /
& .2730E-01, 4.812 , .4032E-01, 4.910 , .1528 , -.1220 ,
& .4893E-02, .3874E-01, -.1900E-01, 5.466 , .3419E-01, 3.976 ,
& .7090E-01, 3.803 , .4021E-01, -.7937E-01, 8.275 , .3077E-01,
& 4.095 , .6978E-01, 4.566 , .3632E-01, -.9475E-01, 9.749 /
DATA (REGEQ(J,16), J=1,24) /
& .3552E-01, 5.542 , .4957E-01, 5.905 , .1968 , -.7718 ,
& .1151E-01, .3192E-01, .1642E-01, 5.020 , .4052E-01, 8.610 ,
& .1285 , 5.563 , .4294E-01, -.3192E-01, 10.34 , .2803E-01,
& 9.019 , .1258 , 5.100 , .2803E-01,0. , 9.019 /
DATA (REGEQ(J,17), J=1,24) /
& .3933E-01, 6.369 , .5175E-01, 7.038 , .2237 , -.9239 ,
& .8433E-03,0. , .2193 , -.7898 , .1702E-01, 10.04 ,
& .1962 , -.3163 , .2976E-02, .1851 , .1459E-01, .1256E-01,
& 11.97 , .2087 , -.4158 ,0. , .2087 , -.4158 /

```

```

DATA (RECEQ(J,18), J=1,24) /
& .4900E-01, 8.421 , .6536E-01, 9.182 , .2804 , -.7587 ,
&-.2689E-02, .4490E-02, .2776 , -.6374 , .2092E-01, 13.78 ,
& .2478 , .6324 , .2960E-02, .2367 , .9615 , .1671E-01,
& 16.54 , .2646 , .9735 , .1677E-02, .2570 , 1.213 /
DATA (REGEQ(J,19), J=1,24) /
& .2664E-01, 4.976 , .4173E-01, 4.878 , .1501 , .1143 ,
&0. , .5211E-01, -.4492E-01, 6.478 , .4369E-01, 4.522 ,
& .5386E-01, 6.646 , .5528E-01, -.1527 , 12.79 , .3340E-01,
& 6.608 , .5996E-01, 8.219 , .4067E-01, -.1243 , 14.02 /
DATA (REGEQ(J,20), J=1,24) /
& .2267E-01, 5.152 , .3434E-01, 5.166 , .1290 , .9440 ,
&-.1938E-02, .3358E-01, .1344E-01, 4.738 , .2421E-01, 7.539 ,
& .1221 , 2.826 , .2086E-01, .4420E-01, 5.145 , .1919E-01,
& 8.967 , .1061 , 4.894 , .1767E-01, .2601E-01, 7.416 /
DATA (REGEQ(J,21), J=1,24) /
& .2267E-01, 5.152 , .3434E-01, 5.166 , .1290 , .9440 ,
&-.1938E-02, .3358E-01, .1344E-01, 4.738 , .2643E-01, 5.879 ,
& .7088E-01, 4.719 , .2938E-01, -.3890E-01, 7.986 , .2116E-01,
& 7.411 , .6768E-01, 6.359 , .2341E-01, -.3835E-01, 9.700 /
DATA (REGEQ(J,22), J=1,24) /
& .1145E-01, 3.971 , .1771E-01, 3.951 , .6620E-01, 1.791 ,
&-.3478E-02, .1804E-01, .1674E-01, 3.447 , .1090E-01, 4.503 ,
& .5371E-01, 2.463 , .9530E-02, .1810E-01, 3.523 , .9921E-02,
& 5.202 , .5166E-01, 3.317 , .9386E-02, .9137E-02, 4.657 /
DATA (REGEQ(J,23), J=1,24) /
& .5917E-01, 9.584 , .7857E-01, 10.55 , .3433 , -1.746 ,
&-.8480E-02, -.4527E-02, .4045 , -3.654 , .3002E-01, 14.98 ,
& .3345 , -2.547 , .6469E-02, .3104 , -1.828 , .2467E-01,
& 17.67 , .3558 , -2.889 , .5244E-02, .3321 , -2.141 /
DATA (REGEQ(J,24), J=1,24) /
& .6066E-01, 9.311 , .8944E-01, 9.551 , .3382 , -1.560 ,
& .1018E-01, .8338E-01, -.2709E-01, 10.25 , .8417E-01, 11.13 ,
& .1869 , 9.912 , .9769E-01, -.1781 , 20.77 , .6998E-01,
& 11.01 , .1614 , 11.89 , .8237E-01, -.2117 , 23.64 /
DATA (REGEQ(J,25), J=1,24) /
& .4805E-01, 7.998 , .7076E-01, 8.195 , .2691 , -.6787 ,
& .5930E-02, .6358E-01, 0. , 8.043 , .6412E-01, 9.905 ,
& .1664 , 7.448 , .7187E-01, -.1022 , 15.44 , .3229E-01,
& 9.621 , .1412 , 5.370 , .3229E-01, 0. , 9.621 /
DATA (REGEQ(J,26), J=1,24) /
& .4805E-01, 7.998 , .7076E-01, 8.195 , .2691 , -.6787 ,
& .5930E-02, .6358E-01, 0. , 8.043 , .4406E-01, 8.684 ,
& .1458 , 4.985 , .4605E-01, -.2626E-01, 10.11 , .3334E-01,
& 9.686 , .1512 , 4.918 , .3334E-01, 0. , 9.686 /
DATA (REGEQ(J,27), J=1,24) /
& .3544E-01, 6.685 , .5209E-01, 6.839 , .1998 , .2129 ,
& .1539E-02, .4370E-01, .2825E-01, 5.794 , .4012E-01, 8.335 ,
& .1106 , 6.386 , .4429E-01, -.5493E-01, 11.31 , .3250E-01,
& 9.002 , .8842E-01, 8.469 , .3718E-01, -.3000E-01, 13.77 /
DATA (REGEQ(J,28), J=1,24) /
& .1770E-01, 5.116 , .2659E-01, 5.152 , .1016 , 1.786 ,

```

```

&-.3664E-02, .2284E-01, .3523E-01, 3.995, .1809E-01, 5.999,
& .7603E-01, 3.449, .1720E-01, .1177E-01, 5.361, .1610E-01,
& 6.028, .6599E-01, 4.216, .1665E-01, -.9456E-02, 6.593 /
DATA (REGEQ(J,29), J=1,24) /
& .6837E-02, 1.319, .9232E-02, 1.417, .4015E-01, -.2052E-01,
&-.1302E-02, 0., .4697E-01, -.2305, .3434E-02, 2.230,
& .4163E-01, .9788E-02, 0., .4163E-01, .9788E-02, .4310E-02,
& 4.652, .8736E-01, -.6990, -.1090E-02, .9230E-01, -.8545 /
DATA (REGEQ(J,30), J=1,24) /
& .8742E-02, 2.193, .1221E-01, 2.283, .5052E-01, .5281,
&-.1722E-02, .4119E-02, .4407E-01, .7576, .4638E-02, 2.901,
& .2947E-01, 1.611, .3353E-02, .1694E-01, 1.984, .4136E-02,
& 3.12, .3418E-01, 1.459, .2907E-02, .2101E-01, 1.873 /
DATA (REGEQ(J,31), J=1,24) /
& .2436E-01, 5.384, .3275E-01, 5.742, .1416, .7017,
&-.5757E-02, 0., .1715, -.2125, .1365E-01, 7.738,
& .1299, 1.186, .5296E-02, .1101, 1.775, .1028E-01,
& 8.858, .1309, 1.502, .3570E-02, .1148, 2.012 /

```

C

C YAW, PITCH AND ROLL

DATA YPRLL/

```

& 46*0., 8.1, 0., -8.1, -8.1, 0., 8.1, 28.8, 80., -28.8, -80., 0., 0., 20., REV1/87
& 0., -44.4, 65.6, -80., -44.4, 65.6, -80., -39.6, -70., -39.6, -70., REV1/87
& 4*0., 14.5, -3.3, 0., -14.5, 3.3, 0., 63.3, 0., -63.3, 0. / REV1/87

```

DATA IPIN/0,0,0,0,0,1,0,0,1,0,0,1,0,1/

END

```

C--SUBROUTINE DIALOG-----C
C   DIALOG WRITES PROMPTS TO THE CRT AND ACCEPTS INPUT FROM THE CRT   C
C   DESCRIBING THE SUBJECT FOR WHICH BODY DESCRIPTION DATA IS TO BE   C
C   PRODUCED.  THE PARAMETERS, IPTR, ICNT, AND ISTRT, ARE RETURNED TO THE C
C   MAIN ROUTINE SPECIFYING THE CHOICES MADE BY THE USER               C
C   IPTR--SPECIFIES ROW OF REGEQ CONTAINING FIRST REGRESSION EQUATION   C
C   COEFFICIENT TO BE USED.  IPTR IS SET TO 25 FOR USER SUPPLIED      C
C   DIMENSIONAL DATA.                                                 C
C   ICNT--NUMBER OF COEFFICIENTS IN THE REGRESSION EQUATION TO BE USED. C
C   ISTRT--FIRST BODY DIMENSION TO BE GENERATED:                      C
C   ISTRT =-1 (AGE) FOR CHILD SUBJECTS                                  C
C   ISTRT =0 (WEIGHT) FOR ADULT SUBJECTS OR USER SUPPLIED DATA       C
C   VALUES ARE ALSO READ INTO PRED FOR THE PREDICTING VARIABLES      C
C-----C

```

```

SUBROUTINE DIALOG (IPTR,ICNT,ISTRT,ISUB)
COMMON /DIMS/ DD(-1:31),REGEQ(24,-1:31),PRED(-1:1),
& RANGE(2,-1:1,3),CONV(-1:1)
COMMON /NAMES/ SUBTYP(10),SEGLAB(15),JNTLAB(14),PLTSYM(29),
& DIMNM(-1:31),TITLE,UNITS(3,-1:2)
CHARACTER SUBTYP*35,SEGLAB*3,JNTLAB*2,PLTSYM*1,ANS*1,DIMNM*24
CHARACTER TITLE*60,UNITS*6
CHARACTER*12 FILE1
INTEGER IPTMAT(-1:2,3),IPT2(13)
EQUIVALENCE (IPTMAT,IPT2)
DATA IPT2 /1,3,5,7,11,11,13,15,18,18,20,22,25/

```

80386

```

WRITE (5,22)
READ (7,333) TITLE
WRITE (5,333) ' '//TITLE
WRITE (9,333) '1'//TITLE

```

```

10 READ (2,*) NUMTYP
   READ (2,*) (SUBTYP(I),I=1,NUMTYP)
   WRITE (5,33) (I,SUBTYP(I),I=1,NUMTYP)
   WRITE (5,44)
   READ (7,222) ISUB
   IF (ISUB.GT.NUMTYP .OR. ISUB.LT.1) GOTO 10
   WRITE (9,333) '0'//SUBTYP(ISUB)
   ISTRT = -1
   IF (ISUB.GT.1) ISTRT = 0
   IF (ISUB.EQ.4) GOTO 100
   IF (ISUB.GT.4) THEN
     IPTR = 0
     ICNT = 0
     ISTRT = 0
   RETURN
   ENDIF

20  WRITE (5,33) (I+1,DIMNM(I),I=ISTRT,1),3,'ALL OF THE ABOVE'
   WRITE (5,55)
   READ (7,222) IDIM

```

```

IDIM = IDIM - 1
IF (IDIM.LT.1STRT .OR. IDIM.GT.2) GOTO 20
IF (IDIM.GE.2) THEN
  LPSTRT = 1STRT
  LPEND = 1
ELSE
  LPSTRT = IDIM
  LPEND = LPSTRT
ENDIF

WRITE (9,555)
DO 50 I = LPSTRT,LPEND
  CALL ASKUN (DIMNM(I),UNITS(1,I),ICHOS,1STRT+3)
  IF (ICHOS.GE.3) THEN
    CALL PTILE (ISUB,I,PRED(I))
  ELSE
    TCONV = 1.
    IF (ICHOS.GE.2) TCONV = CONV(I)
40  WRITE (5,77) DIMNM(I),(RANGE(J,I,ISUB)/TCONV,J=1,2)
    READ (5,444) PRED(I)
    WRITE (9,666) DIMNM(I),PRED(I),UNITS(ICHOS,I)
    PRED(I) = PRED(I)*TCONV
    IF ((PRED(I)-RANGE(1,I,ISUB))*(PRED(I)-RANGE(2,I,ISUB))
&      .GT. 0.) GOTO 40
50  ENDIF
  CONTINUE

IF (IDIM.LT.2) THEN
  PRED(-1) = PRED(IDIM)
ELSE
  IF (ISUB.GT.1) THEN
60  DO 60 I=-1,0
    PRED(I) = PRED(I+1)
  ENDIF
ENDIF
IPTR = IPTMAT(IDIM,ISUB)
ICNT = IPTMAT(IDIM+1,ISUB) - IPTR - 1
RETURN

100 WRITE (5,88)
READ (7,333) ANS
IF (ANS.EQ.'N'.OR.ANS.EQ.'n') STOP 'FROM DIALOG' 80386
IF (ANS.NE.'Y'.AND.ANS.NE.'y') GOTO 100 80386
CLOSE (UNIT=4) 80386
OPEN (UNIT=4,FILE='GEBOD11.DIR',STATUS='OLD') 80386
READ (4,'(///A12)')FILE1 80386
OPEN (UNIT=1,FILE=FILE1,STATUS='OLD') 80386
READ (1,99) (DD(I),I=0,31)
CLOSE (UNIT=1) 80386
IPTR = 25
CALL ASKUN(SUBTYP(4),UNITS(1,2),ICHOS,2)
IF (ICHOS.GE.2) THEN

```

```

        DD(0) = DD(0)*CONV(0)
        DO 120 I=1,31
120      DD(I) = DD(I)*CONV(I)
        ENDIF
        RETURN

22      FORMAT (// ' PROGRAM GEBOD'/' GEBOD GENERATES BODY DESCRIPTION',
&          ' DATA'/' SUITABLE FOR INPUT TO THE ATB MODEL'/
&          ' PLEASE ENTER A DESCRIPTION OF THE SUBJECT' ,
&          ' (<60 CHARS.)')
33      FORMAT (//(I12,')',3X,A/)
44      FORMAT (' ENTER NUMBER CORRESPONDING TO DESIRED SUBJECT TYPE')
55      FORMAT (' ENTER NUMBER CORRESPONDING TO '/' PREDICTING',
&          ' DIMENSION(S) TO BE SUPPLIED')
77      FORMAT (' ENTER VALUE FOR ',A24/' IN THE RANGE ',G10.4,',',G10.4)
88      FORMAT (// ' TO UTILIZE EXTERNAL DIMENSIONAL DATA, THE DATA MUST',
&          ' BE'/' IN THE PROPER FORMAT AND IN A FILE ASSIGNED TO UNIT 1'/
&          ' HAS THE ABOVE BEEN SATISFIED (Y/N) ?')
99      FORMAT (8F10.0)
222     FORMAT (BN,I60)
333     FORMAT (A)
444     FORMAT (G60.0)
555     FORMAT ('OSELECTED BODY DIMENSIONS'/)
666     FORMAT (A40,G15.4,A6)
        END

```



```

C--SUBROUTINE ASKUN-----C
C   ASKUN ASKS USER CHOICE OF UNITS TO BE USED WITH SOME INPUT VARIABLE   C
C   OR THE OUTPUT OF RESULTS.  THE PARAMETERS ARE THE FOLLOWING:           C
C   VARB--CHARACTER STRING DESCRIBING WHAT UNITS ARE TO BE SELECTED FOR   C
C   UNITS--UNITS AVAILABLE TO CHOOSE FROM                                  C
C   ICHOS--RETURNS WHICH OF THE UNITS WAS CHOSEN                          C
C   ICNT--NUMBER OF DIFFERENT UNITS AVAILABLE TO CHOOSE FROM              C
C-----C

```

```

SUBROUTINE ASKUN (VARB,UNITS,ICHOS,ICNT)
CHARACTER VARB*(*),UNITS(ICNT)*6,UNITIN*6

```

```

30  WRITE (5,66) VARB
    WRITE (5,55) (I,UNITS(I),I=1,ICNT)
    READ (7,44) ICHOS
    IF ( (ICHOS-1)*(ICHOS-ICNT) .GT. 0)   GOTO 30

    RETURN

44  FORMAT (BN,I60)
55  FORMAT (I10,' '),5X,A)
66  FORMAT (//' SELECT UNITS FOR ',A)
    END

```

```

C--SUBROUTINE PTILE-----C
C   PTILE COMPUTES A PERCENTILE POINT, SPECIFIED BY THE USER, OF A BODY C
C   DIMENSION USED AS A PREDICTOR, BY ASSUMING THE DIMENSION IS C
C   NORMALLY DISTRIBUTED. THE PARAMETERS ARE THE FOLLOWING: C
C   ISUB--SPECIFIES SUBJECT TYPE, AND THUS WHICH POSITION OF THE C
C   MEAN AND STANDARD ARRAYS TO USE. C
C   IDIM--THE DIMENSION A PERCENTILE POINT IS TO BE COMPUTED FOR C
C   PRED--RETURNS THE COMPUTED PERCENTILE POINT C
C   NOTE: THIS ROUTINE REQUIRES USE OF IMSL ROUTINE MDNRIS C
C-----C

```

```

SUBROUTINE PTILE (ISUB, IDIM, PRED)
COMMON /NAMES/ SUBTYP(10), SEGLAB(15), JNTLAB(14), PLTSYM(29),
& DIMNM(-1:31), TITLE, UNITS(3, -1:2) 80386
REAL MEAN(0:1, 2:3), STDEV(0:1, 2:3)
CHARACTER SUBTYP*35, SEGLAB*3, JNTLAB*2, PLTSYM*1, DIMNM*24 80386
CHARACTER TITLE*60, UNITS*6
DATA MEAN, STDEV/127.24, 63.82, 173.54, 69.82, 16.57, 2.36, 21.42, 2.44/

10 WRITE (5, 111) DIMNM(IDIM)
READ (7, 99) PCENT
IF ( (PCENT-1.)*(PCENT-100.) .GE. 0.) GOTO 10
WRITE (9, 222) DIMNM(IDIM), PCENT
PCENT = PCENT/100.

CALL NDTRI (PCENT, X, D, IER)
PRED = MEAN(IDIM, ISUB) + X*STDEV(IDIM, ISUB)

RETURN
99 FORMAT (F40.0)
111 FORMAT (' ENTER DESIRED PERCENTILE FOR ', A/
& ' (A REAL VALUE BETWEEN 0. AND 100.)')
222 FORMAT (A40, G15.4, ' %-TILE')
END

```

```

C--SUBROUTINE NDTRI-----C
C   NDTRI COMPUTES X (THE OUTPUT ARGUMENT SUCH THAT P=Y=THE      C
C   PROBABILITY THAT U, THE RANDOM VARIABLE, IS LESS THAN OR EQUAL  C
C   TO X.), D (THE OUTPUT DENSITY, F(X).), AND IER (THE OUTPUT      C
C   ERROR CODE.) USING P (THE INPUT PROBABILITY.).                  C
C-----C

```

```

      SUBROUTINE NDTRI(P,X,D,IE)
C
      IE=0
      X=.999999E+38                      80386
      D=X
      IF(P)1,4,2
1     IE=-1
      GO TO 12
2     IF (P-1.0)7,5,1
4     X=-.999999E+38                      80386
5     D=0.0
      GO TO 12
C
7     D=P
      IF(D-0.5)9,9,8
8     D=1.0-D
9     T2=ALOG(1.0/(D*D))
      T=SQRT(T2)
      X=T-(2.515517+0.802853*T+0.010328*T2)/(1.0+1.432788*T+0.189269*T2
& +0.001308*T*T2)
      IF(P-0.5)10,10,11
10    X=-X
11    D=0.3989423*EXP(-X*X/2.0)
12    RETURN
      END

```

C--SUBROUTINE CONTAC-----C
 C CONTAC COMPUTES JOINT LOCATIONS AND CONTACT ELLIPSOID SEMIAXES. C
 C-----C

```

SUBROUTINE CONTAC
COMMON /FLOAT/ PI,PI2,PI4,DENS,GRAV
COMMON /SGMNTS/ AN(15),BN(15),CN(15),RMN(15),PHI(3,15),XYZCG(15,3)
COMMON /JNTS/ RNJ(3,28),JNT(14),YPRLL(14,6),IPIN(14)
COMMON /DIMS/ DD(-1:31),REGEQ(24,-1:31),PRED(3),
&             RANGE(2,-1:1,3),CONV(-1:1)

```

80386
80386
80386
80386

C THE CENTERS OF GRAVITY AND JOINT X,Y COORDINATES ARE ALL SET TO ZERO.

```

DO 10 J=1,3
DO 10 I=1,15
10 XYZCG(I,J)=0.0
DO 20 J=1,2
DO 20 I=1,28
20 RNJ(J,I)=0.0

```

C NON-ZERO X,Y JOINT COORDINATES ARE COMPUTED.

```

RNJ(1,21)=DD(29)*0.5
RNJ(1,24)=RNJ(1,21)
RNJ(2,5)=0.5*(DD(16)-DD(24)/PI)
RNJ(2,8)=-RNJ(2,5)
RNJ(2,11)=0.5*(DD(10)-DD(19)/PI)
RNJ(2,13)=-RNJ(2,11)

```

C SMW IS ONE HALF THE HEIGHT OF THE ABDOMEN SEGMENT.

C OLAP IS USED AS AN OVERLAP AMOUNT BETWEEN CERTAIN CONTACT ELLIPSOIDS.

```

SMW = (DD(2) - DD(4))/10.
OLAP = DD(9)/PI

```

C Z COORDINATES OF JOINTS ARE COMPUTED RELATIVE TO SEGMENT JNT(J),

C WHERE J IS THE SECOND SUBSCRIPT OF RNJ

```

RNJ(3,1) = (DD(1) - DD(5) - DD(4) + SMW)/2.
RNJ(3,2) = -SMW
RNJ(3,3) = -9.*SMW/2.
RNJ(3,4) = (DD(8) + DD(2) - DD(1) + OLAP/2.)/2.
RNJ(3,5) = (DD(4) - SMW - DD(1) + DD(5) - DD(24)/PI)/2.
RNJ(3,6)=0.5*(DD(1)-DD(5)-DD(23)+DD(24)/PI)
RNJ(3,7)=0.5*(DD(23)-DD(29)-DD(28)/PI2)
RNJ(3,8) = RNJ(3,5)
RNJ(3,9)=RNJ(3,6)
RNJ(3,10)=RNJ(3,7)
RNJ(3,11)=-DD(2)+DD(3)+DD(19)/PI2
RNJ(3,12)=0.5*(DD(17)-DD(20)/PI)
RNJ(3,13)=RNJ(3,11)
RNJ(3,14)=RNJ(3,12)

```

C Z COORDINATES OF JOINTS ARE COMPUTED RELATIVE TO SEGMENT J-13,

C WHERE J IS THE SECOND SUBSCRIPT OF RNJ

```

RNJ(3,15) = SMW

```

RNJ(3,16) = -RNJ(3,3)
 RNJ(3,17) = (DD(1) - DD(8) - DD(2) - OLAP/2.)/2.
 RNJ(3,18) = DD(8) /2. + OLAP/4.
 RNJ(3,19) = 0.5*(-DD(1)+DD(5)+DD(23)-DD(26)/PI)
 RNJ(3,20) = (DD(29) - DD(23) - DD(28)/PI2 + DD(26)/PI)/2.
 RNJ(3,21) = DD(28)/PI2 - DD(31)/2.
 RNJ(3,22) = RNJ(3,19)
 RNJ(3,23) = RNJ(3,20)
 RNJ(3,24) = RNJ(3,21)
 RNJ(3,25) = 0.5*(DD(19)/PI - DD(17))
 RNJ(3,26) = 0.5*(DD(20)/PI - DD(18))
 RNJ(3,27) = RNJ(3,25)
 RNJ(3,28) = RNJ(3,26)

C SEGMENT CONTACT ELLIPSOID X SEMIAXES ARE COMPUTED.

AN(5) = DD(6)*0.5
 AN(4) = DD(9)/PI2
 AN(3) = DD(11)*0.5
 AN(2) = DD(13)*0.5
 AN(1) = DD(15)*0.5
 AN(6) = (DD(24)+DD(25))/PI4
 AN(7) = DD(27)/PI2
 AN(8) = DD(29)*0.5
 AN(9) = AN(6)
 AN(10) = AN(7)
 AN(11) = AN(8)
 AN(12) = DD(19)/PI2
 AN(13) = DD(21)/PI2
 AN(14) = AN(12)
 AN(15) = AN(13)

C SEGMENT CONTACT ELLIPSOID Y SEMIAXES ARE COMPUTED.

BN(5) = DD(7)*0.5
 BN(4) = AN(4)
 BN(3) = DD(12)*0.5
 BN(2) = DD(14)*0.5
 BN(1) = DD(16)*0.5
 BN(6) = AN(6)
 BN(7) = AN(7)
 BN(8) = DD(30)*0.5
 BN(9) = AN(9)
 BN(10) = AN(10)
 BN(11) = BN(8)
 BN(12) = AN(12)
 BN(13) = AN(13)
 BN(14) = AN(14)
 BN(15) = AN(15)

C SEGMENT CONTACT ELLIPSOID Z SEMIAXES ARE COMPUTED.

CN(5) = (DD(8) + OLAP/2.)/2.
 CN(4) = (DD(1) - DD(8) - DD(2) + OLAP/2.)/2.
 CN(3) = 9.*SMW/2.

CN(2) = SMW + OLAP/2.
 CN(1) = (DD(4)+DD(5)-DD(1) - SMW)*0.5
 CN(6)=(DD(1)-DD(5)-DD(23)+DD(24)/PI+DD(26)/PI)*0.5
 CN(7)=(DD(23)-DD(29)+DD(28)/PI2)*0.5
 CN(8)=DD(31)*0.5
 CN(9)=CN(6)
 CN(10)=CN(7)
 CN(11)=CN(8)
 CN(12)=DD(17)*0.5
 CN(13)=DD(18)*0.5
 CN(14)=CN(12)
 CN(15)=CN(13)

C THE GLOBAL LOCATION OF SEGMENT CENTERS IS PLACED IN XYZCG FOR
 C SEGMENTS THAT WILL USE ELLIP IN COMPUTATION OF MASS AND
 C MOMENTS OF INERTIA.

XYZCG(1,3) = (DD(4) - SMW + D(1) - DD(5))/2.
 XYZCG(2,3) = DD(4)
 XYZCG(3,3) = DD(2) - 9.*SMW/2.
 XYZCG(8,3) = DD(29) - DD(31)/2. + DD(28)/PI2
 XYZCG(11,3) = XYZCG(8,3)

RETURN
 END

```

C--SUBROUTINE SEGMAS-----C
C   SEGMAS COMPUTES APPROXIMATE SEGMENT MASSES.  WTCOR IS THEN COMPUTED   C
C   TO ADJUST BODY DENSITY, DENS, SO THAT TOTAL BODY MASS AGREES        C
C   WITH DD(0).  SEGMENT MASSES ARE THEN RECOMPUTED USING WTCOR.         C
C-----C

```

```

SUBROUTINE SEGMAS
C   EXTERNAL TORSO, FEET 80386
COMMON /FLOAT/ PI,PI2,PI4,DENS,GRAV
COMMON /CYL/ ZEES(0:15),ZH(15),NSEG(15),CGZ(15),WTCOR
COMMON /JNTS/ RNJ(3,28),JNT(14),YPRLL(14,6),IPIN(14) 80386
COMMON /SGMNTS/ ABCN(15,3),RMN(15),PHI(45),XYCG(30),ZCG(15)
COMMON /DIMS/ DD(-1:31),REGEQ(24,-1:31),PRED(3), 80386
& RANGE(2,-1:1,3),CONV(-1:1) 80386
REAL AN(15),BN(15),CN(15)
EQUIVALENCE (AN,ABCN),(BN,ABCN(1,2)),(CN,ABCN(1,3))
SUMM=0.0

```

```

C   POSITIONS 0 THRU 3 OF ZEES DEFINE THE VERTICAL LIMITS OF TORSO
C   SEGMENTS.  10 THRU 11 DEFINE THE LIMITS OF THE FEET.
SMW = (DD(2) - DD(4))/10.
ZEES(0) = DD(1) - DD(5)
ZEES(1) = DD(4) - SMW
ZEES(2) = DD(4) + SMW
ZEES(3) = DD(2)
ZEES(11) = DD(29) + DD(28)/PI2
ZEES(10) = ZEES(11) - DD(31)
DO 5 I = 1,3

```

```

    CALL ELLIP (I,1,RMN(I)) 80386
    ZCG(I) = CGZ(I) - ZCG(I)

```

```

5    SUMM = SUMM + RMN(I)
    CALL ELLIP (11,2,RMN(11)) 80386
    SUMM = SUMM + 2.*RMN(11)
    RMN(8) = RMN(11)
    ZCG(11) = CGZ(11) - ZCG(11)
    ZCG(8) = ZCG(11)

```

```

DO 7 I = 1,3
7    RNJ(3,I) = RNJ(3,i) + ZCG(I)
    RNJ(3,15) = ZCG(2) + RNJ(3,15)
    RNJ(3,16) = ZCG(3) + RNJ(3,16)
    RNJ(3,5) = ZCG(1) + RNJ(3,5)
    RNJ(3,8) = ZCG(1) + RNJ(3,8)
    RNJ(3,11) = RNJ(3,11) + ZCG(3)
    RNJ(3,13) = RNJ(3,13) + ZCG(3)
    RNJ(3,21) = RNJ(3,21) + ZCG(8)
    RNJ(3,24) = RNJ(3,24) + ZCG(11)
DO 10 I = 4,7
    RMN(I) = 4.0/3.0*PI*AN(I)*BN(I)*CN(I)*DENS
    SUMM = SUMM+RMN(I)
    IF (I.GT.5) THEN
        SUMM = SUMM + RMN(I)
        RMN(I+3) = RMN(I)

```

```

    ENDIF
10  CONTINUE
    DO 15 I = 12,13
        RMN(I) = 4.0/3.0*PI*AN(I)*BN(I)*CN(I)*DENS
        SUMM = SUMM+2.*RMN(I)
15  RMN(I+2) = RMN(I)
C
C   THE FOLLOWING ADJUSTS SEGMENT MASSES TO TOTAL BODY WEIGHT
C
    WTCOR = DD(0)/SUMM/GRAV
    DO 20 I=1,15
        RMN(I)=RMN(I)*WTCOR
20  CONTINUE
    RETURN
    END

```


C--SUBROUTINE SGINER-----C
 C SGINER COMPUTES SEGMENT MOMENTS OF INERTIA. C
 C-----C

```

SUBROUTINE SGINER
C  EXTERNAL TORSO, FEET                                80386
COMMON /FLOAT/ PI,PI2,PI4,DENS,GRAV                  80386
COMMON /SGMNTS/ AN(15),BN(15),CN(15),RMN(15),PHI(3,15),XYZCG(15,3)80386

DO 5 I=1,3
5  CALL ELLPMI (I,1,PHI(1,I))                          80386
DO 10 I=4,7
TEMP=RMN(I)/5.0
A2=AN(I)*AN(I)
B2=BN(I)*BN(I)
C2=CN(I)*CN(I)
PHI(1,I)=TEMP*(B2+C2)
PHI(2,I)=TEMP*(A2+C2)
PHI(3,I)=TEMP*(A2+B2)
IF (I.GT.5) THEN
DO 7 J = 1,3
7  PHI(J,I+3) = PHI(J,I)
ENDIF
10 CONTINUE
DO 20 I=12,13
TEMP=RMN(I)/5.0
A2=AN(I)*AN(I)
B2=BN(I)*BN(I)
C2=CN(I)*CN(I)
PHI(1,I)=TEMP*(B2+C2)
PHI(2,I)=TEMP*(A2+C2)
PHI(3,I)=TEMP*(A2+B2)
DO 15 J = 1,3
15  PHI(J,I+2) = PHI(J,I)
20 CONTINUE
CALL ELLPMI (11,2,PHI(1,11))                          80386
DO 25 J=1,3
25  PHI(J,8) = PHI(J,11)
RETURN
END

```

```

C--SUBROUTINE ELLIP-----C
C   ELLIP COMPUTES MASS AND CENTER OF GRAVITY LOCATION OF A RIGHT   C
C   ELLIPTICAL SOLID.  THE PARAMETERS ARE:                          C
C   ISEG--SPECIFIES SEGMENT BEING MODELLED BY RIGHT ELLIPTICAL SOLID C
C   IABEVAL--SPECIFIES SUBROUTINE TO BE USED TO COMPUTE SEMIAXES OF THE C
C   THE SOLID AT SPECIFIC Z HEIGHTS.                                C
C   TMASS--RETURNS THE MASS COMPUTED FOR THE SOLID.                 C
C-----C

```

```

SUBROUTINE ELLIP (ISEG,IABEVAL,TMASS)                                80386
COMMON /CYL/ ZEES(0:15),ZH(15),NSEG(15),CGZ(15),WTCOR
COMMON /FLOAT/ PI,PI2,PI4,DENS,GRAV
REAL IXX,IXXP,IYY,IYYP,IZZ,IZZP
REAL MASS
REAL MI(3)
TMASSO=0.0
ZO = ZEES(ISEG-1)
Z1 = ZEES(ISEG)
NSEG(ISEG)=0.0
10  NSEG(ISEG)=NSEG(ISEG)+10
TCGN=0.0
TMASS=0.0
ZH(ISEG)=(Z1-ZO)/FLOAT(NSEG(ISEG))
DO 100 I=1,NSEG(ISEG)
  Z = ZO + (FLOAT(I-1)+.5)*ZH(ISEG)
  IF (IABEVAL.EQ.1) THEN                                           80386
    CALL TORSO(Z,A,B)                                             80386
  ELSE IF (IABEVAL.EQ.2) THEN                                     80386
    CALL FEET(Z,A,B)                                             80386
  ELSE                                                             80386
    STOP 'FROM ELLIP (IABEVAL)'                                   80386
  END IF                                                         80386
  MASS=PI*A*B*ZH(ISEG)*DENS
  CGN=Z
  CGN=CGN*MASS
  TMASS=TMASS+MASS
100 TCGN=TCGN+CGN
TMASSN=TMASS
IF (ABS(TMASSN-TMASSO)/TMASSN.GT.5.E-5) THEN
  TMASSO = TMASSN
  GOTO10
ENDIF
CGZ(ISEG)=TCGN/TMASS
RETURN

```

```

C ENTRY ELLPMI
C   ELLPMI COMPUTES MOMENTS OF INERTIA OF A RIGHT ELLIPTICAL SOLID, BASED
C   ON THE NUMERICAL INTEGRATION DETERMINED BY ELLIP.  THE FIRST TWO
C   PARAMETERS ARE THE SAME AS THOSE USED BY ELLIP.  THE THIRD, MI,
C   RETURNS THE MOMENTS OF INERTIA OF THE SOLID.

```

```

ENTRY ELLPMI (ISEG,IABEVAL,MI)                                     80386

```

```

DO 5 I=1,3
5  MI(I)=0.0
   ZO = ZEES(ISEG-1)
   DO 200 I=1,NSEG(ISEG)
   Z = ZO + (FLOAT(I-1)+.5)*ZH(ISEG)
   IF (IABEVAL.EQ.1) THEN
   CALL TORSO(Z,A,B)
   ELSE IF (IABEVAL.EQ.2) THEN
   CALL FEET(Z,A,B)
   ELSE
   STOP 'FROM ELLPMI (IABEVAL)'
   END IF
   MASS=PI*A*B*ZH(ISEG)*DENS*WTCOR
   IXX=(MASS/12)*(3.0*B**2+ZH(ISEG)**2)
   IYY=(MASS/12)*(3.0*A**2+ZH(ISEG)**2)
   IZZ=(MASS/4.0)*(A**2+B**2)
   DIST=Z-CGZ(ISEG)
   IXXP=IXX+MASS*DIST**2
   IYYP=IYY+MASS*DIST**2
   IZZP=IZZ
   MI(1)=MI(1)+IXXP
   MI(2)=MI(2)+IYYP
   MI(3)=MI(3)+IZZP
200  Z=Z+ZH(ISEG)
   RETURN
   END

```

```

80386
80386
80386
80386
80386
80386
80386

```

```

C--SUBROUTINE TORSO-----C
C   TORSO COMPUTES SEMIAXIS VALUES FOR THE RIGHT ELLIPTICAL MODEL OF THE   C
C   THREE TORSO SEGMENTS.  THE PARAMETERS ARE THE FOLLOWING:               C
C   Z--VERTICAL LOCATION AT WHICH THE SEMIAXES ARE TO BE COMPUTED         C
C   X--RETURNS X SEMIAXIS                                                 C
C   Y--RETURNS Y SEMIAXIS                                                 C
C-----C

```

```

SUBROUTINE TORSO (Z,X,Y)
COMMON /DIMS/ D(-1:31),REGEQ(24,-1:31),PRED(3),
&           RANGE(2,-1:1,3),CONV(-1:1)
COMMON /FLOAT/ PI,PI2,PI4,DENS,GRAV
REAL H2,H3,H4,H5,H6

```

```

80386
80386
80386

```

```

C   H2--TOP OF SOLID
C   H3--BREAK BETWEEN TOP SEMI-ELLIPSOID AND TOP FRUSTRUM
C   H4--BREAK BETWEEN TWO FRUSTRUM
C   H5--BREAK BETWEEN BOTTOM FRUSTRUM AND BOTTOM SEMI-ELLIPSOID
C   H6--BOTTOM OF SOLID

```

```

H2=D(2)
H3=D(3)
H4=D(4)
H5=(D(1)-D(5))+D(24)/PI2
H6=(D(1)-D(5))
IF (Z.LT.H4)THEN
  IF (Z.LT.H5)THEN
    IF (Z.LT.H6)STOP 'Z IS LESS THAN MIN.'
    Y=SQRT(1. - ((Z - H5)/D(24)/PI2)**2)
    X = D(15)/2.*Y
    Y = D(16)/2.*Y
  ELSE
    Y=(Z-D(4))/(H5-D(4))
    X = (Y*(D(15)-D(13)) +D(13))/2.
    Y = (Y*(D(16)-D(14)) +D(14))/2.
  END IF
ELSE
  IF (Z.LT.H3)THEN
    Y=(Z-D(3))/(D(4)-D(3))
    X= (Y*(D(13)-D(11))+D(11))/2.
    Y= (Y*(D(14)-D(12))+D(12))/2.
  ELSE
    IF (Z.GT.H2) STOP 'Z IS GREATER THAN MAX.'
    Y=SQRT(1. - ((Z - D(3))/(D(2) - D(3)) )**2)
    X= Y*D(11)/2.
    Y= Y*D(12)/2.
  END IF
END IF
END

```

```

C--SUBROUTINE FEET-----C
C   FEET COMPUTES SEMIAXIS VALUES FOR THE RIGHT ELLIPTICAL MODEL OF THE   C
C   FEET.  THE PARAMETERS ARE THE FOLLOWING:                               C
C   Z--Z LOCATION (TOE TO HEEL) AT WHICH SEMIAXES ARE TO BE COMPUTED     C
C   X--RETURNS X SEMIAXIS                                                 C
C   Y--RETURNS Y SEMIAXIS                                                 C
C-----C

```

SUBROUTINE FEET (Z,X,Y)

COMMON /CYL/ ZEES(0:15),ZH(15),HSEG(15),CGZ(15),WTCOR

COMMON /DIHS/ DD(-1:31),RECEQ(24,-1:31),FRED(3),

80386

& RANGE(2,-1:1,3),CONV(-1:1)

80386

COMMON /FLOAT/ PI,PI2,PI4,DENS,GRAV

80386

C H2--TOP OF SOLID

C H1--BREAK BETWEEN ELLIPTICAL CYLINDER AND FRUSTRUM

C H0--BOTTOM OF SOLID

ANKR = DD(28)/PI2

H2 = DD(29) + ANKR

H1 = DD(29) - ANKR

H0 = H2 - DD(31)

IF (Z.GT.H1) THEN

IF (Z.GT.H2) STOP 'Z VALUE TOO LARGE IN FEET'

X = ANKR

Y = ANKR

ELSE

IF (Z.LT.H0) STOP 'Z VALUE TOO SMALL IN FEET'

X = ANKR/(DD(31) - 2.*ANKR)*(Z - DD(29) + DD(31) - ANKR)

Y = (ANKR - DD(30)/2.)/(DD(31) - 2.*ANKR)

Y = Y*(Z - DD(29) + ANKR) + ANKR

ENDIF

RETURN

END

C--SUBROUTINE RESULTS-----C
 C RESULTS WRITES OUT THE COMPUTED BODY DESCRIPTION DATA. C
 C-----C

```

SUBROUTINE RESULTS (ISTRT)
COMMON /SGMNTS/ A(15),B(15),C(15),RMN(15),PHI(3,15),
& XCG(15),YCG(15),ZCG(15)
COMMON /JNTS/ RNJ(3,14,2),JNT(14),YPRLL(14,6),IPIN(14)
COMMON /DIMS/ DD(-1:31),REGEQ(24,-1:31),PRED(3),
& RANGE(2,-1:1,3),CONV(-1:1)
COMMON /NAMES/ SUBTYP(10),SEGLAB(15),JNTLAB(14),PLTSYM(29),
& DIMLAB(-1:31),TITLE,UNITS(3,-1:2)
COMMON /CYL/ ZEES(0:15),ZH(15),NSEG(15),CGZ(15),WTCOR
CHARACTER SUBTYP*35,SEGLAB*3,JNTLAB*2,PLTSYM*1,DIMLAB*24,TITLE*60
CHARACTER ANS*1,UNITS*6,LABL1(2)*6,LABL2(2)*12,LABL3(2)*2,
& LABL4(2)*4,VARB*6
INTEGER ID2(14,3)
DATA LABL1,LABL2,LABL3,LABL4 /'WEIGHT','MASS','LB-SEC**2-IN',
& 'N-SEC**2-M','IN.','M.','LB.','N.'/,VARB/'OUTPUT'/
DATA ID2/6*0,2,0,0,2,0,2,0,2,6*0.3,0,0,3,0,3,0,3.6*0,
& 1,0,0,1,0,1,0,1/

WRITE (9,11)
CALL ASKUN (VARB,UNITS(1,2),IUN,2)
CALL CNVRT (IUN)
IF (ISTRT.LT.0) WRITE (9,22) -1,DIMLAB(-1),DD(-1),UNITS(1,-1)
WRITE (9,22) 0,DIMLAB(0),DD(0),UNITS(IUN,0)
WRITE (9,22) (I,DIMLAB(I),DD(I),UNITS(IUN,1),I=1,31)
WRITE (9,66) WTCOR

WRITE (9,44) '1'//TITLE
WRITE(9,1170) LABL1(IUN),LABL2(IUN),LABL3(IUN),LABL3(IUN),
& LABL4(IUN)
WRITE(9,1180) (I,SEGLAB(I),PLTSYM(I),RMN(I),(PHI(J,I),J=1,3),
& A(I),B(I),C(I),XCG(I),YCG(I),ZCG(I),I=1,15)
WRITE(9,1190) LABL3(IUN),LABL3(IUN)
WRITE(9,1200) (I,JNTLAB(I),PLTSYM(I+15),JNT(I),IPIN(I),
& ((RNJ(J,I,K),J=1,3),K=1,2),(YPRLL(I,I),I=1,6),I=1,14)
WRITE (9,201)

10 WR_ ,33)
READ ,44) ANS
IF (ANS.EQ.'Y'.OR.ANS.EQ.'y') THEN
WRITE (3,77) TITLE(1,52)
WRITE(3,1300) (SEGLAB(I),PLTSYM(I),RMN(I),(PHI(J,I),J=1,3),
& A(I),B(I),C(I),XCG(I),YCG(I),ZCG(I),I=1,15)
C WRITE(3,1320) (JNTLAB(I),PLTSYM(I+15),JNT(I),IPIN(I),
& ((RNJ(J,I,K),J=1,3),K=1,2),
& (YPRLL(I,J),J=1,6),(ID2(I,J),J=1,3),I=1,14)
ELSE
IF (ANS.NE.'N'.AND.ANS.NE.'n') GOTO 10
ENDIF

```

```

20  WRITE (5,55)
    READ (7,44) ANS
    IF (ANS.EQ.'N'.OR.ANS.EQ.'n') STOP 'FROM RESULTS'
    IF (ANS.NE.'Y'.AND.ANS.NE.'y') GOTO 20
    RETURN
80386
80386

11  FORMAT ('-COMPUTED BODY DIMENSIONS'/)
22  FORMAT (I10,A30,G15.4,A6)
33  FORMAT (//' IS ATB MODEL FORMATTED OUTPUT DESIRED (Y/N) ?')
44  FORMAT (A)
55  FORMAT (//' IS IT DESIRED TO PRODUCE ANOTHER BODY DESCRIPTION ',
    &      'DATA SET (Y/N) ?')
66  FORMAT ('OWEIGHT CORRECTION FACTOR =',F10.3)
77  FORMAT (4X,'15',4X,'14',8X,A52,'CARD B.1')
1170 FORMAT(
    1H0,"CRASH VICTIM PARAMETERS (3-D)"/1H0,33X,
    1 25HSEGMENT MOMENT OF INERTIA,
    2 19X,25HSEGMENT CONTACT ELLIPSOID/5X,7HSEGMENT,7X,A6
    3,15X,(' ',A12,')',17X,10HSEMIAXIS (,A2,')',14X,'CENTER (',
    4 A2,')'/3X,12HNO. SYM PLOT,4X,(' ',A4,')',10X,1HX,10X,1HY,10X,
    51HZ,12X,1HX,6X,1HY,6X,1HZ,12X,1HX,7X,1HY,6X,1HZ/1H )
1180 FORMAT ((3X,I2,2X,A4,A2,6X,F6.2,3X,3(4X,F7.5),5X,3(2X,F5.2),5X,3
    1(3X,F5.2)))
1190 FORMAT(1H0,///3X,5HJOINT,16X,'LOCATION(' ,A2,') - SEG(JNT)',5X,
    1 'LOCATION(' ,A2,') - SEG(J+1)',2X,'PRIN. AXIS(DEG) - SEG(JNT)',
    2 ' PRIN. AXIS(DEG) - SEG(J+1)'/2X,'J SYM PLOT JNT PIN',
    3 4X,2(1X,'X',8X,'Y',8X,'Z',8X),2('YAW',5X,'PITCH',5X,'ROLL',6X)/)
1200 FORMAT ((I3,A4,A3,2X,2I3,4(3F9.2,1X)))
1201 FORMAT(77X,'JOINTS RA, LA, RE, & LE ARE ORDERED PITCH, YAW, ROLL')REV1/87
1300 FORMAT(A4,A2,F6.2,3F6.4,6F6.3,6X,'CARD B.2')
1320 FORMAT(A4,1X,A1,2I4,6F6.2,22X,'CARD B.3'/14X,6F6.2,24X,3I2) REV1/87

END

```

```

C--SUBROUTINE CNVRT-----C
C   CNVRT CONVERTS ALL OUTPUT DATA TO THE UNITS SELECTED BY THE USER   C
C-----C

```

```

SUBROUTINE CNVRT (IUN)
COMMON /SGMNTS/ ABC(15,3),RMN(15),PHI(3,15),XYZCG(15,3)
COMMON /JNTS/ RNJ(3,28),JNT(14),YPRLL(14,6),IPIN(14)
COMMON /DIMS/ DD(-1:31),REGEQ(24,-1:31),PRED(3),
& RANGE(2,-1:1,3),CONVT(-1:1)
REAL CONV(2,3)
DATA CONV /1.,.0254,32.17405,143.1072,12.,106.2045/

DO 100 I = 1,15
  DO 10 J = 1,3
    ABC(I,J) = ABC(I,J)*CONV(IUN,1)
    XYZCG(I,J) = XYZCG(I,J)*CONV(IUN,1)
10    PHI(J,I) = PHI(J,I)/CONV(IUN,3)
100    RMN(I) = RMN(I)*CONV(IUN,2)

DO 200 I = 1,28
DO 200 J = 1,3
200    RNJ(J,I) = RNJ(J,I)*CONV(IUN,1)

IF (IUN.GT.1) THEN
  DD(0) = DD(0)/CONV(1,2)*CONV(2,2)
  DO 300 I=1,31
300    DD(I) = DD(I)*CONV(2,1)
ENDIF

RETURN
END

```


C--SUBROUTINE RESULTZ(ISUB)-----C
 C RESULTZ WRITES OUT THE COMPUTED BODY DESCRIPTION DATA FOR ISUB GT 4. C
 C-----C

SUBROUTINE RESULTZ (ISUB)

COMMON /NAMES/ SUBTYP(10),SEGLAB(15),JNTLAB(14),PLTSYM(29),
 & DIMLAB(-1:31),TITLE,UNITS(3,-1:2)

CHARACTER ANS*1,UNITS*6,VARB*6,SUBTYP*35,SEGLAB*3, 80386
 & JNTLAB*2,PLTSYM*1,DIMLAB*24,TITLE*60,HEADER*30

VARB = 'OUTPUT'
 CALL ASKUN (VARB,UNITS(1,2),IUN,2)
 10 WRITE (5,33)
 READ (7,44) ANS
 50 READ (2,*) NUMREC,HEADER
 IF (HEADER.NE.SUBTYP(ISUB)) THEN
 DO 30 I=1,NUMREC
 READ (2)
 30 CONTINUE
 GOTO 50
 ELSE
 IF (IUN.EQ.2) THEN
 DO 40 I=1,NUMREC+1
 READ(2)
 40 CONTINUE
 ENDIF
 ENDIF
 IF (ANS.EQ.'Y'.OR.ANS.EQ.'y') THEN 80386
 CALL ATBOUT (IUN)
 ELSE
 IF (ANS.EQ.'N'.OR.ANS.EQ.'n') THEN 80386
 CALL NOATBOUT (IUN)
 ELSE
 GOTO 10
 ENDIF
 ENDIF
 20 WRITE (5,55)
 READ (7,44) ANS
 IF (ANS.EQ.'N'.OR.ANS.EQ.'n') STOP 'FROM RESULTZ' 80386
 IF (ANS.NE.'Y'.AND.ANS.NE.'y') GOTO 20 80386
 RETURN
 33 FORMAT (//' IS ATB MODEL FORMATTED OUTPUT DESIRED (Y/N) ?')
 44 FORMAT (A)
 55 FORMAT (//' IS IT DESIRED TO PRODUCE ANOTHER BODY DESCRIPTION ',
 & 'DATA SET (Y/N) ?')

END

```

C--SUBROUTINE ABOUT(IUN)-----C
C   ABOUT WRITES OUT THE BODY DESCRIPTION DATA FOR ISUB GT 4           C
C   IN READABLE FORMAT AND ATB INPUT FORMAT                             C
C-----C

```

SUBROUTINE ABOUT (IUN)

```

CHARACTER SEG*4,JOINT(30)*4,CGS*1,JS*1,LABL2(2)*12,BLANK*4,
&      LABL3(2)*3,LABL4(2)*4,BDYTTL(5)*30,KTITLE(5)*20,
&      KTITLE2(5,50)*20
INTEGER IDYPR(6),NS(3),JOINTF(30),NF(5),IPIN(30),
&      IGLOB(30),IFLAG(30)
REAL YPRPMI(3),YPR1(3),YPR2(3),YPR3(3),ANG(3),THETA(50),
&      BD(6),PHI(3),SR(3,60),VISC(7),SPRING(5,90),X(30),
&      Y(30),F(50)

DATA LABL2,LABL3,LABL4 /'LB-SEC**2-IN',' N-SEC**2-M ',
&      'IN.',' M.','LB.',' N.'/

BLANK = ' '
IPAGE = 1
C I/O CARDB.1
READ (2,200) NSEG,NJNT,(BDYTTL(I),I=1,5)
WRITE (3,300) NSEG,NJNT,(BDYTTL(I),I=1,5)
WRITE (9,900) (BDYTTL(I),I=1,5),NSEG,NJNT,IPAGE
WRITE (9, '(T121, 'CARDS B.1')')
IPAGE = IPAGE+1
C I/O B.2 CARDS
WRITE (9,902)
WRITE (9,904) LABL2(IUN),LABL3(IUN),LABL3(IUN)
WRITE (9,906) LABL4(IUN)
DO 1 I=1,NSEG
  READ (2,202) SEG,CGS,W,(PHI(J),J=1,3),
& (BD(J),J=1,6),LPMI
  IF (IUN.EQ.1) THEN
    WRITE (3,302) SEG,CGS,W,(PHI(J),J=1,3),
& (BD(J),J=1,6),LPMI
  ELSE
    IF (IUN.EQ.2) THEN
      WRITE (3,303) SEG,CGS,W,(PHI(J),J=1,3),
& (BD(J),J=1,6),LPMI
    ENDIF
  ENDIF
ENDIF
C I/O B.2.2 CARDS IF LPMI NE 0
DO 14 M=1,3
  14 YPRPMI(M) = 0.0
  IF (LPMI.NE.0) THEN
    READ (2,204) (YPRPMI(J),J=1,3)
    WRITE (3,304) (YPRPMI(J),J=1,3)
  ENDIF
  WRITE (9,908) I,SEG,CGS,W,(PHI(J),J=1,3),
& (BD(J),J=1,6),(YPRPMI(J),J=1,3)

```

```

1 CONTINUE
C I/O B.3 CARDS IF NJNT NE 0
  IF (NJNT.EQ.0) GOTO 5
  WRITE (9,910)
  WRITE (9,912) LABL3(IUN),LABL3(IUN)
  WRITE (9,914)
  DO 2 J=1,NJNT
    READ(2,206) JOINT(J),JS,JNT,IPIN(J),
    & (SR(I,2*J-1),I-1,3),(SR(I,2*J),I-1,3),
    & ISLIP,C1,C2
    WRITE (3,306) JOINT(J),JS,JNT,IPIN(J),
    & (SR(I,2*J-1),I-1,3),(SR(I,2*J),I-1,3),
    & ISLIP,C1,C2
    READ (2,208) (YPR1(I),I-1,3),(YPR2(I),I-1,3),
    & (YPR3(I),I-1,3),(IDYPR(I),I-1,6)
    WRITE (3,308) (YPR1(I),I-1,3),(YPR2(I),I-1,3),
    & (YPR3(I),I-1,3),(IDYPR(I),I-1,6)
    WRITE (9,916) J,JOINT(J),JS,JNT,IPIN(J),
    & (SR(I,2*J-1),I-1,3),(SR(I,2*J),I-1,3),
    & (YPR1(I),I-1,3),(YPR2(I),I-1,3)
2 CONTINUE
C I/O B.4 CARDS
  WRITE (9,918) IPAGE
  IPAGE = IPAGE+1
  WRITE (9,'(T121, 'CARDS B.4')')
  WRITE (9,920)
  WRITE (9,922) LABL3(IUN),LABL4(IUN),LABL3(IUN),LABL4(IUN)
  WRITE (9,924)
  WRITE (9,926)
  DO 3 J=1,NJNT
    READ (2,210) (SPRING(I,3*J-2),I-1,5),
    & (SPRING(I,3*J-1),I-1,5)
    WRITE (3,310) (SPRING(I,3*J-2),I-1,5),
    & (SPRING(I,3*J-1),I-1,5)
    WRITE (9,928) J,JOINT(J),(SPRING(I,3*J-2),I-1,5),
    & (SPRING(I,3*J-1),I-1,5)
C IF JOINT(J) J IS AN EULER JOINT(J), I/O ANOTHER CARD
  IFLAG(J) = 0
  IF (IABS(IPIN(J)).GE.5.AND.IABS(IPIN(J)).LE.7) THEN
    IF (ISLIP.EQ.0) IFLAG(J) = 1
  ELSE
    IF (IABS(IPIN(J)).EQ.4) THEN
      IFLAG(J) = 1
    ELSE
      IF (IPIN(J).LE.-8.AND.IPIN(J).GE.-10) IFLAG(J) = 1
    ENDIF
  ENDIF
  IF (IFLAG(J).EQ.1) THEN
    READ (2,210) (SPRING(I,3*J),I-1,5),
    & (ANG(I),I-1,3)
    WRITE (3,310) (SPRING(I,3*J),I-1,5),
    & (ANG(I),I-1,3)

```

```

        WRITE (9,929) (SPRING(I,3*J),I-1,5)
C    &    (ANG(I),I-1,3)
        ENDIF
    3 CONTINUE
C I/O B.5 CARDS
    WRITE (9,(''-''',119X, ''CARDS B.5'''))
    WRITE (9,930)
    WRITE (9,932)
    WRITE (9,934)
    WRITE (9,936) LABL3(IUN), LABL4(IUN), LABL3(IUN),
&LABL4(IUN), LABL3(IUN), LABL4(IUN), LABL3(IUN), LABL4(IUN)
    DO 4 J-1, NJNT
        READ (2,212) (VISC(I),I-1,7)
        WRITE (3,312) (VISC(I),I-1,7)
        WRITE (9,938) J, JOINT(J), (VISC(I),I-1,7)
C IF JOINT(J) J IS AN EULER JOINT(J), THEN I/O TWO MORE CARDS
        IF (IFLAG(J).EQ.1) THEN
            READ (2,212) (VISC(I),I-1,7)
            WRITE (3,312) (VISC(I),I-1,7)
            WRITE (9,939) (VISC(I),I-1,7)
            READ (2,212) (VISC(I),I-1,7)
            WRITE (3,312) (VISC(I),I-1,7)
            WRITE (9,939) (VISC(I),I-1,7)
        ENDIF
    4 CONTINUE
C I/O NJNTF FROM THE D.1 CARD
    5 READ (2,214) NJNTF
    WRITE (3,314) NJNTF
    WRITE (9,940) IPAGE
    IPAGE = IPAGE+1
    WRITE (9,942) NJNTF
C I/O E.1 CARDS
    NUM = 1
    6 READ (2,216) IONE, (KTITLE(I),I-1,5)
    WRITE (3,316) IONE, (KTITLE(I),I-1,5)
    IF (IONE.GT.50) GOTO 7
    IF (NUM.EQ.1) THEN
        WRITE (9,944) IPAGE
        IPAGE = IPAGE+1
        NUM = 2
    ELSE
        NUM = 1
        WRITE (9,('////////'))
    ENDIF
    WRITE (9,946) IONE, (KTITLE(I),I-1,5)
C READ E.2 CARD
    READ (2,218) DO,D1,D2,D3,D4
    WRITE (3,318) DO,D1,D2,D3,D4
    WRITE (9,948)
    WRITE (9,950) DO,D1,D2,D3,D4
C I/O E.3 AND E.4 CARDS
    IF (D1.EQ.0) THEN

```

```

C      F1 = 'CONSTANT'
      WRITE (9,960) D2
      ELSE
C      IF (D1.LT.0) THEN
      F1 = 'TABULAR'
      READ (2,220) NPI
      READ (2,218) (X(I),Y(I),I-1,NPI)
      WRITE (3,320) NPI
      WRITE (3,322) (X(I),Y(I),I-1,NPI)
      WRITE (9,952) NPI
      WRITE (9,962)
      WRITE (9,964) (X(I),Y(I),I-1,NPI)
C IF D2 = 0, F2 DOES NOT EXIST
      IF (D2.GT.0) THEN
C      F2 = 'POLYNOMIAL'
      READ (2,218) A0,A1,A2,A3,A4,A5
      IF (IUN.EQ.1) THEN
        WRITE (3,324) A0,A1,A2,A3,A4,A5
      ELSE
        WRITE (3,326) A0,A1,A2,A3,A4,A5
      ENDIF
      WRITE (9,958)
      WRITE (9,966)
      WRITE (9,968) A0,A1,A2,A3,A4,A5
      ENDIF
      ELSE
C      IF (D1.GT.0) THEN
      F1 = 'POLYNOMIAL'
      READ (2,218) A0,A1,A2,A3,A4,A5
      IF (IUN.EQ.1) THEN
        WRITE (3,324) A0,A1,A2,A3,A4,A5
      ELSE
        IF (IUN.EQ.2) THEN
          WRITE (3,326) A0,A1,A2,A3,A4,A5
        ENDIF
      ENDIF
      WRITE (9,954)
      WRITE (9,966)
      WRITE (9,968) A0,A1,A2,A3,A4,A5
      IF (D2.LT.0) THEN
C      F2 = 'TABULAR'
      READ (2,220) NPI
      READ (2,218) (X(I),Y(I),I-1,NPI)
      WRITE (3,320) NPI
      WRITE (3,322) (X(I),Y(I),I-1,NPI)
      WRITE (9,956) NPI
      WRITE (9,962)
      WRITE (9,964) (X(I),Y(I),I-1,NPI)
      ELSE
C      IF (D2.GT.0) THEN
      F2 = 'POLYNOMIAL'
      READ (2,218) A0,A1,A2,A3,A4,A5

```

```

        IF (IUN.EQ.1) THEN
            WRITE (3,324) A0,A1,A2,A3,A4,A5
        ELSE
            WRITE (3,326) A0,A1,A2,A3,A4,A5
        ENDIF
        WRITE (9,958)
        WRITE (9,966)
        WRITE (9,968) A0,A1,A2,A3,A4,A5.
    ENDIF
    ENDIF
    ENDIF
    ENDIF
    ENDIF
    GOTO 6
C I/O E.7 CARDS
    7 IF (NJNTF.NE.0) THEN
        DO 8 I=1,NJNTF
            READ (2,216) ITWO,(KTITLE2(J,ITWO),J-1,5)
            WRITE (3,328) ITWO,(KTITLE2(J,ITWO),J-1,5)
            WRITE (9,944) IPAGE
            IPAGE = IPAGE+1
            WRITE (9,970) ITWO,(KTITLE2(J,ITWO),J-1,5)
            READ (2)
            WRITE (3)
            READ (2,222) NTHETA,NPHI
            WRITE (3,330) NTHETA,NPHI
            IF (NTHETA.GT.0) THEN
                DO 15 K=2,NTHETA
                    THETA(K) = DFLOAT(K-1)*180.0/DFLOAT(NTHETA-1)
15                CONTINUE
                    WRITE (9,972) NTHETA,NPHI
                    WRITE (9,974)
                    WRITE (9,976) (THETA(J),J-2,NTHETA)
                    WRITE (9,'('' ''')')
                ELSE
                    IF (NTHETA.LT.0) THEN
                        NPOLY = -NTHETA-1
                        WRITE (9,978) NPOLY,NPHI
                        WRITE (9,980)
                        WRITE (9,982) (BLANK,J,J-1,NPOLY)
                        WRITE (9)
                    ENDIF
                ENDIF
                NTHETE = IABS(NTHETA)
                PHIDEG = 360.0/DFLOAT(NPHI)
                DO 9 K=1,NPHI
                    READ (2,218) THETAO,(F(J),J-2,NTHETE)
                    WRITE (3,332) THETAO,(F(J),J-2,NTHETE)
                    DEG = PHIDEG*DFLOAT(K-1) - 180.0
                    WRITE (9,984) DEG,THETAO,(F(J),J-2,NTHETE)
9                CONTINUE
8                CONTINUE

```

```

        ENDIF
C I/O F.4 CARDS
C INITIALIZE IGLOB(J),J-1,18 SO 'CARD F.4' WILL PRINT OUT
  DO 10 J-1,18
10  IGLOB(J) = 0
    IF (NJNT.NE.0) THEN
      READ (2,224) (IGLOB(J),J-1,NJNT)
      WRITE (3,334) (IGLOB(J),J-1,18)
      IF (NJNT.GT.18) THEN
        WRITE (3,334) (IGLOB(J),J-19,NJNT)
      ENDIF
    DO 11 J-1,NJNT
      IF (IGLOB(J).EQ.1) THEN
        READ (2,226) NJ,(NS(I),I-1,3),
&          (NF(I),I-1,5)
        WRITE (3,226) NJ,(NS(I),I-1,3),
&          (NF(I),I-1,5)
      ENDIF
    11  CONTINUE
    ENDIF
C I/O F.5 CARD
C INITIALIZE JOINTF(J),J-1,18 TO ZERO SO 'CARD F.5' WILL PRINT
  DO 12 J-1,18
12  JOINTF(J) = 0
    IF (NJNT.GT.0.AND.NJNTF.GT.0) THEN
      WRITE (9,986) IPAGE
      IPAGE = IPAGE+1
      READ (2,224) (JOINTF(J),J-1,NJNT)
      WRITE (3,336) (JOINTF(J),J-1,18)
      IF (NJNT.GT.18) THEN
        WRITE (3,336) (JOINTF(J),J-19,NJNT)
      ENDIF
    DO 13 K-1,NJNT
      NUMF = JOINTF(K)
      IF (NUMF.NE.0) THEN
        WRITE (9,988) K,JOINT(K),NUMF,
&          (KTITLE2(I,NUMF),I-1,5)
      ENDIF
    13  CONTINUE
    ENDIF
C FINISHED CARD I/O
  RETURN

200  FORMAT (2I6,8X,5A6)
202  FORMAT (A4,1X,A1,10F6.0,I4)
204  FORMAT (12X,3F6.0)
206  FORMAT (A4,1X,A1,2I4,6F6.0,I4,2F6.0)
208  FORMAT (14X,9F6.0,6I2)
210  FORMAT (2(4F6.0,F12.0))
212  FORMAT (F6.0,4F6.0,18X,2F6.0)
214  FORMAT (48X,I6)
216  FORMAT (I4,4X,5A4)

```

218 FORMAT (6F12.0)
 220 FORMAT (I6)
 222 FORMAT (2I6)
 224 FORMAT (18I4)
 226 FORMAT (9I4)
 300 FORMAT (2I6,8X,5A6,T73,'CARD B.1')
 302 FORMAT (A4,1X,A1,F6.3,3F6.4,6F6.3,I4,T73,'CARD B.2')
 303 FORMAT (A4,1X,A1,F6.2,3F6.4,6F6.3,I4,T73,'CARD B.2')
 304 FORMAT (12X,2F6.2,F6.1)
 306 FORMAT (A4,1X,A1,2I4,6F6.2,I4,2F6.2,T73,'CARD B.3')
 308 FORMAT (14X,9F6.2,6I2)
 310 FORMAT (2(4F6.2,F12.2),'CARD B.4')
 312 FORMAT (F6.3,4F6.0,18X,2F6.0,T73,'CARD B.5')
 314 FORMAT (48('X'),I6,6('X'),T73,'CARD D.1')
 316 FORMAT (I4,4X,5A4,T73,'CARD E.1')
 318 FORMAT (5F12.2,T73,'CARD E.2')
 320 FORMAT (I6,66X,'CARD E.4')
 322 FORMAT (6F12.2,'CARD E.4')
 324 FORMAT (6F12.2,'CARD E.3')
 326 FORMAT (6E12.5,'CARD E.3')
 328 FORMAT (I4,4X,5A4,T73,'CARD E.7')
 330 FORMAT (2I6,T73,'CARD E.7')
 332 FORMAT (6F12.3,'CARD E.7')
 334 FORMAT (18I4,'CARD F.4')
 336 FORMAT (18I4,'CARD F.5')
 900 FORMAT ('1 CRASH VICTIM',5A6,3X,I2,' SEGMENTS',3X,
 &I2,' JOINTS',T124,'PAGE',I4)
 902 FORMAT (T26,'PRINCIPAL MOMENTS OF INERTIA',T68,'SEGMENT',
 &' CONTACT ELLIPSIOD',T121,'CARDS B.2')
 904 FORMAT (T4,'SEGMENT',T17,'WEIGHT',T32,'(',A12,')',T60,
 &'SEMIAXES ('A4,')',T87,'CENTER ('A4,')',
 &T111,'PRINCIPAL AXES (DEG)')
 906 FORMAT (' I SYM PLOT ('A4,')',T30,'X',T39,'Y',T48,
 &'Z',T59,'X',T67,'Y',T75,'Z',T85,'X',T93,'Y',T101,'Z',
 &T110,'YAW',T118,'PITCH',T128,'ROLL',/)
 908 FORMAT (1X,I2,1X,A4,A3,F11.3,2X,3F9.4,2(2X,3F8.3),1X,3F9.2)
 910 FORMAT ('-',T121,'CARDS B.3')
 912 FORMAT (' JOINT',T24,'LOCATION('A4,') - SEG(JNT)',
 &'LOCATION('A4,') - SEG(J+1) PRIN. AXIS(DEG) - SEG',
 &'(JNT) PRIN. AXIS(DEG) - SEG(J+1)')
 914 FORMAT (' J SYM PLOT PIN',T27,'X',T36,'Y',T45,'Z',T55,
 &'X',T64,'Y',T73,'Z',T82,'YAW',T90,'PITCH',T100,'ROLL',
 &T110,'YAW',T118,'PITCH',T128,'ROLL',/)
 916 FORMAT (1X,I2,2A4,I5,I3,2(1X,3F9.3),2(1X,3F9.2))
 918 FORMAT ('1 JOINT TORQUE CHARACTERISTICS',T124,'PAGE',
 &I4)
 920 FORMAT (T24,'FLEXURAL SPRING CHARACTERISTICS',T83,
 &'TORSIONAL SPRING CHARACTERISTICS',/)
 922 FORMAT (8X,2(7X,'SPRING COEF. ('2A4,'/DEG**J)',6X,
 &'ENERGY',5X,'JOINT'))
 924 FORMAT (' JOINT',2(9X,'LINEAR',4X,'QUADRATIC',5X,'CUBIC'
 &,4X,'DISSIPATION STOP'))

80386

926 FORMAT (8X,2(8X,'(J=1)',7X,'(J=2)',7X,'(J=3)',7X,'COEF.',
 &5X,'(DEG)'),/)
 928 FORMAT (1X,I2,A4,2(3X,3F12.3,2F10.3))
 929 FORMAT (10X,3F12.3,2F10.3)
 930 FORMAT (T39,'JOINT VISCOUS CHARACTERISTICS AND LOCK-'
 &,'UNLOCK CONDITIONS',/)
 932 FORMAT (T15,'VISCOUS',T31,'COULOMB',T45,'FULL FRICTION',
 &T63,'MAX TORQUE FOR',T81,'MIN TORQUE FOR',T99,'MIN. ',
 &'ANG. VELOCITY',T123,'IMPULSE')
 934 FORMAT (' JOINT',T13,'COEFFICIENT',T28,'FRICTION COEF. ',
 &'ANGULAR VELOCITY',T63,'A LOCKED JOINT',T81,'UNLOCKED ',
 &'JOINT',T121,'RESTITUTION')
 936 FORMAT (T9,'(',2A4,' SEC/DEG)',T29,'(',2A4,')',T45,
 &'(DEG/ SEC)',T65,'(',2A4,')',T83,'(',2A4,')',T103,
 &'(RAD/ SEC)',T121,'COEFFICIENT',/)
 938 FORMAT (1X,I2,A4,F14.3,2F15.2,4X,2F18.2,F20.2,F17.2)
 939 FORMAT (7X,F14.3,2F15.2,4X,2F18.2,F20.2,F17.2)
 940 FORMAT ('1',T6,'NPL',T13,'NBLT',T21,'NBAG',T29,'NELP',T39.
 &'NQ',T46,'NSD',T51,'NHRNSS',T59,'NWINDF',T68,'NJNTF',
 &T75,'NFORCE',T124,'PAGE',I4)
 942 FORMAT (8(4X,'XXXX'),4X,I4,4X,'XXXX',T121,'CARD D.1')
 944 FORMAT ('1',T124,'PAGE',I4)
 946 FORMAT (' FUNCTION NO.',I4,T22,5A4,T121,'CARDS E.1',/)
 948 FORMAT (T11,'D0',T26,'D1',T41,'D2',T56,'D3',T71,'D4')
 950 FORMAT (1X,F14.4,4F15.4)
 952 FORMAT ('-',T8,'FIRST PART OF FUNCTION - ',I4,' TABULAR ',
 &'POINTS',/)
 954 FORMAT ('-',T8,'FIRST PART OF FUNCTION - 5TH DEGREE ',
 &'POLYNOMIAL',/)
 956 FORMAT ('-',T8,'SECOND PART OF FUNCTION - ',I4,' TABULAR ',
 &'POINTS',/)
 958 FORMAT ('-',T8,'SECOND PART OF FUNTION - 5TH DEGREE ',
 &'POLYNOMIAL',/)
 960 FORMAT ('-',T8,'FUNCTION IS CONSTANT ',F11.6)
 962 FORMAT (T9,'0',T26,'F(0)')
 964 FORMAT (1X,F14.6,F15.4)
 966 FORMAT (T9,'A0',T24,'A1',T39,'A2',T54,'A3',T69,'A4',T84,
 &'A5')
 968 FORMAT (6E15.7)
 970 FORMAT (5X,'JOINT FORCE FUNCTION NO. ',I4,T38,5A4,T121,
 &'CARD E.7'//)
 972 FORMAT ('- FUNCTION IS TABULAR FOR ',I2,' X ',I2,' VALUES',
 &' OF THETA AND PHI',/)
 974 FORMAT (T31,'THETA')
 976 FORMAT (T6,'PHI',T14,'THETAO',F16.3,4F20.3/(15X,5F20.3))
 978 FORMAT ('- FUNCTION IS COEFFICIENTS OF ',I2,' ORDER ',
 &'POLYNOMIALS IN (THETA-THETAO) FOR ',I2,' VALUES OF PHI')
 980 FORMAT ('0',T28,'COEFFICIENTS OF (THETA-THETAO)**N')
 982 FORMAT (T6,'PHI',T14,'THETAO',7X,5(A4,'N-',I2,11X)/
 &(25X,A4,'N-',I2,11X,A4,'N-',I2,11X,A4,'NX-',I2,11X,A4,
 &'N-',I2,11X,A4,'N-',I2))
 984 FORMAT (F9.2,F10.3,5G20.7/(19X,5G20.7))

```
986  FORMAT ('1',122X,'PAGE ',I4/120X,'CARD F.5'/  
&      ' THE FOLLOWING JOINT RESTORING FORCE FUNCTIONS AS DEFINED  
& ON CARDS E.7 WILL BE USED.'//4X,'JOINT',10X,'FUNCTION'//)  
988  FORMAT (I6,'-',A4,I10,'-',5A4)
```

END

```

C--SUBROUTINE NOATBOUT(IUN)-----C
C   ABOUT WRITES OUT THE BODY DESCRIPTION DATA FOR ISUB GT 4           C
C   IN READABLE FORMAT.                                               C
C-----C

```

SUBROUTINE NOATEOUT (IUN)

```

CHARACTER SEG*4,JOINT(30)*4,CGS*1,JS*1,LABL2(2)*12,BLANK*4,
&      LABL3(2)*3,LABL4(2)*4,BDYTTL(5)*30,KTITLE(5)*20,
&      KTITLE2(5,50)*20
INTEGER IDYPR(6),NS(3),JOINTF(30),NF(5),IPIN(30),
&      IGLOB(30),IFLAG(30)
REAL YPRPMI(3),YPR1(3),YPR2(3),YPR3(3),ANG(3),THETA(50),
&      BD(6),PHI(3),SR(3,60),VISC(7),SPRING(5,90),X(30),
&      Y(30),F(50)

DATA LABL2,LABL3,LABL4 /'LB-SEC**2-IN',' N-SEC**2-M ',
&      'IN.',' M.','LB.',' N.'/

BLANK = ' '
IPAGE = 1
C I/O CARDB.1
READ (2,200) NSEG,NJNT,(BDYTTL(I),I-1,5)
WRITE (9,900) (BDYTTL(I),I-1,5),NSEG,NJNT,IPAGE
WRITE (9,'(T121, 'CARDS B.1')')
IPAGE = IPAGE+1
C I/O B.2 CARDS
WRITE (9,902)
WRITE (9,904) LABL2(IUN),LABL3(IUN),LABL3(IUN)
WRITE (9,906) LABL4(IUN)
DO 1 I-1,NSEG
  READ (2,202) SEG,CGS,W,(PHI(J),J-1,3),
& (BD(J),J-1,6),LPMI
C I/O B.2.2 CARDS IF LPMI NE 0
DO 14 M-1,3
  14 YPRPMI(M) = 0.0
  IF (LPMI.NE.0) THEN
    READ (2,204) (YPRPMI(J),J-1,3)
  ENDIF
  WRITE (9,908) I,SEG,CGS,W,(PHI(J),J-1,3),
& (BD(J),J-1,6),(YPRPMI(J),J-1,3)
1 CONTINUE
C I/O B.3 CARDS IF N NT NE 0
IF (NJNT.EQ.0) GOTO 5
WRITE (9,910)
WRITE (9,912) LABL3(IUN),LABL3(IUN)
WRITE (9,914)
DO 2 J-1,NJNT
  READ(2,206) JOINT(J),JS,JNT,IPIN(J),
& (SR(I,2*J-1),I-1,3),(SR(I,2*J),I-1,3),
& ISLIP,C1,C2
  READ (2,208) (YPR1(I),I-1,3),(YPR2(I),I-1,3),

```

```

& (YPR3(I),I-1,3),(IDYPR(I),I-1,6)
WRITE (9,916) J,JOINT(J),JS,JNT,IPIN(J),
& (SR(I,2*J-1),I-1,3),(SP(I,2*J),I-1,3),
& (YPR1(I),I-1,3),(YPR2(I),I-1,3)
2 CONTINUE
C I/O B.4 CARDS
WRITE (9,918) IPAGE
IPAGE = IPAGE+1
WRITE (9,'(T121, ''CARDS B.4'')')
WRITE (9,920)
WRITE (9,922) LABL3(IUN),LABL4(IUN),LABL3(IUN),LABL4(IUN)
WRITE (9,924)
WRITE (9,926)
DO 3 J=1,NJNT
READ (2,210) (SPRING(I,3*J-2),I-1,5),
& (SPRING(I,3*J-1),I-1,5)
WRITE (9,928) J,JOINT(J),(SPRING(I,3*J-2),I-1,5),
& (SPRING(I,3*J-1),I-1,5)
C IF JOINT(J) J IS AN EULER JOINT(J), I/O ANOTHER CARD
IFLAG(J) = 0
IF (IABS(IPIN(J)).GE.5.AND.IABS(IPIN(J)).LE.7) THEN
IF (ISLIP.EQ.0) IFLAG(J) = 1
ELSE
IF (IABS(IPIN(J)).EQ.4) THEN
IFLAG(J) = 1
ELSE
IF (IPIN(J).LE.-8.AND.IPIN(J).GE.-10) IFLAG(J) = 1
ENDIF
ENDIF
IF (IFLAG(J).EQ.1) THEN
READ (2,210) (SPRING(I,3*J),I-1,5),
& (ANG(I),I-1,3)
WRITE (9,929) (SPRING(I,3*J),I-1,5)
C & (ANG(I),I-1,3)
ENDIF
3 CONTINUE
C I/O B.5 CARDS
WRITE (9,'(''-''',119X, ''CARDS B.5'')')
WRITE (9,930)
WRITE (9,932)
WRITE (9,934)
WRITE (9,936) LABL3(IUN),LABL4(IUN),LABL3(IUN),
&LABL4(IUN),LABL3(IUN),LABL4(IUN),LABL3(IUN),LABL4(IUN)
DO 4 J=1,NJNT
READ (2,212) (VISC(I),I-1,7)
WRITE (9,938) J,JOINT(J),(VISC(I),I-1,7)
C IF JOINT(J) J IS AN EULER JOINT(J), THEN I/O TWO MORE CARDS
IF (IFLAG(J).EQ.1) THEN
READ (2,212) (VISC(I),I-1,7)
WRITE (9,939) (VISC(I),I-1,7)
READ (2,212) (VISC(I),I-1,7)
WRITE (9,939) (VISC(I),I-1,7)

```

```

        ENDIF
    4 CONTINUE
C I/O NJNTF FROM THE D.1 CARD
    5 READ (2,214) NJNTF
      WRITE (9,940) IPAGE
      IPAGE = IPAGE+1
      WRITE (9,942) NJNTF
C I/O E.1 CARDS
    NUM = 1
    6 READ (2,216) IONE,(KTITLE(I),I-1,5)
      IF (IONE.GT.50) GOTC 7
      IF (NUM.EQ.1) THEN
        WRITE (9,944) IPAGE
        IPAGE = IPAGE+1
        NUM = 2
      ELSE
        NUM = 1
        WRITE (9,'(////////)')
      ENDIF
      WRITE (9,946) IONE,(KTITLE(I),I-1,5)
C READ E.2 CARD
    READ (2,218) D0,D1,D2,D3,D4
    WRITE (9,948)
    WRITE (9,950) D0,D1,D2,D3,D4
C I/O E.3 AND E.4 CARDS
    IF (D1.EQ.0) THEN
C      F1 = 'CONSTANT'
      WRITE (9,960) D2
    ELSE
      IF (D1.LT.0) THEN
C      F1 = 'TABULAR'
        READ (2,220) NPI
        READ (2,218) (X(I),Y(I),I-1,NPI)
        WRITE (9,952) NPI
        WRITE (9,962)
        WRITE (9,964) (X(I),Y(I),I-1,NPI)
C IF D2 = 0, F2 DOES NOT EXIST
        IF (D2.GT.0) THEN
C      F2 = 'POLYNOMIAL'
          READ (2,218) A0,A1,A2,A3,A4,A5
          WRITE (9,958)
          WRITE (9,966)
          WRITE (9,968) A0,A1,A2,A3,A4,A5
        ENDIF
      ELSE
        IF (D1.GT.0) THEN
C      F1 = 'POLYNOMIAL'
          READ (2,218) A0,A1,A2,A3,A4,A5
          WRITE (9,954)
          WRITE (9,966)
          WRITE (9,968) A0,A1,A2,A3,A4,A5
          IF (D2.LT.0) THEN

```

```

C          F2 - 'TABULAR'
          READ (2,220) NPI
          READ (2,218) (X(I),Y(I),I-1,NPI)
          WRITE (9,956) NPI
          WRITE (9,962)
          WRITE (9,964) (X(I),Y(I),I-1,NPI)
        ELSE
          IF (D2.GT.0) THEN
C          F2 - 'POLYNOMIAL'
          READ (2,218) A0,A1,A2,A3,A4,A5
          WRITE (9,958)
          WRITE (9,966)
          WRITE (9,968) A0,A1,A2,A3,A4,A5
        ENDIF
      ENDIF
    ENDIF
  ENDIF
GOTO 6
C I/O E.7 CARDS
  7 IF (NJNTF.NE.0) THEN
    DO 8 I-1,NJNTF
      READ (2,216) ITWO,(KTITLE2(J,ITWO),J-1,5)
      WRITE (9,944) IPAGE
      IPAGE = IPAGE+1
      WRITE (9,970) ITWO,(KTITLE2(J,ITWO),J-1,5)
      READ (2)
      READ (2,222) NTHETA,NPHI
      IF (NTHETA.GT.0) THEN
        DO 15 K-2,NTHETA
          THETA(K) = DFLOAT(K-1)*180.0/DFLOAT(NTHETA-1)
15      CONTINUE
          WRITE (9,972) NTHETA,NPHI
          WRITE (9,974)
          WRITE (9,976) (THETA(J),J-2,NTHETA)
          WRITE (9,'('' ''')')
        ELSE
          IF (NTHETA.LT.0) THEN
            NPOLY = -NTHETA-1
            WRITE (9,978) NPOLY,NPHI
            WRITE (9,980)
            WRITE (9,982) (BLANK,J,J-1,NPOLY)
            WRITE (9)
          ENDIF
        ENDIF
      NTHETE = IABS(NTHETA)
      PHIDEG = 360.0/DFLOAT(NPHI)
      DO 9 K-1,NPHI
        READ (2,218) THETAO,(F(J),J-2,NTHETE)
        DEG = PHIDEG*DFLOAT(K-1) - 180.0
        WRITE (9,984) DEG,THETAO,(F(J),J-2,NTHETE)
9      CONTINUE

```

```

      8   CONTINUE
        ENDIF
C I/O F.4 CARDS
C   INITIALIZE IGLOB(J),J-1,18 SO 'CARD F.4' WILL PRINT OUT
      DO 10 J-1,18
10    IGLOB(J) = 0
        IF (NJNT.NE.0) THEN
          READ (2,224) (IGLOB(J),J-1,NJNT)
          DO 11 J-1,NJNT
            IF (IGLOB(J).EQ.1) THEN
              READ (2,226) NJ,(NS(I),I-1,3),
&          (NF(I),I-1,5)
              ENDIF
            11   CONTINUE
          ENDIF
C I/O F.5 CARD
C   INITIALIZE JOINTF(J),J-1,18 TO ZERO SO 'CARD F.5' WILL PRINT
      DO 12 J-1,18
12    JOINTF(J) = 0
        IF (NJNT.GT.0.AND.NJNTF.GT.0) THEN
          WRITE (9,986) IPAGE
          IPAGE = IPAGE+1
          READ (2,224) (JOINTF(J),J-1,NJNT)
          DO 13 K-1,NJNT
            NUMF = JOINTF(K)
            IF (NUMF.NE.0) THEN
              WRITE (9,988) K,JOINT(K),NUMF,
&          (KTITLE2(I,NUMF),I-1,5)
              ENDIF
            13   CONTINUE
          ENDIF
C FINISHED CARD I/O
      RETURN

```

```

200  FORMAT (^16,8X,5A6)
202  FORMAT (A4,1X,A1,10F6.0,I4)
204  FORMAT (12X,3F6.0)
206  FORMAT (A4,1X,A1,2I4,6F6.0,I4,2F6.0)
208  FORMAT (14X,9F6.0,6I2)
210  FORMAT (2(4F6.0,F12.0))
212  FORMAT (F6.0,4F6.0,18X,2F6.0)
214  FORMAT (48X,I5)
216  FORMAT (14,4X,5A4)
218  FORMAT (6F12.0)
220  FORMAT (I6)
222  FORMAT (2I6)
224  FORMAT (18I4)
226  FORMAT (9I4)
900  FORMAT ('1 CRASH VICTIM      ',5A6,3X,I2,' SEGMENTS',3X,
&I2,' JOINTS',T124,' PAGE',I4)
902  FORMAT (T26,' PRINCIPAL MOMENTS OF INERTIA',T68,' SEGMENT',
&' CONTACT ELLIPSIOD',T121,' CARDS B.2')

```

80386

904 FORMAT (T4, 'SEGMENT', T17, 'WEIGHT', T32, (' , A12, ') , T60,
 &'SEMIAXES (' , A4, ')', T87, 'CENTER (' , A4, ')',
 &T111, 'PRINCIPAL AXES (DEG)')
 906 FORMAT (' I SYM PLOT (' , A4, ')', T30, 'X', T39, 'Y', T48,
 &'Z', T59, 'X', T67, 'Y', T75, 'Z', T85, 'X', T93, 'Y', T101, 'Z',
 &T110, 'YAW', T118, 'PITCH', T128, 'ROLL', /)
 908 FORMAT (1X, I2, 1X, A4, A3, F11.3, 2X, 3F9.4, 2(2X, 3F8.3), 1X, 3F9.2)
 910 FORMAT (' - ', T121, 'CARDS B.3')
 912 FORMAT (' JOINT', T24, 'LOCATION(' , A4, ') - SEG(JNT) ',
 &'LOCATION(' , A4, ') - SEG(J+1) PRIN. AXIS(DEG) - SEG',
 &'(JNT) PRIN. AXIS(DEG) - SEG(J+1)')
 914 FORMAT (' J SYM PLOT PIN', T27, 'X', T36, 'Y', T45, 'Z', T55,
 &'X', T64, 'Y', T73, 'Z', T82, 'YAW', T90, 'PITCH', T100, 'ROLL',
 &T110, 'YAW', T118, 'PITCH', T128, 'ROLL', /)
 916 FORMAT (1X, I2, 2A4, I5, I3, 2(1X, 3F9.3), 2(1X, 3F9.2))
 918 FORMAT ('1 JOINT TORQUE CHARACTERISTICS', T124, 'PAGE',
 &I4)
 920 FORMAT (T24, 'FLEXURAL SPRING CHARACTERISTICS', T83,
 &'TORSIONAL SPRING CHARACTERISTICS', /)
 922 FORMAT (8X, 2(7X, 'SPRING COEF. (' , 2A4, '/DEG**J)', 6X,
 &'ENERGY', 5X, 'JOINT'))
 924 FORMAT (' JOINT', 2(9X, 'LINEAR', 4X, 'QUADRATIC', 5X, 'CUBIC'
 &, 4X, 'DISSIPATION STOP'))
 926 FORMAT (8X, 2(8X, '(J-1)', 7X, '(J-2)', 7X, '(J-3)', 7X, 'COEF.',
 &5X, '(DEG)'), /)
 928 FORMAT (1X, I2, A4, 2(3X, 3F12.3, 2F10.3))
 929 FORMAT (10X, 3F12.3, 2F10.3)
 930 FORMAT (T39, 'JOINT VISCOUS CHARACTERISTICS AND LOCK-'
 &, 'UNLOCK CONDITIONS', /)
 932 FORMAT (T15, 'VISCOUS', T31, 'COULOMB', T45, 'FULL FRICTION',
 &T63, 'MAX TORQUE FOR', T81, 'MIN TORQUE FOR', T99, 'MIN. ',
 &'ANG. VELOCITY', T123, 'IMPULSE')
 934 FORMAT (' JOINT', T13, 'COEFFICIENT', T28, 'FRICTION COEF. ',
 &'ANGULAR VELOCITY', T63, 'A LOCKED JOINT', T81, 'UNLOCKED ',
 &'JOINT', T121, 'RESTITUTION')
 936 FORMAT (T9, (' , 2A4, ' SEC/DEG)', T29, (' , 2A4, ')', T45,
 &'(DEG/ SEC)', T65, (' , 2A4, ')', T83, (' , 2A4, ')', T103,
 &'(RAD/ SEC)', T121, 'COEFFICIENT', /)
 938 FORMAT (1X, I2, A4, F14.3, 2F15.2, 4X, 2F18.2, F20.2, F17.2)
 939 FORMAT (7X, F14.3, 2F15.2, 4X, 2F18.2, F20.2, F17.2)
 940 FORMAT ('1', T6, 'NPL', T13, 'NBLT', T21, 'NBAG', T29, 'NELP', T39,
 &'NQ', T46, 'NSD', T51, 'NHRNSS', T59, 'NWINDF', T68, 'NJNTF',
 &T75, 'NFORCE', T124, 'PAGE', I4)
 942 FORMAT (8(4X, 'XXXX'), 4X, I4, 4X, 'XXXX', T121, 'CARD D.1')
 944 FORMAT ('1', T124, 'PAGE', I4)
 946 FORMAT (' FUNCTION NO.', I4, T22, 5A4, T121, 'CARDS E.1', /)
 948 FORMAT (T11, 'D0', T26, 'D1', T41, 'D2', T56, 'D3', T71, 'D4')
 950 FORMAT (1X, F14.4, 4F15.4)
 952 FORMAT (' - ', T8, 'FIRST PART OF FUNCTION - ', I4, ' TABULAR ',
 &'POINTS', /)
 954 FORMAT (' - ', T8, 'FIRST PART OF FUNCTION - 5TH DEGREE ',
 &'POLYNOMIAL', /)


```

956  FORMAT ('-',T8,'SECOND PART OF FUNCTION - ',I4,' TABULAR ',
&'POINTS',/)
958  FORMAT ('-',T8,'SECOND PART OF FUNTION - 5TH DEGREE ',
&'POLYNOMIAL',/)
960  FORMAT ('-',T8,'FUNCTION IS CONSTANT ',F11.6)
962  FORMAT (T9,'0',T26,'F(0)')
964  FORMAT (1X,F14.6,F15.4)
966  FORMAT (T9,'A0',T24,'A1',T39,'A2',T54,'A3',T69,'A4',T84,
&'A5')
968  FORMAT (6E15.7)
970  FORMAT (5X,'JOINT FORCE FUNCTION NO. ',I4,T38,5A4,T121,
&'CARD E.7'//)
972  FORMAT ('- FUNCTION IS TABULAR FOR ',I2,' X ',I2,' VALUES',
&' OF THETA AND PHI',/)
974  FORMAT (T31,'THETA')
976  FORMAT (T6,'PHI',T14,'THETAO',F16.3,4F20.3/(15X,5F20.3))
978  FORMAT ('- FUNCTION IS COEFFICIENTS OF ',I2,' ORDER ',
&' POLYNOMIALS IN (THETA-THETAO) FOR ',I2,' VALUES OF PHI')
980  FORMAT ('0',T28,'COEFFICIENTS OF (THETA-THETAO)**N')
982  FORMAT (T6,'PHI',T14,'THETAO',7X,5(A4,'N -',I2,11X)/
&(26X,A4,'N -',I2,11X,A4,'N -',I2,11X,A4,'N -',I2,11X,A4,
&'N -',I2,11X,A4,'N -',I2) )
984  FORMAT (F9.2,F10.3,5G20.7/(19X,5G20.7))
986  FORMAT ('1',122X,'PAGE ',I4/120X,'CARD F.5'/
&      ' THE FOLLOWING JOINT RESTORING FORCE FUNCTIONS AS DEFINED
& ON CARDS E.7 WILL BE USED.'//4X,'JOINT',10X,'FUNCTION'//)
988  FORMAT (I6,'-',A4,I10,'-',5A4)

```

END

APPENDIX I

386-ATB Program Listing

	BLOCK DATA		DECKA
C		REV IV 07/23/86	TWOPI
	IMPLICIT REAL*8 (A-H,O-Z)		DECKA
	COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,		DECKA
*	NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG		PAGE
	COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),		DECKA
*	UNITL,UNITM,UNITT,GRAVITY(3),TWOPI		TWOPI
	COMMON/JBARTZ/ MNPL(30),MNBLT(8),MNSEG(30),MNBAG(6),		DECKA
*	MPL(3,5,30),MBLT(3,5,8),MSEG(3,5,30),MBAG(3,10,6),		DECKA
*	NTPL(5,30),NTBLT(5,8),NTSEG(5,30)		DECKA
	COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),		DECKA
*	BLTTTL(5,8),PLTTL(5,30),BAGTTL(5,6),SEG(30),		DECKA
*	JOINT(30),CGS(30),JS(30)		DECKA
	REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT		DECKA
	LOGICAL*1 CGS,JS		DECKA
	COMMON/FORCES/PSF(7,70),BSF(4,20),SSF(10,40),BAGSF(3,20),		NGFORC
*	PRJNT(7,30),NPANEL(5),NPSF,NBSF,NSSF,NBGSF		DECKA
	COMMON/RSAVE/ XSG(3,20,3),DPMI(3,3,30),LPMI(30),		ATBIII
*	NSG(9),MSG(20,9),MCG,MCGIN(24,5),KREF(20,9)		TTHKREF
	COMMON/CDINT/ UU(4),GH(3,4),		DECKA
*	E(3,240),FF(5,240),GG(5,240),Y(5,240),U(5,240),		DECKA
*	H,HPRINT,TSAVE,TPRINT,START,ICNT,IDBL,IFLAG,IDMMY	80386	
	COMMON/DAMPER/ APSDM(3,20),APSDN(3,20),ASD(5,20),MSDM(20),MSDN(20)		DECKA
	COMMON/HRNESS/ BAR(15,100),BB(100),BBDOT(100),PLOSS(2,100),		DECKA
*	XLONG(20),HTIME(2),IBAR(5,100),NL(2,100),		DECKA
*	NPTSPB(20),NPTPLY(20),NTHRNS(20),NBLTPH(5)		DECKA
C	NOTE: FF REPLACES F.		DECKA
	LOGICAL*1 FREE,LDDMY		80386
	COMMON/TEMPVS/ JTMPVS(24000),FREE(30),LDDMY(2),DDMY(23109)		80386
	COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),		DECKA
*	SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)		DECKA
	COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60),		SLIP
*	RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),		DECKA
*	JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)		DECKA
	COMMON/CNTRSF/ PL(24,30),BELT(20,8),TPTS(6,8),BD(24,40)		EDGE
	COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(1250),TAB(4500)		BUTLER2
	COMMON/VPOSTN/ ZPLT(3),SPLT(3),AXV(3,6),VATAB(6,501,6),		VEHICL
*	VT0(6),VDT(6),TIMEV(6),OMEGV(6),NVTAB(6),INDXV(6)		DECKA
	COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60),		DECKA
*	F(3,30),TQ(3,30),WJ(30),A11(3,3,30)		SLIP
	COMMON/CEULER/ IEULER(30),HIR(3,3,90),ANG(3,30),ANGD(3,30),		JDRIFT
*	FE(3,30),TQE(3,30),CONST(5,30)		JDRIFT
	COMMON/FLXBLE/ HF(4,12,8),B42(3,3,24),V4(3,8),NFLEX(3,8)		DECKA
	COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),		DECKA
*	HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12),		DECKA
*	RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),		DECKA
*	KQ1(12),KQ2(12),KQTYPE(12)		DECKA
	COMMON/TEMPVI/ CREST,TTI(3),R1I(3),R2I(3),JSTOP(4,2,30)		DECKA
	COMMON/INTEST/ SGTEST(3,4,30),XTEST(3,120),SEGT(120),REGT(120)		DECKA
	REAL SEGT		DECKA
	COMMON/COMAIN/ VAR(240),DER(240),DT,HO,HMAX,HMIN,RSTIME,		DECKA
*	ISTEP,NSTEPS,NDINT,NEQ,IRSN,IRSOUT		DECKA

```

COMMON/ABDATA/ ZDEP(3,5),DBR(3,3,5),DPVCTR(3,5),DEPLOY(3,5),      DECKA
*              AB(3,5),B(9,4,5),ZR(3,4,5),BFB(3,4,5),DRR(9,4,5),  DECKA
*              VBAGG(5),VSCS(5),SPRK(5),CK(5),CMASS(5),CYMIN(5),   DECKA
*              CYMOUT(5),BAGPV(5),PD(5),VBAG(5),VOLBP(5),         DECKA
*              PCYV(5),PCYMIN(5),PVBAG(5),TV1(3,4,5),TV2(3,10,5), DECKA
*              SWITCH(5),PYMOUT(5),SCALE(5),PREVT,IFULL(6)        DECKA
COMMON/CYDATA/  CYTD(5),CYP(5),CYSP(5),CYTO(5),CYVO(5),CYCD(5),   DECKA
*              CYK(5),CYR(5),CYAT(5),CYPV(5),CYCD0(5),CYA0(5),   DECKA
*              CYP0(5),CYSS(5),CYL0(5),CYC(5),CYRHO0(5),CYVMAX(5), DECKA
*              CYORFC(5),CYRHO(5),CYT(5),CYP(5),CYV(5)           DECKA
COMMON/WINDFR/ WTIME(30),QFU(3,5),QFV(3,5),WF(3,30),IWIND(30),    WINDOP
*              MWSEG(7,30),NFVSEG(6),NFVNT(5),MOWSEG(30,30)       WINDOP
END                                                         DECKA

```

```

C
C AAMRL ARTICULATED TOTAL BODY (ATBIV) MODEL COMPUTER PROGRAM MAINA
C DEVELOPED BY CALSPAN CORP. AND J&J TECHNOLOGIES INC. ATBIV
C REV IV 07/23/86 TWOPI
C 80386 IMPLEMENTATION BY KETRON, INC. - OCTOBER 31, 1989 80386
C MAIN PROGRAM MAINA
C MAINA
C PERFORMS CARD INPUT, PROGRAM INITIALIZATION, MAINA
C CONTROL OF INTEGRATION LOOP AND OPTIONAL OUTPUT. MAINA
C MAINA
C IMPLICIT REAL*8(A-H,O-Z) MAINA
C COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLF,NBAG,NVEH,NGRND, MAINA
* NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG PAGE
C COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5), MAINA
* BLTTTL(5,8),PLTTL(5,30),BAGTTL(5,6),SEG(30), MAINA
* JOINT(30),CGS(30),JS(30) MAINA
C REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT MAINA
C LOGICAL*1 CGS,JS MAINA
C COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24), MAINA
* UNITL,UNITM,UNITT,GRAVITY(3),TWOPI TWOPI
C COMMON/COMAIN/ VAR(240),DER(240),DT,HO,HMAX,HMIN,RSTIME, MAINA
* ISTEP,NSTEPS,NDINT,NEQ,IRSI,IRSO, MAINA
C LOGICAL NPRT1,NPRT2,NPRT3 MAINA
C COMMON/DEVICE/ IDEF,IOPORT,MODEL 80386
C CHARACTER*12 FILE1,FILE2,FILE3,FILE4,FILE5,FILE6,FILE8,FILE20 80386
C CHARACTER*1 CHR 80386
C CALL ELTIME(1,1) MAINA
C PECONV
C MAKE THE OUTPUT FILES PRINTER CONTROL FILES FOR THE P&E PECONV
C PECONV
C OPEN (UNIT=14,FILE='ATBIV.DIR',STATUS='OLD') 80386
C READ (14,'(6(A12/),A12)')FILE1,FILE2,FILE3,FILE4,FILE5,FILE6,FILE8 80386
C READ (14,'(2(I4/),I4)')IDEF,IOPORT,MODEL 80386
C OPEN (UNIT=5,FILE=FILE5,STATUS='OLD') 80386
C OPEN (UNIT=6,FILE=FILE6,STATUS='NEW',CARRIAGE CONTROL='FORTRAN') 80386
C CALL CARCON(6,1) 80386
C CALL CARCON(2,1) 80386
C MA^NA
C WRITE PROLOGUE ON PRIMARY OUTPUT UNIT. MAINA
C MAINA
C NPG-2 PAGE
C WRITE(6,11) MAINA
11 FORMAT(1H1,30X,'AAMRL ARTICULATED TOTAL BODY (ATB) MODEL',52X, ATBIV
* 'PAGE 1'//// PAGE
* 31X,'DEVELOPED BY CALSPAN CORP., P.O. BOX 400, BUFFALO NY 14225'/BUTLER1
* 31X,'AND BY J&J TECHNOLOGIES INC., ORCHARD PARK, NY 14127' // EDGE
* 31X,'FOR THE AIR FORCE ARMSTRONG AEROSPACE MEDICAL RESEARCH ' / VEHICL
* 31X,'LABORATORY, WRIGHT PATTERSON AIR FORCE BASE ' /ATBIV
* 31X,'UNDER CONTRACTS F33615-75C-5002,-78C-0516 AND -80C-05117' //BUTLER1
* 31X,'AND FOR THE NATIONAL HIGHWAY TRAFFIC SAFETY ADMINISTRATION,'BUTLER1
*/31X,'U.S. DEPARTMENT OF TRANSPORTATION, UNDER CONTRACTS' / BUTLER1
* 31X,'FH-11-7592, HS-053-2-485, HS-6-01300 AND HS-6-01410.' //// BUTLER1

```

```

* 31X, 'PROGRAM DOCUMENTATION: NHTSA REPORT NOS. DOT-HS-801-507' / BUTLER1
* 31X, 'THROUGH 510 (FORMERLY CALSPAN REPORT NO. ZQ-5180-L-1),' / BUTLER1
* 31X, 'AVAILABLE FROM NTIS (ACCESSION NOS. PB-241692,3,4 AND 5),' / BUTLER1
* 31X, 'APPENDIXES A-J TO THE ABOVE (AVAILABLE FROM CALSPAN),' / BUTLER1
* 31X, 'AND REPORT NOS. AMRL-TR-75-14 (NTIS NO. AD-A014 816),' / ATBIV
* 31X, 'AFAMRL-TR-80-14 (NTIS NO. AD-A088 029), AND' / ATBIV
* 31X, 'AFAMRL-TR-83-073 (NTIS NO. AD-B079 184).'/////
* 31X, 'PROGRAM ATB-IV 80386 IMPLEMENTATION COMPLETED BY KETRON,' / 80386
* 31X, 'OCTOBER 31, 1989, UNDER CONTRACT F33615-88-C-0543'///// 80386
C * 31X, 'PROGRAM ATB-IV, EXECUTED ON THE AAMRL/BB CONCURRENT' / 80386
C * 31X, '3250 COMPUTER, WRIGHT-PATTERSON AFB, OHIO'///// 80386
C MAINA
C INPUT CARDS A.1 AND A.2, TEST FOR RESTART. MAINA
C MAINA
C CALL BLKDTA MAINA
C READ(5,12) DATE,IRSIN,IRSOUT,RSTIME,COMENT MAINA
12 FORMAT(3A4,2I4,F8.0/20A4/20A4) MAINA
WRITE(6,13) DATE,IRSIN,IRSOUT,RSTIME,COMENT MAINA
13 FORMAT(/////4X,3A4,' IRSIN-',I4,' IRSOUT-',I4,' RSTIME -',F8.4, MAINA
* 61X,'CARDS A'//1X,20A4/1X,20A4//) MAINA
IF (IRSIN.NE.0) 80386
* OPEN (UNIT=4, FILE=FILE4, STATUS='OLD', FORM='UNFORMATTED') 80386
IF (IRSOUT.NE.0) 80386
* OPEN (UNIT=3, FILE=FILE3, STATUS='NEW', FORM='UNFORMATTED') 80386
IF (IRSIN.NE.0) GO TO 18 MAINA
C MAINA
C INPUT CARDS A.3,A.4 AND A.5. MAINA
C MAINA
C READ(5,14) UNITL,UNITM,UNITT,GRAVTY,G MAINA
14 FORMAT(3A4,4F12.0) MAINA
IF (G.EQ.0.0) G = DSQRT(GRAVTY(1)**2+GRAVTY(2)**2+GRAVTY(3)**2) MAINA
READ(5,15) NDINT,NSTEPS,DT,HO,HMAX,HMIN,NPRT MAINA
15 FORMAT(2I4,4F8.0/36I2) MAINA
WRITE(6,16) UNITL,UNITM,UNITT,GRAVTY,G, CHGIII
* NDINT,NSTEPS,DT,HO,HMAX,HMIN MAINA
16 FORMAT(5X,'UNITL = ',A4,5X,'UNITM = ',A4,5X,'UNITT = ',A4, CHGIII
* 5X,'GRAVITY VECTOR = (' ,F9.4,',',F9.4,',',F9.4,')',5X,'G = ' CHGIII
*F9.4,/,5X,'NDINT = ',I4,5X,'NSTEPS = ',I5,5X,'DT = ',F8.6, MAINA
* 5X,'HO = ',F8.6,5X,'HMAX = ',F8.6,5X,'HMIN = ',F8.6) MAINA
WRITE(6,17) (I,I-1,36),NPRT MAINA
17 FORMAT('0 NPRT ARRAY'/3X,36I3/3X,36I3) MAINA
NPRT4 = NPRT(4) MAINA
IF(NPRT(25).GT.6) STOP 93 TGMOD1
IF (NPRT(4).NE.0) 80386
* OPEN (UNIT=8, FILE=FILE8, STATUS='NEW', FORM='UNFORMATTED') 80386
IF (NPRT(4).LT.0) GO TO 50 MAINA
IF (NPRT(4).EQ.0.OR.NPRT(4).EQ.1.OR.NPRT(4).EQ.4) THEN 80386
READ (14, '(A12)', END=38) FILE20 80386
IDOT=INDEX(FILE20, '.') 80386
IF(IDOT.LT.2.OR.IDOT.GT.9)GO TO 38 80386
CHR=FILE20(IDOT+2:IDOT+2) 80386
IF(LLT(CHR,'0').OR.LGT(CHR,'9'))GO TO 38 80386

```

CHR=FILE20(IDOT+3:IDOT+3)	80386
IF(LLT(CHR,'0').OR.LGT(CHR,'9'))GO TO 38	80386
READ(FILE20(IDOT+2:IDOT+3),'(I2)')IMULT	80386
IF(IMULT.LT.21.OR.IMULT.GT.85)GO TO 38	80386
DO 37 II=21,IMULT	80386
WRITE(FILE20(IDOT+2:IDOT+3),'(I2)')II	80386
OPEN (UNIT=II,FILE=FILE20,STATUS='NEW',	80386
* CARRIAGE CONTROL='FORTRAN')	80386
37 CONTINUE	80386
ENDIF	80386
38 CONTINUE	80386
IF (NPRT(1).NE.0)	80386
* OPEN (UNIT=1,FILE=FILE1,STATUS='NEW',FORM='UNFORMATTED')	80386
IF (NPRT(5).NE.0.OR.NPRT(6).NE.0.OR.NPRT(7).NE.0)	80386
* OPEN (UNIT=2,FILE=FILE2,STATUS='NEW',CARRIAGE CONTROL='FORTRAN')	80386
C	MAINA
C CALL INPUT ROUTINES	MAINA
C	MAINA
CALL BINPUT	MAINA
CALL VINPUT	MAINA
CALL SINPUT	MAINA
CALL CINPUT	MAINA
C	MAINA
C PROGRAM INITIALIZATION	MAINA
C	MAINA
TIME = 0.0	MAINA
CALL INITAL	MAINA
GO TO 19	MAINA
C	MAINA
C READ INPUT DATA FROM RESTART TAPE AND WRITE NEW TAPE.	MAINA
C THE FIVE FUNCTIONS OF SUBROUTINE RSTART ARE:	MAINA
C 1. READ INPUT & INITIALIZATION RECORD FROM OLD RESTART TAPE.	MAINA
C 2. WRITE INPUT & INITIALIZATION RECORD ONTO NEW RESTART TAPE.	MAINA
C 3. READ TIME POINT RECORD FROM OLD RESTART TAPE.	MAINA
C 4. READ NEW INPUT DATA FROM INPUT STREAM FOR RESTART.	MAINA
C 5. WRITE TIME POINT RECORD ONTO NEW RESTART TAPE.	MAINA
C	MAINA
18 CALL RSTART(1,IRSIN)	MAINA
CALL RSTART(4,5)	MAINA
NPRT4 = NPRT(4)	MAINA
19 IF (IRSOUT.NE.0) CALL RSTART(2,IRSOUT)	MAINA
C	MAINA
C INTEGRATION LOOP - ADVANCE TIME BY EITHER INTEGRATING THROUGH	MAINA
C SUBROUTINE DINT OR BY FETCHING TIME POINT RECORD FROM RESTART TAPE	MAINA
C	MAINA
TIME = 0.0	MAINA
ISTEP = 0	MAINA
20 IF (IRSIN.EQ.0) GO TO 23	MAINA
IF (TIME.GT.RSTIME+0.5*DT) GO TO 23	MAINA
IF (DABS(TIME-RSTIME).LT.0.5*DT) GO TO 21	MAINA
CALL RSTART(3,IRSIN)	MAINA
GO TO 24	MAINA

21	CALL RSTART(4,5)	MAINA
	IF (NPRT(4).LT.0) GO TO 50	MAINA
23	CALL DINT	MAINA
C		MAINA
C	OPTIONAL OUTPUT	MAINA
C	1. PRINTER PLOT ON OUTPUT UNIT 2 CONTROLLED BY NPRT(5) & (6).	MAINA
C		MAINA
24	CALL PRIPLT	MAINA
C		MAINA
C	2. RESTART DATA ON UNIT IRSOUT CONTROLLED BY IRSOUT # 0.	MAINA
C		MAINA
	IF (IRSOUT.NE.0) CALL RSTART(5,IRSOUT)	MAINA
C		MAINA
C	3. SUBROUTINE PRINT ON PRIMARY OUTPUT UNIT CONTROLLED BE NPRT(3).	MAINA
C		MAINA
	NPRT3 = (NPRT(3).EQ.1)	MAINA
	IF (NPRT(3).GT.1) NPRT3 = (MOD(ISTEP,NPRT(3)).EQ.0)	MAINA
	IF (NPRT3) CALL PRINT(6HMAIN3D)	MAINA
C		MAINA
C	4. PROGRAM VIEW PLOT DATA ON UNIT 1 CONTROLLED BY NPRT(1).	MAINA
C		MAINA
	NPRT1 = (NPRT(1).EQ.1)	MAINA
	IF (NPRT(1).GT.1) NPRT1 = (MOD(ISTEP,NPRT(1)).EQ.0)	MAINA
	IF (NPRT1) CALL UNIT1(0)	MAINA
C		MAINA
C	5. SUBROUTINE ELTIME ON PRIMARY OUTPUT UNIT CONTROLLED BY NPRT(2).	MAINA
C		MAINA
	NPRT2 = (NPRT(2).EQ.1)	MAINA
	IF (NPRT(2).GT.1) NPRT2 = (MOD(ISTEP,NPRT(2)).EQ.0)	MAINA
	IF (NPRT2) CALL ELTIME(NPG,1)	PAGE
C		MAINA
C	END OF INTEGRATION LOOP.	MAINA
C		MAINA
	ISTEP = ISTEP+1	MAINA
	IF (ISTEP.LE.NSTEPS) GO TO 20	MAINA
C		MAINA
C	6. SUBROUTINE POSTPR ON PRIMARY OUTPUT UNIT CONTROLLED BY NPRT(4).	MAINA
C		MAINA
50	IF (NPRT4 GT.0) END FILE 8	MAINA
	IF (NPRT(4).EQ.0 .OR. NPRT(4).EQ.4) GO TO 60	MAINA
	PRDT = 1000.0*DT	MAINA
	CALL POSTPR (PRDT)	MAINA
	IF (NPRT2) CALL ELTIME(NPG,1)	PAGE
C		MAINA
C	7. END OF RUN - CALL ELTIME IF NOT CALLED ABOVE.	MAINA
C		MAINA
60	IF (.NOT.NPRT2) CALL ELTIME(NPG,1)	PAGE
	STOP 1	MAINA
	END	MAINA

	SUBROUTINE ADJUST (M,D1)		ADJUST
C		REV IV 07/23/86	TWOPI
	IMPLICIT REAL*8 (A-H,O-Z)		ADJUST
	COMMON/CNSNTS/ PI,RADIAN,G,THIRD, EPS(24),		ADJUST
	* UNITL,UNITH,UNITT, GRAVITY(3), TWOPI		TWOPI
	COMMON/CDINT/ UU(4),GH(3,4),		ADJUST
	* E(3,240), F(5,240), GG(5,240), Y(5,240), U(5,240),		ADJUST
	* H,HPRINT,HS,TPRINT,TSTART, ICNT, IDBL, IFLAG, IDMMY		80386
	COMMON/COMAIN/ VAR(240), DER(240), DT, HO, HMAX, HMIN, RSTIME,		ADJUST
	* ISTEP, NSTEPS, NDINT, NEQ, IRSIN, IRSOUT		ADJUST
	IF (M.NE.1) GO TO 12		ADJUST
C			ADJUST
C	M = 1:		ADJUST
C			ADJUST
	DO 11 I=1,NEQ		ADJUST
	W = VAR(I) - GG(1,I)		ADJUST
	Z = DER(I) - GG(2,I)		ADJUST
	ZZ = Z - GG(5,I)*W - GG(3,I)*UU(3) - GG(4,I)*UU(4)		ADJUST
	GG(3,I) = GG(3,I) + ZZ*UU(1)		ADJUST
	GG(4,I) = GG(4,I) + ZZ*UU(2)		ADJUST
	Y(1,I) = VAR(I)		ADJUST
11	Y(2,I) = DER(I)		ADJUST
	GO TO 99		ADJUST
12	IF (M.EQ.3) GO TO 23		ADJUST
C			ADJUST
C	M = 2,4,5:		ADJUST
C			ADJUST
	H1 = EPS(1)/H		ADJUST
	N2 = NEQ/2		ADJUST
	DO 20 I=1,NEQ,3		ADJUST
	ZA = 0.0		ADJUST
	IF (I.LE.N2) GO TO 20		ADJUST
	IF (M.EQ.4) GO TO 16		ADJUST
	VARX = VAR(I) - Y(1,I)		ADJUST
	VARY = VAR(I+1) - Y(1,I+1)		ADJUST
	VARZ = VAR(I+2) - Y(1,I+2)		ADJUST
	DERX = DER(I) - Y(2,I)		ADJUST
	DERY = DER(I+1) - Y(2,I+1)		ADJUST
	DERZ = DER(I+2) - Y(2,I+2)		ADJUST
	GO TO 17		ADJUST
16	VARX = VAR(I) - U(1,I)		ADJUST
	VARY = VAR(I+1) - U(1,I+1)		ADJUST
	VARZ = VAR(I+2) - U(1,I+2)		ADJUST
	DERX = DER(I) - U(2,I)		ADJUST
	DERY = DER(I+1) - U(2,I+1)		ADJUST
	DERZ = DER(I+2) - U(2,I+2)		ADJUST
17	U(3,I) = U(3,I) + VARX*DERX + VARY*DERY + VARZ*DERZ		ADJUST
	U(4,I) = U(4,I) + VARX**2 + VARY**2 + VARZ**2		ADJUST
	IF (U(4,I).EQ.0.0) GO TO 18		FIXADJ
	ZA = H1		FIXADJ
	IF (U(3,I).LT.H1*U(4,I)) ZA = U(3,I)/U(4,I)		FIXADJ
18	GG(5,I+2) = ZA		FIXADJ

	GG(5,I+1) = ZA	ADJUST
20	GG(5,I) = ZA	ADJUST
	GO TO (99,21,99,23,25),M	ADJUST
C		ADJUST
C	M = 2:	ADJUST
C		ADJUST
21	DO 22 I=1,NEQ	ADJUST
	ZA = GG(5,I)	ADJUST
	Y1 = Y(4,I) - ZA*Y(3,I)	ADJUST
	Y2 = GG(2,I) - ZA*GG(1,I)	ADJUST
	Y3 = DER(I) - ZA*VAR(I)	ADJUST
	GG(3,I) = -Y1*GH(1,1) + Y2*GH(2,1) + Y3*GH(3,1)	ADJUST
	GG(4,I) = Y1*GH(1,2) - Y2*GH(2,2) + Y3*GH(3,2)	ADJUST
	Y(1,I) = 0.5*(Y(1,I)+VAR(I))	ADJUST
22	Y(2,I) = 0.5*(Y(2,I)+DER(I))	ADJUST
	GO TO 99	ADJUST
C		ADJUST
C	M = 3,4:	ADJUST
C		ADJUST
23	DO 24 I=1,NEQ	ADJUST
	ZA = GG(5,I)	ADJUST
	Y1 = GG(2,I) - ZA*GG(1,I)	ADJUST
	Y2 = Y(2,I) - ZA*Y(1,I)	ADJUST
	Y3 = DER(I) - ZA*VAR(I)	ADJUST
	GG(3,I) = -Y1*GH(1,3) + Y2*GH(2,3) - Y3*GH(3,3)	ADJUST
	GG(4,I) = Y1*GH(1,4) - Y2*GH(2,4) + Y3*GH(3,4)	ADJUST
	U(1,I) = VAR(I)	ADJUST
24	U(2,I) = DER(I)	ADJUST
	GO TO 99	ADJUST
C		ADJUST
C	M = 5:	ADJUST
C		ADJUST
25	DO 26 I=1,NEQ	ADJUST
	ZA = GG(5,I)	ADJUST
	Y1 = GG(2,I) - ZA*GG(1,I)	ADJUST
	Y2 = DER(I) - ZA*VAR(I)	ADJUST
	Y3 = U(2,I) - ZA*U(1,I)	ADJUST
	GG(3,I) = -Y1*GH(1,3) + Y2*GH(2,3) - Y3*GH(3,3)	ADJUST
	GG(4,I) = Y1*GH(1,4) - Y2*GH(2,4) + Y3*GH(3,4)	ADJUST
	Y(i,I) = VAR(I)	ADJUST
26	Y(2,I) = DER(I)	ADJUST
99	RETURN	ADJUST
	END	ADJUST

SUBROUTINE AIRBAG

AIRBAG

REV IV 07/24/86SLIP

AIRBAG ROUTINE CALLED BY SUBROUTINE CONTCT TO DETERMINE THE INTER-AIRBAG ACTION OF THE BAG WITH REACTION PANELS AND BODY SEGMENTS BY USE OFAIRBAG SUBROUTINE BGG. THE DIFFERENTIAL PRESSURE, FORCE AND TORQUE ON THE AIRBAG BAG IS EVALUATED AND THE RESULTING FORCE AND TORQUE ON THE BODY AIRBAG SEGMENTS ARE ADDED TO THE U1 AND U2 ARRAYS. AIRBAG

IMPLICIT REAL*8 (A-H,O-Z)

COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND, AIRBAG
 * NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG PAGE
 COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30), AIRBAG
 * SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30) AIRBAG
 COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60), SLIP
 * RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90), AIRBAG
 * JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30) AIRBAG
 COMMON/JBARTZ/ MNPL(30),MNBLT(8),MNSEG(30),MNBAG(6), AIRBAG
 * MPL(3,5,30),MBLT(3,5,8),MSEG(3,5,30),MBAG(3,10,6), AIRBAG
 * NTPL(5,30),NTBLT(5,8),NTSEG(5,30) AIRBAG
 COMMON/FORCES/PSF(7,70),BSF(4,20),SSF(10,40),BAGSF(3,20), NCFORC
 * PRJNT(7,30),NPANEL(5),NPSF,NBSF,NSSF,NBGSF AIRBAG
 COMMON/CNTRSFR/ PL(24,30),BELT(20,8),TPTS(6,8),BD(24,40) EDGE
 COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24), AIRBAG
 * UNITL,UNITM,UNITT,GRAVTY(3),TWOPI TWOPI
 COMMON/ABDATA/ ZDEP(3,5),DBR(3,3,5),DPVCTR(3,5),DEPLOY(3,5), AIRBAG
 * AB(3,5),B(9,4,5),ZR(3,4,5),BFB(3,4,5),DRR(9,4,5), AIRBAG
 * VBAGG(5),VSCS(5),SPRK(5),CK(5),CMASS(5),CYMIN(5), AIRBAG
 * CYMOUT(5),BAGPV(5),PD(5),VBAG(5),VOLBP(5), AIRBAG
 * PCYV(5),PCYMIN(5),PVBAG(5),TV1(3,4,5),TV2(3,10,5), AIRBAG
 * SWITCH(5),PYMOUT(5),SCALE(5),PREVT,IFULL(6) AIRBAG
 COMMON/CYDATA/ CYTD(5),CYPA(5),CYSP(5),CYTO(5),CYVO(5),CYCD(5), AIRBAG
 * CYK(5),CYR(5),CYAT(5),CYPV(5),CYCDO(5),CYA0(5), AIRBAG
 * CYPO(5),CYSS(5),CYLO(5),CYC(5),CYRHO0(5),CYVMAX(5), AIRBAG
 * CYORFC(5),CYRHO(5),CYT(5),CYP(5),CYV(5) AIRBAG
 COMMON/TEMPVS/ TMP(9),TMP1(3),TORQ(3),FORGE(3,5),TORA(3,5), AIRBAG
 * TQB(3,10),FRB(3,10),VOL(10),DELF(3),VOLP(4,5),FRA(4,5) AIRBAG
 * ,DMY(34955) 80386

NOTE: THIS COMMON/TEMPVS/ IS SHARED BY AIRBAG AND AIRBGG. AIRBAG

CALL ELTIME(1,24) AIRBAG

DELT = TIME-PREVT AIRBAG

NBGSF = 0 AIRBAG

DO 70 J=1,NBAG AIRBAG

IF (MNBAG(J).EQ.0) GO TO 70 AIRBAG

IF (IFULL(J).LE.0) GO TO 69 AIRBAG

CALL AIRBGG(J) AIRBAG

COMPUTE CMOUT: MASS FLOW OUT OF BAG AIRBAG

BAGPV: UNDISTORTED BAG VOLUME AIRBAG

IF (PD(J).GT.CYPV(J)) CYMOUT(J) = PYMOUT(J) AIRBAG

* + DELT*CYORFC(J)*DSQRT(PD(J)) AIRBAG

BAGPV(J) = CYPA(J)*((CYMIN(J)-CYMOUT(J))*SWITCH(J))**CYK(J) AIRBAG

C		AIRBAG
C	BAG IS FULLY INFLATED, COMPUTE DIFFERENTIAL PRESSURE	AIRBAG
C		AIRBAG
	PD(J) = BAGPV(J)/(VBAG(J)-VOLBP(J))*CYK(J) - CYP(A(J)	AIRBAG
	JB = NVEH + J	AIRBAG
	KP = NPANEL(J)	AIRBAG
	KBAG = MNBAG(J)	AIRBAG
C		AIRBAG
C	OPTIONAL DIAGNOSTIC OUTPUT	AIRBAG
C		AIRBAG
	IF (NPRT(21).NE.0) WRITE(6,41)	AIRBAG
	* ((FRB(I,K),I=1,3),(TQB(I,K),I=1,3),K=1,KBAG),(FORCE(I,J),I=1,3),	AIRBAG
	* (TORA(I,J),I=1,3),TORQ,((FRA(I,K),I=1,3),VOLP(K,J),K=1,KP),	AIRBAG
	* (VOL(K),K=1,KBAG),VOLBP(J),CYMOUT(J),BAGPV(J),PD(J)	AIRBAG
41	FORMAT ('OAIRBAG CONTCT'/(1X,9G14.6))	AIRBAG
	IF (PD(J).LT.0.0) PD(J) = 0.0	AIRBAG
	IF (PD(J).EQ.0.0) GO TO 46	AIRBAG
C		AIRBAG
C	SET UP BAGSF ARRAY FOR OUTPUT ROUTINE	AIRBAG
C		AIRBAG
	KBSF = NBSF+5	AIRBAG
	DO 42 K=1,KP	AIRBAG
	KBSF = KBSF+1	AIRBAG
	DO 42 I=1,3	AIRBAG
42	BAGSF(I,KBSF) = PD(J)*FRA(I,K)	AIRBAG
	DO 45 I=1,KBAG	AIRBAG
	KBSF = KBSF+1	AIRBAG
	IF (VOL(I).EQ.0.0) GO TO 45	AIRBAG
	M = MBAG(2,I,J)	AIRBAG
C		AIRBAG
C	FINAL COMPUTATIONS OF FORCE AND TORQUE ON AIRBAG	AIRBAG
C		AIRBAG
	DO 44 K=1,3	AIRBAG
	FRB(K,I) = PD(J)*FRB(K,I)	AIRBAG
	BAGSF(K,KBSF) = FRB(K,I)	AIRBAG
	U1(K,M) = U1(K,M) - FRB(K,I)	AIRBAG
44	U2(K,M) = U2(K,M) + PD(J)*TQB(K,I)	AIRBAG
45	CONTINUE	AIRBAG
46	DO 47 K=1,3	AIRBAG
	FORCE(K,J) = PD(J)*FORCE(K,J)	AIRBAG
47	TORA (K,J) = PD(J)*TORA (K,J)	AIRBAG
	IF (VOLP(1,J).NE.0.0) GO TO 55	AIRBAG
C		AIRBAG
C	AIRBAG IS NOT INTERSECTING PRIMARY REACTION PANEL.	AIRBAG
C	COMPUTE ARTIFICIAL FORCE AND TORQUE WITH A LINEAR SPRING FUNCTION	AIRBAG
C	IN AN ATTEMPT TO TIE +X SEMIAXIS ENDPOINT OF AIRBAG TO DEPLOYMENT	AIRBAG
C	POINT ON REACTION PANEL.	AIRBAG
C		AIRBAG
	DO 51 K=1,3	AIRBAG
51	TMP(K) = BFB(K,1,J) + ZDEP(K,J)	AIRBAG
	CALL DOT31 (D(1,1,NVEH),TMP,TMP1)	AIRBAG
	DO 52 K=1,3	AIRBAG

	DELFL(K) = TMP1(K) + SEGLP(K,NVEH) - SEGLP(K,JB)	AIRBAG
52	TMP(K) = BD(K+3,JB)	AIRBAG
	TMP(1) = TMP(1) + BD(1,JB)	AIRBAG
	CALL DOT31 (D(1,1,JB),TMP,TMP1)	AIRBAG
	DO 53 K=1,3	AIRBAG
	DELFL(K) = SPRK(J)*(DELFL(K)-TMP1(K))	AIRBAG
	BAGSF(K,NBGSF+5) = DELFL(K)	AIRBAG
53	FORCE(K,J) = FORCE(K,J) + DELFL(K)	AIRBAG
	CALL MAT31 (D(1,1,JB),DELFL,TMP1)	AIRBAG
	CALL CROSS (TMP,TMP1,DELFL)	AIRBAG
	DO 54 K=1,3	AIRBAG
54	TORA(K,J) = TORA(K,J) + DELFL(K)	AIRBAG
55	XDD = CYMIN(J) - CYMOUT(J) + W(JB)	AIRBAG
	FMASS = CMASS(J)*XDD/G	AIRBAG
	TMASS = CMASS(J)*(XDD+W(JB)*2.0/3.0)/G	AIRBAG
	DO 56 I=1,3	AIRBAG
56	TMP(I) = WMEG(I,JB)*PHI(I,JB)	AIRBAG
	CALL CROSS (WMEG(1,JB),TMP,TMP1)	AIRBAG
	DO 57 I=1,3	AIRBAG
	SEGLA(I,JB) = FORCE(I,J)/FMASS + GRAVITY(I)	AIRBAG
57	WMEGD(I,JB) = (TORA(I,J)/TMASS-TMP1(I))*RPHI(I,JB)	AIRBAG
69	NBGSF = NBGSF + 5 + NPANEL(J) + MNBAG(J)	AIRBAG
70	CONTINUE	AIRBAG
	CALL ELTIME(2,24)	AIRBAG
	RETURN	AIRBAG
	END	AIRBAG

```

SUBROUTINE AIRBGG(J)                                     AIRBGG
C                                                         REV III.5 10/17/85EDGE
C CALLED BY SUBROUTINES AIRBAG AND AIRBG3 TO COMPUTE VOLUMES OF AIRBGG
C INTERSECTION BETWEEN AIRBAGS AND PANELS AND SEGMENTS. AIRBGG
C                                                         AIRBGG
IMPPLICIT REAL*8 (A-H,O-Z)                               AIRBGG
COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND, AIRBGG
*               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG PAGE
COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30), AIRBGG
*               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30) AIRBGG
COMMON/JBARTZ/ MNPL( 30),MNBLT( 8),MNSEG( 30),MNBAG( 6), AIRBGG
*               MPL(3,5,30),MBLT(3,5,8),MSEG(3,5,30),MBAG(3,10,6), AIRBGG
*               NTPL( 5,30),NTBLT( 5,8),NTSEG( 5,30) AIRBGG
COMMON/FORCES/PSF(7,70),BSF(4,20),SSF(10,40),BAGSF(3,20), AIRBGG
*               PRJNT(7,30),NPANEL(5),NPSF,NBSF,NSSF,NBGSF AIRBGG
COMMON/CNTRF/ PL(24,30),BELT(20,8),TPTS(6,8),BD(24,40)   EDGE
COMMON/ABDATA/ ZDEP(3,5),DBR(3,3,5),DPVCTR(3,5),DEPLOY(3,5), AIRBGG
*               AB(3,5),B(9,4,5),ZR(3,4,5),BFB(3,4,5),DRR(9,4,5), AIRBGG
*               VBAGG(5),VSGS(5),SPRK(5),CK(5),CMASS(5),CYMIN(5), AIRBGG
*               CYMOUT(5),BAGPV(5),PD(5),VBAG(5),VOLBP(5), AIRBGG
*               PCYV(5),PCYMIN(5),PVBAG(5),TV1(3,4,5),TV2(3,10,5), AIRBGG
*               SWITCH(5),PYMOUT(5),SCALE(5),PREVT,IFULL(6) AIRBGG
COMMON/CYDATA/ CYTD(5),CYPA(5),CYSP(5),CYT0(5),CYV0(5),CYCD(5), AIRBGG
*               CYK(5),CYR(5),CYAT(5),CYPV(5),CYCD0(5),CYA0(5), AIRBGG
*               CYP0(5),CYSS(5),CYL0(5),CYC(5),CYRH00(5),CYVMAX(5), AIRBGG
*               CYORFC(5),CYRHO(5),CYT(5),CYP(5),CYV(5) AIRBGG
COMMON/TEMPVS/ TMP(9),TMP1(3),TORQ(3),FORCE(3,5),TORA(3,5), AIRBGG
*               TQB(3,10),FRB(3,10),VOL(10),DELF(3),VOLP(4,5),FRA(4,5) AIRBGG
*               ,DDMY(34955) 80386
C NOTE: THIS COMMON/TEMPVS/ IS SHARED BY AIRBAG AND AIRBGG. AIRBGG
JB = NVEH + J AIRBGG
VOLBP(J) = 0.0 AIRBGG
C                                                         AIRBGG
C COMPUTE THERMODYNAMIC PROPERTIES OF AIRBAG AIRBGG
C     CYRHO : DENSITY AIRBGG
C     CYT   : TEMPERATURE AIRBGG
C     CYP   : PRESSURE AIRBGG
C     CYMIN : MASS FLOW INTO BAG AIRBGG
C     VBCALC : CALCULATED VOLUME AIRBGG
C                                                         AIRBGG
Q - 1.0 AIRBGG
Q1 - 1.0 AIRBGG
Q2 - 1.0 AIRBGG
IF (TIME.LE.CYTD(J)) GO TO 13 AIRBGG
Q = 1.0 + CYC(J)*(TIME-CYTD(J)) AIRBGG
CYK1 = 2.0/(CYK(J)-1.0) AIRBGG
Q1 = 1.0/Q**CYK1 AIRBGG
Q2 = 1.0/Q**(CYK(J)*CYK1) AIRBGG
13  CYRHO(J) = CYRH00(J)*Q1 AIRBGG
     CYT(J) = CYT0(J)/Q**2 AIRBGG
     CYP(J) = CYP0(J)*Q2 AIRBGG
     CYMIN(J) = CYV0(J)*(CYRH00(J) - CYRHO(J)) AIRBGG

```

```

CYV(J)      - CYVMAX(J)*(1.0-Q2)
IF (TIME.LT.CYTD(J)) GO TO 31
IF (BD(1,JB).EQ.0.0) GO TO 31
IF (TIME.LE.0.0)      GO TO 31
VOLB = 0.0

C
C COMPUTE AIRBAG ELLIPSOID MATRIX AND ZERO BAG FORCE AND TORQUE.
C
IF (IFULL(J).NE.0) GO TO 21
SAB = SCALE(J)*AB(1,J)
DO 19 I=1,3
19 TMP(I) = DEPLOY(I,J) + SAB*DPVCTR(I,J)
CALL DOT31 (D(1,1,NVEH),TMP,SEGLP(1,JB))
DO 20 I=1,3
20 SEGLP(I,JB) = SEGLP(I,JB) + SEGLP(I,NVEH)
21 DO 23 I=1,3
FORCE(I,J) = 0.0
23 TORA (I,J) = 0.0

C
C COMPUTE FORCE,TORQUE AND VOLUME OF INTERSECTION
C OF AIRBAG WITH REACTION PANEL ELLIPSOIDS.
C
KP = NPANEL(J)
DO 26 K=1,KP
CALL BGG(
*   BD(7,JB),SEGLP(1,JB),D(1,1,JB),BD(4,JB),SEGLV(1,JB),WMEG(1,JB),AIRBGG
*   B(1,K,J),SEGLP(1,NVEH),D(1,1,NVEH),BFB(1,K,J),SEGLV(1,NVEH),AIRBGG
*   WMEG(1,NVEH),VSCS(J),IFULL(J),TV1(1,K,J),
*   FRA(1,K),TORQ,TQB,VOLP(K,J))
VOLBP(J) = VOLBP(J) + VOLP(K,J)
DO 26 I=1,3
FORCE(I,J) = FORCE(I,J) + FRA(I,K)
26 TORA (I,J) = TORA (I,J) + TORQ(I)

C
C COMPUTE FORCE,TORQUE AND VOLUME OF INTERSECTION
C OF AIRBAG WITH CONTACTING SEGMENT ELLIPSOIDS.
C
KBAG = MNBAG(J)
DO 30 I=1,KBAG
M = MBAG(2,I,J)
MM = MBAG(3,I,J)
CALL BGG(
*   BD(7,JB),SEGLP(1,JB),D(1,1,JB),BD(4,JB),SEGLV(1,JB),WMEG(1,JB),AIRBGG
*   BD(7,MM),SEGLP(1,M),D(1,1,M),BD(4,MM),SEGLV(1,M),WMEG(1,M),AIRBGG
*   VSCS(J),IFULL(J),TV2(1,I,J),FRB(1,I),TORQ,TQB(1,I),VOL(I))
IF (VOL(I).EQ.0.0) GO TO 30
VOLB = VOLB + VOL(I)
DO 28 K=1,3
FORCE(K,J) = FORCE(K,J) + FRB(K,I)
28 TORA (K,J) = TORA (K,J) + TORQ(K)
30 CONTINUE
VOLBP(J) = VOLBP(J) + VOLB

```

31 RETURN
END

AIRBGG
AIRBGG

	SUBROUTINE AIRBG1		AIRBG1
C		REV IV	07/24/86SLIP
C	READS AND PRINTS THE INPUT CARDS THAT DESCRIBE THE PHYSICAL		AIRBG1
C	DIMENSIONS AND GAS DYNAMICS OF THE AIRBAG RESTRAINTS AND		AIRBG1
C	PERFORMS INITIALIZATION REQUIRED BY THE AIRBAG ROUTINE.		AIRBG1
C			AIRBG1
	IMPLICIT REAL*8 (A-H,O-Z)		AIRBG1
	COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,		AIRBG1
*	NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG		PAGE
	COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),		AIRBG1
*	SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)		AIRBG1
	COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60),		SLIP
*	RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),		AIRBG1
*	JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)		AIRBG1
	COMMON/FORCES/PSF(7,70),BSF(4,20),SSF(10,40),BAGSF(3,20),		NCFORC
*	PRJNT(7,30),NPANEL(5),NPSF,NBSF,NSSF,NBSGF		AIRBG1
	COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),		AIRBG1
*	BLTTTL(5,8),PLTTL(5,30),BAGTTL(5,6),SEG(30),		AIRBG1
*	JOINT(30),CGS(30),JS(30)		AIRBG1
	REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT		AIRBG1
	LOGICAL*1 CGS,JS		AIRBG1
	COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),		AIRBG1
*	UNITL,UNITM,UNITT,GRAVTY(3),TWOPI		TWOPI
	COMMON/CNTRSF/ PL(24,30),BELT(20,8),TPTS(6,8),BD(24,40)		EDGE
	COMMON/INTEST/ SGTEST(3,4,30),XTEST(3,120),SEGT(120),REGT(120)		AIRBG1
	REAL SEGT		AIRBG1
	COMMON/ABDATA/ ZDEP(3,5),DBR(3,3,5),DPVCTR(3,5),DEPLOY(3,5),		AIRBG1
*	AB(3,5),B(9,4,5),ZR(3,4,5),BFB(3,4,5),DRR(9,4,5),		AIRBG1
*	VBAGG(5),VSCS(5),SPRK(5),CK(5),CMASS(5),CYMIN(5),		AIRBG1
*	CYMOUT(5),BAGPV(5),PD(5),VBAG(5),VOLBP(5),		AIRBG1
*	PCYV(5),PCYMIN(5),PVBAG(5),TV1(3,4,5),TV2(3,10,5),		AIRBG1
*	SWITCH(5),PYMOUT(5),SCALE(5),PREVT,IFULL(6)		AIRBG1
	COMMON/CYDATA/ CYTD(5),CYPA(5),CYSP(5),CYT0(5),CYV0(5),CYCD(5),		AIRBG1
*	CYK(5),CYR(5),CYAT(5),CYPV(5),CYCD0(5),CYA0(5),		AIRBG1
*	CYP0(5),CYSS(5),CYL0(5),CYC(5),CYRHO0(5),CYVMAX(5),		AIRBG1
*	CYORFC(5),CYRHO(5),CYT(5),CYP(5),CYV(5)		AIRBG1
	COMMON/TEMPVS/ TMP(9),TMP1(3),DMMY(35101)		80386
	DIMENSION YB(3),YP(3),IDYPR(3)		AIRBG1
	REAL BAG(6)		AIRBG1
	DATA BAG/4HBAG1,4HBAG2,4HBAG3,4HBAG4,4HBAG5,4HBAG /		AIRBG1
	DATA IDYPR/3,2,1/		AIRBG1
	DATA MAXNPL/4/,MAXSEG/30/		CHGIII
C			AIRBG1
C	MAKE ROOM FOR BAG DATA IN SEGMENT ARRAYS BETWEEN VEH AND GRND.		AIRBG1
C			AIRBG1
	MSEG = 0		CHGIII
	IF (NVEH.GT.NSEG) MSEG = NVEH - NSEG		CHGIII
	L = NSEG + NBAG + MSEG + 1		CHGIII
	K = NSEG + MSEG + 1		CHGIII
	W(L) = W(K)		AIRBG1
	RW(L) = RW(K)		AIRBG1
	SEG(L) = SEG(K)		AIRBG1

```

ISING(L) - ISING(K)
IF (L-1.GT.NJNT) JNT (L-1) = 0
IF (L-1.GT.NJNT) IPIN(L-1) = 0
DO 19 I=1,3
SEGLP(I,L) = SEGLP(I,K)
SEGLV(I,L) = SEGLV(I,K)
SEGLA(I,L) = SEGLA(I,K)
WMEG (I,L) = WMEG (I,K)
WMEGD(I,L) = WMEGD(I,K)
PHI (I,L) = PHI (I,K)
RPHI (I,L) = RPHI (I,K)
DO 18 J=1,3
D(I,J,L) = D(I,J,K)
18 SGTEST(I,J,L) = SGTEST(I,J,K)
19 SGTEST(I,4,L) = SGTEST(I,4,K)
NGRND = NSEG + NBAG + MSEG + 1
IF (NGRND.GT.MAXSEG) STOP 75
DO 40 J=1,NBAG
JB = NVEH + J
C
C READ AND PRINT CARDS D.4.A -D.4.F FOR THE JTH AIRBAG.
C
READ(5,13) (BAGTTL(I,J),I = 1,5),NPANEL(J),
* (AB(I,J),I=1,3) , (BD(I,JB),I=4,6),
* YB,(ZDEP(I,J),I=1,3),
* W(JB),CYTD(J),CYP(A,J),CYSP(J),CYTO(J),CYVO(J),
* CYCD(J),CYK(J),CYR(J),CYAT(J),CYPV(J),CYCDO(J),
* CYAO(J),SPRK(J),VSCS(J),CK(J),CMASS(J)
13 FORMAT (5A4,I4/(6F12.0))
IF (NPANEL(J).GT.MAXNPL) STOP 76
IF (MOD(J,2).EQ.1) WRITE(6,15) NPG
IF (MOD(J,2).EQ.1) NPG=NPG+1
15 FORMAT('1',122X,'PAGE',15/' AIRBAG INPUTS',105X,'CARDS D.4')
WRITE(6,14) J,(BAGTTL(I,J),I = 1,5 ),
* (AB(I,J),I=1,3) , (BD(I,JB),I=4,6),
* YB,(ZDEP(I,J),I=1,3),
* W(JB),CYTD(J),CYP(A,J),CYSP(J),CYTO(J),CYVO(J),
* CYCD(J),CYK(J),CYR(J),CYAT(J),CYPV(J),CYCDO(J),
* CYAO(J),SPRK(J),VSCS(J),CK(J),CMASS(J)
14 FORMAT('0 AIRBAG NO.',I4,4X,5A4//
* 29X,'AIR BAG SEMIAXES',46X,'C.G. OFFSET'/6X,6G20.9//
* 15X,'YAW',16X,'PITCH',15X,'ROLL',30X,'DEPLOYMENT POINT'
* /6X,6G20.9//
* 15X,'XBM',16X,'CYTD',16X,'CYP(A',16X,'CYSP',16X,'CYTO',16X,'CYVO'
* /6X,6G20.9//
* 14X,'CYCD',17X,'CYK',17X,'CYR',16X,'CYAT',16X,'CYPV',16X,'CYCDO'
* /6X,6G20.9//
*14X,'CYAO',16X,'SPRK',16X,'VSCS',17X,'CK',17X,'CMASS'/6X,5G20.9)
KP = NPANEL(J)
DO 25 K=1,KP
C
C READ AND PRINT CARDS D.4.G AND D.4.H FOR THE KTH PANEL TO

```

```

C CONTACT THE JTH AIRBAG. THESE PANELS ARE APPROXIMATED BY AIRBG1
C ELLIPSOIDS. THE FIRST PANEL (K=1) IS THE REACTION PANEL THAT AIRBG1
C INCLUDES THE DEPLOYMENT POINT. AIRBG1
C AIRBG1
READ(5,11) (B(I,K,J),I=1,3),(BFB(I,K,J),I=1,3), AIRBG1
* (ZR(I,K,J),I=1,3),YP AIRBG1
11 FORMAT(6F12.0) AIRBG1
WRITE(6,12) K,(B(I,K,J),I=1,3),(BFB(I,K,J),I=1,3), AIRBG1
* (ZR(I,K,J),I=1,3),YP AIRBG1
12 FORMAT('0 PANEL NO.',I4// AIRBG1
* 24X,'PANEL ELLIPSOID SEMIAXES',43X,'C.G. OFFSET'/6X,6G20.9// AIRBG1
* 29X,'PANEL LOCATION',32X,'YAW',16X,'PITCH',15X,'ROLL'/6X,6G20.9) AIRBG1
C AIRBG1
C CONVERT B FROM ELLIPSOID SEMIAXES TO MATRIX AIRBG1
C AIRBG1
DO 21 I=1,3 AIRBG1
21 TMP(I) = B(I,K,J) AIRBG1
DO 22 I=1,9 AIRBG1
22 B(I,K,J) = 0.0 AIRBG1
DO 23 I=1,3 AIRBG1
23 B(4*I-3,K,J) = 1.0/TMP(I)**2 AIRBG1
CALL DRCYPR (DRR(1,K,J),YP,IDYPR) AIRBG1
CALL MAT33 (B(1,K,J),DRR(1,K,J),TMP) AIRBG1
CALL DOT33 (DRR(1,K,J),TMP,B(1,K,J)) AIRBG1
CALL DOT31 (DRR(1,K,J),BFB(1,K,J),TMP) AIRBG1
DO 24 I=1,3 AIRBG1
24 BFB(I,K,J) = TMP(I) + ZR(I,K,J) AIRBG1
25 CONTINUE AIRBG1
C AIRBG1
C COMPUTE GEOMETRY OF DEPLOYMENT POINT ON FIRST PANEL. AIRBG1
C AIRBG1
CALL DRCYPR (DBR(1,1,J),YB,IDYPR) AIRBG1
CALL DOT31 (DRR(1,1,J),ZDEP(1,J),DEPLOY(1,J)) AIRBG1
DO 31 I=1,3 AIRBG1
DPVCTR(I,J) = -DBR(1,I,J) AIRBG1
31 DEPLOY(I,J) = DEPLOY(I,J) + BFB(I,1,J) AIRBG1
CALL PANEL (DBR(1,1,J),DEPLOY(1,J),JB) AIRBG1
C AIRBG1
C INITIALIZATION OF AIRBAG GEOMETRY. AIRBG1
C AIRBG1
VBAGG(J) = 4.0/3.0*PI*AB(1,J)*AB(2,J)*AB(3,J) AIRBG1
PHI(1,JB) = (AB(2,J)**2+AB(3,J)**2)/5.0 AIRBG1
PHI(2,JB) = (AB(3,J)**2+AB(1,J)**2)/5.0 AIRBG1
PHI(3,JB) = (AB(1,J)**2+AB(2,J)**2)/5.0 AIRBG1
JNT(JB-1) = 0 AIRBG1
IPIN(JB-1) = 0 AIRBG1
SEG(JB) = BAG(J) AIRBG1
IF (NBAG.EQ.1) SEG(JB) = BAG(6) AIRBG1
ISING(JB) = -1 AIRBG1
RW(JB) = G/W(JB) AIRBG1
DO 36 I=1,3 AIRBG1
BD(I,JB) = 0.0 AIRBG1

```

RPHI(I,JB) = 1.0/PHI(I,JB)	AIRBG1
DO 36 K=1,4	AIRBG1
36 SGTEST(I,K,JB) = 0.0	AIRBG1
DO 35 I=7,24	AIRBG1
35 BD(I,JB) = 0.0	AIRBG1
IFULL(J) = 0	AIRBG1
CYMOUT(J) = 0.0	AIRBG1
PYMOUT(J) = 0.0	AIRBG1
DO 38 I=1,3	AIRBG1
DO 37 K=1,4	AIRBG1
37 TV1(I,K,J) = 0.0	AIRBG1
DO 38 K=1,10	AIRBG1
38 TV2(I,K,J) = 0.0	AIRBG1
C	AIRBG1
C	AIRBG1
C	AIRBG1
AIR CYLINDER INITIALIZATION	AIRBG1
CYPO(J) = CYSP(J)+CYP(A,J)	AIRBG1
CYSS(J) = DSQRT(CYK(J)*CYR(J)*CYT0(J)*G)	AIRBG1
CYLO(J) = CYVO(J)/CYAT(J)	AIRBG1
CYK1 = CYK(J)-1.0	AIRBG1
CYK2 = 0.5*(CYK(J)+1.0)	AIRBG1
CYK3 = CYK2**(-CYK2/CYK1)	AIRBG1
CYC(J) = 0.5*CYK1*CYSS(J)*CYCD(J)/CYLO(J)*CYK3	AIRBG1
CYRHOO(J) = CYPO(J)/(CYR(J)*CYT0(J))	AIRBG1
CYVMAX(J) = CYVO(J)/CYK(J)*CYPO(J)/CYP(A,J)	AIRBG1
CYORFC(J) = CYCDO(J)*CYA0(J)*G*DSQRT(2.0*CYP(A,J)*CYK(J))/CYSS(J)	AIRBG1
IF (NPRT(22).NE.0) WRITE(6,39)	AIRBG1
* (SEGLP(I,JB),I-1,3),(SEGLV(I,JB),I-1,3),(WMEG(I,JB),I-1,3),	AIRBG1
* VBAGG(J),CYPO(J),CYSS(J),CYC(J),CYRHOO(J),CYVMAX(J),CYORFC(J)	AIRBG1
39 FORMAT('0 AIRBAG SINPOT'/(1X,9G14.6))	AIRBG1
40 CONTINUE	AIRBG1
PREVT = 0.0	AIRBG1
RETURN	AIRBG1
END	AIRBG1

```

SUBROUTINE AIRBG3(IRESET)
C
C
C THIS SUBROUTINE IS CALLED BY SUBROUTINE UPDATE AT START (IRESET=1)
C AND END (IRESET=2) OF EACH INTEGRATION STEP TO DETERMINE IF EACH
C AIRBAG HAS BEEN FULLY INFLATED.
C
IMPLICIT REAL*8 (A-H,O-Z)
COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,
* NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG
COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),
* SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)
COMMON/JBARTZ/ MNPL( 30),MNBLT( 8),MNSEG( 30),MNBAG( 6),
* MPL(3,5,30),MBLT(3,5,8),MSEG(3,5,30),MBAG(3,10,6),
* NTPL( 5,30),NTBLT( 5,8),NTSEG( 5,30)
COMMON/FORCES/PSF(7,70),BSF(4,20),SSF(10,40),BAGSF(3,20),
* PRJNT(7,30),NPANEL(5),NPSF,NBSF,NSSF,NBGSF
COMMON/CNTRF/ PL(24,30),BELT(20,8),TPTS(6,8),BD(24,40)
COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),
* UNITL,UNITM,UNITT,GRAVTY(3),TWOPI
COMMON/ABDATA/ ZDEP(3,5),DBR(3,3,5),DPVCTR(3,5),DEPLOY(3,5),
* AB(3,5),B(9,4,5),ZR(3,4,5),BFB(3,4,5),DRR(9,4,5),
* VBAGG(5),VSCS(5),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),TV2(3,10,5),
* SWITCH(5),PYMOUT(5),SCALE(5),PREVT,IFULL(6)
COMMON/CYDATA/ CYTD(5),CYPA(5),CYSP(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),CYRHO0(5),CYVMAX(5),
* CYORFC(5),CYRHO(5),CYT(5),CYP(5),CYV(5)
COMMON/TEMPVS/ TMP(9),TMP1(3),DDMY(35101)
CALL ELTIME(1,29)
JRESET = IRESET
IF (JRESET.EQ.1) PREVT = TIME
NBGSF = 0
DO 50 J=1,NBAG
IF (MNBAG(J).EQ.0) GO TO 50
JB = NVEH + J
JFULL = IFULL(J) + 2
IF (JFULL.LT.1 .OR. JFULL.GT.3) GO TO 11
IF (JRESET=1) 13,13,14
11 WRITE(6,12) TIME
12 FORMAT ('0 ERROR IN SUBROUTINE AIRBG3 AT TIME -',F10.6)
STOP 32
13 IF (JFULL=2) 41,49,49
14 IF (JFULL=2) 11,21,31
C
C END OF INTEGRATION STEP WHEN IFULL=0. TEST FOR FULL INFLATION.
C
21 PD(J) = 0.0
PCYV(J) = CYV(J)
PCYMIN(J) = CYMIN(J)

```

	PVBAG(J) - VBAG(J)	AIRBG3
22	CALL AIRBGG(J)	AIRBG3
	VBAG(J) - CYV(J) + VOLBP(J)	AIRBG3
	IF (SCALE(J).EQ.1.0) GO TO 23	AIRBG3
	SCALE(J) - (VBAG(J)/VBAGG(J))*THIRD	AIRBG3
	IF (SCALE(J).LT.1.0) GO TO 24	AIRBG3
	SCALE(J) - 1.0	AIRBG3
	GO TO 22	AIRBG3
23	IFULL(J) = -1	AIRBG3
	CYHOUT(J) = 0.0	AIRBG3
	PSW1 = (VBAG(J) - VBAGG(J)) * PCYV(J) / PCYMIN(J)	AIRBG3
	PSW2 = (VBAGG(J) - PVBAG(J)) * CYV(J) / CYMIN(J)	AIRBG3
	SWITCH(J) = (PSW1 + PSW2) / (VBAG(J) - PVBAG(J))	AIRBG3
	BAGPV(J) = CYP(A(J)) * (CYMIN(J) * SWITCH(J)) ** CYK(J)	AIRBG3
	PD(J) = BAGPV(J) / (CYV(J) ** CYK(J)) - CYP(A(J))	AIRBG3
24	DO 25 K=1,3	AIRBG3
	BD(K,JB) = SCALE(J) * AB(K,J)	AIRBG3
	IF (SCALE(J).EQ.0.0) GO TO 25	AIRBG3
	BD(4*K+12,JB) = BD(K,JB) ** 2	AIRBG3
	BD(4*K+3,JB) = 1.0 / BD(4*K+12,JB)	AIRBG3
25	TMP(K) = DEPLOY(K,J) + BD(1,JB) * DPVCTR(K,J)	AIRBG3
	CALL PANEL (DBR(1,1,J),TMP,JB)	AIRBG3
C		AIRBG3
C	SET UP BAGSF ARRAY FOR OUTPUT.	AIRBG3
C		AIRBG3
31	BAGSF(1,NBGSF+1) = CYP(J)	AIRBG3
	BAGSF(2,NBGSF+1) = CYT(J)	AIRBG3
	BAGSF(3,NBGSF+1) = PD(J)	AIRBG3
	CALL DOT31 (D(1,1,JB),BD(4,JB),TMP)	AIRBG3
	DC 32 K=1,3	AIRBG3
	B/GSF(K,NBGSF+3) = BD(K,JB)	AIRBG3
32	TMP(K) = TMP(K) + SEGLP(K,JB) - SEGLP(K,NVEH)	AIRBG3
	CALL MAT31 (D(1,1,NVEH),TMP,BAGSF(1,NBGSF+2))	AIRBG3
	CALL YPRDEG (D(1,1,JB),BAGSF(1,NBGSF+4))	AIRBG3
	NBGSF = NBGSF + 5 + NPANEL(J) + MNBAG(J)	AIRBG3
	GO TO 50	AIRBG3
C		AIRBG3
C	START OF INTEGRATION STEP WITH IFULL = -1, RESET INTEGRATOR.	AIRBG3
C		AIRBG3
41	IFULL(J) = 1	AIRBG3
	IRESET = -1	AIRBG3
49	PYMOUT(J) = CYMOUT(J)	AIRBG3
50	CONTINUE	AIRBG3
	CALL ELTIME(2,29)	AIRBG3
	RETURN	AIRBG3
	END	AIRBG3

	SUBROUTINE BELTG (ZA,ZB,ZC,BD)		BELTG
C		REV IV 07/23/86	TWOPI
C	COMPUTE TANGENT POINTS, UNIT VECTORS FROM TANGENT POINTS TO		BELTG
C	ANCHOR POINTS AND LENGTHS OF THE BELT SEGMENTS.		BELTG
C			BELTG
C	ARGUMENTS:		BELTG
C			BELTG
C	ZA,ZB - ANCHOR POINTS RELATIVE TO ELLIPSOID CENTER.		BELTG
C	ZC - FIXED POINT OF BELT ON SEGMENT ELLIPSOID.		BELTG
C	BD - SEGMENT ELLIPSOID SEMIAXES AND CENTER.		BELTG
C			BELTG
C	RESULTS ARE RETURNED TO CALLING ROUTINE VIA COMMON/TEMPVS/.		BELTG
C			BELTG
	IMPLICIT REAL*8 (A-H,O-Z)		BELTG
	DIMENSION ZA(3),ZB(3),ZC(3),BD(24)		BELTG
	COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,		BELTG
	* NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG		PAGE
	COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),		BELTG
	* UNITL,UNITM,UNITT,GRAVTY(3),TWOPI		TWOPI
C	NOTE: BELTRT AND BELTG SHARE FIRST PART OF TEMPVS		BELTG
	COMMON/TEMPVS/ APA(3),UVA(3),DLGA,AAA,APB(3),UVB(3),DLGB,UBB		BELTG
	* ,TA(3),TB(3),TC(3),UP(3),B(3)		BELTG
	* ,UC(3),AX(3),XE(3),BX(3),ACA(3),ACB(3),DMY(35064)		80386
C			BELTG
C	COMPUTE		BELTG
C	TC: NORMALIZED VECTOR OF BELT PLANE DETERMINED		BELTG
C	BY ANCHOR POINTS AND FIXED POINT.		BELTG
C			BELTG
	DO 10 K=1,3		BELTG
	TA(K) = ZC(K)-ZA(K)		BELTG
10	TB(K) = ZC(K)-ZB(K)		BELTG
	CALL CROSS(TB,TA,TC)		BELTG
	S = DSQRT(TC(1)**2 + TC(2)**2 + TC(3)**2)		BELTG
	TC(1) = TC(1)/S		BELTG
	TC(2) = TC(2)/S		BELTG
	TC(3) = TC(3)/S		BELTG
C			BELTG
C	GET DISTANCE OF BELT PLANE TO CENTER OF ELLIPSIOD.		BELTG
C			BELTG
	BET = TC(1)*ZC(1)+TC(2)*ZC(2)+TC(3)*ZC(3)		BELTG
C			BELTG
C	COMPUTE		BELTG
C	XE: CENTER OF ELLIPSE DETERMINED BY INTERSECTION		BELTG
C	OF BELT PLANE AND SEGMENT ELLIPSOID.		BELTG
C			BELTG
	CALL MAT31 (BD(16),TC,XE)		BELTG
	GG = BET/(TC(1)*XE(1)+TC(2)*XE(2)+TC(3)*XE(3))		BELTG
	DLGA = 0.0		BELTG
	DLGB = 0.0		BELTG
	DO 15 K=1,3		BELTG
	XE(K) = XE(K)*GG		BELTG
	UC(K) = ZC(K)-XE(K)		BELTG

	APA(K) - UC(K)	BELTG
15	APB(K) - UC(K)	BELTG
	YAY - GG*BET	BELTG
	YAY1 = 1.0-YAY	BELTG
	IF (YAY1.LE.EPS(6)) GO TO 70	BELTG
C		BELTG
C	CALCULATE POSSIBLE TANGENT POINTS FROM	BELTG
C	UVA,UVB: VECTORS FROM ELLIPSE CENTER TO MIDPOINT OF	BELTG
C	LINE CONNECTING POSSIBLE TANGENT POINTS.	BELTG
C	ACA,ACB: VECTORS FROM UVA,UVB TO TANGENT POINTS (POSITIVE).	BELTG
C		BELTG
	CALL MAT31 (BD(7),ZA,AX)	BELTG
	CALL MAT31 (BD(7),ZB,BX)	BELTG
	ZZA - AX(1)*ZA(1)+AX(2)*ZA(2)+AX(3)*ZA(3)	BELTG
	IF(ZZA.LE.1.0) STOP 88	CHGIII
	ZZB - BX(1)*ZB(1)+BX(2)*ZB(2)+BX(3)*ZB(3)	BELTG
	IF(ZZB.LE.1.0) STOP 89	CHGIII
	C2A - YAY1/(ZZA-YAY)	BELTG
	C2B - YAY1/(ZZB-YAY)	BELTG
	CALL CROSS(TC,AX,ACA)	BELTG
	CALL CROSS(TC,BX,ACB)	BELTG
	TTA - 0.0	BELTG
	TTB - 0.0	BELTG
	DO 21 I=1,3	BELTG
	DO 21 J=1,3	BELTG
	K = 3*J+I+3	BELTG
	TTA = TTA + ACA(I)*BD(K)*ACA(J)	BELTG
21	TTB = TTB + ACB(I)*BD(K)*ACB(J)	BELTG
	C3A - DSQRT((1.0 - C2A)*YAY1/TTA)	CHGIII
	C3B - DSQRT((1.0 - C2B)*YAY1/TTB)	CHGIII
	TT = DSQRT(UC(1)**2 + UC(2)**2 + UC(3)**2)	BELTG
	DO 24 K=1,3	BELTG
	UVA(K) - C2A*(ZA(K)-XE(K))	BELTG
	UVB(K) - C2B*(ZB(K)-XE(K))	BELTG
	ACA(K) - C3A*ACA(K)	BELTG
	ACB(K) - C3B*ACB(K)	BELTG
	UC(K) - UC(K)/TT	BELTG
24	B(K) - 0.0	BELTG
C		BELTG
C	OBTAIN EQUATION OF ELLIPSE	BELTG
C	B1*X**2 + 2*B2*X*Y + B3*Y**2 - 1	BELTG
C	IN UC,UP COORDINATES WHERE UC POINTS TO FIXED POINT.	BELTG
C		BELTG
	CALL CROSS(TC,UC,UP)	BELTG
	DO 22 I=1,3	BELTG
	DO 22 J=1,3	BELTG
	K = 3*J+I+3	BELTG
	B(1) - B(1) + UC(I)*BD(K)*UC(J)	BELTG
	B(2) - B(2) + UC(I)*BD(K)*UP(J)	BELTG
22	B(3) - B(3) + UP(I)*BD(K)*UP(J)	BELTG
	B(1) - B(1)/YAY1	BELTG
	B(2) - B(2)/YAY1	BELTG

	B(3) - B(3)/YAY1	BELTG
C		BELTG
C	COMPUTE ANGLES FROM FIXED POINT TO POSSIBLE TANGENT POINTS.	BELTG
C		BELTG
	UCUVA = UC(1)*UVA(1) + UC(2)*UVA(2) + UC(3)*UVA(3)	BELTG
	UCUVB = UC(1)*UVB(1) + UC(2)*UVB(2) + UC(3)*UVB(3)	BELTG
	UCACA = UC(1)*ACA(1) + UC(2)*ACA(2) + UC(3)*ACA(3)	BELTG
	UCACB = UC(1)*ACB(1) + UC(2)*ACB(2) + UC(3)*ACB(3)	BELTG
	UPUVA = UP(1)*UVA(1) + UP(2)*UVA(2) + UP(3)*UVA(3)	BELTG
	UPUVB = UP(1)*UVB(1) + UP(2)*UVB(2) + UP(3)*UVB(3)	BELTG
	UPACA = UP(1)*ACA(1) + UP(2)*ACA(2) + UP(3)*ACA(3)	BELTG
	UPACB = UP(1)*ACB(1) + UP(2)*ACB(2) + UP(3)*ACB(3)	BELTG
	TH1 = DATAN2(UPUVA-UPACA,UCUVA-UCACA)	BELTG
	TH2 = DATAN2(UPUVA+UPACA,UCUVA+UCACA)	BELTG
	TH3 = DATAN2(UPUVB-UPACB,UCUVB-UCACB)	BELTG
	TH4 = DATAN2(UPUVB+UPACB,UCUVB+UCACB)	BELTG
	IF (TH1.LT.0.0) TH1 = TWOPI + TH1	BELTG
	IF (TH2.LT.0.0) TH2 = TWOPI + TH2	BELTG
	IF (TH3.LT.0.0) TH3 = TWOPI + TH3	BELTG
	IF (TH4.LT.0.0) TH4 = TWOPI + TH4	BELTG
C		BELTG
C	CHOOSE PROPER TANGENT POINTS AND BELT ARC LENGTHS.	BELTG
C		BELTG
	THMIN = DMIN1(TH1,TH2,TH3,TH4)	BELTG
	IF (THMIN.EQ.TH1.AND.DMIN1(TH2,TH3,TH4).NE.TH4) GO TO 61	BELTG
	IF (THMIN.EQ.TH2.AND.DMAX1(TH1,TH3,TH4).EQ.TH4) GO TO 61	BELTG
	IF (THMIN.EQ.TH3.AND.DMIN1(TH1,TH2,TH4).NE.TH2) GO TO 63	BELTG
	IF (THMIN.EQ.TH4.AND.DMAX1(TH1,TH2,TH3).EQ.TH2) GO TO 63	BELTG
	GO TO 70	BELTG
61	THA = TH1	BELTG
	THB = TWOPI-TH4	BELTG
	DO 62 K=1,3	BELTG
	APA(K) = UVA(K)-ACA(K)	BELTG
62	APB(K) = UVB(K)+ACB(K)	BELTG
	GO TO 65	BELTG
63	THA = TWOPI-TH2	BELTG
	THB = TH3	BELTG
	DO 64 K=1,3	BELTG
	APA(K) = UVA(K)+ACA(K)	BELTG
64	APB(K) = UVB(K)-ACB(K)	BELTG
65	CONTINUE	BELTG
	EPS1 = EPS(1)	BELTG
	DLGA = DABS(ELONG(B(1),B(2),B(3),EPS1,THA))	BELTG
	DLGB = DABS(ELONG(B(1),B(2),B(3),EPS1,THB))	BELTG
C		BELTG
C	CALCULATE BELT LENGTHS AND UNIT VECTORS	BELTG
C	FROM TANGENT POINTS TO ANCHOR POINTS.	BELTG
C		BELTG
70	UAA=0.	BELTG
	UBB=0.	BELTG
	DO 80 K=1,3	BELTG
	APA(K) = APA(K)+XE(K)	BELTG

	APB(K) = APB(K)+XE(K)	BELTG
	UVA(K)=ZA(K)-APA(K)	BELTG
	UVB(K)=ZB(K)-APB(K)	BELTG
	APA(K)=APA(K)+BD(K+3)	BELTG
	APB(K)=APB(K)+BD(K+3)	BELTG
	UAA=UAA+UVA(K)**2	BELTG
	UBB=UBB+UVB(K)**2	BELTG
80	CONTINUE	BELTG
	UAA=DSQRT(UAA)	BELTG
	UBB=DSQRT(UBB)	BELTG
	DO 90 K=1,3	BELTG
	UVA(K)=UVA(K)/UAA	BELTG
	UVB(K)=UVB(K)/UBB	BELTG
90	CONTINUE	BELTG
C		BELTG
C	OPTIONAL OUTPUT	BELTG
C		BELTG
	IF (NPRI(15).EQ.0) GO TO 99	BELTG
	WRITE(6,50)	BELTG
50	FORMAT(1X,'BELT RESTRAINT')	BELTG
	WRITE(6,60) APA,UVA,DLGA,UAA	BELTG
	WRITE(6,60) APB,UVB,DLGB,UBB	BELTG
60	FORMAT (1X,1P8D15.5)	BELTG
99	CONTINUE	BELTG
	RETURN	BELTG
	END	BELTG

```

SUBROUTINE BELTRT(I,II,MM,M NT)                                BELTRT
C                                                                REV IV 07/23/86TWOPI
C THE ROUTINE CALLS SUBROUTINE BELTG TO COMPUTE THE TANGENT POINTS BELTRT
C AND BELT LENGTHS AND APPLIES THE RESTRAINT FORCES TO THE U1 ARRAY BELTRT
C AND BELT TORQUES TO THE U2 ARRAY FOR ELLIPSOID(II) ATTACHED TO BELTRT
C BODY SEGMENT (I) BY BELT (M) ATTACHED TO SEGMENT (MM). BELTRT
C BELTRT
C IMPLICIT REAL*8(A-H,O-Z) BELTRT
C COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND, BELTRT
* NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG PAGE
C COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30), BELTRT
* SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30) BELTRT
C COMMON/CNTRSRF/ PL(24,30),BELT(20,8),TPTS(6,8),BD(24,40) EDGE
C COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(1250),TAB(4500) DIMENB
C COMMON/FORCES/PSF(7,70),BSF(4,20),SSF(10,40),BAGSF(3,20), NCFORC
* PRJNT(7,30),NPANEL(5),NPSF,NBSF,NSSF,NBSGF BELTRT
C COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24), BELTRT
* UNITL,UNITM,UNITT,GRAVTY(3),TWOPI TWOPI
C COMMON/RSAVE/ XSG(3,20,3),DPMI(3,3,30),LPMI(30), TGMOD4
* NSG(9),MSG(20,9),MCG,MCGIN(24,5),KREF(20,9) TTHKREF
C NOTE: BELTRT AND BELTG SHARE FIRST PART OF TEMPVS BELTRT
C COMMON/TEMPVS/ APA(3),UVA(3),DLGA,UAA,APB(3),UVB(3),DLGB,UBB BELTRT
* ,DMMY(35097) 80386
C DIMENSION TA(3),TB(3),ZA(3),ZB(3),TT(3),TTT(3),TA1(3),TB1(3) TGMOD4
C BELTRT
C CALL ELTIME(1,22) BELTRT
C BELTRT
C CONVERT SEGMENT POSITION TO SEGMENT REFERENCE. BELTRT
C BELTRT
C MA = MOD(MM,100) JTF984
C MB = MM/100 JTF984
C IF (MB.EQ.0) MB=MA JTF984
C CALL DOT31 (D(1,1,MA),BELT(1,M),TA) BELTRT
C CALL DOT31 (D(1,1,MB),BELT(4,M),TB) BELTRT
C DO 10 K=1,3 BELTRT
C TA(K) = SEGLP(K,MA) + TA(K) - SEGLP(K,I) BELTRT
10 TB(K) = SEGLP(K,MB) + TB(K) - SEGLP(K,I) BELTRT
C CALL MAT31 (D(1,1,I),TA,ZA) BELTRT
C CALL MAT31 (D(1,1,I),TB,ZB) BELTRT
C DO 13 K=1,3 BELTRT
C ZA(K) = ZA(K) - BD(K+3,II) BELTRT
13 ZB(K) = ZB(K) - BD(K+3,II) BELTRT
C BELTRT
C COMPUTE NEW BELT LENGTHS AND EXPANSION. BELTRT
C BELTRT
C CALL BELTG (ZA, ZB, BELT(7,M), BD(1,II)) BELTRT
C TLA = DLGA+UAA BELTRT
C TLB = DLGB+UBB BELTRT
C TL = TLA+TLB BELTRT
C IF (TIME.NE.0.0) GO TO 11 BELTRT
C BELTRT
C IF TIME=0, COMPUTE INITIAL BELT LENGTHS BELTRT

```

C	AND STORE RESULTS IN BELT ARRAY.	BELTRT
C		BELTRT
	IF (BELT(11,M).LT.0.0) BELT(11,M)=-BELT(11,M)-TL	BELTRT
	IF (BELT(11,M).LT.0.0) BELT(11,M)=0.0	BELTRT
	BELT(12,M) = TLA+TLA/TL*BELT(11,M)	BELTRT
	BELT(13,M) = TLB+TLB/TL*BELT(11,M)	BELTRT
	B1213 = BELT(12,M) + BELT(13,M)	BELTRT
	BELT(10,M) = B1213	BELTRT
	DO 305 LL=1,3	TGMOD4
	TA1(LL) = APA(LL)	TGMOD4
305	TB1(LL) = APB(LL)	TGMOD4
	IF(LPMI(I).EQ.0) GO TO 306	TGMOD4
	CALL DOT31(DPMI(1,1,I),APA,TA1)	TGMOD4
	CALL DOT31(DPMI(1,1,I),APB,TB1)	TGMOD4
306	CONTINUE	TGMOD4
	WRITE (6,14) M, B1213, BELT(12,M), BELT(13,M), UNITL,I,TA1, TB1	TGMOD4
14	FORMAT('0 INITIAL LENGTHS OF BELT NO.',I3,' AND ITS SEGMENTS ARE',	BELTRT
	* 3F12.4,1X,A4/'0 INITIAL TANGENT POINTS IN LOCAL REFERENCE	TGMOD4
	*OF SEGMENT ',I2,' ARE:',/,2(3X,3F12.3))	TGMOD4
C		BELTRT
C	CONVERT TANGENT POINTS TO INERTIAL REFERENCE AND STORE.	BELTRT
C		BELTRT
11	CALL DOT31 (D(1,1,I),APA,TPTS(1,M))	BELTRT
	CALL DOT31 (D(1,1,I),APB,TPTS(4,M))	BELTRT
	DO 12 K=1,3	BELTRT
	TPTS(K ,M) = TPTS(K ,M) + SEGLP(K,I)	BELTRT
12	TPTS(K+3,M) = TPTS(K+3,M) + SEGLP(K,I)	BELTRT
	SDOT = 0.0	BELTRT
	NCF = NTAB(NT+5)	BELTRT
	IF (NCF.NE.0) GO TO 15	BELTRT
C		BELTRT
C	ZERO BELT FRICTION, COMPUTE STRAIN AND FORCE OF ENTIRE BELT.	BELTRT
C		BELTRT
	B1213 = BELT(12,M)+BELT(13,M)	BELTRT
	S = (TL-B1213)/B1213	BELTRT
	SA = S	BELTRT
	SB = S	BELTRT
	IF (S.LT.0.0) S = 0.0	BELTRT
	CALL FRCDFL (S,SDOT,NT,1,FA,ELOSS)	BELTRT
	FB = FA	BELTRT
	GO TO 17	BELTRT
C		BELTRT
C	FULL BELT FRICTION, COMPUTE STRAIN AND FORCE OF EACH PART OF BELT.	BELTRT
C		BELTRT
15	IF (TL.GT.BELT(10,M)) GO TO 16	BELTRT
	FA = 0.0	BELTRT
	FB = 0.0	BELTRT
	SA = (TL-BELT(10,M))/BELT(10,M)	BELTRT
	SB = SA	BELTRT
	BELT(12,M) = TLA	BELTRT
	BELT(13,M) = TLB	BELTRT
	GO TO 17	BELTRT

16	S = (TLA-BELT(12,M))/BELT(12,M)	BELTRT
	SA = S	BELTRT
	IF (S.LT.0.0) S = 0.0	BELTRT
	CALL FRCDFL (S,SDOT,NT,1,FA,ELOSS)	BELTRT
	S = (TLB-BELT(13,M))/BELT(13,M)	BELTRT
	SB = S	BELTRT
	IF (S.LT.0.0) S = 0.0	BELTRT
	CALL FRCDFL (S,SDOT,NT+6,1,FB,ELOSS)	BELTRT
	BELT(10,M) = 0.0	BELTRT
17	BSF(1,NBSF) = SA	BELTRT
	BSF(2,NBSF) = FA	BELTRT
	BSF(3,NBSF) = SB	BELTRT
	BSF(4,NBSF) = FB	BELTRT
	IF (FA+FB.LE.0.0) GO TO 31	BELTRT
C		BELTRT
C	COMPUTE FORCE VECTORS.	BELTRT
C		BELTRT
	DO 20 K=1,3	BELTRT
	UVA (K) = FA*UVA(K)	BELTRT
20	UVB(K) = FB*UVB(K)	BELTRT
C		BELTRT
C	CONVERT FORCES TO INERTIAL REFERENCE AND ADD TO U1 ARRAY.	BELTRT
C		BELTRT
	CALL DOT31(D(1,1,I),UVA,TT)	BELTRT
	CALL DOT31(D(1,1,I),UVB,TTT)	BELTRT
	DO 30 K=1,3	BELTRT
	U1(K,MA) = U1(K,MA) - TT(K)	JTF984
	U1(K,MB) = U1(K,MB) - TTT(K)	JTF984
30	U1(K,I) = U1(K,I)+TTT(K) - TT(K)	JTF984
C		BELTRT
C	CONVERT TORQUES TO LOCAL REFERENCE AND ADD TO U2 ARRAY.	BELTRT
C		BELTRT
	CALL MAT31(D(1,1,MA),TT,ZA)	JTF984
	CALL MAT31(D(1,1,MB),TTT,ZB)	JTF984
	CALL CROSS(BELT(1,M),ZA,TA)	JTF984
	CALL CROSS(BELT(4,M),ZB,TB)	JTF984
	CALL CROSS(APA,UVA,TT)	BELTRT
	CALL CROSS(APB,UVB,TTT)	BELTRT
	DO 40 K=1,3	BELTRT
	U2(K,MA) = U2(K,MA) - TA(K)	JTF984
	U2(K,MB) = U2(K,MB) - TB(K)	JTF984
40	U2(K,I) = U2(K,I)+(TT(K)+TTT(K))	BELTRT
31	CONTINUE	BELTRT
	CALL ELTIME(2,22)	BELTRT
	RETURN	BELTRT
	END	BELTRT

```

SUBROUTINE BGG(A, ZA, DA, BFA, VA, WA,                                BGG
*       B, ZB, DB, BFB, VB, WB,                                    BGG
*       VSCS, IFULL, TV, FRA, TORQ, TQB, VOL)                       BGG
C
C                                     REV IV 07/23/86TWOPI
C COMPUTES THE VOLUME OF INTERSECTION OF AN ELLIPSOIDAL AIRBAG    BGG
C WITH AN ELLIPSOIDAL BODY SEGMENT OR REACTION PANEL.           BGG
C ALSO COMPUTES THE FORCE PER UNIT PRESSURE AND TORQUE PER UNIT   BGG
C PRESSURE ON BOTH THE BAG AND THE INTERSECTING OBJECT.         BGG
C
C ARGUMENTS:                                                       BGG
C AIRBAG INPUTS : A(3,3) - ELLIPSOID MATRIX                      BGG
C                   ZA(3) - C.G.                                  BGG
C                   DA(3,3)- DIRECTION COSINE MATRIX              BGG
C                   BFA(3) - OFFSET                               BGG
C                   VA(3) - CG VELOCITY(INERTIAL REF.)           BGG
C                   WA(3) - ANGULAR VELOCITY (LOCAL REF.)       BGG
C
C CONTACT SURFACE B(3,3) - ELLIPSOID MATRIX                      BGG
C                   ZB(3) - C.G.                                  BGG
C                   DB(3,3)- DIRECTION COSINE MATRIX              BGG
C                   BFB(3) - OFFSET                               BGG
C                   VB(3) - CG VELOCITY (INERTIAL REF.)          BGG
C                   WB(3) - ANGULAR VELOCITY (LOCAL REF.)       BGG
C                   VSCS - COEFFICIENT OF SLIDING FRICTION       BGG
C                   IFULL - IF ZERO, COMPUTE VOL ONLY.           BGG
C                   TV(3) - MEMORY FOR SUBROUTINES INTERS & EDEPTH. BGG
C
C OUTPUT : FRA(3) - FORCE ON BAG                                    BGG
C           TORQ(3)- TORQUE ON BAG                                 BGG
C           TOB(3) - TORQUE ON CONTACT SURFACE                   BGG
C           VOL - VOLUME OF INTERSECTION                         BGG
C
C IMPLICIT REAL*8 (A-H,O-Z)                                        BGG
C DIMENSION A(3,3), ZA(3), DA(3,3), BFA(3), VA(3), WA(3), B(3,3), ZB(3), BGG
*       DB(3,3), BFB(3), VB(3), WB(3), FRA(3), TORQ(3), TQB(3), TV(3) BGG
C COMMON/CNSNTS/ PI, RADIANS, G, THIRD, EPS(24),                 BGG
*       UNITL, UNITM, UNITT, GRAVITY(3), TWOPI                    TWOPI
C COMMON/TEMPVS/ DUMMY(200), DAB(3,3), BA(3,4), TEMP(3,3), Y(3), CPA(3), BGG
*       CPB(3), PLANE(4,3), FORCE(3), CBB(3), VLM(3), FRB(3),     BGG
*       YFA(3), YFB(3), ZBB(3), T1(3), T2(3), T3(3), T4(3), T5(3), T6(3) BGG
*       , DMMY(34823)                                             80386
C NOTE: DUMMY IS USED BY SUBROUTINES AIRBAG AND AIRBGG.         BGG
C
C INITIALIZATION                                                  BGG
C
C S3TEST = 10.0                                                  BGG
C VOL=0.                                                          BGG
C DO 5 I=1,3                                                       BGG
C   FRA(I) = 0.0                                                  BGG
C   TORQ(I) = 0.0                                                 BGG
C   TQB(I) = 0.0                                                  BGG
C   BA(I,4)=-BFA(I)                                              BGG
C DO 5 J=1,3                                                       BGG

```

```

      BA(I,4)=BA(I,4)+DA(I,J)*(ZB(J)-ZA(J))      BGG
      DAB(I,J)=0.                                BGG
      DO 5 K=1,3                                  BGG
5     DAB(I,J)=DAB(I,J)+DA(I,K)*DB(J,K)         BGG
C
C     COMPUTE DISTANCE BETWEEN ELLIPSOID CENTERS AND BGG
C     CONVERT ELLIPSOID MATRIX OF OBJECT TO AIRBAG REFERENCE. BGG
C
      DO 10 I=1,3                                  BGG
      DO 10 J=1,3                                  BGG
      TEMP(I,J) = 0.0                             BGG
      BA(I,4)=BA(I,4)+DAB(I,J)*BFB(J)           BGG
      DO 10 K=1,3                                  BGG
10    TEMP(I,J) = TEMP(I,J) + B(I,K)*DAB(J,K)   BGG
      CALL MAT33(DAB,TEMP,BA)                    BGG
C
C     CHECK FOR INTERSECTION AND DETERMINE POINTS OF MAXIMUM PENETRATION BGG
C
      TB = 1.0                                     BGG
      CALL INTERS (A,BA,BA(1,4),TB,Y,TV(1),T1)   BGG
      IF (TB.GT.1.0) RETURN                       BGG
      CALL EDEPTH (A,BA,BA(1,4),TB,Y,CPA,CPB,TV(2),TV(3)) BGG
C
C     SET UP ORTHOGONAL SYSTEM USING VECTOR BETWEEN POINTS BGG
C     OF MAXIMUM PENETRATION AS ONE AXIS.        BGG
C
      P = 0.                                       BGG
      DO 20 I=1,3                                  BGG
      PLANE(I,3) = CPA(I)-CPB(I)                 BGG
20    P = PLANE(I,3)**2+P                         BGG
      IF (P.LT.EPS(6)) GO TO 99                 BGG
      PP = DSQRT(P)                               BGG
      DO 25 I=1,3                                  BGG
25    TEMP(I,1) = PLANE(I,3)/PP                 BGG
      CALL ORTHO(PLANE,TEMP,4)                  BGG
C
C     DEFINE PLANES AT MAXIMUM PENETRATION POINTS. BGG
C
      DO 40 I=1,3                                  BGG
      PLANE(4,I) = 0.0                           BGG
      DO 40 J=1,3                                  BGG
40    PLANE(4,I) = PLANE(4,I)+PLANE(J,I)*CPB(J) BGG
      DO 45 K=1,3                                  BGG
45    CBB(K)=CPB(K)-BA(K,4)                   BGG
C
C     ESTIMATES OF VOLUME AND AREA BASED ON RADII OF CURVATURE BGG
C     AND PENETRATION.                            BGG
C
      IF=2                                         BGG
      AREA=PI                                     BGG
      DO 70 L=1,2                                  BGG
      RA=RCRT(A,PLANE,CPA,L)                   BGG

```

```

RB=RCRT(BA, PLANE, CBB, L) BGG
IF(PP.GT.RA)RA=PP BGG
R=(RA-RB)*.5 BGG
RC=(RA+RB)*.5 BGG
VP=PP/(RA+RB) BGG
VD=VP BGG
ALP=RC*DSQRT(VP*(2.-VP)) BGG
IF(R.GE.0.)GO TO 60 BGG
AB=RA+RB-PP BGG
BET=(RA**2-RB**2+AB**2)*.5/AB BGG
ALP=DSQRT(RA**2-BET**2) BGG
R=0. BGG
VD=1.-BET/RA BGG
VP=(PP+BET-RA)/RB BGG
60 VLM(L)=RB*(RB*VP)**2*(1.-VP/3.)+RA*(RA*VD)**2*(1.-VD/3.) BGG
IF(R.GT.0.)VLM(L)=VLM(L)-ALP*R*R*(PI-2.*(DASIN(1.-VP)+
* (1.-VP)*ALP/RC)) BGG
VLM(L)=VLM(L)*PI BGG
AREA=AREA*ALP BGG
70 IP=1 BGG
VOL=(VLM(1)+VLM(2))*0.5 BGG
IF (IFULL.EQ.0) GO TO 99 BGG
C BGG
C SET UP FORCE VECTOR ALONG LINE OF MAXIMUM PENETRATION. BGG
C BGG
CALL DOT31(DAB,CBB,ZBB) BGG
DO 76 K=1,3 BGG
YFA(K)=CPB(K)+BFA(K) BGG
YFB(K)=ZBB(K)+BFB(K) BGG
FORCE(K) = -AREA*PLANE(K,3) BGG
76 T1(K) = VA(K)-VB(K) BGG
C BGG
C COMPUTE ANGULAR VELOCITY COMPONENTS,RELATIVE VELOCITY, COMPONENTS BGG
C OF RELATIVE VELOCITY ALONG MAX PENETRATION LINE AND MAGNITUDE OF BGG
C FORCE. BGG
C BGG
CALL MAT31(DA,T1,T2) BGG
CALL CROSS(WA,YFA,T1) BGG
CALL CROSS(WB,YFB,T3) BGG
CALL MAT31(DAB,T3,T4) BGG
FM = 0.0 BGG
SUM = 0.0 BGG
DO 77 K=1,3 BGG
T5(K) = T2(K)+T1(K)-T4(K) BGG
SUM = SUM+T5(K)*PLANE(K,3) BGG
77 FM = FM+FORCE(K)**2 BGG
C BGG
C COMPUTE COMPONENTS OF RELATIVE VELOCITY IN TANGENT PLANE, BGG
C FRICTION FORCE AND TOTAL FORCE VECTOR. BGG
C BGG
S3 = 0.0 BGG
DO 78 K=1,3 BGG

```


	T6(K) = T5(K) - SUM*PLANE(K, 3)	BGG
78	S3 = S3+T6(K)**2	BGG
	SQ3 = DSQRT(S3)	BGG
	IF (SQ3.LT.S3TEST) SQ3=S3TEST/(2.0-SQ3/S3TEST)	BGG
	FF = VSCS*DSQRT(FM)/SQ3	BGG
	DO 79 K=1, 3	BGG
79	FORCE(K) = FORCE(K) - FF*T6(K)	BGG
C		BGG
C	COMPUTE FRB: FORCE ON REACTION SURFACE IN ITS LOCAL REFERENCE.	BGG
C	TORQ: TORQUE ON AIRBAG IN AIRBAG REFERENCE.	BGG
C	TQB: TORQUE ON REACTION SURFACE IN ITS LOCAL REFERENCE.	BGG
C	FRA: FORCE ON AIRBAG IN INERTIAL REFERENCE.	BGG
C		BGG
	CALL DOT31(DAB, FORCE, FRB)	BGG
	CALL CROSS(YFA, FORCE, TORQ)	BGG
	CALL CROSS(FRB, YFB, TQB)	BGG
	CALL DOT31(DA, FORCE, FRA)	BGG
99	RETURN	BGG
	END	BGG

	SUBROUTINE BINPUT	REV IV 07/24/86	BINPUT
C			SLIP
C	READS THE INPUT CARDS THAT CONTAINS THE PHYSICAL DIMENSIONS AND		BINPUT
C	CHARACTERISTICS OF THE CRASH VICTIM'S BODY SEGMENTS AND JOINTS.		BINPUT
C			BINPUT
	IMPLICIT REAL*8(A-H,O-Z)		BINPUT
	COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,		BINPUT
*	NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG		PAGE
	COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60),		SLIP
*	RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),		BINPUT
*	JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)		BINPUT
	COMMON/CNTRSF/ PL(24,30),BELT(20,8),TPTS(6,8),BD(24,40)		EDGE
	COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),		BINPUT
*	BLTTTL(5,8),PLTTL(5,30),BAGTTL(5,6),SEG(30),		BINPUT
*	JOINT(30),CGS(30),JS(30)		BINPUT
	REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT		BINPUT
	LOGICAL*1 CGS,JS		BINPUT
	COMMON/INTEST/ SGTEST(3,4,30),XTEST(3,120),SEGT(120),REGT(120)		BINPUT
	REAL SEGT		BINPUT
	COMMON/FLXBLE/ HF(4,12,8),B42(3,3,24),V4(3,8),NFLEX(3,8)		BINPUT
	COMMON/GEULER/ IEULER(30),H1R(3,3,90),ANG(3,30),ANGD(3,30),		JDRIFT
*	FE(3,30),TQE(3,30),CONST(5,30)		JDRIFT
	COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),		BINPUT
*	UNITL,UNITM,UNITT,GRAVTY(3),TWOPI		TWOPI
	COMMON/RSAVE/ XSG(3,20,3),DPMI(3,3,30),LPMI(30),		ATBIII
*	NSG(9),MSG(20,9),MCG,MCGIN(24,5),KREF(20,9)		TTHKREF
	LOGICAL*1 EULER,SLIP,LDMY		80386
	COMMON/TEMPVS/ YPR1(3,30),YPR2(3,30),YPR3(3,30),YPRPMI(3,30),		BINPUT
*	T1(6),TMP1(3,3),TMP2(3,3),KNT(30),IDYPR(6,30),		SLIP
*	EULER(30),LDMY(2),DMY(34620)		80386
	DATA MXNSEG/30/,MXNJNT/30/,MNFLX/8/		MISC
	CALL ELTIME(1,2)		BINPUT
	IDYPRT = 0		TGMOD5
C			BINPUT
C	INPUT CARD B.1		BINPUT
C			BINPUT
	READ (5,11) NSEG,NJNT,BDYTTL		BINPUT
11	FORMAT (2I6,8X,5A4)		BINPUT
	IF (NSEG.GT.MXNSEG) STOP 77		MISC
	IF (NJNT.GT.MXNJNT) STOP 78		MISC
C			BINPUT
C	INPUT CARDS B.2.I FOR EACH SEGMENT		BINPUT
C			BINPUT
	DO 12 I=1,NSEG		BINPUT
	READ (5,13) SEG(I),CGS(I),W(I),(PHI(J,I),J=1,3),		BINPUT
*	(BD(J,I),J=1,6),LPMI(I)		BINPUT
13	FORMAT(A4,1X,A1,10F6.0,14)		BINPUT
	DO 81 J=1,3		BINPUT
	IDYPR(J,I) = 4 J		BINPUT
81	YPRPMI(J,I) = 0.0		BINPUT
	IF (LPMI(I).EQ.0) GO TO 12		BINPUT
	READ (5,82) (YPRPMI(J,I),J=1,3)		BINPUT

82	FORMAT(12X,3F6.0)	BINPUT
12	CALL DRCYPR (DPMI(1,1,I),YPRPMI(1,I),IDYPR(1,I))	BINPUT
C		BINPUT
C	INPUT CARDS B.3.J FOR EACH JOINT.	BINPUT
C		BINPUT
	NFLX = 0	BINPUT
	IF (NJNT.EQ.0) GO TO 27	BINPUT
	SLIP = .FALSE.	SLIP
	DO 14 J=1,NJNT	BINPUT
	READ (5,15) JOINT(J),JS(J),JNT(J),IPIN(J),(SR(I,2*J-1),I-1,3),	BINPUT
	* (SR(I,2*J),I-1,3),IEULER(J),CONST(1,J),CONST(2,J),	SLIP
	* (YPR1(I,J),I-1,3),(YPR2(I,J),I-1,3),	SLIP
	* (YPR3(I,J),I-1,3),(IDYPR(I,J),I-1,6)	BINPUT
	ID1 = IDYPR(1,J)	BINPUT
	ID4 = IDYPR(4,J)	BINPUT
	EULER(J) = .FALSE.	SLIP
	IF (IPIN(J).EQ.4) EULER(J) = .TRUE.	SLIP
	IF (IEULER(J).EQ.0.AND.IPIN(J).LE.-4) EULER(J) = .TRUE.	SLIP
	IF (.NOT.EULER(J).AND.IABS(IPIN(J)).GE.5) SLIP = .TRUE.	SLIP
	IF(ID1.NE.0.OR.ID4.NE.0) IDYPRT = 1	TGMOD5
	DO 479 II=1,6	TGMOD5
479	IF(IABS(IDYPR(II,J)).GT.3) STOP 101	TGMOD5
	DO 14 I=1,3	BINPUT
	IF (ID1.EQ.0) IDYPR(I ,J) = 4-I	BINPUT
14	IF (ID4.EQ.0) IDYPR(I+3,J) = 4-I	BINPUT
15	FORMAT(A4,1X,A1,2I4,6F6.0,I4,2F6.0/14X,9F6.0,6I2)	SLIP
C		BINPUT
C	COMPUTE NFLX AND NFLEX ARRAY FROM NEGATIVE VALUES OF JNT(J).	BINPUT
C	NFLX WILL BE NUMBER OF CONSTRAINT TORQUES FOR FLEXIBLE SEGMENTS.	BINPUT
C	NFLEX(1,) REFERENCE SEGMENT (LOWEST NUMBERED SEGMENT OF CHAIN)	BINPUT
C	NFLEX(2,) INTERIOR SEGMENT NUMBERS	BINPUT
C	NFLEX(3,) TERMINATING SEGMENT (HIGHEST NUMBERED SEGMENT IN CHAIN)	BINPUT
C	VALUES OF NFLEX NEED NOT BE SEQUENTIAL BUT MUST BE ORDERED.	BINPUT
C	FLEXIBLE SEGMENT MUST BE SIMPLE CHAIN, I.E., BRANCHING SEGMENTS	BINPUT
C	CANNOT BE ATTACHED TO INTERIOR SEGMENTS BUT MAY BE ATTACHED TO	BINPUT
C	REFERENCE OR TERMINATING SEGMENTS.	BINPUT
C		BINPUT
	DO 16 J=1,NJNT	BINPUT
16	KNT(J) = JNT(J)	BINPUT
	DO 22 J=1,NJNT	BINPUT
	IF (KNT(J).GE.0) GO TO 22	BINPUT
	NFA = NFLX+1	BINPUT
	IT = J+1	BINPUT
	IF (IT.GT.NJNT) GO TO 18	BINPUT
	JP1 = J+1	BINPUT
	DO 17 L=JP1,NJNT	BINPUT
	IF (IABS(KNT(L)).NE.IT) GO TO 17	BINPUT
	KL = KNT(L)	BINPUT
	KNT(L) = 0	BINPUT
	IF (KL.GT.0) GO TO 18	BINPUT
	NFLX = NFLX+1	BINPUT
	NFLEX(1,NFLX) = IABS(KNT(J))	BINPUT

	NFLEX(2,NFLX) - IT	BINPUT
	IT - L+1	BINPUT
	17 CONTINUE	BINPUT
	18 IF (NFLX.GE.NFA) GO TO 20	BINPUT
	WRITE (6,19)	BINPUT
	19 FORMAT('OERROR IN DEFINING FLEXIBLE SEGMENTS, ONLY ONE NEGATIVE JN	BINPUT
	*T IN STRING. PROGRAM TERMINATED.')	BINPUT
	STOP 3	BINPUT
	20 DO 21 K=NFA,NFLX	BINPUT
	21 NFLEX(3,K) - IT	BINPUT
	22 CONTINUE	BINPUT
C		BINPUT
C	INPUT CARDS B.4.J FOR EACH JOINT.	BINPUT
C		BINPUT
	DO 23 J=1,NJNT	BINPUT
	READ (5,24) (SPRING(I,3*J-2),I-1,5),(SPRING(I,3*J-1),I-1,5)	BINPUT
	23 IF (EULER(J)) READ(5,24)(SPRING(I,3*J),I-1,5),(ANG(I,J),I-1,3)	SLIP
	24 FORMAT(2(4F6.0,F12.0))	BINPUT
C		BINPUT
C	INPUT CARDS B.5.J FOR EACH JOINT.	BINPUT
C		BINPUT
	DO 25 J=1,NJNT	BINPUT
	READ (5,26) (VISC(I,3*J-2),I-1,7)	BINPUT
	IF (.NOT.EULER(J)) GO TO 25	SLIP
	READ (5,26) (VISC(I,3*J-1),I-1,7)	BINPUT
	READ (5,26) (VISC(I,3*J),I-1,7)	BINPUT
	25 CONTINUE	BINPUT
	26 FORMAT(5F6.0,18X,2F6.0)	BINPUT
C		BINPUT
C	INPUT CARDS B.6.I FOR EACH SEGMENT.	BINPUT
C		BINPUT
	27 DO 28 I=1,NSEG	BINPUT
	28 READ (5,29) ((SGTEST(J,K,I),J=1,3),K=1,4)	BINPUT
	29 FORMAT(12F6.0)	BINPUT
C		BINPUT
C	PRINT CARD B.1	BINPUT
C		BINPUT
	WRITE (6,30) BDYTTL,NSEG,NJNT,NPG,UNITM,UNITT,UNITL,UNITL,	PAGE
	* UNITL,UNITM	PAGE
	NPG=NPG+1	PAGE
	30 FORMAT('1 CRASH VICTIM',5X,5A4,I5,' SEGMENTS',I5,' JOINTS',58X,	PAGE
	* 'PAGE',I5/120X,'CARD B.1'/25X,'PRINCIPAL MOMENTS OF INERTIA',	PAGE
	* 14X,'SEGMENT CONTACT ELLIPSOID',28X,'CARDS B.2'/	BINPUT
	* 3X,'SEGMENT',6X,'WEIGHT',7X,'(',A4,'-',A4,'**2',A4,')',	BINPUT
	* 11X,'SEMIAXES (',A4,')',12X,'CENTER (',A4,')',	BINPUT
	* 11X,'PRINCIPAL AXES (DEG)'/	BINPUT
	* ' I SYM PLOT (',A4,')',7X,'X',8X,'Y',8X,'Z ',	BINPUT
	* 2(9X,'X',7X,'Y',7X,'Z'),8X,'YAW',5X,'PITCH',5X,'ROLL'/)	BINPUT
C		BINPUT
C	PRINT CARDS B.2.I FOR EACH SEGMENT.	BINPUT
C		BINPUT
	DO 31 I=1,NSEG	BINPUT

31	WRITE (6,32) I,SEG(I),CGS(I),W(I),(PHI(J,I),J-1,3),	BINPUT
*	(BD(J,I),J-1,6),(YPRPMI(J,I),J-1,3)	BINPUT
32	FORMAT(I3,1X,A4,2X,A1,F11.3,2X,3F9.4,2(2X,3F8.3),1X,3F9.2)	BINPUT
	IF (NJNT.EQ.0) GO TO 50	BUTLER1
C		BINPUT
C	PRINT CARDS B.3.J FOR EACH JOINT.	BINPUT
C		BINPUT
	IF(IDYPRT.EQ.0) WRITE(6,33) UNITL,UNITL	TGMOD5
	IF(IDYPRT.EQ.1) WRITE(6,733) UNITL,UNITL	TGMOD5
33	FORMAT(///120X,'CARDS B.3'/	BINPUT
*	3X,'JOINT',15X,'LOCATION(' ,A4,') - SEG(JNT)',	BINPUT
*	3X,'LOCATION(' ,A4,') - SFJ(J+1)',	BINPUT
*	2X,'PRIN. AXIS(DEG) - SEG(JNT)',	BINPUT
*	2X,'PRIN. AXIS(DEG) - SEG(J+1)'/	BINPUT
*	' J SYM PLOT JNT PIN', 2(6X,'X',8X,'Y',8X,'Z',3X),	BINPUT
*	2(5X,'YAW',5X,'PITCH',5X,'ROLL',1X) /)	BINPUT
733	FORMAT(///120X,'CARDS B.3'/	TGMOD5
*	3X,'JOINT',15X,'LOCATION(' ,A4,') - SEG(JNT)',	TGMOD5
*	3X,'LOCATION(' ,A4,') - SEG(J+1)',	TGMOD5
*	2X,'PRIN. AXIS(DEG) - SEG(JNT)',	TGMOD5
*	2X,'PRIN. AXIS(DEG) - SEG(J+1)'/	TGMOD5
*	' J SYM PLOT JNT PIN', 2(6X,'X',8X,'Y',8X,'Z',3X),	TGMOD5
*	'ID1 YAW ID2 PITCH ID3 ROLL ',	TGMOD5
*	'ID4 YAW ID5 PITCH ID6 ROLL ',/)	TGMOD5
DO 34	J=1,NJNT	BINPUT
IF(IDYPRT.EQ.0)		TGMOD5
*WRITE (6,35) J,JOINT(J),JS(J),JNT(J),JPIN(J),	(SR(I,2*J-1),I-1,3),	TGMOD5
*	(SR(I,2*J),I-1,3),(YPR1(I,J),I-1,3),(YPR2(I,J),I-1,3)	BINPUT
IF(IDYPRT.EQ.1)		TGMOD5
*WRITE(6,735) J,JOINT(J),JS(J),JNT(J),JPIN(J),	(SR(I,2*J-1),I-1,3),	TGMOD5
*	(SR(I,2*J),I-1,3),(IDYPR(I,J),YPR1(I,J),I-1,3),	TGMOD5
*	(IDYPR(I+3,J),YPR2(I,J),I-1,3)	TGMOD5
IF (.NOT.EULER(J)) GO TO 34		SLIP
IEULER(J) = 8		BINPUT
IF (IPIN(J).EQ.4) GO TO 34		BINPUT
IEULER(J) = 11 + IPIN(J)		BINPUT
IPIN(J) = -4		BINPUT
34	CONTINUE	BINPUT
35	FORMAT(I3,1X,A4,2X,A1,2X,2I3,2(1X,3F9.3),2(1X,3F9.2))	BINPUT
735	FORMAT(I3,1X,A4,2X,A1,2X,2I3,2(1X,3F9.3),2(1X,3(1X,I1,F7.2)))	TGMOD5
IF (.NOT.SLIP) GO TO 89		SLIP
WRITE (6,83) UNITM,UNITM		SLIP
83	FORMAT(// ' UNLOCK CONDITIONS FOR SLIP JOINTS'/	SLIP
*	' JOINT TENSION COMPRESSION'/	SLIP
*	14X,'(' ,A4,')',7X,'(' ,A4,')' /)	SLIP
DO 85	J = 1,NJNT	SLIP
IF (EULER(J)) GO TO 85		SLIP
IF (IABS(IPIN(J)).LT.5) GO TO 85		SLIP
WRITE (6,84) J,CONST(1,J),CONST(2,J)		SLIP
84	FORMAT(1X,I6,4X,F10.3,3X,F10.3)	SLIP
85	CONTINUE	SLIP
C		BINPUT

C	SET UP HT MATRIX FROM YPR1 & YPR2 INPUT.	BINPUT
C	HA IS 3RD COLUMN & HB IS 2ND COLUMN OF HT.	BINPUT
C	FOR A SLIP JOINT(IPIN=7),HB IS 3RD COLUMN OF HT.	SLIP
C		BINPUT
89	IF (NPRT(23).NE.0) WRITE (6,36) NPG	SLIP
	IF (NPRT(23).NE.0) NPG=NPG+1	PAGE
	36 FORMAT('1 HT ARRAY AS COMPUTED FROM YPR1 & YPR2 INPUT.',77X,	PAGE
	* 'PAGE',15)	PAGE
	DO 38 J=1,NJNT	BINPUT
	SR(4,2*J-1) = 0.0	SLIP
	SR(4,2*J) = 0.0	SLIP
	CALL DRCYPR (TMP1,YPR1(1,J),IDYPR(1,J))	BINPUT
	CALL DRCYPR (TMP2,YPR2(1,J),IDYPR(4,J))	BINPUT
	DO 37 I=1,3	BINPUT
	ANGD(I,J) = 0.0	BINPUT
	HA(I,2*J-1) = 0.0	BINPUT
	HA(I,2*J) = 0.0	BINPUT
	K = 2	SLIP
	IF (IABS(IPIN(J)).EQ.7) K = 3	SLIP
	HB(I,2*J-1) = TMP1(K,I)	SLIP
	HB(I,2*J) = TMP2(K,I)	SLIP
	DO 77 K=1,3	SLIP
	HT(I,K,2*J-1) = TMP1(K,I)	SLIP
77	HT(I,K,2*J) = TMP2(K,I)	SLIP
	IF (.NOT.EULER(J)) GO TO 37	SLIP
	CONST(I,J) = YPR3(I,J)*RADIAN	SLIP
	ANG(I,J) = ANG(I,J)*RADIAN - CONST(I,J)	SLIP
	37 CONTINUE	SLIP
	38 IF (NPRT(23).NE.0) WRITE (6,39) J,JOINT(J),	BINPUT
	* ((HT(I,K,2*J-1),K=1,3),(HT(I,K,2*J),K=1,3),I=1,3)	BINPUT
	39 FORMAT('0',I4,2X,A4,3X,3F12.6,3X,3F12.6/(14X,3F12.6,3X,3F12.6))	BINPUT
C		BINPUT
C	PRINT CARDS B.4.J FOR EACH JOINT.	BINPUT
C		BINPUT
	WRITE (6,41) NPG,UNITL,UNITM,UNITL,UNITM	PAGE
	NPG=NPG+1	PAGE
	41 FORMAT('1 JOINT TORQUE CHARACTERISTICS',93X,	PAGE
	* 'PAGE',15/120X,'CARDS B.4'/	PAGE
	*23X,'FLEXURAL SPRING CHARACTERISTICS',28X,'TORSIONAL SPRING' ,	BINPUT
	*' CHARACTERISTICS'//	BINPUT
	*15X,'SPRING COEF. (' ,2A4,'/DEG**J)',6X,'ENERGY JOINT',	BINPUT
	* 7X,'SPRING COEF. (' ,2A4,'/DEG**J)',6X,'ENERGY JOINT'/	JBINPUT
	*OINT ',2(8X.'LINEAR QUADRATIC CUBIC DISSIPATION STOP ')	BINPUT
	*/8X,2(8X,'(J=1)',7X,'(J=2)',7X,'(J=3)',7X,'COEF. (DEG)')//	BINPUT
	DO 42 J=1,NJNT	BINPUT
	J1 = 3*J-2	BINPUT
	J2 = 3*J-1	BINPUT
	J3 = 3*J	BINPUT
	WRITE (6,43) J,JOINT(J),((SPRING(I,JJ),I=1,5),JJ=J1,J2)	BINPUT
42	IF (EULER(J)) WRITE (6,44) (SPRING(I,J3),I=1,5)	SLIP
43	FORMAT(I3,1X,A4,2(3X,3F12.3,2F10.3))	BINPUT
44	FORMAT(11X,3F12.3,2F10.3)	BINPUT

C		BINPUT
C	PRINT CARDS B.5.J FOR EACH JOINT.	BINPUT
C		BINPUT
	WRITE (6,46) (UNITL,UNITM,UNITT,I-1,2),(UNITL,UNITM,I-1,2),UNITT	BINPUT
46	FORMAT(///120X,'CARDS B.5'/	BINPUT
	*38X,'JOINT VISCOUS CHARACTERISTICS AND LOCK-UNLOCK CONDITIONS'//	BINPUT
	*14X,'VISCOUS',9X,'COULOMB',7X,'FULL FRICTION',5X,'MAX TORQUE FOR',	BINPUT
	*4X,'MIN TORQUE FOR',4X,'MIN. ANG. VELOCITY',6X,'IMPULSE'/	BINPUT
	*2X,'JOINT',5X,'COEFFICIENT',4X,'FRICTION COEF. ANGULAR VELOCITY',	BINPUT
	*4X,'A LOCKED JOINT',4X,'UNLOCKED JOINT',4X,'FOR UNLOCKED JOINT',	BINPUT
	*4X,'RESTITUTION'/	BINPUT
	*8X,'(',3A4,'/DEG) (' ,2A4,')',6X,'(DEG/' ,A4,')',10X,'(' ,2A4,')',	BINPUT
	*8X,'(' ,2A4,')',10X,'(RAD/' ,A4,')',8X,'COEFFICIENT'/)	BINPUT
	DO 47 J=1,NJNT	BINPUT
	J1 = 3*J-2	BINPUT
	J2 = 3*J-1	BINPUT
	J3 = 3*J	BINPUT
	WRITE (6,48) J,JOINT(J),(VISC(I,J1),I-1,7)	BINPUT
47	IF (EULER(J)) WRITE (6,49) ((VISC(I,JJ),I-1,7),JJ-J2,J3)	SLIP
48	FORMAT(I3,1X,A4,F13.3,2F15.2,F22.2,F18.2,F20.2,F17.3)	BINPUT
49	FORMAT(8X,F13.3,2F15.2,F22.2,F18.2,F20.2,F17.3)	BINPUT
C		BINPUT
C	PRINT CARDS B.6. FOR EACH SEGMENT.	BINPUT
C		BINPUT
	50 WRITE (6,51) NPG.(UNITT,UNITL,UNITT,I-1,2)	PAGE
	NPG=NPG+1	PAGE
	51 FORMAT('1',122X,'PAGE',I5/20X,	PAGE
	* 'SEGMENT INTEGRATION CONVERGENCE TEST INPUT',58X,'CARDS B.6'//PAGE	
	* 17X,'ANGULAR VELOCITIES', 11X,'LINEAR VELOCITIES',	BINPUT
	* 10X,'ANGULAR ACCELERATIONS',9X,'LINEAR ACCELERATIONS'//	BINPUT
	* 21X,'(RAD/' ,A4,')', 18X,'(' ,A4, '/' ,A4,')',	BINPUT
	* 17X,'(RAD/' ,A4, '**2)', 16X,'(' ,A4, '/' ,A4, '**2)'/	BINPUT
	* ' SEGMENT', 4(' MAG. ABS. REL. ') /	BINPUT
	* ' NO. SYM', 4(' TEST ERROR ERROR') //)	BINPUT
	DO 52 I=1,NSEG	BINPUT
52	WRITE (6,53) I,SEG(I),((SGTEST(J,K,I),J-1,3),K-1,4)	BINPUT
53	FORMAT(I3,1X,A4,4(F11.3,F9.3,F9.4))	BINPUT
	IF (NFLX.EQ.0) GO TO 62	BINPUT
C		BINPUT
C	INPUT AND PRINT CARDS B.7	BINPUT
C	CARD B.7.A NFX: NO. OF INTERIOR SEGMENTS OF FLEXIBLE ELEMENTS.	BINPUT
C	KNT(J),J-1,NFX: THE SEGMENT NUMBERS.	BINPUT
C		BINPUT
	READ (5,54) NFX,(KNT(J),J-1,NFX)	BINPUT
54	FORMAT(18I4)	BINPUT
	IF (NFX.NE.NFLX) WRITE (6,55) NFX,NFLX	BINPUT
55	FORMAT('OINPUT ERROR ON CARD B.7.A, NFX =',I4, ' BUT NFLX =',I4/	BINPUT
	* ' AS COMPUTED FROM CARDS B.3. PROGRAM TERMINATED.')	BINPUT
	IF (NFX.NE.NFLX) STOP 4	BINPUT
	WRITE (6,56) NPG	PAGE
	NPG=NPG+1	PAGE
56	FORMAT('1',122X,'PAGE',I5/121X,'CARDS B.7')	PAGE

	DO 60 JJ=1,NFX	BINPUT
	DO 57 K=1,NFLX	BINPUT
	IF (KNT(JJ).EQ.NFLEX(2,K)) GO TO 59	BINPUT
	57 CONTINUE	BINPUT
	WRITE (6,58) KNT(JJ)	BINPUT
	58 FORMAT('OINPUT ERROR ON CARD B.7.J, SEGMENT NO.',I4,' IS NOT AN	INBINPUT
	*TERIOR SEGMENT OF A FLEXIBLE ELEMENT FROM DATA ON CARDS B.3.'/	BINPUT
	* ' PROGRAM TERMINATED.')	BINPUT
	STOP 5	BINPUT
	59 IF(NFLX.GT.MNFLX) STOP 99	TGMOD5
C		BINPUT
C	CARDS B.7.J HF ARRAY FOR SEGMENT KNT(JJ)	BINPUT
C		BINPUT
	READ (5,29) ((HF(I,J,K),J=1,12),I=1,4)	TGMOD5
	DO 737 LL=1,3	TGMOD5
	L = (LL-1)*4	TGMOD5
	DO 737 I=1,4	TGMOD5
	DO 737 J=1,4	TGMOD5
	737 IF(HF(I,J+L,K).NE.HF(J,I+L,K)) STOP 100	TGMOD5
	60 WRITE (6,61) KNT(JJ),K,(NFLEX(I,K),I=1,3),	BINPUT
	* ((HF(I,J,K),J=1,12),I=1,4)	BINPUT
	61 FORMAT('O HF ARRAY FOR INTERIOR SEGMENT NO.',I4,20X,	BINPUT
	* '(NFLEX(I,' ,I1,'),I=1,3) =',3I6//	BINPUT
	* (3X,4F10.4,3X,4F10.4,3X,4F10.4))	BINPUT
	62 IF (NJNT.EQ.0) GO TO 65	BINPUT
C		BINPUT
C	CHANGE SPRING AND VISC FROM DEG TO RAD	BINPUT
C		BINPUT
	DO 64 I=1,NJNT	BINPUT
	J1 = 3*I-2	BINPUT
	J2 = 3*I-1	BINPUT
	IF (EULER(I)) J2= 3*I	SLIP
	DO 63 J=J1,J2	BINPUT
	SPRING(1,J) = SPRING(1,J)/RADIAN	BINPUT
	SPRING(2,J) = SPRING(2,J)/RADIAN**2	BINPUT
	SPRING(3,J) = SPRING(3,J)/RADIAN**3	BINPUT
	SPRING(5,J) = SPRING(5,J)*RADIAN	BINPUT
	63 CONTINUE	BINPUT
	IF (.NOT.EULER(I)) J2 = J1	SLIP
	DO 64 J=J1,J2	BINPUT
	VISC (1,J) = VISC (1,J)/RADIAN	BINPUT
	64 VISC (3,J) = VISC (3,J)*RADIAN	BINPUT
C		BINPUT
C	W ARRAY HAS BEEN SUPPLIED IN LBS. SET UP RECIPROCAL MASS (RW)	BINPUT
C	AND MOMENT OF INERTIA (RPHI) ARRAYS. HOWEVER, IF W OR ANY ELEMENT	BINPUT
C	OF PHI IS ZERO, SEGMENT WILL BE CONSIDERED SINGULAR (ISING=1) AND	BINPUT
C	ALL RECIPROCAL WILL BE ZERO SO AS TO NULLIFY COMPUTATIONS IN THE	BINPUT
C	DAUX ROUTINES. NS IS THE NUMBER OF SINGULAR SEGMENTS.	BINPUT
C		BINPUT
	65 NS = 0	BINPUT
	DO 68 I=1,NSEG	BINPUT
	ISING(I) = 0	BINPUT

RW(I) = 0.0	BINPUT
IF (W(I).EQ.0.0) ISING(I) = 1	BINPUT
DO 66 K=1,3	BINPUT
IF (PHI(K,I).EQ.0.0) ISING(I) = 1	BINPUT
66 RPHI(K,I) = 0.0	BINPUT
IF (ISING(I).EQ.1) NS = NS+1	BINPUT
IF (ISING(I).EQ.1) GO TO 68	BINPUT
RW(I) = G/W(I)	BINPUT
DO 67 K=1,3	BINPUT
67 RPHI(K,I) = 1.0/PHI(K,I)	BINPUT
68 CONTINUE	BINPUT
C	BINPUT
C SET UP ELLIPSOID MATRIX AND INVERSE (ASSUME YAW,PITCH,ROLL = 0)	BINPUT
C FOR 1ST NSEG ELLIPSOIDS IN BD(7-15) AND BD(16-24).	BINPUT
C	BINPUT
DO 71 J=1,NSEG	BINPUT
DO 70 I=7,24	BINPUT
70 BD(I,J) = 0.0	BINPUT
DO 71 I=1,3	BINPUT
BD(4*I+3,J) = 1.0/BD(I,J)**2	BINPUT
71 BD(4*I+12,J) = BD(I,J)**2	BINPUT
RETURN	BINPUT
END	BINPUT

	SUBROUTINE BLKDTA		REV IV 07/23/86	TWOPI	BLKDTA
C					BLKDTA
C	THIS SUBROUTINE REPLACES THE BLOCK DATA SUBPROGRAM OF PREVIOUS				BLKDTA
C	VERSIONS OF CVS-III TO INITIALIZE COMMON/CNSNTS/ IN A MANNER				BLKDTA
C	THAT IS INDEPENDENT OF THE COMPUTER SYSTEM BEING UTILIZED.				BLKDTA
C					BLKDTA
	IMPLICIT REAL*8 (A-H,O-Z)				BLKDTA
	COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),				BLKDTA
	* UNITL,UNITM,UNITT,GRAVTY(3),TWOPI				TWOPI
	COMMON/TEMPVS/ ZERO,ONE,THREE,TEN,ONE80,DMMY(35108)				80386
	DATA UM/8H LBS / , UT/8H SEC / , UL/8H IN /				BLKDTA
	ZERO = 0.0				BLKDTA
	ONE = 1.0				BLKDTA
	UNITM = UM				BLKDTA
	UNITT = UT				BLKDTA
	UNITL = UL				BLKDTA
	G = 386.088D0				BLKDTA
	GRAVTY(1) = ZERO				BLKDTA
	GRAVTY(2) = ZERO				BLKDTA
	GRAVTY(3) = G				BLKDTA
	THREE = 3.0				BLKDTA
	TEN = 10.0				BLKDTA
	ONE80 = 180.0				BLKDTA
	PI = DATAN2(ZERO,-ONE)				BLKDTA
	TWOPI = 2.0*PI				TWOPI
	RADIAN = PI/ONE80				BLKDTA
	THIRD = ONE/THREE				BLKDTA
	EPS(1) = ONE/TEN				BLKDTA
	DO 10 I=2,24				BLKDTA
10	EPS(I) = EPS(I-1)/TEN				BLKDTA
	RETURN				BLKDTA
	END				BLKDTA

	SUBROUTINE CFACTT(A,B,D)		CFACTT
C		REV 03	05/31/73CFACTT
C	GIVEN 3X3 MATRIX A		CFACTT
C	COMPUTE B TRANSPOSE OF COFACTORS (SIGNED MINORS)		CFACTT
C	AND D THE VALUE OF THE DETERMINANT OF A.		CFACTT
C	INVERSE OF A IS B(J,K)/D		CFACTT
C			CFACTT
	IMPLICIT REAL*8 (A-H,O-Z)		CFACTT
	DIMENSION A(3,3),B(3,3)		CFACTT
	M = 4		CFACTT
	L = 2		CFACTT
	N = 3		CFACTT
	D = 0.0		CFACTT
	DO 20 J=1,3		CFACTT
	B(J,J) = A(L,L)*A(N,N)-A(L,N)*A(N,L)		CFACTT
	IF (J.EQ.3) GO TO 20		CFACTT
	L = N		CFACTT
	N = J		CFACTT
	KK = J+1		CFACTT
	DO 15 K=KK,3		CFACTT
	M = M-1		CFACTT
	B(K,J) = A(K,M)*A(M,J)-A(K,J)*A(M,M)		CFACTT
15	B(J,K) = A(J,M)*A(M,K)-A(J,K)*A(M,M)		CFACTT
20	D = D+A(1,J)*B(J,1)		CFACTT
	RETURN		CFACTT
	END		CFACTT

```

SUBROUTINE CHAIN(ISKIP)                                JDRIFT
C                                                       REV IV 07/24/86SLIP
C COMPUTES THE LINEAR POSITION AND VELOCITY IN INERTIAL REFERENCE CHAIN
C OF BODY SEGMENTS FROM THOSE OF THE REFERENCE SEGMENTS CHAIN
C (I.E., SEGMENT NO. 1 AND EACH SEGMENT J FOR WHICH JNT(J)=0). CHAIN
C CHAIN
C IMPLICIT REAL*8(A-H,O-Z) CHAIN
COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND, CHAIN
* NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG PAGE
COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30), CHAIN
* SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30) CHAIN
COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60), SLIP
* RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90), CHAIN
* JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30) CHAIN
COMMON/CEULER/ IEULER(30),HIR(3,3,90),ANG(3,30),ANGD(3,30), SLIP
* FE(3,30),TQE(3,30),CONST(5,30) SLIP
COMMON/TEMPVS/ T1(3),T2(3),T3(3),T4(3),T5(3),T6(3),T7(3) SLIP
* ,DMMY(35092) 80386
DATA IFIRST/1/ SLIP
CALL ELTIME (1,11) ATBIII
IF (NJNT.EQ.0) GO TO 71 ATBIII
IF (ISKIP.NE.0) CALL DRIFT JDRIFT
DO 70 J=1,NJNT ATBIII
K = IABS(JNT(J)) ATBIII
IF (K.EQ.0) GO TO 70 ATBIII
IF (ISING(J+1).LT.0) GO TO 70 ATBIII
C ATBIII
C COMPUTE SEGMENT POSITIONS BY ATBIII
C P(J+1) = P(K) + D(K)'*R(K,J) - D(J+1)'*R(J+1,J) ATBIII
C ATBIII
C COMPUTE SEGMENT VELOCITIES BY ATBIII
C V(J+1) = V(K) + D(K)'*W(K) X R(K,J) - D(J+1)'*W(J+1) X R(J+1,J) ATBIII
C ATBIII
CALL CROSS (WMEG(1,K),SR(1,2*J-1),T1) JDRIFT
CALL DOT31 (D(1,1,K),T1,T3) ATBIII
CALL CROSS (WMEG(1,J+1),SR(1,2*J),T2) ATBIII
CALL DOT31 (D(1,1,J+1),T2,T4) ATBIII
CALL DOT31 (D(1,1,K),SR(1,2*J-1),T1) ATBIII
CALL DOT31 (D(1,1,J+1),SR(1,2*J),T2) ATBIII
IF (IABS(IPIN(J)).LT.5) GO TO 50 SLIP
IF (IEULER(J).EQ.-1)GO TO 50 SLIP
IF (IFIRST.EQ.1) GO TO 50 SLIP
DO 40 I = 1,3 SLIP
T5(I) = SEGLP(I,J+1) + T2(I) - SEGLP(I,K) - T1(I) SLIP
40 T6(I) = SEGLV(I,J+1) + T4(I) - SEGLV(I,K) - T3(I) SLIP
CALL DOT31 (D(1,1,K),HT(1,3,2*J-1),T7) SLIP
SR(4,2*J-1) = T5(1)*T7(1) + T5(2)*T7(2) + T5(3)*T7(3) SLIP
SR(4,2*J) = T6(1)*T7(1) + T6(2)*T7(2) + T6(3)*T7(3) SLIP
CALL CROSS (WMEG(1,K),HT(1,3,2*J-1),T5) SLIP
CALL DOT31 (D(1,1,K),T5,T6) SLIP
DO 45 I = 1,3 SLIP
T1(I) = T1(I) + SR(4,2*J-1)*T7(I) SLIP

```

45	T3(I) = T3(I) + SR(4,2*J)*T7(I) + SR(4,2*J-1)*T6(I)	SLIP
50	DO 60 I=1,3	SLIP
	SEGLP(I,J+1) = SEGLP(I,K) + T1(I) - T2(I)	ATBIII
60	SEGLV(I,J+1) = SEGLV(I,K) + T3(I) - T4(I)	ATBIII
70	CONTINUE	CHAIN
	IFIRST = 0	SLIP
C		CHAIN
C	OPTIONAL OUTPUT	CHAIN
C		CHAIN
71	IF (NPRT(20).NE.0) WRITE(6,90) TIME	CHAIN
*	,((SEGLP(I,J),I=1,3),J=1,NSEG)	CHAIN
*	,((SEGLV(I,J),I=1,3),J=1,NSEG)	CHAIN
90	FORMAT('0 LINEAR POSITIONS AND VELOCITIES OF BODY SEGMENTS FROM CH	CHAIN
	*AIN FOR TIME =',F12.6/(9F13.5))	CHAIN
	CALL ELTIME(2,11)	CHAIN
	RETURN	CHAIN
	END	CHAIN

	SUBROUTINE CINPUT	CINPUT
		REV III.2 08/08/84REVIII
C	INPUT CARDS E.1 - E.4 FOR THE FORCE-DEFLECTION, INERTIAL SPIKE,	CINPUT
C	R FACTOR, G FACTOR AND FRICTION COEFFICIENT FUNCTION DEFINITIONS	CINPUT
C		CINPUT
	IMPLICIT REAL*8(A-H,O-Z)	CINPUT
	COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,	PAGE
*	NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG	PAGE
	COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(1250),TAB(4500)	DIMENB
	COMMON/TEMPVS/JTITLE(5,51),NF(5),NT(3),KTITLE(31),DMMY(34966)	80386
	REAL JTITLE,KTITLE	CINPUT
C		CINPUT
	IS = 0	CINPUT
	DO 10 I = 1,50	CINPUT
10	NTI(I) = 0	CINPUT
	J1 = 1	CINPUT
C		CINPUT
C	INPUT CARD E.1 - FUNCTION NO. AND TITLE, IF NO. > 50 SKIP OUT.	CINPUT
C		CINPUT
11	READ(5,12) I,(KTITLE(J),J = 1,5)	CINPUT
12	FORMAT (I4,4X,5A4)	CINPUT
	IF (I.GT.50) GO TO 30	CINPUT
	DO 13 J = 1,5	CINPUT
13	JTITLE(J,I) = KTITLE(J)	CINPUT
C		CINPUT
C	HAS FUNCTION NO. BEEN ALREADY USED?	CINPUT
C		CINPUT
	IF (NTI(I).NE.0) WRITE(6,14) I	CINPUT
14	FORMAT('0 FUNCTION NO.',I4,' HAS ALREADY BEEN INPUTTED AND WILL BEC	CINPUT
	*REPLACED BY NEXT FUNCTION')	CINPUT
	NTI(I) = J1	CINPUT
	J2 = J1+4	CINPUT
C		CINPUT
C	INPUT CARD E.2	CINPUT
C		CINPUT
	READ(5,15) (TAB(J),J = J1,J2)	CINPUT
15	FORMAT (6F12.0)	CINPUT
	IS = 1-IS	CINPUT
	IF (IS.EQ.0) WRITE(6,16)	CINPUT
	IF (IS.EQ.0) GOTO 40	PAGE
	WRITE(6,41) NPG	PAGE
41	FORMAT('1',122X,'PAGE',I5)	PAGE
	NPG=NPG+1	PAGE
16	FORMAT(/////)	CINPUT
40	WRITE(6,17) I,(JTITLE(J,I),J=1,5),I,NTI(I),(TAB(J),J=J1,J2)	PAGE
17	FORMAT(' FUNCTION NO.',I4,4X,5A4,20X,'NTI(',I2,') =',I5,45X,	PAGE
	* 'CARDS E'//10X,'D0',13X,'D1',13X,'D2',13X,'D3',13X,'D4'/5F15.4//)	CINPUT
	DO = TAB(J1)	CINPUT
	D1 = TAB(J1+1)	CINPUT
	D2 = TAB(J1+2)	CINPUT
	J1 = J2+1	CINPUT
	IF (D1) 22,18,20	CINPUT

C		CINPUT
C	FUNCTION IS CONSTANT D2 FOR ALL D.	CINPUT
C		CINPUT
	18 WRITE(6,19) D2	CINPUT
	19 FORMAT(7X,'FUNCTION IS CONSTANT',F12.6)	CINPUT
	GO TO 11	CINPUT
C		CINPUT
C	5TH ORDER POLYNOMIAL ... 1ST FUNCTION	CINPUT
C	INPUT CARD E.3	CINPUT
C		CINPUT
	20 J2 = J1+5	CINPUT
	READ(5,15)(TAB(J),J = J1,J2)	CINPUT
	WRITE(6,21) (TAB(J),J = J1,J2)	CINPUT
	21 FORMAT(7X,'FIRST PART OF FUNCTION - 5TH DEGREE POLYNOMIAL'//	CINPUT
	* 8X,'A0',13X,'A1',13X,'A2',13X,'A3',13X,'A4',13X,'A5',13X/CINPUT	CINPUT
	* 6F15.6//)	CINPUT
	J1 = J2+1	CINPUT
	GO TO 25	CINPUT
C		CINPUT
C	TABLE LOAD ... 1ST FUNCTION	CINPUT
C	INPUT CARDS E.4.A-E.4.N	CINPUT
C		CINPUT
	22 READ(5,23) NPI	CINPUT
	23 FORMAT (12I6)	CINPUT
	TAB(J1) = NPI	CINPUT
	J1 = J1+1	CINPUT
	J2 = J1+2*NPI-1	CINPUT
	READ(5,15)(TAB(J),J = J1,J2)	CINPUT
	WRITE(6,24) NPI, (TAB(J) ,J = J1, J2)	CINPUT
	24 FORMAT(7X,'FIRST PART OF FUNCTION - ',I4,' TABULAR POINTS'//	CINPUT
	* 8X,'D',16X,'F(D)' /(F15.6,F15.4))	CINPUT
	J1 = J2+1	CINPUT
C		CINPUT
C	CHECK FOR SECOND FUNCTION	CINPUT
C		CINPUT
	25 IF(D2) 28,11,26	CINPUT
C		CINPUT
C	SECOND FUNCTION ... 5TH ORDER POLYNOMIAL	CINPUT
C	INPUT CARD E.3	CINPUT
C		CINPUT
	26 J2 = J1+5	CINPUT
	READ(5,15)(TAB(J),J = J1,J2)	CINPUT
	WRITE(6,27) (TAB(J),J = J1,J2)	CINPUT
	27 FORMAT(7X,'SECOND PART OF FUNCTION - 5TH DEGREE POLYNOMIAL'//	CINPUT
	* 8X,'B0',13X,'B1',13X,'B2',13X,'B3',13X,'B4',13X,'B5',13X/CINPUT	CINPUT
	* 6F15.6//)	CINPUT
	J1 = J2+1	CINPUT
	GO TO 11	CINPUT
C		CINPUT
C	SECOND FUNCTION ... TABLE LOAD	CINPUT
C	INPUT CARDS E.4.A-E.4.N	CINPUT
C		CINPUT

28	READ(5,23) NPI	CINPUT
	TAB(J1) = NPI	CINPUT
	J1 = J1+1	CINPUT
	J2 = J1+2*NPI-1	CINPUT
	READ(5,15)(TAB(J),J = J1,J2)	CINPUT
	WRITE(6,29) NPI, (TAB(J), J = J1,J2)	CINPUT
29	FORMAT(7X,'SECOND PART OF FUNCTION - ',I4,' TABULAR POINTS'//	CINPUT
	* 8X,'D',16X,'F(D)' /(F15.6,F15.4))	CINPUT
	J1 = J2+1	CINPUT
	GO TO 11	CINPUT
30	MXTB1 = J1-1	CINPUT
	CALL KINPUT	CINPUT
	CALL FINPUT	CINPUT
	CALL HINPUT	CINPUT
	RETURN	CINPUT
	END	CINPUT

C	<pre> SUBROUTINE CMPUTE (K,M,FT) IMPLICIT REAL*8 (A-H,O-Z) COMMON/CONTRL/ TIME,NSEC,NJNT,NPL,NBLT,NBAG,NVEH,NGRND, * NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPC COMMON/CDINT/ UU(4),GH(3,4), * E(3,240), F(5,240),GG(5,240),Y(5,240),U(5,240), * H,HPRINT,HS,TPRINT,TSTART,ICHT,IDBL,IFLAG,IDMY COMMON/COMAIN/ VAR(240),DER(240),DT,HO,HMAX,HMIN,RSTIME, * ISTEP,NSTEPS,NDINT,NEQ,IRSIN,IRSOUT TIME = TSTART + FT CALL DZP (NEQ,VAR,GG,E,FT,M) IF (NPRT(26).EQ.2) CALL OUTPUT(0) CALL PDAUX (VAR,DER,NEQ,K) IF (NPRT(26).EQ.2) CALL OUTPUT(1) RETURN END </pre>	<pre> CMPUTE REV III.2 08/08/84REVIII CMPUTE CMPUTE PAGE CMPUTE CMPUTE 80386 CMPUTE CMPUTE CMPUTE CMPUTE CMPUTE CMPUTE </pre>
---	--	---

	SUBROUTINE CONTACT	CONTACT
C		REV III.2 08/08/84REVIII
C	CONTROLS THE CALLING OF SUBROUTINES REQUIRED TO COMPUTE THOSE	CONTACT
C	EXTERNAL FORCES AND TORQUES ACTING ON THE BODY SEGMENTS.	CONTACT
C		CONTACT
	IMPLICIT REAL*8 (A-H,O-Z)	CONTACT
	COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,	CONTACT
*	NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG	PAGE
	COMMON/JBARTZ/ MNPL(30),MNBLT(8),MNSEG(30),MNBAG(6),	CONTACT
*	MPL(3,5,30),MBLT(3,5,8),MSEG(3,5,30),MBAG(3,10,6),	CONTACT
*	NTPL(5,30),NTBLT(5,8),NTSEG(5,30)	CONTACT
	COMMON/FORCES/PSF(7,70),BSF(4,20),SSF(10,40),BAGSF(3,20),	NCFORC
*	PRJNT(7,30),NPANEL(5),NPSF,NBSF,NSSF,NBSGF	CONTACT
	COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(1250),TAB(4500)	DIMENB
	COMMON/HRNESS/ BAR(15,100).BB(100).BBDOT(100),PLOSS(2,100),	CONTACT
,*	XLONG(20),HTIME(2),IBAR(5,100),NL(2,100),	CONTACT
*	NPTSPB(20),NPTPLY(20),NTHRNS(20),NBLTPH(5)	CONTACT
	COMMON/WINDFR/ WTIME(30),QFU(3.5),QFV(3.5),WF(3,30),IWIND(30),	WINDOP
*	MWSEG(7,30),NFVSEG(6),NFVNT(5),MOWSEG(30,30)	WINDOP
	DATA MAXPSF/70/,MAXBSF/20/,MAXSSF/46/	NCFORC
C		CHGIII
C	MAXSSF SHOULD BE 40 BUT IT IS ALLOWED TO OVERFLOW INTO BAGSF	NCFORC
C		CHGIII
	CALL ELTIME(1,12)	CONTACT
	NPSF = 0	CONTACT
	NBSF = 0	CONTACT
	NSSF = 0	CONTACT
	IF (NPL.LE.0) GO TO 21	CONTACT
C		CONTACT
C	CALL PLELP ROUTINE FOR EACH ALLOWED PLANE-SEGMENT CONTACT.	CONTACT
C		CONTACT
	DO 20 J=1,NPL	CONTACT
	IF(MNPL(J).EQ.0) GO TO 20	CONTACT
	KPL = MNPL(J)	CONTACT
	DO 19 I=1,KPL	CONTACT
	NPSF = NPSF+1	CONTACT
	IF(NPSF.GT.MAXPSF) STOP 57	CHGIII
	M1 = MPL(1,I,J)	CONTACT
	M2 = MPL(2,I,J)	CONTACT
	M3 = MPL(3,I,J)	CONTACT
	NT = NTPL(I,J)	CONTACT
	JT = NTAB(NT)	CONTACT
	TAB(JT) = 0.0	CONTACT
19	CALL PLELP(M2,M3,M1,J,NT)	CONTACT
20	CONTINUE	CONTACT
21	IF(NBLT.LE.0) GO TO 41	CONTACT
C		CONTACT
C	CALL BELTRT ROUTINE FOR EACH ALLOWED BELT-SEGMENT CONTACT.	CONTACT
C		CONTACT
	DO 30 J=1,NBLT	CONTACT
	IF(MNBLT(J).EQ.0) GO TO 30	CONTACT
	KBLT = MNBLT(J)	CONTACT

DO 29 I=1,KBLT	CONTCT
NBSF = NBSF+1	CONTCT
IF(NBSF.GT.MAXBSF) STOP 58	CHGIII
M1 = MBLT(1,I,J)	CONTCT
M2 = MBLT(2,I,J)	CONTCT
M3 = MBLT(3,I,J)	CONTCT
NT = NTBLT(I,J)	CONTCT
JT = NTAB(NT)	CONTCT
TAB(JT) = 0.0	CONTCT
NF = NTAB(NT+5)	CONTCT
IF (NF.NE.0) JT = NTAB(NT+6)	CONTCT
IF (NF.NE.0) TAB(JT) = 0.0	CONTCT
29 CALL BELTRT(M2,M3,M1,J,NT)	CONTCT
30 CONTINUE	CONTCT
C	CONTCT
C CALL SEGSEG ROUTINE FOR EACH ALLOWED SEGMENT-SEGMENT CONTACT.	CONTCT
C	CONTCT
41 DO 50 J=1,NSEG	CONTCT
IF(MNSEG(J).EQ.0) GO TO 50	CONTCT
KSEG = MNSEG(J)	CONTCT
DO 49 I=1,KSEG	CONTCT
NSSF = NSSF+1	CONTCT
IF(NSSF.GT.MAXSSF) STOP 59	CHGIII
M1 = MSEG(1,I,J)	CONTCT
M2 = MSEG(2,I,J)	CONTCT
M3 = MSEG(3,I,J)	CONTCT
NT = NTSEG(I,J)	CONTCT
JT = NTAB(NT)	CONTCT
TAB(JT) = 0.0	CONTCT
49 CALL SEGSEG(J,M1,M2,M3,NT)	CONTCT
50 CONTINUE	CONTCT
C	CONTCT
C CALL AIRBAG ROUTINE FOR ALLOWED BAG-SEGMENT CONTACTS, IF ANY.	CONTCT
C	CONTCT
IF (NBAG.NE.0) CALL AIRBAG	CONTCT
C	CONTCT
C CALL WINDY ROUTINE FOR WIND FORCES ON EACH SEGMENT.	CONTCT
C	CONTCT
DO 60 J=1,NSEG	CONTCT
IF (MWSEG(1,J).EQ.0) GO TO 60	CONTCT
M=MWSEG(1,J)	WINDOP
M1 = MWSEG(2,J)	CONTCT
M2 = MWSEG(3,J)	CONTCT
M3 = MWSEG(4,J)	CONTCT
NT = MWSEG(5,J)	CONTCT
CALL WINDY (M,M1,M2,M3,NT)	WINDOP
60 CONTINUE	CONTCT
C	CONTCT
C CALL WINDY FOR FORCE FUNCE FUNCTION CALCULATIONS.	CONTCT
C	CONTCT
NFORCE = NFVSEG(6)	CONTCT
IF (NFORCE.GT.0) CALL WINDY (0,M1,M2,M3,NT)	WINDOP

C		CONTCT
C	CALL HBELT ROUTINE FOR EACH HARNESS-BELT SYSTEM.	CONTCT
C		CONTCT
	IF (NHRNSS.LE.0) GO TO 80	CONTCT
	J1 = 1	CONTCT
	KNLO = 0	CONTCT
	DO 70 I=1,NHRNSS	CONTCT
	IF (NBLTPH(I).LE.0) GO TO 70	CONTCT
	J2 = J1 + NBLTPH(I) - 1	CONTCT
	CALL HBELT (J1,J2,KNLO,0)	CONTCT
	J1 = J2+1	CONTCT
	70 CONTINUE	CONTCT
C		CONTCT
C	CALL SPDAMP FOR SPRING DAMPER FORCES, IF ANY	CONTCT
C		CONTCT
	80 IF (NSD.NE.0) CALL SPDAMP	CONTCT
	CALL ELTIME (2,12)	CONTCT
	RETURN	CONTCT
	END	CONTCT

C	SUBROUTINE CROSS(A,B,C)		CROSS
C	COMPUTES VECTOR CROSS PRODUCT $C = A \times B$.	REV 03	05/31/73CROSS
C			CROSS
C	ARGUMENTS		CROSS
C	A,B,C: VECTORS OF LENGTH 3 WHERE $C=AXB$.		CROSS
C			CROSS
	IMPLICIT REAL*8 (A-H,O-Z)		CROSS
	DIMENSION A(3),B(3),C(3)		CROSS
	$C(1) = A(2)*B(3) - A(3)*B(2)$		CROSS
	$C(2) = A(3)*B(1) - A(1)*B(3)$		CROSS
	$C(3) = A(1)*B(2) - A(2)*B(1)$		CROSS
	RETURN		CROSS
	END		CROSS

	SUBROUTINE DAUX(I1)		DAUX
C		REV IV	07/24/86SLIP
C	COMPUTES DERIVATIVES FOR INTEGRATOR ROUTINE BY		DAUX
C	(1) SET UP INITIAL VALUES FOR ARRAY OF SYSTEM EQUATIONS.		DAUX
C	(2) MODIFY ARRAYS BY CONSTRAINTS.		DAUX
C	(3) SOLVE SYSTEM OF EQUATION FOR F, TQ, QQ AND V4.		DAUX
C	(4) EVALUATE DERIVATIVES SEGLA AND WMEGD.		DAUX
C			DAUX
	IMPLICIT REAL*8(A-H,O-Z)		DAUX
	COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,		DAUX
*	NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG		FACE
	COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),		DAUX
*	SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)		DAUX
	COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60),		SLIP
*	RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),		DAUX
*	JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)		DAUX
	COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60),		DAUX
*	F(3,30),TQ(3,30),WJ(30),A11(3,3,30)		SLIP
	COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),		DAUX
*	HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12),		DAUX
*	RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),		DAUX
*	KQ1(12),KQ2(12),KQTYPE(12)		DAUX
	COMMON/FLXBLE/ HF(4,12,8),B42(3,3,24),V4(3,8),NFLEX(3,8)		DAUX
	COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),		DAUX
*	UNITL,UNITM,UNITT,GRAVITY(3),TWOPI		TWOPI
	COMMON/RSAVE/ XSG(3,20,3),DPMI(3,3,30),LPMI(30),		ATBIII
*	NSG(9),MSG(20,9),MCG,MCGIN(24,5),KREF(20,9)		TTHKREF
C			DAUX
C	NOTE: DAUX SHARES /TEMPVS/ WITH DAUX11,12,22,31,32 &33.		DAUX
C			DAUX
	LOGICAL*1 FREE,LDMY		80386
	COMMON/TEMPVS/ C(3,3,600),RHS(3,54),IJK(54,54),IJ,NQ2S,		SLIP
*	IDUM(458),FREE(30),LDMY(2),DMY(27859)		80386
	DIMENSION T1(3),T2(3),T3(3)		TGMOD2
	CALL ELTIME(1,9)		DAUX
C			DAUX
C	IF I1#0, U1 AND U2 HAVE BEEN SET UP BY CALLING ROUTINE.		DAUX
C			DAUX
	IF (I1.NE.0) GO TO 8		DAUX
C			DAUX
C	SET UP INITIAL VALUES OF A & B ARRAYS AND U & V VECTORS.		DAUX
C	MODIFY U1 & U2 ARRAYS BY CONTACT AND JOINT FORCES.		DAUX
C			DAUX
	CALL CHAIN(NPRT(36))		JDRIFT
	CALL SETUP1		DAUX
	CALL VEHPOS		DAUX
	CALL CONTCT		DAUX
	CALL VISPR(0,0)		DAUX
	CALL EJOINT(0,0)		DAUX
	CALL SETUP2		DAUX
	IF (NFLX.GT.0) CALL FLXSEG		DAUX
C			DAUX

C	MODIFY U1,U2 AND ADD G TO U1.	DAUX
C		DAUX
	DO 5 J=1,NGRND	DAUX
	IF (ISING(J)) 1,3,5	DAUX
1	DO 2 I=1,3	DAUX
	U1(I,J) = SEGLA(I,J)	DAUX
2	U2(I,J) = WMEGD(I,J)	DAUX
	GO TO 5	DAUX
3	DO 4 I=1,3	DAUX
	U1(I,J) = U1(I,J)*RW(J) + GRAVITY(I)	DAUX
4	U2(I,J) = U2(I,J)*RPHI(I,J)	DAUX
5	CONTINUE	DAUX
C		DAUX
C	SET UP BODY SEGMENT SYMMETRY	DAUX
C	NSYM(J) = 0 3D MOTION	DAUX
C	NSYM(J) = J CENTRAL SEGMENT 2D MOTION, NO LATERAL MOTION	DAUX
C	NSYM(J) = K SEGMENT J SYMMETRIC TO SEGMENT K, ALL MOTION	DAUX
C	IN THE X-Z PLANE, NO LATERAL MOTION	DAUX
C	NSYM(J) = -K SEGMENT J MIRROR SYMMETRIC TO SEGMENT K, EQUAL	DAUX
C	BUT OPPOSITE LATERAL MOTION PERMITTED	DAUX
C		DAUX
	DO 20 J=1,NGRND	DAUX
	IF (NSYM(J).EQ.0) GO TO 20	DAUX
	K = IABS(NSYM(J))	DAUX
	DO 205 L=1,3	TGMOD2
	T1(L) = U2(L,J)	TGMOD2
	T2(L) = U2(L,K)	TGMOD2
	T3(L) = U2(L,J)	TGMOD2
205	CONTINUE	TGMOD2
	IF(LPMI(J).EQ.0.AND.LPMI(K).EQ.0) GO TO 201	TGMOD2
	IF(LPMI(J).NE.0.AND.LPMI(K).EQ.0) GO TO 202	TGMOD2
	IF(LPMI(J).EQ.0.AND.LPMI(K).NE.0) GO TO 203	TGMOD2
	CALL DOT31(DPMI(1,1,J),U2(1,J),T1)	TGMOD2
	CALL DOT31(DPMI(1,1,K),U2(1,K),T2)	TGMOD2
	GO TO 201	TGMOD2
202	CALL DOT31(DPMI(1,1,J),U2(1,J),T1)	TGMOD2
	GO TO 201	TGMOD2
203	CALL DOT31(DPMI(1,1,K),U2(1,K),T2)	TGMOD2
201	CONTINUE	TGMOD2
	IF (NSYM(J).EQ.J) GO TO 19	DAUX
	IF (K.LT.J) GO TO 16	DAUX
	U1(1,J) = 0.5*(U1(1,J) + U1(1,K))	DAUX
	U1(3,J) = 0.5*(U1(3,J) + U1(3,K))	DAUX
	T3(2) = 0.5*(T1(2) + T2(2))	TGMOD2
	GO TO 17	DAUX
16	U1(1,J) = U1(1,K)	DAUX
	U1(3,J) = U1(3,K)	DAUX
	T3(2) = T2(2)	DAUX
17	IF (NSYM(J).GT.0) GO TO 19	DAUX
	IF (K.LT.J) GO TO 18	DAUX
	U1(2,J) = 0.5*(U1(2,J) - U1(2,K))	DAUX
	T3(1) = 0.5*(T1(1) - T2(1))	TGMOD2

	T3(3) = 0.5*(T1(3) - T2(3))	TGMOD2
	GO TO 206	DAUX
18	U1(2,J) = -U1(2,K)	DAUX
	T3(1) = -T2(1)	TGMOD2
	T3(3) = -T2(3)	TGMOD2
	GO TO 206	DAUX
19	U1(2,J) = 0.0	DAUX
	T3(1) = 0.0	TGMOD2
	T3(3) = 0.0	TGMOD2
206	IF(LPMI(J).EQ.0) GO TO 207	TGMOD2
	CALL MAT31(DPMI(1,1,J),T3,U2(1,J))	TGMOD2
	GO TO 20	TGMOD2
207	U2(1,J) = T3(1)	TGMOD2
	U2(2,J) = T3(2)	TGMOD2
	U2(3,J) = T3(3)	TGMOD2
20	CONTINUE	TGMOD2
C		DAUX
C	INITIALIZE IJK ARRAY AND IJ COUNTER TO ZERO.	DAUX
C		DAUX
8	NQ2S = 2*NS + NFLX + NQ	DAUX
	NJ2 = NQ2S + 2*NJNT	DAUX
	IF (NJ2.GT.54) WRITE (6,11) NS,NFLX,NQ,NJNT,NJ2	DAUX
11	FORMAT('ONS=',I6,',NFLX=',I6,',NQ=',I6,',NJNT=',I6,', AND NJ2=',I6)/AFREVS	
	*' THE VALUE OF NJ2 EXCEEDS THE ARRAY SIZES FOR RHS AND IJK IN SUBRDAUX	
	*OUTLINE DAUX. PROGRAM TERMINATED.'	DAUX
	IF (NJ2.GT.54) STOP 34	DAUX
	MJ2 = NJ2	DAUX
	DO 10 I=1,NJ2	DAUX
	DO 10 J=1,NJ2	DAUX
10	IJK(I,J) = 0	DAUX
	IJ = 0	DAUX
C		DAUX
C	ELMINATE SEGLA AND WMEGD FROM SYSTEM OF EQUATIONS.	DAUX
C		DAUX
	IF (NS.GT.0) CALL DAUX55	DAUX
	IF (NJNT.EQ.0) GO TO 12	DAUX
	IF (NFLX.GT.0) CALL DAUX44	DAUX
	CALL DAUX11	DAUX
	CALL DAUX12	DAUX
	CALL DAUX22	DAUX
12	IF (NQ.LE.0) GO TO 15	DAUX
	IF (NJNT.EQ.0) GO TO 13	DAUX
	CALL DAUX31	DAUX
	CALL DAUX32	DAUX
13	CALL DAUX33	DAUX
	DO 14 I=1,NQ	DAUX
14	IF (KQTYPE(I).GE.4) MJ2 = -NJ2	DAUX
15	IF (NPRT(8).EQ.0) GO TO 28	DAUX
21	WRITE (6,22) NPG,(J,J=1,NJ2)	PAGE
	NPG=NPG+1	PAGE
22	FORMAT('1 DAUX PRINT OF IJK MATRIX',97X,'PAGE',I5//6X,40I3)	PAGE
	DO 23 I=1,NJ2	DAUX

23	WRITE (6,24) I,(IJK(I,J),J=1,NJ2)	DAUX
24	FORMAT(I3,3X,40I3)	DAUX
	WRITE (6,29)	DAUX
29	FORMAT('0 DAUX PRINT OF RHS ARRAY'//)	DAUX
	DO 30 K=1,NJ2	DAUX
30	WRITE (6,27) K,(RHS(I,K),I=1,3)	DAUX
	WRITE (6,25) NPG	PAGE
	NPG=NPG+1	PAGE
25	FORMAT('1 DAUX PRINT OF C ARRAY ELEMENTS',91X,'PAGE',15//)	PAGE
	DO 26 K=1,IJ	DAUX
26	WRITE (6,27) K,((C(I,J,K),J=1,3),I=1,3)	DAUX
27	FORMAT(I6,9G14.7)	DAUX
28	IF (NPRT(8).EQ.-2) GO TO 31	DAUX
C		DAUX
C	SOLVE SYSTEM OF EQUATIONS FOR F,TQ,QQ & V4.	DAUX
C		DAUX
	CALL FMSOL (C,RHS,IJK,MJ2,IJ,54,600)	CHGIII
	IF (NPRT(8).EQ. 2) NPRT(8) = -2	DAUX
	IF (NPRT(8).EQ.-2) GO TO 21	DAUX
31	IF (NPRT(8).EQ.-2) NPRT(8) = 0	DAUX
	EPS12 = EPS(12)	JDRIFT
	IF (NJNT.EQ.0) GO TO 49	DAUX
	DO 51 I=1,NJNT	DAUX
	NJ = NQ2S + I	DAUX
	NI = NJ+NJNT	DAUX
	DO 51 K=1,3	DAUX
	IF (DABS(RHS(K,NJ)).LT.EPS12) RHS(K,NJ) = 0.0	DAUX
	IF (DABS(RHS(K,NI)).LT.EPS12) RHS(K,NI) = 0.0	DAUX
	TQ(K,I) = TQ(K,I) - RHS(K,NI)	DAUX
51	F(K,I) = RHS(K,NJ)	DAUX
49	IF (NQ.EQ.0) GO TO 53	DAUX
	DO 52 I=1,NQ	DAUX
	J = 2*NS + NFLX + I	DAUX
	DO 52 K=1,3	DAUX
	IF (KQTYPE(I).I.T.0) RHS(K,J) = 0.0	DAUX
	IF (DABS(RHS(K,J)).LT.EPS12) RHS(K,J) = 0.0	DAUX
52	QQ(K,I) = RHS(K,J)	DAUX
53	IF (NFLX.EQ.0) GO TO 70	DAUX
	DO 54 I=1,NFLX	DAUX
	J = 2*NS + I	DAUX
	DO 54 K=1,3	DAUX
	IF (DABS(RHS(K,J)).LT.EPS12) RHS(K,J) = 0.0	DAUX
54	V4(K,I) = RHS(K,J)	DAUX
C		DAUX
C	BACKUP SOLUTION FOR SEGLA AND WMEGD.	DAUX
C		DAUX
70	DO 71 J=1,NGRND	DAUX
	DO 71 I=1,3	DAUX
	SEGLA(I,J) = U1(I,J)	DAUX
71	WMEGD(J,J) = U2(I,J)	DAUX
	IF (NS.EQ.0) GO TO 79	DAUX
C		DAUX

C	SET UP SEGLA & WMEGD FOR SINGULAR SEGMENTS.	DAUX
C		DAUX
	IS = 0	DAUX
	DO 78 J-1,NGRND	DAUX
	IF (ISING(J).LE.0) GO TO 78	DAUX
	IS = IS+2	DAUX
	DO 77 I-1,3	DAUX
	IF (DABS(RHS(I,IS-1)).LT.EPS12) RHS(I,IS-1) = 0.0	DAUX
	SEGLA(I,J) = SEGLA(I,J) + RHS(I,IS-1)	DAUX
	IF (DABS(RHS(I,IS)).LT.EPS12) RHS(I,IS) = 0.0	DAUX
	77 WMEGD(I,J) = WMEGD(I,J) + RHS(I,IS)	DAUX
	78 CONTINUE	DAUX
	79 IF (NJNT.EQ.0) GO TO 80	DAUX
C		DAUX
C	ELIMINATE F	DAUX
C		DAUX
	DO 75 M-1,NJNT	DAUX
	N = IABS(JNT(M))	DAUX
	IF (N.EQ.0) GO TO 73	DAUX
	DO 72 I-1,3	DAUX
	DO 72 J-1,3	DAUX
	SEGLA(I,N) = SEGLA(I,N) - A11(I,J,M)*RW(N)*F(J,M)	SLIP
	SEGLA(I,M+1) = SEGLA(I,M+1) + A11(I,M)*RW(M+1)*F(J,M)	SLIP
	WMEGD(I,N) = WMEGD(I,N) - B1(I,2*M-1)*RPHI(I,N)*F(J,M)	DAUX
	72 WMEGD(I,M+1) = WMEGD(I,M+1) - B12(J,I,2*M)*RPHI(I,M+1)*F(J,M)	DAUX
C		DAUX
C	ELIMINATE TQ	DAUX
C		DAUX
	73 IF (FREE(M)) GO TO 75	SLIP
	L = NQ2S + NJNT + M	DAUX
	DO 74 I-1,3	DAUX
	DO 74 J-1,3	DAUX
	WMEGD(I,N) = WMEGD(I,N) - A22(I,J,2*M-1)*RPHI(I,N)*RHS(J,L)	DAUX
	74 WMEGD(I,M+1) = WMEGD(I,M+1) + A22(I,J,2*M)*RPHI(I,M+1)*RHS(J,L)	DAUX
	75 CONTINUE	DAUX
	80 IF (NQ.EQ.0) GO TO 83	DAUX
C		DAUX
C	ELIMINATE QQ	DAUX
C		DAUX
	DO 82 K-1,NQ	DAUX
	IF (KQTYPE(K).LT.0) GO TO 82	DAUX
	N = KQ1(K)	DAUX
	M = KQ2(K)	DAUX
	DO 81 I-1,3	DAUX
	DO 81 J-1,3	DAUX
	SEGLA(I,N) = SEGLA(I,N) - A13(I,J,2*K-1)*RW(N)*QQ(J,K)	DAUX
	SEGLA(I,M) = SEGLA(I,M) - A13(I,J,2*K)*RW(M)*QQ(J,K)	DAUX
	WMEGD(I,N) = WMEGD(I,N) - A23(I,J,2*K-1)*RPHI(I,N)*QQ(J,K)	DAUX
	81 WMEGD(I,M) = WMEGD(I,M) - A23(I,J,2*K)*RPHI(I,M)*QQ(J,K)	DAUX
	82 CONTINUE	DAUX
	83 IF (NFLX.EQ.0) GO TO 90	DAUX
C		DAUX

C	ELIMINATE V4 (TORQUES FOR FLEXIBLE SEGMENTS)	DAUX
C		DAUX
	DO 84 N=1,NFLX	DAUX
	N1 = NFLEX(1,N)	DAUX
	N2 = NFLEX(2,N)	DAUX
	N3 = NFLEX(3,N)	DAUX
	DO 84 I=1,3	DAUX
	DO 84 J=1,3	DAUX
	WMEGD(I,N1) = WMEGD(I,N1) - B42(J,I,3*N-2)*RPHI(I,N1)*V4(J,N)	DAUX
	WMEGD(I,N2) = WMEGD(I,N2) - B42(J,I,3*N-1)*RPHI(I,N2)*V4(J,N)	DAUX
84	WMEGD(I,N3) = WMEGD(I,N3) - B42(J,I,3*N)*RPHI(I,N3)*V4(J,N)	DAUX
90	DO 91 J=1,NGRND	DAUX
	DO 91 I=1,3	DAUX
	IF (DABS(WMEGD(I,J)).LE.EPS12) WMEGD(I,J) = 0.0	DAUX
91	IF (DABS(SEGLA(I,J)).LE.EPS12) SEGLA(I,J) = 0.0	DAUX
C		DAUX
C	OPTIONAL OUTPUT OF FUNCTIONS AND DERIVATIVES.	DAUX
C		DAUX
	IF (NPRT(9).NE.0) CALL PRINT(6H DAUX)	DAUX
C		DAUX
	CALL ELTIME(2,9)	DAUX
	RETURN	DAUX
	END	DAUX

```

SUBROUTINE DAUX11                                DAUX11
C                                                     REV IV 07/24/86SLIP
C CALLED BY SUBROUTINE DAUX TO COMPUTE          DAUX11
C                                                     DAUX11
C             -1                               -1      DAUX11
C    (C11) = (B11)(M) (A11) + (B12)(PHI) (A21)  DAUX11
C                                                     DAUX11
C             -1                               -1      DAUX11
C    (R1) = (B11)(M) (U1) + (B12)(PHI) (U2) - (V1) DAUX11
C                                                     DAUX11
C    IMPLICIT REAL*8(A-H,O-Z)                    DAUX11
C    COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND, DAUX11
C *           NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG PAGE
C    COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30), DAUX11
C *           SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30) DAUX11
C    COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60), SLIP
C *           RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90), DAUX11
C *           JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30) DAUX11
C    COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60), DAUX11
C *           F(3,30),TQ(3,30),WJ(30),A11(3,3,30) SLIP
C    COMMON/TEMPVS/ C(3,3,600),RHS(3,54),IJK(54,54),IJ,NQ2S CHGIII
C *           ,DN(3,3),DM(3,3),SN(3,3),SM(3,3),HH(3,3),BN(3) DAUX11
C *           ,DMMY(28044) 80386
C    CALL ELTIME(1,14) DAUX11
C    DO 30 M=1,NJNT DAUX11
C    N = IABS(JNT(M)) DAUX11
C    MQ = NQ2S + M DAUX11
C    IJ = IJ+1 DAUX11
C    IJK(MQ,MQ) = IJ DAUX11
C    IF (N.GT.0) GO TO 13 DAUX11
C                                                     DAUX11
C    IF (N < 1) SET C11(M,M) = I DAUX11
C                                                     DAUX11
C             AND RHS(M) = V1(M) DAUX11
C                                                     DAUX11
C    DO 12 I=1,3 DAUX11
C    DO 11 J=1,3 DAUX11
C 11 C(I,J,IJ) = 0.0 DAUX11
C    C(I,I,IJ) = 1.0 DAUX11
C 12 RHS(I,MQ) = V1(I,M) DAUX11
C    IJK(MQ,MQ) = -IJ DAUX11
C    GO TO 30 DAUX11
C                                                     DAUX11
C    IF (N > 0) SET RHS(M) = U1(N) - U1(M+1) - V1(M) DAUX11
C             + B12(M,N)U2(N) + B12(M,M+1)U2(M+1) DAUX11
C                                                     DAUX11
C             AND C11(M,N) = RW(N) + RW(M+1) DAUX11
C             + B12(M,N)PHI(N)'A21(N,M) DAUX11
C             + B12(M,M+1)PHI(M+1)'A21(M+1,M) DAUX11
C                                                     DAUX11
C 13 DO 15 I=1,3 DAUX11
C    T1 = -V1(I,M) SLIP

```

```

DO 15 J = 1,3
T1 = T1 + B12(I,J,2*M-1)*U2(J,N) + B12(I,J,2*M)*U2(J,M+1)
*      + A11(I,J,M)*(U1(J,N) - U1(J,M+1))
IF (J.LT.I) GO TO 15
T2 = 0.0
IF (J.EQ.I) T2 = RW(N) + RW(M+1)
DO 14 K=1,3
14 T2 = T2 + B12(I,K,2*M-1)*RPHI(K,N )*B12(J,K,2*M-1)
*      + B12(I,K,2*M )*RPHI(K,M+1)*B12(J,K,2*M )
C(I,J,IJ) = T2
C(J,I,IJ) = T2
15 RHS(I,MQ) = T1
IF (ISING(N).NE.0) GO TO 30
L = 0
IF (N.GT.1) L = IABS(JNT(N-1))
IF (L.EQ.0) GO TO 18
C
C IF (N > 1) AND (L = JNT(N-1) > 0)
C
C SET C11(M,N-1) = -RW(N) + B12(M,N)PHI(N)'A21(N,N-1)
C
C AND C11(N-1,M) = C(M,N-1) T
C
C KJNT = NQ2S + N - 1
C IJ = IJ+1
C IJK(MQ,KJNT) = IJ
C IJK(KJNT,MQ) = IJ+1
C DO 17 I=1,3
C DO 17 J=1,3
C C(I,J,IJ) = 0.0
C DO 16 K=1,3
16 C(I,J,IJ) = C(I,J,IJ) + B12(I,K,2*M-1)*RPHI(K,N)*B12(J,K,2*N-2)
*      - A11(I,K,M)*RW(N)*A11(J,K,N-1)
SLIP
17 C(J,I,IJ+1) = C(I,J,IJ)
C IJ = IJ+1
18 IF (M.EQ.NJNT) GO TO 30
C M1 = M+1
C DO 21 L=M1,NJNT
C IF (IABS(JNT(L)).NE.N) GO TO 21
C
C IF (L > M) AND (JNT(L) = N)
C
C SET C11(M,L) = RW(N) + B12(M,N)PHI(N)'A21(N,L)
C
C AND C11(L,M) = C11(M,L) T
C
C KJNT = NQ2S + L
C IJ = IJ+1
C IJK(MQ,KJNT) = IJ
C IJK(KJNT,MQ) = IJ+1

```

DO 20 I=1,3	DAUX11
DO 20 J=1,3	DAUX11
C(I,J,IJ) = 0.0	DAUX11
DO 19 K=1,3	DAUX11
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
* + A11(I,K,M)*RW(N)*A11(J,K,L)	SLIP
20 C(J,I,IJ+1) = C(I,J,IJ)	DAUX11
IJ = IJ+1	DAUX11
21 CONTINUE	DAUX11
30 CONTINUE	DAUX11
CALL ELTIME(2,14)	DAUX11
RETURN	DAUX11
END	DAUX11

	SUBROUTINE DAUX12	DAUX12
C		
	REV IV 07/24/86	SLIP
C	CALLED BY SUBROUTINE DAUX TO COMPUTE	DAUX12
C		DAUX12
C		DAUX12
C	-1	DAUX12
C	(C12) = (B12)(PHI) (A22)	DAUX12
C		DAUX12
C		DAUX12
C	T	DAUX12
C	(C21) = (C12)	DAUX12
C		DAUX12
	IMPLICIT REAL*8(A-H,O-Z)	DAUX12
	COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,	DAUX12
*	NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG	PAGE
	COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60),	SLIP
*	RPHI(3,30).HT(3,3,60),SPRING(5,90),VISC(7,90),	DAUX12
*	JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)	DAUX12
	COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60),	DAUX12
*	F(3,30),TQ(3,30),WJ(30),A11(3,3,30)	SLIP
	LOGICAL*1 FREE,LDMMY	80386
	COMMON/TEMPVS/ C(3,3,600),RHS(3,54),IJK(54,54),IJ,NQ2S	CHGIII
*	, DN(3,3),DM(3,3),SN(3,3),SM(3,3),HH(3,3),BN(3)	DAUX12
*	, IDUM(362),FREE(30),LDMMY(2),DMMY(27859)	80386
	CALL ELTIME(1,15)	DAUX12
	NQSJNT = NQ2S + NJNT	DAUX12
	DO 60 M=1,NJNT	DAUX12
	N = IABS(JNT(M))	DAUX12
	IF (N.EQ.0) GO TO 60	DAUX12
	MQ = NQ2S + M	DAUX12
	IF (FREE(M)) GO TO 37	SLIP
	MJNT = NQSJNT + M	DAUX12
	IJ = IJ+1	DAUX12
	IJK(MQ,MJNT) = IJ	DAUX12
	IJK(MJNT,MQ) = IJ+1	DAUX12
	DO 36 I=1,3	DAUX12
	DO 36 J=1,3	DAUX12
	SN(I,J) = 0.0	DAUX12
	SM(I,J) = 0.0	DAUX12
	DO 35 K=1,3	DAUX12
	SN(I,J) = SN(I,J) + B12(I,K,2*M-1) * RPHI(K,N) * A22(K,J,2*M-1)	DAUX12
35	SM(I,J) = SM(I,J) + B12(I,K,2*M) * RPHI(K,M+1) * A22(K,J,2*M)	DAUX12
	C(I,J,IJ) = SN(I,J) - SM(I,J)	DAUX12
36	C(J,I,IJ+1) = C(I,J,IJ)	DAUX12
	IJ = IJ+1	DAUX12
37	IF (ISING(N).NE.0) GO TO 50	DAUX12
	IF (N.EQ.1) GO TO 43	DAUX12
	IF (FREE(N-1)) GO TO 43	SLIP
	MJNT = NQSJNT + N-1	DAUX12
	IJ = IJ+1	DAUX12
	IJK(MQ,MJNT) = IJ	DAUX12
	IJK(MJNT,MQ) = IJ+1	DAUX12
	DO 42 I=1,3	DAUX12
	DO 42 J=1,3	DAUX12

SN(I,J) = 0.0	DAUX12
DO 41 K=1,3	DAUX12
41 SN(I,J) = SN(I,J) + B12(I,K,2*M-1) * RPHI(K,N) * A22(K,J,2*N-2)	DAUX12
C(I,J,IJ) = -SN(I,J)	DAUX12
42 C(J,I,IJ+1) = -SN(I,J)	DAUX12
IJ = IJ+1	DAUX12
43 DO 49 L=N,NJNT	DAUX12
IF (L.EQ.M) GO TO 49	DAUX12
IF (IABS(JNT(L)).NE.N) GO TO 49	DAUX12
IF (FREE(L)) GO TO 49	SLIP
MJNT = NQSJNT + L	DAUX12
IJ = IJ+1	DAUX12
IJK(MQ,MJNT) = IJ	DAUX12
IJK(MJNT,MQ) = IJ+1	DAUX12
DO 48 I=1,3	DAUX12
DO 48 J=1,3	DAUX12
SN(I,J) = 0.0	DAUX12
DO 47 K=1,3	DAUX12
47 SN(I,J) = SN(I,J) + B12(I,K,2*M-1) * RPHI(K,N) * A22(K,J,2*L-1)	DAUX12
C(I,J,IJ) = SN(I,J)	DAUX12
48 C(J,I,IJ+1) = SN(I,J)	DAUX12
IJ = IJ +1	DAUX12
49 CONTINUE	DAUX12
50 IF (M.EQ.NJNT) GO TO 60	DAUX12
IF (ISING(M+1).NE.0) GO TO 60	DAUX12
M1 = M+1	DAUX12
DO 59 L=M1,NJNT	DAUX12
IF (IABS(JNT(L)).NE.M1) GO TO 59	DAUX12
IF (FREE(L)) GO TO 59	SLIP
MJNT = NQSJNT + L	DAUX12
IJ = IJ+1	DAUX12
IJK(MQ,MJNT) = IJ	DAUX12
IJK(MJNT,MQ) = IJ+1	DAUX12
DO 58 I=1,3	DAUX12
DO 58 J=1,3	DAUX12
SM(I,J) = 0.0	DAUX12
DO 57 K=1,3	DAUX12
57 SM(I,J) = SM(I,J) + B12(I,K,2*M) * RPHI(K,M+1) * A22(K,J,2*L-1)	DAUX12
C(I,J,IJ) = SM(I,J)	DAUX12
58 C(J,I,IJ+1) = SM(I,J)	DAUX12
IJ = IJ +1	DAUX12
59 CONTINUE	DAUX12
60 CONTINUE	DAUX12
CALL ELTIME(2,15)	DAUX12
RETURN	DAUX12
END	DAUX12

	SUBROUTINE DAUX22		DAUX22
C		REV IV	07/24/86SLIP
C	CALLED BY SUBROUTINE DAUX TO COMPUTE		DAUX22
C			DAUX22
C			DAUX22
C		-1	DAUX22
C	(C22) - (B22)(PHI) (A22) - (B24)		DAUX22
C			DAUX22
C		-1	DAUX22
C	(R2) - (B22)(PHI) (U2) - (V2)		DAUX22
C			DAUX22
	IMPLICIT REAL*8(A-H,O-Z)		DAUX22
	COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,		DAUX22
*	NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG		PAGE
	COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),		DAUX22
*	SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)		DAUX22
	COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60),		SLIP
*	RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),		DAUX22
*	JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)		DAUX22
	COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60),		DAUX22
*	F(3,30),TQ(3,30),WJ(30),A11(3,3,30)		SLIP
	COMMON/CEULER/ IEULER(30),HIR(3,3,90),ANG(3,30),ANGD(3,30),		JDRIFT
*	FE(3,30),TQE(3,30),CONST(5,30)		JDRIFT
	LOGICAL*1 FREE,LDDMY		80386
	COMMON/TEMPVS/ C(3,3,600),RHS(3,54),IJK(54,54),IJ,NQ2S		CHGIII
*	,DN(3,3),DM(3,3),SN(3,3),SM(3,3),HH(3,3),BN(3)		DAUX22
*	,IDUM(362),FREE(30),LDDMY(2),DDMY(27859)		80386
	LOGICAL TEST		DAUX22
	CALL ELTIME(1,16)		DAUX22
	NQSJNT = NQ2S + NJNT		DAUX22
	DO 90 M=1,NJNT		DAUX22
	MJNT = NQSJNT + M		DAUX22
	DO 60 I=1,3		DAUX22
60	RHS(I,MJNT) = V2(I,M)		DAUX22
	N = IABS(JNT(M))		DAUX22
	IF (N.EQ.0) GO TO 90		DAUX22
	IF (FREE(M)) GO TO 90		SLIP
	IJ = IJ+1		DAUX22
	IJK(MJNT,MJNT) = IJ		DAUX22
	DO 61 J=1,3		DAUX22
	DO 61 I=1,3		DAUX22
61	HH(I,J) = 0.0		DAUX22
	LGO = IPIN(M)+8		SLIP
	TEST = .FALSE.		DAUX22
	GO TO (64,64,64,62,64,64,64,64,63,64,64,64,64,63,63),LGO		SLIP
62	IF (IEULER(M).GE.7) GO TO 64		DAUX22
	TEST = IEULER(M).LT.4		DAUX22
63	AN = 0.0		DAUX22
	DO 51 J=1,3		DAUX22
51	AN = AN + HB(J,2*M-1)**2 * RPHI(J N)		DAUX22
*	+ HB(J,2*M)**2 * RPHI(J,M+1)		DAUX22
	IF (TEST) GO TO 64		DAUX22

CALL DOT31 (D(1,1,N),HB(1,2*M-1),BN)	DAUX22
DO 53 J=-1,3	DAUX22
DO 53 I=-1,3	DAUX22
53 HH(I,J) = AN*BN(I)*BN(J)	DAUX22
64 DO 67 I=-1,3	DAUX22
RHS(I,MJNT) = -V2(I,M)	DAUX22
DO 66 J=-1,3	DAUX22
RHS(I,MJNT) = RHS(I,MJNT) + A22(J,I,2*M-1)*U2(J,N)	DAUX22
* - A22(J,I,2*M)*U2(J,M+1)	DAUX22
SN(I,J) = 0.0	DAUX22
IF (TEST) GO TO 66	DAUX22
DO 65 K=-1,3	DAUX22
65 SN(I,J) = SN(I,J) + A22(K,I,2*M-1) * RPHI(K,N) * A22(K,J,2*M-1)	DAUX22
* + A22(K,I,2*M) * RPHI(K,M+1) * A22(K,J,2*M)	DAUX22
66 C(I,J,IJ) = SN(I,J) + HH(I,J)	DAUX22
67 IF (TEST) C(I,I,IJ) = AN	DAUX22
IF (ISING(N).NE.0) GO TO 90	DAUX22
IF (N.EQ.1) GO TO 80	DAUX22
IF (FREE(N-1)) GO TO 80	DAUX22
N1JNT = NQSJNT + N -1	SLIP
IJ = IJ+1	DAUX22
IJK(MJNT,N1JNT) = IJ	DAUX22
IJK(N1JNT,MJNT) = IJ+1	DAUX22
DO 77 I=-1,3	DAUX22
DO 77 J=-1,3	DAUX22
SN(I,J) = 0.0	DAUX22
DO 76 K=-1,3	DAUX22
76 SN(I,J) = SN(I,J) + A22(K,I,2*M-1) * RPHI(K,N) * A22(K,J,2*N-2)	DAUX22
C(I,J,IJ) = -SN(I,J)	DAUX22
77 C(J,I,IJ+1) = -SN(I,J)	DAUX22
IJ = IJ+1	DAUX22
80 IF (M.EQ.NJNT) GO TO 90	DAUX22
M1 = M+1	DAUX22
DO 88 L=M1,NJNT	DAUX22
IF (IABS(JNT(L)).NE.N) GO TO 88	DAUX22
IF (FREE(L)) GO TO 88	DAUX22
LJNT = NQSJNT + L	SLIP
IJ = IJ+1	DAUX22
IJK(MJNT,LJNT) = IJ	DAUX22
IJK(LJNT,MJNT) = IJ+1	DAUX22
DO 87 I=-1,3	DAUX22
DO 87 J=-1,3	DAUX22
SN(I,J) = 0.0	DAUX22
DO 86 K=-1,3	DAUX22
86 SN(I,J) = SN(I,J) + A22(K,I,2*M-1) * RPHI(K,N) * A22(K,J,2*L-1)	DAUX22
C(I,J,IJ) = SN(I,J)	DAUX22
87 C(J,I,IJ+1) = SN(I,J)	DAUX22
IJ = IJ+1	DAUX22
88 CONTINUE	DAUX22
90 CONTINUE	DAUX22
CALL ELTIME(2,16)	DAUX22
RETURN	DAUX22

END

DAUX22

	SUM = SUM + B12(I,K,2*K1-2)*RPHI(K,K1)*A23(K,J,2*N-1)	DAUX31
	* - A11(I,K,K1-1)*RW(K1)*A13(K,J,2*N-1)	SLIP
11	TUM = TUM + B32(I,K,2*N-1)*RPHI(K,K1)*B12(J,K,2*K1-2)	DAUX31
	* - B31(I,K,2*N-1)*RW(K1)*A11(K,J,K1-1)	SLIP
	C(I,J,IJ) = SUM	DAUX31
12	C(I,J,IJ+1) = TUM	DAUX31
	IJ = IJ+1	DAUX31
13	IF (K2.LE.1) GO TO 16	DAUX31
	IF (IABS(JNT(K2-1)).EQ.0) GO TO 16	DAUX31
	IF (ISING(K2).NE.0) GO TO 16	DAUX31
C		DAUX31
C		DAUX31
C	C13(K2-1,N) = B11(K2-1,K2)M ⁻¹ (K2)A13(K2,N)	DAUX31
C		DAUX31
C	+ B12(K2-1,K2)PHI ⁻¹ (K2)A23(K2,N)	DAUX31
C		DAUX31
C		DAUX31
C	C31(N,K2-1) = B31(N,K2)M ⁻¹ (K2)A11(K2,K2-1)	DAUX31
C		DAUX31
C	+ B32(N,K2)PHI ⁻¹ (K2)A21(K2,K2-1)	DAUX31
C		DAUX31
	MQ = NQ2S + K2 - 1	DAUX31
	IJ = IJ+1	DAUX31
	IJK(MQ,NNS) = IJ	DAUX31
	IJK(NNS,MQ) = IJ+1	DAUX31
	DO 15 I=1,3	DAUX31
	DO 15 J=1,3	DAUX31
	SUM = 0.0	SLIP
	TUM = 0.0	SLIP
	DO 14 K=1,3	DAUX31
	SUM = SUM + B12(I,K,2*K2-2)*RPHI(K,K2)*A23(K,J,2*N)	DAUX31
	* - A11(I,K,K2-1)*RW(K2)*A13(K,J,2*N)	SLIP
14	TUM = TUM + B32(I,K,2*N)*RPHI(K,K2)*B12(J,K,2*K2-2)	DAUX31
	* - B31(I,K,2*N)*RW(K2)*A11(K,J,K2-1)	SLIP
	C(I,J,IJ) = SUM	DAUX31
15	C(I,J,IJ+1) = TUM	DAUX31
	IJ = IJ+1	DAUX31
16	IF (NJNT.LE.0) GO TO 30	DAUX31
	DO 26 L=1,NJNT	DAUX31
	IF (IABS(JNT(L)).NE.K1) GO TO 21	DAUX31
	IF (ISING(K1).NE.0) GO TO 21	DAUX31
C		DAUX31
C	FOR ANY L SUCH THAT JNT(L) = K1	DAUX31
C		DAUX31
C		DAUX31
C	C13(L,N) = B11(L,K1)M ⁻¹ (K1)A13(K1,N)	DAUX31
C		DAUX31
C	+ B12(L,K1)PHI ⁻¹ (K1)A23(K1,N)	DAUX31
C		DAUX31
C		DAUX31
C	C31(N,L) = B31(N,K1)M ⁻¹ (K1)A11(K1,L)	DAUX31
C		DAUX31
C		DAUX31

C	+ B32(N,K1)PHI (K1)A21(K1,L)	DAUX31
C		DAUX31
	MQ = NQ2S + L	DAUX31
	IF (IJK(MQ,NNS).NE.0) GO TO 18	DAUX31
	IJ = IJ+1	DAUX31
	IJK(MQ,NNS) = IJ	DAUX31
	IJK(NNS,MQ) = IJ+1	DAUX31
	DO 17 J=-1,3	DAUX31
	DO 17 I=-1,3	DAUX31
	C(I,J,IJ) = 0.0	DAUX31
17	C(I,J,IJ+1) = 0.0	DAUX31
	IJ = IJ+1	DAUX31
18	JJ = IJK(MQ,NNS)	DAUX31
	DO 20 I=-1,3	DAUX31
	DO 20 J=-1,3	DAUX31
	SUM = C(I,J,JJ)	DAUX31
	TUM = C(I,J,JJ+1)	SLIP
	DO 19 K=-1,3	SLIP
	SUM = SUM + B12(I,K,2*L-1)*RPHI(K,K1)*A23(K,J,2*N-1)	DAUX31
	* +A11(I,K,L)*RW(K1)*A13(K,J,2*N-1)	DAUX31
19	TUM = TUM + B32(I,K,2*N-1)*RPHI(K,K1)*B12(J,K,2*L-1)	SLIP
	* +B31(I,K,2*N-1)*RW(K1)*A11(J,K,L)	DAUX31
	C(I,J,JJ) = SUM	SLIP
20	C(I,J,JJ+1) = TUM	DAUX31
21	IF (IABS(JNT(L)).NE.K2) GO TO 26	DAUX31
	IF (ISING(K2).NE.0) GO TO 26	DAUX31
C		DAUX31
C	FOR ANY L SUCH THAT JNT(L) = K2	DAUX31
C		DAUX31
C		DAUX31
C		DAUX31
C	C13(L,N) = B11(L,K2)M ⁻¹ (K2)A13(K2,N)	DAUX31
C		DAUX31
C	+ B12(L,K2)PHI ⁻¹ (K2)A23(K2,N)	DAUX31
C		DAUX31
C		DAUX31
C	C31(N,L) = B31(N,K2)M ⁻¹ (K2)A11(K2,L)	DAUX31
C		DAUX31
C	+ B32(N,K2)PHI ⁻¹ (K2)A21(K2,L)	DAUX31
C		DAUX31
	MQ = NQ2S + L	DAUX31
	IF (IJK(MQ,NNS).NE.0) GO TO 23	DAUX31
	IJ = IJ+1	DAUX31
	IJK(MQ,NNS) = IJ	DAUX31
	IJK(NNS,MQ) = IJ+1	DAUX31
	DO 22 J=-1,3	DAUX31
	DO 22 I=-1,3	DAUX31
	C(I,J,IJ) = 0.0	DAUX31
22	C(I,J,IJ+1) = 0.0	DAUX31
	IJ = IJ+1	DAUX31
23	JJ = IJK(MQ,NNS)	DAUX31
	DO 25 I=-1,3	DAUX31
	DO 25 J=-1,3	DAUX31

SUM = C(I,J,JJ)	SLIP
TUM = C(I,J,JJ+1)	SLIP
DO 24 K=1,3	DAUX31
SUM = SUM + B12(I,K,2*L-1)*RPHI(K,K2)*A23(K,J,2*N)	DAUX31
* + A11(I,K,L)*RW(K2)*A13(K,J,2*N)	SLIP
24 TUM = TUM + B32(I,K,2*N)*RPHI(K,K2)*B12(J,K,2*L-1)	DAUX31
* + B31(I,K,2*N)*RW(K2)*A11(J,K,L)	SLIP
C(I,J,JJ) = SUM	DAUX31
25 C(I,J,JJ+1) = TUM	DAUX31
26 CONTINUE	DAUX31
30 CONTINUE	DAUX31
CALL ELTIME(2,17)	DAUX31
RETURN	DAUX31
END	DAUX31

	SUBROUTINE DAUX32		DAUX32
C		REV IV	07/24/86SLIP
C	CALLED BY SUBROUTINE DAUX TO COMPUTE		DAUX32
C			DAUX32
C		-1	DAUX32
C	(C23) - (B22)(PHI) (A23)		DAUX32
C			DAUX32
C		-1	DAUX32
C	(C32) - (B32)(PHI) (A22)		DAUX32
C			DAUX32
	IMPLICIT REAL*8 (A-H,O-Z)		DAUX32
	COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,		DAUX32
*	NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG		PAGE
	COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60),		SLIP
*	RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),		DAUX32
*	JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)		DAUX32
	COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60),		DAUX32
*	F(3,30),TQ(3,30),WJ(30),A11(3,3,30)		SLIP
	COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),		DAUX32
*	HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12),		DAUX32
*	RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),		DAUX32
*	KQ1(12),KQ2(12),KQTYPE(12)		DAUX32
	LOGICAL*1 FREE,LDDMY		80386
	COMMON/TEMPVS/ C(3,3,600),RHS(3,54),IJK(54,54),IJ,NQ2S		CHGIII
*	,DN(3,3),DM(3,3),BN(3),IDUM(416),FREE(30)		SLIP
*	,LDDMY(2),DDMY(27859)		80386
	CALL ELTIME(i,18)		DAUX32
	NQSJNT = NQ2S + NJNT		DAUX32
	DO 60 N=1,NQ		DAUX32
	IF (KQTYPE(N).LT.0) GO TO 60		DAUX32
	K1 = KQ1(N)		DAUX32
	K2 = KQ2(N)		DAUX32
	NNS = NQ2S - NQ + N		DAUX32
	IF (K1.LE.1) GO TO 43		DAUX32
	IF (IABS(JNT(K1-1)).EQ.0) GO TO 43		DAUX32
	IF (FREE(K1-1)) GO TO 43		SLIP
	IF (ISING(K1).NE.0) GO TO 43		DAUX32
C			DAUX32
C		-1	DAUX32
C	C23(K1-1,N) - B22(K1-1,K1)PHI (K1)A23(K1,N)		DAUX32
C			DAUX32
C		-1	DAUX32
C	C32(N,K1-1) - B32(N,K1)PHI (K1)A22(K1,K1-1)		DAUX32
C			DAUX32
	KJNT = NQSJNT + K1 - 1		DAUX32
	IJ = IJ+1		DAUX32
	IJK(KJNT,NNS) = IJ		DAUX32
	IJK(NNS,KJNT) = IJ+1		DAUX32
	DO 42 I=1,3		DAUX32
	DO 42 J=1,3		DAUX32
	SUM = 0.0		DAUX32
	TUM = 0.0		DAUX32

	DO 41 K=1,3	DAUX32
	SUM = SUM + A22(K,I,2*K1-2) * RPHI(K,K1) * A23(K,J,2*N-1)	DAUX32
	41 TUM = TUM + B32(I,K,2*N-1) * RPHI(K,K1) * A22(K,J,2*K1-2)	DAUX32
	C(I,J,IJ) = -SUM	DAUX32
	42 C(I,J,IJ+1) = -TUM	DAUX32
	IJ = IJ+1	DAUX32
	43 IF (K2.LE.1) GO TO 46	DAUX32
	IF (IABS(JNT(K2-1)).EQ.0) GO TO 46	DAUX32
	IF (FREE(K2-1)) GO TO 46	SLIP
	IF (ISING(K2).NE.0) GO TO 46	DAUX32
C		DAUX32
C		DAUX32
C	C23(K2-1,N) = B22(K2-1,K2)PHI (K2)A23(K2,N)	DAUX32
C		DAUX32
C		DAUX32
C	C32(N,K2-1) = B32(N,K2)PHI (K2)A22(K2,K2-1)	DAUX32
C		DAUX32
	KJNT = NQSJNT + K2 - 1	DAUX32
	IJ = IJ+1	DAUX32
	IJK(KJNT,NNS) = IJ	DAUX32
	IJK(NNS,KJNT) = IJ+1	DAUX32
	DO 45 I=1,3	DAUX32
	DO 45 J=1,3	DAUX32
	SUM = 0.0	DAUX32
	TUM = 0.0	DAUX32
	DO 44 K=1,3	DAUX32
	SUM = SUM + A22(K,I,2*K2-2) * RPHI(K,K2) * A23(K,J,2*N)	DAUX32
	44 TUM = TUM + B32(I,K,2*N) * RPHI(K,K2) * A22(K,J,2*K2-2)	DAUX32
	C(I,J,IJ) = -SUM	DAUX32
	45 C(I,J,IJ+1) = -TUM	DAUX32
	IJ = IJ+1	DAUX32
	46 IF (NJNT.LE.0) GO TO 60	DAUX32
	DO 56 L=1,NJNT	DAUX32
	IF (FREE(L)) GO TO 56	SLIP
	IF (IABS(JNT(L)).NE.K1) GO TO 51	DAUX32
	IF (ISING(K1).NE.0) GO TO 51	DAUX32
C		DAUX32
C		DAUX32
C	FOR ANY L SUCH THAT JNT(L) = K1	DAUX32
C		DAUX32
C		DAUX32
C	C23(L,N) = B22(L,K1)PHI (K1)A23(K1,N)	DAUX32
C		DAUX32
C		DAUX32
C	C32(N,L) = B32(N,K1)PHI (K1)A22(K1,L)	DAUX32
C		DAUX32
	KJNT = NQSJNT + L	DAUX32
	IF (IJK(KJNT,NNS).NE.0) GO TO 48	DAUX32
	IJ = IJ+1	DAUX32
	IJK(KJNT,NNS) = IJ	DAUX32
	IJK(NNS,KJNT) = IJ+1	DAUX32
	DO 47 J=1,3	DAUX32
	DO 47 I=1,3	DAUX32

	C(I,J,IJ) = 0.0	DAUX32
47	C(I,J,IJ+1) = 0.0	DAUX32
	IJ = IJ+1	DAUX32
48	JJ = IJK(KJNT,NNS)	DAUX32
	DO 50 I=1,3	DAUX32
	DO 50 J=1,3	DAUX32
	SUM = C(I,J,JJ)	DAUX32
	TUM = C(I,J,JJ+1)	DAUX32
	DO 49 K=1,3	DAUX32
	SUM = SUM + A22(K,I,2*L-1) * RPHI(K,K1) * A23(K,J,2*N-1)	DAUX32
49	TUM = TUM + B32(I,K,2*N-1) * RPHI(K,K1) * A22(K,J,2*L-1)	DAUX32
	C(I,J,JJ) = SUM	DAUX32
50	C(I,J,JJ+1) = TUM	DAUX32
51	IF (IABS(JNT(L)).NE.K2) GO TO 56	DAUX32
	IF (ISING(K2).NE.0) GO TO 56	DAUX32
C		DAUX32
C	FOR ANY L SUCH THAT JNT(L) = K2	DAUX32
C		DAUX32
C	-1	DAUX32
C	C23(L,N) = B22(L,K2)PHI (K2)A23(K2,N)	DAUX32
C		DAUX32
C	-1	DAUX32
C	C32(N,L) = B32(N,K2)PHI (K2)A22(K2,L)	DAUX32
C		DAUX32
	KJNT = NQSJNT + L	DAUX32
	IF (IJK(KJNT,NNS).NE.0) GO TO 53	DAUX32
	IJ = IJ+1	DAUX32
	IJK(KJNT,NNS) = IJ	DAUX32
	IJK(NNS,KJNT) = IJ+1	DAUX32
	DO 52 J=1,3	DAUX32
	DO 52 I=1,3	DAUX32
	C(I,J,IJ) = 0.0	DAUX32
52	C(I,J,IJ+1) = 0.0	DAUX32
	IJ = IJ+1	DAUX32
53	JJ = IJK(KJNT,NNS)	DAUX32
	DO 55 I=1,3	DAUX32
	DO 55 J=1,3	DAUX32
	SUM = C(I,J,JJ)	DAUX32
	TUM = C(I,J,JJ+1)	DAUX32
	DO 54 K=1,3	DAUX32
	SUM = SUM + A22(K,I,2*L-1) * RPHI(K,K2) * A23(K,J,2*N)	DAUX32
54	TUM = TUM + B32(I,K,2*N) * RPHI(K,K2) * A22(K,J,2*L-1)	DAUX32
	C(I,J,JJ) = SUM	DAUX32
55	C(I,J,JJ+1) = TUM	DAUX32
56	CONTINUE	DAUX32
60	CONTINUE	DAUX32
	CALL ELTIME(2,18)	DAUX32
	RETURN	DAUX32
	END	DAUX32

```

SUBROUTINE DAUX33                                DAUX33
C                                                REV IV    07/24/86SLIP
C CALLED BY SUBROUTINE DAUX TO COMPUTE          DAUX33
C                                                DAUX33
C              -1                               -1   DAUX33
C (C33) = (B31)(M) (A13) + (B32)(PHI) (A23) - (B35) DAUX33
C                                                DAUX33
C              -1                               -1   DAUX33
C (R3) = (B31)(M) (U1) + (B32)(PHI) (U2) - (V3)  DAUX33
C                                                DAUX33
IMPLICIT REAL*8 (A-H,O-Z)                        DAUX33
COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND, DAUX33
* NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30), DAUX33
* SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)      DAUX33
COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60), SLIP
* RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),    DAUX33
* JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)  DAUX33
COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60), DAUX33
* F(3,30),TQ(3,30),WJ(30),A11(3,3,30)             SLIP
COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24), DAUX33
* HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12), DAUX33
* RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),          DAUX33
* KQ1(12),KQ2(12),KQTYPE(12)                     DAUX33
COMMON/TEMPVS/ C(3,3,600),RHS(3,54),IJK(54,54),IJ,NQ2S,DMMY(28092) 80386
CALL ELTIME(1,19)                                DAUX33
DO 90 N=1,NQ                                     DAUX33
IF (KQTYPE(N).LT.0) GO TO 90                     DAUX33
K1 = KQ1(N)                                       DAUX33
K2 = KQ2(N)                                       DAUX33
NNS = NQ2S - NQ + N                              DAUX33
C                                                DAUX33
C              -1                               -1   DAUX33
C RHS(N) = B31(N,K1)M (K1)U1(K1) + B32(N,K1)PHI (K1)U2(K1) DAUX33
C              -1                               -1   DAUX33
C      + B31(N,K2)M (K2)U1(K2) + B32(N,K2)PHI (K2)U2(K2) DAUX33
C                                                DAUX33
C      - V3(N)                                     DAUX33
C                                                DAUX33
DO 63 I=1,3                                       DAUX33
SUM = 0.0                                         DAUX33
DO 62 K=1,3                                       DAUX33
62 SUM = SUM + B31(I,K,2*N-1)*U1(K,K1) + B32(I,K,2*N-1)*U2(K,K1) DAUX33
*      + B31(I,K,2*N )*U1(K,K2) + B32(I,K,2*N )*U2(K,K2) DAUX33
63 RHS(I,NNS) = SUM - V3(I,N)                    DAUX33
C                                                DAUX33
C              -1                               -1   DAUX33
C C33(N,N) = B31(N,K1)M (K1)A13(K1,N) + B32(N,K1)PHI (K1)A23(K1,N) DAUX33
C              -1                               -1   DAUX33
C      + B31(N,K2)M (K2)A13(K2,N) + B32(N,K2)PHI (K2)A23(K2,N) DAUX33
C                                                DAUX33
C      - B35(N,N)                                 DAUX33

```

C	IJ = IJ+1	DAUX33
	IJK(NNS,NNS) = IJ	DAUX33
	IF (KQTYPE(N).EQ.2) GO TO 51	DAUX33
	IF (KQTYPE(N).EQ.4) GO TO 51	DAUX33
	DO 65 I=1,3	DAUX33
	DO 65 J=1,3	DAUX33
	SUM = -HHT(I,J,N)	DAUX33
	IF (I.EQ.J) SUM = 1.0+SUM	DAUX33
	DO 64 K=1,3	DAUX33
64	SUM = SUM + B31(I,K,2*N-1)* RW(K1)*A13(K,J,2*N-1)	DAUX33
	* + B31(I,K,2*N)* RW(K2)*A13(K,J,2*N)	DAUX33
	* + B32(I,K,2*N-1)*RPHI(K,K1)*A23(K,J,2*N-1)	DAUX33
	* + B32(I,K,2*N)*RPHI(K,K2)*A23(K,J,2*N)	DAUX33
65	C(I,J,IJ) = SUM	DAUX33
	GO TO 59	DAUX33
C		DAUX33
C	FOR KQTYPE = 2 OR 4. SET C33(N,N) = B*I	DAUX33
C	WHERE B = SUM OF DIAGONAL ELEMENTS OF	DAUX33
C	-1 -1	DAUX33
C	(B31)(M) (A13) + (B32)(PHI) (A23)	DAUX33
C		DAUX33
51	SUM = 0.0	DAUX33
	DO 55 I=1,3	DAUX33
	DO 55 K=1,3	DAUX33
55	SUM = SUM + B31(I,K,2*N-1)* RW(K1)*A13(K,I,2*N-1)	DAUX33
	* + B31(I,K,2*N)* RW(K2)*A13(K,I,2*N)	DAUX33
	* + B32(I,K,2*N-1)*RPHI(K,K1)*A23(K,I,2*N-1)	DAUX33
	* + B32(I,K,2*N)*RPHI(K,K2)*A23(K,I,2*N)	DAUX33
	DO 57 I=1,3	DAUX33
	DO 56 J=1,3	DAUX33
56	C(I,J,IJ) = 0.0	DAUX33
57	C(I,I,IJ) = SUM	DAUX33
59	IF (N.EQ.NQ) GO TO 90	DAUX33
	N1 = N+1	DAUX33
	DO 85 M=N1,NQ	DAUX33
	IF (KQTYPE(M).LT.0) GO TO 85	DAUX33
	MNS = NQ2S - NQ + M	DAUX33
	IF (ISING(K1).NE.0) GO TO 75	DAUX33
	IF (K1.NE.KQ1(M)) GO TO 70	DAUX33
	IF (IJK(MNS,NNS).NE.0) GO TO 67	DAUX33
C		DAUX33
C	FOR ANY M>N SUCH THAT K1(N) = K1(M)	DAUX33
C		DAUX33
C	-1	DAUX33
C	C33(N,M) = C(N,M) + B31(N,K1) M (K1)A13(K1,M)	DAUX33
C	-1	DAUX33
C	+ B32(N,K1)PHI (K1)A23(K1,M)	DAUX33
C		DAUX33
C	-1	DAUX33
C	C33(M,N) = C(M,N) + B31(M,K1) M (K1)A13(K1,N)	DAUX33
C	-1	DAUX33

C		+ B32(M,K1)PHI (K1)A23(K1,N)	DAUX33
C			DAUX33
	IJ - IJ+1		DAUX33
	IJK(MNS,NNS) - IJ		DAUX33
	IJK(NNS,MNS) - IJ+1		DAUX33
	DO 66 J-1,3		DAUX33
	DO 66 I-1,3		DAUX33
	C(I,J,IJ) = 0.0		DAUX33
66	C(I,J,IJ+1) = 0.0		DAUX33
	IJ - IJ+1		DAUX33
67	JJ - IJK(MNS,NNS)		DAUX33
	DO 69 I-1,3		DAUX33
	DO 69 J-1,3		DAUX33
	SUM = C(I,J,JJ)		DAUX33
	TUM = C(I,J,JJ+1)		DAUX33
	DO 68 K-1,3		DAUX33
	SUM = SUM + B31(I,K,2*N-1)* RW(K1)*A13(K,J,2*M-1)		DAUX33
	* + B32(I,K,2*N-1)*RPHI(K,K1)*A23(K,J,2*M-1)		DAUX33
68	TUM = TUM + B31(I,K,2*M-1)* RW(K1)*A13(K,J,2*N-1)		DAUX33
	* + B32(I,K,2*M-1)*RPHI(K,K1)*A23(K,J,2*N-1)		DAUX33
	C(I,J,JJ) = SUM		DAUX33
69	C(I,J,JJ+1) = TUM		DAUX33
70	IF (K1.NE.KQ2(M)) GO TO 75		DAUX33
	IF (IJK(MNS,NNS).NE.0) GO TO 72		DAUX33
C			DAUX33
C	FOR ANY M>N SUCH THAT K1(N) = K2(M)		DAUX33
C			DAUX33
C		-1	DAUX33
C	C33(N,M) = C(N,M) + B31(N,K1) M (K1)A13(K2,M)		DAUX33
C		-1	DAUX33
C	+ B32(N,K1)PHI (K1)A23(K2,M)		DAUX33
C		-1	DAUX33
C	C33(M,N) = C(M,N) + B31(M,K2) M (K1)A13(K1,N)		DAUX33
C		-1	DAUX33
C	+ B32(M,K2)PHI (K1)A23(K1,N)		DAUX33
C			DAUX33
	IJ - IJ+1		DAUX33
	IJK(MNS,NNS) - IJ		DAUX33
	IJK(NNS,MNS) - IJ+1		DAUX33
	DO 71 J-1,3		DAUX33
	DO 71 I-1,3		DAUX33
	C(I,J,IJ) = 0.0		DAUX33
71	C(I,J,IJ+1) = 0.0		DAUX33
	IJ - IJ+1		DAUX33
72	JJ - IJK(MNS,NNS)		DAUX33
	DO 74 I-1,3		DAUX33
	DO 74 J-1,3		DAUX33
	SUM = C(I,J,JJ)		DAUX33
	TUM = C(I,J,JJ+1)		DAUX33
	DO 73 K-1,3		DAUX33
	SUM = SUM + B31(I,K,2*N-1)* RW(K1)*A13(K,J,2*M)		DAUX33

	* + B32(I,K,2*N-1)*RPHI(K,K1)*A23(K,J,2*M)	DAUX33
73	TUM = TUM + B31(I,K,2*M)* RW(K1)*A13(K,J,2*N-1)	DAUX33
	* + B32(I,K,2*M)*RPHI(K,K1)*A23(K,J,2*N-1)	DAUX33
	C(I,J,JJ) = SUM	DAUX33
74	C(I,J,JJ+1) = TUM	DAUX33
75	IF (ISING(K2).NE.0) GO TO 85	DAUX33
	IF (K2.NE.KQ1(M)) GO TO 80	DAUX33
	IF (IJK(MNS,NNS).NE.0) GO TO 77	DAUX33
C		DAUX33
C	FOR ANY M>N SUCH THAT K2(N) = K1(M)	DAUX33
C		DAUX33
C		DAUX33
C		DAUX33
C	C33(N,M) = C(N,M) + B31(N,K2) M (K2)A13(K1,M)	DAUX33
C		DAUX33
C	+ B32(N,K2)PHI (K2)A23(K1,M)	DAUX33
C		DAUX33
C		DAUX33
C		DAUX33
C	C33(M,N) = C(M,N) + B31(M,K1) M (K2)A13(K2,N)	DAUX33
C		DAUX33
C	+ B32(M,K1)PHI (K2)A23(K2,N)	DAUX33
C		DAUX33
	IJ = IJ+1	DAUX33
	IJK(MNS,NNS) = IJ	DAUX33
	IJK(NNS,MNS) = IJ+1	DAUX33
	DO 76 J=1,3	DAUX33
	DO 76 I=1,3	DAUX33
	C(I,J,IJ) = 0.0	DAUX33
76	C(I,J,IJ+1) = 0.0	DAUX33
	IJ = IJ+1	DAUX33
77	JJ = IJK(MNS,NNS)	DAUX33
	DO 79 I=1,3	DAUX33
	DO 79 J=1,3	DAUX33
	SUM = C(I,J,JJ)	DAUX33
	TUM = C(I,J,JJ+1)	DAUX33
	DO 78 K=1,3	DAUX33
	SUM = SUM + B31(I,K,2*N)* RW(K2)*A13(K,J,2*M-1)	DAUX33
	* + B32(I,K,2*N)*RPHI(K,K2)*A23(K,J,2*M-1)	DAUX33
78	TUM = TUM + C(I,K,2*M-1)* RW(K2)*A13(K,J,2*N)	DAUX33
	* + C(I,K,2*M-1)*RPHI(K,K2)*A23(K,J,2*N)	DAUX33
	C(I,J,JJ) = SUM	DAUX33
79	C(I,J,JJ+1) = TUM	DAUX33
80	IF (K2.NE.KQ2(M)) GO TO 85	DAUX33
	IF (IJK(MNS,NNS).NE.0) GO TO 82	DAUX33
C		DAUX33
C	FOR ANY M>N SUCH THAT K2(N) = K2(M)	DAUX33
C		DAUX33
C		DAUX33
C		DAUX33
C	C33(N,M) = C(N,M) + B31(N,K2) M (K2)A13(K2,M)	DAUX33
C		DAUX33
C	+ B32(N,K2)PHI (K2)A23(K2,M)	DAUX33
C		DAUX33
C		DAUX33

C	C33(M,N) = C(M,N) + B31(M,K2) M (K2)A13(K2,N)	DAUX33
C		DAUX33
C		DAUX33
C		DAUX33
	IJ = IJ+1	DAUX33
	IJK(MNS,NNS) = IJ	DAUX33
	IJK(NNS,MNS) = IJ+1	DAUX33
	DO 81 J=1,3	DAUX33
	DO 81 I=1,3	DAUX33
	C(I,J,IJ) = 0.0	DAUX33
81	C(I,J,IJ+1) = 0.0	DAUX33
	IJ = IJ+1	DAUX33
82	JJ = IJK(MNS,NNS)	DAUX33
	DO 84 I=1,3	DAUX33
	DO 84 J=1,3	DAUX33
	SUM = C(I,J,JJ)	DAUX33
	TUM = C(I,J,JJ+1)	DAUX33
	DO 83 K=1,3	DAUX33
	SUM = SUM + B31(I,K,2*N) * RW(K2)*A13(K,J,2*M)	DAUX33
	* + B32(I,K,2*N) *RPHI(K,K2)*A23(K,J,2*M)	DAUX33
83	TUM = TUM + B31(I,K,2*M) * RW(K2)*A13(K,J,2*N)	DAUX33
	* + B32(I,K,2*M) *RPHI(K,K2)*A23(K,J,2*N)	DAUX33
	C(I,J,JJ) = SUM	DAUX33
84	C(I,J,JJ+1) = TUM	DAUX33
85	CONTINUE	DAUX33
90	CONTINUE	DAUX33
	CALL ELTIME(2,19)	DAUX33
	RETURN	DAUX33
	END	DAUX33

	SUBROUTINE DAUX44	REV IV	07/24/86	SLIP	DAUX44
C	IMPLICIT REAL*8(A-H,O-Z)				DAUX44
	COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,				DAUX44
*	NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG				PAGE
	COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),				DAUX44
*	SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)				DAUX44
	COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60),				SLIP
*	RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),				DAUX44
*	JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)				DAUX44
	COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60),				DAUX44
*	F(3,30),TQ(3,30),WJ(30),A11(3,3,30)				SLIP
	COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),				DAUX44
*	HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12),				DAUX44
*	RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),				DAUX44
*	KQ1(12),KQ2(12),KQTYPE(12)				DAUX44
	COMMON/FLXBLE/ HF(4,12,8),B42(3,3,24),V4(3,8),NFLEX(3,8)				DAUX44
	LOGICAL*1 FREE,LDMY				80386
	COMMON/TEMPVS/ C(3,3,600),RHS(3,54),IJK(54,54),IJ,NQ2S				CHGIII
*	,IDUM(458),FREE(30),LDMY(2),DMMY(27859)				80386
	IF (NFLX.EQ.0) GO TO 99				DAUX44
	CALL ELTIME(1,33)				DAUX44
	DO 90 L=1,NFLX				DAUX44
	N1 = NFLEX(1,L)				DAUX44
	N2 = NFLEX(2,L)				DAUX44
	N3 = NFLEX(3,L)				DAUX44
	IJ = IJ+1				DAUX44
	DO 10 I=1,3				DAUX44
	DO 10 J=1,3				DAUX44
	C(I,J,IJ) = 0.0				DAUX44
	DO 10 K=1,3				DAUX44
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
*	+ B42(I,K,3*L-1)*RPHI(K,N2)*B42(J,K,3*L-1)				DAUX44
*	+ B42(I,K,3*L)*RPHI(K,N3)*B42(J,K,3*L)				DAUX44
	NSL = 2*NS+L				DAUX44
	IJK(NSL,NSL) = IJ				DAUX44
	DO 20 I=1,3				DAUX44
	RHS(I,NSL) = -V4(I,L)				DAUX44
	DO 20 J=1,3				DAUX44
20	RHS(I,NSL) = RHS(I,NSL) + B42(I,J,3*L-2)*U2(I,N1)				DAUX44
*	+ B42(I,J,3*L-1)*U2(I,N2)				DAUX44
*	+ B42(I,J,3*L)*U2(I,N3)				DAUX44
	IF (L.EQ.NFLX) GO TO 30				DAUX44
	LP1 = L+1				DAUX44
	DO 29 M=LP1,NFLX				DAUX44
	DO 28 II=1,3,2				DAUX44
	IL = NFLEX(II,L)				DAUX44
	IF (ISING(IL).NE.0) GO TO 28				DAUX44
	DO 27 JJ=1,3,2				DAUX44
	IF (NFLEX(II,L).NE.NFLEX(JJ,M)) GO TO 27				DAUX44
	NSM = 2*NS+M				DAUX44
	JK = IJK(NSL,NSM)				DAUX44

KJ = IJK(NSM,NSL)	DAUX44
IF (JK.GT.0) GO TO 22	DAUX44
IJK(NSL,NSM) = IJ+1	DAUX44
IJK(NSM,NSL) = IJ+2	DAUX44
JK = IJ+1	DAUX44
KJ = IJ+2	DAUX44
IJ = IJ+2	DAUX44
DO 21 I=1,3	DAUX44
DO 21 J=1,3	DAUX44
21 C(I,J,JK) = 0.0	DAUX44
22 LI = 3*L+II-3	DAUX44
MJ = 3*M+JJ-3	DAUX44
DO 24 I=1,3	DAUX44
DO 24 J=1,3	DAUX44
DO 23 K=1,3	DAUX44
23 C(I,J,JK) = C(I,J,JK) + B42(I,K,LI)*RPHI(K,IL)*B42(J,K,MJ)	DAUX44
24 C(J,I,KJ) = C(I,J,JK)	DAUX44
27 CONTINUE	DAUX44
28 CONTINUE	DAUX44
29 CONTINUE	DAUX44
30 IF (NQ.EQ.0) GO TO 40	DAUX44
DO 39 M=1,NQ	DAUX44
IF (KQTYPE(M).LT.0) GO TO 39	DAUX44
DO 38 II=1,3	DAUX44
LM = 0	DAUX44
IF (NFLEX(II,L).EQ.KQ1(M)) LM = 2*M-1	DAUX44
IF (NFLEX(II,L).EQ.KQ2(M)) LM = 2*M	DAUX44
IF (LM.EQ.0) GO TO 38	DAUX44
IL = NFLEX(II,L)	DAUX44
IF (ISING(IL).NE.0) GO TO 38	DAUX44
NSM = 2*NS+NFLX+M	DAUX44
JK = IJK(NSL,NSM)	DAUX44
KJ = IJK(NSM,NSL)	DAUX44
IF (JK.GT.0) GO TO 32	DAUX44
IJK(NSL,NSM) = IJ+1	DAUX44
IJK(NSM,NSL) = IJ+2	DAUX44
JK = IJ+1	DAUX44
KJ = IJ+2	DAUX44
IJ = IJ+2	DAUX44
DO 31 I=1,3	DAUX44
DO 31 J=1,3	DAUX44
C(I,J,JK) = 0.0	DAUX44
31 C(I,J,KJ) = 0.0	DAUX44
32 LI = 3*L+II-3	DAUX44
DO 33 I=1,3	DAUX44
DO 33 J=1,3	DAUX44
DO 33 K=1,3	DAUX44
C(I,J,JK) = C(I,J,JK) + B42(I,K,LI)*RPHI(K,IL)*A23(K,J,LM)	DAUX44
33 C(I,J,KJ) = C(I,J,KJ) + B32(I,K,LM)*RPHI(K,IL)*B42(J,K,LI)	DAUX44
38 CONTINUE	DAUX44
39 CONTINUE	DAUX44
40 IF (NJNT.EQ.0) GO TO 90	DAUX44

DO 59 M=1,NJNT	DAUX44
IF (JNT(M).EQ.0) GO TO 59	DAUX44
DO 58 II=1,3	DAUX44
LM = 0	DAUX44
IF (NFLEX(II,L).EQ.IABS(JNT(M))) LM = 2*M-1	DAUX44
IF (NFLEX(II,L).EQ.M+1) LM = 2*M	DAUX44
IF (LM.EQ.0) GO TO 58	DAUX44
IL = NFLEX(II,L)	DAUX44
IF (ISING(IL).NE.0) GO TO 58	DAUX44
NSM = 2*NS+NFLX+NQ+M	DAUX44
JK = IJK(NSL,NSM)	DAUX44
KJ = IJK(NSM,NSL)	DAUX44
IF (JK.GT.0) GO TO 42	DAUX44
IJK(NSL,NSM) = IJ+1	DAUX44
IJK(NSM,NSL) = IJ+2	DAUX44
JK = IJ+1	DAUX44
KJ = IJ+2	DAUX44
IJ = IJ+2	DAUX44
DO 41 I=1,3	DAUX44
DO 41 J=1,3	DAUX44
41 C(I,J,JK) = 0.0	DAUX44
42 LI = 3*L+II-3	DAUX44
DO 44 I=1,3	DAUX44
DO 44 J=1,3	DAUX44
DO 43 K=1,3	DAUX44
43 C(I,J,JK) = C(I,J,JK) + B42(I,K,LI)*RPHI(K,IL)*B12(J,K,LM)	DAUX44
44 C(J,I,KJ) = C(I,J,JK)	DAUX44
IF (FREE(M)) GO TO 58	SLIP
NSM = 2*NS+NFLX+NQ+NJNT+M	DAUX44
JK = IJK(NSL,NSM)	DAUX44
KJ = IJK(NSM,NSL)	DAUX44
IF (JK.GT.0) GO TO 52	DAUX44
IJK(NSL,NSM) = IJ+1	DAUX44
IJK(NSM,NSL) = IJ+2	DAUX44
JK = IJ+1	DAUX44
KJ = IJ+2	DAUX44
IJ = IJ+2	DAUX44
DO 51 I=1,3	DAUX44
DO 51 J=1,3	DAUX44
51 C(I,J,JK) = 0.0	DAUX44
52 SET = 1.0	DAUX44
IF (IL.EQ.M+1) SET = -1.0	DAUX44
DO 54 I=1,3	DAUX44
DO 54 J=1,3	DAUX44
DO 53 K=1,3	DAUX44
53 C(I,J,JK) = C(I,J,JK) + SET*B42(I,K,LI)*RPHI(K,IL)*A22(K,J,LM)	DAUX44
54 C(J,I,KJ) = C(I,J,JK)	DAUX44
58 CONTINUE	DAUX44
59 CONTINUE	DAUX44
90 CONTINUE	DAUX44
CALL ELTIME(2,33)	DAUX44
99 RETURN	DAUX44

END

DAUX44

	SUBROUTINE DAUX55	REV IV	07/24/86	SLIP	DAUX55
C	IMPLICIT REAL*8(A-H,O-Z)				DAUX55
	COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,				DAUX55
	* NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG				PAGE
	COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),				DAUX55
	* SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)				DAUX55
	COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60),				SLIP
	* RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),				DAUX55
	* JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)				DAUX55
	COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60),				DAUX55
	* F(3,30),TQ(3,30),WJ(30),A11(3,3,30)				SLIP
	COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),				DAUX55
	* HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12),				DAUX55
	* RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),				DAUX55
	* KQ1(12),KQ2(12),KQTYPE(12)				DAUX55
	COMMON/FLXBLE/ HF(4,12,8),B42(3,3,24),V4(3,8),NFLEX(3,8)				DAUX55
	COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),				DAUX55
	* UNITL,UNITM,UNITT,GRAVTY(3),TWOPI				TWOPI
	LOGICAL*1 FREE,LDDMY				80386
	COMMON/TEMPVS/ C(3,3,600),RHS(3,54),IJK(54,54),IJ,NQ2S				CHG111
	* ,IDUM(458),FREE(30),LDDMY(2),DDMY(27859)				80386
	CALL ELTIME(1,30)				DAUX55
	IS = 0				DAUX55
	DO 99 I=1,NGRND				DAUX55
	IF (ISING(I).LE.0) GO TO 99				DAUX55
	IS = IS+1				DAUX55
	IJ = IJ+1				DAUX55
	IJK(IS , IS) = IJ				DAUX55
	IJK(IS+1,IS+1) = IJ+1				DAUX55
	DO 11 J=1,3				DAUX55
	RHS(J,IS) = U1(J,I) + W(I)*GRAVTY(J)/G				DAUX55
	RHS(J,IS+1) = U2(J,I)				DAUX55
	U1(J,I) = 0.0				DAUX55
	U2(J,I) = 0.0				DAUX55
	DO 10 K=1,3				DAUX55
	C(J,K,IJ) = 0.0				DAUX55
10	C(J,K,IJ+1) = 0.0				DAUX55
	C(J,J,IJ) = W(I)/G				DAUX55
11	C(J,J,IJ+1) = PHI(J,I)				DAUX55
	IJ = IJ+1				DAUX55
	IF (NFLX.EQ.0) GO TO 19				DAUX55
	DO 15 N=1,NFLX				DAUX55
	LN = 0				DAUX55
	IF (NFLEX(1,N).EQ.I) LN = 3*N-2				DAUX55
	IF (NFLEX(2,N).EQ.I) LN = 3*N-1				DAUX55
	IF (NFLEX(3,N).EQ.I) LN = 3*N				DAUX55
	IF (LN.EQ.0) GO TO 15				DAUX55
	DO 14 J=1,3				DAUX55
	DO 14 K=1,3				DAUX55
	C(J,K,IJ+1) = B42(K,J,LN)				DAUX55
14	C(J,K,IJ+2) = B42(J,K,LN)				SLIP

NNS = 2*NS+N	DAUX55
IJK(IS+1,NNS) = IJ+1	DAUX55
IJK(NNS,IS+1) = IJ+2	DAUX55
IJ = IJ+2	DAUX55
15 CONTINUE	DAUX55
19 IF (NQ.EQ.0) GO TO 30	DAUX55
DO 25 N=1,NQ	DAUX55
IF (KQTYPE(N).LT.0) GO TO 25	DAUX55
LN = 0	DAUX55
IF (I.EQ.KQ1(N)) LN = 2*N-1	DAUX55
IF (I.EQ.KQ2(N)) LN = 2*N	DAUX55
IF (LN.EQ.0) GO TO 25	DAUX55
DO 20 J=1,3	DAUX55
DO 20 K=1,3	DAUX55
C(J,K,IJ+1) = A13(J,K,LN)	DAUX55
C(J,K,IJ+2) = A23(J,K,LN)	DAUX55
C(J,K,IJ+3) = B31(J,K,LN)	SLIP
20 C(J,K,IJ+4) = B32(J,K,LN)	SLIP
NNS = 2*NS+NFLX+N	DAUX55
IJK(IS ,NNS) = IJ+1	DAUX55
IJK(IS+1,NNS) = IJ+2	DAUX55
IJK(NNS,IS) = IJ+3	DAUX55
IJK(NNS,IS+1) = IJ+4	DAUX55
IJ = IJ+4	DAUX55
25 CONTINUE	DAUX55
30 IF (NJNT.EQ.0) GO TO 98	DAUX55
DO 65 N=1,NJNT	DAUX55
IF (JNT(N).EQ.0) GO TO 65	DAUX55
LN = 0	DAUX55
IF (I.EQ.IABS(JNT(N))) LN = 2*N-1	DAUX55
IF (I.EQ.N+1) LN = 2*N	DAUX55
IF (LN.EQ.0) GO TO 65	DAUX55
SET = 1.0	DAUX55
IF (I.EQ.N+1) SET = -1.0	DAUX55
DO 40 J=1,3	DAUX55
DO 40 K=1,3	SLIP
C(J,K,IJ+1) = SET*A11(J,K,N)	SLIP
C(J,K,IJ+3) = SET*A11(K,J,N)	SLIP
C(J,K,IJ+2) = B12(K,J,LN)	DAUX55
40 C(J,J,IJ+4) = B12(J,K,LN)	SLIP
NNS = NQ2S + N	DAUX55
IJK(IS ,NNS) = IJ+1	DAUX55
IJK(IS+1,NNS) = IJ+2	DAUX55
IJK(NNS,IS) = IJ+3	DAUX55
IJK(NNS,IS+1) = IJ+4	DAUX55
IJ = IJ+4	DAUX55
IF (FREE(N)) GO TO 65	SLIP
DO 60 J=1,3	DAUX55
DO 60 K=1,3	DAUX55
C(J,K,IJ+1) = SET*A22(J,K,LN)	DAUX55
60 C(J,K,IJ+2) = SET*A22(K,J,LN)	SLIP
NNS = NQ2S + NJNT + N	DAUX55

```
IJK(IS+1,NNS) = IJ+1
IJK(NNS,IS+1) = IJ+2
IJ = IJ+2
65 CONTINUE
98 IS = IS+1
99 CONTINUE
CALL ELTIME(2,30)
RETURN
END
```

```
DAUX55
DAUX55
DAUX55
DAUX55
DAUX55
DAUX55
DAUX55
DAUX55
DAUX55
DAUX55
```

	SUBROUTINE DHHPIN(DD,BN,L,M,N)		DHHPIN
C		REV IV	07/24/86SLIP
C	SETS DD = D(L) IF JOINT M IS NOT PINNED		DHHPIN
C	OR DD = (I-HH.)(D(L)) IF PINNED		DHHPIN
C			DHHPIN
	IMPLICIT REAL*8 (A-H,O-Z)		DHHPIN
	COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),		DHHPIN
*	SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)		DHHPIN
	COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60),		SLIP
*	RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),		DHHPIN
*	JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)		DHHPIN
	COMMON/CEULER/ IEULER(30),HIR(3,3,90),ANG(3,30),ANGD(3,30),		JDRIFT
*	FE(3,30),TQE(3,30),CONST(5,30)		JDRIFT
	DIMENSION DD(3,3),BN(3)		DHHPIN
	DO 10 J=1,3		DHHPIN
	BN(J) = 0.0		DHHPIN
	DO 10 I=1,3		DHHPIN
10	DD(I,J) = D(I,J,L)		DHHPIN
	LGO = IPIN(M)+8		SLIP
	TSIGN = -1.0		DHHPIN
	GO TO (90,90,90,20,90,90,90,90,30,90,90,90,90,30,30),LGO		SLIP
20	IF (IEULER(M).GE.7) GO TO 90		DHHPIN
	IF (IEULER(M).GE.4) GO TO 30		DHHPIN
	TSIGN = 1.0		DHHPIN
	DO 21 J=1,3		DHHPIN
	DO 21 I=1,3		DHHPIN
21	DD(I,J) = 0.0		DHHPIN
30	DO 31 J=1,3		DHHPIN
	BN(J) = HB(1,N)*D(1,J,L) + HB(2,N)*D(2,J,L) + HB(3,N)*D(3,J,L)		DHHPIN
	DO 31 I=1,3		DHHPIN
31	DD(I,J) = DD(I,J) + TSIGN*BN(J)*HB(I,N)		DHHPIN
90	RETURN		DHHPIN
	END		DHHPIN

```

SUBROUTINE DINT
C
C                                     REV IV   07/23/86TWOPI
IMPLICIT REAL*8 (A-H,O-Z)
COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,
*
*   NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG
COMMON/INTEST/ SGTEST(3,4,30),XTEST(360  ),SEGT(120),REGT(120)
C
NOTE: XTEST SINGLY DIMENSIONED HERE.
REAL   SEGT
COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),
*
*   UNITL,UNITM,UNITT,GRAVTY(3),TWOPI
COMMON/CDINT/ UU(4),GH(3,4),
*
*   E(3,240), F(5,240),GG(5,240),Y(5,240),U(5,240),
*
*   H,HPRINT,HS,TPRINT,TSTART,ICNT,IDBL,IFLAG,IDMMY
COMMON/COMAIN/ VAR(240),DER(240),DT,HO,HMAX,HMIN,RSTIME,
*
*   ISTEP,NSTEPS,NDINT,NEQ,IRSIN,IRSOUT
LOGICAL LNRT
CALL ELTIME(1,3)
IF (ISTEP.NE.0) GO TO 11
C
C
C   IN#0: INITIAL CALL TO INTEGRATOR - INITIALIZE AND RESET PARAMETERS
C   NOTE: FOR EARLIER VERSIONS OF CVS, THE VARIABLE 'IN'(ISTEP IN THE
C   CALLING PROGRAM) RAN FROM 1 TO NSTEPS+1, NOW IT RUNS FROM
C   0 TO NSTEPS.
C
C   TPRINT = TIME
C   IDBL = 2
C   K = 0
C   GO TO 13
C
C
C   IN#0: ADVANCE TPRINT - TIME TO RETURN TO CALLING PROGRAM.
C
C
C   11 TPRINT = TPRINT + DT
C   H = HPRINT
C
C
C   ENTRY TO ADVANCE INTEGRATOR
C
C   12 K = 1
C   CALL UPDATE(K)
C
C   NEGATIVE K FROM UPDATE IS INDICATOR TO RESET INTEGRATOR.
C
C   IF (K.EQ.1) GO TO 15
C
C
C   RESET OR INITIALIZE INTEGRATOR.
C
C
C   13 H = HO
C   HPRINT = HO
C   HS = 0.0
C   ICNT = -2
C   IF (ISTEP.EQ.0 .OR. NPRT(26).EQ.2) CALL OUTPUT(0)
C   CALL PDAUX (VAR,DER,NEQ,K)
C   IF (ISTEP.NE.0 .AND. NPRT(26).EQ.2) CALL OUTPUT(1)

```


	DO 14 I=1,NEQ	DINT
	F(1,I) = VAR(I)	DINT
	F(2,I) = DER(I)	DINT
	DO 14 J=3,5	DINT
	F(J,I) = 0.0	DINT
	U(J,I) = 0.0	DINT
14	Y(J,I) = 0.0	DINT
	IF (ISTEP.EQ.0) GO TO 65	DINT
	K = 1	DINT
C		DINT
C	ADJUST H (CURRENT TIME STEP) IF IT WILL ADVANCE T BEYOND TPRINT.	DINT
C		DINT
15	IF (H+EPS(8).GE.TPRINT-TIME) H = TPRINT-TIME	DINT
C		DINT
C	BACKUP ENTRY POINT IF H HAS BEEN HALVED.	DINT
C		DINT
16	D1 = 0.5*H	D1
	CALL TRIGFS	DINT
	TSTART = TIME	DINT
	DO 20 I=1,NEQ	DINT
	U(3,I) = Y(5,I)	DINT
	U(4,I) = U(5,I)	DINT
	DO 20 J=1,5	DINT
20	GG(J,I) = F(J,I)	DINT
	CALL CMPUTE (K,1,D1)	DINT
	IF (K.LT.0) GO TO 50	DINT
	CALL ADJUST (1,D1)	DINT
	K = 2	DINT
	CALL CMPUTE (K,0,D1)	DINT
	IF (K.LT.0) GO TO 50	DINT
	CALL ADJUST (2,D1)	DINT
	NQUAT = K	DINT
	K = 3	DINT
	CALL CMPUTE (K,1, H)	DINT
	IF (K.LT.0) GO TO 50	DINT
	CALL ADJUST (3,D1)	DINT
	DO 49 L=1,NDINT	DINT
	M = 1	DINT
	IF (L.EQ.1) M = 0	DINT
	IF (NPRT(26).NE.2) CALL OUTPUT(0)	DINT
	CALL CMPUTE (K,M, H)	DINT
	IF (K.LT.0) GO TO 50	DINT
	FAIL = 1.0	DINT
	JJ = 0	DINT
	DO 47 II=1,NEQ,3	DINT
	JJ = JJ+1	DINT
	IF (XTEST(II).LE.0.0) GO TO 47	DINT
	TT = DER(II)**2 + DER(II+1)**2 + DER(II+2)**2	DINT
	TX = VAR(II)**2 + VAR(II+1)**2 + VAR(II+2)**2	DINT
	TE = 0.0	DINT
	TY = 0.0	DINT
	I2 = II+2	DINT

```

DO 45 I=II,I2                                DINT
Z = GG(5,I)*(VAR(I)-GG(1,I)) + GG(2,I) + H*(GG(3,I)+H*GG(4,I)) DINT
TE = TE + (DER(I)-Z)**2                       DINT
TYD = TT + TX*GG(5,I)**2                      DINT
IF (TYD.EQ.0.0) TYD = 1.0                     DINT
45 TY = TY + (DER(I)-Z)**2/TYD                 DINT
TM = 1000.0*TIME                              DINT
IF (NPRT(25).NE.0) WRITE (6,46) TM,SEGT(JJ),REGT(JJ),TT,TE,TY, DINT
*                                               (XTEST(I),I-II,I2) DINT
46 FORMAT ('O DINT CONV. TEST',F10.3,2X,A4,2X,A8,6G12.4) DINT
IF (TT.LT.XTEST(II)) GO TO 47                 DINT
IF (XTEST(II+1).GT.0.0 .AND. TE.LT.XTEST(II+1)) GO TO 47 DINT
IF (TY.GT.XTEST(II+2)) GO TO 48              DINT
47 CONTINUE                                    DINT
FAIL = 0.0                                     DINT
48 CALL ADJUST (4,D1)                          DINT
IF (FAIL.EQ.0.0) GO TO 60                     DINT
IF (L.EQ.NDINT) GO TO 49                     DINT
CALL CMPUTE (K,1,D1)                          DINT
IF (K.LT.0) GO TO 50                          DINT
CALL ADJUST (5,D1)                            DINT
49 CONTINUE                                    DINT
IF (NPRT(25).EQ.0) WRITE (6,46) TM,SEGT(JJ),REGT(JJ),TT,TE,TY, DINT
*                                               (XTEST(I),I-II,I2) DINT
50 WRITE (6,51) TIME,H                         DINT
51 FORMAT('O TEST FAILED AT TIME = ',F10.6,' FOR H = ',F10.6) DINT
ICNT = 0                                       DINT
IDBL = IDBL+2                                  DINT
IF (IDBL.GT.6) IDBL = 6                       DINT
IF (K.GE.0) GO TO 58                          DINT
IF (H.GT.HMIN+EPS(8)) GO TO 59                DINT
WRITE (6,52)                                   DINT
52 FORMAT('O PROGRAM TERMINATED. PDAUX NEG SQRT. H < HMIN+EPS8.'/ DINT
* ' RERUN PROGRAM WITH SMALLER HMIN ON INPUT CARD A.4') DINT
STOP 31                                        DINT
58 IF (H.LE.HMIN+EPS(8)) GO TO 61             DINT
IF (NPRT(26).EQ.2) CALL OUTPUT(1)            DINT
59 TIME = TSTART                              DINT
H = 0.5*H                                     DINT
HPRINT = 0.5*HPRINT                          DINT
K = 2                                         DINT
GO TO 16                                       DINT
60 IF (H.GT.0.74*HPRINT) ICNT = ICNT+1       DINT
61 K = 4                                       DINT
M = 0                                         DINT
IF (H.GT.HMIN .AND. IDBL.GT.2) IDBL = IDBL-1 DINT
GG4 = 2.0*H                                   DINT
GG5 = DEXP(-1600.0*H)                        DINT
DO 63 I=1,NEQ                                 DINT
F(3,I) = GG(3,I) + GG4*GG(4,I)               DINT
F(4,I) = GG(4,I)                             DINT
F(5,I) = GG(5,I)                             DINT

```

	Y(3,I) - Y(1,I)	DINT
	Y(4,I) - Y(2,I)	DINT
	Y(5,I) - GG5*U(3,I)	DINT
63	U(5,I) - GG5*U(4,I)	DINT
	CALL QSET(F,Y,VAR,DER,NQUAT)	DINT
	CALL PDAUX (VAR,DER,M,K)	DINT
	DO 64 I=1,NEQ	DINT
	F(1,I) - VAR(I)	DINT
64	F(2,I) - DER(I)	DINT
	HS = H	DINT
	IF (ICNT.LT.IDBL) GO TO 65	DINT
	ICNT = 0	DINT
	H = DMIN1(2.0*H,HMAX)	DINT
	HPRINT = DMIN1(2.0*HPRINT,HMAX)	DINT
65	CALL UPDATE(2)	DINT
	XPRINT = TPRINT - TIME	TGMOD1
	IF(XPRINT.GE.EPS(8).AND.NPRT(26).NE.3.AND.NPRT(26).GE.0)	TGMOD1
	* CALL OUTPUT(1)	TGMOD1
	IF(XPRINT.GE.EPS(8)) GO TO 12	TGMOD1
	LNRT = .FALSE.	TGMOD1
	IF(NPRT(26).GE.0) LNRT = .TRUE.	TGMOD1
	IF(NPRT(26).LT.0) INRT = IABS(NPRT(26))	TGMOD1
	IF(NPRT(26).LT.0) LNRT = (MOD(ISTEP,INRT).EQ.0)	TGMOD1
	IF(LNRT) CALL OUTPUT(1)	TGMOD1
	CALL ELTIME(2,3)	DINT
	RETURN	DINT
	END	DINT

	SUBROUTINE DOTT31 (A,B,C)		DOTT31
C		REV 17	12/20/76DOTT31
C	PERFORMS MATRIX MULTIPLICATION C = AB'		DOTT31
C	WHERE C IS A 3X3 MATRIX, AND A AND B ARE VECTORS OF LENGTH 3.		DOTT31
C			DOTT31
	IMPLICIT REAL*8 (A-H,O-Z)		DOTT31
	DIMENSION A(3) , B(3) , C(3,3)		DOTT31
	DO 10 I=1,3		DOTT31
	DO 10 J=1,3		DOTT31
10	C(I,J) = A(I)*B(J)		DOTT31
	RETURN		DOTT31
	END		DOTT31

	SUBROUTINE DOTT33 (A,B,C)		DOTT33
C		REV 17	01/03/77DOTT33
C	PERFORMS MATRIX MULTIPLICATION C = AB'		DOTT33
C	WHERE A, B AND C ARE ALL 3X3 MATRICEES.		DOTT33
C			DOTT33
	IMPLICIT REAL*8 (A-H,O-Z)		DOTT33
	DIMENSION A(3,3) , B(3,3) , C(3,3)		DOTT33
	DO 10 I=1,3		DOTT33
	DO 10 J=1,3		DOTT33
10	C(I,J) = A(I,1)*B(J,1) + A(I,2)*B(J,2) + A(I,3)*B(J,3)		DOTT33
	RETURN		DOTT33
	END		DOTT33

	SUBROUTINE DOT31 (A,B,C)		DOT31
C		REV 17	01/03/77DOT31
C	PERFORMS MATRIX MULTIPLICATION $C = A'B$		DOT31
C	WHERE A IS A 3X3 MATRIX, AND B AND C ARE VECTORS OF LENGTH 3.		DOT31
C			DOT31
	IMPLICIT REAL*8 (A-H,O-Z)		DOT31
	DIMENSION A(3,3) , B(3) , C(3)		DOT31
	$C(1) = A(1,1)*B(1) + A(2,1)*B(2) + A(3,1)*B(3)$		DOT31
	$C(2) = A(1,2)*B(1) + A(2,2)*B(2) + A(3,2)*B(3)$		DOT31
	$C(3) = A(1,3)*B(1) + A(2,3)*B(2) + A(3,3)*B(3)$		DOT31
	RETURN		DOT31
	END		DOT31

	SUBROUTINE DOT33 (A,B,C)		DOT33
C		REV 17	01/03/77DOT33
C	PERFORMS MATRIX MULTIPLICATION C = A'B		DOT33
C	WHERE A, B AND C ARE ALL 3X3 MATRICEES.		DOT33
C			DOT33
	IMPLICIT REAL*8 (A-H,O-Z)		DOT33
	DIMENSION A(3,3) , B(3,3) . C(3,3)		DOT33
	DO 10 I=1,3		DOT33
	DO 10 J=1,3		DOT33
10	C(I,J) = A(1,I)*B(1,J) + A(2,I)*B(2,J) + A(3,I)*B(3,J)		DOT33
	RETURN		DOT33
	END		DOT33

	SUBROUTINE DRCIJK (D,ANG,ID,HT,J)		DRCIJK
C		REV 18	02/24/78DRCIJK
	IMPLICIT REAL*8 (A-H,O-Z)		DRCIJK
	DIMENSION D(9,22),HT(9,42),ANG(3,2-),ID(4,22),T1(9),T2(9)		DRCIJK
	M = ID(4,J)		DRCIJK
	IF (M.NE.0) GO TO 10		DRCIJK
	CALL DRCYPR (D(1,J),ANG(1,J),ID(1,J))		DRCIJK
	GO TO 99		DRCIJK
10	CALL DRCYPR (T1,ANG(1,J),ID(1,J))		DRCIJK
	IF (M.LT.0) GO TO 20		DRCIJK
	CALL MAT33 (T1,D(1,M),D(1,J))		DRCIJK
	GO TO 99		DRCIJK
20	M = -M		DRCIJK
	CALL DOT33 (HT(1,2*J-3),D(1,M),D(1,J))		DRCIJK
	CALL MAT33 (T1,D(1,J),T2)		DRCIJK
	CALL MAT33 (HT(1,2*J-2),T2,D(1,J))		DRCIJK
99	RETURN		DRCIJK
	END		DRCIJK

	SUBROUTINE DRCQUA(DC,Q)	DRCQUA
C		REV III.5 07/31/85JTF785
C	COMPUTES DIRECTION COSINE MATRIX FROM QUATERNIONS	DRCQUA
	IMPLICIT REAL*8(A-H,O-Z)	DRCQUA
	DIMENSION DC(3,3),Q(4)	DRCQUA
	C = Q(1)**2 - Q(2)**2 - Q(3)**2 - Q(4)**2	JTF785
	DO 12 I = 1,3	DRCQUA
	DO 10 J = 1,3	DRCQUA
10	DC(I,J) = 2.0*Q(I+1)*Q(J+1)	DRCQUA
12	DC(I,I) = DC(I,I) + C	DRCQUA
	E = Q(1) + Q(1)	DRCQUA
	DO 14 I = 1,3	DRCQUA
	J = 1 + MOD(I,3)	DRCQUA
	K = 1 + MOD(I+1,3)	DRCQUA
	D = E*Q(I+1)	DRCQUA
	DC(K,J) = DC(K,J) - D	DRCQUA
14	DC(J,K) = DC(J,K) + D	DRCQUA
	DO 18 I = 1,3	DRCQUA
	DO 18 J = 1,3	DRCQUA
18	IF(DABS(DC(I,J)).GT.1.0D0)DC(I,J) = DSIGN(1.0D0,DC(I,J))	DRCQUA
	RETURN	DRCQUA
	END	DRCQUA


```

SUBROUTINE DRIFT
CORRECTS FOR DRIFT IN CONSTRAINED JOINTS
IMPLICIT REAL*8(A-H,O-Z)
COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,
* NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG
COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),
* SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)
COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60),
* RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),
* JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)
COMMON/CEULER/ IEULER(30),HIR(3,3,90),ANG(3,30),ANGD(3,30),
* FE(3,30),TQE(3,30),CONST(5,30)
COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),
* UNITL,UNITM,UNITT,GRAVTY(3),TWOPI
COMMON/TEMPVS/ T1(3),T2(3),T3(3),T4(3),TP(3,3),H1(3),H2(3)
* ,DDMY(35086)
IF (NJNT.EQ.0) GO TO 51
DO 50 J=1,NJNT
K = IABS(JNT(J))
IF (K.EQ.0) GO TO 50
IF (ISING(J+1).LT.0) GO TO 50
M = 0
IF (IPIN(J).EQ.1) M = 4
IF (IPIN(J).EQ.6) M = 4
IF (IPIN(J).EQ.7) M = 4
IF (IABS(IPIN(J)).NE.4) GO TO 15
IF (IEULER(J).EQ.1) M = 2
IF (IEULER(J).EQ.2) M = 3
IF (IEULER(J).EQ.3) M = 1
IF (IEULER(J).EQ.4) M = 4
IF (IEULER(J).EQ.5) M = 4
IF (IEULER(J).EQ.6) M = 4
15 IF (M.EQ.0) GO TO 50
IF(M.EQ.4)GO TO 23
IF(M.NE.3)GO TO 21
CALL EJOINT(-1,J)
CALL CROSS(HIR(1,2,2*J+29),HIR(1,1,2*J+29),T1)
DO 17 I = 1,3
H1(I) = CONST(4,J)*HIR(I,1,2*J+29) + CONST(5,J)*T1(I)
17 H2(I) = HIR(I,3,2*J+30)
GO TO 25
21 DO 22 I = 1,3
H1(I) = HIR(I,M,2*J+29)
22 H2(I) = HIR(I,M+1,2*J+30)
GO TO 25
23 DO 24 I = 1,3
H1(I) = HB(I,2*J-1)
24 H2(I) = HB(I,2*J)

```

```

C
C   ** ADJUST DC MATRIX FOR CONSTRAINED JOINTS **
C
25 CALL DOT31(D(1,1,K),H1,T1)
   CALL MAT31 (D(1,1,J+1),T1,T2)
   CT = T2(1)*H2(1) + T2(2)*H2(2) + T2(3)*H2(3)
   IF(M.GE.3)GO TO 28
   ST = 1.0/DSQRT((1.0 - CT)*(1.0 + CT))
   DO 27 I = 1,3
27 T2(I) = (H2(I) - CT*T2(I))*ST
   CT = 1.0/ST
28 CALL CROSS(H2,T2,T3)
   DO 30 L=1,3
   CALL CROSS (T3,D(1,L,J+1),T4)
   ST = T3(1)*D(1,L,J+1) + T3(2)*D(2,L,J+1) + T3(3)*D(3,L,J+1)
   ST = ST/(1.0 + CT)
   DO 30 I=1,3
30 D(I,L,J+1) = CT*D(I,L,J+1) - T4(I) + ST*T3(I)
C
C   ** RENORMALIZATION OF DIRECTION COSINE MATRIX BY **
C   ** AVERAGING MATRIX AND TRANSPOSE OF ITS INVERSE **
C
   DO 33 ITER= 1,10
   CALL CFACTT (D(1,1,J+1),TP,DET)
   DO 32 L = 1,3
   DO 32 I = 1,3
   D(I,L,J+1) = 0.5*(D(I,L,J+1)+TP(L,I)/DET)
32 IF (DABS(D(I,L,J+1)).LT.EPS(15)) D(I,L,J+1) = 0.0
   IF (DABS(DET-1.0).LT.EPS(6)) GO TO 41
33 CONTINUE
   WRITE (6,34) J,TIME,DET
34 FORMAT (44H0 DRIFT RENORMALIZATION DID NOT CONVERGE FOR,
*       10H JOINT NO.,I3,7H TIME =,F10.6,6H DET =,F10.6)
C
C   ** ADJUST WMEG FOR CONSTRAINED JOINTS **
C
41 IF(M.NE.4)GO TO 43
   HW = H2(1)*WMEG(1,J+1) - H1(1)*WMEG(1,K)
*   + H2(2)*WMEG(2,J+1) - H1(2)*WMEG(2,K)
*   + H2(3)*WMEG(3,J+1) - H1(3)*WMEG(3,K)
   CALL DOT31 (D(1,1,K),WMEG(1,K),T1)
   CALL MAT31 (D(1,1,J+1),T1,WMEG(1,J+1))
   DO 42 I=1,3
42 WMEG(I,J+1) = WMEG(I,J+1) + HW*H2(I)
   GO TO 50
43 IF(M.NE.3)GO TO 47
   CALL DOT31(D(1,1,K),HIR(1,2,2*J+29),T1)
   CALL MAT31(D(1,1,J+1),T1,H1)
   GO TO 48
47 CALL MAT31(D(1,1,J+1),T1,T2)
   CALL CROSS(T2,H2,H1)
48 CALL DOT31(D(1,1,K),WMEG(1,K),T1)

```

```
CALL MAT31(D(1,1,J+1),T1,T2) DRIFT
HW = H1(1)*(T2(1) - WMEG(1,J+1)) DRIFT
* + H1(2)*(T2(2) - WMEG(2,J+1)) DRIFT
* + H1(3)*(T2(3) - WMEG(3,J+1)) DRIFT
DO 49 I = 1,3 DRIFT
49 WMEG(I,J+1) = WMEG(I,J+1) + HW*H1(I) DRIFT
50 CONTINUE DRIFT
51 RETURN DRIFT
END DRIFT
```


	SUBROUTINE DSMSOL (A, KK, LL)		DSMSOL
C		REV 03	07/08/74 DSMSOL
C	SOLVES A SET OF SIMULTANEOUS LINEAR EQUATIONS AX=B.		DSMSOL
C			DSMSOL
C	ARGUMENTS:		DSMSOL
C	A: 2-DIMENSIONAL(KK, KK+1) MATRIX OF COEFFICIENTS.		DSMSOL
C	KK: NUMBER OF EQUATIONS AND UNKNOWNNS.		DSMSOL
C	LL: 1ST DIMENSION OF A IN CALLING PROGRAM.		DSMSOL
C			DSMSOL
C	CALLING PROGRAM SETUP:		DSMSOL
C	A(I, J) FOR I, J=1, KK		DSMSOL
C	A(I, KK+1) = B(I) FOR I=1, KK		DSMSOL
C	THE SOLUTION X IS RETURNED IN COLUMN KK+1 OF A.		DSMSOL
C	MATRIX A IS DESTROYED BY SUBROUTINE.		DSMSOL
C			DSMSOL
	IMPLICIT REAL*8(A-H, O-Z)		DSMSOL
	DIMENSION A(LL, 1)		DSMSOL
	N = KK		DSMSOL
	N1 = N+1		DSMSOL
	DO 50 L=1, N		DSMSOL
	L1 = L+1		DSMSOL
	BIG = 0.0		DSMSOL
	DO 25 I=L, N		DSMSOL
	IF (DABS(A(I, L)).LE.DABS(BIG)) GO TO 25		DSMSOL
	K = I		DSMSOL
	BIG = A(I, L)		DSMSOL
25	CONTINUE		DSMSOL
	IF (BIG.NE.0.0) GO TO 30		DSMSOL
	WRITE (6, 26)		DSMSOL
26	FORMAT('O DSMSOL MATRIX SINGULAR, PROGRAM TERMINATED.')		DSMSOL
	STOP 41		DSMSOL
30	BIG = 1.0/BIG		DSMSOL
	DO 40 J=L, N1		DSMSOL
	B = A(K, J)		DSMSOL
	A(K, J) = A(L, J)		DSMSOL
40	A(L, J) = B*BIG		DSMSOL
	IF (L.EQ.N) GO TO 50		DSMSOL
	DO 48 I=L1, N		DSMSOL
	IF (A(I, L).EQ.0.0) GO TO 48		DSMSOL
	DO 45 J=L1, N1		DSMSOL
45	A(I, J) = A(I, J) - A(I, L)*A(L, J)		DSMSOL
48	CONTINUE		DSMSOL
50	CONTINUE		DSMSOL
	IF (N.EQ.1) GO TO 71		DSMSOL
	N2 = N-1		DSMSOL
	DO 60 L=1, N2		DSMSOL
	I = N-L		DSMSOL
	L1 = I+1		DSMSOL
	DO 60 J=L1, N		DSMSOL
60	A(I, N1) = A(I, N1) - A(I, J)*A(J, N1)		DSMSOL
71	CONTINUE		DSMSOL
	RETURN		DSMSOL

END

DSMSOL

```

SUBROUTINE DZP(N,X,GG,E,R,M)
C
C          REV IV    07/23/86TWOPI
C          COMPUTES THE STATE VARIABLES (X) FROM THE PARAMETRIC FORM ASSUMED
C          IN THE INTEGRATION ROUTINE DINT. ALSO EVALUATES THE EXPONENTIAL
C          WEIGHTS (E) IF M IS NOT ZERO.
C
C          IMPLICIT REAL*8 (A-H,O-Z)
C          DIMENSION X(1),GG(5,1),E(3,1)
C          COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),
*          UNITL,UNITH,UNITT,GRAVITY(3),TWOPI
C
C          CALL ELTIME(1,5)
C          IF(M.NE.0) GO TO 10
C
C          COMPUTE STATE VARIABLES ONLY.
C
C          DO 5 I=1,N
5   X(I) = GG(1,I) + R*(GG(2,I)*E(1,I)
      + R*(GG(3,I)*E(2,I)
      + R*(GG(4,I)*E(3,I) )))
      GO TO 90
C
C          COMPUTE EXPONENTIAL WEIGHTS AND STATE VARIABLES.
C
10  DO 50 I=1,N
      E(1,I) = 1.0
      E(2,I) = 0.5
      E(3,I) = THIRD
      IF (GG(5,I).EQ.0.0) GO TO 50
      Z = R*GG(5,I)
      W = 0.
      IF (DABS(Z).GT.0.004) GO TO 20
      W = 4.
      A = E(3,I)
      E(3,I) = 0.
15  E(3,I) = E(3,I)+A
      A = A*Z/W
      W = W+1.0
      IF(E(3,I)+A.NE.E(3,I)) GO TO 15
      E(2,I) = 0.5+0.5*Z*E(3,I)
      E(1,I) = 1.+Z*E(2,I)
      GO TO 50
20  IF(Z.GT.-40.) W = DEXP(Z)
      E(1,I) = (W-1.)/Z
      E(2,I) = (E(1,I)-1.)/Z
      E(3,I) = (2.*E(2,I)-1.)/Z
50  X(I) = GG(1,I) + R*(GG(2,I)*E(1,I)
      + R*(GG(3,I)*E(2,I)
      + R*(GG(4,I)*E(3,I) )))
C
90  CALL ELTIME(2,5)
      RETURN

```

END

DZP

```

SUBROUTINE EDEPTH (A,B,XM,T,Y,XA,XB,XL,XU)                                EDEPTH
C                                                                                   REV IV 07/23/86TWOPI
C DETERMINES XA AND XB, THE POINTS OF MAXIMUM PENETRATION OF TWO EDEPTH
C INTERSECTING ELLIPSOIDS A AND B. EDEPTH
C ARGUMENTS A,B,XM,T AND X SAME AS FOR SUBROUTINE INTERS. EDEPTH
C ARGUMENTS XL AND XU, IF NONZERO, ARE FINAL RESULTS OF LAST CALL. EDEPTH
C EDEPTH
C IMPLICIT REAL*8 (A-H,O-Z) EDEPTH
C DIMENSION A(3,3),B(3,3),XM(3),Y(3),XA(3),XB(3) EDEPTH
C COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND. EDEPTH
* NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG PAGE
C COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24), EDEPTH
* UNITL,UNITM,UNITT,GRAVITY(3),TWOPI TWOPI
C DIMENSION C1(3,4),C2(3,4),C3(3,4),XBM(3),PXBL(3),PXAU(3),AB(3,3) EDEPTH
C DIMENSION AXA(3),BxBM(3),PXAL(?),PXBU(3) EDEPTH
C EQUIVALENCE (XBM(1),C1(1,4)), (FXBL(1),C2(1,4)), (PXAU(1),C3(1,4)) EDEPTH
C EDEPTH
C INITIAL GUESSES EDEPTH
C XA = Y/T EDEPTH
C XB = M+(Y-M)/T EDEPTH
C L = -!XB-XA!/!AXA! EDEPTH
C U = -!XB-XA!/!B(XB-M)! EDEPTH
C EDEPTH
D1 = 0.0 EDEPTH
D2 = 0.0 EDEPTH
DO 9 I=1,3 EDEPTH
XA(I) = Y(I)/T EDEPTH
XBM(I) = (Y(I)-XM(I))/T EDEPTH
XB (I) = XBM(I)+XM(I) EDEPTH
9 D1 = D1+(XB(I)-XA(I))**2 EDEPTH
IF (DABS(T-1.0).LE.EPS(6)) GO TO 31 EDEPTH
ITER = 0 EDEPTH
CALL MAT33 (A,B,AB) EDEPTH
IF (XL.NE.0.0) GO TO 11 EDEPTH
IF (XU.NE.0.0) GO TO 11 EDEPTH
D3 = 0.0 EDEPTH
DO 10 I=1,3 EDEPTH
AXA(I) = A(I,1)*XA(1) EDEPTH
* + A(I,2)*XA(2) EDEPTH
* + A(I,3)*XA(3) EDEPTH
D2 = D2 + AXA(I)**2 EDEPTH
BxBM(I) = B(I,1)*XBM(1) EDEPTH
* + B(I,2)*XBM(2) EDEPTH
* + B(I,3)*XBM(3) EDEPTH
10 D3 = D3+BxBM(I)**2 EDEPTH
XL = -DSQRT(D1/D2) EDEPTH
XU = -DSQRT(D1/D3) EDEPTH
C EDEPTH
C START OF ITERATION EDEPTH
C EDEPTH
11 ITER = ITER+1 EDEPTH
IF (NPRT(17).NE.0) WRITE (6,12) ITER,XL,XU,XA,XB EDEPTH

```


C		*	DU	EDEPTH
C				EDEPTH
	C12 = 0.0			EDEPTH
	C22 = 0.0			EDEPTH
	DO 26 I=1,3			EDEPTH
	PXBU(I) = PXAU(I) + XL*(A(I,1)*PXAU(1)			EDEPTH
	* + A(I,2)*PXAU(2) + A(I,3)*PXAU(3))			EDEPTH
	C12 = C12 + AXA(I)*PXAU(I)			EDEPTH
	26 C22 = C22 + BXBM(I)*PXBU(I)			EDEPTH
C				EDEPTH
C			SOLVE FOR DL AND DU	EDEPTH
C			C11*DL + C12*DU = C13	EDEPTH
C			C21*DL + C22*DU = C23	EDEPTH
C				EDEPTH
	DET = C11*C22-C12*C21			EDEPTH
	DL = (C13*C22-C12*C23)/DET			EDEPTH
	DU = (C11*C23-C13*C21)/DET			EDEPTH
C				EDEPTH
C			INCREMENT L AND U	EDEPTH
C			TEST FOR CONVERGENCE	EDEPTH
C				EDEPTH
	XL = XL + DL			EDEPTH
	XU = XU + DU			EDEPTH
	IF (DABS(DL/XL).GT.EPS(12)) GO TO 11			EDEPTH
	IF (DABS(DU/XU).GT.EPS(12)) GO TO 11			EDEPTH
	31 CONTINUE			EDEPTH
	RETURN			EDEPTH
	END			EDEPTH

	DOUBLE PRECISION FUNCTION EFUNCT (TH,THD,SPR,JSTOP)	EFUNCT
C		REV 20 04/29/80EFUNCT
C	COMPUTES NONLINEAR SRRING TORQUE FOR EULER JOINTS.	EFUNCT
C		EFUNCT
C	ARGUMENTS:	EFUNCT
C	TH - THETA IS THE ANGLE OF THE EULER AXIS	EFUNCT
C	THD - THETA DOT	EFUNCT
C	SPR - ARRAY OF 5 VALUES DESCRIBING FUNCTION EVALUATION	EFUNCT
C	JSTOP - INDICATOR TO BE SET TO ONE IF IN STOP	EFUNCT
C		EFUNCT
	IMPLICIT REAL*8(A-H,O-Z)	EFUNCT
	DIMENSION SPR(5)	EFUNCT
	JSTOP = 0	EFUNCT
	EFUNCT = TH*SPR(1)	EFUNCT
	TEN = 10.0	EFUNCT
	Q = DSIGN(TEN*THD,TH*THD)	EFUNCT
	IF (Q.GT.1.0) Q = 1.0	EFUNCT
	IF (Q.LT.-1.0) Q = -1.0	EFUNCT
	X = 0.5*(1.0+SPR(4)+Q*(1.0-SPR(4)))	EFUNCT
	IF (SPR(5).GT.0.0) GO TO 10	EFUNCT
	EFUNCT = X*EFUNCT	EFUNCT
	GO TO 99	EFUNCT
10	IF (DABS(TH).LT.SPR(5)) GO TO 99	EFUNCT
	JSTOP = 1	EFUNCT
	Z = DABS(TH) - SPR(5)	EFUNCT
	EFUNCT = EFUNCT + DSIGN(X*(SPR(2)+Z*SPR(3))*Z**2,TH)	EFUNCT
99	RETURN	EFUNCT
	END	EFUNCT

DO 12 I=1,3	EJOINT
12 ANG(I,J) = ANG(I,J) + CONST(I,J)	EJOINT
IC = IEULER(J)	EJOINT
CALL EULRAD (TH,ANG(1,J),IC)	EJOINT
CALL ROT(H2,3,-ANG(1,J))	EJOINT
DO 13 I=1,3	EJOINT
ANG(I,J) = ANG(I,J) - CONST(I,J)	EJOINT
HIR(I,1,J) = DH1(I,3)	EJOINT
HIR(I,3,J) = DH4(I,3)	EJOINT
HIM(I,1) = HT(I,3,2*J-1)	EJOINT
HIJ(I,3) = HT(I,3,2*J)	EJOINT
LSKIP(I) = .FALSE.	EJOINT
FE(I,J) = 0.0	EJOINT
CV(I) = 0.0	EJOINT
CS(I) = 0.0	EJOINT
V2(I,J) = 0.0	EJOINT
TQE(I,J) = 0.0	EJOINT
13 TQ(I,J) = 0.0	EJOINT
WJ(J) = 0.0	EJOINT
TQC = 0.0	EJOINT
IF (IJ.EQ.4) GO TO 55	EJOINT
CALL MAT31 (HT(1,1,2*J-1),H2(1,1),HIM(1,2))	EJOINT
CALL MAT31 (HT(1,1,2*J-1),H2(1,2),HIM(1,3))	EJOINT
CALL DOT31 (D(1,1,M),HIM(1,2),H2(1,2))	EJOINT
CALL DOT31 (D(1,1,M),HIM(1,3),H2(1,3))	EJOINT
CALL CROSS (H2(1,2),HIR(1,3,J),H2(1,1))	EJOINT
CALL DOT31 (D(1,1,M),WMEG(1,M),TM)	EJOINT
CALL DOT31 (D(1,1,J+1),WMEG(1,J+1),TJ)	EJOINT
SWJ = 0.0	EJOINT
DO 14 I=1,3	EJOINT
HIR(I,2,J) = H2(I,2)	EJOINT
WMJ(I) = TJ(I) - TM(I)	EJOINT
14 SWJ = SWJ + WMJ(I)**2	EJOINT
WJ(J) = DSQRT(SWJ)	EJOINT
CALL DOT31 (HIR(1,1,J),WMJ,AD)	EJOINT
CALL CROSS (TM,HIR(1,1,J),HDT(1,1))	EJOINT
CALL CROSS (TM,HIR(1,2,J),HDT(1,2))	EJOINT
CALL CROSS (TJ,HIR(1,3,J),HDT(1,3))	EJOINT
CALL MAT31 (D(1,1,J+1),HIR(1,1,J),HIJ(1,1))	EJOINT
CALL MAT31 (D(1,1,J+1),HIR(1,2,J),HIJ(1,2))	EJOINT
CALL MAT31 (D(1,1,M),HIR(1,3,J),HIM(1,3))	EJOINT
N = IEULER(J)	EJOINT
DO 15 I=1,3	EJOINT
SH(I) = AD(I)	JDRIFT
DO 15 K=1,3	JDRIFT
HIR(I,K,2*J+29) = HIM(I,K)	JDRIFT
15 HIR(I,K,2*J+30) = HIJ(I,K)	JDRIFT
IF (N.EQ.8) GO TO 19	EJOINT
IF (N.GT.3) GO TO 16	EJOINT
SH(N) = 0.0	EJOINT
GO TO 18	EJOINT
16 DO 17 I=1,3	EJOINT

```

17 IF (I.NE.N-3) SH(I) = 0.0 EJOINT
18 IF (N.NE.2) GO TO 21 EJOINT
19 HX = H2(1,1)*HIR(1,1,J) + H2(2,1)*HIR(2,1,J) + H2(3,1)*HIR(3,1,J) EJOINT
   IF (DABS(HX).GE.EPS(6)) GO TO 20 EJOINT
   SH(1) = ANGD(1,J) . EJOINT
   SH(3) = ANGD(3,J) EJOINT
   GO TO 21 EJOINT
20 CALL DOT31 (H2,WMJ,SH) EJOINT
   SH(1) = SH(1)/HX EJOINT
   IF (N.EQ.2) SH(2) = 0.0 EJOINT
   SH(3) = SH(3)/HX EJOINT
21 DO 22 I=1,3 EJOINT
   ANGD(I,J) = SH(I) EJOINT
22 HDT(I,2) = HDT(I,2) + SH(1)*H2(I,3) EJOINT
   IF (NJ.NE.0) N = IJ+3 EJOINT
   IF (N.GT.3) GO TO 30 EJOINT
   N4 = 4-N EJOINT
   IF (N.EQ.2) AHDT = HDT(1,2)*WMJ(1)+HDT(2,2)*WMJ(2)+HDT(3,2)*WMJ(3) EJOINT
   IF (N.NE.2) AHDT = -(SH(2)*HDT(1,2)+SH(N4)*HDT(1,N4))*H2(1,N) EJOINT
   * -(SH(2)*HDT(2,2)+SH(N4)*HDT(2,N4))*H2(2,N) EJOINT
   * -(SH(2)*HDT(3,2)+SH(N4)*HDT(3,N4))*H2(3,N) EJOINT
   CALL MAT31 (D(1,1,M ),H2(1,N),HB(1,2*J-1)) EJOINT
   CALL MAT31 (D(1,1,J+1),H2(1,N),HB(1,2*J )) EJOINT
   DO 25 I=1,3 EJOINT
   V2(I,J) = AHDT*H2(I,N) EJOINT
25 IF (N.EQ.1) LSKIP(I) = .TRUE. EJOINT
   GO TO 42 EJOINT
30 IF (N.GT.6) GO TO 40 EJOINT
   K3J = 3*J-2 EJOINT
   DO 32 I=1,3 EJOINT
   IF (NJ.EQ.0) GO TO 31 EJOINT
   IF (I.EQ.N-3) CREST = VISC(7,K3J) EJOINT
   TQE(I,J) = H2(I,N-3) EJOINT
   GO TO 32 EJOINT
31 V2(I,J) = -HDT(I,N-3)*AD(N-3) EJOINT
   HB(I,2*J-1) = HIM(I,N-3) EJOINT
   HB(I,2*J ) = HIJ(I,N-3) EJOINT
   IF (I.NE.N-3) LSKIP(I) = .TRUE. EJOINT
32 K3J = K3J + 1 EJOINT
   IF (NJ) 35,42,35 EJOINT
40 IF (N.EQ.7) GO TO 97 EJOINT
42 IF(IJ.NE.0) GOTO 98 EJOINT
   DO 41 I=1,3 EJOINT
   IF (LSKIP(I)) GO TO 41 EJOINT
   K3J = 3*J-3+I EJOINT
   CV(I) = ANGD(I,J)*VISCOS(DABS(ANGD(I,J)),VISC(1,K3J),HA(I,2*J)) EJOINT
   CS(I) = EFUNCT(ANG(I,J),ANGD(I,J),SPRING(1,K3J),JSTOP(I,1,J)) EJOINT
   FE(I,J) = CS(1) + CV(I) + HA(I,2*J)*HA(I,2*J-1) EJOINT
41 CONTINUE EJOINT
   CALL MAT31(HIR(1,1,J),FE(1,J),TQE(1,J)) EJOINT
   IF(NJ.GT.0) GO TO 34 EJOINT
55 IF (IGLOB(J).EQ.0) GO TO 34 EJOINT

```

HD3(1) = TH(3,1)	EJOINT
HD3(2) = TH(3,2)	EJOINT
HD3(3) = TH(3,3)	EJOINT
CALL GLOBAL (J,HD3,DH1,TQC,T9,ANGL)	EJOINT
34 CONTINUE	EJOINT
C	EJOINT
C ADD TORQUE CONVERTED TO LOCAL REFERENCE TO U2 ARRAY BY	EJOINT
C U2(M) = U2(M) + D(M)*TQ	EJOINT
C U2(J+1) = U2(J+1) - D(J+1)*TQ	EJOINT
C	EJOINT
35 DO 51 I=1,3	EJOINT
TQ(I,J) = TQE(I,J)+TQC*T9(I)	EJOINT
TTI(I) = TQ(I,J)	EJOINT
DO 51 K=1,3	EJOINT
U2(K,M) = U2(K,M) + D(K,I,M)*TQ(I,J)	EJOINT
51 U2(K,J+1) = U2(K,J+1) - D(K,I,J+1)*TQ(I,J)	EJOINT
C	EJOINT
C STORE DATA INTO PRJNT ARRAY FOR OUTPUT ROUTINE	EJOINT
C	EJOINT
97 PRJNT(1,J) = IEULER(J)	EJOINT
PRJNT(2,J) = ANG(1,J)	EJOINT
PRJNT(3,J) = ANG(2,J)	EJOINT
PRJNT(4,J) = ANG(3,J)	EJOINT
PRJNT(5,J) = CS(1)**2 + CS(3)**2 + 2.0*CS(1)*CS(3)*TH(3,3) + CS(2)**2	JTF785
PRJNT(6,J) = CV(1)**2 + CV(3)**2 + 2.0*CV(1)*CV(3)*TH(3,3) + CV(2)**2	JTF785
PRJNT(7,J) = TQ(1,J)**2 + TQ(2,J)**2 + TQ(3,J)**2	EJOINT
98 CONTINUE	EJOINT
CALL ELTIME(2,31)	EJOINT
99 RETURN	EJOINT
END	EJOINT

	DOUBLE PRECISION FUNCTION ELONG(A,B,C,D,E)		ELONG
C		REV 01 10/05/72	ELONG
C	COMPUTES ARC LENGTH OF ELLIPSE	$AX^{**2} + 2BXY + CY^{**2} - 1$	ELONG
C	FROM THETA=0 (POSITIVE X AXIS) TO THETA=E (RADIANS)		ELONG
C	WHERE D IS NOMINAL INCREMENT OF INTEGRATION.		ELONG
C			ELONG
	IMPLICIT REAL*8(A-H,O-Z)		ELONG
	N=DABS(E/D)		ELONG
	N=N+N		ELONG
	IF(N.EQ.0)N=2		ELONG
	Z=N		ELONG
	T=E/Z		ELONG
	F = DSQRT ((1.+(B/A)**2)/A)		ELONG
	CS=1.		ELONG
	SN=0.		ELONG
	DCS=DCOS(T)		ELONG
	DSN=DSIN(T)		ELONG
	S=F/2.		ELONG
	AC = A+C		ELONG
	BAC = B*B-A*C		ELONG
	DO 10 I=1,N,2		ELONG
	CSS=CS*DCS-SN*DSN		ELONG
	SN=SN*DCS+CS*DSN		ELONG
	CS=CSS		ELONG
	G=(A*CS+B*SN)*CS+(B*CS+C*SN)*SN		ELONG
	G = G**2/(AC + BAC/G)		ELONG
	F=(F+1./(F*G))/2.		ELONG
	S=S+F+F		ELONG
	CSS=CS*DCS-SN*DSN		ELONG
	SN=SN*DCS+CS*DSN		ELONG
	CS=CSS		ELONG
	G=(A*CS+B*SN)*CS+(B*CS+C*SN)*SN		ELONG
	G = G**2/(AC + BAC/G)		ELONG
	F=(F+1./(F*G))/2.		ELONG
	S=S+F		ELONG
10	CONTINUE		ELONG
	ELONG=(S+S-F)*T/3.		ELONG
	RETURN		ELONG
	END		ELONG

C		ELTIME
C	CALL AT END OF NTH SUBROUTINE.	ELTIME
C		ELTIME
	30 CALL TIMER(INOW)	80386
	MTOUT = INOW-ISTART	80386
	NDIFF = MTOUT-MTIN(N)	ELTIME
	MTIN(N) = -1	ELTIME
	IF (NDIFF.EQ.0) GO TO 99	80386
	NT(N) = NT(N) + NDIFF	ELTIME
	DO 31 I=1,40	ELTIME
	IF (MTIN(I).NE.-1) MTIN(I) = MTIN(I) + NDIFF	ELTIME
	31 CONTINUE	ELTIME
	GO TO 99	80386
C		ELTIME
C	SUBSEQUENT CALLS FROM MAIN PROGRAM, PRINT SUMMARY TABLE.	ELTIME
C		ELTIME
	40 CALL TIMER(INOW)	80386
	NTSUM = INOW-ISTART	80386
	NT(1) = NTSUM - MTIN(1)	ELTIME
	TIME = FLOAT(NTSUM)/100.0	ELTIME
	WRITE (6,41) TIME,L	PAGE
	41 FORMAT('1 ELAPSED CPU TIME = ',F10.2,' SECONDS',85X,'PAGE',I5//	PAGE
	* ' SUB CALLS TIME % '//)	ELTIME
	PCSUM = 0.0	ELTIME
	NTSUM = 0	ELTIME
	DO 42 I=1,NSUB	ELTIME
	J = IND(I)	ELTIME
	PC = FLOAT(NT(J))/TIME	ELTIME
	PCSUM = PCSUM + PC	ELTIME
	NTSUM = NTSUM + NT(J)	ELTIME
	42 WRITE (6,43) SUB(J),NC(J),NT(J),PC	ELTIME
	43 FORMAT(A10,2I10,F10.2)	ELTIME
	WRITE (6,44) NTSUM,PCSUM	ELTIME
	44 FORMAT('0TOTAL',14X,I10,F10.2)	ELTIME
	99 RETURN	ELTIME
	END	ELTIME

*	'PAGE', I5/120X, 'CARD G.4'/)	PAGE
	ICARD = 4	EQUILB
	JCARD = 0	EQUILB
	IF (NVAR.LT.1 .OR. NVAR.GT.10) GO TO 65	EQUILB
	IF (NCON.LT.0 .OR. NCON.GT.5) GO TO 65	EQUILB
	WRITE (6,52)	EQUILB
52	FORMAT('0',4X,'J',4X,'NTV',3X,'N11',3X,'NSG',8X,'GX',12X,'XDEV',	EQUILB
	*7X,'JPL',3X,'JSG',3X,'NAV',3X,'KSG(I,J),I-1,NAV',28X,'CARDS G.5'/)	EQUILB
	ICARD = 5	EQUILB
	DO 58 J=1,NVAR	EQUILB
	JCARD = J	EQUILB
	READ (5,53) NTV(J),N11(J),NSG(J),GX(J),XDEV(J),	EQUILB
	* JPL(J),JSG(J),IAV,(KSG(I,J),I-1,IAV)	EQUILB
53	FORMAT(3I4,2F8.0,8I4)	EQUILB
	NAV(J) = IAV	EQUILB
	WRITE (6,54) J,NTV(J),N11(J),NSG(J),GX(J),XDEV(J),	EQUILB
	* JPL(J),JSG(J),IAV,(KSG(I,J),I-1,IAV)	EQUILB
54	FORMAT(4I6,2F15.6,8I6)	EQUILB
	IF (NTV(J).LT.1 .OR. NTV(J).GT.2) GO TO 65	EQUILB
	IF (N11(J).LT.1 .OR. N11(J).GT.3) GO TO 65	EQUILB
	IF (NSG(J).LT.1 .OR. NSG(J).GT.NSEG) GO TO 65	EQUILB
	IF (NAV(J).LT.0 .OR. NAV(J).GT.5) GO TO 65	EQUILB
	IF (JPL(J).LT.1 .OR. JPL(J).GT.NPL) GO TO 65	EQUILB
	IF (JSG(J).LT.1 .OR. JSG(J).GT.NSEG) GO TO 65	EQUILB
	K = JPL(J)	EQUILB
	NNPL = MNPL(K)	EQUILB
	IF (NNPL.LT.1 .OR. NNPL.GT.5) GO TO 65	EQUILB
	DO 55 I=1,NNPL	EQUILB
	IF (JSG(J).NE.MPL(2,I,K)) GO TO 55	EQUILB
	JSG(J) = I	EQUILB
	GO TO 56	EQUILB
55	CONTINUE	EQUILB
	GO TO 65	EQUILB
56	IF (NAV(J).LE.0) GO TO 58	EQUILB
	DO 57 I=1,IAV	EQUILB
	IF (KSG(I,J).LT.1 .OR. KSG(I,J).GT.NSEG) GO TO 65	EQUILB
57	CONTINUE	EQUILB
58	CONTINUE	EQUILB
	IF (NCON.LE.0) GO TO 17	EQUILB
	WRITE (5,59)	EQUILB
59	FORMAT('0',4X,'I',4X,'IPL',3X,'ISG',2X,'LTYPE',2X,'INDGX',	EQUILB
	* 87X,'CARDS G.6'/)	EQUILB
	ICARD = 6	EQUILB
	DO 64 I=1,NCON	EQUILB
	JCARD = I	EQUILB
	READ (5,60) IPL(I),ISG(I),LTYPE(I),INDGX(I)	EQUILB
	WRITE (6,61) I,IPL(I),ISG(I),LTYPE(I),INDGX(I)	EQUILB
60	FORMAT(4I4)	EQUILB
61	FORMAT(5I6)	EQUILB
	IF (IPL(I).LT.1 .OR. IPL(I).GT.NPL) GO TO 65	EQUILB
	IF (ISG(I).LT.1 .OR. ISG(I).GT.NSEG) GO TO 65	EQUILB
	IF (LTYPE(I).LT.3 .OR. LTYPE(I).GT.4) GO TO 65	EQUILB

	CALL OUTPUT(0)	EQUILB
	CALL DAUX(0)	EQUILB
	CALL PRINT(6H USER)	EQUILB
	CALL OUTPUT(1)	EQUILB
C		EQUILB
C	START FDF FORCE -> CONSTRAINT FORCE ITERATION	EQUILB
C		EQUILB
20	PENDOT = 0.0	EQUILB
	DO 50 JITTER=1,10	EQUILB
C		EQUILB
C	ITERATE INPUT (X) SUCH THAT F(X) = G(X)	EQUILB
C		EQUILB
	MVAR = 2	EQUILB
	IF (NVAR.EQ.1) MVAR = 1	EQUILB
	DO 32 M=1,2	EQUILB
	DO 32 I=MVAR,NVAR	EQUILB
	DO 32 J=1,I	EQUILB
	NITER = 10	EQUILB
	IF (DXP(J).EQ.0.0) NITER = 50	EQUILB
	DX(J) = 0.25	EQUILB
	N1 = M1(J)	EQUILB
	N2 = M2(J)	EQUILB
	N3 = M3(J)	EQUILB
	NP = JPL(J)	EQUILB
	NT = MT(J)	EQUILB
	I1 = NI1(J)	EQUILB
	I2 = NSG(J)	EQUILB
	IAV = NAV(J)	EQUILB
	IF (NTV(J).NE.2) GO TO 15	EQUILB
	CALL DRCIJK (D,YPR,IYPR,HT,I2)	EQUILB
	IF (NAV(J).LE.0) GO TO 15	EQUILB
	DO 14 K=1,IAV	EQUILB
	J2 = KSG(K,J)	EQUILB
14	CALL DRCIJK (D,YPR,IYPR,HT,J2)	EQUILB
15	DO 29 ITER=1,NITER	EQUILB
	CALL CHAIN(0)	JDRIFT
	PEN1 = PEN	EQUILB
	NPSF = 1	EDGE
	CALL PLELP(N2,N3,N1,NP,NT)	MISDOT
	PEN = PSF(1,1)	EDGE
	FX1(J) = FX(J)	EQUILB
	FXJ = 0.0	EQUILB
	IF (PEN.GT.0.0) FXJ = PSF(2,1)	EDGE
	IF (PEN.GT.0.0) CALL FRCDL (PEN,PENDOT,NT,1,FXJ,ELOSS)	EQUILB
	FX(J) = FXJ	EQUILB
	IF (JX(J)-2) 23,21,25	EQUILB
21	IF (FX(J)*FX1(J).GT.0.0) GO TO 22	EQUILB
	IF (FX1(J).EQ.0.0) JX(J) = 1	EQUILB
	FX(J) = FX1(J)	EQUILB
	PEN = PEN1	EQUILB
	DX(J) = 0.5*DX(J)	EQUILB
	X(J) = X(J) - DX(J)	EQUILB

```

      GO TO 27
22  F2 = FX(J) - GX(J)
      F1 = FX1(J) - GX(J)
      IF (F1*F2.LE.0.0) GO TO 24
      IF (DABS(F2).LT.DABS(F1)) GO TO 23
26  FX(J) = FX1(J)
      DX(J) = -DX(J)
      PEN = PEN1
      X(J) = X(J) + 2.0*DX(J)
      GO TO 27
23  JX(J) = 1
      IF (PEN.GT.0.0) JX(J) = 2
      IF (ITER.GT.1 .AND. PEN.LT.0.0 .AND. PEN.LT.PEN1) GO TO 26
      X(J) = X(J) + DX(J)
      GO TO 27
24  DXP(J) = DX(J)/(FX(J)-FX1(J))
      JX(J) = 3
25  IF (DABS(FX(J)-GX(J)).LT.EPS(6)) GO TO 30
      IF (PEN.LT.0.0) CALL FRCDL (-PEN,PENDOT,NT,1,FXJ,ELOSS)
      IF (PEN.LT.0.0) FX(J) = -FXJ
      X(J) = X(J) - DXP(J)*(FX(J)-GX(J))
27  IF (XDEV(J).LE.0.0) GO TO 42
      IF (DABS(X(J)-SX(J)).LE.XDEV(J)) GO TO 42
      WRITE (6,41) J,X(J),SX(J),XDEV(J)
41  FORMAT('O PROGRAM IS BEING TERMINATED IN SUBROUTINE EQUILB.'//
*        ' ITERATION FOR VARIABLE NO.',I3,' IS NOT CONVERGING.'//
*        ' VALUE OF X IS OUT OF RANGE. VALUES OF X,SX,XDEV ARE'//
*        3G20.8)
      STOP 27
42  IF (NTV(J).EQ.1) SEGLP(I1,I2) = X(J)
      IF (NTV(J).EQ.2) YPR(I1,I2) = X(J)
      IF (NTV(J).EQ.2) CALL DRCIJK (D,YPR,IYPR,HT,I2)
      IF (NAV(J).LE.0) GO TO 29
      DO 28 K=1,IAV
      J2 = KSG(K,J)
      IF (NTV(J).EQ.1) SEGLP(I1,J2) = X(J) - DPN(K,J)
      IF (NTV(J).EQ.2) YPR(I1,J2) = X(J) - DPN(K,J)
28  IF (NTV(J).EQ.2) CALL DRCIJK (D,YPR,IYPR,HT,J2)
29  CONTINUE
30  IF (NPRT(27).NE.0) WRITE (6,31) M,I,J,ITER,X(J),FX(J)
31  FORMAT(4I3,4X,2F12.6)
32  CONTINUE
      IF (NQ.LE.0) GO TO 40
C
C      COMPUTE VEHICLE COORDINATES FOR FIXED POINT CONSTRAINTS.
C
      DO 35 K=1,NQ
      IF (KQTYPE(K).NE.1) GO TO 35
      IF (KQ2(K).NE.NVEH) GO TO 35
      L = KQ1(K)
      CALL DOT31(D(1,1,L),RK1(1,K),T)
      DO 34 I=1,3

```

```

34 T(I) = T(I) + SEGLP(I,L) - SEGLP(I,NVEH)           EQUILB
    CALL MAT31(D(1,1,NVEH),T,RK2(1,K))                EQUILB
35 CONTINUE                                           EQUILB
40 IF (NPRT(27).EQ.0) GO TO 36                       EQUILB
C                                                     EQUILB
C SOLVE SYSTEM EQUATIONS WITH CONSTRAINTS OFF.       EQUILB
C                                                     EQUILB
    CALL OUTPUT(0)                                     EQUILB
    CALL DAUX(0)                                       EQUILB
    CALL PRINT(6HEQUIL2)                              EQUILB
    CALL OUTPUT(1)                                     EQUILB
C                                                     EQUILB
C SET UP CONSTRAINTS TO PRODUCE ZERO ACCELERATIONS. EQUILB
C                                                     EQUILB
36 NQ = NQORG                                         EQUILB
    IF (NCON.LE.0) GO TO 81                           EQUILB
    DO 37 I=1,NCON                                    EQUILB
    NQ = NQ+1                                         EQUILB
    J = IPL(I)                                        EQUILB
    K = ISG(I)                                        EQUILB
    NT = NTPL(K,J)                                    EQUILB
    NTNQ(I) = NTAB(NT+1)                             EQUILB
    NTAB(NT+1) = -NQ                                  EQUILB
    KQ1(NQ) = MPL(2,K,J)                              EQUILB
    KQ2(NQ) = MPL(1,K,J)                              EQUILB
37 KQTYPE(NQ) = LTYPE(I)                             EQUILB
C                                                     EQUILB
C SOLVE SYSTEM EQUATIONS WITH CONSTRAINTS ON.       EQUILB
C                                                     EQUILB
    CALL OUTPUT(0)                                     EQUILB
    CALL DAUX(0)                                       EQUILB
    IF (NPRT(27).NE.0.AND.JITTER.EQ.1) CALL PRINT(6HEQUIL1) EQUILB
C                                                     EQUILB
C FETCH CONSTRAINTS FORCES NORMAL TO PLANE SURFACES. EQUILB
C STORE FRICTION FORCE AND TURN OFF CONSTRAINTS.     EQUILB
C                                                     EQUILB
    CONV = 1.0                                         EQUILB
    DO 39 I=1,NCON                                    EQUILB
    MQ = NQORG+I                                       EQUILB
    J = IPL(I)                                        EQUILB
    K = ISG(I)                                        EQUILB
    NT = NTPL(K,J)                                    EQUILB
    NTAB(NT+1) = NTNQ(I)                             EQUILB
    M = MPL(2,K,J)                                    EQUILB
    N = MPL(1,K,J)                                    EQUILB
    CALL DOT31(D(1,1,N),PL(1,J),TEMP)                 EQUILB
    T(I) = TEMP(1)*QQ(1,MQ) + TEMP(2)*QQ(2,MQ)+ TEMP(3)*QQ(3,MQ) EQUILB
    I1 = INDGX(I)                                     EQUILB
    IF (I1.GT.0 .AND. DABS(GX(I1)+T(I)).GT.EPS(2)) CONV = 0.0 EQUILB
    IF (I1.GT.0) GX(I1) = 0.5*(GX(I1)-T(I))          EQUILB
    DO 38 L=1,3                                       EQUILB
38 TEMP(L) = QQ(L,MQ) - T(I)*TEMP(L)                EQUILB

```

```

      LT = NTAB(NT)
39  CALL MAT31(D(1,1,M),TEMP,TAB(LT+19))
      NQ = NQORG
      IF (CONV.EQ.1.0) GO TO 81
50  CONTINUE
C
C  PRINT INPUT AND CHANGES MADE.
C
81  IF (NJNT.LE.0) GO TO 86
      CALL OUTPUT(0)
      CALL DAUX(0)
      IPRINT = 0
      DO 84 J=1,NJNT
      IF (IPIN(J).GE.0) GO TO 84
      IF (VISC(4,3*J-2).GT.0.0) GO TO 84
      IF (IPIN(J).EQ.-1) T1 = DABS(XDY(HB(1,2*J),D(1,1,J+1),TQ(1,J)))
      IF (IPIN(J).LE.-2) T1 = DSQRT(TQ(1,J)**2+TQ(2,J)**2+TQ(3,J)**2)
      VISC(4,3*J-2) = 1.5*T1
      IF (IPRINT.EQ.0) WRITE (6,82)
82  FORMAT('0 THE FOLLOWING VALUES FOR THE MAX TORQUE FOR A LOCKED JOEQUILB
*INT ON CARDS B.5 HAVE BEEN SET UP BY SUBROUTINE EQUILB: '//
*      ' J SYM LPIN T1=VISC(4)' /)
      IPRJNT = 1
      WRITE (6,83) J,JOINT(J),IPIN(J),VISC(4,3*J-2)
83  FORMAT(I6,1X,A4,I6,F15.6)
84  CONTINUE
86  IF (NQ.LE.0) GO TO 91
      IPRINT = 0
      DO 89 K=1,NQ
      IF (KQTYPE(K).NE.1) GO TO 89
      IF (KQ2(K).NE.NVEH) GO TO 89
      IF (IPRINT.EQ.0) WRITE (6,87)
87  FORMAT('0 THE FOLLOWING VALUES FOR RK2 ON CARDS D.6 FOR FIXED POIEQUILB
*NT CONSTRAINTS HAVE BEEN CHANGED BY SUBROUTINE EQUILB: '//
*      ' 5X, 'K', 3X, 'KQTYPE', 4X, 'KQ1', 5X, 'KQ2', 8X, 'RK2(X)',
*      ' 9X, 'RK2(Y)', 9X, 'RK2(Z)' /)
      IPRINT = 1
      WRITE (6,88) K,KQTYPE(K),KQ1(K),KQ2(K),(RK2(I,K),I=1,3)
88  FORMAT(I6,3I8,3F15.6)
89  CONTINUE
91  WRITE (6,92)
92  FORMAT('0 THE FOLLOWING VARIABLES ON CARDS G.2 AND G.3 ',
*      ' HAVE BEEN CHANGED BY SUBROUTINE EQUILB: '//)
      DO 95 J=1,NVAR
      IO = NTV(J)
      I1 = NI1(J)
      I2 = NSG(J)
      WRITE (6,93) WORD(IO),I1,I2,SX(J),X(J),BLANK,J,SGX(J),GX(J)
93  FORMAT(4X,A6,'( ',I2,', ',I2,') FROM',F12.6,' TO',F12.6,
*      ' A4, 'AND GX(',I2,') FROM',F12.6,' TO',F12.6)
      IF (NAV(J).LE.0) GO TO 95
      IAV = NAV(J)

```

```
DO 94 I=1,IAV
J2 = KSG(I,J)
ZSX = SX(J) - DPN(I,J)
ZXX = X(J) - DPN(I,J)
94 WRITE (6,93) WORD(I0),I1,J2,ZSX,ZXX
95 CONTINUE
CALL ELTIME (2,35)
RETURN
END
```

```
EQUILB
EQUILB
EQUILB
EQUILB
EQUILB
EQUILB
EQUILB
EQUILB
EQUILB
```


26	TMAX = 0.0	EULRAD
	J = 3	EULRAD
	DO 30 I=1,6	EULRAD
	T(I) = DMOD(T(I),TWOPI)	EULRAD
	IF (DABS(T(I)).GT.PI) T(I) = T(I) - DSIGN(TWOPI,T(I))	EULRAD
	IF (DABS(T(I)).LT.TMAX) GO TO 30	EULRAD
	TMAX = DABS(T(I))	EULRAD
	IF (I.GT.3) J = 0	EULRAD
30	CONTINUE	EULRAD
	IF (Z.LT.0.0) T(J+3) = -T(J+3)	EULRAD
	DO 40 I=1,3	EULRAD
	IJ = I+J	EULRAD
40	A(I) = A(I) + T(IJ)	EULRAD
	RETURN	EULRAD
	END	EULRAD

```

DOUBLE PRECISION FUNCTION EVALFD (D,N,L)                                EVALFD
C                                                                                   REV IV 07/23/86JTF786
C EVALUATE FUNCTION THAT IS DEFINED AT LOCATION N OF TAB ARRAY          EVALFD
C FOR ABSCISSA VALUE D. EVALUATES DERIVATIVE, FUNCTION OR INTEGRAL     EVALFD
C AS L EQUALS 0, 1, OR 2. TAB ARRAY IS DEFINED AS FOLLOWS:           EVALFD
C   TAB(N) - DO (NO RESTRICTIONS ON DO)                                JTF786
C   TAB(N+1) - D1 (F1 DEFINED FOR  $DO < D < |D1|$ )                    JTF786
C   TAB(N+2) - D2 (F2 DEFINED FOR  $|D1| < D < |D2|$ )                    JTF786
C   TAB(N+3) - (NOT CURRENTLY USED)                                    EVALFD
C   TAB(N+4) - (NOT CURRENTLY USED)                                    EVALFD
C   TAB(N+5) - START OF DEFINITION OF 1ST PART OF FUNCTION (F1)     EVALFD
C WHICH IS FOLLOWED BY DEFINITION OF 2ND PART OF FUNCTION (F2),     EVALFD
C IF ANY.                                                            EVALFD
C 2ND PART OF FUNCTION EXISTS IF D2 IS NON-ZERO.                    EVALFD
C SIGN OF D1 DETERMINES FORM OF DEFINITION FOR 1ST PART OF         EVALFD
C THE FUNCTION.                                                      EVALFD
C                                                                                   EVALFD
C   D1 ZERO INDICATES THAT FUNCTION IS CONSTANT D2 FOR ALL D.       EVALFD
C                                                                                   EVALFD
C   D1 POSITIVE INDICATES THAT TAB(N+5)-TAB(N+10) CONTAINS          EVALFD
C   A0,A1,...A5. THE COEFFICIENTS OF A 5TH ORDER POLYNOMIAL.       EVALFD
C                                                                                   EVALFD
C   D1 NEGATIVE INDICATES THAT TAB(N+5) CONTAINS NP (REAL)         EVALFD
C   FOLLOWED BY D(1), F(1), D(2), F(2) ..., D(NP), F(NP)           EVALFD
C                                                                                   EVALFD
C   WARNING- TABULAR FUNCTION MUST BE DEFINED FOR WHOLE RANGE,     EVALFD
C   THAT IS, FROM DO TO D1 INCLUSIVE, OR D1 TO D2 INCLUSIVE.     EVALFD
C                                                                                   EVALFD
C   SIMILARLY, THE SIGN OF D2 (IF NON-ZERO) DETERMINES FORM OF    EVALFD
C   DEFINITION OF 2ND PART OF FUNCTION, IF ANY.                    EVALFD
C                                                                                   EVALFD
C   IF  $D < DO$  AND  $D1 \neq 0$ , DERIVATIVE = 0 OR FUNCTION = F1(DO)      JTF786
C   IF  $D > |D1|$  AND  $D2 = 0$ , DERIVATIVE = 0 OR FUNCTION = F1(|D1|)    JTF786
C   IF  $D > |D2|$  AND  $D2 \neq 0$ , DERIVATIVE = 0 OR FUNCTION = F2(|D2|)  JTF786
C                                                                                   EVALFD
C NOTE: PREVIOUS VERSIONS ASSUMED THAT DO WAS NON-NEGATIVE AND     JTF786
C THAT  $F = 0$  FOR  $D < DO$ .                                           JTF786
C                                                                                   JTF786
C IMPLICIT REAL*8(A-H,O-Z)                                           EVALFD
C COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(1250),TAB(4500) EVALFD
C F = 0.0                                                            EVALFD
C IOUTR = 0                                                         EVALFD
C DO = TAB(N)                                                        EVALFD
C D1 = TAB(N+1)                                                      EVALFD
C D2 = TAB(N+2)                                                      EVALFD
C IF (D1.NE.0.0) GO TO 26                                           EVALFD
C IF (L-1) 40,24,25                                                JTF786
24 F = D2                                                            EVALFD
C GO TO 40                                                            EVALFD
25 F = (D-DO)*D2                                                    EVALFD

```

	GO TO 40	EVALFD
C		EVALFD
C	COMPUTE INDEX OF F1 DEFINITION	EVALFD
C		EVALFD
	26 NP = N+5	EVALFD
	IF (L.EQ.2) GO TO 41	EVALFD
C		EVALFD
C	DERIVATIVES AND FUNCTIONS HERE, INTEGRALS HAVE OTHER LOGIC	EVALFD
C		EVALFD
	IF (D.GE.D0) GOTO 22	JTF786
C		JTF786
C	D < D0, RETURN F=0 FOR L=0, OR F=F1(D0) FOR L=1.	JTF786
C		JTF786
	IF (L.EQ.0) GOTO 40	JTF786
	X = D0	JTF786
	IF (D1.GT.0.0) GOTO 37	JTF786
	F = TAB(NP+2)	JTF786
	GOTO 40	JTF786
22	IF (D.LT.DABS(D1)) GOTO 31	EVALFD
	IF (D2.NE.0.0) GO TO 32	EVALFD
C		EVALFD
C	D .GE. !D1! , D2 = 0	EVALFD
C		EVALFD
	IF (D1.LE.0.0) GO TO 33	EVALFD
C		EVALFD
C	IOUTR.EQ.1 INDICATES D BEYOND RANGE. DERIVATIVE = 0.	EVALFD
C	IOUTR.EQ.0 INDICATES D.LE. !D1!. COMPUTE POLY DERIVATIVE	EVALFD
C		EVALFD
	IF (D.GT.DABS(D1)) IOUTR = 1	EVALFD
	X = D1	EVALFD
	GO TO 37	EVALFD
C		EVALFD
C	D0 < D < !D1!	EVALFD
C		EVALFD
	31 IF (D1.LT.0.0) GO TO 35	EVALFD
	X = D	EVALFD
	GO TO 37	EVALFD
C		EVALFD
C	D .GE. !D1!, D2 NON-ZERO, USE F2	EVALFD
C		EVALFD
	32 MP = 6	EVALFD
C		EVALFD
C	COMPUTE INDEX OF F2 DEFINITION	EVALFD
C		EVALFD
	IF (D1.LT.0.0) MP = 2.0 * TAB(NP)+1.0	EVALFD
	NP = NP+MP	EVALFD
	IF (D.LT.DABS(D2)) GO TO 34	EVALFD
	IF (D2.LT.0.0) GO TO 33	EVALFD
C		EVALFD
C	IOUTR.EQ.1 INDICATES D BEYOND RANGE. DERIVATIVE = 0.	EVALFD
C	IOUTR.EQ.0 INDICATES D.LE. !D2!. COMPUTE POLY DERIVATIVE	EVALFD
C		EVALFD

	IF (D.GT.DABS(D2)) IOUTR = 1	EVALFD
C		EVALFD
C	D .GE. D2 (POSITIVE), EVALUATE F2 FOR D2	EVALFD
C		EVALFD
	X = D2	EVALFD
	GO TO 37	EVALFD
C		EVALFD
C	D EXCEEDS TABULAR DEFINITION, SET F = F(NP)	EVALFD
C	IF TABLE DEFINITION EXTENDS BEYOND RANGE, USE TABLE VALUES	EVALFD
C		EVALFD
	33 MB = TAB(NP)	EVALFD
	NB = NP+MB+MB	EVALFD
	IF (D .LE. TAB(NB-1)) GO TO 35	EVALFD
	IF (L.EQ.1) F=TAB(NB)	EVALFD
	GO TO 40	EVALFD
C		EVALFD
C	!D1! .LE. D < !D2!	EVALFD
C		EVALFD
	34 IF (D2.LT.0.0) GO TO 35	EVALFD
	X = D	EVALFD
	GO TO 37	EVALFD
C		EVALFD
C	EVALUATE F FROM TABULAR DEFINITION	EVALFD
C		EVALFD
	35 MB = TAB(NP)	EVALFD
	K1 = NP+3	EVALFD
	K2 = NP+MB+MB	EVALFD
	DO 36 K=K1,K2,2	EVALFD
	IF (D.GT.TAB(K)) GO TO 36	EVALFD
	IF (L-1) 28,27,40	EVALFD
C		EVALFD
C	EVALUATE DERIVATIVE FROM TABLE	EVALFD
C		EVALFD
	28 F = (TAB(K+1)-TAB(K-1))/(TAB(K)-TAB(K-2)) .	EVALFD
	GO TO 40	EVALFD
C		EVALFD
C	EVALUATE FUNCTION FROM TABLE	EVALFD
C		EVALFD
	27 R2 = TAB(K)-TAB(K-2)	EVALFD
	R1 = (D-TAB(K-2))/R2	EVALFD
	R2 = (TAB(K)-D)/R2	EVALFD
	F = R1*TAB(K+1)+R2*TAB(K-1)	EVALFD
	GO TO 40	EVALFD
	36 CONTINUE	EVALFD
	IF (L.EQ.1) F = TAB(K2)	EVALFD
	GO TO 40	EVALFD
	37 IF (IOUTR.EQ.1 .AND. L.EQ.0) GO TO 40	EVALFD
	IF (L-1) 38,39,40	EVALFD
C		EVALFD
C	EVALUATE DERIVATIVE OF 5TH DEGREE POLYNOMIAL	EVALFD
C		EVALFD
	38 F = TAB(NP+1)+X*(2.0*TAB(NP+2)+X*(3.0*TAB(NP+3)+X*(4.0*TAB(NP+4)+	EVALFD

```

*          X*5.0*TAB(NP+5))))          EVALFD
GO TO 40                                EVALFD
C                                         EVALFD
C          EVALUATE 5TH DEGREE POLYNOMIAL          EVALFD
C                                         EVALFD
39 F =      TAB(NP) + X*(TAB(NP+1)+X*(TAB(NP+2)
*          +X*(TAB(NP+3)+X*(TAB(NP+4)+X*TAB(NP+5))))
GO TO 40                                EVALFD
C                                         EVALFD
C          L-2: COMPUTE INTEGRAL OF FUNCTION FROM D0 TO F.
C                                         EVALFD
C                                         EVALFD
41 IF (D.EQ.D0) GO TO 40                EVALFD
X0 = D0                                EVALFD
X1 = D1                                EVALFD
DO 50 I=1,2                             EVALFD
IF (X1) 43,49,42                         EVALFD
42 A0 = TAB(NP )                         EVALFD
A1 = TAB(NP+1)/2.0                       EVALFD
A2 = TAB(NP+2)/3.0                       EVALFD
A3 = TAB(NP+3)/4.0                       EVALFD
A4 = TAB(NP+4)/5.0                       EVALFD
A5 = TAB(NP+5)/6.0                       EVALFD
NP = NP+6                                EVALFD
X = X0                                    EVALFD
IF (X.NE.0.0) F=F-X*(A0+X*(A1+X*(A2+X*(A3+X*(A4+X*A5))))
X = DMIN1(D,X1)                           EVALFD
IF (X.NE.0.0) F=F+X*(A0+X*(A1+X*(A2+X*(A3+X*(A4+X*A5))))
IF(D.LE.X1) GO TO 40                      EVALFD
IF(I.EQ.1.AND.D2.NE.0.0) GO TO 49        EVALFD
C                                         EVALFD
C          NOTE - NP WAS UPDATED NP=NP+6 BEFORE THIS, READY FOR SECOND PASS
C                                         EVALFD
C                                         EVALFD
F = F + (D-X1)*(TAB(NP-6)+X1*(TAB(NP-5)+X1*(TAB(NP-4)
*          +X1*(TAB(NP-3)+X1*(TAB(NP-2)+X1*TAB(NP-1))))))
GO TO 40                                EVALFD
43 MB = TAB(NP)                           EVALFD
K1 = NP+3                                 EVALFD
K2 = NP+MB+MB                             EVALFD
NP = K2+1                                 EVALFD
DL = DMIN1(D,DABS(X1))                    EVALFD
DO 44 K=K1,K2,2                           EVALFD
IF (X0.GE.TAB(K)) GO TO 44                EVALFD
Z1 = DMAX1(X0,TAB(K-2))                    EVALFD
Z2 = DMIN1(DL,TAB(K))                      EVALFD
FYX = TAB(K-1)*TAB(K) - TAB(K+1)*TAB(K-2) EVALFD
FY = TAB(K+1) - TAB(K-1)                  EVALFD
F = F +(FYX + 0.5*FY*(Z1+Z2)) *(Z2-Z1)/ (TAB(K)-TAB(K-2))
IF (Z2.NE.DL) GO TO 44                    EVALFD
IF(I.EQ.1.AND.D2.NE.0.0) GO TO 49        EVALFD
IF(Z2.EQ.0) GO TO 40                      EVALFD
F = F +(D-Z2)*(FYX+Z2*FY)/ (TAB(K)-TAB(K-2))
GO TO 40                                EVALFD

```

44 CONTINUE
49 X0 = DABS(D1)
50 X1 = D2
40 EVALFD = F
RETURN
END

EVALFD
EVALFD
EVALFD
EVALFD
EVALFD
EVALFD


```

*          ' PROGRAM TERMINATED.'
IF (MXNTB.GT.1250) WRITE (6,63) MXNTB
63 FORMAT ('0 ERROR IN SUBROUTINE FDINIT, SIZE OF NTAB ARRAY =',I8//
*          ' PROGRAM TERMINATED.')
IF (MXTB2.GT.4500.OR.MXNTB.GT.1250) STOP 16
RETURN
END
FDINIT
DIMENB
FDINIT
FDINIT
DIMENB
FDINIT
FDINIT

```



```

SUBROUTINE FINPUT                                FINPUT
C                                               REV IV   02/01/88MISDOT
C INPUT CARDS F.1-F.5 SPECIFYING THE ALLOWED CONTACTS OF THE CRASH FINPUT
C VICTIM BODY SEGMENTS WITH VEHICLE PANELS, BELTS, AIRBAGS AND OTHER FINPUT
C BODY SEGMENTS ALONG WITH THE ASSOCIATED FUNCTIONS TO BE USED FOR FINPUT
C EACH CONTACT.                                FINPUT
C ALSO SETS UP TABLES TO CONTROL TIME HISTORY INFORMATION FOR FINPUT
C EACH FUNCTION FOR EACH ALLOWED CONTACT.      FINPUT
C                                               FINPUT
C IMPLICIT REAL*8(A-H,O-Z)                      FINPUT
COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND, FINPUT
* NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG PAGE
COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60), SLIP
* RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90), FINPUT
* JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30) FINPUT
COMMON/JBARTZ/ MNPL( 30),MNBLT( 8),MNSEG( 30),MNBAG( 6), FINPUT
* MPL(3,5,30),MBLT(3,5,8),MSEG(3,5,30),MBAG(3,10,6), FINPUT
* NTPL( 5,30),NTBLT( 5,8),NTSEG( 5,30) FINPUT
COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(1250),TAB(4500)DIMENB
COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5), FINPUT
* BLTTTL(5,8),PLTTL(5,30),BAGTTL(5,6),SEG(30), FINPUT
* JOINT(30),CGS(30),JS(30) FINPUT
REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT FINPUT
LOGICAL*1 CGS,JS FINPUT
COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24), FINPUT
* HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12), FINPUT
* RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12), FINPUT
* KQ1(12),KQ2(12),KQTYPE(12) FINPUT
COMMON/WINDFR/ WTIME(30),QFU(3,5),QFV(3,5),WF(3,30),IWIND(30), WINDOP
* MWSEG(7,30),NFVSEG(6),NFVNT(5),MOWSEG(30,30) WINDOP
COMMON/TEMPVS/JTITLE(5,51),NF(5),MS(3),KTITLE(31),DMMY(34966) 80386
C                                               FINPUT
REAL JTITLE,KTITLE,BLANK,SURFCE(2,3) FINPUT
DATA BLANK/4H / FINPUT
DATA SURFCE/4H PL,4HANE ,4H BE,4HLT ,4H SEG,4HMENT/ FINPUT
C                                               FINPUT
MXNTI = 50 FINPUT
MXNTB = 0 FINPUT
MXTB2 = MXTB1 FINPUT
C                                               FINPUT
C INPUT ALLOWED CONTACTS AND FUNCTIONS BY REF. NO. FINPUT
C                                               FINPUT
WRITE (6,31) NPG PAGE
NPG=NPG+1 PAGE
31 FORMAT('1 ALLOWED CONTACTS AND ASSOCIATED FUNCTIONS',80X, PAGE
* 'PAGE',15) PAGE
DO 61 I=1,4 FINPUT
IJK = 0 FINPUT
GO TO (32,34,35,36),I FINPUT
32 IF (NPL.LE.0) GO TO 61 FINPUT
C                                               FINPUT
C INPUT NO. OF SEGMENTS TO CONTACT EACH PLANE. FINPUT

```

C	INPUT CARD F.1.A	FINPUT
C		FINPUT
	READ (5,33) (MNPL(J),J=1,NPL)	FINPUT
33	FORMAT(18I4)	FINPUT
	NJJ = NPL	FINPUT
	GO TO 37	FINPUT
34	IF (NBLT.LE.0) GO TO 61	FINPUT
C		FINPUT
C	INPUT NO. OF SEGMENTS TO CONTACT EACH BELT.	FINPUT
C	INPUT CARD F.2.A	FINPUT
C		FINPUT
	READ (5,33) (MNBLT(J),J=1,NBLT)	FINPUT
	NJJ = NBLT	FINPUT
	GO TO 37	FINPUT
35	IF (NSEG.LE.0) GO TO 61	FINPUT
C		FINPUT
C	INPUT NO. OF SEGMENTS TO CONTACT EACH SEGMENT.	FINPUT
C	INPUT CARD F.3.A	FINPUT
C		FINPUT
	READ (5,33) (MNSEG(J),J=1,NSEG)	FINPUT
	NJJ = NSEG	FINPUT
	NSEG1 = NSEG+1	FINPUT
	DO 26 J=NSEG1,NGRND	FINPUT
26	MNSEG(J) = 0	FINPUT
	GO TO 37	FINPUT
36	IF (NJNT.LE.0) GO TO 61	FINPUT
C		FINPUT
C	INPUT CARD F.4.A	FINPUT
C	SUPPLY IGLOB(J)=1 FOR EACH GLOBALGRAPHIC JOINT J=1,NJNT	FINPUT
C		FINPUT
	READ (5,33) (IGLOB(J),J=1,NJNT)	FINPUT
	NJJ = NJNT	FINPUT
C		FINPUT
C	START OF LOOP TO READ CONTACTS FOR PLANES (I=1), BELTS (I=2),	FINPUT
C	SEGMENTS (I=3) AND FUNCTIONS FOR GLOBALGRAPHIC JOINTS (I=4).	FINPUT
C		FINPUT
37	DO 60 J=1,NJJ	FINPUT
	IF (I.EQ.1) NK = MNPL(J)	FINPUT
	IF (I.EQ.2) NK = MNBLT(J)	FINPUT
	IF (I.EQ.3) NK = MNSEG(J)	FINPUT
	IF (I.EQ.4) NK = IGLOB(J)	FINPUT
	IF (NK.LE.0) GO TO 60	FINPUT
	DO 59 K=1,NK	FINPUT
	IF (IJK.EQ.0) WRITE (6,38) I	FINPUT
38	FORMAT('0',119X,'CARDS F.',I1)	FINPUT
	IF (IJK.EQ.0 .AND. I.NE.4) WRITE (6,39) SURFCE(1,I),SURFCE(2,I)	FINPUT
39	FORMAT('0',3X,2A4,8X,'SEGMENT',2X,'FORCE DEFLECTION',6X,'INERTIAL	FINPUT
	*SPIKE',10X,'R FACTOR',13X,'G FACTOR',10X,'FRICTION COEF. OPT')	EDGE
	IF (IJK.EQ.0 .AND. I.EQ.4) WRITE (6,40)	FINPUT
40	FORMAT('0',5X,'JOINT (GLOBALGRAPHIC)',2X,'TORQUE DEFLECTION',6X,'HFINPUT	FINPUT
	*ERRON FORMULA',10X,'R FACTOR',13X,'G FACTOR',10X,'FRICTION COEF.')	FINPUT
	IJK = 1	FINPUT

C		FINPUT
C	INPUT CONTACT SURFACE NO., SEGMENT NO., AND FUNCTION NOS.	FINPUT
C	INPUT CARD F.(I).(K)	FINPUT
C		FINPUT
	READ (5,33) NJ,MS,NF,NX	EDGE
	WRITE (6,41) NJ,MS,NF,NX	EDGE
41	FORMAT('0',I7,'-',I3,I11,'-',I3,I8,4I21,I12)	EDGE
	IF (NJ.NE.J) WRITE (6,42)	FINPUT
42	FORMAT(' CONTACT INPUT ERROR. PROGRAM TERMINATED.')	FINPUT
	IF (NJ.NE.J) STOP 14	FINPUT
	IF (I.NE.2.AND.NF(5).EQ.0) WRITE(6,20)	MISDOT
20	FORMAT(' FRICTION FUNCTION NUMBER CAN NOT BE ZERO FOR THIS TYPE OF	MISDOT
	* CONTACT')	MISDOT
	IF (I.NE.2.AND.NF(5).EQ.0) STOP 105	MISDOT
	NLT = 1	FINPUT
	DO 43 JJ = 1,31	FINPUT
43	KTITLE(JJ) = BLANK	FINPUT
	GO TO (44,46,48,49),I	FINPUT
C		FINPUT
C	PLACE SEGMENT NO. AND INDEX TO NTAB ARRAY INTO M- AND NT- ARRAYS.	FINPUT
C		FINPUT
44	MPL(1,K,J) = MS(1)	FINPUT
	MPL(2,K,J) = MS(2)	FINPUT
	MPL(3,K,J) = MS(3)	FINPUT
	NTPL(K,J) = MXNTB+1	FINPUT
	DO 45 JJ = 1,5	FINPUT
45	KTITLE(JJ) = PLTTL (JJ,J)	FINPUT
	GO TO 50	FINPUT
46	MBLT(1,K,J) = MS(1)	FINPUT
	MBLT(2,K,J) = MS(2)	FINPUT
	MBLT(3,K,J) = MS(3)	FINPUT
	NTBLT(K,J) = MXNTB+1	FINPUT
	DO 47 JJ = 1,5	FINPUT
47	KTITLE(JJ) = BLTTTL (JJ,J)	FINPUT
C		FINPUT
C	SET UP TWO TABLES FOR FULL BELT FRICTION	FINPUT
C		FINPUT
	IF (NF(5).NE.0) NLT = 2	FINPUT
	GO TO 50	FINPUT
48	MSEG(1,K,J) = MS(1)	FINPUT
	MSEG(2,K,J) = MS(2)	FINPUT
	MSEG(3,K,J) = MS(3)	FINPUT
	NTSEG(K,J) = MXNTB+1	FINPUT
	KTITLE (3) = SEG(J)	FINPUT
	GO TO 50	FINPUT
C		FINPUT
C	NOTE: GLOBALGRAPHIC JOINT WILL SAVE NT IN IGLOB ARRAY	FINPUT
C		FINPUT
49	IGLOB(J) = MXNTB+1	FINPUT
	KTITLE(2) = JOINT(J)	FINPUT
C		FINPUT
C	SET UP POINTERS TO TAB ARRAY IN NTAB ARRAY.	FINPUT

C	50 NFJ = MS(2)	FINPUT
	IF (NFJ.GT.0) KTITLE(6) = SEG(NFJ)	FINPUT
	DO 51 JJ=1,NLT	FINPUT
	51 CALL FDINIT	FINPUT
	WRITE (6,53) KTITLE	FINPUT
	53 FORMAT(1X,5A4,1X,A4,5(1X,5A4))	FINPUT
	LT = NTAB(MXNTB-5)	EDGE
	IF (I.EQ.1) TAB(LT+22) = NX	EDGE
	IF (NF(1).NE.0) GO TO 59	EDGE
C		FINPUT
C	IF FORCE DEFLECTION FUNCTION NO. IS ZERO,	FINPUT
C	SET UP FOR ROLLING CONSTRAINT	FINPUT
C		FINPUT
	NQ = NQ+1	FINPUT
	NTAB(MXNTB-4) = -NQ	FINPUT
	KQTYPE(NQ) = -4	FINPUT
	KQ1(NQ) = MS(2)	FINPUT
	KQ2(NQ) = MS(1)	FINPUT
	IF (I.NE.3) GO TO 59	EDGE
	KQ1(NQ) = J	FINPUT
	KQ2(NQ) = MS(2)	FINPUT
	59 CONTINUE	FINPUT
	60 CONTINUE	FINPUT
	61 CONTINUE	FINPUT
C		FINPUT
C	INPUT CARD F.5 - JOINT FUNCTIONS TO BE USED.	FINPUT
C		FINPUT
	IF (NJNT.LE.0) GO TO 81	FINPUT
	IF (NJNTF.NE.0) GO TO 76	FINPUT
	DO 75 J=1,NJNT	FINPUT
	75 JOINTF(J) = 0	FINPUT
	GO TO 81	FINPUT
	76 READ (5,33) (JOINTF(J),J=1,NJNT)	FINPUT
	IJK = 0	FINPUT
	DO 80 J=1,NJNT	FINPUT
	IF (JOINTF(J).EQ.0) GO TO 80	FINPUT
	IF (IJK.EQ.0) WRITE (6,77) NPG	PAGE
	IF (IJK.EQ.0) NPG=NPG+1	PAGE
	77 FORMAT('1',122X,'PAGE',15/120X,'CARD F.5'/	PAGE
	* ' THE FOLLOWING JOINT RESTORING FORCE FUNCTIONS AS DEFINED	FINPUT
	*ON CARDS E.7 WILL BE USED.'//4X,'JOINT',10X,'FUNCTION'//)	FINPUT
	JF = JOINTF(J)	FINPUT
	IJK = 1	FINPUT
	WRITE (6,78) J,JOINT(J),JF,(JTITLE(I,JF),I=1,5)	FINPUT
	78 FORMAT(I6,'-',A4,I10,'-',5A4)	FINPUT
	IF (NTI(JF).EQ.0) WRITE (6,42)	FINPUT
	IF (NTI(JF).EQ.0) STOP 17	FINPUT
	80 CONTINUE	FINPUT
C		FINPUT
C	INPUT CONTACT SEGMENTS FOR AIRBAG, IF ANY.	FINPUT
C		FINPUT

81	IF (NBAG.LE.0) GO TO 69	FINPUT
	IJK = 0	FINPUT
	DO 68 J=1,NBAG	FINPUT
C		FINPUT
C	INPUT CARD F.6.(J)*	FINPUT
C		FINPUT
	READ (5,63) K,NK,(MBAG(2,I,J),MBAG(3,I,J),I=1,NK)	FINPUT
63	FORMAT(2I4,20I2)	FINPUT
	MNBAG(J) = NK	FINPUT
	IF (NK.EQ.0) GO TO 68	FINPUT
	IF (IJK.EQ.0) WRITE (6,64)	FINPUT
64	FORMAT(////5X,'AIRBAG',4X,'VS.',4X,'SEGMENTS',90X,'CARDS F.6')	FINPUT
	IF (K.NE.J) WRITE (6,42)	FINPUT
	IF (K.NE.J) STOP 20	FINPUT
	WRITE (6,65) J,(MBAG(2,I,J),MBAG(3,I,J),I=1,NK)	FINPUT
65	FORMAT('0 NO.',I2,12X,10(I3,'-',I3))	FINPUT
	DO 66 I=1,NK	FINPUT
	K = MBAG(2,I,J)	FINPUT
66	KTITLE(I) = SEG(K)	FINPUT
	WRITE (6,67) (BAGTTL(I,J),I=1,5),(KTITLE(I),I=1,NK)	FINPUT
67	FORMAT(1X,5A4,10(3X,A4))	FINPUT
68	CONTINUE	FINPUT
C		FINPUT
C	INPUT CARDS F.7.A F.7.B FOR SUBROUTINE WINDY.	FINPUT
C		FINPUT
69	DO 85 J=1,NGRND	FINPUT
85	MWSEG(1,J) = 0	FINPUT
	IF (NWINDF.EQ.0) GO TO 99	FINPUT
	READ (5,33) (MWSEG(1,J),J=1,NSEG)	FINPUT
	IPAGE = 0	FINPUT
	DO 73 J=1,NSEG	FINPUT
	IWIND(J) = 0	FINPUT
	WTIME(J) = 0.0	FINPUT
	IF (MWSEG(1,J).EQ.0) GO TO 73	FINPUT
	IF (IPAGE.EQ.0) WRITE (6,70) NPG	PAGE
	IF (IPAGE.EQ.0) NPG=NPG+1	PAGE
70	FORMAT('1 SEGMENT WIND FORCES',102X,'PAGE',I5/120X,'CARDS F.7'/	PAGE
*	75X,'DRAG COEFFICIENT BLOCKING'/	WINDOP
*	' SEGMENT-ELLIPSOID SEGMENT-PLANE',	WINDOP
*	16X,'WIND FORCE FUNCTION',10X,'FUNCTION',9X,	WINDOP
*	'SEGMENTS-ELLIPSOID')	WINDOP
	IPAGE = 1	FINPUT
	READ(5,86)(MWSEG(I,J),I=1,7),(MOWSEG(J,K),K=1,2*MWSEG(7,J))	WINDOP
86	FORMAT (7I4,22I2/(I30,7I2))	WINDOP
	WRITE(6,71)(MWSEG(I,J),I=1,6)	OUT385
71	FORMAT(1H0,I6,2H -,I3,I13,2H ,I3,I31,I23)	OUT385
	IF (IABS(MWSEG(1,J)).NE.J) WRITE (6,42)	WINDOP
	IF (IABS(MWSEG(1,J)).NE.J) STOP 21	WINDOP
	M3 = MWSEG(3,J)	FINPUT
	M4 = MWSEG(4,J)	FINPUT
	M5 = MWSEG(5,J)	FINPUT
	M6 = MWSEG(6,J)	WINDOP

M7 = MWSEG(7,J)	OUT385
DO 172 II=1,5	FIXWBS
KTITLE(II)=BLANK	FIXWBS
172 IF (M6.NE.0) KTITLE(II)=JTITLE(II,M6)	FIXWBS
WRITE (6,72) SEG(J),SEG(M3),(PLTTL(I,M4),I=1,5)	FINPUT
* , (JTITLE(I,M5),I=1,5),(KTITLE(I),I=1,5)	FIXWBS
* , (MOWSEG(J,K),K=1,2*M7)	OUT385
72 FORMAT(3X,A4,14X,A4,1H-,5A4,3X,5A4,3X,5A4,2X,3(5(I3,1H-,I3)/94X))	OUT385
73 CONTINUE	FINPUT
99 RETURN	FINPUT
END	FINPUT

	'SUBROUTINE FLXSEG		FLXSEG
C		REV IV 07/23/86TWOPI	
	IMPLICIT REAL*8(A-H,O-Z)		FLXSEG
	COMMON/CONTRL/ TIME,NSEG,NJNT,NPI,NBLT,NBAG,NVEH,NGRND,		FLXSEG
	* NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG		PAGE
	COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),		FLXSEG
	* SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)		FLXSEG
	COMMON/FLXBLE/ HF(4,12,8),B42(3,3,24),V4(3,8),NFLEX(3,8)		FLXSEG
	COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),		FLXSEG
	* UNITL,UNITM,UNITT,GRAVTY(3),TWOPI		TWOPI
	COMMON/TEMPVS/ TT(3,3), THN(4), CN1(3,3), CN(3,3), WNM1(3),		FLXSEG
	* THND(4), PTD(3), WCSN(3), RHSN(3), RHS1(3),		FLXSEG
	* RHS2(3), GF(3,4), GC(3,3), CGC(3,3), THA(3),		FLXSEG
	* THAD(3), THADEG(3), DN2N1(3,3), RMG(3),DMMY(35009)		80386
	DIMENSION IDYPR(3)		FLXSEG
	DATA IDYPR/3,2,1/		FLXSEG
	IF (NFLX.EQ.0) GO TO 99		FLXSEG
	CALL ELTIME(1,34)		FLXSEG
	IFX = 1		FLXSEG
11	N1 = NFLEX(1,IFX)		FLXSEG
	N3 = NFLEX(3,IFX)		FLXSEG
	CALL DOT33(D(1,1,N3),D(1,1,N1),TT)		FLXSEG
	THN(1) = DATAN2(TT(1,2),TT(1,1))		FLXSEG
	THN(2) = -DASIN(TT(1,3))		FLXSEG
	THN(3) = DATAN2(TT(2,3),TT(3,3))		FLXSEG
	THN(4) = 1.0		FLXSEG
	CT22 = 1.0-TT(1,3)**2		FLXSEG
	CT2 = DSQRT(CT22)		FLXSEG
	ST2 = -TT(1,3)		FLXSEG
	CT1 = TT(1,1)/CT2		FLXSEG
	ST1 = TT(1,2)/CT2		FLXSEG
	CN1(1,1) = -TT(1,1)*TT(1,3)/CT22		FLXSEG
	CN1(1,2) = -TT(1,2)*TT(1,3)/CT22		FLXSEG
	CN1(1,3) = 1.0		FLXSEG
	CN1(2,1) = -ST1		FLXSEG
	CN1(2,2) = CT1		FLXSEG
	CN1(2,3) = 0.0		FLXSEG
	CN1(3,1) = TT(1,1)/CT22		FLXSEG
	CN1(3,2) = TT(1,2)/CT22		FLXSEG
	CN1(3,3) = 0.0		FLXSEG
	CALL DOT31(TT,WMEG(1,N3),WNM1)		FLXSEG
	DO 12 I=1,3		FLXSEG
12	WNM1(I) = WNM1(I) - WMEG(I,N1)		FLXSEG
	CALL MAT31(CN1,WNM1,THND)		FLXSEG
	THND(4) = 0.0		FLXSEG
	CALL CROSS(WMEG(1,N1),WNM1,WCSN)		FLXSEG
	RHSN(1) = ((-THND(1)*ST1*ST2 + THND(2)*CT1/CT2)*WNM1(1)		FLXSEG
	* +(THND(1)*CT1*ST2 + THND(2)*ST1/CT2)*WNM1(2))/CT2		FLXSEG
	RHSN(2) = -THND(1)*(CT1*WNM1(1) + ST1*WNM1(2))		FLXSEG
	RHSN(3) = ((-THND(1)*ST1 + THND(2)*CT1*ST2/CT2)*WNM1(1)		FLXSEG
	* +(THND(1)*CT1 + THND(2)*ST1*ST2/CT2)*WNM1(2))/CT2		FLXSEG
13	N2 = NFLEX(2,IFX)		FLXSEG

	M = 0	FLXSEG
	DO 15 I=1,3	FLXSEG
	DO 14 J=1,4	FLXSEG
	JM = J+M	FLXSEG
	GF(I,J) = 0.0	FLXSEG
	DO 14 K=1,4	FLXSEG
	14 GF(I,J) = GF(I,J) + HF(K,JM,IFX)*THN(K)	FLXSEG
	15 M = M+4	FLXSEG
	DO 17 I=1,3	FLXSEG
	THA(I) = 0.0	FLXSEG
	THAD(I) = 0.0	FLXSEG
	DO 16 J=1,4	FLXSEG
	THA (I) = THA (I) + GF(I,J)*THN (J)	FLXSEG
	16 THAD(I) = THAD(I) + GF(I,J)*THND(J)	FLXSEG
	THA (I) = 0.5*THA(I)	FLXSEG
	17 THADEG(I) = THA(I)/RADIAN	FLXSEG
	CALL DRCYPR (DN2N1,THADEG, IDYPR)	FLXSEG
	CALL MAT33(DN2N1,D(1,1,N1),D(1,1,N2))	FLXSEG
	CSC = DCOS(THA(2))	FLXSEG
	CSS = DSIN(THA(2))	FLXSEG
	CN(1,1) = 0.0	FLXSEG
	CN(2,1) = 0.0	FLXSEG
	CN(3,1) = 1.0	FLXSEG
	CN(1,2) = -DSIN(THA(1))	FLXSEG
	CN(2,2) = DCOS(THA(1))	FLXSEG
	CN(3,2) = 0.0	FLXSEG
	CN(1,3) = CSC*CN(2,2)	FLXSEG
	CN(2,3) = -CSC*CN(1,2)	FLXSEG
	CN(3,3) = -CSS	FLXSEG
	CALL MAT33(GF, CN1, GC)	FLXSEG
	CALL MAT33(CN, GC, CGC)	FLXSEG
	CALL DOT33 (D(1,1,N1),CGC,B42(1,1,3*IFX-2))	FLXSEG
	CALL DOTT33(B42(1,1,3*IFX-2),TT,B42(1,1,3*IFX))	FLXSEG
	DO 20 I=1,3	FLXSEG
	DO 20 J=1,3	FLXSEG
	B42(I,J,3*IFX-2) = B42(I,J,3*IFX-2) - D(J,I,N1)	FLXSEG
	B42(I,J,3*IFX-1) = D(J,I,N2)	FLXSEG
	20 B42(I,J,3*IFX) = -B42(I,J,3*IFX)	FLXSEG
C		FLXSEG
C	COMPUTE V4	FLXSEG
C		FLXSEG
	CALL MAT31(CGC,WNM1,RHS1)	FLXSEG
	DO 21 I=1,3	FLXSEG
	21 RMG(I) = RHS1(I) + WMEG(I,N1)	FLXSEG
	CALL MAT31(DN2N1,RMG,WMEG(1,N2))	FLXSEG
	CALL CROSS(WMEG(1,N1),RHS1,RHS2)	FLXSEG
	CALL MAT31(CGC,WCSN,RHS1)	FLXSEG
	DO 25 I=1,3	FLXSEG
	25 RHS1(I) = RHS2(I) - RHS1(I)	FLXSEG
	CALL MAT31(GC,WNM1,RHS2)	FLXSEG
	RHS1(1) = RHS1(1) - THAD(1)*(CN(2,2)*RHS2(2)-CN(1,2)*CSC*RHS2(3))	FLXSEG
	* - THAD(2)*CN(2,2)*CSS*RHS2(3)	FLXSEG


```

RHS1(2) = RHS1(2) + THAD(1)*(CN(1,2)*RHS2(2)+CN(2,2)*CSC*RHS2(3)) FLXSEG
*          + THAD(2)*CN(1,2)*CSS*RHS2(3) FLXSEG
RHS1(3) = RHS1(3) - THAD(2)*CSC*RHS2(3) FLXSEG
CALL MAT31(GF, RHSN, RHS2) FLXSEG
M = 1 FLXSEG
DO 30 I=1,3 FLXSEG
DO 26 J=1,3 FLXSEG
PTD(J) = 0.0 FLXSEG
DO 26 K=1,3 FLXSEG
KK = K+M-1 FLXSEG
26 PTD(J) = PTD(J) + HF(J, KK, IFX)*THAD(K) FLXSEG
RHS2(I) = RHS2(I) + XDY(PTD, CN1, WMI) FLXSEG
30 M = M+4 FLXSEG
CALL MAT31(CN, RHS2, PTD) FLXSEG
DO 35 I=1,3 FLXSEG
35 RHS1(I) = RHS1(I) + PTD(I) FLXSEG
CALL DOT31(D(1,1,N1), RHS1, V4(1, IFX)) FLXSEG
IF (IFX.EQ.NFLX) GO TO 98 FLXSEG
IFX = IFX+1 FLXSEG
IF (NFLEX(1, IFX).EQ.N1 .AND. NFLEX(3, IFX).EQ.N3) GO TO 13 FLXSEG
GO TO 11 FLXSEG
98 CALL ELTIME(2, 34) FLXSEG
99 RETURN FLXSEG
END FLXSEG

```

```

      DOUBLE PRECISION FUNCTION FINTERP(THETA,PHI,NT)                                FINTERP
C                                                                                   REV IV   04/10/87FNFIX
C COMPUTES THE RESTORING TORQUE OF A JOINT AS A FUNCTION OF THE FINTERP
C FLEXURE ANGLE (THETA) AND THE AZIMUTH ANGLE (PHI) AS DEFINED BY FINTERP
C FUNCTION NO. NT . FINTERP
C FINTERP
C ASSUMES 0 < THETA < PI FINTERP
C          -PI < PHI < PI FINTERP
C          DATA IN TAB ARRAY CONTAINS NTHETA,NPHI FOLLOWED BY FINTERP
C          TWO DIMENSIONAL ARRAY OF FUNCTIONAL VALUES (NTHETA > 0) FINTERP
C          OR POLYNOMIAL COEFFICIENTS (NTHETA < 0) FOR EQUALLY FINTERP
C          SPACED VALUES OF PHI. FINTERP
C
C          THETA(I) = (I-1)*PI/(NTHETA-1) FOR I=1,NTHETA FINTERP
C          PHI(J) = -PI + (J-1)*2*PI/NPHI FOR J=1,NPHI FINTERP
C          F(THETA,PI) = F(THETA,-PI) FINTERP
C
C SUBROUTINE EVALUATES G1(THETA) = F(THETA,PHI(J) ) FINTERP
C          G2(THETA) = F(THETA,PHI(J+1)) FINTERP
C          FOR PHI(J) < PHI < PHI(J+1) FINTERP
C BY LINEAR INTERPOLATION OR POLYNOMIAL EVALUATION AND THEN LINEAR FINTERP
C INTERPOLATES BETWEEN G1 AND G2 TO OBTAIN F(THETA,PHI). FINTERP
C IF F < 0. F IS SET TO ZERO, THEREFORE A DEAD BAND IS OBTAINED FINTERP
C BY NEGATIVE VALUES IN THE TABLE. FINTERP
C
C IMPLICIT REAL*8 (A-H,O-Z) FINTERP
C COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24), FINTERP
C *          UNITL,UNITM,UNITT,GRAVTY(3),TWOPI TWOPI
C COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(1250),TAB(4500)DIMENB
C IERROR = 0 FINTERP
C IF (PHI.LT.-PI) IERROR = 1 FINTERP
C IF (PHI.GT.PI) IERROR = 2 FINTERP
C IF (THETA.LT.0.0) IERROR = 3 FINTERP
C IF (THETA.GT.PI) IERROR = 4 FINTERP
C IF (IERROR.NE.0) WRITE (6,11) IERROR,THETA,PHI,NT FINTERP
11 FORMAT('0 IMPROPER ARGUMENTS TO FUNCTION FINTERP. ERROR CODE =',I4)FINTERP
C *          '0 THETA =',G25.15, ' PHI =',G25.15,' NT =',I6) FINTERP
C IF (IERROR.NE.0) STOP 36 FINTERP
C NF = NTI(NT) + 5 FINTERP
C NTHETA = TAB(NF) FINTERP
C NPHI = TAB(NF+1) FINTERP
C
C DETERMINE INDEX AND INTERPOLATION PARAMETERS FOR PHI. FINTERP
C
C IF (PHI.GE.PI-EPS(15)) PHI=0.0-PI FNFIX
C XNP = (PHI+PI)/TWOPI*TAB(NF+1) TWOPI
C NP1 = XNP FINTERP
C NP2 = NP1+1 FINTERP
C IF (NP2.GE.NPHI) NP2 = 0 FINTERP
C RP2 = XNP - DFLOAT(NP1) FINTERP
C RP1 = 1.0 - RP2 FINTERP
C NTH = IABS(NTHETA) FINTERP

```

	IP1 = NF+1+NP1*NTH	FNTERP
	IP2 = NF+1+NP2*NTH	FNTERP
C		FNTERP
C	DETERMINE INDEX AND INTERPOLATION PARAMETERS FOR THETA.	FNTERP
C		FNTERP
	IF (NTHETA.LT.0) GO TO 20	FNTERP
	XNT = THETA/PI*(TAB(NF)-1.0)	FNTERP
	NT1 = XNT	FNTERP
	RT2 = XNT - DFLOAT(NT1)	FNTERP
	RT1 = 1.0 - RT2	FNTERP
	IT1 = IP1 + NT1	FNTERP
	IT2 = IP2 + NT1	FNTERP
	G1 = RT1*TAB(IT1+1) + RT2*TAB(IT1+2)	FNTERP
	G2 = RT1*TAB(IT2+1) + RT2*TAB(IT2+2)	FNTERP
	GO TO 23	FNTERP
C		FNTERP
C	COMPUTE FOR POLYNOMIALS IN THETA FOR FIXED PHI.	FNTERP
C		FNTERP
20	NPOLY = -NTHETA-1	FNTERP
	IT1 = IP1 + NPOLY + 2	FNTERP
	IT2 = IP2 + NPOLY + 2	FNTERP
	THETA1 = THETA - TAB(IP1+1)	FNTERP
	THETA2 = THETA - TAB(IP2+1)	FNTERP
	G1 = 0.0	FNTERP
	G2 = 0.0	FNTERP
	DO 21 I=1, NPOLY	FNTERP
	IT1 = IT1-1	FNTERP
	IT2 = IT2-1	FNTERP
	G1 = THETA1*(TAB(IT1)+G1)	FNTERP
21	G2 = THETA2*(TAB(IT2)+G2)	FNTERP
	IF (THETA1.LT.0.0) G1=0.0	FNFIX
	IF (THETA2.LT.0.0) G2=0.0	FNFIX
23	FNTERP = RP1*G1 + RP2*G2	FNTERP
	IF (FNTERP.LT.0.0) FNTERP = 0.0	FNTERP
	RETURN	FNTERP
	END	FNTERP

```

SUBROUTINE FRCDFL (D,RATE,M,N,FRCDF,ELOSS)                               FRCDFL
C                                                                               REV III.2 08/08/84REVIII
C   EVALUATE FORCE DEFLECTION FUNCTION AT POINT D, WHERE DEFINITION      FRCDFL
C   OF FUNCTION IS CONTROLLED BY M INDEX OF NTAB ARRAY.                FRCDFL
C   DERIVATIVE, FUNCTION OR INTEGRAL IS EVALUATED AS N = 0,1 OR 2.     FRCDFL
C       NTAB(M) - INDEX TO TAB ARRAY FOR REAL DATA                    FRCDFL
C       NTAB(M+1) - INDEX TO TAB ARRAY FOR BASE FUNCTION               FRCDFL
C       NTAB(M+2) - INDEX TO TAB ARRAY FOR INERTIAL FUNCTION, IF ANY FRCDFL
C
C   ASSUMES 0 < DG < DCUBIC < DREF < DMAX                             FRCDFL
C           BUT ANY < MAY BE LESS THAN OR EQUAL TO                    FRCDFL
C
C   IMPLICIT REAL*8(A-H,O-Z)                                           FRCDFL
C   COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(1250),TAB(4500) DIMENB
C   F = 0.0                                                            FRCDFL
C   ELOSS = 0.0                                                         FRCDFL
C   L = NTAB(M)                                                         FRCDFL
C   TAB(L) = D                                                           FRCDFL
C   IF (D.LT.0.0) GO TO 99                                              FRCDFL
C   DMAX = TAB(L+8)                                                      FRCDFL
C   IF (D.LT.DMAX) GO TO 10                                             FRCDFL
C
C   DMAX < D , USE MAX VALUE                                            FRCDFL
C
C   IF (N-1) 99,9,99                                                    FRCDFL
C   9  FDMAX = TAB(L+10)                                                 FRCDFL
C   F = FDMAX                                                            FRCDFL
C   GO TO 40                                                             FRCDFL
C   10 DREF = TAB(L+7)                                                   FRCDFL
C   IF (D.GE.DREF) GO TO 30                                             FRCDFL
C   DCUBIC = TAB(L+6)                                                   FRCDFL
C   IF (DCUBIC GE.DREF) GO TO 20                                         FRCDFL
C   IF (D.LE.DCUBIC) GO TO 20                                           FRCDFL
C
C   DCUBIC < D < DREF , USE CUBIC                                       FRCDFL
C
C   LC = L+14                                                            FRCDFL
C   DCO = TAB(L+18)                                                      FRCDFL
C   X = D-DCO                                                            FRCDFL
C   IF (N-1) 12,11,95                                                  FRCDFL
C
C   USE CUBIC DEFINITION                                               FRCDFL
C
C   11 F = TAB(LC) + X *(TAB(LC+1)+X*(TAB(LC+2)+X*TAB(LC+3)))          FRCDFL
C   GO TO 40                                                             FRCDFL
C
C   USE DERIVATIVE OF CUBIC                                           FRCDFL
C
C   12 F = TAB(LC+1)+X*(2.0*TAB(LC+2)+X*3.0*TAB(LC+3))                FRCDFL
C   GO TO 99                                                             FRCDFL
C   20 DG = TAB(L+5)                                                    FRCDFL
C   IF (D.LE.DG) GO TO 40                                              FRCDFL

```

C		FRCDFL
C	DG < D < DCUBIC , USE QUADRATIC	FRCDFL
C		FRCDFL
	LQ = L+11	FRCDFL
	X = D-DG	FRCDFL
	IF (N-1) 22,21,99	FRCDFL
C		FRCDFL
C	USE QUADRATIC DEFINITION	FRCDFL
C		FRCDFL
21	F = TAB(LQ)+X*(TAB(LQ+1)+X*TAB(LQ+2))	FRCDFL
	GO TO 40	FRCDFL
C		FRCDFL
C	USE DERIVATIVE OF QUADRATIC.	FRCDFL
C		FRCDFL
22	F = TAB(LQ+1)+X*2.0*TAB(LQ+2)	FRCDFL
	GO TO 99	FRCDFL
C		FRCDFL
C	DREF < D < DMAX, USE BASE FUNCTION	FRCDFL
C		FRCDFL
30	IF (N-1) 31,31,99	FRCDFL
31	NB = NTAB(M+1)	FRCDFL
C		FRCDFL
C	EVALUATE BASE FUNCTION	FRCDFL
C		FRCDFL
	IF (NB.GT.0) F = EVALFD(D,NB,N)	FRCDFL
	NI = NTAB(M+2)	FRCDFL
C		FRCDFL
C	ADD INERTIAL FUNCTION , IF ANY	FRCDFL
C		FRCDFL
	IF (NI.GT.0) F = F+EVALFD(D,NI,N)	FRCDFL
40	IF (N.NE.1) GO TO 99	FRCDFL
C		FRCDFL
C	COMPUTE AND ADD RATE DEPENDENT FUNCTIONS, IF ANY.	FRCDFL
C		FRCDFL
C	CURRENT RESTRICTIONS:	FRCDFL
C		FRCDFL
C	1) COMPUTED FOR N=1 (FUNCTION) ONLY.	FRCDFL
C		FRCDFL
C	2) FUNCTION NOS. M+2,M+3 AND M+4 (USED FOR INERTIAL SPIKE,	FRCDFL
C	R FACTOR AND G FACTOR FUNCTIONS) MUST BE NEGATIVE OR ZERO,	FRCDFL
C	I.E., THESE FUNCTIONS CANNOT BE USED IN CONJUNCTION WITH	FRCDFL
C	THE RATE DEPENDENT FUNCTIONS.	FRCDFL
C		FRCDFL
C	3) ASSUMES THE FUNCTIONAL FORM	FRCDFL
C		FRCDFL
C	$F(D,D') = F1(D) + F2(D)*F3(D') + F4(D')$	FRCDFL
C		FRCDFL
C	WHERE F1(D) IS DEFINED BY FUNCTION NTAB(M+1)>0,	FRCDFL
C	I.E., NORMAL FORCE DEFLECTION FUNCTION WITH NO	FRCDFL
C	INERTIAL SPIKE FUNCTION AND DEFAULT VALUES	FRCDFL
C	R=1 AND G=0 (UNLOADING AND RELOADING SAME AS	FRCDFL
C	ORIGINAL LOADING);	FRCDFL

C		FRCDFL
C	F2(D) IS DEFINED BY FUNCTION NTAB(M+2)<0,	FRCDFL
C	IF NTAB(M+2)=0, F2(D)=0;	FRCDFL
C		FRCDFL
C	F3(D') IS DEFINED BY FUNCTION NTAB(M+3)<0,	FRCDFL
C	IF NTAB(M+3)=0, F3(D')=0;	FRCDFL
C		FRCDFL
C	AND F4(D') IS DEFINED BY FUNCTION NTAB(M+4)<0,	FRCDFL
C	IF NTAB(M+4)=0, F4(D')=0.	FRCDFL
C		FRCDFL
C	NOTE: FUNCTIONAL FORM CAN BE CHANGED BY REVISING PROGRAM	FRCDFL
C	BETWEEN STATEMENTS 40 AND 99.	FRCDFL
C		FRCDFL
	F2 = 0.0	FRCDFL
	F3 = 0.0	FRCDFL
	F4 = 0.0	FRCDFL
	N2 = -NTAB(M+2)	FRCDFL
	N3 = -NTAB(M+3)	FRCDFL
	N4 = -NTAB(M+4)	FRCDFL
	IF (N2.GT.0) F2 = EVALFD (D, N2,N)	FRCDFL
	IF (N3.GT.0) F3 = EVALFD (RATE,N3,N)	FRCDFL
	IF (N4.GT.0) F4 = EVALFD (RATE,N4,N)	FRCDFL
	F = F + F2*F3 + F4	FRCDFL
	ELOSS = RATE*(F2*F3+F4)	FRCDFL
99	FRCDF = F	FRCDFL
	RETURN	FRCDFL
	END	FRCDFL

```

SUBROUTINE FSMSOL (C,R,NN,MX,MAXN,JN,MAXDIM)                                FSMSOL
C                                                                           REV III.2 08/08/84REVIII
C SOLVES A SET OF SIMULTANEOUS EQUATIONS OF SIZE 3*MM                     FSMSOL
C WHERE THE MATRIX CONSISTS OF A SET OF 3*3 SUBMATRICES                   FSMSOL
C STORED IN C(3,3,IJ). THE LOCATION OF THE I,J ELEMENT                     FSMSOL
C IS STORED IN NN(I,J). I.E. IJ= NN(I,J)                                   FSMSOL
C                                                                           FSMSOL
C A NEGATIVE IJ IMPLIES THAT C( , ,!IJ!) IS AN                           FSMSOL
C IDENTITY AND THE RIGHT SIDE IS ZERO. A NEGATIVE                         FSMSOL
C IJ WILL ONLY OCCUR ON A DIAGONAL ENTRY OF NN.                           FSMSOL
C                                                                           FSMSOL
C THE BASIC EQUATION IS CX=R                                              FSMSOL
C                                                                           FSMSOL
C DURING THE SOLUTION THE C MATRIX IS DESTROYED ,IT MAY                   FSMSOL
C BE NECESSARY TO ADD TO THE C ARRAY.                                       FSMSOL
C THE SOLUTION IS STORED IN R.                                             FSMSOL
C                                                                           FSMSOL
C INPUT                                                                      FSMSOL
C                                                                           FSMSOL
C C(3,3,K) GIVEN ARRAY                                                       FSMSOL
C R(3,MM) GIVEN RIGHT HAND SIDE                                             FSMSOL
C NN(JJ,JJ) GIVEN ARRAY CONTAINING LOCATIONS OF I,J,ELEMENT               FSMSOL
C MX SIZE OF SYSTEM OF SUBMATRICES (POSITIVE INDICATES                    FSMSOL
C THAT C MATRIX IS SYMMETRIC, NEGATIVE IT IS NOT.)                         FSMSOL
C MAXN LARGEST VALUE IN NN ARRAY                                           FSMSOL
C JN DIMENSION OF NN                                                         FSMSOL
C MAXDIM THIRD DIMENSION OF C IN CALLING ROUTINE                           FSMSOL
C                                                                           FSMSOL
C IMPLICIT REAL*8 (A-H,O-Z)                                                 FSMSOL
C COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,                   PAGE
* NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG                           PAGE
C DIMENSION C(3,3,1),R(3,1),NN(JN,1)                                       FSMSOL
C CALL ELTIME(1,20)                                                           FSMSOL
C MM = IABS(MX)                                                                FSMSOL
C IF (MM.LE.0) GO TO 99                                                         FSMSOL
C MM1 = MM-1                                                                    FSMSOL
C MP1 = MM+1                                                                    FSMSOL
C DO 50 II=1,MM                                                                FSMSOL
C I = MP1-II                                                                    FSMSOL
C                                                                           FSMSOL
C START PIVOT AT BOTTOM - FIND PIVOT - INVERT.                             FSMSOL
C                                                                           FSMSOL
C L = NN(I,I)                                                                    FSMSOL
C IF (L.LE.0) GO TO 50                                                         FSMSOL
C DO 14 M=1,3                                                                    FSMSOL
C B = 1.0/C(M,M,L)                                                                FSMSOL
C C(M,M,L) = 1.0                                                                FSMSOL
C C(M,1,L) = B*C(M,1,L)                                                         FSMSOL
C C(M,2,L) = B*C(M,2,L)                                                         FSMSOL
C C(M,3,L) = B*C(M,3,L)                                                         FSMSOL
C R(M,I) = B*R(M,I)                                                             FSMSOL
C DO 13 N=1,3                                                                    FSMSOL

```

IF (N.EQ.M) GO TO 13	FSMSOL
B = C(N,M,L)	FSMSOL
C(N,M,L) = 0.0	FSMSOL
C(N,1,L) = C(N,1,L) - B*C(M,1,L)	FSMSOL
C(N,2,L) = C(N,2,L) - B*C(M,2,L)	FSMSOL
C(N,3,L) = C(N,3,L) - B*C(M,3,L)	FSMSOL
R(N,I) = R(N,I) - B*R(M,I)	FSMSOL
13 CONTINUE	FSMSOL
14 CONTINUE	FSMSOL
C	FSMSOL
C CHECK IF DONE.	FSMSOL
C	FSMSOL
IF (I.EQ.1) GO TO 50	FSMSOL
IM1 = I-1	FSMSOL
C	FSMSOL
C CALCULATE PIVOT ROW.	FSMSOL
C	FSMSOL
DO 20 J=1,IM1	FSMSOL
IF (NN(I,J).EQ.0) GO TO 20	FSMSOL
M = NN(I,J)	FSMSOL
DO 15 N=1,3	FSMSOL
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
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
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
C(1,N,M) = A	FSMSOL
C(2,N,M) = B	FSMSOL
15 C(3,N,M) = D	FSMSOL
20 CONTINUE	FSMSOL
C	FSMSOL
C DONE WITH PIVOT ROW - ZERO COLUMN I ABOVE DIAGONAL.	FSMSOL
C	FSMSOL
C 1,1	FSMSOL
C	FSMSOL
C	FSMSOL
C	FSMSOL
C	FSMSOL
C K,K . . K,J . . K,I	C = C - C *C
C	KJ KJ KI IJ
C	FSMSOL
C J,K . . J,J . . J,I	C = C - C *C
C	JK JK JI IK
C	FSMSOL
C I,K . . I,J . . I,I	C = 0
C	KI
C	FSMSOL
C	FSMSOL
C M,M	FSMSOL
C	FSMSOL
C	FSMSOL
DO 40 K=1,IM1	FSMSOL
KI = NN(K,I)	FSMSOL
IK = NN(I,K)	FSMSOL
IF (KI.EQ.0 .AND. IK.EQ.0) GO TO 40	FSMSOL
DO 30 J=K,IM1	FSMSOL
IJ = NN(I,J)	FSMSOL
JI = NN(J,I)	FSMSOL

IF (KI.EQ.0 .OR. IJ.EQ.0) GO TO 24	FSMSOL
KJ = NN(K,J)	FSMSOL
IF (KJ.NE.0) GO TO 22	FSMSOL
MAXN = MAXN+1	FSMSOL
IF (MAXN.GT.MAXDIM) GO TO 41	FSMSOL
KJ = MAXN	FSMSOL
NN(K,J) = KJ	FSMSOL
DO 21 M=1,3	FSMSOL
DO 21 N=1,3	FSMSOL
21 C(N,M,KJ) = 0.0	FSMSOL
22 DO 23 M=1,3	FSMSOL
DO 23 N=1,3	FSMSOL
23 C(N,M,KJ) = C(N,M,KJ) - C(N,1,KI)*C(1,M,IJ)	FSMSOL
* - C(N,2,KI)*C(2,M,IJ)	FSMSOL
* - C(N,3,KI)*C(3,M,IJ)	FSMSOL
24 IF (J.EQ.K) GO TO 30	FSMSOL
IF (JI.EQ.0 .OR. IK.EQ.0) GO TO 30	FSMSOL
JK = NN(J,K)	FSMSOL
IF (JK.NE.0) GO TO 26	FSMSOL
MAXN = MAXN+1	FSMSOL
IF (MAXN.GT.MAXDIM) GO TO 41	FSMSOL
JK = MAXN	FSMSOL
NN(J,K) = JK	FSMSOL
DO 25 M=1,3	FSMSOL
DO 25 N=1,3	FSMSOL
25 C(N,M,JK) = 0.0	FSMSOL
26 IF (MX.LT.0) GO TO 28	FSMSOL
DO 27 M=1,3	FSMSOL
DO 27 N=1,3	FSMSOL
27 C(N,M,JK) = C(M,N,KJ)	FSMSOL
GO TO 30	FSMSOL
28 DO 29 M=1,3	FSMSOL
DO 29 N=1,3	FSMSOL
29 C(N,M,JK) = C(N,M,JK) - C(N,1,JI)*C(1,M,IK)	FSMSOL
* - C(N,2,JI)*C(2,M,IK)	FSMSOL
* - C(N,3,JI)*C(3,M,IK)	FSMSOL
30 CONTINUE	FSMSOL
IF (KI.EQ.0) GO TO 40	FSMSOL
DO 35 N=1,3	FSMSOL
35 R(N,K) = R(N,K) - C(N,1,KI)*R(1,I)	FSMSOL
* - C(N,2,KI)*R(2,I)	FSMSOL
* - C(N,3,KI)*R(3,I)	FSMSOL
40 CONTINUE	FSMSOL
50 CONTINUE	FSMSOL
GO TO 51	FSMSOL
41 WRITE (6,49) MAXDIM,NPG,(L,L=1,MM)	FSMSOL
NPG=NPG+1	PAGE
DO 42 I=1,MM	PAGE
42 WRITE (6,43) I,(NN(I,L),L=1,MM)	FSMSOL
43 FORMAT(I3,3X,40I3,3X/6X,40I3)	FSMSOL
WRITE (6,44) NPG	FSMSOL
NPG=NPG+1	PAGE
	PAGE

44	FORMAT('1 FSMSOL PRINT OF RHS ARRAY',96X,'PAGE',15//)	PAGE
	DO 45 K=1,MM	FSMSOL
45	WRITE (6,46) K,(R(I,K),I=1,3)	FSMSOL
46	FORMAT(16,9G14.7)	FSMSOL
	WRITE (6,47) NPG	PAGE
	NPG=NPG+1	PAGE
47	FORMAT('1 FSMSOL PRINT OF C ARRAY ELEMENTS',89X,'PAGE',15//)	PAGE
	DO 48 K=1,MAXN	FSMSOL
48	WRITE (6,46) K,((C(I,L,K),L=1,3),I=1,3)	FSMSOL
49	FORMAT('1 MAXIMUM DIMENSION OF',14,' ON C ARRAY HAS BEEN EXCEEDED	FSMSOL
	*IN SUBROUTINE FSMSOL.',46X,'PAGE',15//' IF 600, CALL IS FROM SUBROPAGE	
	*UTINE DAUX. IF 200'	PAGE
	* ,' CALL IS FROM SUBROUTINE HPTURB.'//' PROGRAM IS BEING TERMINATE	PAGE
	*D. COMPLETE PRINT-OUT OF IJK, RHS AND C ARRAYS FOLLOW.'//	FSMSOL
	*' FSMSOL PRINT OF IJK MATRIX'//(6X,40I3))	FSMSOL
	STOP 35	FSMSOL
C		FSMSOL
C	BACKDOWN SOLUTION	FSMSOL
C		FSMSOL
51	IF (MM.EQ.1) GO TO 99	FSMSOL
	DO 90 J=1,MM1	FSMSOL
	IP = J+1	FSMSOL
	DO 80 I=IP,MM	FSMSOL
	IF (NN(I,J).EQ.0) GO TO 80	FSMSOL
	IJ = NN(I,J)	FSMSOL
	DO 75 N=1,3	FSMSOL
75	R(N,I) = R(N,I) - C(N,1,IJ)*R(1,J)	FSMSOL
	* - C(N,2,IJ)*R(2,J)	FSMSOL
	* - C(N,3,IJ)*R(3,J)	FSMSOL
80	CONTINUE	FSMSOL
90	CONTINUE	FSMSOL
99	CALL ELTIME(2,20)	FSMSOL
	RETURN	FSMSOL
	END	FSMSOL

C	SUBROUTINE GLOBAL (J,HD3,DH1,T9,ANGL)	REV IV	07/24/86	GLOBAL
	IMPLICIT REAL*8 (A-H,O-Z)			SLIP
	DIMENSION HD3(3),DH1(3,3),T9(3),ANGL(3),CC(3)			GLOBAL
	COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60),			GLOBAL
*	RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),			SLIP
*	JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)			GLOBAL
	COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(1250),TAB(4500)			GLOBAL
	COMMON/TEMPVI/ CREST,TTI(3),RII(3),R2I(3),JSTOP(4,2,30)			DIMENB
	COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),			GLOBAL
*	UNITL,UNITM,UNITT,GRAVTY(3),TWOPI			GLOBAL
	IF (DABS(HD3(3)).GT.1.0-EPS(6)) GO TO 34			TWOPI
	ANGL(1) = DACOS(HD3(3))			GLOBAL
	NT = IGLOB(J)			GLOBAL
	NT1 = NTAB(NT+2)			GLOBAL
	CALL HERRON(HD3,NT1,THETO,THETOP)			GLOBAL
	JSTOP(4,1,J) = 0			GLOBAL
	IF (ANGL(1).LE.THETO) GO TO 34			GLOBAL
	JSTOP(4,1,J) = 1			GLOBAL
	MT = NTAB(NT+5)			GLOBAL
	CREST = TAB(MT+3)			GLOBAL
	STH2 = 1.0-HD3(3)**2			GLOBAL
	STH = DSQRT(STH2)			GLOBAL
	CTH = HD3(3)/STH			GLOBAL
	CST = DSQRT(STH2+THETOP**2)			GLOBAL
	DR = (ANGL(1)-THETO)*STH/CST			GLOBAL
	LT = NTAB(NT)			GLOBAL
	TAB(LT) = DR			GLOBAL
	NTAB(NT+2) = 0			GLOBAL
	DRDOT = 0.0			GLOBAL
	CALL FRCDFL (DR,DRDOT,NT,1,TQF,ELOSS)			GLOBAL
	NTAB(NT+2) = NT1			GLOBAL
	TQC = TQF/CST			GLOBAL
	CC(1) = -HD3(2)+HD3(1)*CTH*THETOP			GLOBAL
	CC(2) = HD3(1)+HD3(2)*CTH*THETOP			GLOBAL
	CC(3) = -STH*THETOP			GLOBAL
	DO 28 L=1,3			GLOBAL
28	T9(L) = CC(1)*DH1(L,1) + CC(2)*DH1(L,2) + CC(3)*DH1(L,3)			GLOBAL
34	RETURN			GLOBAL
	END			GLOBAL

	SUBROUTINE HBELT (J1,J2,KNLO,IND)		HBELT
C		REV IV	02/01/88MISDOT
C	ARGUMENTS:		HBELT
C	J1,J2 - FIRST AND LAST INDEX FOR BELTS.		HBELT
C	KNLO - ZERO VALUE FOR KNL INDEX.		HBELT
C	IND - 0: CALL IS FROM SUBROUTINE CONTCT		HBELT
C	1: CALL IS FROM SUBROUTINE UPDATE		HBELT
C			HBELT
	IMPLICIT REAL*8 (A-H,O-Z)		HBELT
	COMMON/CNTRSF/ PL(24,30),BELT(20,8),TPTS(6,8),BD(24,40)		EDGE
	COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),		MISDOT
*	UNITL,UNITM,UNITT,GRAVITY(3),TWOPI		MISDOT
	COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),		HBELT
*	SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)		HBELT
	COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(1250),TAB(4500)		DIMENB
	COMMON/FORCES/PSF(7,70),BSF(4,20),SSF(10,40),BAGSF(3,20),		NCFORC
*	PRJNT(7,30),NPANEL(5),NPSF,NBSF,NSSF,NBGSF		HBELT
	COMMON/HRNESS/ BAR(15,100),BB(100),BBDOT(100),PLOSS(2,100),		HBELT
*	XLONG(20),HTIME(2),IBAR(5,100),NL(2,100),		HBELT
*	NPTSPB(20),NPTPLY(20),NTHRNS(20),NBLTPH(5)		HBELT
C	THIS COMMON/TEMPVS/ IS SHARED BY HPTURB, HBPLAY, HBELT AND HSETC.		HBELT
	COMMON/TEMPVS/ B(3,3,3),S(3,3),T(3),R(3),V(3),T1(3),T2(3),		HBELT
*	E(3,3,50),EDOT(3,50),FCE(3,50),FR(3,50),ZR(3,50),		HBELT
*	TR(3,50),U(3,50),PTLOSS(2,50),BL(50),FB(50),FP(50),		HBELT
*	OLDBB(100),RHS(3,54),C(3,3,200),IJK(54,54)		HBELT
*	,DMMY(29942)		80386
	CALL ELTIME (1,38)		HBELT
	NTP = 0		HBELT
	K2 = 0		HBELT
	DO 31 JB=J1,J2		HBELT
	IF (IND.EQ.0) NBSF = NBSF + 1		HBELT
	IF (NPTPLY(JB).LE.0) GO TO 31		HBELT
C			HBELT
C	FIRST LOOP ON K		HBELT
C	COMPUTE Z(K),ZR(K),E3(K),U(K-1),BL(K-1),FB(K-1)		HBELT
C	NEED NL(K),BB(K-1)		HBELT
C	NOTE: AN INDEX K-1 REFERS TO BELT SEGMENT BETWEEN K-1 AND K.		HBELT
C			HBELT
	K1 = K2 + 1		HBELT
	K2 = K2 + NPTPLY(JB)		HBELT
	DO 20 K=K1,K2		HBELT
	KNL = KNLO + K		HBELT
	KI = NL(1,KNL)		HBELT
C			HBELT
C	HERE K IS INDEX OF POINTS IN PLAY ON EACH HARNESS		HBELT
C	KNL IS INDEX OF ALL POINTS IN PLAY		HBELT
C	KI IS INDEX OF ALL POINTS		HBELT
C			HBELT
	KS = IABS(IBAR(1,KI))		HBELT
	IF (KS.GT.100) NTP = 1		HBELT
	IF (KS.GT.100) KS = MOD(KS,100)		HBELT
	KE = IBAR(2,KI)		HBELT

CALL DOT31 (D(1,1,KS),BAR(4,KI),T1)	HBELT
CALL DOT31 (D(1,1,KS),BAR(7,KI),T2)	HBELT
DO 11 J=1,3	HBELT
R(J) = V(J)	HBELT
V(J) = BAR(J+3,KI) + BAR(J+6,KI)	HBELT
TR(J,K) = T1(J)	HBELT
ZR(J,K) = T1(J) + T2(J)	HBELT
S (J,2) = S(J,1)	HBELT
11 S (J,1) = SEGLP(J,KS) + ZR(J,K)	HBELT
CALL CROSS (WMEG(1,KS),V,T)	HBELT
IF (KE.EQ.0) GO TO 12	HBELT
CALL MAT31 (BD(7,KE),BAR(4,KI),T2)	HBELT
CALL DOT31 (D(1,1,KS),T2,T1)	HBELT
12 DO 13 J=1,3	HBELT
T(J) = T(J) + BAR(J+12,KI)	HBELT
13 E(J,3,K) = T1(J)	HBELT
CALL DOT31 (D(1,1,KS),T,V)	HBELT
DO 14 J=1,3	HBELT
14 V(J) = V(J) + SEGLV(J,KS)	HBELT
FB(K) = 0.0	HBELT
FP(K) = 0.0	HBELT
IF (K.EQ.K1) GO TO 20	HBELT
DO 15 J=1,3	HBELT
15 U(J,K-1) = S(J,1) - S(J,2)	HBELT
BL(K-1) = DSQRT(U(1,K-1)**2 + U(2,K-1)**2 + U(3,K-1)**2)	HBELT
DO 16 J=1,3	HBELT
16 U(J,K-1) = U(J,K-1)/BL(K-1)	HBELT
STRAIN = (BL(K-1)/BB(KNL-1)) - 1.0	HBELT
IF (STRAIN.LT.EPS(12)) STRAIN = 0.0	HBELT
NT = NL(2,KNL)	MISDOT
BLDOT = U(1,K-1)*(V(1)-R(1))	HBELT
* + U(2,K-1)*(V(2)-R(2))	HBELT
* + U(3,K-1)*(V(3)-R(3))	HBELT
STRDOT = (BB(KNL-1)*BLDOT-BL(K-1)*BBDOT(KNL-1))/BB(KNL-1)**2	HBELT
CALL FRCDFL (STRAIN,STRDOT,NT,0,FBK,ELOSS)	HBELT
CALL FRCDFL (STRAIN,STRDOT,NT,1,FBK,ELOSS)	HBELT
PTLOSS(1,K-1) = BB(KNL-1)*ELOSS	HBELT
FP(K-1) = FP*	HBELT
FB(K-1) = FBK	HBELT
IF (IND.NE.0) GO TO 20	HBELT
IF (K.NE.K1+1) GO TO 19	ENDPFX
BSF(1,NBSF) = STRAIN	ENDPFX
BSF(2,NBSF) = FBK	ENDPFX
19 IF (K.NE.K2) GO TO 20	ENDPFX
BSF(3,NBSF) = STRAIN	ENDPFX
BSF(4,NBSF) = FBK	ENDPFX
20 CONTINUE	HBELT
C	HBELT
C SECOND LOOP ON K	HBELT
C COMPUTE FCE(K),E1(K),E2(K),EDOT(K),FR(K),U1(KS),U2(KS))	HBELT
C NEED FB(K&K-1),U(K&K-1),ZR(K),E3(K)	HBELT
C	HBELT

	DO 30 K=K1,K2	HBELT
	KNL = KNLO + K	HBELT
	KI = NL(1,KNL)	HBELT
	KS = IABS(IBAR(1,KI))	HBELT
	IF (KS.GT.100) KS = MOD(KS,100)	HBELT
	DO 21 J=1,3	HBELT
	FCE(J,K) = 0.0	HBELT
	IF (K.NE.K2) FCE(J,K) = FB(K)*U(J,K)	BUTLER1
	IF (K.NE.K1) FCE(J,K) = FCE(J,K) - FB(K-1)*U(J,K-1)	EJTLER1
21	NT = IBAR(3,KI)	HBELT
	NF = NTAB(NT+5)	HBELT
	IF (NF.EQ.0 .AND. IND.EQ.0) GO TO 30	HBELT
	IF (IBAR(4,KI).EQ.0) GO TO 22	HBELT
	CALL DOT31 (D(1,1,KS),BAR(10,KI),T1)	HBELT
	GO TO 24	HBELT
22	DO 23 J=1,3	HBELT
	T1(J) = 0.0	HBELT
	IF (K.NE.K2) T1(J) = U(J,K)	HBELT
23	IF (K.NE.K1) T1(J) = T1(J) + U(J,K-1)	HBELT
24	CALL CROSS (T1,E(1,3,K),E(1,1,K))	HBELT
	CALL CROSS (E(1,3,K),E(1,1,K),E(1,2,K))	HBELT
	DO 25 J=1,3	HBELT
	EDOT(J,K) = DSQRT(E(1,J,K)**2 + E(2,J,K)**2 + E(3,J,K)**2)	HBELT
	DO 25 I=1,3	HBELT
25	E(I,J,K) = E(I,J,K)/EDOT(J,K)	HBELT
	CALL DOT31 (E(1,1,K),FCE(1,K),FR(1,K))	HBELT
30	CONTINUE	HBELT
31	CONTINUE	HBELT
	IF (NTP.LE.0) GO TO 41	HBELT
C		HBELT
C	SUM FCE,FR FOR TIE-POINTS	HBELT
C		HBELT
	KNL1 = KNLO + 2	HBELT
	KNL2 = KNLO + K2	HBELT
	DO 40 KNL=KNL1,KNL2	HBELT
	KI = NL(1,KNL)	HBELT
	KS = IABS(IBAR(1,KI))	HBELT
	IF (KS.LT.100) GO TO 40	HBELT
	KS1 = KS/100	HBELT
	KH = KNL - KNLO	HBELT
	MH = 0	HBELT
	DO 38 JNL=KNL1,KNL	HBELT
	K1 = NL(1,JNL-1)	HBELT
	KS = IABS(IBAR(1,KI))	HBELT
	IF (KS.LT.100) GO TO 38	HBELT
	KS2 = KS/100	HBELT
	IF (KS2.NE.KS1) GO TO 38	HBELT
	JH = JNL - KNLO	HBELT
	IF (MH.EQ.0) MH = JH	HBELT
	DO 37 J=1,3	HBELT
	IF (MH.EQ.JH) FCE(J,MH) = FCE(J,MH) + FCE(J,KH)	HBELT
37	FCE(J,JH) = FCE(J,MH)	HBELT

	CALL DOT31 (E(1,1,JH),FCE(1,JH),FR(1,JH))	HBELT
38	CONTINUE	HBELT
	IF (MH.EQ.0) GO TO 40	HBELT
	KI = NL(1,KNL)	HBELT
	IBAR(1,KI) = -IABS(IBAR(1,KI))	HBELT
	DO 39 J=1,3	HBELT
39	FCE(J,KH) = FCE(J,MH)	HBELT
	CALL DOT31 (E(1,1,KH),FCE(1,KH),FR(1,KH))	HBELT
40	CONTINUE	HBELT
C		HBELT
C	IF CALL IS FROM SUBROUTINE CONTECT,	HBELT
C	ADD FORCES (FCE) MODIFIED BY FRICTION TO U1,U2 ARRAYS.	HBELT
C		HBELT
41	IF (IND.NE.0) GO TO 52	HBELT
	K2 = 0	HBELT
	DO 51 JB=J1,J2	HBELT
	IF (NPTPLY(JB).LE.0) GO TO 51	HBELT
	K1 = K2 + 1	HBELT
	K2 = K2 + NPTPLY(JB)	HBELT
	DO 50 K=K1,K2	HBELT
	KNL = KNLO + K	HBELT
	KI = NL(1,KNL)	HBELT
	IF (IBAR(1,KI).LT.0) GO TO 50	HBELT
	KS = IBAR(1,KI)	HBELT
	IF (KS.GT.100) KS = MOD(KS,100)	HBELT
	NT = IBAR(3,KI)	HBELT
	NF = NTAB(NT+5)	HBELT
	IF (NF.EQ.0) GO TO 43	HBELT
	DO 42 J=1,3	HBELT
42	T1(J) = FR(J,K)	HBELT
	FR1 = TAB(NF+2)*DABS(T1(3))	HBELT
	FR2 = TAB(NF+4)*DABS(T1(3))	HBELT
	IF (DABS(T1(1)).GT.FR1) T1(1) = DSIGN(FR1,T1(1))	HBELT
	IF (DABS(T1(2)).GT.FR2) T1(2) = DSIGN(FR2,T1(2))	HBELT
	CALL MAT31 (E(1,1,K),T1,FCE(1,K))	HBELT
43	CALL CROSS (ZR(1,K),FCE(1,K),T2)	HBELT
	CALL MAT31 (D(1,1,KS),T2,T1)	HBELT
	DO 44 J=1,3	HBELT
	U1(J,KS) = U1(J,KS) + FCE(J,K)	HBELT
44	U2(J,KS) = U2(J,KS) + T1(J)	HBELT
50	CONTINUE	HBELT
51	CONTINUE	HBELT
52	KNLO = KNLO + K2	HBELT
	CALL ELTIME (2,38)	HBELT
	RETURN	HBELT
	END	HBELT

```

      SUBROUTINE HBPLAY                                         HBPLAY
C                                                REV III.5 10/17/85EDGE
      IMPLICIT REAL*8 (A-H,O-Z)                                HBPLAY
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,   HBPLAY
*      NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG        PAGE
      COMMON/CNTRSF/ PL(24,30),BELT(20,8),TPTS(6,8),BD(24,40)  EDGE
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30), HBPLAY
*      SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)          HBPLAY
      COMMON/HRNESS/ BAR(15,100),BB(100),BBDOT(100),PLOSS(2,100), HBPLAY
*      XLONG(20),HTIME(2),IBAR(5,100),NL(2,100),             HBPLAY
*      NPTSPB(20),NPTPLY(20),NTHRNS(20),NBLTPH(5)            HBPLAY
C      THIS COMMON/TEMPVS/ IS SHARED BY HPTURB, HBPLAY, HBELT AND HSETC. HBPLAY
      COMMON/TEMPVS/ B(3,3,3),S(3,3),T(3),R(3),V(3),T1(3),T2(3), HBPLAY
*      E(3,3,50),EDOT(3,50),FCE(3,50),FR(3,50),ZR(3,50).    HBPLAY
*      TR(3,50),U(3,50),PTLOSS(2,50),BL(50),FB(50),FP(50), HBPLAY
*      OLDBB(100),RHS(3,54),C(3,3,200),IJK(54,54)           HBPLAY
*      ,DMMY(29942)                                           80386
      IF (NHRNSS.LE.0) GO TO 99                                HBPLAY
C                                                                    HBPLAY
C      SAVE PREVIOUS NL,BB AND PLOSS ARRAYS.                  HBPLAY
C      USE IJK,OLDBB AND PTLOSS AS TEMP STORAGE.             HBPLAY
C                                                                    HBPLAY
      DO 10 I=1,100                                           HBPLAY
      IJK(I,1) = NL(1,I)                                       HBPLAY
      PTLOSS(I,1) = PLOSS(1,I)                                  HBPLAY
10     OLDBB(I) = BB(I)                                         HBPLAY
      JNL = 1                                                  HBPLAY
      J1 = 1                                                    HBPLAY
      J2 = 1                                                    HBPLAY
      LL = 0                                                    HBPLAY
      DO 90 NH=1,NHRNSS                                       HBPLAY
      IF (NBLTPH(NH).LE.0) GO TO 90                             HBPLAY
      J2 = J1 + NBLTPH(NH) - 1                                  HBPLAY
      DO 80 NB=J1,J2                                           HBPLAY
      LI = LL                                                  HBPLAY
      IF (NPTSPB(NB).LE.0) GO TO 80                             HBPLAY
      K2 = K1 + NPTSPB(NB) - 1                                  HBPLAY
      KB = 0                                                    HBPLAY
      DO 30 K=K1,K2                                            HBPLAY
      KB = KB + 1                                              HBPLAY
C                                                                    HBPLAY
C      HERE K IS INDEX OF ALL POINTS                           HBPLAY
C      KB IS INDEX OF POINTS ON A SINGLE BELT                  HBPLAY
C      LL IS INDEX OF ALL POINTS IN PLAY                       HBPLAY
C      JB IS INDEX OF PREVIOUS POINT ON BELT IN PLAY           HBPLAY
C                                                                    HBPLAY
      KS = IABS(IBAR(1,K))                                       HBPLAY
      IF (KS.GT.100) KS = MOD(KS,100)                            HBPLAY
      CALL DOT31 (D(1,1,KS),BAR(4,K),T1)                         HBPLAY
      CALL DOT31 (D(1,1,KS),BAR(7,K),T2)                         HBPLAY
      DO 11 J=1,3                                               HBPLAY
11     U(J,KB) = SEGLP(J,KS) + T1(J) + T2(J)                  HBPLAY

```



```

      IF (K.EQ.K1) GO TO 30
      LL = LL + 1
12   JJ = NL(1,LL)
      JB = JJ - K1 + 1
      DSS = 0.0
      DO 13 J=1,3
      ZR(J,KB) = U(J,KB) - U(J,JB)
13   DSS = DSS + ZR(J,KB)**2
      BL(LL) = DSQRT(DSS)
      IF (JJ.EQ.K1 .OR. IABS(IBAR(1,JJ)).GT.100) GO TO 30
      JS = IBAR(1,JJ)
      JE = IBAR(2,JJ)
      IF (JE.LE.0) GO TO 30
      CALL MAT31 (BD(7,JE),BAR(4,JJ),T2)
      CALL DOT31 (D(1,1,JS),T2,R)
      DPR = 0.0
      DO 17 J=1,3
17   DPR = DPR + R(J)*(ZR(J,KB)/BL(LL) - ZR(J,JB)/BL(LL-1))
      IF (DPR.LT.0.0) GO TO 30
      LL = LL - 1
      GO TO 12
30   NL(1,LL+1) = K
      L2 = L1 + 1
      LL = LL + 1
      L3 = LL-1
      DO 31 J=L2,LL
31   NL(2,J) = NTHRNS(NB)
      IF (XLONG(NB).EQ.0.0) GO TO 35
C
C   FIRST TIME IN ROUTINE, SET INITIAL BB ARRAY.
C   INPUT XLONG MUST BE NON-ZERO TO TRIGGER THIS TEST.
C
      XLG = 0.0
      DO 32 J=L2,L3
32   XLG = XLG + BL(J)
      XLG = 1.0 + XLONG(NB)/XLG
      DO 33 J=L2,L3
33   BB(J) = XLG*BL(J)
      XLONG(NB) = 0.0
      GO TO 52
C
C   DETERMINE IF NEW NL ARRAY IS DIFFERENT FROM PREVIOUS NL ARRAY.
C   IF SO, RECOMPUTE BB ELEMENTS FOR POINTS THAT ARE DIFFERENT.
C
35   IF (NL(1,L2).EQ.IJK(JNL,1)) GO TO 61
      WRITE (6,62)
62   FORMAT ('O LOGIC ERROR IN SUB HBPLAY. PROGRAM TERMINATED.')
      STOP 42
61   LTEST = 0
      M = L2
      N = JNL
36   IF (NL(1,M+1)-IJK(N+1,1)) 39,37,41

```

37	BB(M) = OLDBB(N)	HBPLAY
	PLOSS(1,M) = PTLOSS(N,1)	HBPLAY
38	M = M+1	HBPLAY
	N = N+1	HBPLAY
	IF (M-LL) 36,51,51	HBPLAY
C		HBPLAY
C	POINT M+1 IS NEW.	HBPLAY
C		HBPLAY
39	MO = M	HBPLAY
	NO = N	HBPLAY
	LTEST = 1	HBPLAY
40	M = M+1	HBPLAY
C		HBPLAY
C	MODIFY NEW POINT TO LIE IN BELT PLANE	CHGIII
C		CHGIII
	IP1 = N - 1	CHGIII
	IF (N.GT.JNL) GO TO 63	CHGIII
	IP1 = N	CHGIII
C	(IS THIRD POINT AVAILABLE FROM OLD POINTS IN PLAY?)	CHGIII
	IF (IJK(N+1,1).EQ.NL(1,LL)) GO TO 43	CHGIII
63	DO 64 I=1,3	CHGIII
	IP = IP1 + I - 1	CHGIII
C	(USE OLD POINTS IP = N-1,N,N+1 IF N > JNL	CHGIII
C	OR IP = N,N+1,N+2 IF N = JNL AND N+2 EXISTS)	CHGIII
	NI = IJK(IP,1)	CHGIII
	NSI = IABS(IBAR(1,NI))	NSFIX
	IF (NSI.GT.100) NSI = MOD(NSI,100)	NSFIX
	CALL DOT31 (D(1,1,NSI),BAR(4,NI),T1)	NSFIX
	CALL DOT31 (D(1,1,NSI),BAR(7,NI),T2)	NSFIX
	DO 64 J=1,3	CHGIII
64	S(J,I) = SEGLP(J,NSI)+ T1(J) + T2(J)	NSFIX
	DO 65 J=1,3	CHGIII
	S(J,3) = S(J,3) - S(J,2)	CHGIII
65	S(J,2) = S(J,2) - S(J,1)	CHGIII
C	(S(*,1) IS POINT P1 IN INERTIAL REFERENCE)	CHGIII
C	(S(*,2) IS VECTOR (P2-P1) IN INERTIAL REFERENCE)	CHGIII
C	(S(*,3) IS VECTOR (P3-P2) IN INERTIAL REFERENCE)	CHGIII
	CALL CROSS (S(1,3),S(1,2),T2)	CHGIII
	ABST = DSQRT(T2(1)**2 + T2(2)**2 + T2(3)**2)	CHGIII
	DO 66 J=1,3	CHGIII
66	T2(J) = T2(J)/ABST	CHGIII
C	(T2 IS T, THE NORMALIZED PLANE VECTOR IN INERTIAL REFERENCE)	CHGIII
	MI = NL(1,M)	CHGIII
	MS = IABS(IBAR(1,MI))	CHGIII
	IF (MS.GT.100) MS = MOD(MS,100)	CHGIII
	ME = IBAR(2,MI)	CHGIII
	CALL MAT31 (D(1,1,MS),T2,T1)	CHGIII
C	(T1 IS T IN ELLIPSOID REFERENCE OF NEW POINT M)	CHGIII
	D1 = T2(1)*S(1,1) + T2(2)*S(2,1) + T2(3)*S(3,1)	CHGIII
	D2 = T1(1)*BAR(7,MI) + T1(2)*BAR(8,MI) + T1(3)*BAR(9,MI)	CHGIII
	D3 = T2(1)*SEGLP(1,MS) + T2(2)*SEGLP(2,MS) + T2(3)*SEGLP(3,MS)	CHGIII
	DD = D1 - D2 - D3	CHGIII

```

C      (DD IS D, THE DISTANCE OF ELLIPSOID CENTER TO PLANE)
CALL MAT31 (BD(16,ME),T1,R)
BX = DD/(T1(1)*R(1) + T1(2)*R(2) + T1(3)*R(3))
D4 = T1(1)*BAR(4,MI) + T1(2)*BAR(5,MI) + T1(3)*BAR(6,MI)
DO 67 J=1,3
R(J) = BX*R(J)
C      (R IS S, THE CENTER OF THE ELLIPSE)
67 V(J) = BAR(J+3,MI) + (DD-D4)*T1(J)
C      (BAR(J+3,MI) IS P, THE NEW POINT TO BE ADDED)
C      (V IS Q, THE PROJECTION OF POINT P ONTO THE PLANE)
AX = DSQRT( (BX*DD-1.0) / (BX*DD-XDY(V,BD(7,ME),V)) )
DO 68 J=1,3
68 BAR(J+3,MI) = R(J) + AX*(V(J)-R(J))
C      (BAR(J+3,MI) IS R = S + A(Q - S), Q EXTENDED TO ELLIPSOID)
GO TO 43
C
C      POINT N+1 IS DROPPED.
C
C
41 MO = M
NO = N
LTEST = 1
42 N = N+1
43 IF (NL(1,M+1)-IJK(N+1,1)) 40,44,42
C
C      POINTS NO TO N+1 ARE BEING REPLACED WITH POINTS MO TO M+1.
C
44 SUMBL = 0.0
DO 45 J=MO,M
45 SUMBL = SUMBL + BL(J)
SUMPL = 0.0
SUMBB = 0.0
DO 46 J=NO,N
SUMPL = SUMPL + PTLOSS(J,1)
46 SUMBB = SUMBB + OLDBB(J)
RATPL = SUMPL/SUMBL
RATIO = SUMBB/SUMBL
DO 47 J=MO,M
PLOSS(1,J) = RATPL*BL(J)
47 BB(J) = RATIO*BL(J)
GO TO 38
51 JNL = N+1
IF (LTEST.EQ.0) GO TO 79
C
C      PRINT NEW POINT ARRAY IF DIFFERENT.
C
52 NPTS = LL - L1
USEC = 1000.0*TIME
WRITE (6,53) USEC,NH,NB,NPTS,NTHRNS(NB)
53 FORMAT ('0 HBPLAY TIME =',F10.3,' MSEC. NH,NB,NPTS NT=',4I6)
WRITE (6,54) (NL(1,J),J=L2,LL)
54 FORMAT (' NL(1)=',15I8/(8X,15I8))
WRITE (6,55) (BB(J),J=L2,L3)

```

55	FORMAT (' BB =',6X,14F8.3/(6X,15F8.3))	HBPLAY
79	K1 = K2 + 1	HBPLAY
80	NPTPLY(NB) = LL - L1	HBPLAY
	J1 = J2 + 1	HBPLAY
90	CONTINUE	HBPLAY
99	RETURN	HBPLAY
	END	HBPLAY

```

SUBROUTINE HEDING (LINES,LPP)
C
REV IV 02/01/88MISDOT
IMPLICIT REAL*8 (A-H,O-Z)
COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,
* NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG
COMMON/JBARTZ/ MNPL( 30),MNBLT( 8),MNSEG( 30),MNBAG( 6),
* MPL(3,5,30),MBLT(3,5,8),MSEG(3,5,30),MBAG(3,10,6),
* NTPL( 5,30),NTBLT( 5,8),NTSEG( 5,30)
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)
REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT
LOGICAL*1 CGS,JS
COMMON/FORCES/PSF(7,70),BSF(4,20),SSF(10,40),BAGSF(3,20),
* PRJNT(7,30),NPANEL(5),NPSF,NBSF,NSSF,NBGSF
COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),
* UNITL,UNITM,UNITT,GRAVTY(3),TWOPI
COMMON/RSAVE/ XSG(3,20,3),DPMI(3,3,30),LPMI(30),
* NSG(9),MSG(20,9),MCG,MCGIN(24,5),KREF(20,9)
COMMON/DAMPER/ APSDM(3,20),APSDN(3,20),ASD(5,20),MSDM(20),MSDN(20)
COMMON/HRNESS/ 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)
C NOTE: SUBROUTINES POSTPR & HEDING SHARE THIS COMMON/TEMPVS/.
C SEE COMMENT IN POSTPR ABOUT FIRST DIMENSION OF PLDATA.
REAL HEAD,PHED,BLANK,PLDATA,USEC,ZTTH,AHED,AHEAD,GHED,ZZZ
COMMON/TEMPVS/ TDATA(14,65),HEAD(20),NOPL(150),MOPL(150),
* M1PL(150),PLDATA(97,20),USEC(45),ZZZ(1000,25),ZTTH(14,45,65)
* ,IDMMY
LOGICAL LOLD ,LNEW
DIMENSION PHED(5),HEDJ(4,2),HEADJJ(4,2),HEADR(20)
DATA HEDJ/8HIPIN FL,8HEXURE A,8HZIMUTH ,8HTORSION ,
* 8HIEULER ,8HPREC. N,8HUTATION ,8H SPIN /
DIMENSION AHED(5,2),AHEAD(5,20),GHED(2)
DATA AHED/4H IN ,4H ,4H REF,4HEREN,4HCE ,
* 4H AC,4HCELE,4HROME,4HTER ,4H /
DATA GHED/4H(0G),4H(1G)/
DATA BLANK/4H /
DATA PHED/4HSPRF,4HPNL1,4HPNL2,4HPNL3,4HPNL4/
NPRT4 = NPRT(4) + 4
IF (NPRT4.LE.0 .OR. NPRT4.GT.8) STOP 40
GO TO (11,11,82,12,12,11,11,12) , NPRT4
11 LOLD = .FALSE.
LNEW = .TRUE.
GO TO 13
12 LOLD = .TRUE.
LNEW = .FALSE.
13 MT = 20
NLINES = MOD(LINES-1,LPP)+1
XPAGE = 0.01*FLOAT((LINES + LPP-1)/LPP)
C
C NOTE: MT WILL BE THE PAGE OR OUTPUT UNIT COUNTER

```

C	NT WILL BE THE ACTUAL OUTPUT UNIT NUMBER	HEDING
C	IT WILL BE THE INDEX TO THE DATA ARRAY	HEDING
C	NLINES WILL BE THE NUMBER OF LINES TO BE PRINTED	HEDING
C		HEDING
C		HEDING
C	EVERY LPP LINES PRINT HEADINGS FOR 9 TYPES OF OUTPUT ABOVE.	WINDOP
C		HEDING
	DO 20 K=1,9	WINDOP
	IF (NSG(K).LE.0) GO TO 20	HEDING
	KSG = NSG(K)	HEDING
	IF (K.EQ.9) GO TO 455	WINDOP
	J3 = 3	HEDING
	IF (K.EQ.7) J3 = 2	HEDING
	DO 19 J1=1,KSG,J3	HEDING
	MT = MT + 1	HEDING
	NT = MT	HEDING
	IF (LNEW) NT = 6	HEDING
	IT = MT - 20	HEDING
	PAGE = FLOAT(MT) + XPAGE	HEDING
C	P & E PRINTER CARRIAGE CONTROL	PECONV
C	CALL CARCON(NT,1)	80386
	IF (NT.EQ.6) WRITE(NT,121) DATE,BLANK,NPG	PAGE
	IF (NT.NE.6) WRITE(NT,121) DATE	PAGE
	IF (NT.EQ.6) NPG=NPG+1	PAGE
	WRITE (NT,21) COMENT,PAGE,VPSTTL,BDYTTL	PAGE
	IF (K.EQ.1) WRITE (NT,22)	TTHKREF
	IF (K.EQ.2) WRITE (NT,23) UNITL,UNITT	TTHKREF
	IF (K.EQ.3) WRITE (NT,24) UNITL	TTHKREF
	IF (K.EQ.4) WRITE (NT,25) UNITT	TTHKREF
	IF (K.EQ.5) WRITE (NT,26) UNITT	TTHKREF
	IF (K.EQ.6) WRITE (NT,27)	TTHKREF
	IF (K.EQ.7) WRITE (NT,28)	HEDING
	IF (K.EQ.8) WRITE(NT,200) UNITM	TTHKREF
	J2 = MINO(J1+J3-1,KSG)	HEDING
	DO 14 J=J1,J2	HEDING
	KK = MSG(J,K)	HEDING
	HEAD(J) = SEG(IABS(KK))	ACCEL
	IF ((K.LT.7).OR.(K.EQ.8)) GO TO 214	TTHKREF
	KK = IABS(KK)	HEDING
	HEAD(J) = JOINT(KK)	HEDING
	JJ2 = J-J1+1	HEDING
	K2 = 1	HEDING
	IF (MSG(J,K).LT.0) K2 = 2	HEDING
	DO 35 K1=1,4	HEDING
	35 HEADJJ(K1,JJ2) = HEDJ(K1,K2)	HEDING
	GO TO 14	TTHKREF
	214 IF (MSG(J,K).LT.0) GOTO 302	ACCEL
	IF (KREF(J,K).EQ.0) HEADR(J)=SEG(NVEH)	ACCEL
	IF (K.EQ.8) HEADR(J)=SEG(NGRND)	TTHKREF
	IF (K.EQ.1 .OR. K.EQ.4) HEADR(J)=SEG(KK)	TTHKREF
	IF (KREF(J,K).NE.0) HEADR(J)=SEG(KREF(J,K))	TTHKREF
	DO 301 II=1,5	ACCEL

301	AHEAD(II,J)=AHED(II,1)	ACCEL
	AHEAD(2,J)=HEADR(J)	ACCEL
	GOTO 14	ACCEL
302	HEADR(J)=SEG(IABS(KK))	ACCEL
	DO 303 II=1,4	ACCEL
303	AHEAD(II,J)=AHED(II,2)	ACCEL
	AHEAD(5,J)=GHED(KREF(J,K)+1)	ACCEL
14	CONTINUE	HEDING
	IF (K.LE.3) WRITE (NT,29) (BLANK,(XSG(I,J,K),I=1,3),J=J1,J2)	HEDING
	IF (K.LE.6) WRITE (NT,30) (BLANK,MSG(J,K),HEAD(J),J=J1,J2)	HEDING
	IF (K.EQ.8) WRITE (NT,30) (BLANK,MSG(J,K),HEAD(J),J=J1,J2)	WINDOP
	IF (K.LE.6 .OR. K.EQ.8) WRITE (NT,230)	ACCEL
*	(BLANK,(AHEAD(II,J),II=1,5),J=J1,J2)	ACCEL
	IF ((K.LE.5).OR.(K.EQ.8)) WRITE (NT,31) (BLANK,J=J1,J2)	WINDOP
	IF (K.EQ.6) WRITE (NT,32) (BLANK,J=J1,J2)	HEDING
	IF ((K.LT.7).OR.(K.EQ.8)) GOTO 15	WINDOP
	WRITE (NT,33) (BLANK,MSG(J,K),HEAD(J),J=J1,J2)	HEDING
	WRITE (NT,36) (BLANK,UNITL,UNITM,J=J1,J2)	HEDING
	WRITE (NT,37) (BLANK,(HEADJJ(K1,J),K1=1,4),J=1,JJ2)	HEDING
15	WRITE (NT,38)	HEDING
	IF (.NOT.LNEW) GO TO 19	HEDING
	IF (K.EQ.7) GO TO 17	HEDING
	JJ = 4*(J2-J1+1)	HEDING
	DO 16 I=1,NLINES	HEDING
16	WRITE (NT,39) USEC(I),(ZTTH(J,I,IT),J=1,JJ)	HEDING
	GO TO 19	HEDING
17	JJ = 7*(J2-J1+1)	HEDING
	DO 18 I=1,NLINES	HEDING
18	WRITE (NT,40) USEC(I),(ZTTH(J,I,IT),J=1,JJ)	HEDING
19	CONTINUE	HEDING
	GO TO 20	CHGIII
C		CHGIII
C	PRINT HEADING FOR JOINT FORCES & TORQUES	CHGIII
C		CHGIII
455	CONTINUE	CHGIII
	DO 860 II=1,KSG	CHGIII
	IF(KREF(II,K).EQ.0) KRF = NVEH	TTHKREF
	IF(KREF(II,K).NE.0) KRF = KREF(II,K)	TTHKREF
	JRF = MSG(II,9)	WINDOP
	MT = MT + 1	CHGIII
	NT = MT	CHGIII
	IF (LNEW) NT = 6	CHGIII
C	P & E CARRIAGE CONTROL	PECONV
C	CALL CARCON(NT,1)	80386
	IT = MT - 20	CHGIII
	PAGE = FLOAT (MT) + XPAGE	CHGIII
	IF (NT.EQ.6) WRITE(NT,121) DATE,BLANK,NPG	PAGE
	IF (NT.NE.6) WRITE(NT,121) DATE	PAGE
	IF (NT.EQ.6) NPG=NPG+1	PAGE
	WRITE (NT,21) COMENT,PAGE,VPSTTL,BDYTTL	PAGE
	WRITE (NT,850) JOINT(JRF),SEG(JRF+1),SEG(KRF)	OUT385
	WRITE (NT,38)	CHGIII

' WRITE (NT,851) UNITM,UNITL,UNITM	CHGIII
WRITE (NT,852)	CHGIII
WRITE (NT,38)	CHGIII
IF (.NOT.LNEW) GO TO 857	CHGIII
DO 858 JK=1,NLINES	CHGIII
WRITE (NT,856) USEC(JK),(ZTTH(J,JK,IT),J=1,6)	CHGIII
858 CONTINUE	CHGIII
857 CONTINUE	CHGIII
850 FORMAT(' '/47X,	TTHKREF
* A4,' JOINT FORCES & TORQUES ON ',A4,' IN ',A4,' REFERENCE')	OUT385
851 FORMAT(4X,4HTIME,7X,13HJOINT FORCE (,A4,7H 10**2),10X,	CHGIII
*14HJOINT TORQUE (,A4,1H-,A4,7H 10**2))	CHGIII
852 FORMAT(3X,6H(MSEC),8X,1HX,8X,1HY,8X,1HZ,14X,1HX,11X,1HY,11X,1HZ)	CHGIII
856 FORMAT(F9.3,3X,3F9.3,3X,3(2X,D10.3))	CHGIII
860 CONTINUE	CHGIII
20 CONTINUE	HEDING
121 FORMAT('1',18X,'DATE:',3X,4A4,80X,'PAGE',I5)	PAGE
21 FORMAT(8X,'RUN DESCRIPTION:',3X,20A4/27X,20A4,'PAGE:',F6.2/	PAGE
* 3X,'VEHICLE DECELERATION:',3X,20A4/	HEDING
* 11X,'CRASH VICTIM:',3X,5A4)	HEDING
22 FORMAT(' '47X,	TTHKREF
* 'POINT TOTAL ACCELERATION (G'S)'/)	TTHKREF
23 FORMAT(' '47X,	TTHKREF
* 'POINT REL. VELOCITY (' ,A4,'/' ,A4,')'/)	TTHKREF
24 FORMAT(' '47X,	TTHKREF
* 'POINT REL. LINEAR DISPLACEMENT (' ,A4,')'/)	TTHKREF
25 FORMAT(' '/47X,	TTHKREF
* 'SEGMENT ANGULAR ACCELERATION (REV/' ,A4,'**2)'/)	TTHKREF
26 FORMAT(' '/47X,	TTHKREF
* 'SEGMENT REL. ANGULAR VELOCITY (REV/' ,A4,')'/)	TTHKREF
27 FORMAT(' '/47X,	TTHKREF
* 'SEGMENT REL. ANGULAR DISPLACEMENT (DEG)'/)	TTHKREF
28 FORMAT(' '/47X,'JOINT PARAMETERS'//)	TTHKREF
200 FORMAT(' '/47X,'SEGMENT WIND FORCE (' ,A4,')'//)	TTHKREF
29 FORMAT(9X,3(A4,3X,'POINT (' ,F6.2,' ,',F6.2,' ,',F6.2,') ON '))	HEDING
30 FORMAT(' ' ,3(A4,9X,'SEGMENT NO.',I3,' - ',A4,5X))	TTHKREF
230 FORMAT(' TIME ',3(A4,9X,5A4,6X))	ACCEL
31 FORMAT(' (MSEC)',3(A4,5X,'X',8X,'Y',8X,'Z',7X,'RES',1X))	HEDING
32 FORMAT(' (MSEC)',3(A4,4X,'YAW',5X,'PITCH',5X,'ROLL',5X,'RES '))	HEDING
33 FORMAT(9X,2(A1,21X,'JOINT NO.',I3,' - ',A4,20X))	HEDING
36 FORMAT(' TIME ',2(A1,'STATE',5X,'JOINT ANGLES (DEG)',8X,	HEDING
* 'TOTAL TORQUE ('.2A4,') '))	HEDING
37 FORMAT(' (MSEC)',2(A1,4A8,4X,'SPRING VISCOUS RES. '))	HEDING
38 FORMAT(1X)	HEDING
39 FORMAT(F9.3,3(3X,4F9.3))	HEDING
40 FORMAT(F9.3,2(F5.0,3F9.3,2X,3F9.3))	HEDING
C	ATBIII
C PRINT BODY PROPERTIES CONTROLLED BY H.10 CARDS	WINDOP
C	ATBIII
IF (MCG.EQ.0) GO TO 131	ATBIII
DO 130 NCG=1,MCG	ATBIII
MT = MT +1	ATBIII

	NT = MT	ATBIII
	IF (LNEW) NT = 6	ATBIII
C	P & E CARRIAGE CONTROL	PECONV
C	CALL CARCON(NT,1)	80386
	IT = MT - 20	ATBIII
	PAGE = FLOAT(MT) + XPAGE	ATBIII
	IF (NT.EQ.6) WRITE(NT,121) DATE, BLANK, NPG	PAGE
	IF (NT.NE.6) WRITE(NT,121) DATE	PAGE
	IF (NT.EQ.6) NPG=NPG+1	PAGE
	WRITE (NT,21) COMENT, PAGE, VPSTTL, BDYTTL	PAGE
	M = MCGIN(1, NCG)	ATBIII
	WRITE (NT,132) M, SEG(M)	ATBIII
	N = MCGIN(2, NCG)	ATBIII
	WRITE (NT,133) (MCGIN(I+2, NCG), I-1, N)	ATBIII
	WRITE (NT,38)	ATBIII
	WRITE (NT,134) UNITL, UNITM, UNITT, UNITL, UNITM, UNITT, UNITM, UNITL	KINETIC
	WRITE (NT,38)	ATBIII
	IF (.NOT.LNEW) GO TO 130	ATBIII
	DO 129 I=1, N LINES	ATBIII
	129 WRITE (NT,135) USEC(I), (ZTTH(J, I, IT), J=1, 12)	KINETIC
	130 CONTINUE	ATBIII
	131 CONTINUE	ATBIII
	132 FORMAT(' ', 47X, 39H BODY PROPERTIES - REFERENCE SEGMENT NO.,	TTHKREF
	* I3, 2H (, A4, 1H))	ATBIII
	133 FORMAT(15X, 21H INCLUDED SEGMENT NOS: , 20I3)	ATBIII
	134 FORMAT(14X, 17H CENTER OF GRAVITY, 13X, 15H LINEAR MOMENTUM, 17X,	KINETIC
	* 16H ANGULAR MOMENTUM, 18X, 14H KINETIC ENERGY/	KINETIC
	* 4X, 4H TIME, 11X, 1H (, A4, 1H), 21X, 1H (, A4, 1H- , A4, 1H), 19X,	KINETIC
	* 1H (, A4, 1H- , A4, 1H- , A4, 1H), 20X, 1H (, A4, 1H- , A4, 1H)/	MISC
	* 3X, 6H (MSEC), 5X, 1HX, 7X, 1HY, 7X, 1HZ,	KINETIC
	* 2(10X, 1HX, 10X, 1HY, 10X, 1HZ), 6X, 6H LINEAR, 5X,	KINETIC
	* 7H ANGULAR, 5X, 5H TOTAL)	KINETIC
	135 FORMAT(F9.3, 3F8.3, 9(1X, D10.3))	KINETIC
C		HEDING
C	PLANE FORCES HEADINGS	HEDING
C		HEDING
	MPSF = 0	HEDING
	IF (NPL.EQ.0) GO TO 52	HEDING
	IF (NPRT(18).EQ.1.OR.NPRT(18).EQ.7) GO TO 52	VARTH
	IF (NPRT(18).EQ.10.OR.NPRT(18).EQ.11) GO TO 52	VARTH
	IF (NPRT(18).GE.14) GO TO 52	VARTH
	DO 42 J=1, NPL	HEDING
	IF (MNPL(J).EQ.0) GO TO 42	HEDING
	KPL = MNPL(J)	HEDING
	DO 41 I=1, KPL	HEDING
	MPSF = MPSF + 1	HEDING
	NOPL(MPSF) = J	HEDING
	IF (MPL(3, I, J).LT.0) M1PL(MPSF) = MPL(2, I, J)	CHGIII
	IF (MPL(3, I, J).GE.0) M1PL(MPSF) = MPL(1, I, J)	CHGIII
	41 MOPL(MPSF) = MPL(2, I, J)	HEDING
	42 CONTINUE	HEDING
	IF (MPSF.EQ.0) GO TO 52	HEDING

```

DO 44 J1=1,MPSF,2
J2 = MINO(J1+1,MPSF)
MT = MT + 1
NT = MT
IF (LNEW) NT = 6
C P & E CARRIAGE CONTROL
C CALL CARCON(NT,1)
IT = MT - 20
PAGE = FLOAT(MT) + XPAGE
IF (NT.EQ.6) WRITE(NT,121) DATE,BLANK,NPG
IF (NT.NE.6) WRITE(NT,121) DATE
IF (NT.EQ.6) NPG=NPG+1
WRITE (NT,21) COMENT,PAGE,VPSTTL,BDYTTL
WRITE (NT,45)
N1 = NOPL(J1)
N2 = NOPL(J2)
M1 = MOPL(J1)
M2 = MOPL(J2)
MM1 = M1PL(J1)
MM2 = M1PL(J2)
IF (J1.EQ.J2) WRITE (NT,46)
* BLANK,N1,( PLTTL(I,N1),I=1,5),M1,SEG(M1)
IF (J1.NE.J2) WRITE (NT,46)
* BLANK,N1,( PLTTL(I,N1),I=1,5),M1,SEG(M1),
* BLANK,N2,( PLTTL(I,N2),I=1,5),M2,SEG(M2)
WRITE (NT,47) (BLANK,UNITL,J=J1,J2)
IF (J1.EQ.J2) WRITE (NT,48) BLANK,SEG(MM1)
IF (J1.NE.J2) WRITE (NT,448) BLANK,SEG(MM1),BLANK,SEG(MM2)
WRITE (NT,49) (BLANK,UNITL,UNITM,UNITM,UNITM,J=J1,J2)
WRITE (NT,38)
IF (.NOT.LNEW) GO TO 44
JJ = 7*(J2-J1+1)
DO 43 I=1,NLINES
43 WRITE (NT,50) USEC(I),(ZTTH(J,I,IT),J=1,JJ)
44 CONTINUE
45 FORMAT(27X,'CONTACT FORCES - SEGMENT PANELS VS. SEGMENTS' )
46 FORMAT(' '/8X,2(A4,' PANEL',I3,' (' ,5A4,' ) VS. SEGMENT',I3,
* (' ,A4,' ) ) )
47 FORMAT(' ',8X,A4,'DEFL- NORMAL FRICTION RESULTANT CONTACT LOCATHEDING
*ION (' ,A4,' )',A2,'DEFL- NORMAL FRICTION RESULTANT CONTACT LOCATHEDING
*ION (' ,A4,' )')
48 FORMAT(' TIME',A4,'ECTION FORCE FORCE FORCE (' ,A4 CHGIII
*,' REFERENCE') )
448 FORMAT(' TIME',A4,'ECTION FORCE FORCE FORCE (' ,A4 CHGIII
*,' REFERENCE'),' ,2X,A4,'ECTION FORCE FORCE FORCE (' ,A4 CHGIII
*,' REFERENCE') )
49 FORMAT(' (MSEC)',2(A3,'(' ,A4,' )',2X,'(' ,A4,' )',4X,'(' ,A4,' )',3X,
* (' ,A4,' , X Y Z ') )
50 FORMAT(F9.3,2(F9.3,3F9.2,3F8.3) )
51 FORMAT(3X,'(MSEC)',4(A1,9X,'X',8X,'Y',8X,'Z',1X))
C
C BELT FORCES HEADINGS

```

C		HEDING
	52 MBSF = 0	HEDING
	IF (NPRT(18).EQ.2.OR.NPRT(18).GE.13) GO TO 83	VARTTH
	IF (NPRT(18).GE.7.AND.NPRT(18).LE.9) GO TO 83	VARTTH
	IF (NBLT.EQ.0) GO TO 83	HEDING
	DO 54 J=1,NBLT	HEDING
	IF (MNBLT(J).EQ.0) GO TO 54	HEDING
	MBSF = MBSF+1	HEDING
	NOPL(MBSF) = J	HEDING
	MOPL(MBSF) = MBLT(2,1,J)	HEDING
	54 CONTINUE	HEDING
	IF (MBSF.EQ.0) GO TO 83	HEDING
	DO 56 J1=1,MBSF,2	HEDING
	J2 = MINO(J1+1,MBSF)	HEDING
	MT = MT + 1	HEDING
	NT = MT	HEDING
	IF (LNEW) NT = 6	HEDING
C	P & E CARRIAGE CONTROL	PECONV
C	CALL CARCON(NT,1)	80386
	IT = MT - 20	HEDING
	PAGE = FLOAT(MT) + XPAGE	HEDING
	IF (NT.EQ.6) WRITE(NT,121) DATE,BLANK,NPG	PAGE
	IF (NT.NE.6) WRITE(NT,121) DATE	PAGE
	IF (NT.EQ.6) NPG=NPG+1	PAGE
	WRITE (NT,21) COMENT, PAGE,VPSTTL,BDYTTL	PAGE
	WRITE (NT,57)	HEDING
	N1 = NOPL(J1)	HEDING
	N2 = NOPL(J2)	HEDING
	M1 = MOPL(J1)	HEDING
	M2 = MOPL(J2)	HEDING
	IF (J1.EQ.J2) WRITE (NT,58)	HEDING
*	BLANK,N1,(BLTTTL(I,N1),I=1,5),M1,SEG(M1)	HEDING
*	IF (J1.NE.J2) WRITE (NT,58)	HEDING
*	BLANK,N1,(BLTTTL(I,N1),I=1,5),M1,SEG(M1),	HEDING
*	BLANK,N2,(BLTTTL(I,N2),I=1,5),M2,SEG(M2)	HEDING
	WRITE (NT,59) (BLANK,J=J1,J2)	HEDING
	WRITE (NT,60) (BLANK,J=J1,J2)	HEDING
	WRITE (NT,61) (BLANK,UNITL,UNITL,UNITM,UNITL,UNITL,UNITM,J=J1,J2)	HEDING
	WRITE (NT,38)	HEDING
	IF (.NOT.LNEW) GO TO 56	HEDING
	JJ = 4*(J2-J1+1)	HEDING
	DO 55 I=1,NLINES	HEDING
	55 WRITE (NT,62) USEC(I),(ZTTH(J,I,IT),J=1,JJ)	HEDING
	56 CONTINUE	HEDING
	57 FORMAT('0',26X,'CONTACT FORCES - BELTS VS. SEGMENTS')	HEDING
	58 FORMAT(' ',7X,2(A4,' BELT',I3,' (' ,5A4,') VS. SEGMENT',I3,	HEDING
	* ' (' ,A4,') ')	HEDING
	59 FORMAT(' ',2X,2(A4,11X,'ANCHOR POINT A',14X,'ANCHOR POINT B'))	HEDING
	60 FORMAT(4X,'TIME',2(A4,5X,'STRAIN',7X,'FORCE',12X,	HEDING
	* 'STRAIN',7X,'FORCE',3X))	HEDING
	61 FORMAT(3X,'(MSEC)',2(A4,2X,' (' ,A4,' /',A4,')',4X,' (' ,A4,')',9X,	HEDING
	* ' (' ,A4,' /',A4,')',4X,' (' ,A4,')',3X))	HEDING

62	FORMAT(F9.3,4(F15.6,F12.2,3X))	HEDING
C		HEDING
C	HARNES BELT ENDPOINTS FORCES HEADINGS	HEDING
C		HEDING
83	IF (NHRNSS.LE.0) GO TO 91	HEDING
	MBSF = 0	HEDING
	IF (NPRT(18).EQ.3.OR.NPRT(18).EQ.11) GO TO 91	VARTH
	IF (NPRT(18).EQ.9.OR.NPRT(18).EQ.8) GO TO 91	VARTH
	IF (NPRT(18).EQ.13.OR.NPRT(18).EQ.14) GO TO 91	VARTH
	IF (NPRT(18).EQ.16) GO TO 91	VARTH
	J1 = 1	HEDING
	K1 = 1	HEDING
	DO 85 I=1,NHRNSS	HEDING
	IF (NBLTPH(I).LE.0) GO TO 85	HEDING
	J2 = J1 + NBLTPH(I) - 1	HEDING
	DO 84 J=J1,J2	HEDING
	MBSF = MBSF + 1	HEDING
	IF (NPTSPB(J).LE.0) GO TO 84	HEDING
	K2 = K1 + NPTSPB(J) - 1	HEDING
	NOPL(2*MBSF-1) = J	HEDING
	NOPL(2*MBSF) = I	HEDING
	MOPL(2*MBSF-1) = K1	HEDING
	MOPL(2*MBSF) = K2	HEDING
	K1 = K2 + 1	HEDING
84	CONTINUE	HEDING
	J1 = J2 + 1	HEDING
85	CONTINUE	HEDING
	DO 87 J1=1,MBSF,2	HEDING
	J2 = MINO(J1+1,MBSF)	HEDING
	MT = MT + 1	HEDING
	NT = MT	HEDING
	IF (LNEW) NT = 6	HEDING
C	P & E CARRIAGE CONTROL	PECONV
C	CALL CARCON(NT,1)	80386
	IT = MT - 20	HEDING
	PAGE = FLOAT(MT) + XPAGE	HEDING
	IF (NT.EQ.6) WRITE(NT,121) DATE,BLANK,NPG	PAGE
	IF (NT.NE.6) WRITE(NT,121) DATE	PAGE
	IF (NT.EQ.6) NPG=NPG+1	PAGE
	WRITE (NT,21) COMENT,PAGE,VPSTTL,BDYTTL	PAGE
	WRITE (NT,88)	HEDING
	WRITE (NT,89) (BLANK,NOPL(2*J-1),NOPL(2*J),J=J1,J2)	HEDING
	WRITE (NT,90) (BLANK,MOPL(2*J-1),MOPL(2*J),J=J1,J2)	HEDING
	WRITE (NT,60) (BLANK,J-1,J2)	HEDING
	WRITE (NT,61) (BLANK,UNITL,UNITL,UNITM,UNITL,UNITL,UNITM,J=J1,J2)	HEDING
	WRITE (NT,38)	HEDING
	IF (.NOT.LNEW) GO TO 87	HEDING
	JJ = 4*(J2-J1+1)	HEDING
	DO 86 I=1,NLINES	HEDING
86	WRITE (NT,62) USEC(I),(ZTTH(J,I,IT),J=1,JJ)	HEDING
87	CONTINUE	HEDING
88	FORMAT('0',26X,'HARNES SYSTEM BELT ENDPOINT FORCES')	HEDING

	89	FORMAT(9X,2(A4,11X,'BELT NO.',14,' OF HARNESS NO.',13,15X))	HEDING
	90	FORMAT(9X,2(A4,6X,'POINT NO.',15,16X,'POINT NO.',15,6X))	HEDING
C			HEDING
C		SPRING DAMPER FORCES HEADINGS	HEDING
C			HEDING
	91	IF (NSD.LE.0) GO TO 63	HEDING
		IF (NPRT(18).EQ.4.OR NPRT(18).EQ.9) GO TO 63	VARTTH
		IF (NPRT(18).GE.12) GO TO 63	VARTTH
		DO 94 J1=1,NSD,4	HEDING
		J2 = MINO(J1+3,NSD)	HEDING
		MT = MT + 1	HEDING
		NT = MT	HEDING
		IF (LNEW) NT = 6	HEDING
C		P & E CARRIAGE CONTROL	PECONV
C		CALL CARCON(NT,1)	80386
		IT = MT - 20	HEDING
		PAGE = FLOAT(MT) + XPAGE	HEDING
		IF (NT.EQ.6) WRITE(NT,121) DATE,BLANK,NPG	PAGE
		IF (NT.NE.6) WRITE(NT,121) DATE	PAGE
		IF (NT.EQ.6) NPG=NPG+1	PAGE
		WRITE (NT,21) COMENT,PAGE,VPSTTL,BDYTTL	PAGE
		WRITE (NT,95) (BLANK,J,J=J1,J2)	HEDING
		DO 92 J=J1,J2	HEDING
		M1 = MSDM(J)	HEDING
		N1 = MSDN(J)	HEDING
C		POSSIBLE OVERFLOW INTO NOPL ARRAY IS INTENTIONAL.	HEDING
		HEAD(2*J-1) = SEG(M1)	HEDING
	92	HEAD(2*J) = SEG(N1)	HEDING
		WRITE (NT,96) (BLANK,MSDM(J),HEAD(2*J-1),MSDN(J),HEAD(2*J),J-J1,J2)	HEDING
		WRITE (NT,97) (BLANK,J=J1,J2)	HEDING
		WRITE (NT,98) (BLANK,UNITL,UNITM,J=J1,J2)	HEDING
		WRITE (NT,38)	HEDING
		IF (.NOT.LNEW) GO TO 94	HEDING
		JJ = 2*(J2-J1+1)	HEDING
		DO 93 I=1,NLINES	HEDING
	93	WRITE (NT,99) USEC(I),(ZTTH(J,I,IT),J=1,JJ)	HEDING
	94	CONTINUE	HEDING
	95	FORMAT('0',26X,'SPRING DAMPER FORCES'/	HEDING
		* 9X,4(A3,3X,'SPRING DAMPER NO.',13,4X))	HEDING
	96	FORMAT(9X,4(A3,'SEG',13,'(',A4,') - SEG',13,'(',A4,')'))	HEDING
	97	FORMAT(4X,'TIME',1X,4(A3,5X,'LENGTH',7X,'FORCE',4X))	HEDING
	98	FORMAT(3X,'(MSEC)',4(A3,5X,'(',A4,')',6X,'(',A4,')',4X))	HEDING
	99	FORMAT (F9.3,4(F14.3,F12.2,4X))	HEDING
C			HEDING
C		SEGMENT FORCES HEADINGS	HEDING
C			HEDING
	63	MSSF = 0	HEDING
		IF (NPRT(18).EQ.5.OR.NPRT(18).EQ.13) GO TO 161	VARTTH
		IF (NPRT(18).EQ.10.OR.NPRT(18).EQ.11) GO TO 161	VARTTH
		IF (NPRT(18).GE.15) GO TO 161	VARTTH
		DO 65 J=1,NSEG	HEDING
		IF (MNSEG(J).EQ.0) GO TO 65	HEDING

	LSEG = MNSEG(J)	HEDING
	DO 64 I=1,LSEG	HEDING
	MSSF = MSSF+1	HEDING
	NOPL(MSSF) = J	HEDING
64	MOPL(MSSF) = MSEG(2,I,J)	HEDING
65	CONTINUE	HEDING
	IF (MSSF.EQ.0) GO TO 70	HEDING
	DO 67 J=1,MSSF	HEDING
	MT = MT + 1	HEDING
	NT = MT	HEDING
	IF (LNEW) NT = 6	HEDING
C	P & E CARRIAGE CONTROL	PECONV
C	CALL CARCON(NT,1)	80386
	IT = MT - 20	HEDING
	PAGE = FLOAT(MT) + XPAGE	HEDING
	IF (NT.EQ.6) WRITE(NT,121) DATE,BLANK,NPG	PAGE
	IF (NT.NE.6) WRITE(NT,121) DATE	PAGE
	IF (NT.EQ.6) NPG=NPG+1	PAGE
	WRITE (NT,21) COMENT,PAGE,VPSTTL,BDYTTL	PAGE
	N1 = NOPL(J)	HEDING
	M1 = MOPL(J)	HEDING
	WRITE (NT,68) N1,SEG(N1),M1,SEG(M1),UNITL,N1,M1	HEDING
	* ,UNITL,UNITM,UNITM,UNITM	HEDING
	IF (.NOT.LNEW) GO TO 67	HEDING
	DO 66 I=1,NLINES	HEDING
66	WRITE (NT, 69) USEC(I),(ZTTH(JJ,I,IT),JJ=1,10)	HEDING
67	CONTINUE	HEDING
68	FORMAT('0',26X,'CONTACT FORCES - SEGMENT NO.',I3,' (' ,A4,	HEDING
	* ') VS. SEGMENT NO.',I3,' (' ,A4,')'//	HEDING
	* 13X,'DEFL- NORMAL FRICTION RESULTANT',	HEDING
	* 14X,'CONTACT LOCATION (' ,A4,')'/	HEDING
	* 4X,'TIME ECTION',3(3X,'FORCE',1X),	HEDING
	* 2(' SEG.',I3,' LOCAL REFERENCE ')/	HEDING
	* 3X,'(MSEC)',3X,' (' ,A4,')', 3(3X,' (' ,A4,')'),	HEDING
	* 2(5X,'X',7X,'Y',7X,'Z',4X)/1X)	HEDING
69	FORMAT(2F9.3,3F9.2,3F8.3,2X,3F8.3)	HEDING
161	CONTINUE	VARTTH
C		HEDING
C	AIRBAG FORCES HEAD!	HEDING
C		HEDING
70	IF (NBAG.EQ.0) GO TO 82	HEDING
	IF (NPRT(18).EQ.6.OR.NPRT(18).EQ.9) GO TO 82	VARTTH
	IF (NPRT(18).GE.12) GO TO 82	VARTTH
	DO 77 J=1,NBAG	HEDING
	IF (MNBAG(J).EQ.0) GO TO 77	HEDING
	MT = MT + 1	HEDING
	NT = MT	HEDING
	IF (LNEW) NT = 6	HEDING
C	P & E CARRIAGE CONTROL	PECONV
C	CALL CARCON(NT,1)	80386
	IT = MT - 20	HEDING
	PAGE = FLOAT(MT) + XPAGE	HEDING

	IF (NT.EQ.6) WRITE(NT,121) DATE,BLANK,NPG	PAGE
	IF (NT.NE.6) WRITE(NT,121) DATE	PAGE
	IF (NT.EQ.6) NPG=NPG+1	PAGE
	WRITE (NT,21) COMENT,PAGE,VPSTTL,BDYTTL	PAGE
	WRITE (NT,78) J,(BAGTTL(I,J),I=1,5)	HEDING
	IF (.NOT.LNEW) GO TO 72	HEDING
	DO 71 I=1,NLINES	HEDING
71	WRITE (NT, 79) USEC(I),(ZTTH(JJ,I,IT),JJ=1,12)	HEDING
72	KBAG = 0	HEDING
	KP = NPANEL(J)+1	HEDING
	DO 73 K=1,KP	HEDING
	KBAG = KBAG+1	HEDING
73	HEAD(KBAG) = PHED(K)	HEDING
	KP = MNBAG(J)	HEDING
	DO 74 K=1,KP	HEDING
	KBAG = KBAG+1	HEDING
	M = MBAG(2,K,J)	HEDING
74	HEAD(KBAG) = SEG(M)	HEDING
	DO 76 J1=1,KBAG,4	HEDING
	J2 = MINO(J1+3,KBAG)	HEDING
	MT = MT + 1	HEDING
	NT = MT	HEDING
	IF (LNEW) NT = 6	HEDING
C	P & E CARRIAGE CONTROL	PECONV
C	CALL CARCON(NT,1)	80386
	IT = MT - 20	HEDING
	PAGE = FLOAT(MT) + XPAGE	HEDING
	IF (NT.EQ.6) WRITE(NT,121) DATE,BLANK,NPG	PAGE
	IF (NT.NE.6) WRITE(NT,121) DATE	PAGE
	IF (NT.EQ.6) NPG=NPG+1	PAGE
	WRITE (NT,21) COMENT,PAGE,VPSTTL,BDYTTL	PAGE
	WRITE (NT,80)UNITM,J,(BAGTTL(I,J),I=1,5),(BLANK,J,HEAD(K),K=J1,J2)	HEDING
	WRITE (NT,51) (BLANK,K=J1,J2)	HEDING
	WRITE (NT,38)	HEDING
	IF (.NOT.LNEW) GO TO 76	HEDING
	JJ = 3*(J2-J1+1)	HEDING
	DO 75 I=1,NLINES	HEDING
75	WRITE (NT, 81) USEC(I),(ZTTH(K,I,IT),K=1,JJ)	HEDING
76	CONTINUE	HEDING
77	CONTINUE	HEDING
78	FORMAT('0',26X,'PARAMETERS FOR AIRBAG NO.',12,4X,5A4//	HEDING
	* 16X,'SUPPLY CYLINDER STATIC'/	HEDING
	* 4X,'TIME',8X,'PRES.',4X,'TEMP.',4X,'PRES.',12X,'AIRBAG',	HEDING
	* 3X,'CENTER',14X,'AIRBAG SEMIAXES',12X,'ORIENTATION (DEG.)'/	HEDING
	* 3X,'(MSEC)',7X,'(PSIG) (DEG.R) (PSIG)',8X,'X',8X,'Y',8X,'Z',	HEDING
	* 11X,'A',8X,'B',8X,'C',10X,'YAW',4X,'PITCH',5X,'ROLL'/)	HEDING
79	FORMAT (F9.3,3X,3F9.2,2(3X,3F9.3).3X,3F9.2)	HEDING
80	FORMAT('0',26X,'CONTACT FORCES ('A4,') ON AIRBAG NO.',12,4X,5A4//	HEDING
	* /4X,'TIME',4(A1,11X,'AIRBAG',12,' VS. ',A4,1X))	HEDING
81	FORMAT (F9.3,4(3X,3F9.2))	HEDING
82	RETURN	HEDING
	END	HEDING

```

SUBROUTINE HERRON(HD3,NT1,THETO,THETOP)                                HERRON
C                                                                    REV IV    07/23/86TWOPI
C COMPUTES THETO - ANGLE OF JOINT STOP                               HERRON
C     THETOP- DERIVATIVE OF THETO WITH RESPECT TO PHI              HERRON
C                                                                    HERRON
C     FROM HD3 - COMPONENTS OF VECTOR DEFINING PHI                HERRON
C     NT1 - INDEX TO TAB ARRAY DEFINING FUNCTION                  HERRON
C                                                                    HERRON
C     IMPLICIT REAL*8(A-H,O-Z)                                     HERRON
C     COMMON/TABLES/MXNT1,MXNTB,MXTB1,MXTB2,NT1(50),NTAB(1250),TAB(4500)DIMENB
C     COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                   HERRON
C     *      UNITL,UNITM,UNITT,GRAVTY(3),TWOPI                    TWOPI
C     DIMENSION HD3(3)                                            HERRON
C     IF (TAB(NT1+1).LE.0.0) GO TO 30                             HERRON
C     IF (TAB(NT1+2).LE.0.0) GO TO 30                             HERRON
C                                                                    HERRON
C     THETO = P1(CP) + SP*P2(CP)                                  HERRON
C                                                                    HERRON
C     THETOP = -SP*P1'(CP) + CP*P2(CP) - SP**2*P2'(CP)         HERRON
C                                                                    HERRON
C     WHERE P1(X),P2(X) ARE THE TWO 5TH ORDER POLYNOMIALS DEFINED HERRON
C           IN TAB(NT1+5) AND TAB(NT1+11)                         HERRON
C           P1'(X),P2'(X) ARE THEIR DERIVATIVES WITH RESPECT TO PHI HERRON
C           AND SP,CP ARE SIN(PHI) AND COS(PHI)                   HERRON
C                                                                    HERRON
C     STH2 = 1.0-HD3(3)**2                                         HERRON
C     STH = DSQRT(STH2)                                            HERRON
C     CP = HD3(1)/STH                                              HERRON
C     SP = HD3(2)/STH                                              HERRON
C     P1 = TAB(NT1+5 )+ CP*(TAB(NT1+6 )                            HERRON
C *          + CP*(TAB(NT1+7 )                                     HERRON
C *          + CP*(TAB(NT1+8 )                                     HERRON
C *          + CP*(TAB(NT1+9 )                                     HERRON
C *          + CP*(TAB(NT1+10) )))) .                             HERRON
C     P2 = TAB(NT1+11)+ CP*(TAB(NT1+12)                           HERRON
C *          + CP*(TAB(NT1+13)                                     HERRON
C *          + CP*(TAB(NT1+14)                                     HERRON
C *          + CP*(TAB(NT1+15)                                     HERRON
C *          + CP*(TAB(NT1+16) ))))                                HERRON
C     P1P = TAB(NT1+6 )+ CP*(2.0*TAB(NT1+7 )                       HERRON
C *          + CP*(3.0*TAB(NT1+8 )                               HERRON
C *          + CP*(4.0*TAB(NT1+9 )                               HERRON
C *          + CP*(5.0*TAB(NT1+10) ))))                           HERRON
C     P2P = TAB(NT1+12)+ CP*(2.0*TAB(NT1+13)                       HERRON
C *          + CP*(3.0*TAB(NT1+14)                               HERRON
C *          + CP*(4.0*TAB(NT1+15)                               HERRON
C *          + CP*(5.0*TAB(NT1+16) ))))                           HERRON
C     THETO = P1 + SP*P2                                           HERRON
C     THETOP = CP*P2 - SP*(P1P + SP*P2P)                           HERRON
C     GO TO 99                                                     HERRON
C                                                                    HERRON
C     EVALUATE THETO AND THETOP FROM REGULAR FUNCTION DEFINITION WHERE HERRON

```



```

C   · THETO (ORDINATE) IS A FUNCTION OF PHI (ABSCISSA) (0 < PHI < 2*PI)HERRON
C
30 PHI = DATAN2(HD3(2),HD3(1)) HERRON
   IF (PHI.LT.0.0) PHI = PHI + TWOPI HERRON
   THETO = EVALFD(PHI,NT1,1) HERRON
   THETOP = EVALFD(PHI,NT1,0) HERRON
99 RETURN HERRON
   END HERRON

```



```

DO 17 I=2,NPTS
H2 = SQRT(Z(I,JH)) * Z(I,JH)**2
DT = 0.5*(Z(I,1) - Z(I-1,1))
AREA(I) = AREA(I-1) + DT*(Z(I-1,JH)+Z(I,JH))
HSI = HSI + DT*(HL+H2)
IF (HMX.GT.Z(I,JH)) GO TO 17
HMX = Z(I,JH)
HMT = Z(I,1)
17 H1 = H2
HSI = 0.001*HSI
C
C COMPUTE HIC - HEAD INJURY CRITERION - AND TIME DURATION HT1,HT2
C
DO 19 K=2,NPTS
DO 18 L=K,NPTS
DT = Z(L,1) - Z(K-1,1)
DH = AREA(L) - AREA(K-1)
HT = DH/DT
HM = DT*SQRT(HT)*HT**2
IF (HM.LE.HIC) GO TO 18
HIC = HM
HT1 = Z(K-1,1)
HT2 = Z(L,1)
HA2 = Z(L,JH)
HA1 = Z(K-1,JH)
AVE = HT
18 CONTINUE
19 CONTINUE
HIC = 0.001*HIC
WRITE (6,21) HIC,HT1,HT2,HA1,HA2,AVE
21 FORMAT (1H0, ' HEAD INJURY CRITERION'//
* ' HIC = ', F8.2,
* 9X, 'TIME DURATION = ', F9.3, ' TO ', F9.3, ' MSEC'//
* 20X, 'WITH HEAD RESULTANTS = ', F9.3, ', AND ', F9.3, ' G''S'//
*14X, 'AVERAGE HEAD RESULTANT FOR TIME DURATION = ', F9.3, ' G''S')
WRITE (6,22) HSI,HMX,HMT
22 FORMAT (1H0, ' HEAD SEVERITY INDEX'//
* ' HSI = ', F8.2//
* ' MAX HEAD RESULTANT = ', F9.3, ' G''S AT ', F9.3, ' MSEC')
23 IF (JDTPTS(2).EQ.0) GO TO 25
WRITE (6,24) CSI,CMX,CMT
24 FORMAT (1H0, ' CHEST SEVERITY INDEX'//
* ' CSI = ', F8.2//
* ' MAX CHEST RESULTANT = ', F9.3, ' G''S AT ', F9.3, ' MSEC')
25 CONTINUE
IF(NPTS.LT.25) WRITE(6,101) NPTS
101 FORMAT(1X,/,2X,'HIC, HSI AND CSI NOT COMPUTED BECAUSE THE NUMBER
*OF POINTS TO BE USED IN THE COMPUTATION =',12,',',/,2X
* 'WHICH IS LESS THAN THE MINIMUM REQUIREMENT OF 25 POINTS.',/)
RETURN
END

```

```

SUBROUTINE HINPUT                                HINPUT
C                                               REV IV    07/23/86TWOPI
C CONTROLS THE INPUT OF CARDS F.8.A - F.8.D CONTAINING THE SETUP ANDHINPUT
C CONTROL OF THE HARNESS BELT SYSTEM.          HINPUT
C                                               HINPUT
C                                               HINPUT
C IMPLICIT REAL*8(A-H,O-Z)                      HINPUT
COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND, HINPUT
* NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),        HINPUT
* UNITL,UNITM,UNITT,GRAVITY(3),TWOPI            TWOPI
COMMON/HRNESS/ BAR(15,100),BB(100),BBDOT(100),PLOSS(2,100), HINPUT
* XLONG(20),HTIME(2),IBAR(5,100),NL(2,100),      HINPUT
* NPTSPB(20),NPTPLY(20),NTHRNS(20),NBLTPH(5)     HINPUT
COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(1250),TAB(4500)DIMENB
COMMON/CNTRSRF/ PL(24,30),BELT(20,8),TPTS(6,8),BD(24,40)  EDGE
COMMON/TITLES/ DATE(3),COMENT(40),VPS TTL(20),BDYTTL(5),  HINPUT
* BLTTTL(5,8),PLTTTL(5,30),BAGTTL(5,6),SEG(30),    HINPUT
* JOINT(30),CGS(30),JS(30)                      HINPUT
REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTTL,BAGTTL,SEG,JOINT HINPUT
LOGICAL*1 CGS,JS                                HINPUT
C THIS COMMON/TEMTVS/ IS SHARED BY CINPUT, FINPUT, HINPUT AND FDINITHINPUT
COMMON/TEMPVS/ JTITLE(5,51),NF(5),MS(3),KTITLE(31),DDMY(34966) 80386
REAL JTITLE,KTITLE                             HINPUT
IF (NHRNSS.EQ.0) GO TO 99                       HINPUT
C                                               HINPUT
C INPUT CARD F.8.A                              HINPUT
C (NOTE: NHRNSS NOW SUPPLIED ON INPUT CARD D.1) HINPUT
C NBLTPH - NO. OF BELTS PER HARNESS           HINPUT
C                                               HINPUT
READ (5,11) (NBLTPH(I),I=1,NHRNSS)             HINPUT
11 FORMAT(18I4)                                 HINPUT
WRITE (6,12) NPG,NHRNSS,(NBLTPH(I),I=1,NHRNSS) PAGE
NPG=NPG+1                                       PAGE
12 FORMAT('1 HARNESS-BELT SYSTEM INPUT',96X,'PAGE',I5/120X,    PAGE
* 'CARDS F.8'/' NO. OF HARNESSES =',I4//        PAGE
* ' NO. OF BELTS PER HARNESS =',5I6)          HINPUT
J1 = 1                                          HINPUT
K1 = 1                                          HINPUT
DO 90 I=1,NHRNSS                               HINPUT
IF (NBLTPH(I).LE.0) GO TO 90                   HINPUT
J2 = J1 + NBLTPH(I) -1                        HINPUT
C                                               HINPUT
C INPUT CARD F.8.B - NPTSPB - NO. OF POINTS PER BELT. HINPUT
C                                               HINPUT
C                                               HINPUT
READ (5,11) (NPTSPB(J),J=J1,J2)              HINPUT
WRITE (6,13) I,(NPTSPB(J),J=J1,J2)          HINPUT
13 FORMAT('0 FOR HARNESS NO.',I3,' NO. OF POINTS PER BELT =',20I4) HINPUT
DO 80 J=J1,J2                                  HINPUT
IF (NPTSPB(J).EQ.0) GO TO 80                 HINPUT
C                                               HINPUT
C INPUT CARD F.8.C - 5 FUNCTION NOS AND LENGTH OF EACH BELT. HINPUT

```

C		HINPUT
	READ (5,14) NF,XLONG(J)	HINPUT
14	FORMAT(5I4,F12.6)	HINPUT
	WRITE (6,15) I,J,NF,XLONG(J),UNITL	HINPUT
15	FORMAT('0 HARNESS NO.',I3,' BELT NO.',I3,' FUNCTION NOS.',5I6,	HINPUT
*	' REFERENCE SLACK = ',F9.3,1X,A4/)	HINPUT
	IF (XLONG(J).EQ.0.0) XLONG(J) = EPS(24)	HINPUT
	WRITE (6,16)	HINPUT
16	FORMAT ('0 K KS KE NT NPD NDR FUNCTION NOS.',	HINPUT
*	66X,'CARDS F.8.D'/)	CHGIII
C		HINPUT
C	SET UP POINTERS IN NTAB AND INITIAL VALUES OF TAB FOR BELT J	HINPUT
C	AS WAS DONE FOR OTHER CONTACTS IN SUBROUTINE FINPUT.	HINPUT
C		HINPUT
	NTHRNS(J) = MXNTB+1	HINPUT
	CALL FDINIT	HINPUT
	K2 = K1 + NPTSPB(J) - 1	HINPUT
	DO 70 K=K1,K2	HINPUT
C		HINPUT
C	INPUT CARD F.8.D	HINPUT
C		HINPUT
	READ (5,21) KS,KE,NPD,NDR,NF, (BAR(L,K),L=1,3)	HINPUT
21	FORMAT (9I4,3F12.0)	HINPUT
	READ (5,22) (BAR(L,K),L=7,12)	HINPUT
22	FORMAT (6F12.0)	HINPUT
	ICHEC = 0	CHGIII
	IF (K.EQ.K1.OR.K.EQ.K2) ICHEC = 1	CHGIII
	IF (ICHEC.EQ.1.AND.NPD.EQ.0) STOP 60	CHGIII
	IF (ICHEC.EQ.1.AND.NDR.EQ.0) STOP 61	CHGIII
	IF (NDR.EQ.0.AND.NPD.NE.0) STOP 62	CHGIII
	IBAR(1,K) = KS	HINPUT
	IBAR(2,K) = KE	HINPUT
	IBAR(4,K) = NPD	HINPUT
	IBAR(5,K) = NDR	HINPUT
	IBAR(3,K) = MXNTB+1	HINPUT
	CALL FDINIT	HINPUT
	SQRER = 1.0	HINPUT
	IF (KE.NE.0) SQRER = DSQRT(XDY(BAR(1,K),BD(7,KE),BAR(1,K)))	HINPUT
	DO 26 L=1,3	HINPUT
	IF (KE.NE.0) BAR(L+6,K) = BD(L+3,KE)	HINPUT
26	BAR(L+3,K) = BAR(L,K)/SQRER	HINPUT
	WRITE (6,31) K,(IBAR(L,K),L=1,5),NF	HINPUT
31	FORMAT (11I6)	HINPUT
70	CONTINUE	HINPUT
	WRITE (6,71) UNITL,UNITL,UNITL,UNITL	HINPUT
71	FORMAT ('0',12X,'BASE REFERENCE (' , A4,')',	HINPUT
*	7X,'ADJUSTED REFERFNCE (' , A4,')',	HINPUT
*	11X,'OFFSET (' , A4,')',	HINPUT
*	11X,'PREFERRED DIRECTION (' ,A4,')'/	HINPUT
*	5X,'K', 4(8X,'X',8X,'Y',8X,'Z',3X) /)	HINPUT
	WRITE (6,72) (K,(BAR(L,K),L=1,12),K=K1,K2)	HINPUT
72	FORMAT (I6,3X,3F9.3,3X,3F9.3,3X,3F9.3,3X,3F9.3)	HINPUT

```
      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
HINPUT
HINPUT
HINPUT
HINPUT
HINPUT
HINPUT
HINPUT
HINPUT
HINPUT
HINPUT
```


	IF (NBLTPH(NH).LE.0) GO TO 61	HPTURB
	ITER = 1	HPTURB
	KNL1 = KNLO	HPTURB
	KNLN = 0	CHGIII
C		HPTURB
C	START OF DO 59 * ITER=1,MAXITR LOOP	HPTURB
C		HPTURB
	13 NJ2 = 54	HPTURB
	DO 14 I=1,NJ2	HPTURB
	DO 14 J=1,NJ2	HPTURB
	14 IJK(I,J) = 0	HPTURB
	KNLO = KNL1	HPTURB
	J2 = J1 + NBLTPH(NH) - 1	HPTURB
	NTP = 0	HPTURB
	IJ = 0	HPTURB
	CALL HBELT (J1,J2,KNLO,1)	HPTURB
	KHO = 0	HPTURB
	KNLO = KNL1	HPTURB
	DO 15 NB=J1,J2	HPTURB
	IF (NPTPLY(NB).LE.0) GO TO 15	HPTURB
	NPTS = NPTPLY(NB)	HPTURB
	CALL HSETC (NPTS,KHO,KNLO,NTP,IJ)	HPTURB
	KHO = KHO + NPTS	HPTURB
	KNLO = KNLO + NPTS	HPTURB
	15 CONTINUE	HPTURB
	KNLN = KNLO	CHGIII
C		HPTURB
C	SET UP C AND IJK ELEMENTS FOR TIE-POINTS.	HPTURB
C		HPTURB
	KNLO = KNL1	HPTURB
	KNLK = KNLO + 1	HPTURB
	K1 = KNLK	HPTURB
	DO 22 NB=J1,J2	HPTURB
	IF (NPTPLY(NB).LE.0) GO TO 22	HPTURB
	K2 = K1 + NPTPLY(NB) - 1	HPTURB
	DO 21 KNL=K1,K2	HPTURB
	KI = NL(1,KNL)	HPTURB
	KS = IABS(IBAR(1,KI))	HPTURB
	IF (KS.LT.100) GO TO 21	HPTURB
	KS1 = KS/100	HPTURB
	DO 16 K=KNLK,KNL	HPTURB
	KK = K	HPTURB
	KI = NL(1,K)	HPTURB
	KS = IABS(IBAR(1,KI))	HPTURB
	IF (KS.LT.100) GO TO 16	HPTURB
	KS2 = KS/100	HPTURB
	IF (KS2.EQ.KS1) GO TO 17	HPTURB
	16 CONTINUE	HPTURB
	17 IF (KK.EQ.KNL) GO TO 21	HPTURB
	KK1 = KK - KNLO	HPTURB
	KK2 = KNL - KNLO	HPTURB
	IQ1 = MAX0(1,KK2-1)	HPTURB


```

      DO 46 J=1,3
46 BAR(J+12,K) = 0.0
      IF (DHT.EQ.0.0) GO TO 49
      DO 48 K=K1,K2
      KH = KH + 1
      KI = NL(1,K)
      PLOSS(2,KI) = PLOSS(2,KI) + DHT*PTLOSS(2,KH)
      IF (K.EQ.K1) GO TO 47
      BBDOT(K-1) = (BB(K-1)-OLDBB(K-1))/DHT
      PLOSS(1,K-1) = PLOSS(1,K-1) + DHT*PTLOSS(1,KH-1)
      BLOSS(1,NB) = BLOSS(1,NB) + PLOSS(1,K-1)
47 DO 48 J=1,3
48 BAR(J+12,KI) = (BAR(J+3,KI)-BAR(J,KI))/DHT
      BBDOT(K2) = 0.0
      PLOSS(1,K2) = 0.0
49 K1 = K2+1
      DO 50 K=K1,K2
50 BLOSS(2,NB) = BLOSS(2,NB) + PLOSS(2,K)
      HLOSS(1,NH) = HLOSS(1,NH) + BLOSS(1,NB)
      HLOSS(2,NH) = HLOSS(2,NH) + BLOSS(2,NB)
51 CONTINUE
52 IF (NPRT(28).EQ.0) GO TO 59
      IF (.NOT.LAST .AND. IABS(NPRT(28)).EQ.1) GO TO 59
      K1 = K0
      KI = 0
      DO 57 NB=J1,J2
      IF (NPTPLY(NB).LE.0) GO TO 57
      WRITE (6,53) NB,NH
53 FORMAT ('0 BELT NO.',I4,' OF HARNESS NO.',I4)
      K2 = K1 + NPTPLY(NB) - 1
      DO 54 K=K1,K2
      KH = KH + 1
      KI = NL(1,K)
      KS = IBAR(1,KI)
      BK = 0.0
      IF (K.NE.K1) BK = BB(K-1)
      PLS = 0.0
      IF (K.NE.K1) PLS = PLOSS(1,K-1)
      T(1) = BAR(4,KI)
      T(2) = BAR(5,KI)
      T(3) = BAR(6,KI)
      KJ = MOD(IABS(KS),100)
      IF (LPMI(KJ).NE.0) CALL DOT 1 (DPMI(1,1,KJ),BAR(4,KI),T)
54 WRITE (6,55) K,KI,KS,BK,PLS,(T(J),J=1,3)
      *
      (FCE(J,KH),J=1,3),PLOSS(2,KI)
55 FORMAT ('I8,F10.3,F12.3,2X,3F9.3,3X,3F11.3,3X,F12.3)
      IF (LAST) WRITE (6,56) BLOSS(1,NB),BLOSS(2,NB)
56 FORMAT ('0 TOTAL BELT ENERGY LOSS',7X,F12.3,68X,F12.3)
      K1 = K2 + 1
57 CONTINUE
      IF (LAST) WRITE (6,58) HLOSS(1,NH),HLOSS(2,NH)
58 FORMAT ('0 TOTAL HARNESS ENERGY LOSS',7X,F12.3,68X,F12.3)

```

59	ITER = ITER + 1	HPTURB
C		HPTURB
C	END OF DO 59 ITER-1,MAXITR LOOP	HPTURB
C		HPTURB
	IF (.NOT.LAST) GO TO 13	HPTURB
	IF (ITER.GT.MAXITR) WRITE (6,60) MAXITR,TSEC,DELMAX,SCALE	HPTURB
60	FORMAT ('0 HPTURB ITER =',I4,' AT TIME =',F8.3,	HPTURB
	* ' MSEC. DELMAX =',F10.6,' SCALE =',F10.6)	HPTURB
	J1 = J2 + 1	HPTURB
	K0 = K1	HPTURB
	KHLO = KNLN	CHGIII
61	CONTINUE	HPTURB
	IF (NPRT(28).LT.0) NPRT(28) = 0	HPTURB
	CALL ELTIME (2,39)	HPTURB
	RETURN	HPTURB
	END	HPTURB

```

SUBROUTINE HSETC (NPTS,KHO,KNLO,NTP,IJ)                                HSETC
C                                                                    REV III.2 08/08/84REVIII
IMPLICIT REAL*8 (A-H,O-Z)                                           HSETC
COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),HSETC
* SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)                       HSETC
COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(1250),TAB(4500)DIMENB
COMMON/HRNESS/ BAR(15,100),BB(100),BBDOT(100),PLOSS(2,100),       HSETC
* XLONG(20),HTIME(2),IBAR(5,100),NI(2,100),                       HSETC
* NPTSPB(20),NPTPLY(20),NTHRNS(20),NBLTPH(5)                       HSETC
C THIS COMMON/TEMPVS/ IS SHARED BY HPTURB, HBPLAY, HEELT AND HSETC. HSETC
COMMON/TEMPVS/ B(3,3,3),S(3,3),T(3),F(3),V(3),T1(3),T2(3),       HSETC
* E(3,3,50),EDOT(3,50),FCE(3,50),FR(3,50),ZR(3,50),             HSETC
* TR(3,50),U(3,50),PTLOSS(2,50),BL(50),FB(50),FP(50),           HSETC
* OLDBB(100),RHS(3,54),C(3,3,200),IJK(54,54)                     HSETC
* DMMY(29942) -                                                    80386
DIMENSION KM(3),MK(2)                                              HSETC
ONE = 1.0                                                            HSETC
KNL = KNLO                                                            HSETC
KH = KHO                                                              HSETC
K1 = KHO + NTP + 1                                                  HSETC
K2 = KHO + NTP + NPTS                                              HSETC
DO 60 K=K1,K2                                                       HSETC
C                                                                    HSETC
C HERE K IS INDEX OF IJK AND RHS ARRAYS                             HSETC
C KH IS INDEX OF POINTS IN PLAY ON EACH HARNESS                    HSETC
C KNL IS INDEX OF ALL POINTS IN PLAY                                HSETC
C KI IS INDEX OF ALL POINTS                                         HSETC
C                                                                    HSETC
KH = KH + 1                                                         HSETC
KNL = KNL + 1                                                       HSETC
C                                                                    HSETC
C ZERO C(K,K) , C(K,K-1) , C(K,K+1) & RHS(K); SET IJK(K,K) = IJ  HSETC
C                                                                    HSETC
KM(1) = K+1                                                         HSETC
KM(2) = K-1                                                         HSETC
KM(3) = K                                                           HSETC
IF (K.EQ.K2) KM(1) = 0                                              HSETC
IF (K.EQ.K1) KM(2) = 0                                              HSETC
KK = IJ                                                             HSETC
DO 12 L=1,3                                                         HSETC
RHS(L,K) = 0.0                                                      HSETC
IF (KM(L).EQ.0) GO TO 12                                           HSETC
KK = KK+1                                                           HSETC
DO 11 I=1,3                                                         HSETC
DO 11 J=1,3                                                         HSETC
11 C(I,J,KK) = 0.0                                                  HSETC
12 CONTINUE                                                         HSETC
IJ = IJ+1                                                           HSETC
IJK(K,K) = IJ                                                       HSETC
C                                                                    HSETC
C COMPUTE CNORM, IF ZERO, SET C(F,K) = 1                            HSETC
C                                                                    HSETC

```

	CNORM = 0.0	HSETC
	IF (K.NE.K2) CNORM = FB(KH)/BL(KH)	HSETC
	IF (K.NE.K1) CNORM = CNORM + FB(KH-1)/BL(KH-1)	HSETC
	KI = NL(1,KNL)	HSETC
	IF (IABS(IBAR(1,KI)).GT.100) GO TO 14	HSETC
C	IF (CNORM.NE.0.0) GO TO 14	BUTLER1
	KK = IJK(K,K)	HSETC
	DO 13 I=1,3	HSETC
13	C(I,I,KK) = ONE	HSETC
	IF (CNORM.EQ.0.0) GO TO 60	BUTLER1
14	KK = IBAR(3,KI)	HSETC
	NFD = NTAB(KK+1)	HSETC
	NFR = NTAB(KK+5)	HSETC
C		HSETC
C	SET UP B(3,3,3) AND S(3,3)	HSETC
C		HSETC
	MK(1) = KH	HSETC
	MK(2) = KH-1	HSETC
	IF (K.EQ.K2) MK(1) = 0	HSETC
	IF (K.EQ.K1) MK(2) = 0	HSETC
	DO 18 M=1,2	HSETC
	KK = MK(M)	HSETC
	IF (KK.NE.0 .AND. CNORM.NE.0.0) GO TO 16	HSETC
	DO 15 I=1,3	HSETC
	S(I,M) = 0.0	HSETC
	DO 15 J=1,3	HSETC
15	B(I,J,M) = 0.0	HSETC
	GO TO 18	HSETC
16	CALL DOT31 (E(1,1,KH),U(1,KK),T)	HSETC
	KIM = KNL + 1 - M	HSETC
	FB1 = FB(KK)/BL(KK)	HSETC
	FB2 = FP(KK)/BB(KIM) - FB1	HSETC
	FB3 = FP(KK)*BL(KK)/BB(KIM)**2	HSETC
	DO 17 I=1,3	HSETC
	SGN = ONE	HSETC
	IF (FR(I,KH).LT.0.0) SGN = -ONE	HSETC
	S(I,M) = SGN*(FB3*T(I))	HSETC
	DO 17 J=1,3	HSETC
17	B(I,J,M) = SGN*(FB1*E(J,I,KH) + FB2*T(I)*U(J,KK))	HSETC
18	CONTINUE	HSETC
	DO 19 I=1,3	HSETC
	S(I,3) = -(S(I,1) + S(I,2))	HSETC
	DO 19 J=1,3	HSETC
19	B(I,J,3) = -(B(I,J,1) + B(I,J,2))	HSETC
	IF (NFR.EQ.0) GO TO 20	HSETC
	R(1) = TAB(NFR+2)	HSETC
	R(2) = TAB(NFR+4)	HSETC
20	R(3) = 0.0	HSETC
	DO 50 M=1,3	HSETC
	RH = 0.0	HSETC
	IF (M.EQ.3) GO TO 31	HSETC
	IF (NFR.EQ.0) GO TO 48	HSETC

C		HSETC
C	CONSTRAINTS 1 AND 2	HSETC
C		HSETC
	SGN = -ONE	HSETC
	FR3 = DABS(FR(M,KH)) - R(M)*DABS(FR(3,KH))	HSETC
	IF (IBAR(1,KI).GT.0) RH = FR3	HSETC
	IF (FR3.LE.0.0) GO TO 48	HSETC
	GO TO 40	HSETC
C		HSETC
C	CONSTRAINT NO. 3	HSETC
C		HSETC
31	IF (NFD.EQ.0) GO TO 48	HSETC
	IF (IBAR(1,KI).LT.0) GO TO 40	HSETC
	SGN = ONE	HSETC
	RMAG2 = TR(1,KH)**2 + TR(2,KH)**2 + TR(3,KH)**2	HSETC
	RMAG = DSQRT(RMAG2)	HSETC
	RER2 = TR(1,KH)*E(1,3,KH)+ TR(2,KH)*E(2,3,KH)+ TR(3,KH)*E(3,3,KH)	HSETC
	RER2 = EDOT(3,KH)*RER2	HSETC
	RER = DSQRT(RER2)	HSETC
	PEN = RMAG/RER - RMAG	HSETC
	RRDOT = BAR(4,KI)*BAR(13,KI)	HSETC
*	+ BAR(5,KI)*BAR(14,KI)	HSETC
*	+ BAR(6,KI)*BAR(15,KI)	HSETC
	KS = IABS(IBAR(1,KI))	HSETC
	IF (KS.GT.100) KS = MOD(KS,100)	HSETC
	CALL DOT31 (D(1,1,KS),BAR(13,KI),T)	HSETC
	ERDOT = E(1,3,KH)*T(1) + E(2,3,KH)*T(2) + E(3,3,KH)*T(3)	HSETC
	C1 = PEN/RMAG2	HSETC
	C2 = RMAG*EDOT(3,KH)/(RER*RER2)	HSETC
	PDOT = C1*RRDOT - C2*ERDOT	HSETC
	NFDZ = IBAR(3,KI)	CHGIII
	CALL FRCDFL (PEN,PDOT,NFDZ,0,FDP,ELOSS)	CHGIII
	CALL FRCDFL (PEN,PDOT,NFDZ,1,FD,ELOSS)	CHGIII
	K' = FD + FR(3,KH)	HSETC
	P1LOSS(2,KH) = ELOSS	HSETC
	C1 = FDP*C1	HSETC
	C2 = FDP*C2	HSETC
	SGNB3 = -DSIGN(ONE,FR(3,KH))	HSETC
	DO 32 J=1,3	HSETC
32	B(3,J,3) = SGNB3*B(3,J,3) - C1*TR(J,KH) + C2*E(J,3,KH)	HSETC
40	DO 47 LL=1,3	HSETC
	L = 4 - LL	HSETC
	IF (KM(L).EQ.0) GO TO 47	HSETC
	DO 42 J=1,3	HSETC
42	V(J) = R(M)*B(3,J,L) + SGN*B(M,J,L)	HSETC
	KL = KM(L)	HSETC
	KML = KNL + KL - K	HSETC
	KIL = NL(1,KML)	HSETC
	IF (IBAR(5,KIL).NE.0) GO TO 43	HSETC
	KHL = KH + KL - K	HSETC
	CALL DOT31 (E(1,1,KHL),V,T)	HSETC
	T(2) = R(M)*S(3,L) + SGN*S(M,L)	HSETC

CALL MAT31 (E(1,1,KHL),T,V)	HSETC
43 IF (LL.NE.1) GO TO 44	HSETC
VE = V(1)*E(1,M,KH) + V(2)*E(2,M,KH) + V(3)*E(3,M,KH)	HSETC
EV = 1.0	HSETC
IF (IABS(IBAR(1,KI)).LT.100)	HSETC
* EV = DSIGN(ONE,VE)/DSQRT(V(1)**2+V(2)**2+V(3)**2)	HSETC
RH = EV*RH	HSETC
44 IF (IJK(K,KL).NE.0) GO TO 45	HSETC
IJ = IJ+1	HSETC
IJK(K,KL) = IJ	HSETC
45 KK = IJK(K,KL)	HSETC
DO 46 J=1,3	HSETC
VEV = EV*V(J)	HSETC
DO 46 I=1,3	HSETC
46 C(I,J,KK) = C(I,J,KK) + E(I,M,KH)*VEV	HSETC
47 CONTINUE	HSETC
DO 41 I=1,3	HSETC
41 RHS(I,K) = RHS(I,K) + RH*E(I,M,KH)	HSETC
GO TO 50	HSETC
48 IF (IBAR(1,KI).LE.0) GO TO 50	HSETC
KK = IJK(K,K)	HSETC
DO 49 I=1,3	HSETC
DO 49 J=1,3	HSETC
49 C(I,J,KK) = C(I,J,KK) + E(I,M,KH)*E(J,M,KH)	HSETC
50 CONTINUE	HSETC
60 CONTINUE	HSETC
RETURN	HSETC
END	HSETC

	SUBROUTINE HYABF(B,Z,A,F)		HYABF
C		REV IV	02/07/87HYABF
	IMPLICIT REAL*8(A-H,O-Z)		HYABF
C			HYABF
C	CALCULATES A, AZ, Z.AZ: OLD FORM MUST BE DIAGONAL		HYABF
C			HYABF
	DIMENSION B(24),Z(1),A(3,3)		HYABF
	P2 = 0.0		HYABF
	IF(B(1).LT.0.0)P2 = -B(1) - 2.0		HYABF
	F = 0.0		HYABF
	DO 30 I = 1,3		HYABF
	J = I		HYABF
	IF(B(1).LT.0.0)GO TO 10		HYABF
	A(I,1) = 1.0/B(I)**2		HYABF
	GO TO 15		HYABF
10	A(I,1) = B(I+16)		HYABF
	J = J + 1		HYABF
	A(I,1) = HYFCN(A(I,1),Z(I),B(J),P2)		HYABF
C	IF(P2.GT.0.0)A(I,1) = A(I,1)*DABS(Z(I)/B(J))**P2		HYABF
15	DO 20 J = 2,3		HYABF
20	A(I,J) = A(I,J-1)*Z(I)		HYABF
30	F = F + A(I,3)		HYABF
	RETURN		HYABF
	END		HYABF

	SUBROUTINE HYBND(M,Z,IV,U,C,X)		HYBND
C		REV IV	02/07/87HYBND
	IMPLICIT REAL*8(A-H,O-Z)		HYBND
C			HYBND
C	SEARCHES FOR POINT NEAREST CORNER - DIRECTION C*U		HYBND
C			HYBND
	DIMENSION Z(3,12),IV(12),U(3),X(3)		HYBND
	DO 20 I = 1,M,2		HYBND
	J = IV(I)		HYBND
	ATST = C*(U(1)*Z(1,J) + U(2)*Z(2,J) + U(3)*Z(3,J))		HYBND
	IF(I.EQ.1)GO TO 10		HYBND
	TEST = AMAX - ATST		HYBND
	COMP = DMAX1(DABS(AMAX),DABS(ATST))		HYBND
C	PRECISION TEST - TRY >1000??		HYBND
	IF(1000.*DABS(TEST).LT.COMP)TEST = 0.0		HYBND
	IF(TEST)10,15,20		HYBND
C	IF(AMAX-ATST)10,15,20		HYBND
	10 AMAX = ATST		HYBND
	J1 = J		HYBND
	15 J2 = J		HYBND
	20 CONTINUE		HYBND
	DO 25 I = 1,3		HYBND
	25 X(I) = 0.5*(Z(I,J1) + Z(I,J2))		HYBND
	RETURN		HYBND
	END		HYBND

	SUBROUTINE HYBOX(E,T,P,N,Z,IV)		HYBOX
C		REV IV	02/07/87HYBOX
	IMPLICIT REAL*8(A-H,O-Z)		HYBOX
C			HYBOX
C	COMPUTES THE INTERSECTION OF A PLANE WITH THE EDGES OF A BOX		HYBOX
C			HYBOX
	DIMENSION T(3),E(3),TU(3),T2(3),P(3)		HYBOX
C	TO BE SAFE IV AND Z SHOULD BE DIMENSION 14 IN CALLING PROGRAM		HYBOX
	DIMENSION IV(12)		HYBOX
	DIMENSION Z(3,12)		HYBOX
	DATA ONE/1.0D0/		HYBOX
C	T - PLANE VECTOR, P POINT IN PLANE		HYBOX
	TUV = 0.0		HYBOX
	DO 10 I = 1,3		HYBOX
	TU(I) = T(I)*E(I)		HYBOX
	T2(I) = 2.0*TU(I)		HYBOX
10	TUV = TUV + T(I)*(E(I) + P(I))		HYBOX
	N = 0		HYBOX
	J = 2		HYBOX
	K = 3		HYBOX
	DO 45 I = 1,3		HYBOX
	CK = -E(K)		HYBOX
	P1 = TUV		HYBOX
	DO 40 LL = 1,2		HYBOX
	P2 = P1 - T2(I)		HYBOX
	P3 = P2 - T2(J)		HYBOX
	P4 = P1 - T2(J)		HYBOX
	M = N		HYBOX
	IF(DSIGN(ONE,P2).EQ.DSIGN(ONE,P1))GO TO 15		HYBOX
	M = M + 1		HYBOX
	Z(I,M) = (P1/TU(I) - 1.0)*E(I)		HYBOX
	Z(J,M) = -E(J)		HYBOX
	Z(K,M) = CK		HYBOX
15	IF(DSIGN(ONE,P3).EQ.DSIGN(ONE,P2))GO TO 20		HYBOX
	M = M + 1		HYBOX
	Z(I,M) = E(I)		HYBOX
	Z(J,M) = (P2/TU(J) - 1.0)*E(J)		HYBOX
	Z(K,M) = CK		HYBOX
20	IF(DSIGN(ONE,P3).EQ.DSIGN(ONE,P4))GO TO 25		HYBOX
	M = M + 1		HYBOX
	Z(I,M) = (P4/TU(I) - 1.0)*E(I)		HYBOX
	Z(J,M) = E(J)		HYBOX
	Z(K,M) = CK		HYBOX
25	IF(DSIGN(ONE,P4).EQ.DSIGN(ONE,P1))GO TO 30		HYBOX
	M = M + 1		HYBOX
	Z(I,M) = -E(I)		HYBOX
	Z(J,M) = (P1/TU(J) - 1.0)*E(J)		HYBOX
	Z(K,M) = CK		HYBOX
30	IF(M.EQ N)GO TO 35		HYBOX
	CHECK FOR PRECISION (+-+- .OR -+-+)		HYBOX
	IF(M.EQ.N+4)GO TO 35		HYBOX
C	DELETE 0 LENGTH SIDE		HYBOX

	IF((Z(I,M-1).EQ.Z(I,M)).AND.(Z(J,M-1).EQ.Z(J,M)))GO TO 35	HYBOX
	N = M	HYBOX
35	P1 = P1 - T2(K)	HYBOX
40	CK = -CK	HYBOX
	J = K	HYBOX
45	K = I	HYBOX
C		HYBOX
	IF(N.LT.6)GO TO 65	HYBOX
	IV(1) = 1	HYBOX
	IV(2) = 2	HYBOX
	M = 2	HYBOX
	DO 60 J = 3,N,2	HYBOX
	D = DABS(Z(1,M)) + DABS(Z(2,M)) + DABS(Z(3,M))	HYBOX
	DO 55 L = 3,N	HYBOX
	DO 50 LL = 2,J	HYBOX
	IF(IV(LL-1).EQ.L) GO TO 55	HYBOX
50	CONTINUE	HYBOX
	F = DABS(Z(1,M)-Z(1,L))+DABS(Z(2,M)-Z(2,L))+DABS(Z(3,M)-Z(3,L))	HYBOX
	IF(F.GT.D)GO TO 55	HYBOX
	D = F	HYBOX
	K = L	HYBOX
55	CONTINUE	HYBOX
	M = K + 1	HYBOX
	IF(MOD(K,2).EQ.0)M = K - 1	HYBOX
	IV(J) = K	HYBOX
	IV(J+1) = M	HYBOX
60	CONTINUE	HYBOX
65	RETURN	HYBOX
	END	HYBOX

	SUBROUTINE HYDAD(D,A,DAD)		HYDAD
C		REV IV	02/07/87HYDAD
	IMPLICIT REAL*8(A-H,O-Z)		HYDAD
	COMPUTES D'A(*,1)D		HYDAD
	DIMENSION D(3,3),A(3),DAD(3,3)		HYDAD
	DO 10 I = 1,3		HYDAD
	DO 10 J = 1,3		HYDAD
	DAD(I,J) = 0.0		HYDAD
	DO 10 K = 1,3		HYDAD
	10 DAD(I,J) = DAD(I,J) + D(K,I)*A(K)*D(K,J)		HYDAD
	RETURN		HYDAD
	END		HYDAD

SUBROUTINE HYEST(BM,BN,TAB)		HYEST
C	REV IV	07/23/87HYEST
C LINEAR PROGRAM		HYEST
IMPLICIT REAL*8(A-H,O-Z)		HYEST
DIMENSION BM(24),BN(24),TAB(8)		HYEST
COMMON/TEMPVS/D12(3,3),A(3,3),B(3,3),XMN(3),RLN(3),XMM(3),		HYEST
* T(3),R(3),C(3,3),V(7),DMMY(35055)		80386
C R GOES FROM M TO N D12 = DM*DN'		HYEST
C R = 0 CANNOT BE SOLVED WITH THIS METHOD		HYEST
BE = 1.0		HYEST
RR = R(1)**2 + R(2)**2 + R(3)**2		HYEST
IF(RR.EQ.0.0)GO TO 30		HYEST
C R.R = 0 INVALID		HYEST
M = 1		HYEST
N = 1		HYEST
IF(BM(1).LT.0.0)M = 2		HYEST
IF(BN(1).LT.0.0)N = 2		HYEST
PM = 2.		HYEST
PN = 2.		HYEST
IF(M.EQ.2)PM = -BM(1)		HYEST
IF(N.EQ.2)PN = -BN(1)		HYEST
DO 10 I = 1,3		HYEST
T(I) = R(I)		HYEST
DO 10 J = 1,3		HYEST
10 B(I,J) = D12(I,J)		HYEST
IF(N.EQ.2)CALL DOTF33(D12,BN(8),B)		HYEST
DO 15 I = 1,3		HYEST
DO 15 J = 1,3		HYEST
15 C(I,J) = B(I,J)		HYEST
IF(M.EQ.2)CALL MAT33(BM(8),B,C)		HYEST
C C WILL TRANSFORM FROM NN TO MM		HYEST
IF(M.EQ.2)CALL MAT31(BM(8),R,T)		HYEST
CALL HYLPM(BM(M),BN(N))		HYEST
BE = V(7)		HYEST
IF(V(7).LE.1.0)GO TO 30		HYEST
CALL HYABF(BM(1),V(1),A,F1)		HYEST
CALL HYABF(BN(1),V(4),B,F2)		HYEST
C ESTIMATE ALPHA		HYEST
AA = A(1,2)**2 + A(2,2)**2 + A(3,2)**2		HYEST
BB = B(1,2)**2 + B(2,2)**2 + B(3,2)**2		HYEST
ALP = DSQRT(AA/BB)		HYEST
RA = F1**(1.0/PM)		HYEST
RB = F2**(1.0/PN)		HYEST
ALP = ALP*RA*F2/(RB*F1)		HYEST
C SCALE POINTS TO ELLIPSOIDS		HYEST
DO 20 I = 1,3		HYEST
V(I) = V(I)/RA		HYEST
20 V(I+3) = V(I+3)/RB		HYEST
C ESTIMATE BETA		HYEST
CALL MAT31(C,V(4),T)		HYEST
BE = (V(1)-T(1))**2 + (V(2)-T(2))**2 + (V(3) - T(3))**2		HYEST
BE = DSQRT(BE/RR)		HYEST

```
C STORE VALUES IN TAB ARRAY FOR CONTACT
  TAB(1) = ALP
  DO 25 I = 1,6
25 TAB(I+2) = V(I)
30 TAB(2) = BE
  RETURN
  END
```

```
HYEST
HYEST
HYEST
HYEST
HYEST
HYEST
HYEST
```

```

C      DOUBLE PRECISION FUNCTION HYFCN(C,Z,A,P)
      IMPLICIT REAL*8(A-H,O-Z)
      HYFCN = C
      IF(P.EQ.0.0)GO TO 10
      HYFCN = 0.0
      IF(Z.EQ.0.0)GO TO 10
      Q = P*(DLOG(DABS(Z)) - DLOG(A))
      IF(Q.GT.-88.5) HYFCN = C*DEXP(Q)
10  RETURN
      END

```

```

REV IV 02/07/87HYFCN
HYFCN
HYFCN
HYFCN
HYFCN
HYFCN
HYFCN
HYFCN
HYFCN
HYFCN
HYFCN

```


	SUBROUTINE HYLIM(A,U,B,V,C,W,Z,BD)		HYLIM
C	IMPLICIT REAL*8(A-H,O-Z)	REV IV	12/11/87HYFIX
C	GIVEN Z, FIND A,B,Z: ZEZ = 1, ZEV = 0, TZ = TP		HYLIM
	DIMENSION BD(24)		HYLIM
	DIMENSION U(3),V(3),W(3),EI(3),EJ(3),T(3),TV(3)		HYLIM
	DIMENSION Z(3),Z(3),EV(3),Q(3),S(3),EZ(3)		HYLIM
	DIMENSION SM(3,3)		HYLIM
	LOGICAL PASS,USEV		HYLIM
	PASS = .FALSE.		HYLIM
	ITER = 100		HYLIM
	PP = -1./BD(1)		HYLIM
	POW = -BD(1) - 2.0		HYLIM
	P1 = -BD(1) - 1.0		HYLIM
	PO1 = 1.0/P1		HYLIM
	P2 = -BD(1)/P1		HYLIM
	DO 10 I = 1,3		HYLIM
	TV(I) = 0.0		HYLIM
10	IF(V(I).NE.0.0)TV(I) = HYFCN(1.0/V(I),V(I),BD(I+1),P2)		HYFIX
C	GET RECIPROCAL SET		HYLIM
	CALL CROSS(V,W,EI)		HYLIM
	CALL CROSS(W,U,EJ)		HYLIM
	CALL CROSS(U,V,T)		HYLIM
	EIU = EI(1)*U(1) + EI(2)*U(2) + EI(3)*U(3)		HYLIM
	G = C*EIU		HYLIM
C			HYLIM
	DO 55 IT = 1,ITER		HYLIM
	EVM = 0.0		HYLIM
	EVZ = 0.0		HYLIM
	ZEZ = 0.0		HYLIM
	USEV = .FALSE.		HYLIM
	DO 15 I = 1,3		HYLIM
	E(I) = HYFCN(BD(I+16),Z(I),BD(I+1),POW)		HYLIM
	EV(I) = E(I)*V(I)		HYLIM
	IF(EV(I).EQ.0.0)USEV = .TRUE.		HYLIM
	IF(DABS(EV(I)).GT.EVM)EVM = DABS(EV(I))		HYLIM
	EZ(I) = E(I)*Z(I)		HYLIM
	IF(DABS(EZ(I)).GT.EVZ)EVZ = DABS(EZ(I))		HYLIM
15	ZEZ = ZEZ + Z(I)*EZ(I)		HYLIM
	RHO = ZEZ**PP		HYLIM
	DO 20 I = 1,3		HYLIM
20	Z(I) = Z(I)/RHO		HYLIM
	IF(PASS)GO TO 60		HYLIM
	RHOZ = ZEZ/RHO		HYLIM
	RHOV = EVM*RHOZ/RHO		HYLIM
	RHOZ = EVZ*RHOZ		HYLIM
	IF(.NOT.USEV)GO TO 30		HYLIM
	RHOV = 1.0		HYLIM
	DO 25 I = 1,3		HYLIM
25	EV(I) = TV(I)		HYLIM
C	WHAT IF NO TV IS 0 AND EV ARE ALL 0 ?		HYLIM
	30 DO 35 I = 1,3		HYLIM

EV(I) = EV(I)/RHOV	HYLIM
35 EZ(I) = EZ(I)/RHOZ	HYLIM
C SET UP MATRIX	HYLIM
CALL CROSS(EV, T, SM(1,1))	HYLIM
CALL CROSS(T, EZ, SM(1,2))	HYLIM
CALL CROSS(EZ, EV, SM(1,3))	HYLIM
TZV = T(1)*SM(1,3) + T(2)*SM(2,3) + T(3)*SM(3,3)	HYLIM
TZ = T(1)*Z(1) + T(2)*Z(2) + T(3)*Z(3)	HYLIM
ZEV = Z(1)*EV(1) + Z(2)*EV(2) + Z(3)*EV(3)	HYLIM
IF(TZV.EQ.0.0)STOP 39	HYLIM
ZEV = ZEV/TZV	HYLIM
Q(1) = 0.0	HYLIM
Q(2) = -ZEV	HYLIM
IF(.NOT.USEEV)Q(2) = Q(2)/PI	HYLIM
Q(3) = (G - TZ)/TZV	HYLIM
CALL MAT31(SM,Q,S)	HYLIM
SS = 0.0	HYLIM
ZZ = 0.0	HYLIM
DO 50 I = 1,3	HYLIM
SS = SS + DABS(S(I))	HYLIM
IF(DABS(Z(I)).LT.0.1*BD(I+1))GO TO 45	HYLIM
IF(DABS(S(I)).GT.DABS(Z(I)))S(I) = DSIGN(0.5*Z(I),S(I))	HYLIM
45 Z(I) = Z(I) + S(I)	HYLIM
IF(DABS(Z(I)).GT.BD(I+1))Z(I) = DSIGN(BD(I+1),Z(I))	HYLIM
50 ZZ = ZZ + DABS(Z(I))	HYLIM
IF(SS.LT.1.0E-10*ZZ)PASS = .TRUE.	HYLIM
55 CONTINUE	HYLIM
C	HYLIM
60 A = (EI(1)*Z(1) + EI(2)*Z(2) + EI(3)*Z(3))/EIU	HYLIM
B = (EJ(1)*Z(1) + EJ(2)*Z(2) + EJ(3)*Z(3))/EIU	HYLIM
RETURN	HYLIM
END	HYLIM

	SUBROUTINE HYLPR(J1,J2, ID,C,S,E,T)		HYLPR
C	IMPLICIT REAL*8(A-H,O-Z)	REV IV	02/07/87HYLPR
	DIMENSION ID(16),C(16),S(9,8),E(9).T(7)		HYLPR
	C LINEAR PROGRAM ROUTINE USING SIMPLEX METHOD		HYLPR
	C J1 = J2 FORCED PIVOT ON COLUMN J1		HYLPR
	CALCULATE COSTS		HYLPR
	J = J1		HYLPR
	IF(J.EQ.J2)GO TO 30		HYLPR
	10 DO 20 L = 1,7		HYLPR
	T(L) = -C(L)		HYLPR
	IF(C(L).EQ.10.)GO TO 20		HYLPR
	DO 15 I = 1,9		HYLPR
	15 T(L) = T(L) + S(I,L)*C(I+7)		HYLPR
	20 CONTINUE		HYLPR
	IF(J1.EQ.J2)GO TO 65		HYLPR
	C FIND PIVOT COLUMN		HYLPR
	DO 25 L = 1,7		HYLPR
	J = L		HYLPR
	IF(T(L).GT.0.0)GO TO 30		HYLPR
	25 CONTINUE		HYLPR
	GO TO 65		HYLPR
	C FIND PIVOT ROW		HYLPR
	30 K = 0		HYLPR
	DO 40 I = 1,9		HYLPR
	C SAVE PIVOT COLUMN		HYLPR
	E(I) = S(I,J)		HYLPR
	IF(S(I,J).LE.0.0)GO TO 40		HYLPR
	IF(K.EQ.0)GO TO 35		HYLPR
	IF(S(I,8).GE.Z*S(I,J))GO TO 40		HYLPR
	35 K = I		HYLPR
	Z = S(I,8)/S(I,J)		HYLPR
	40 CONTINUE		HYLPR
	C REPLACE COLUMNS		HYLPR
	IF(K.EQ.0)GO TO 65		HYLPR
	M = ID(J)		HYLPR
	ID(J) = ID(K+7)		HYLPR
	ID(K+7) = M		HYLPR
	Q = C(J)		HYLPR
	C(J) = C(K+7)		HYLPR
	C(K+7) = Q		HYLPR
	P = S(K,J)		HYLPR
	DO 45 I = 1,9		HYLPR
	45 S(I,J) = 0.0		HYLPR
	S(K,J) = 1.0		HYLPR
	DO 50 L = 1,8		HYLPR
	50 S(K,L) = S(K,L)/P		HYLPR
	E(K) = 1.0		HYLPR
	DO 60 I = 1,9		HYLPR
	IF(I.EQ.K)GO TO 60		HYLPR
	IF(E(I).EQ.0.0)GO TO 60		HYLPR
	DO 55 M = 1,8		HYLPR

55 $S(I,M) = S(I,M) - E(I)*S(K,M)$
60 CONTINUE
GO TO 10
65 RETURN
END

HYLPR
HYLPR
HYLPR
HYLPR
HYLPR

C TEST SHOULD PROBABLY BE AN EPSILON TEST, DABS(T(I)).GT.EPS	HYLPX
IF((T(I).NE.0.0).OR.(C(I).EQ.10.))GO TO 45	HYLPX
NZ = NZ + 1	HYLPX
IP(NZ) = I	HYLPX
45 CONTINUE	HYLPX
C SET PASS COUNT	HYLPX
NP = 1	HYLPX
IF(NZ.GT.0)NP = 2**NZ	HYLPX
C	HYLPX
DO 55 M = 1, NP	HYLPX
NM = NK	HYLPX
NK = NK + 1	HYLPX
DO 50 I = 1, 16	HYLPX
K = ID(I)	HYLPX
IF(K.GT.7)GO TO 50	HYLPX
W = 0.0	HYLPX
IF(I.GT.7)W = S(I-7,8)	HYLPX
V(K) = (W - A(K) + NM*V(K))/NK	HYLPX
50 CONTINUE	HYLPX
C LOOK FOR ALL SOLUTIONS	HYLPX
IF(M.EQ.NP)GO TO 55	HYLPX
J1 = IP(1)	HYLPX
IP(1) = IP(2)	HYLPX
IP(2) = J1	HYLPX
IF(T(J1).NE.0.0)GO TO 55	HYLPX
J2 = J1	HYLPX
CALL HYLPR(J1,J2, ID,C,S,E,T)	HYLPX
55 CONTINUE	HYLPX
RETURN	HYLPX
END	HYLPX

	SUBROUTINE HYNTR(BM,BN,TAB)		HYNTR
C		REV IV 02/07/87	HYNTR
	IMPLICIT REAL*8(A-H,O-Z)		HYNTR
	CALCULATIONS IN SEGMENT M'S REFERENCE		HYNTR
	DIMENSION BM(24),BN(24),TAB(8)		HYNTR
	COMMON/TEMPVS/D12(3,3),A(3,3),B(3,3),XMN(3),RLN(3),XMM(3),		HYNTR
	* AZ(3),R(3),Z(3),DNM(3,3),DAD(3,3),DBD(3,3),		HYNTR
	* BMD(3,3),TMP(3,3),S(5,6),ZVR(3),BZV(3),V(3),		HYNTR
	* ZM(3),VN(3),F(2),BV(3),DMY(34973)		80386
	P1 = 2.0		HYNTR
	P2 = 2.0		HYNTR
	IF(BM(1).LT.-2.0)P1 = -BM(1)		HYNTR
	IF(BN(1).LT.-2.0)P2 = -BN(1)		HYNTR
	C1 = P1 - 1.0		HYNTR
	CN = P2 - 1.0		HYNTR
C	DNM TRANSFORMS FROM M TO NN		HYNTR
	K = 8		HYNTR
	DO 15 J = 1,3		HYNTR
	DO 10 I = 1,3		HYNTR
	BMD(I,J) = 0.0		HYNTR
	IF(BM(1).LT.0.0)BMD(I,J) = BM(K)		HYNTR
	DNM(I,J) = D12(I,J)		HYNTR
	10 K = K + 1		HYNTR
	15 IF(BM(1).GT.0.0)BMD(J,J) = 1.0		HYNTR
	IF(BN(1).LT.0.0)CALL DOT33(BN(8),D12,DNM)		HYNTR
	ALP = TAB(1)		HYNTR
	BET = TAB(2)		HYNTR
	DO 20 I = 1,3		HYNTR
	ZM(I) = TAB(I+2)		HYNTR
	20 VN(I) = TAB(I+5)		HYNTR
C	PUT VECTORS INTO M'S REFERENCE		HYNTR
	CALL DOT31(BMD,ZM,Z)		HYNTR
	CALL DOT31(DNM,VN,V)		HYNTR
	DO 25 I = 1,3		HYNTR
	25 ZVR(I) = Z(I) - V(I) - BET*R(I)		HYNTR
C			HYNTR
	DO 40 ITER = 1,100		HYNTR
	CALL HYABF(BM,ZM,A,F(1))		HYNTR
	CALL HYABF(BN,VN,B,F(2))		HYNTR
	CALL DOT31(BMD,A(1,2),AZ)		HYNTR
	CALL DOT31(DNM,B(1,2),BV)		HYNTR
	CALL HYDAD(BMD,A,DAD)		HYNTR
	CALL HYDAD(DNM,B,DBD)		HYNTR
	CALL MAT31(DBD,R,S(1,5))		HYNTR
	CALL MAT31(DBD,ZVR,BZV)		HYNTR
	C2 = CN*ALP		HYNTR
C	SET UP S MATRIX		HYNTR
	S(4,4) = 0.0		HYNTR
	S(5,5) = 0.0		HYNTR
	S(4,6) = (1.0 - F(2))/P2 - V(1)*BZV(1) - V(2)*BZV(2) - V(3)*BZV(3)		HYNTR
	S(5,6) = (1.0 - F(1))/P1		HYNTR
	S(4,5) = -BV(1)*R(1) - BV(2)*R(2) - BV(3)*R(3)		HYNTR

	DOUBLE PRECISION FUNCTION HYPEN(BD,E,V)		HYPEN
C		REV IV	02/07/87HYPEN
C	POINT OF MAXIMUM PENETRATION		HYPEN
C	SOLVES FOR VALUE OF ALP USED BY PLELP		HYPEN
C	POWERS OF HYPERELLIPSOID MAY BE DIFFERENT		HYPEN
	IMPLICIT REAL*8(A-H,O-Z)		HYPEN
	DIMENSION BD(24),E(3),V(3)		HYPEN
	FX(A) = A**E(1)*V(1)+A**E(2)*V(2)+A**E(3)*V(3)-1.0		HYPEN
	L = 1		HYPEN
	VM = V(1)		HYPEN
	DO 10 I = 2,3		HYPEN
	IF (V(I).LE.VM) GO TO 10		HYPEN
	L = I		HYPEN
	VM = V(I)		HYPEN
10	CONTINUE		HYPEN
	A = V(1) + V(2) + V(3)		HYPEN
	A = 1.0/A**(1.0/E(L))		HYPEN
	DEL = A/2.0		HYPEN
	AP = 0.0		HYPEN
12	F = FX(A)		HYPEN
	IF (DABS(F).LT.1.D-08) GO TO 40		HYPEN
	IF (F) 16,40,14		HYPEN
14	IF (A-DEL.LE.0.0) DEL = A/2.0		HYPEN
	AP = A		HYPEN
	FP = F		HYPEN
	A = A - DEL		HYPEN
	GO TO 12		HYPEN
16	IF (AP.NE.0.0) GO TO 18		HYPEN
	A = A + DEL		HYPEN
	GO TO 12		HYPEN
18	AM = A		HYPEN
	FM = F		HYPEN
20	IF (FP.EQ.FM) GO TO 40		HYPEN
	DEL = -FM*(AP - AM)/(FP - FM)		HYPEN
	AN = AM + DEL		HYPEN
	IF (AN.EQ.A) GO TO 40		HYPEN
	A = AN		HYPEN
	F = FX(A)		HYPEN
	IF (DABS(F).LT.1.D-08) GO TO 40		HYPEN
	IF (F) 18,40,22		HYPEN
22	FP = F		HYPEN
	AP = A		HYPEN
	GO TO 20		HYPEN
40	HYPEN = A		HYPEN
	RETURN		HYPEN
	END		HYPEN

	SUBROUTINE HYREA (L,H,AREA,AB,BB)			
C	IMPLICIT REAL*8(A-H,O-Z)	REV IV	12/11/87	HYFIX
	DIMENSION H(2,2,5)			HYREA
	AREA = 0.0			HYFIX
	AB = 0.0			HYREA
	BB = 0.0			HYREA
	IF (L.LT.2) GO TO 20			HYREA
	DO 15 I = 1,L			HYFIX
	AR = H(1,1,I)*H(2,2,I) - H(1,2,I)*H(2,1,I)			HYFIX
	IF (AR.EQ.0.0) GO TO 5			HYFIX
	AB = AB + AR*(H(1,1,I) + H(1,2,I))			HYFIX
	BB = BB + AR*(H(2,1,I) + H(2,2,I))			HYFIX
	AREA = AREA + AR			HYFIX
	5 AR = H(1,2,I)*H(2,1,I+1) - H(1,1,I+1)*H(2,2,I)			HYREA
	IF (AR.EQ.0.0) GO TO 15			HYFIX
	AB = AB + AR*(H(1,1,I+1) + H(1,2,I))			HYFIX
	BB = BB + AR*(H(2,1,I+1) + H(2,2,I))			HYFIX
	AREA = AREA + AR			HYFIX
	15 CONTINUE			HYREA
	IF (AREA.LE.0.0) GO TO 20			HYFIX
	AREA = 3.0*AREA			HYFIX
	AB = AB/AREA			HYREA
	BB = BB/AREA			HYREA
	AREA = AREA/6.0			HYREA
C	20 RETURN			HYREA
	END			HYREA

	SUBROUTINE HYSOL(A,N,ND)		HYSOL
C	IMPLICIT REAL*8(A-H,O-Z)	REV IV	02/01/88MISDOT
	DIMENSION A(ND,6)		HYSOL
C	ASSUMES PIVOT ON DIAGONAL , BYPASS 0'S		MISDOT
	N1 = N + 1		HYSOL
	DO 20 L = 1,N		HYSOL
	IF(A(L,L).EQ.0.0)GO TO 20		HYSOL
	L1 = L + 1		HYSOL
	DO 10 J = L1,N1		HYSOL
10	A(L,J) = A(L,J)/A(L,L)		HYSOL
	IF(L.EQ.N)GO TO 20		HYSOL
	DO 21 I = L1,N		HYSOL
	IF(A(I,L).EQ.0.0)GO TO 21		HYSOL
	DO 15 J = L1,N1		HYSOL
15	A(I,J) = A(I,J) - A(I,L)*A(L,J)		HYSOL
21	CONTINUE		HYSOL
20	CONTINUE		HYSOL
	IF(N.EQ.1)GO TO 30		HYSOL
C	BACKUP		HYSOL
	DO 25 L = 2,N		HYSOL
	I = N1 - L		HYSOL
	L1 = I + 1		HYSOL
	DO 25 J = L1,N		HYSOL
25	A(I,N1) = A(I,N1) - A(I,J)*A(J,N1)		HYSOL
30	RETURN		HYSOL
	END		HYSOL

	SUBROUTINE HYVAL(A,U,R,BD,L)		HYVAL
C	IMPLICIT REAL*8(A-H,O-Z)	REV IV	12/11/87HYFIX
C	GIVEN A,U,R: COMPUTE A		HYVAL
	Z = A*U + R		HYVAL
	DIMENSION BD(24),U(3),R(3),RM(2)		HYFIX
	ONE = 1.0		HYFIX
	POW = -BD(1) - 2.0		HYVAL
C	ARE THESE THE CORRECT TESTS??		HYFIX
	TEST = -BD(1)*0.000001		HYFIX
	TESD = 0.000001		HYFIX
	CALL HYVBX(U,R,BD(2),M,RM)		HYVAL
	A = 0.0		HYFIX
	IF (M.LT.L) GO TO 50		HYFIX
C	THIS SHOULD NEVER HAPPEN - IMPLIES R IS OUTSIDE BOX		HYFIX
	A = RM(L)		HYVAL
	IF (DABS(A).LT.TESD) GO TO 50		HYFIX
	DEL = A/5.0		HYFIX
	NSTEP = 0		HYFIX
C	ITERATION LOOP		HYFIX
10	DEL = DEL/4.0		HYFIX
	NSTEP = NSTEP + 1		HYFIX
	IF (NSTEP.LT.100) GO TO 12		HYFIX
	WRITE(6,11) M,A,DEL,F1,F2,L,RM(1),RM(2),U,R,BD		HYFIX
11	FORMAT(' HYV ',I4,4F11.6,I3,2F11.6/4X,3F11.6,4X,3F11.6/ * 4(2X,7F10.4/))		HYFIX
	STOP 102		HYFIX
12	F2 = HYVFN(A,U,R,BD,POW)		HYFIX
	IF (DABS(F2).LT.TEST) GO TO 50		HYFIX
	IF (F2) 20,50,30		HYFIX
15	F2 = HYVFN(A,U,R,BD,POW)		HYFIX
	NSTEP = NSTEP + 1		HYFIX
	IF (NSTEP.LT.100) GO TO 17		HYFIX
	WRITE(6,11) M,A,DEL,F1,F2,L,RM(1),RM(2),U,R,BD		HYFIX
	STOP 103		HYFIX
17	IF (DABS(F2).LT.TEST) GO TO 50		HYFIX
	IF (F2) 20,50,35		HYFIX
20	IF (DSIGN(ONE,A).EQ.DSIGN(ONE,A+DEL)) GO TO 22		HYFIX
	A = A/2.0		HYFIX
	DL = -A		HYFIX
	GO TO 23		HYFIX
22	DL = DEL		HYFIX
	A = A + DEL		HYFIX
23	F1 = F2		HYFIX
	GO TO 15		HYFIX
25	F2 = HYVFN(A,U,R,BD,POW)		HYFIX
	NSTEP = NSTEP + 1		HYFIX
	IF (NSTEP.LT.100) GO TO 27		HYFIX
	WRITE(6,11) M,A,DEL,F1,F2,L,RM(1),RM(2),U,R,BD		HYFIX
	STOP 104		HYFIX
27	IF (DABS(F2).LT.TEST) GO TO 50		HYFIX
	IF (F2) 35,50,30		HYFIX
30	IF (DSIGN(ONE,A).EQ.DSIGN(ONE,A-DEL)) GO TO 32		HYFIX

A = A/2.0	HYFIX
DL = -A	HYFIX
GO TO 33	HYFIX
32 DL = -DEL	HYFIX
A = A - DEL	HYFIX
33 F1 = F2	HYFIX
GO TO 25	HYFIX
35 IF (F1.EQ.F2) GO TO 50	HYFIX
A = A + F2*DL/(F1 - F2)	HYFIX
IF (DABS(DEL).GT.TESD) GO TO 10	HYFIX
C	HYVAL
50 RETURN	HYVAL
END	HYVAL

	SUBROUTINE HYVBX(Q,S,B,M,RM)		HYVBX
C	IMPLICIT REAL*8(A-H,O-Z)	REV IV	02/07/87HYVBX
	DIMENSION Q(3),S(3),B(3),RM(2)		HYVBX
C	FINDS LIMITS OF BOX IN DIRECTION Q, Z = R*Q + S		HYVBX
	LOGICAL VAL		HYVBX
	M = 0		HYVBX
	C = -1.0		HYVBX
	DO 30 I = 1,3		HYVBX
	IF(Q(I).EQ.0.0)GO TO 30		HYVBX
	DO 25 K = 1,2		HYVBX
	VAL = .TRUE.		HYVBX
	D = C*B(I) - S(I)		HYVBX
	DO 10 J = 1,3		HYVBX
	IF(J.EQ.I)GO TO 10		HYVBX
	IF(DABS(D*Q(J) + S(J)*Q(I)).GT.DABS(B(J)*Q(I)))VAL = .FALSE.		HYVBX
C	IF(DABS(R*Q(J) + S(J)).GT.B(J))VAL = .FALSE.		HYVBX
10	CONTINUE		HYVBX
	IF(.NOT.VAL)GO TO 25		HYVBX
	R = D/Q(I)		HYVBX
	IF(M.EQ.0)GO TO 20		HYVBX
	DO 15 L = 1,M		HYVBX
	IF(R.EQ.RM(L)) GO TO 25		HYVBX
15	CONTINUE		HYVBX
20	M = M + 1		HYVBX
	RM(M) = R		HYVBX
25	C = -C		HYVBX
30	CONTINUE		HYVBX
	IF(M.EQ.0)GO TO 35		HYVBX
	IF(RM(1).LT.RM(2))GO TO 35		HYVBX
	R = RM(1)		HYVBX
	RM(1) = RM(2)		HYVBX
	RM(2) = R		HYVBX
35	RETURN		HYVBX
	END		HYVBX

	DOUBLE PRECISION FUNCTION HYVFN(A,U,R,B,P)		HYVFN
C		REV IV	12/11/87HYVFN
	IMPLICIT REAL*8(A-H,O-Z)		HYVFN
	DIMENSION U(3),R(3),B(24)		HYVFN
	F = -1.0		HYVFN
	DO 10 I = 1,3		HYVFN
	Z = A*U(I) + R(I)		HYVFN
	C = B(I+16)		HYVFN
	IF (P.GT.0.0) C = HYFCN(C,Z,B(I+1),P)		HYVFN
10	F = F + C*Z**2		HYVFN
	HYVFN = F		HYVFN
	RETURN		HYVFN
	END		HYVFN

19	V4(I,K) = 0.0	IMPLS2
18	DO 16 I=1,3	IMPLS2
	U2(I,M) = RPHI(I,M)*D(I,L,M)	IMPLS2
16	U2(I,N) = -RPHI(I,N)*D(I,L,N)	IMPLS2
	CALL DAUX(L)	IMPLS2
	DO 17 K=1,NGRND	IMPLS2
	DO 17 I=1,3	IMPLS2
	TLA(I,L,K) = SEGLA(I,K)	IMPLS2
17	TWA(I,L,K) = WMEGD(I,K)	IMPLS2
20	CONTINUE	IMPLS2
	CALL DOT33(D(1,1,M),TWA(1,1,M),TM)	IMPLS2
	CALL DOT33(D(1,1,N),TWA(1,1,N),TN)	IMPLS2
	CALL DOT31(D(1,1,M),WMEG(1,M),SM)	IMPLS2
	CALL DOT31(D(1,1,N),WMEG(1,N),SN)	IMPLS2
	DO 22 I=1,3	IMPLS2
	DO 21 K=1,3	IMPLS2
	T(I,K) = TM(I,K) - TN(I,K)	IMPLS2
21	TT(I,K) = T(I,K)	IMPLS2
	T(I,4) = SN(I) - SM(I)	IMPLS2
22	TT(I,4) = H(I)	IMPLS2
	IF (MODE.GE.0) CALL DSMSOL(T,3,3)	IMPLS2
	IF (MODE.GT.0) CALL DSMSOL(TT,3,3)	IMPLS2
	IF (MODE) 24,29,25	IMPLS2
24	ST = 0.0	IMPLS2
	STT = XDY(H,T,H)	IMPLS2
	GO TO 26	IMPLS2
25	ST = 1.0	IMPLS2
	STT = -(H(1)*TT(1,4) + H(2)*TT(2,4) + H(3)*TT(3,4))	IMPLS2
26	STT = (H(1)*T(1,4) + H(2)*T(2,4) + H(3)*T(3,4))/STT	IMPLS2
	DO 27 I=1,3	IMPLS2
27	T(I,4) = ST*T(I,4) + STT*TT(I,4)	IMPLS2
29	DO 30 K=1,NGRND	IMPLS2
	DO 30 I=1,3	IMPLS2
	DO 30 L=1,3	IMPLS2
	SEGLV(I,K) = SEGLV(I,K) + T(L,4)*TLA(I,L,K)	IMPLS2
30	WMEG(I,K) = WMEG(I,K) + T(L,4)*TWA(I,L,K)	IMPLS2
	IF (NPRT(3).NE.0) CALL PRINT(6HIMPLS2)	IMPLS2
	CALL ELTIME(2,28)	IMPLS2
	RETURN	IMPLS2
	END	IMPLS2

	DO 32 I=1,3	IMPULS
	U1(I,J) = 0.0	IMPULS
32	U2(I,J) = 0.0	IMPULS
	IF (NJNT.LE.0) GO TO 21	IMPULS
	DO 33 J=1,NJNT	IMPULS
	DO 33 I=1,3	IMPULS
	V1(I,J) = 0.0	IMPULS
33	V2(I,J) = 0.0	IMPULS
21	IF (NFLX.EQ.0) GO TO 23	IMPULS
	DO 22 J=1,NFLX	IMPULS
	DO 22 I=1,3	IMPULS
22	V4(I,J) = 0.0	IMPULS
C		IMPULS
C	REPLACE CALLS TO CONTACT AND VISPR WITH SINGLE CALL	IMPULS
C	AT FIRST CONTACT IF NOT CONSTRAINT.	IMPULS
C		IMPULS
23	IF (I1.NE.1) GO TO 34	IMPULS
	NT = NTPL(I2,I3)	IMPULS
	M1 = MPL(1,I2,I3)	IMPULS
	M2 = MPL(2,I2,I3)	IMPULS
	M3 = MPL(3,I2,I3)	IMPULS
	CALL PLELP(M2,M3,M1,I3,NT)	IMPULS
	IF (NTAB(NT+1).LT.0) GO TO 37	IMPULS
	K1 = M2	IMPULS
	K2 = M1	IMPULS
	GO TO 39	IMPULS
34	IF (I1.NE.3) GO TO 35	IMPULS
	NT = NTSEG(I2,I3)	IMPULS
	M1 = MSEG(1,I2,I3)	IMPULS
	M2 = MSEG(2,I2,I3)	IMPULS
	M3 = MSEG(3,I2,I3)	IMPULS
	CALL SEGSEG(I3,M1,M2,M3,NT)	IMPULS
	IF (NTAB(NT+1).LT.0) GO TO 37	IMPULS
	K1 = I3	IMPULS
	K2 = M2	IMPULS
	GO TO 39	IMPULS
35	IF (I1.NE.4) WRITE (6,36) I1,I2,I3	IMPULS
36	FORMAT('0 IMPROPER ARGUMENTS TO SUBROUTINE IMPULS'/	IMPULS
	* ' ARGUMENTS = ', 3I6 /	IMPULS
	* ' PROGRAM TERMINATED')	IMPULS
	IF (I1.NE.4) STOP 33	IMPULS
C		IMPULS
C	RECALL VISPR FOR JOINT STOP.	IMPULS
C		IMPULS
	IF (IABS(IPIN(I3)).NE.4) GO TO 25	IMPULS
	CALL EJOINT(I2,I3)	IMPULS
	GO TO 26	IMPULS
25	CALL VISPR(I2,I3)	IMPULS
26	K1 = IABS(JNT(I3))	IMPULS
	K2 = I3+1	IMPULS
	GO TO 39	IMPULS
C		IMPULS

C	SET UP SPECIAL U1,U2 FOR FIRST CONTACT OF CONSTRAINT.	IMPULS
C		IMPULS
37	KQ = -NTAB(NT+1)	IMPULS
	KQTEST = 1	IMPULS
	KQTYPE(KQ) = -IABS(KQTYPE(KQ))	IMPULS
	K1 = KQ1(KQ)	IMPULS
	K2 = KQ2(KQ)	IMPULS
	IF (K1.GT.NSEG) GO TO 38	IMPULS
	CALL MAT31(A13(1,1,2*KQ-1),QQ(1,KQ),U1(1,K1))	IMPULS
	CALL MAT31(A23(1,1,2*KQ-1),QQ(1,KQ),U2(1,K1))	IMPULS
38	IF (K2.GT.NSEG) GO TO 39	IMPULS
	CALL MAT31(A13(1,1,2*KQ),QQ(1,KQ),U1(1,K2))	IMPULS
	CALL MAT31(A23(1,1,2*KQ),QQ(1,KQ),U2(1,K2))	IMPULS
C		IMPULS
C	FINAL SETUP OF U1 AND U2	IMPULS
C		IMPULS
39	DO 40 J=1,NGRND	IMPULS
	DO 40 I=1,3	IMPULS
	U1(I,J) = U1(I,J)*RW(J)	IMPULS
40	U2(I,J) = U2(I,J)*RPHI(I,J)	IMPULS
	CALL DAUX(I1)	IMPULS
	IF (KQTEST.EQ.1) KQTYPE(KQ) = IABS(KQTYPE(KQ))	IMPULS
	IF (NPRT(10).NE.0) CALL PRINT(6HPREIMP)	IMPULS
	IF (I1.GT.3) GO TO 51	IMPULS
	IF (NPRT(10).NE.0) WRITE (6,42) R1I,R2I	IMPULS
42	FORMAT ('0'/(6G20.8))	IMPULS
	CALL CROSS(WMEG (1,K1),R1I(1),TEMP)	IMPULS
	CALL DOT31(D(1,1,K1),TEMP,DWR1(1))	IMPULS
	CALL CROSS(WMEG (1,K2),R2I(1),TEMP)	IMPULS
	CALL DOT31(D(1,1,K2),TEMP,DWR2(1))	IMPULS
	CALL CROSS(WMEGD(1,K1),R1I(1),TEMP)	IMPULS
	CALL DOT31(D(1,1,K1),TEMP,DWR3(1))	IMPULS
	CALL CROSS(WMEGD(1,K2),R2I(1),TEMP)	IMPULS
	CALL DOT31(D(1,1,K2),TEMP,DWR4(1))	IMPULS
	TVREL = 0.0	IMPULS
	TDV = 0.0	IMPULS
	DO 50 I=1,3	IMPULS
	VREL(I) = SEGLV(I,K1)+DWR1(I) - SEGLV(I,K2)-DWR2(I)	IMPULS
	DV (I) = SEGLA(I,K1)+DWR3(I) - SEGLA(I,K2)-DWR4(I)	IMPULS
	TVREL = TVREL + TTI(I)*VREL(I)	IMPULS
50	TDV = TDV + TTI(I)*DV (I)	IMPULS
	GO TO 53	IMPULS
51	CALL DOT31(D(1,1,K1),WMEG (1,K1),DWR1(1))	IMPULS
	CALL DOT31(D(1,1,K2),WMEG (1,K2),DWR2(1))	IMPULS
	CALL DOT31(D(1,1,K1),WMEGD(1,K1),DWR3(1))	IMPULS
	CALL DOT31(D(1,1,K2),WMEGD(1,K2),DWR4(1))	IMPULS
	TVREL = 0.0	IMPULS
	TDV = 0.0	IMPULS
	DO 52 I=1,3	IMPULS
	VREL(I) = DWR1(I) - DWR2(I)	IMPULS
	DV (I) = DWR3(I) - DWR4(I)	IMPULS
	TVREL = TVREL + TTI(I)*VREL(I)	IMPULS

```

52 TDV  = TDV  + TTI(I)*DV  (I)           IMPULS
53 ALPHA = 0.0                             IMPULS
C                                             IMPULS
C NOTE: CREST IS SUPPLIED AS (1+E)/2 WHERE E IS THE CLASSICAL IMPULS
C COEFFICIENT OF RESTITUTION BUT WITH A RANGE OF -1 TO +1.      IMPULS
C CREST HAS A RANGE OF 0 TO +1 WHERE 0 (E=-1) REPRESENTS NO IMPULSE. IMPULS
C                                             IMPULS
IF (TDV.NE.0.0) ALPHA = -2.0*CREST*TVREL/TDV          IMPULS
IF (NPRT(10).NE.0) WRITE (6,42) DWR1,DWR2,DWR3,DWR4 , IMPULS
*                                             TTI ,VREL ,DV , IMPULS
*                                             TVREL ,TDV ,CREST ,ALPHA IMPULS
DO 60 J=1,NGRND                                       IMPULS
DO 60 I=1,3                                           IMP' LS
SEGLV(I,J) = SEGLV(I,J) + ALPHA*SEGLA(I,J)          IMPULS
60 WMEG (I,J) = WMEG (I,J) + ALPHA*WMEGD(I,J)       IMPULS
IF (NPRT(10).NE.0) CALL OUTPUT(1)                   IMPULS
IF (NPRT( 3).NE.0) CALL PRINT(6HIMPULS)             IMPULS
CALL ELTIME(2,27)                                    IMPULS
99 RETURN                                           IMPULS
END                                                  IMPULS

```

```

SUBROUTINE INITIAL                                REV IV    07/24/86SLIP
C PERFORMS CARD INPUT AND COMPUTATIONS FOR INITIAL
C POSITIONING OF THE CRASH VICTIM'S BODY SEGMENTS.
C
  IMPLICIT REAL*8(A-H,O-Z)
  COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,
*               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG
  COMMON/SGRNITS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),
*               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)
  COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60),
*               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),
*               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)
  COMMON/VPOSTN/ ZPLT(3),SPLT(3),AXV(3,6),VATAB(6,501,6),
*               VTO(6),VDT(6),TIMEV(6),OMEGV(6),NVTAB(6),INDXV(6)
  COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),
*               BLTTTL(5,8),PLTTTL(5,30),BAGTTL(5,6),SEG(30),
*               JOINT(30),CGS(30),JS(30)
  COMMON/CEULER/ IEULER(30),HIR(3,3,90),ANG(3,30),ANGD(3,30),
*               FE(3,30),TQE(3,30),CONST(5,30)
  REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTTL,BAGTTL,SEG,JOINT
  LOGICAL*1 CGS,JS
  COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),
*               UNITL,UNITM,UNITT,GRAVTY(3),TWOPI
  COMMON/TEMPVS/ TMP(140),WMGDEG(3,30),T(3),S(3),A(3,2),Z(3,3)
*               ,DMY(34862)
C           NOTE : CHAIN ALSO USES TEMPVS.
  DIMENSION YPR(3,30) , IYPR(4,30)
C
C INPUT CARD G.1.A (PLOT COORDINATES OF VEHICLE REFERENCE ORIGIN)
C
  READ(5,22) ZPLT,I1,J1,I2,J2,I3
22 FORMAT(3F10.0,5I4)
  S(1) = 10.0
  S(2) = 6.0
  S(3) = 1.0
C
C IF J1#0, INPUT CARD G.1.B (PLOT SCALING INPUT)
C
  IF (J1.NE.0) READ (5,22) S
  SPLT(1) = 1.0/S(3)
  SPLT(2) = 1.0/S(3)
  SPLT(3) = -(S(1)/S(2))/S(3)
  WRITE (6,23) NPG,ZPLT,I1,J1,I2,J2,I3,S
  NPG=NPG+1
23 FORMAT('1 SUBROUTINE INITIAL INPUT',98X,'PAGE',I5/120X,'CARD G.1'/
* ' ZPLT(X) ZPLT(Y) ZPLT(Z) I1 J1 I2 J2 I3',
* ' SPLT(1) SPLT(2) SPLT(3)'/3F10.0,5I6,3F10.2)
C
C INPUT CARDS G.2.A - G.2.N
C
C INITIAL LINEAR POSITION (IN) AND (IF I3=1) VELOCITY (IN/SEC)

```

C	OF EACH BASE BODY SEGMENT. IF I3=0, VELOCITY WILL BE SET TO	INITAL
C	INITIAL VELOCITY OF VEHICLE. INPUTS IN INERTIAL REFERENCE.	INITAL
C		INITAL
	DO 37 J=1,NSEG	INITAL
	IF(J.GT.1.AND.IABS(JNT(J-1)).GT.0) GO TO 37	INITAL
	READ(5,24) (SEGLP(I,J),I=1,3),(SEGLV(I,J),I=1,3)	INITAL
24	FORMAT (6F10.0 , 4I3)	INITAL
	IF(I3.GT.0) GO TO 37	INITAL
	DO 36 I=1,3	INITAL
36	SEGLV(I,J) = SEGLV(I,NVEH)	INITAL
37	CONTINUE	INITAL
C		INITAL
C	INPUT CARDS G.3.A - G.3.N	INITAL
C		INITAL
C	FOR EACH BODY SEGMENT SUPPLY YAW, PITCH AND ROLL (DEGREES)	INITAL
C	AND (IF I3=1) THE ANGULAR VELOCITY IN LOCAL REFERENCE (DEG/SEC).	INITAL
C	IF I3=0, THE ANGULAR VELOCITY (BLANK ON INPUT CARDS) WILL BE SET	INITAL
C	EQUAL TO THE INITIAL ANGULAR VELOCITY OF THE VEHICLE.	INITAL
C		INITAL
	FIRST = 0.0	INITAL
	DO 40 J=1,NSEG	INITAL
	READ (5,24) (YPR(I,J),I=1,3),(WMGDEG(I,J),I=1,3),(IYPR(I,J),I=1,4)	INITAL
	ID1 = IYPR(1,J)	INITAL
	DO 38 I=1,3	INITAL
	IF (ID1.EQ.0) IYPR(I,J) = I	INITAL
38	WMEG(I,J) = WMGDEG(I,J)*RADIAN	INITAL
	IF (ID1.GE.0) GO TO 60	INITAL
C		INITAL
C	READ CARD G.3.J2 FOR SEGMENT NO. J WHEN IYPR(1,J) IS NEGATIVE.	INITAL
C		INITAL
	READ (5,24) A,II,IK,JJ,JK	INITAL
	IJ = II	INITAL
	LK = IK	INITAL
	DO 54 K=1,2	INITAL
	IF (IJ.GT.0) GO TO 52	INITAL
	DO 51 I=1,3	INITAL
51	Z(I,LK) = A(I,K)	INITAL
	GO TO 53	INITAL
52	DA1 = A(1,K)*RADIAN	INITAL
	DA2 = A(2,K)*RADIAN	INITAL
	SA1 = DSIN(DA1)	INITAL
	SA2 = DSIN(DA2)	INITAL
	CA1 = DCOS(DA1)	INITAL
	CA2 = DCOS(DA2)	INITAL
	IJ1 = IJ+1	INITAL
	IJ2 = IJ+2	INITAL
	IF (IJ1.GT.3) IJ1= IJ1-3	INITAL
	IF (IJ2.GT.3) IJ2= IJ2-3	INITAL
	SGN = 1.0	INITAL
	IF (SA1.LT.0.0 .AND. CA2.LT.0.0) SGN = -1.0	INITAL
	Z(IJ ,LK) = SGN*SA1*CA2	INITAL
	Z(IJ1,LK) = SGN*SA1*SA2	INITAL

```

      Z(IJ2,LK) = SGN*CA1*CA2
53 IJ = JJ
54 LK = JK
      ZDOTIJ = Z(1,IK)*Z(1,JK) + Z(2,IK)*Z(2,JK) + Z(3,IK)*Z(3,JK)
      ZDOTII = Z(1,IK)*Z(1,IK) + Z(2,IK)*Z(2,IK) + Z(3,IK)*Z(3,IK)
      RATIO = ZDOTIJ/ZDOTII
      DO 55 I=1,3
55 Z(I,JK) = Z(I,JK) - RATIO*Z(I,IK)
      LK = 6-IK-JK
      IT = MOD(JK-IK+3,3)
      IF (IT.EQ.1) CALL CROSS(Z(1,IK),Z(1,JK),Z(1,LK))
      IF (IT.EQ.2) CALL CROSS(Z(1,JK),Z(1,IK),Z(1,LK))
      DO 57 K=1,3
      IYPR(K,J) = 4-K
      SUM = 0.0
      DO 56 I=1,3
56 SUM = SUM + Z(I,K)**2
      SQUM = DSQRT(SUM)
      DO 57 I=1,3
57 D(K,I,J) = Z(I,K)/SQUM
      CALL YPRDEG (D(1,1,J),YPR(1,J))
      IF (FIRST.EQ.0.0) WRITE (6,58)
58 FORMAT('0 INITIAL ANGULAR ROTATIONS COMPUTED FROM CARDS G.3.J2'//
*        ' SEGMENT',10X,'SEGMENT PRIMARY AXIS',
*        12X,'SEGMENT SECONDARY AXIS',30X,'ANGULAR ROTATIONS (DEG)'/
*        ' NO. SEG',9X,'A1',8X,'A2',8X,'A3',11X,'B1',8X,'B2',8X,
*        'B3',7X,'II IK JJ JK',9X,'YAW',6X,'PITCH',5X,'ROLL'//
      FIRST = 1.0
      WRITE (6,59) J,SEG(J),A,II,IK,JJ,JK,(YPR(I,J),I=1,3)
59 FORMAT (I4,1X,A4,3X,3F10.3,3X,3F10.3,3X,4I4,3X,3F10.3)
60 M = IYPR(4,J)
      IF (M.EQ.0) M=NGRND
      IF (M.GE.J .AND. M.LE.NSEG) STOP 24
      IF (J.EQ.1) GO TO 80
      IF (M.LT.0 .AND. -M.NE.IABS(JNT(J-1))) STOP 25
80 CALL DRCIJK (D,YPR,IYPR,HT,J)
      IF (I3.GT.0) GO TO 40
      CALL DOT31(D(1,1,NVEH),WMEG(1,NVEH),T)
      CALL MAT31(D(1,1,J),T,WMEG(1,J))
      DO 39 I=1,3
39 WMGDEG(I,J) = WMEG(I,J)/RADIAN
40 CONTINUE
      CALL VEHPOS
      IF(NJNT.EQ.0) GOTO 41
      CALL CHAIN(0)
      CALL EJOINT(1,0)
      DO 62 J=1,NJNT
      IF(IABS(IPIN(J)).NE.4) GOTO 62
      IF(IEULER(J).NE.2) GOTO 62
      DA1 = ANG(2,J) + CONST(2,J)
      CONST(4,J) = DCOS(DA1)
      CONST(5,J) = DSIN(DA1)

```


62	CONTINUE	JDRIFT
C		INITAL
C	OUTPUT INITIAL BODY SEGMENT POSITIONS.	INITAL
C		INITAL
41	WRITE (6,42) UNITL,UNITL,UNITT	JDRIFT
42	FORMAT('0 INITIAL POSITIONS (INERTIAL REFERENCE)',70X,'CARDS G.2')//INITAL	
*	/' SEGMENT',11X,'LINEAR POSITION (' ,A4,')',	INITAL
*	14X,'LINEAR VELOCITY (' ,A4,',' /',A4,')'/'	AFREVS
*	' NO. SEG',2(9X,'X',11X,'Y',11X,'Z',5X))	INITAL
	WRITE (6,43) (J,SEG(J),(SEGLP(I,J),I=1,3),(SEGLV(I,J),I=1,3)	INITAL
*	,J=1,NSEG)	INITAL
43	FORMAT(I4,1X,A4,3X,3F12.5,3X,3F12.5)	INITAL
	WRITE (6,44) UNITT	INITAL
44	FORMAT('0 INITIAL ANGULAR ROTATION AND VELOCITY',71X,'CARDS G.3')//INITAL	
*	' SEGMENT',11X,'ANGULAR ROTATION (DEG)',	AFREVS
*	14X,'ANGULAR VELOCITY (DEG/' ,A4,')'/'	INITAL
*	' NO. SEG',8X,'YAW',8X,'PITCH',7X,'ROLL',	INITAL
*	13X,'X',11X,'Y',11X,'Z',15X,'IYPR')	INITAL
	WRITE (6,46) (J,SEG(J),(YPR(I,J),I=1,3),(WMGDEG(I,J),I=1,3),	INITAL
*	(IYPR(I,J),I=1,4),J=1,NSEG)	INITAL
46	FORMAT(I4,1X,A4,3X,3F12.5,3X,3F12.5,3X,4I4)	INITAL
	IF (I3.EQ.0) WRITE (6,45)	INITAL
45	FORMAT('0 LINEAR AND ANGULAR VELOCITIES HAVE BEEN SET EQUAL TO THEINITAL	
*	INITIAL VEHICLE VELOCITIES.')	INITAL
	IF (NHRNSS.NE.0) CALL HBPLAY	INITAL
	IF (I1.EQ.15) CALL EQUILB (YPR,IYPR)	INITAL
	CALL UNITI(0)	JDRIFT
	CALL ROTATE	INITAL
	CALL ELTIME(2,2)	INITAL
	RETURN	INITAL
	END	INITAL

```

SUBROUTINE INTERS(A,B,XM,T,X,V,AX)
C
C DETERMINES INTERSECTION OF ELLIPSOIDS
C   X'AX = 1
C   (X'-M')B(X-M) = 1
C WHERE A AND B ARE ELLIPSOID MATRICES
C IF T ENTERS AS +1.0 , A IS EXTERNAL TO B AND
C   AS -1.0 , A IS INTERNAL TO B.
C
C IF V ENTERS AS NON-ZERO, WILL USE PREVIOUS VALUE FOR START.
C (AX) RETURNS AS (A)*(X).
C
C RETURNS T>1 - NO INTERSECTION
C   T<1 - INTERSECTION IN WHICH CASE X WILL
C   CONTAIN COORDINATES OF CONTACT OF
C   CONTRACTED ELLIPSOIDS.
C
C IMPLICIT REAL*8 (A-H,O-Z)
C DIMENSION A(3,3),B(3,3),XM(3),X(3)
C DIMENSION C(3,4),Z(3),BM(3),AX(3),AM(3)
C EQUIVALENCE (Z(1),C(1,4))
C COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),
*   UNITL,UNITM,UNITT,GRAVITY(3),TWOPI
C   INITIALIZATION
C   EVALUATE BM,M'AM,M'BM
C   SET N=0, V=M'BM/M'AM
C
C N = 0
C BMM = 0.0
C AMM = 0.0
C DO 11 I=1,3
C   BM(I) = 0.0
C   AM(I) = 0.0
C DO 10 J=1,3
C   IF (DABS(A(I,J)).LT.EPS(20)) A(I,J) = 0.0
C   AM(I) = AM(I) + A(I,J)*XM(J)
C   IF (DABS(B(I,J)).LT.EPS(20)) B(I,J) = 0.0
10  BM(I) = BM(I) + B(I,J)*XM(J)
C   BMM = BMM + XM(I)*BM(I)
11  AMM = AMM + XM(I)*AM(I)
C   IF (V.EQ.0.0) V=T*DSQRT(BMM/AMM)
C   IDONE = 0
C
C   NEWTON-RAPHSON ITERATION FOR
C   G(V) = FA(V)-FB(V) = 0
C   SOLVE (VA+B)X = BM FOR X
C
C   ITER = 0
20  ITER = ITER+1
C   DO 22 I=1,3
C   DO 21 J=1,3
21  C(I,J) = V*A(I,J) + B(I,J)
22  Z(I) = BM(I)
C   CALL DSMSOL(C,3,3)
C
C   EVALUATE AX

```

C		$FA(V) = X'AX$	INTERS
C		$FB(V) = -V(X' - M')AX$	INTERS
	FA = 0.0		INTERS
	FB = 0.0		INTERS
	CALL MAT31(A,Z,AX)		INTERS
	DO 30 I=1,3		INTERS
	X(I) = Z(I)		INTERS
	FA = FA+X(I)*AX(I)		INTERS
30	FB = FB+(X(I)-XM(I))*AX(I)		INTERS
	FB = -V*FB		INTERS
	IF (T.LT.0.0) FA = 1.0/FA		INTERS
	IF (IDONE.EQ.1) GO TO 60		INTERS
C		TEST FOR INTERSECTION	INTERS
	IF (FA-FB) 32,60,31		INTERS
C		IF FA>FB>1, NO INTERSECTION	INTERS
31	IF (T.GT.0.0.AND.FB.LT.1.0) GO TO 40		INTERS
	IF (T.LT.0.0.AND.FA.GT.1.0) GO TO 40		INTERS
	IF (N.EQ.0) GO TO 60		INTERS
	GO TO 62		INTERS
C		IF FA<FB<1, INTERSECTION	INTERS
32	IF (T.GT.0.0.AND.FB.LE.1.0) N=1		INTERS
	IF (T.LT.0.0.AND.FA.GE.1.0) N=1		INTERS
C		SOLVE (VA+B)Z = AX FOR Z	INTERS
40	DO 42 I=1,3		INTERS
	DO 41 J=1,3		INTERS
41	C(I,J) = V*A(I,J) + B(I,J)		INTERS
42	Z(I) = AX(I)		INTERS
	CALL DSMSOL(C,3,3)		INTERS
C		F'A(V) = -2X'AZ	INTERS
	CALL MAT31 (A,Z,AX)		INTERS
	FPA = X(1)*AX(1)		INTERS
	* + X(2)*AX(2)		INTERS
	* + X(3)*AX(3)		INTERS
	FPA = -(FPA+FPA)		INTERS
C		DV = -G(V)/G'(V)	INTERS
	DV = 1.0 + V		INTERS
	IF (T.LT.0.0) DV = V-FA**2		INTERS
	DV = (FB-FA)/(DV*FPA)		INTERS
	IF (ITER.GE.50) GO TO 62		INTERS
C		TEST FOR CONVERGENCE	INTERS
	IF (T*(V+DV).LE.0.0) DV = -0.5*V		INTERS
	V = V+DV		INTERS
	DV = DABS(DV/V)		INTERS
	IF (DV.LE.EPS(12)) IDONE=1		INTERS
	GO TO 20		INTERS
C		FA(V) = FB(B), RETURN	INTERS
60	IF (T.LT.0.0) FA = 1.0/FB		INTERS
	T = DSQRT(FA)		INTERS
	IF (FA.GT.1.0) GO TO 61		INTERS
	N = 1		INTERS
	GO TO 71		INTERS
61	IF (N.EQ.0) GO TO 71		INTERS

```
62 WRITE (6,63)
63 FORMAT(' INTERS ITERATION DID NOT CONVERGE')
71 CONTINUE
   RETURN
   END
```

```
INTERS
INTERS
INTERS
INTERS
INTERS
```


60	FORMAT(3F12.0,2I12)	WINDOP
	TAB(J2-1) = DFLOAT(NSV)	WINDOP
	TAB(J2) = DFLOAT(NSR)	WINDOP
	IF (TAB(J1).EQ.0.0) GOTO 22	WINDOP
	WRITE(6,23) (TAB(J),J=J1,J2-2),NSV,SEG(NSV),NSR,SEG(NSR)	WINDOP
23	FORMAT(' SPEC. HEAT RATIO SONIC VEL. ABS. PRESS.',7X,	WINDOP
	* 'SEGMENT REF. SEGMENT',/3F15.4,2(I11,A4)//)	WINDOP
	J1=J2+1	WINDOP
	GOTO 30	MISC
22	WRITE (6,18) (TAB(J),J=J1,J2)	KINPUT
17	FORMAT(6F12.0)	KINPUT
18	FORMAT(10X,'D0',13X,'D1',13X,'D2',13X,'D3',8X,'REF. SEGMENT',	WINDOP
	* /5F15.4//)	WINDOP
	J1 = J2+1	KINPUT
C		KINPUT
C	INPUT CARD E.6.C - NTMPTS	KINPUT
C		KINPUT
	READ (5,11) NTMPTS	KINPUT
	WRITE (6,19) NTMPTS	KINPUT
19	FORMAT('0 WIND FORCE TABLES FOR ',I6,' TIME POINTS.'//	KINPUT
	* 11X,'T',14X,'FX(T)',15X,'FY(T)',15X,'FZ(T)' /)	KINPUT
	TAB(J1) = NTMPTS	KINPUT
	J1 = J1+1	KINPUT
	J2 = J1+4*NTMPTS-1	KINPUT
C		KINPUT
C	INPUT CARDS E.6.D-E.6.N - NTMPTS CARDS OF T,FX(T),FY(T),FZ(T)	KINPUT
C		KINPUT
	READ (5,20) (TAB(J),J=J1,J2)	KINPUT
	WRITE (6,21) (TAB(J),J=J1,J2)	KINPUT
20	FORMAT(4F12.0)	KINPUT
21	FORMAT(3X,F12.6,3G20.6)	KINPUT
	J1 = J2+1	KINPUT
30	CONTINUE	KINPUT
31	IF (NJNTE.LE.0) GO TO 51	KINPUT
	DO 50 K=1,NJNTE	KINPUT
C		KINPUT
C	INPUT CARD E.7.A - FUNCTION NO. AND TITLE	KINPUT
C		KINPUT
	READ (5,12) I,(KTITLE(J),J=1,5)	KINPUT
	WRITE (6,32) I,(KTITLE(J),J=1,5),I,J1,NPG	PAGE
	NPG=NPG+1	PAGE
32	FORMAT('1 JOINT FORCE FUNCTION NO.',I4,4X,5A4,10X,'NTI(',I2,') =',	KINPUT
	* I5,45X,'PAGE',I5/120X,'CARDS E.7'//)	PAGE
	IF (I.LE.0.OR.I.GT.50) WRITE (6,14)	KINPUT
	IF (I.LE.0.OR.I.GT.50) STOP 12	KINPUT
	IF (NTI(I).NE.0) WRITE (6,15) I	KINPUT
	NTI(I) = J1	KINPUT
	DO 33 J=1,5	KINPUT
33	JTITL(J,I) = KTITL(J)	KINPUT
C		KINPUT
C	INPUT CARD E.7.B - D0,D1,D2,D3,D4 (FOR NOW A BLANK CARD).	KINPUT
C		KINPUT

```

J2 = J1+4
READ (5,17) (TAB(J),J=J1,J2)
WRITE (6,18) (TAB(J),J=J1,J2)
J1 = J2+1
C
C INPUT CARD E.7.C - NTHETA,NPHI
C
READ (5,11) NTHETA,NPHI
TAB(J1 ) = NTHETA
TAB(J1+1) = NPHI
J1 = J1+2
IF (NTHETA.LT.0) GO TO 38
DO 35 J=1,NTHETA
35 TH(J) = DFLOAT(J-1)*180.0/DFLOAT(NTHETA-1)
WRITE (6,36) NTHETA,NPHI,(TH(J),J=2,NTHETA)
36 FORMAT('O FUNCTION IS TABULAR FOR' ,I3,' X',I3,' VALUES OF THETA AKINPUT
*ND PHI'//30X,'THETA'/5X,'PHI',5X,'THETA0',F16.3,4F20.3/
* (15X,5F20.3))
37 FORMAT(F9.2,F10.3,5G20.7/(19X,5G20.7))
GO TO 40
38 NPOLY = -NTHETA -1
WRITE (6,39) NPOLY,NPHI,(BLANK,J,J=1,NPOLY)
39 FORMAT('O FUNCTION IS COEFFICIENTS OF' ,I3,' ORDER POLYNOMIALS IN KINPUT
*(THETA-THETA0) FOR',I3,' VALUES OF PHI.'//
* 27X,'COEFFICIENTS OF (THETA-THETA0)**N'/
* 5X,'PHI',5X,'THETA0',7X,5(A4,'N =',I2,11X)/(26X,A4,'N =',I2,11X,KINPUT
* A4,'N =',I2,11X,A4,'N =',I2,11X,A4,'N =',I2,11X,A4,'N =',I2) ) KINPUT
40 WRITE (6,21)
DO 49 I=1,NPHI
PHIDEG = DFLOAT(I-1)*360.0/DFLOAT(NPHI) - 180.0
C
C INPUT CARDS E.7.D - E.7.N NPHI SETS WITH NTHETA ITEMS PER SET. KINPUT
C EACH SET I IS FOR PHI(I) = -180 +(I-1)*360/NPHI DEGREES AND KINPUT
C ASSUMES DATA FOR PHI(NPHI+1) = 180 IS SAME AS PHI(1) = -180. KINPUT
C
J2 = J1 + IABS(NTHETA) -1
READ (5,17) (TAB(J),J=J1,J2)
WRITE (6,37) PHIDEG,(TAB(J),J=J1,J2)
IF (NTHETA.LT.0) TAB(J1) = TAB(J1)*RADIAN
IF (NTHETA.LT.0) GO TO 49
C
C FOR TABULAR DATA, FILL IN ZERO VALUES WITH INTERPOLATED NEGATIVE KINPUT
C VALUES. OVERWRITE VALUE IN FIRST COLUMN (SUPPLIED AS THETA0) WITH KINPUT
C VALUE FOR THETA = 0 AND ALL OTHER ZERO VALUES. KINPUT
C
THETA0 = TAB(J1)
IF (THETA0.EQ.0.0) GO TO 49
JJ = THETA0*DFLOAT(NTHETA-1)/180.0 + 1.0 + EPS(6)
JJ1 = J1+JJ
IERROR = 0
IF (JJ1.GT.J2) IERROR = 1
IF (TAB(JJ1).LE.0.0) IERROR = 2

```

IF (IERROR.NE.0) GO TO 46	KINPUT
DO 45 J=1,JJ	KINPUT
J1J = J1+J-1	KINPUT
IF (J.NE.1.AND.TAB(J1J).GT.0.0) IERROR = 3	KINPUT
45 TAB(J1J) = TAB(JJ1)*(TH(J)-THETA0)/(TH(JJ+1)-THETA0)	KINPUT
46 IF (IERROR.NE.0) WRITE (6,47) IERROR	KINPUT
47 FORMAT('0 INPUT ERROR. INCONSISTENT VALUE OF THETA0. IERROR =',I2,	KINPUT
* ' PROGRAM TERMINATED.')	KINPUT
IF (IERROR.NE.0) STOP 13	KINPUT
49 J1 = J2+1	KINPUT
50 CONTINUE	KINPUT
51 MXTB1 = J1-1	KINPUT
RETURN	KINPUT
END	KINPUT

C	TICX1 = -TICX1	LOGAXS
	TICY1 = - TICY1	LOGAXS
	TICX2 = -TICX2	LOGAXS
	TICXA = - TICXA	LOGAXS
	TICYA = -TICYA	LOGAXS
	TICXB = -TICXB	LOGAXS
	TICYB = - TICYB	LOGAXS
C		LOGAXS
50	COST = COST*SPDEC	LOGAXS
	SINT = SINT* SPDEC	LOGAXS
	TICX2 = TICXA	LOGAXS
	TICY2 = TICYA	LOGAXS
C		LOGAXS
	XD = XO	LOGAXS
	YD = YO	LOGAXS
	ND = 1	LOGAXS
	N = 0	LOGAXS
C		LOGAXS
C	*****GO TO START POS.*****	LOGAXS
	CALL PLOT(X0+TICXB,Y0+TICYB,3)	LOGAXS
	CALL PLOT(X0,Y0,2)	LOGAXS
C		LOGAXS
60	N = N+1	LOGAXS
	Q = XL(N)	LOGAXS
	IF(.NOT. REVERS) GO TO 65	LOGAXS
	M = 18-N	LOGAXS
	Q = 1.0-XL(M)	LOGAXS
65	X = XD + Q*COST	LOGAXS
	Y = YD + Q*SINT	LOGAXS
	CALL PLOT(X,Y,2)	LOGAXS
	CALL PLOT(X+TICX1,Y+TICY1,2)	LOGAXS
	CALL PLOT (X,Y,2)	LOGAXS
C		LOGAXS
	N = N+1	LOGAXS
	Q = XL(N)	LOGAXS
	IF(.NOT. REVERS) GO TO 75	LOGAXS
	M = 18-N	LOGAXS
	Q = 1.0 - XL(M)	LOGAXS
.5	X = XD + Q*COST	LOGAXS
	Y = YD + Q*SINT	LOGAXS
	CALL PLGT(X,Y,2)	LOGAXS
	CALL PLOT (X+TICX2,Y+TICY2,2)	LOGAXS
	CALL PLOT(X,Y,2)	LOGAXS
C		LOGAXS
	IF(N-16) 60,80,100	LOGAXS
C		LOGAXS
80	TICX2 = TICXB	LOGAXS
	TICY2 = TICYB	LOGAXS
	GO TO 60	LOGAXS
C		LOGAXS
100	IF(ND .EQ. NODEC) GO TO 200	LOGAXS

TICX2 = TICXA	LOGAXS
TICY2 = TICYA	LOGAXS
N = 0	LOGAXS
XD = X	LOGAXS
YD = Y	LOGAXS
ND = ND+1	LOGAXS
GO TO 60	LOGAXS
C	LOGAXS
200 RETURN	LOGAXS
END	LOGAXS

	SUBROUTINE MAT31 (A,B,C)		MAT31
C		REV 17	01/03/77MAT31
C	PERFORMS MATRIX MULTIPLICATION C = AB		MAT31
C	WHERE A IS A 3X3 MATRIX, AND B AND C ARE VECTORS OF LENGTH 3.		MAT31
C			MAT31
	IMPLICIT REAL*8 (A-H,O-Z)		MAT31
	DIMENSION A(3,3) , B(3) , C(3)		MAT31
	C(1) = A(1,1)*B(1) + A(1,2)*B(2) + A(1,3)*B(3)		MAT31
	C(2) = A(2,1)*B(1) + A(2,2)*B(2) + A(2,3)*B(3)		MAT31
	C(3) = A(3,1)*B(1) + A(3,2)*B(2) + A(3,3)*B(3)		MAT31
	RETURN		MAT31
	END		MAT31

	SUBROUTINE MAT33 (A,B,C)			MAT33
C		REV 17	01/03/77	MAT33
C	PERFORMS MATRIX MULTIPLICATION C = AB			MAT33
C	WHERE A, B AND C ARE ALL 3X3 MATRICEES.			MAT33
C				MAT33
	IMPLICIT REAL*8 (A-H,O-Z)			MAT33
	DIMENSION A(3,3) , B(3,3) , C(3,3)			MAT33
	DO 10 I=1,3			MAT33
	DO 10 J=1,3			MAT33
10	C(I,J) = A(I,1)*B(1,J) + A(I,2)*B(2,J) + A(I,3)*B(3,J)			MAT33
	RETURN			MAT33
	END			MAT33


```

SUBROUTINE OUTPUT(IJK)
C
C REV IV 02/01/88MISDOT
C CONTROLS TABULATED OUTPUT ON FORTRAN UNITS (STARTING WITH NO. 21) OUTPUT
C OF SELECTED OPTIONAL SEGMENT LINEAR AND ANGULAR ACCELERATIONS, OUTPUT
C VELOCITIES AND DISPLACEMENTS, JOINT PARAMETERS AND SELECTED DATA OUTPUT
C FROM ALL ALLOWED CONTACT FORCE COMPUTATIONS BETWEEN BODY SEGMENTS OUTPUT
C AND VEHICLE COMPONENTS. OUTPUT
C
C IMPLICIT REAL*8 (A-H,O-Z) OUTPUT
COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NCRND, OUTPUT
* NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG PAGE
COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60), OUTPUT
* F(3,30),TQ(3,30),WJ(30),A11(3,3,30) SLIP
COMMON/SCMNTS/ D(3,3,30),WHEG(3,30),WHEGD(3,30),U1(3,30),U2(3,30), OUTPUT
* SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30) OUTPUT
COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60), SLIP
* RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90), OUTPUT
* JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30) OUTPUT
COMMON/JBARTZ/ MNPL( 30),MNBLT( 8),MNSEG( 30),MNBAG( 6), OUTPUT
* MPL(3,5,30),MBLT(3,5,8),MSEG(3,5,30),MBAG(3,10,6), OUTPUT
* NTPL( 5,30),NTBLT( 5,8),NTSEG( 5,30) OUTPUT
COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5), OUTPUT
* BLTTTL(5,8),PLTTL(5,30),BAGTTL(5,6),SEG(30), OUTPUT
* JOINT(30),CGS(30),JS(30) OUTPUT
REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT OUTPUT
LOGICAL*1 CGS,JS OUTPUT
COMMON/FORCES/PSF(7,70),BSF(4,20),SSF(10,40),BAGSF(3,20), NCFORC
* PRJNT(7,30),NPANEL(5),NPSF,NBST,NSSF,NBSGF OUTPUT
COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24), OUTPUT
* UNITL,UNITM,UNITT,GRAVTY(3),TWOPI TWOPI
COMMON/RSAVE/ XSG(3,20,3),DPMI(3,3,30),LPMI(30), ATBIII
* NSG(9),MSG(20,9),MCG,MCGIN(24,5),KREF(20,9) TTHKREF
COMMON/COMAIN/VAR(240),DER(240),DT,HO,HMAX,HMIN,RSTIME, OUTPUT
* ISTEP,NSTEPS,NDINT,NEQ,IRSN,IRSOUT OUTPUT
COMMON/DAMPER/ AFSDM(3,20),APSDN(3,20),ASD(5,20),MSDM(20),MSDN(20) OUTPUT
COMMON/HRNESS/ BAR(15,100),BB(100),BBDOT(100),PLOSS(2,100), OUTPUT
* XLONG(20),HTIME(2),IBAR(5,100),NL(2,100), OUTPUT
* NPTSPB(20),NPTPLY(20),NTHRNS(20),NBLTPH(5) OUTPUT
COMMON/WINDFR/ WTIME(30),QFU(3,5),QFV(3,5),WF(3,30),IWIND(30), WINDOP
* MWSEG(7,30),NFVSEG(6),NFVNT(5),MOWSEG(30,30) WINDOP
COMMON/TEMPVS/ TDATA(14,65),ACC(7,20),T1(3),T2(3),T3(3),T4(9) CHGIII
* T5(3,3),T6(3,3),T7(3),DMY(34024) 80386
LOGICAL LTAPE8 , LTHIST OUTPUT
DATA LINES/-1/,LPP/45/,NTMAX/65/ CHGIII
DATA KMAX/20/,NMAX/22/,MCGMAX/5/ CHGIII
C
C IF (IJK.NE.0) GO TO 13 OUTPUT
C
C SET ALL FORCE ARRAYS TO ZERO. OUTPUT
C
C DO 2 I=1,7 OUTPUT
DO 2 J=1,70 MISDOT
MISDOT

```

2	PSF(I,J) = 0.0	MISDOT
	DO 3 I=1,4	MISDOT
	DO 3 J=1,20	MISDOT
3	BSF(I,J) = 0.0	MISDOT
	DO 4 I=1,10	MISDOT
	DO 4 J=1,40	MISDOT
4	SSF(I,J) = 0.0	MISDOT
	DO 5 I=1,3	MISDOT
	DO 5 J=1,20	MISDOT
5	BAGSF(I,J) = 0.0	MISDOT
	DO 6 I=1,7	MISDOT
	DO 6 J=1,30	MISDOT
6	PRJNT(I,J) = 0.0	MISDOT
	GO TO 66	OUTPUT
C		OUTPUT
C	LTHIST = TRUE MEANS PRINT LINE OF TIME HISTORY DATA FOR THIS	OUTPUT
C	TIME POINT ON EACH OUTPUT UNIT (NT).	OUTPUT
C		OUTPUT
C	LTAPE8 = TRUE MEANS WRITE TIME HISTORY DATA ON TAPE 8.	OUTPUT
C		OUTPUT
13	NPRT4 = NPRT(4) + 4	OUTPUT
	IF (NPRT4.LE.0 .OR. NPRT4.GT.8) STOP 37	OUTPUT
	IF(NPRT(26).EQ.6) GO TO 66	TGMOD1
	GO TO (66,66,66,15,16,17,17,16) , NPRT4	OUTPUT
15	LTAPE8 = .FALSE.	OUTSTP
	LTHIST = .TRUE.	TGMOD1
	GO TO 116	TGMOD1
16	LTHIST = .TRUE.	TGMOD1
	LTAPE8 = .TRUE.	TGMOD1
	GO TO 116	TGMOD1
17	LTHIST = .FALSE.	TGMOD1
	LTAPE8 = .TRUE.	TGMOD1
	GO TO 217	TGMOD1
116	TEST = DMOD(TIME,DT)	OUTSTP
	TEST = DMIN1(TEST,DABS(DT-TEST))	OUTSTP
	IF ((NPRT(26).EQ.0.OR.NPRT(26).EQ.3).AND.TEST.GE.EPS(8))	TGMOD1
	* LTHIST=.FALSE.	TGMOD1
	IF(.NOT.LTAPE8.AND..NOT.LTHIST) GO TO 66	FIXTTH
217	CONTINUE	TGMOD1
	IF(NPRT(26).EQ.4) LTHIST=.FALSE.	TGMOD1
	IF(NPRT(26).EQ.5) LTAPE8=.FALSE.	TGMOD1
	IF(.NOT.LTAPE8.AND..NOT.LTHIST) GO TO 66	TGMOD1
	CALL ELTIME (1,8)	OUTPUT
	IF (LINES.GE.0) GO TO 21	FIXTTH
	PREVT = -999.0	OUTPUT
	LINES = 0	FIXTTH
	IF (IRSIN.NE.0) GO TO 10	OUTPUT
C		OUTPUT
C	1ST TIME IN ROUTINE, READ CARD INPUT FOR OUTPUT CONTROL.	OUTPUT
C		OUTPUT
C	1. NO. OF POINT TOTAL ACCELERATIONS ,POINT NOS. AND LOCATION	CHGIII
C	2. NO. OF POINT REL. VELOCITIES .POINT NOS. AND LOCATION	CHGIII

C	3. NO. OF POINT REL. LINEAR DISPLACEMENTS ,POINT NOS. AND LOCATIC	CHGIII
C	4. NO. OF SEGMENT ANGULAR ACCELERATIONS AND SEGMENT NOS.	CHGIII
C	5. NO. OF SEGMENT REL. ANGULAR VELOCITIES AND SEGMENT NOS.	CHGIII
C	6. NO. OF SEGMENT REL. ANGULAR DISPLACEMENTS AND SEGMENT NOS.	CHGIII
C	7. NO. OF JOINT PARAMETERS AND JOINT NOS.	OUTPUT
C	8. NO. OF SEGMENT WIND FORCES AND SEGMENT NOS.	WINDOP
C	9. NO. OF JOINT FORCES AND TORQUE NOS.	WINDOP
C	10. NO. OF CENTER OF GRAVITY AND RELATED INFORMATION	WINDOP
C		OUTPUT
	WRITE(6,478)	CHGIII
478	FORMAT(1X,/,2X,'TABULAR TIME HISTORY CONTROL PARAMETERS')	CHGIII
	WRITE(6,479)	CHGIII
479	FORMAT(3X,'TYPE KSG SELECTED SEGMENTS OR JOINTS')	TTHKREF
	DO 20 K=1,9	WINDOP
C		OUTPUT
C	INPUT CARDS H.(K).(J) FOR K=1,3	OUTPUT
C		OUTPUT
	IF (K.LE.3) READ (5,18) KSG,KREF(1,K),MSG(1,K),(XSG(I,1,K),I=1,3)	TTHKREF
18	FORMAT (16,2I3,3F12.6)	TTHKREF
	IF (KSG.GT.KMAX) STOP 84	CHGIII
	IF (K.GT.3) GO TO 201	ATBIII
	IF (KSG.LE.1) READ(5,213) IDUMMY	ATBIII
213	FORMAT(I2)	ATBIII
	IF (KSG.LE.1) GO TO 201	ATBIII
	DO 205 J=2,KSG	ATBIII
	READ (5,210) KREF(J,K),MSG(J,K),(XSG(I,J,K),I=1,3)	TTHKREF
210	FORMAT (19,I3,3F12.6)	TTHKREF
205	CONTINUE	ATBIII
201	CONTINUE	ATBIII
C		OUTPUT
C	INPUT CARDS H.(K) FOR K=4,9	WINDOP
C		OUTPUT
	IF (K.GT.3) READ (5,19) KSG,(KREF(J,K),MSG(J,K),J=1,KSG)	TTHKREF
19	FORMAT(16,22I3/(19,21I3))	TTHKREF
	IF (KSG.GT.KMAX) STOP 85	CHGIII
	WRITE (6,78) K,KSG,(MSG(J,K),J=1,KSG)	TTHKREF
	WRITE (6,81) (KREF(J,K),J=1,KSG)	TTHKREF
78	FORMAT(' H.',1I,1X,I3,3X,20I3)	TTHKREF
81	FORMAT(' REF ',20I3)	TTHKREF
	DO 80 J=1,KSG	TTHKREF
	IF(KREF(J,K).GT.NGRND.OR.KREF(J,K).LT.0) STOP 55	TTHKREF
80	CONTINUE	TTHKREF
	IF (K.NE.7 .OR. KSG.EQ.0) GO TO 20	OUTPUT
	DO 12 J=1,KSG	OUTPUT
	L = MSG(J,K)	OUTPUT
	IF (IABS(IPIN(L)).EQ.4) MSG(J,K) = -L	OUTPUT
12	CONTINUE	OUTPUT
20	NSG(K) = KSG	OUTPUT
C		ATBIII
C	READ INPUT CARDS H.10	WINDOP
C		ATBIII
	READ (5,111) MCG	ATBIII

111	FORMAT(I6)	ATBIII
	IF (MCG.GT.MCGMAX) STOP 86	CHGIII
	IF (MCG.EQ.0) GO TO 114	ATBIII
	DO 113 K=1,MCG	ATBIII
	READ (5,112) M,N,(MCGIN(I+2,K),I=1,N)	ATBIII
112	FORMAT (24I3)	ATBIII
	IF (N.GT.NMAX) STOP 87	CHGIII
	WRITE (6,117) N,(MCGIN(I+2,K),I=1,N)	TTHKREF
117	FORMAT(' H.10',I3,3X,22I3)	TTHKREF
	WRITE (6,81) M	TTHKREF
	MCGIN(1,K) = M	ATBIII
113	MCGIN(2,K) = N	ATBIII
114	CONTINUE	ATBIII
10	IF (.NOT.LTAPE8) GO TO 21	OUTPUT
	WRITE (8) NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,NPANEL,	OUTPUT
	* MNPL,MNBLT,MNSEG,MNBAG,MPL,MBLT,MSEG,MBAG	OUTPUT
	WRITE (8) DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,	OUTPUT
	* SEG,JOINT,UNITL,UNITM,UNITT,NSG,MSG,XSG,MCG,	ATBIII
	* MCGIN,KREF,NHRNSS,NBLTPH,NPTSPB,NSD,MSDM,MSDN	CHGIII
21	IF(LTHIST) LINES= LINES + 1	FIXTTH
	IF (MOD(LINES,LPP).EQ.1 .AND. LTHIST) CALL HEDING (LINES,LPP)	OUTPUT
	NT = 20	OUTPUT
	USEC = 1000.0*TIME	OUTPUT
C		OUTPUT
C	COMPUTE AND PRINT DATA FOR 9 TYPES OF OUTPUT ABOVE	WINDOP
C		OUTPUT
	DO 44 K=1,9	WINDOP
	IF (NSG(K).LE.0) GO TO 44	OUTPUT
	KSG = NSG(K)	OUTPUT
	IF (K.GT.8) GO TO 440	WINDOP
	J3 = 3	OUTPUT
	IF (K.EQ.7) J3 = 2	OUTPUT
	DO 43 J1=1,KSG,J3	OUTPUT
	J2 = MINO(J1+J3-1,KSG)	OUTPUT
	NT =NT + 1	OUTPUT
C	SETUP LOGICAL UNIT CONTROL (FOR PRINTER) FOR PERKIN & ELMER	PECONV
C	CALL CARCON(NT,1)	80386
	DO 38 J=J1,J2	OUTPUT
	L = IABS(MSG(J,K))	OUTPUT
	GO TO (2,24,26,29,31,34,35,601),K	WINDOP
C		OUTPUT
C	1. POIN" TOTAL ACCELERATION IN KREF(1) REFERENCE	CHGIII
C		OUTPUT
22	IF(LPMI(L).EQ.0) GO TO 521	CHGIII
	CALL MAT31(LPMI(1,1,L),XSG(1,J,K),T7)	CHGIII
	GO TO 523	CHGIII
521	DO 522 JL=1,3	CHGIII
522	T7(JL) = XSG(JL,J,K)	CHGIII
523	CALL CROSS (WMEC(1,L),T7,T1)	CHGIII
	CALL CROSS (WMEG(1,L),T1,T2)	OUTPUT
	CALL CROSS (WMEGD(1,L),T7,T3)	CHGIII
	CALL MAT31(D(1,1,L),GRAVITY,T7)	ACCEL

CALL MAT31(D(1,1,L),SEGLA(1,L),T4)	OUTPUT
DO 23 I=1,3	OUTPUT
IF(MSG(J,K).LT.0) T4(I)=T4(I)+T7(I)	ACCEL
ACC(I,J) = (T4(I)+T3(I)+T2(I))/G	OUTPUT
23 T1(I) = ACC(I,J)	OUTPUT
IF(MSG(J,K).GE.0) GO TO 405	ACCEL
KRF=L	ACCEL
IF(LPMI(KRF).NE.0) CALL DOT31(DPMI(1,1,KRF),T1,ACC(1,J))	ACCEL
IF(KREF(J,K).EQ.1) GOTO 33	ACCEL
DO 600 II=1,3	ACCEL
600 ACC(II,J)=ACC(II,J)-GRAVITY(II)/G	ACCEL
GOTO 33	ACCEL
C	OUTPUT
C	CHGIII
C	OUTPUT
24 IF(KREF(J,2).EQ.0) KRF = NVEH	TTHKREF
IF(KREF(J,2).NE.0) KRF = KREF(J,2)	TTHKREF
IF(LPMI(L).EQ.0) GO TO 524	CHGIII
CALL MAT31(DPMI(1,1,L),XSG(1,J,K),T7)	CHGIII
GO TO 525	CHGIII
524 DO 526 JL=1,3	CHGIII
526 T7(JL) = XSG(JL,J,K)	CHGIII
525 CALL CROSS (WMEG(1,L),T7,T1)	CHGIII
CALL DOT31(D(1,1,L),T1,T2)	OUTPUT
DO 25 I=1,3	OUTPUT
25 T3(I) = T2(I) + SEGLV(I,L) - SEGLV(I,KRF)	CHGIII
GO TO 28	OUTPUT
C	OUTPUT
C	CHGIII
C	OUTPUT
26 IF(KREF(J,3).EQ.0) KRF = NVEH	TTHKREF
IF(KREF(J,3).NE.0) KRF = KREF(J,3)	TTHKREF
IF (LPMI(L).EQ.0) GO TO 76	CHGIII
CALL DOT33 (DPMI(1,1,L),D(1,1,L),T4)	OUTPUT
CALL DOT31 (T4,XSG(1,J,K),T1)	OUTPUT
GO TO 77	OUTPUT
76 CALL DOT31 (D(1,1,L),XSG(1,J,K),T1)	OUTPUT
77 DO 27 I=1,3	OUTPUT
27 T3(I) = T1(I) + SEGLP(I,L) - SEGLP(I,KRF)	CHGIII
28 IF (LPMI(KRF).EQ.0) GO TO 403	CHGIII
CALL DOT33(DPMI(1,1,KRF),D(1,1,KRF),T5)	CHGIII
CALL MAT31(T5,T3,ACC(1,J))	CHGIII
GO TO 33	OUTPUT
403 CALL MAT31(D(1,1,KRF),T3,ACC(1,J))	CHGIII
33 ACC(4,J) = DSQRT(ACC(1,J)**2+ACC(2,J)**2+ACC(3,J)**2)	CHGIII
GO TO 38	CHGIII
C	OUTPUT
C	CHGIII
C	OUTPUT
29 DO 30 I=1,3	OUTPUT
ACC(I,J) = WMEGD(I,L)/(2.0*PI)	OUTPUT
30 T1(I) = ACC(I,J)	OUTPUT

405	CONTINUE	CHGIII
	IF(KREF(J,K).EQ.0) GO TO 401	TTHKREF
	KRF = KREF(J,K)	TTHKREF
	IF(LPMI(KRF).EQ.0) GO TO 402	CHGIII
	CALL DOT33(DPMI(1,1,KRF),D(1,1,KRF),T5)	CHGIII
	CALL DOTT33(T5,D(1,1,L),T6)	CHGIII
	CALL MAT31(T6,T1,ACC(1,J))	CHGIII
	GO TO 33	CHGIII
402	CALL DOTT33(D(1,1,KRF),D(1,1,L),T6)	CHGIII
	CALL MAT31(T6,T1,ACC(1,J))	CHGIII
	GO TO 33	CHGIII
401	KRF = L	CHGIII
	IF(LPMI(KRF).NE.0) CALL DOT31(DPMI(1,1,KRF),T1,ACC(1,J))	CHGIII
	GO TO 33	OUTPUT
C		OUTPUT
C	5. SEGMENT REL. ANGULAR VELOCITY IN KREF(5) REFERENCE	CHGIII
C		OUTPUT
31	IF(KREF(J,5).EQ.0) KRF = NVEH	TTHKREF
	IF(KREF(J,5).NE.0) KRF = KREF(J,5)	TTHKREF
	CALL DOT31 (D(1,1,L),WMEG(1,L),T1)	CHGIII
	CALL MAT31 (D(1,1,KRF),T1,T2)	CHGIII
	DO 32 I=1,3	OUTPUT
	IF (KRF.NE.L) T2(I)=T2(I)-WMEG(I,KRF)	PLTINC
32	T3(I) = T2(I)/(2.0*PI)	PLTINC
	IF(LPMI(KRF).EQ.0) GO TO 449	CHGIII
	CALL DOT31(DPMI(1,1,KRF),T3,ACC(1,J))	CHGIII
	GO TO 483	CHGIII
449	CONTINUE	CHGIII
	DO 457 KJL=1,3	CHGIII
457	ACC(KJL,J) = T3(KJL)	CHGIII
483	ACC(4,J) = DSQRT(ACC(1,J)**2+ACC(2,J)**2+ACC(3,J)**2)	CHGIII
	GO TO 38	OUTPUT
C		OUTPUT
C	6. SEGMENT REL. ANGULAR DISPLACEMENT IN KREF(6) REFERENCE	CHGIII
C		OUTPUT
34	IF(KREF(J,6).EQ.0) KRF = NVEH	TTHKREF
	IF(KREF(J,6).NE.0) KRF = KREF(J,6)	TTHKREF
	IF (LPMI(KRF).EQ.0.AND.LPMI(L).EQ.0) GO TO 36	CHGIII
	IF (LPMI(L).EQ.0) GO TO 435	CHGIII
	CALL DOT33(DPMI(1,1,L),D(1,1,L),T4)	CHGIII
435	IF (LPMI(KRF).EQ.0) GO TO 436	CHGIII
	CALL DOT33(DPMI(1,1,KRF),D(1,1,KRF),T5)	CHGIII
436	IF (LPMI(L).NE.0) GO TO 438	CHGIII
	CALL DOTT33(D(1,1,L),T5,T1)	CHGIII
	GO TO 37	CHGIII
438	IF (LPMI(KRF).NE.0) GO TO 439	CHGIII
	CALL DOTT33(T4,D(1,1,KRF),T1)	CHGIII
	GO TO 37	CHGIII
439	CALL DOTT33(T4,T5,T1)	CHGIII
	GO TO 37	CHGIII
36	CALL DOTT33(D(1,1,L),D(1,1,KRF),T1)	CHGIII
37	CALL YPRDEG(T1,ACC(1,J))	OUTPUT

	TRACE = 0.5*(T1(1)+T2(2)+T3(3)-1.0)	OUTPUT
	IF (TRACE.GT. 1.0) TRACE = 1.0	OUTPUT
	IF (TRACE.LT.-1.0) TRACE = -1.0	OUTPUT
	ACC(4,J) = DACOS(TRACE)/RADIAN	OUTPUT
	GO TO 38	OUTPUT
C		OUTPUT
C	7. JOINT PARAMETERS	OUTPUT
C		OUTPUT
	35 ACC(1,J) = PRJNT(1,L)	OUTPUT
	ACC(2,J) = PRJNT(2,L)/RADIAN	OUTPUT
	ACC(3,J) = PRJNT(3,L)/RADIAN	OUTPUT
	ACC(4,J) = PRJNT(4,L)/RADIAN	OUTPUT
	ACC(5,J) = DSQRT(PRJNT(5,L))	OUTPUT
	ACC(6,J) = DSQRT(PRJNT(6,L))	OUTPUT
	ACC(7,J) = DSQRT(PRJNT(7,L))	OUTPUT
	GOTO 38	WINDOP
C		WINDOP
C	8. SEGMENT WIND FORCE IN KREF(8) REFERENCE	WINDOP
C		WINDOP
	601 IF(KREF(J,8).EQ.0) KRF = NGRND	TTHKREF
	IF(KREF(J,8).NE.0) KRF = KREF(J,8)	TTHKREF
	CALL MAT31 (D(1,1,KRF),WF(1,L),T2)	WINDOP
	IF(LPMI(KRF).EQ.0) GO TO 602	WINDOP
	CALL DOT31(DPMI(1,1,KRF),T2,ACC(1,J))	WINDOP
	GO TO 604	WINDOP
	602 CONTINUE	WINDOP
	DO 603 KJL=1,3	WINDOP
	603 ACC(KJL,J) = T2(KJL)	WINDOP
	604 ACC(4,J) = DSQRT(ACC(1,J)**2+ACC(2,J)**2+ACC(3,J)**2)	WINDOP
	38 CONTINUE	OUTPUT
	IF (.NOT.LTAPE8) GO TO 40	OUTPUT
	KK = 0	OUTPUT
	I2 = 4	OUTPUT
	IF (K.EQ.7) I2 = 7	OUTPUT
	DO 39 J=J1,J2	OUTPUT
	DO 39 I=1,I2	OUTPUT
	KK = KK+1	OUTPUT
	39 TDATA(KK,NT-20) = ACC(I,J)	OUTPUT
	40 IF (.NOT.LTHIST) GO TO 43	OUTPUT
	IF (K.LE.6) WRITE (NT,41) USEC,((ACC(I,J),I=1,4),J=J1,J2)	OUTPUT
	IF (K.EQ.8) WRITE (NT,41) USEC,((ACC(I,J),I=1,4),J=J1,J2)	WINDOP
	41 FORMAT(F9.3,3(3X,4F9.3))	OUTPUT
	IF (K.EQ.7) WRITE (NT,42) USEC,((ACC(I,J),I=1,7),J=J1,J2)	OUTPUT
	42 FORMAT(F9.3,2(F5.0,3F9.3,2X,3F9.3))	OUTPUT
	43 CONTINUE	OUTPUT
	GO TO 44	CHGIII
C		ATBIII
C	9. JOINT FORCES & TORQUES IN KREF(9) GEOMETRIC COORDINATE SYSTEM	WINDOP
C		CHGIII
	440 DO 860 L=1,KSG	PLTINC
	KRF = NVEH	PLTINC
	IF(KREF(L,9).NE.0) KRF = KREF(L,9)	PLTINC

LL=MSG(L,K)	CHGIII
IF (LPMI(KRF).EQ.0) GO TO 851	CHGIII
CALL DOT33 (DPMI(1,1,KRF),D(1,1,KRF),T5)	CHGIII
CALL MAT31 (T5,F(1,LL),T1)	CHGIII
CALL MAT31 (T5,TQ(1,LL),T2)	CHGIII
DO 852 JJ=1,3	CHGIII
T1(JJ) = T1(JJ)/100.0	CHGIII
852 T2(JJ) = -T2(JJ)/100.0	OUT385
GO TO 859	CHGIII
851 CONTINUE	CHGIII
CALL MAT31 (D(1,1,KRF),F(1,LL),T1)	CHGIII
CALL MAT31 (D(1,1,KRF),TQ(1,LL),T2)	CHGIII
DO 853 JJ=1,3	CHGIII
T1(JJ) = T1(JJ)/100.0	CHGIII
853 T2(JJ) = -T2(JJ)/100.0	OUT385
859 NT = NT + 1	CHGIII
C P & E CARRIAGE CONTROL	PECONV
C CALL CARCON(NT,1)	80386
IF (.NOT.LTAPE8) GO TO 855	CHGIII
DO 854 JL=1,3	CHGIII
TDATA (JL ,NT-20) = T1(JL)	CHGIII
854 TDATA (JL+3,NT-20) = T2(JL)	CHGIII
855 CONTINUE	CHGIII
IF (LTHIST) WRITE (NT,857) USEC,T1,T2	CHGIII
857 FORMAT(F9.3,3X,3F9.3,3X,3(2X,D10.3))	CHGIII
860 CONTINUE	CHGIII
44 CONTINUE	CHGIII
C	ATBIII
C 10. PRINT BODY PROPERTIES	WINDOP
C	ATBIII
IF (MCG.EQ.0) GO TO 131	ATBIII
DO 130 NCG=1,MCG	ATBIII
M = MCGIN(1,NCG)	ATBIII
N = MCGIN(2,NCG)	ATBIII
DO 120 J=1,9	ATBIII
120 T4(J) = 0.0	ATBIII
SUMW = 0.0	ATBIII
T7(1)=0.0	KINETIC
T7(2)=0.0	KINETIC
DO 123 I=1,N	ATBIII
K = MCGIN(I+2,NCG)	ATBIII
WG = W(K)/G	ATBIII
V=(SEGLV(1,K)-SEGLV(1,M))**2	KINETIC
* +(SEGLV(2,K)-SEGLV(2,M))**2	KINETIC
* +(SEGLV(3,K)-SEGLV(3,M))**2	KINETIC
T7(1)=T7(1)+0.5*WG*V	KINETIC
SUMW = SUMW + WG	ATBIII
DO 121 J=1,3	ATBIII
T7(2)=T7(2)+0.5*PHI(J,K)*(WMEG(J,K)-WMEG(J,M))**2	KINETIC
121 T1(J) = PHI(J,K)*WMEG(J,K)	ATBIII
CALL DOT31 (D(1,1,K),T1,T2)	ATBIII
CALL CROSS (SEGLP(1,K),SEGLV(1,K),T1)	ATBIII

DO 122 J=1,3	ATBIII
T4(J) = T4(J) + WG*SEGLP(J,K)	ATBIII
T4(J+3) = T4(J+3) + WG*SEGLV(J,K)	ATBIII
122 T4(J+6) = T4(J+6) + WG*T1(J) + T2(J)	ATBIII
123 CONTINUE	ATBIII
T7(3)=T7(1)+T7(2)	KINETIC
DO 124 J=1,3	ATBIII
124 T4(J) = T4(J)/SUMW - SEGLP(J,M)	ATBIII
C	ATBIII
C TRANSFORM FROM PRINCIPAL AXES TO LOCAL AXES	TGMOD1
C	ATBIII
IF (LPM1(M).EQ.0) GO TO 330	ATBIII
CALL DOT33(DPM1(1,1,M),D(1,1,M),T5)	ATBIII
CALL MAT31(T5,T4(1),T1)	ATBIII
CALL MAT31(T5,T4(4),T2)	ATBIII
CALL MAT31(T5,T4(7),T3)	ATBIII
GO TO 333	ATBIII
330 CONTINUE	ATBIII
CALL MAT31 (D(1,1,M),T4(1),T1)	ATBIII
CALL MAT31 (D(1,1,M),T4(4),T2)	ATBIII
CALL MAT31 (D(1,1,M),T4(7),T3)	ATBIII
333 CONTINUE	ATBIII
NT = NT + 1	ATBIII
IF (.NOT.LTAPE8) GO TO 126	ATBIII
DO 125 J=1,3	ATBIII
TDATA (J ,NT-20) = T1(J)	ATBIII
TDATA (J+3,NT-20) = T2(J)	ATBIII
TDATA(J+9,NT-20) = T7(J)	KINETIC
125 TDATA(J+6,NT-20) = T3(J)	ATBIII
126 IF (LTHIST) WRITE (NT,127) USEC,T1,T2,T3,T7	KINETIC
127 FORMAT (F9.3,3F8.3,9(1X,D10.3))	KINETIC
130 CONTINUE	ATBIII
131 CONTINUE	ATBIII
C	OUTPUT
C PRINT PLANE FORCES	OUTPUT
C	OUTPUT
MPSF = 0	OUTPUT
IF (NPL.EQ.0) GO TO 49	OUTPUT
IF (NPRT(18).EQ.1.OR.NPRT(18).EQ.7) GO TO 49	VARTTH
IF (NPRT(18).EQ.10.OR.NPRT(18).EQ.11) GO TO 49	VARTTH
IF (NPRT(18).GE.14) GO TO 49	VARTTH
DO 45 J=1,NPL	OUTPUT
45 MPSF = MPSF + MNPL(J)	OUTPUT
IF (MPSF.EQ.0) GO TO 49	OUTPUT
DO 47 J1=1,MPSF,2	OUTPUT
J2 = MINO(J1+1,MPSF)	OUTPUT
NT = NT+1	OUTPUT
C SETUP LOGICAL UNIT CONTROL (PRINTER CONTROL) FOR P & E	PECONV
C CALL CARCON(NT,1)	80386
IF (.NOT.LTAPE8) GO TO 47	OUTPUT
KK = 0	OUTPUT
DO 46 J=J1,J2	OUTPUT

DO 46 I=1,7	OUTPUT
KK = KK+1	OUTPUT
46 TDATA(KK,NT-20) = PSF(I,J)	OUTPUT
47 IF (LTHIST) WRITE (NT,48) USEC,((PSF(I,J),I=1,7),J=J1,J2)	OUTPUT
48 FORMAT(F9.3,2(F9.3,3F9.2,3F8.3))	OUTPUT
C	OUTPUT
C PRINT BELT FORCES	OUTPUT
C	OUTPUT
49 MBSF = 0	OUTPUT
IF (NBLT.EQ.0) GO TO 67	OUTPUT
IF (NPRT(18).EQ.2.OR.NPRT(18).GE.13) GO TO 67	VARTTH
IF (NPRT(18).GE.7.AND.NPRT(18).LE.9) GO TO 67	VARTTH
DO 50 J=1,NBLT	OUTPUT
50 MBSF = MBSF + MNBLT(J)	OUTPUT
IF (MBSF.EQ.0) GO TO 67	OUTPUT
DO 52 J1=1,MBSF,2	OUTPUT
J2 = MINO(J1+1,MBSF)	OUTPUT
NT = NT+1	OUTPUT
C LOGICAL UNIT (PRINTER CONTROL) FOR P & E	PECONV
C CALL CARCON(NT,1)	80386
IF (.NOT.LTAPE8) GO TO 52	OUTPUT
KK = 0	OUTPUT
DO 51 J=J1,J2	OUTPUT
DO 51 I=1,4	OUTPUT
KK = KK+1	OUTPUT
51 TDATA(KK,NT-20) = BSF(I,J)	OUTPUT
52 IF (LTHIST) WRITE (NT,53) USEC,((BSF(I,J),I=1,4),J=J1,J2)	OUTPUT
53 FORMAT(F9.3,4(F15.6,F12.2,3X))	OUTPUT
C	OUTPUT
C PRINT HARNESS-BELT ENDPOINT FORCES (STORED IN BSF ARRAY).	OUTPUT
C	OUTPUT
67 IF (NHRNSS.LE.0) GO TO 71	OUTPUT
IF (NPRT(18).EQ.3.OR.NPRT(18).EQ.11) GO TO 71	VARTTH
IF (NPRT(18).EQ.9.OR.NPRT(18).EQ.8) GO TO 71	VARTTH
IF (NPRT(18).EQ.13.OR.NPRT(18).EQ.14) GO TO 71	VARTTH
IF (NPRT(18).GE.16) GO TO 71	VARTTH
MBSF1 = MBSF + 1	OUTPUT
DO 68 I=1,NHRNSS	OUTPUT
68 MBSF = MBSF + NBLTPH(I)	OUTPUT
DO 70 J1=MBSF1,MBSF,2	OUTPUT
J2 = MINO(J1+1,MBSF)	OUTPUT
NT = NT+1	OUTPUT
C LOGICAL UNIT (PRINTER CONTROL) FOR P & E	PECONV
C CALL CARCON(NT,1)	80386
IF (.NOT.LTAPE8) GO TO 70	OUTPUT
KK = 0	OUTPUT
DO 69 J=J1,J2	OUTPUT
DO 69 I=1,4	OUTPUT
KK = KK+1	OUTPUT
69 TDATA(KK,NT-20) = BSF(I,J)	OUTPUT
70 IF (LTHIST) WRITE (NT,53) USEC,((BSF(I,J),I=1,4),J=J1,J2)	OUTPUT
C	OUTPUT

C	PRINT SPRING DAMPER FORCES (STORED IN BSF ARRAY).	OUTPUT
C		OUTPUT
	71 IF (NSD.LE.0) GO TO 54	OUTPUT
	IF (NPRT(18).EQ.4.OR.NPRT(18).EQ.9) GO TO 54	VARTTH
	IF (NPRT(18).GE.12) GO TO 54	VARTTH
	MBSF1 = MBSF + 1	OUTPUT
	MBSF = MBSF + (NSD:)/2	OUTPUT
	DO 73 J1=MBSF1,MBSF,2	OUTPUT
	J2 = MINO(J1+1,MBSF)	OUTPUT
	NT = NT+1	OUTPUT
C	LOGICAL UNIT (PRINTER CONTROL) FOR P & E	PECONV
C	CALL CARCON(NT,1)	80386
	IF (.NOT.LTAPE8) GO TO 73	OUTPUT
	KK = 0	OUTPUT
	DO 72 J=J1,J2	OUTPUT
	DO 72 I=1,4	OUTPUT
	KK = KK+1	OUTPUT
	72 TDATA(KK,NT-20) = BSF(I,J)	OUTPUT
	73 IF (LTHIST) WRITE (NT,74) USEC,((BSF(I,J),I=1,4),J=J1,J2)	OUTPUT
	74 FORMAT (F9.3,4(F14.3,F12.2,4X))	OUTPUT
C		OUTPUT
C	PRINT SEGMENT CONTACT FORCES	OUTPUT
C		OUTPUT
	54 MSSF = 0	OUTPUT
	IF (NPRT(18).EQ.5.OR.NPRT(18).EQ.13) GO TO 161	VARTTH
	IF (NPRT(18).EQ.10.OR.NPRT(18).EQ.11) GO TO 161	VARTTH
	IF (NPRT(18).GE.15) GO TO 161	VARTTH
	DO 55 J=1,NSEG	OUTPUT
	55 MSSF = MSSF + MNSEG(J)	OUTPUT
	IF (MSSF.EQ.0) GO TO 59	OUTPUT
	DO 57 J=1,MSSF	OUTPUT
	NT = NT+1	OUTPUT
C	LOGICAL UNIT (PRINTER CONTROL) FOR P & E	PECONV
C	CALL CARCON(NT,1)	80386
	IF (.NOT.LTAPE8) GO TO 57	OUTPUT
	DO 56 I=1,10	OUTPUT
	56 TDATA(I,NT-20) = SSF(I,J)	OUTPUT
	57 IF (LTHIST) WRITE (NT,58) USEC,(SSF(I,J),I=1,10)	OUTPUT
	58 FORMAT(2F9.3,3F9.2,3F8.3,2X,3F8.3)	OUTPUT
	161 CONTINUE	VARTTH
C		OUTPUT
C	PRINT AIRBAG FORCES	OUTPUT
C		OUTPUT
	59 IF (NBAG.EQ.0) GO TO 65	OUTPUT
	IF (NPRT(18).EQ.6.OR.NPRT(18).EQ.9) GO TO 65	VARTTH
	IF (NPRT(18).GE.12) GO TO 65	VARTTH
	K1 = 1	OUTPUT
	DO 64 J=1,NBAG	OUTPUT
	IF (MNBAG(J).EQ.0) GO TO 64	OUTPUT
	KBAG = MNBAG(J)+NPANEL(J)+5	OUTPUT
	DO 63 J1=1,KBAG,4	OUTPUT
	J2 = MINO(J1+3,KBAG)	OUTPUT

	K2 = K1+J2-J1	OUTPUT
	NT = NT+1	OUTPUT
C	LOGICAL UNIT (PRINTER CONTROL) FOR P & E	PECONV
C	CALL CARCON(NT,1)	30386
	IF (.NOT.LTAPE8) GO TO 61	OUTPUT
	KK = 0	OUTPUT
	DO 60 K=K1,K2	OUTPUT
	DO 60 I=1,3	OUTPUT
	KK = KK+1	OUTPUT
60	TDATA(KK,NT-20) = BAGSF(I,K)	OUTPUT
61	IF (.NOT.LTHIST) GO TO 63	OUTPUT
	IF (J1.EQ.1) WRITE (NT,75) USEC,((BAGSF(I,K),I=1,3),K=K1,K2)	OUTPUT
	IF (J1.NE.1) WRITE (NT,62) USEC,((BAGSF(I,K),I=1,3),K=K1,K2)	OUTPUT
75	FORMAT (F9.3,3X,3F9.2,2(3X,3F9.3),3X,3F9.2)	OUTPUT
62	FORMAT(F9.3,4(3X,3F9.2))	OUTPUT
63	K1 = K2+1	OUTPUT
64	CONTINUE	OUTPUT
65	NT = NT-20	OUTPUT
	IF(NT.GT.NTMAX) STOP 56	CHGIII
	IF (LTAPE8) WRITE (8) NT,USEC,((TDATA(I,J),I=1,14),J=1,NT)	OUTPUT
	PREVT = TIME	OUTPUT
	CALL ELTIME(2,8)	OUTPUT
66	RETURN	OUTPUT
	END	OUTPUT

	SUBROUTINE PANEL (DRR,ZR,JB)	PANEL
		REV III.2 08/08/84REVIII
C	COMPUTES AIRBAG PARAMETERS DURING INFLATION OF BAG.	PANEL
C		PANEL
C	GIVEN: DRR - DC MATRIX RELATIVE TO VEHICLE	PANEL
C	ZR - CG LOCATION IN VEHICLE REFERENCE	PANEL
C		PANEL
C	COMPUTE: SEGLP,SEGLV,SEGLA,D,WMEG & WMEGD FOR SEGMENT JB.	PANEL
C		PANEL
	IMPLICIT REAL*8 (A-H,O-Z)	PANEL
	DIMENSION DRR(3,3),ZR(3),T1(3),T2(3)	PANEL
	COMMON/CONTROL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,	PANEL
	* NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG	PAGE
	COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),	PANEL
	* SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)	PANEL
	CALL MAT33 (DRR,D(1,1,NVEH),D(1,1,JB))	PANEL
	CALL MAT31 (DRR,WMEG(1,NVEH),WMEG(1,JB))	PANEL
	CALL DOT31 (D(1,1,NVEH),ZR,SEGLP(1,JB))	PANEL
	CALL CROSS (WMEG(1,NVEH),ZR,T1)	PANEL
	CALL DOT31 (D(1,1,NVEH),T1,SEGLV(1,JB))	PANEL
	CALL CROSS (WMEG(1,NVEH),T1,T2)	PANEL
	CALL DOT31 (D(1,1,NVEH),T2,SEGLA(1,JB))	PANEL
	DO 10 I=1,3	PANEL
	SEGLP(I,JB) = SEGLP(I,JB) + SEGLP(I,NVEH)	PANEL
	SEGLV(I,JB) = SEGLV(I,JB) + SEGLV(I,NVEH)	PANEL
	SEGLA(I,JB) = SEGLA(I,JB) + SEGLA(I,NVEH)	PANEL
10	WMEGD(I,JB) = WMEGD(I,NVEH)	PANEL
	RETURN	PANEL
	END	PANEL

```

SUBROUTINE PDAUX (VAR,DER,NEQ,KDINT)                                PDAUX
C                                                                    REV IV    07/24/86SLIP
C    PURPOSE IS TO ACT AS INTERFACE BETWEEN INTEGRATOR AND DAUX TO  PDAUX
C    ACCOMODATE VARIABLE NUMBER OF FUNCTIONS TO BE INTEGRATED.    PDAUX
C                                                                    PDAUX
C    ARGUMENTS:                                                    PDAUX
C    VAR - ARRAY OF NEQ STATE VARIABLES UPDATED BY DINT.          PDAUX
C    DER - ARRAY OF NEQ DERIVATIVES TO BE SUPPLIED BY DAUX.      PDAUX
C    NEQ - NUMBER OF STATE VARIABLES AND DERIVATIVES.           PDAUX
C    KDINT - INTEGRATION STEP NUMBER IN DINT.                    PDAUX
C                                                                    PDAUX
    IMPLICIT REAL*8 (A-H,O-Z)                                     PDAUX
    DIMENSION VAR(3,1),DER(3,1)                                  PDAUX
    COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,      PDAUX
*    NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG           PAGE
    COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30), PDAUX
*    SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)             PDAUX
    COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60), SLIP
*    RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),          PDAUX
*    JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)         PDAUX
    COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),    PDAUX
*    BLTTTL(5,8),PLTTL(5,30),BAGTTL(5,6),SEG(30),            PDAUX
*    JCINT(30),CGS(30),JS(30)                                  PDAUX
    REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,FLTTL,BAGTTL,SEG,JOINT PDAUX
    LOGICAL*1 CGS,JS                                           PDAUX
    COMMON/INTEST/ SGTST(3,4,30),XTEST(3,120),SEGT(120),REGT(120) PDAUX
    REAL SEGT                                                  PDAUX
    COMMON/FLXBLE/ HF(4,12,8),B42(3,3,24),V4(3,8),NFLEX(3,8)   PDAUX
    COMMON/CEULER/ IEULER(30),HIR(3,3,90),ANG(3,30),ANGD(3,30), SLIP
*    FE(3,30),TQE(3,30),CONST(5,30)                          SLIP
    COMMON/TEMPVS/ T(3,30),VXT(3),DDMY(35020)                  80386
    DIMENSION SD(3,3,30),E1(30),NTST(30),LSEG(30),RGTTL(4)    PDAUX
    LOGICAL LSEG                                              PDAUX
    DATA NTST/30*0/                                           PDAUX
    DATA RGTTL/8HANG VEL ,8HLIN VEL ,8HANG ACC ,8HLIN ACC /   PDAUX
    CALL ELTIME(1,6)                                           PDAUX
    MBAG = NGRND                                               PDAUX
    IF (NTST(1).NE.0) GO TO 10                                  PDAUX
    LSEG(1) = .FALSE.                                          VAXCHG
    NTST(1) = 1                                               ATBIII
    DO 5 M=2,MBAG                                             ATBIII
    LSEG(M) = ISING(M).GE.0 .AND. JNT(M-1).NE.0              ATBIII
    IF (IABS(IPIN(M-1)).GE.5 .AND. IEULER(M-1).GE.0) LSEG(M)=.FALSE. SLIP
5 NTST(M) = M                                               PDAUX
    NTST(NGRND) = -NGRND                                       PDAUX
    LSEG(NGRND) = .TRUE.                                       PDAUX
    IF (NFLX.EQ.0) GO TO 10                                    PDAUX
    DO 6 J=1,NFLX                                             PDAUX
    M = NFLEX(2,J)                                           PDAUX
6 NTST(M) = -M                                               PDAUX
10 IF (KDINT.EQ.4) GO TO 48                                   PDAUX
    IF (KDINT.GT.0) GO TO 20                                   PDAUX

```

C		PDAUX
C	KDINT=0 IMPLIES INITIAL CALL FROM DINT. PDAUX TO SUPPLY INITIAL	PDAUX
C	VALUES TO STATE VARIABLES AND COMPUTE VALUE OF NEQ.	PDAUX
C		PDAUX
C		PDAUX
C	(A) SET Q TO IDENTITY QUATERNION	PDAUX
C		PDAUX
	N = 0	PDAUX
	DO 12 M=1,MBAG	PDAUX
	IF (NTST(M).LT.0) GO TO 12	PDAUX
	N = N+1	PDAUX
	REGT(N) = RGTTL(1)	PDAUX
	SEGT(N) = SEG(M)	PDAUX
	E1(N) = 1.0	PDAUX
	DO 11 I=1,3	PDAUX
	XTEST(I,N) = SGTEST(I,1,M)**2	PDAUX
	11 VAR(I,N) = 0.0	PDAUX
	12 CONTINUE	PDAUX
C		PDAUX
C	(B) SEGLP OF REFERENCE SEGMENTS	PDAUX
C		PDAUX
	DO 14 M=1,MBAG	PDAUX
	IF (LSEG(M)) GO TO 14	PDAUX
	N = N+1	PDAUX
	REGT(N) = RGTTL(2)	PDAUX
	SEGT(N) = SEG(M)	PDAUX
	DO 13 I=1,3	PDAUX
	XTEST(I,N) = SGTEST(I,2,M)**2	PDAUX
	13 VAR(I,N) = SEGLP(I,M)	PDAUX
	14 CONTINUE	PDAUX
C		PDAUX
C	(C) WMEG	PDAUX
C		PDAUX
	DO 16 M=1,MBAG	PDAUX
	IF (NTST(M).LT.0) GO TO 16	PDAUX
	N = N+1	PDAUX
	REGT(N) = RGTTL(3)	PDAUX
	SEGT(N) = SEG(M)	PDAUX
	DO 15 I=1,3	PDAUX
	XTEST(I,N) = SGTEST(I,3,M)**2	PDAUX
	15 VAR(I,N) = WMEG(I,M)	PDAUX
	16 CONTINUE	PDAUX
C		PDAUX
C	(D) SEGLV OF REFERENCE SEGMENTS	PDAUX
C		PDAUX
	DO 18 M=1,MBAG	PDAUX
	IF (LSEG(M)) GO TO 18	PDAUX
	N = N+1	PDAUX
	REGT(N) = RGTTL(4)	PDAUX
	SEGT(N) = SEG(M)	PDAUX
	DO 17 I=1,3	PDAUX
	XTEST(I,N) = SGTEST(I,4,M)**2	PDAUX

17	VAR(I,N) = SEGLV(I,M)	PDAUX
18	CONTINUE	PDAUX
	NEQ = 3*N	PDAUX
	GO TO 40	PDAUX
20	IF (KDINT.NE.1) GO TO 30	PDAUX
C		PDAUX
C	KDINT = 1, 1ST STEP IN ADVANCING INTEGRATING INTERVAL,	PDAUX
C	SAVE DC MATRICES IF TIME HAS ADVANCED.	PDAUX
C		PDAUX
	N = 0	PDAUX
	DO 22 M=1,MBAG	PDAUX
	IF (NTST(M).LT.0) GO TO 22	PDAUX
	N = N+1	PDAUX
	DO 21 J=1,3	PDAUX
	DO 21 I=1,3	PDAUX
21	SD(I,J,N) = D(I,J,M)	PDAUX
22	CONTINUE	PDAUX
C		PDAUX
C	KDINT > 0,1 - FETCH SAVED DC MATRICES AND UPDATE BY CURRENT THETA.	PDAUX
C		PDAUX
C	(A) UPDATE D BY Q	PDAUX
C		PDAUX
30	N = 0	PDAUX
	DO 32 M=1,MBAG	PDAUX
	IF (NTST(M).LT.0) GO TO 32	PDAUX
	N = N+1	PDAUX
	EDOTE = VAR(1,N)**2 + VAR(2,N)**2 + VAR(3,N)**2	PDAUX
	IF (EDOTE.GE.1.0) KDINT = -KDINT	PDAUX
	IF (KDINT.LE.0) GO TO 99	PDAUX
	E1(N) = DSQRT(1.0-EDOTE)	PDAUX
	CALL DSETQ(SD(1,1,N),VAR(1,N),EDOTE,E1(N),D(1,1,M))	PDAUX
32	CONTINUE	PDAUX
C		PDAUX
C	KDINT > 6 - STORE STATE VARIABLES INTO PROGRAM ARRAYS.	PDAUX
C		PDAUX
C	(B) SEGLP OF REFERENCE SEGMENTS	PDAUX
C		PDAUX
	DO 35 M=1,MBAG	PDAUX
	IF (LSEG(M)) GO TO 35	PDAUX
	N = N+1	PDAUX
	DO 34 I=1,3	PDAUX
34	SEGLP(I,M) = VAR(I,N)	PDAUX
35	CONTINUE	PDAUX
C		PDAUX
C	(C) WMEG	PDAUX
C		PDAUX
	DO 31 M=1,MBAG	PDAUX
	IF (NTST(M).LT.0) GO TO 31	PDAUX
	N = N+1	PDAUX
	DO 36 I=1,3	PDAUX
36	WMEG(I,M) = VAR(I,N)	PDAUX
31	CONTINUE	PDAUX

C		PDAUX
C	(D) SEGLV OF REFERENCE SEGMENTS	PDAUX
C		PDAUX
	DO 38 M=1,MBAG	PDAUX
	IF (LSEG(M)) GO TO 38	PDAUX
	N = N+1	PDAUX
	DO 37 I=1,3	PDAUX
	37 SEGLV(I,M) = VAR(I,N)	PDAUX
	38 CONTINUE	PDAUX
C		PDAUX
C	CALL DAUX ROUTINE TO COMPUTE DERIVATIVES	PDAUX
C		PDAUX
	40 CALL DAUX(0)	PDAUX
C		PDAUX
C	STORE DERIVATIVES FOR INTEGRATING SUBROUTINE.	PDAUX
C		PDAUX
C	(A) DERIVATIVE OF Q	PDAUX
C		PDAUX
	N = 0	PDAUX
	DO 39 M=1,MBAG	PDAUX
	IF (NTST(M).LT.0) GO TO 39	PDAUX
	N = N+1	PDAUX
	CALL CROSS(VAR(1,N),WMEG(1,M),VXT)	PDAUX
	DO 41 I=1,3	PDAUX
	41 DER(I,N) = 0.5*(E1(N)*WMEG(I,M) + VXT(I))	PDAUX
	39 CONTINUE	PDAUX
	NQUAT = N	PDAUX
C		PDAUX
C	(B) SEGLV OF REFERENCE SEGMENTS	PDAUX
C		PDAUX
	DO 43 M=1,MBAG	PDAUX
	IF (LSEG(M)) GO TO 43	PDAUX
	N = N+1	PDAUX
	DO 42 I=1,3	PDAUX
	42 DER(I,N) = SEGLV(I,M)	PDAUX
	43 CONTINUE	PDAUX
C		PDAUX
C	(C) WMEGD	PDAUX
C		PDAUX
	DO 47 M=1,MBAG	PDAUX
	IF (NTST(M).LT.0) GO TO 47	PDAUX
	N = N+1	PDAUX
	DO 44 I=1,3	PDAUX
	44 DER(I,N) = WMEGD(I,M)	PDAUX
	47 CONTINUE	PDAUX
C		PDAUX
C	(D) SEGLA OF REFERENCE SEGMENTS	PDAUX
C		PDAUX
	DO 46 M=1,MBAG	PDAUX
	IF (LSEG(M)) GO TO 46	PDAUX
	N = N+1	PDAUX
	DO 45 I=1,3	PDAUX

45 DER(I,N) = SEGLA(I,M)	PDAUX
46 CONTINUE	PDAUX
IF (KDINT.NE.4) GO TO 99	PDAUX
48 N = 0	PDAUX
DO 51 M=1,MBAG	PDAUX
IF (NTST(M).LT.0) GO TO 51	PDAUX
N = N+1	PDAUX
E1(N) = 1.0	PDAUX
DO 50 I=1,3	PDAUX
DER(I,N) = 0.5*WMEG(I,M)	PDAUX
50 VAR(I,N) = 0.0	PDAUX
51 CONTINUE	PDAUX
99 IF (KDINT.EQ.2) KDINT = NQUAT	PDAUX
CALL ELTIME(2,6)	PDAUX
RETURN	PDAUX
END	PDAUX

	SUBROUTINE PLEDG(AREAL, BD, PL)		PLEDG
C		REV IV 12/11/87	HYFIX
	IMPLICIT REAL*8(A-H,O-Z)		PLEDG
	LOGICAL AREAL		PLEDG
	DIMENSION BD(24), PL(24)		HYFIX
	DIMENSION HAREA(2,2,5), ZC(3,14), X(3), UV(3,2), IV(14)		HYFIX
C	SHARED WITH PLELP-PLSEGF		HYFIX
	COMMON/TEMPVS/DMNT(3,3), DHNT(3,3), DUM1(18), TM(3), R(3), RM(3),		HYFIX
X	DUM2(9), UP(3), VP(3), U(3), V(3), EU(3), EV(3), ET(3),		HYFIX
X	A(2), B(2), CC(2), DUM4(12), TH(3), XH(3), RMD(3), RND(3),		HYFIX
X	APT(2,2,2), AC(2,2), BC(2,2), AFP, E(2,2), DELT, AREA,		HYFIX
X	AB, BB, BT(2), XNC(3), UH(3), P, AMR, FM, T4(3), ALIM(2,2)		HYFIX
*	, DMY(34965)		80386
	EQUIVALENCE (UV(1,1), U(1))		HYFIX
	EQUIVALENCE (ALIM(1,1), BMIN), (ALIM(1,2), AMIN)		HYFIX
	EQUIVALENCE (ALIM(2,1), BMAX), (ALIM(2,2), AMAX)		HYFIX
	EQUIVALENCE (AC(1,1), BB1), (AC(1,2), AA1)		HYFIX
	EQUIVALENCE (AC(2,1), BB2), (AC(2,2), AA2)		HYFIX
	EQUIVALENCE (BC(1,1), AB1), (BC(1,2), BA1)		HYFIX
	EQUIVALENCE (BC(2,1), AB2), (BC(2,2), BA2)		HYFIX
C			HYFIX
	AREA = 0.0		PLEDG
	AREAL = .FALSE.		PLEDG
	CALCULATE CENTER OF ELLIPSE IN PLANE		PLEDG
C	T4 IS VECTOR FROM CENTER OF ELLIPSOID TO CENTER OF ELLIPSE		PLEDG
	DO 10 I = 1,3		PLEDG
	T4(I) = FM*XH(I)		HYFIX
	10 XNC(I) = XNC(I) + T4(I)		PLEDG
C	XNC P1 TO CENTER OF ELLIPSE		PLEDG
C	PUT PLANE VECTORS IN ELLIPSE SYSTEM TH IS PLANE VECTOR		PLEDG
	IF (BD(1).LT.0.0) CALL MAT33(BD(8), DMNT, DHNT)		HYPER
	IF (BD(1).LT.0.0) GO TO 20		HYPER
	DO 15 I = 1,3		HYPER
	DO 15 J = 1,3		HYPER
	15 DHNT(I,J) = DMNT(I,J)		HYPER
	20 CALL MAT31(DHNT, PL(8), UP)		HYPER
	CALL MAT31(DHNT, PL(13), VP)		HYPER
	CALL MAT31(DHNT, PL(18), U)		HYPER
	CALL MAT31(DHNT, PL(21), V)		HYPER
C	U IS P2 - P1, V IS P3 - P1, PLANE VECTOR IS TM		PLEDG
	CALCULATE CENTER FROM P1 IN U, V COORDINATES		PLEDG
	B(1) = (UP(1)*XNC(1) + UP(2)*XNC(2) + UP(3)*XNC(3))/PL(12)		PLEDG
	B(2) = (VP(1)*XNC(1) + VP(2)*XNC(2) + VP(3)*XNC(3))/PL(17)		PLEDG
	AMIN = -B(1)		HYFIX
	AMAX = 1.0 - B(1)		HYFIX
	BMIN = -B(2)		HYFIX
	BMAX = 1.0 - B(2)		HYFIX
C	GET ELLIPSE EQUATION		PLEDG
	DO 25 I = 1,2		HYPER
	DO 25 J = 1,2		HYPER
	25 E(I,J) = 0.0		HYPER
	IF (BD(1).GT.0.0) GO TO 35		HYPER

C TREAT HYPER AS ELLIPSE FOR FIRST GUESS	HYPER
DO 30 I = 1,3	HYPER
EU(I) = U(I)*BD(I+16)	HYPER
30 EV(I) = V(I)*BD(I+16)	HYPER
C GET INTERSECTION OF PLANE WITH BOX	HYFIX
CALL HYBOX(BD(2),TH,T4,MB,ZC,IV)	HYPER
IF (MB.LT.6) GO TO 140	HYPER
GO TO 40	HYPER
35 CALL MAT31(BD(7),U,EU)	HYPER
CALL MAT31(BD(7),V,EV)	HYPER
40 DO 45 K = 1,3	HYPER
E(1,1) = E(1,1) + U(K)*EU(K)	HYPER
E(1,2) = E(1,2) + V(K)*EU(K)	HYPER
45 E(2,2) = E(2,2) + V(K)*EV(K)	HYPER
DELT = E(1,1)*E(2,2) - E(1,2)**2	PLEDGE
C WHAT ABOUT AMR FOR HYPER?? 1 - FM**P ?	HYFIX
R2D = AMR/DELT	HYFIX
COMPUTE BOUNDS OF ELLIPSOID LOCATION OF MAX AND MIN ALPHA	HYFIX
AA2 = DSQRT(E(2,2)*R2D)	HYFIX
AA1 = -AA2	HYFIX
C BA IS VALUE OF BETA AT AT ALPHA MAX	HYFIX
BA1 = E(1,2)*AA2/E(2,2)	HYFIX
BA2 = -BA1	HYFIX
IF (BD(1).GE.-2.0) GO TO 50	HYPER
CALL HYBND(MB,ZC,IV,UP,-1.,X)	HYPER
CALL HYLIM(AA1,U,BA1,V,FM,XH,X,BD)	HYFIX
50 AMIN = DMAX1(AA1,AMIN)	HYFIX
IF (AMIN.GE.AMAX) GO TO 140	HYFIX
IF (BD(1).GE.-2.0) GO TO 55	HYPER
CALL HYBND(MB,ZC,IV,UP,1.,X)	HYPER
CALL HYLIM(AA2,U,BA2,V,FM,XH,X,BD)	HYFIX
55 AMAX = DMIN1(AA2,AMAX)	HYFIX
IF (AMIN.GE.AMAX) GO TO 140	HYPER
COMPUTE BOUNDS OF ELLIPSOID LOCATION OF MAX AND MIN BETA	HYFIX
BB2 = DSQRT(E(1,1)*R2D)	HYFIX
BB1 = -BB2	HYFIX
C AB IS VALUE OF ALPHA AT AT BETA MAX	HYFIX
AB1 = E(1,2)*BB2/E(1,1)	HYFIX
AB2 = -AB1	HYFIX
IF (BD(1).GE.-2.0) GO TO 60	HYPER
CALL HYBND(MB,ZC,IV,VP,-1.,X)	HYPER
CALL HYLIM(BB1,V,AB1,U,FM,XH,X,BD)	HYFIX
60 BMIN = LMAX1(BB1,BMIN)	HYFIX
IF (BMIN.GE.BMAX) GO TO 140	HYFIX
IF (BD(1).GE.-2.0) GO TO 65	HYPER
CALL HYBND(MB,ZC,IV,VP,1.,X)	HYPER
CALL HYLIM(BB2,V,AB2,U,FM,XH,X,BD)	HYFIX
65 BMAX = DMIN1(BB2,BMAX)	HYFIX
IF (BMIN.GE.BMAX) GO TO 140	HYPER
COMPUTE ALPHA'S AT BMIN AND BMAX; BETA'S AT AMIN AND AMAX IF NOT ON	HYFIX
C ELLIPSOID	HYFIX
IF (BD(1).LT.-2.0) GO TO 80	HYPER

HAREA(1,1,L+1) = AMAX	HYFIX
HAREA(2,1,L+1) = APT(1,2,2)	HYFIX
HAREA(1,2,L+1) = AMAX	HYFIX
HAREA(2,2,L+1) = APT(2,2,2)	HYFIX
IF (APT(2,2,2).GE.APT(1,2,2)) L = L + 1	HYFIX
HAREA(1,1,L+1) = APT(2,2,1)	HYFIX
HAREA(2,1,L+1) = BMAX	HYFIX
HAREA(1,2,L+1) = APT(1,2,1)	HYFIX
HAREA(2,2,L+1) = BMAX	HYFIX
IF (APT(2,2,1).GE.APT(1,2,1)) L = L + 1	HYFIX
IF (L.LE.1) GO TO 140	HYFIX
HAREA(1,1,L+1) = HAREA(1,1,1)	HYFIX
HAREA(2,1,L+1) = HAREA(2,1,1)	HYFIX
IF (BD(1).GE.-2) CALL PLREA(L,HAREA,AREA,AB,BB,E,DELT,AMR)	HYFIX
IF (BD(1).LT.-2) CALL HYREA(L,HAREA,AREA,AB,BB)	HYFIX
AREAL = AREA.GT.0.0	HYFIX
IF (.NOT.AREAL) GO TO 140	HYPER
C	HYPER
DO 120 I = 1,3	HYPER
RM(I) = AB*U(I) + BB*V(I) + T4(I)	HYPER
120 RMD(I) = RM(I)	HYPER
COMPUTE POINT ON ELLIPSOID BELOW CENTROID (CONTACT POINT?)	PLEDGE
CONVERT PLANE VECTOR, ET = E*TM	PLEDGE
C TRY TO USE OTHER LOGIC	HYFIX
IF(BD(1).LT.0.0)GO TO 130	HYPER
CALL MAT31(BD(7),TM,ET)	PLEDGE
A2 = TM(1)*ET(1) + TM(2)*ET(2) + TM(3)*ET(3)	PLEDGE
A1 = AB*(TM(1)*EU(1)+TM(2)*EU(2)+TM(3)*EU(3))	HYFIX
1+FM+ BB*(TM(1)*EV(1)+TM(2)*EV(2)+TM(3)*EV(3))	HYFIX
A1 = A1/A2	HYFIX
A0 = (AB**2*E(1,1) + 2.*AB*BB*E(1,2) + BB**2*E(2,2) - AMR)/A2	HYFIX
DISC = A1**2 - A0	PLEDGE
IF(DISC.LT.0.0)DISC = 0.0	PLEDGE
P = A1 + DSQRT(DISC)	PLEDGE
GO TO 140	HYPER
COMPUTE FOR HYPER	HYPER
130 CALL HYVAL(CA,TH,RM,BD,1)	HYFIX
P = -CA	HYFIX
CALL DOT31(BD(8),RMD,RM)	HYPER
140 RETURN	HYPER
END	PLEDGE

	SUBROUTINE PLELP(M,MM,N,NN,NT)		PLELP
C		REV IV 02/07/87	HYPER
	IMPLICIT REAL*8(A-H,O-Z)		PLELP
	LOGICAL AREAL		EDGE
	COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(1250),TAB(4500)		PLELP
	COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),		PLELP
*	SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)		PLELP
	COMMON/FORCES/PSF(7,70),BSF(4,20),SSF(10,40),BAGSF(3,20),		NCFORC
*	PRJNT(7,30),NPANEL(5),NPSF,NBSF,NSSF,NBGSF		PLELP
	COMMON/CNTRSF/ PL(24,30),BELT(20,8),TPTS(6,8),BD(24,40)		EDGE
	COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),		PLELP
*	HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12),		PLELP
*	RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),		PLELP
*	KQ1(12),KQ2(12),KQTYPE(12)		PLELP
	COMMON/RSAVE/ XSG(3,20,3),DPMI(3,3,30),LPMI(30),		TGMOD7
*	NSG(9),MSG(20,9),MCG,MCGIN(24,5),KREF(20,9)		TGMOD7
	COMMON/TEMPVS/DMNT(3,3),TEMP(3,3),B(3,3),XMN(3),RLN(3),XMM(3),		PLELP
*	TM(3),R(3),RM(3),DMNWN(3),RLM(3),RN(3),VMN(3),VR(3),		PLELP
*	WNM(3),WCM(3),WCN(3),VREL(3),FFM(3),FR(3),TQM(3),		PLELP
*	TQN(3),TQNT(3),T(3),H(3),TH(3),XH(3),RMD(3),RND(3),		EDGE
*	TD(3),TT4(3,4),TT5(3,4),XNC(3),UH(3),P,AMR,FM,CF,		EDGE
*	VRM,VRT,VRTS,VRTST,TF,ELOSS,MCF,NCF,DMMY(34964)		80386
	CALL ELTIME(1,21)		PLELP
	CALL DOT33(D(1,1,M),D(1,1,N),DMNT)		PLELP
	DO 10 I = 1,3		PLELP
10	XMN(I) = SEGLP(I,M) - SEGLP(I,N)		PLELP
	CALL MAT31(D(1,1,M),XMN,XMM)		PLELP
	CALL MAT31(DMNT,PL(1,NN),TM)		PLELP
	CALL MAT31(DMNT,PL(5,NN),TD)		EDGE
	BET = 0.0		EDGE
	J = 3		HYPER
	IF(BD(1,MM).LT.0.0) J = 4		HYPER
	DO 15 I=1,3		EDGE
	J = J + 1		HYPER
	XNC(I) = XMM(I) + BD(J,MM) - TD(I)		HYPER
15	BET = BET - TM(I)*XNC(I)		EDGE
C			EDGE
C	BET IS FROM CENTER OF FIGURE TO PLANE		EDGE
	IF(BD(1,MM).GT.0.0)GO TO 30		HYPER
C	PUT PLANE VECTOR INTO HYPER		HYPER
	CALL MAT31(BD(8,MM),TM,TH)		HYPER
	CALL MAT31(BD(8,MM),XNC,UH)		HYPER
	DO 20 I = 1,3		HYPER
	XNC(I) = UH(I)		HYPER
	UH(I) = DABS(TH(I))*BD(I+1,MM)/BD(I+19,MM)		HYPER
	R(I) = BD(I+19,MM)/(BD(I+19,MM) - 1.0)		HYPER
20	RND(I) = UH(I)**R(I)		HYPER
	ALP = HYPEN(BD(1,MM),R,RND)		HYPER
	DO 25 I = 1,3		HYPER
	POW = 1.0/(BD(I+19,MM) - 1.0)		HYPER
	XH(I) = -DSIGN(BD(I+1,MM))*(UH(I)*ALP)**POW,TH(I))		HYPER
25	RND(I) = XH(I)		HYPER

```

BTE = TH(1)*XH(1) + TH(2)*XH(2) + TH(3)*XH(3)          HYPER
FM = BET/BTE                                             HYPER
AMR = 1.0 - DABS(FM)**(-BD(1,MM))                       HYPER
GO TO 35                                                 HYPER
C CODE FOR ELLIPSE          XH = E'T                   EDGE
C                                                                EDGE
30 CALL MAT31(BD(16,MM),TM,XH)                          HYPER
   BTS = TM(1)*XH(1) + TM(2)*XH(2) + TM(3)*XH(3)      EDGE
   BTE = - DSQRT(BTS)                                   PLELP
   FM = BET/BTS                                         EDGE
   AMR = 1.0 -BET*FM                                    EDGE
C                                                                EDGE
35 P = BET - BTE                                         HYPER
   PSF(1,NPSF) = P                                       PLELP
   MCF = NTAB(NT+1)                                       PLELP
   NCF = -MCF                                             PLELP
   IF(NCF.GT.0)CFQQ(NCF) = -999.                         PLELP
   IF(P.LE.0.0) GO TO 85                                  HYPER
C                                                                EDGE
C CALL EDGE ROUTINE TO FIND IF ELLIPSOID INTERSECTS FINITE PLANE  EDGE
C IF IT DOES; AREAL WILL BE TRUE, P WILL BE PENETRATION AT CENTROID  EDGE
C AND RM WILL BE LOCATION OF CENTROID                   EDGE
C RM IS REFERENCED TO CENTER OF ELLIPSOID              EDGE
C USE OLD FORMULA FOR ROLL-SLIDE?, I.E. ROLL-SLIDE SHOULDN'T  EDGE
C CALL PLEDG                                             EDGE
C                                                                EDGE
   LT = NTAB(NT)                                         EDGE
   IF(TAB(LT+22).LE.0.0)GO TO 40                         HYPER
C                                                                EDGE
   IF (AMR.LE.0.0) GO TO 85                               HYPER
   IF (BD(1,MM).LT.0.0.AND.BD(23,MM).NE.0.0) STOP 22  HYPER
   CALL PLEDG(AREAL,BD(1,MM),PL(1,NN))                  EDGE
   IF(.NOT.AREAL)GO TO 85                               HYPER
   PSF(1,NPSF) = P                                       EDGE
C                                                                EDGE
40 IF (TAB(LT+22).GT.-2.0.AND.AMR.LE.0.0) GO TO 85     HYPER
   RHO = 0.0                                             HYPER
   IF(MCF.GT.0)RHO = TAB(MCF+4)                          PLELP
   BETE = 1.0 + RHO*P/BTE                                HYPER
   IF(BD(1,MM).GT.0.0)BETE = BETE/BTE                  HYPER
   IF(BD(1,MM).LT.0.0)CALL DOT31(BD(8,MM),RND,XH)      HYPER
   TRT = P*(1.0 - RHO)                                   EDGE
   J = 3                                                 HYPER
   IF(BD(1,MM).LT.0.0)J = 4                             HYPER
   DO 45 I = 1,3                                         HYPER
   J = J + 1                                             HYPER
   IF(TAB(LT+22).LE.0.0)RM(I) = BETE*XH(I)             EDGE
   IF(TAB(LT+22).GT.0.0)RM(I) = RM(I) - TRT*TM(I)      EDGE
   RLM(I) = RM(I) + BD(J,MM)                            HYPER
45 RN(I) = RLM(I) + XMM(I)                              HYPER
   CALL DOT31(DMNT,RN,RLN)                              PLELP
   IF (TAB(LT+22).GT.0.0) GO TO 55                     HYPER

```


	IF (TAB(LT+22).GT.-3.0.AND.TAB(LT+22).LT.0.0) GO TO 55	HYPER
C		EDGE
C	CHECK BOUNDARY USING OLD METHOD	EDGE
	DO 50 I = 8,13,5	HYPER
	IF(PL(I+4,NN).LE.0.0)GO TO 50	HYPER
	DIST = RLN(1)*PL(I ,NN)	PLELP
	* + RLN(2)*PL(I+1,NN)	PLELP
	* + RLN(3)*PL(I+2,NN) - PL(I+3,NN)	PLELP
	IF((DIST.LE.0.0).OR.(DIST.GT.PL(I+4,NN))) GO TO 85	HYPER
	50 CONTINUE	HYPER
C		EDGE
	55 CALL PLSEGF(M,N,NT)	HYPER
C	DMNWN,VMN,VR,WNM,WCM,WCN,VREL,FFM,FR,TQM,TQN,TQNT,T	EDGE
C	FM,CF,VRM,VRT,VRTS,VRTEST,TF,ELOSS	EDGE
C		EDGE
C	STORE RESULTS	EDGE
	DO 60 I = 1,3	HYPER
	60 PSF(I+4,NPSF) = RLN(I)	HYPER
	IF(LPMI(N).NE.0) CALL DOT31(DPMI(1,1,N),RLN,PSF(5,NPSF))	EDGE
	IF(MCF.LT.0)GO TO 65	HYPER
	PSF(2,NPSF) = FM	PLELP
	PSF(3,NPSF) = 0.0	PLELP
	TRT = TF**2 - FM**2	PLELP
	IF(TRT.GT.0.0) PSF(3,NPSF) = DSQRT(TRT)	PLELP
	PSF(4,NPSF) = TF	PLELP
	GO TO 85	HYPER
C		PLELP
C	ROLL-SLIDE	REVISÉ
		8/18/85
	65 DO 70 I = 1,3	HYPER
	70 PSF(I+1,NPSF) = T(I)	HYPER
	IF(BD(1,MM).LT.0.0) STOP 28	HYPER
	CALL CROSS(TM,WNM,TH)	EDGE
	CALL MAT31(BD(16,MM),TH,UH)	EDGE
	TRT = (TM(1)*UH(1) + TM(2)*UH(2) + TM(3)*UH(3))/BTS	EDGE
	DO 75 I = 1,3	HYPER
	75 RMD(I) = DABS(BETE)*(UH(I) - TRT*XH(I))	HYPER
	CALL CROSS(DMNWN, TM, TH)	EDGE
	CALL CROSS(WNM,RMD,XNC)	EDGE
	SQQ(NCF) = 0.0	PLELP
	DO 80 I = 1,3	HYPER
	80 SQQ(NCF) = SQQ(NCF) + TM(I)*XNC(I) - 2.0*TH(I)*VR(I)	HYPER
	CALL DOT31(D(1,1,M),XNC,RQQ(1,NCF))	EDGE
	85 CALL ELTIME(2,21)	HYPER
	RETURN	PLELP
	END	PLELP

	SUBROUTINE PLREA(L,H,AREA,AB,BB,E,D,R)		HYFIX
C		REV IV 12/11/87	HYFIX
	IMPLICIT REAL*8(A-H,O-Z)		PLREA
	COMPUTES AREA AND CENTROID (TRUE AREA = AREA* UxV /6)		HYFIX
	C UxV IS NEVER COMPUTED UxV = UxV.T = AREA OF PARALLELOGRAM		PLREA
	C THIS ROUTINE WILL ONLY BE CALLED IF THERE IS AN INTERSECTION		PLREA
	DIMENSION H(2,2,5),E(2,2)		HYFIX
	AREA = 0.0		PLREA
	AB = 0.0		PLREA
	BB = 0.0		PLREA
	IF (L.LE.1) GO TO 15		HYFIX
	C = R/DSQRT(D)		HYFIX
	C12 = 2.0*R/D		HYFIX
	C11 = C12*E(1,1)		HYFIX
	C22 = C12*E(2,2)		HYFIX
	C12 = C12*E(1,2)		HYFIX
	DO 10 I = 1,L		HYFIX
	COMPUTE FOR STRAIGHT LINE SEGMENTS		HYFIX
	AR = H(1,1,I)*H(2,2,I) - H(1,2,I)*H(2,1,I)		HYFIX
	IF (AR.EQ.0.0) GO TO 5		HYFIX
	AB = AB + AR*(H(1,1,I) + H(1,2,I))		HYFIX
	BB = BB + AR*(H(2,1,I) + H(2,2,I))		HYFIX
	AREA = AREA + AR		HYFIX
	COMPUTE FOR ELLIPSE		HYFIX
	5 AR = H(1,2,I)*H(2,1,I+1) - H(1,1,I+1)*H(2,2,I)		HYFIX
	IF (AR.EQ.0.0) GO TO 10		HYFIX
	ARC = AR/C		HYFIX
	IF (DABS(ARC).GT.1.0) ARC = DSIGN(1.0D0,ARC)		HYFIX
	AR = C*DASIN(ARC)		HYFIX
	X21 = H(1,1,I+1) - H(1,2,I)		HYFIX
	Y21 = H(2,1,I+1) - H(2,2,I)		HYFIX
	AB = AB + C12*X21 + C22*Y21		HYFIX
	BB = BB - C11*X21 - C12*Y21		HYFIX
	AREA = AREA + AR		HYFIX
	10 CONTINUE		HYFIX
	IF (AREA.LE.0.0) GO TO 15		HYFIX
	AREA = 3.0*AREA		HYFIX
	AB = AB/AREA		PLREA
	BB = BB/AREA		PLREA
C	AREA = AREA/6.0		HYFIX
	15 RETURN		PLREA
	END		PLREA

```

SUBROUTINE PLSEGF(M,N,NT)
C                                     REV III.5 09/03/85TGMOD7
IMPLICIT REAL*8 (A-H,O-Z)
COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),
*          SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)
COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),
*          HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12),
*          RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),
*          KQ1(12),KQ2(12),KQTYPE(12)
COMMON/TEMPVI/ CREST,TTI(3),R1I(3),R2I(3),JSTOP(4,2,30)
COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(1250),TAB(4500)DIMENB
C THIS COMMON/TEMPVS/ IS SHARED BY PLELP, PLSEGF AND SEGSEG.
COMMON/TEMPVS/DMNT(3,3),TEMP(3,3),B(3,3),XMN(3),RLN(3),XMM(3),
*          TM(3),R(3),RM(3),DMNWN(3),RLM(3),RN(3),VMN(3),VR(3),
*          WMN(3),WCM(3),WCN(3),VREL(3),FFM(3),FR(3),TQM(3),
*          TQN(3),TQNT(3),T(3),H(3),T1(3),T2(3),RMD(3),RND(3),
*          TD(3),TT4(3,4),TT5(3,4),T3(3),T4(3),P,AMR,FM,CF,
*          VRM,VRT,VRTS,VRTEST,TF,ELOSS,MCF,NCF,T5(3),T6(3)
*          ,DDMY(34958)
VRTEST = 2.0
CALL MAT31(DMNT,WMEG(1,N),DMNWN)
DO 15 I=1,3
  VMN(I) = SEGLV(I,M) - SEGLV(I,N)
15 WMN(I) = DMNWN(I) - WMEG(I,M)
  CALL DOT31(D(1,1,M),TM,T)
  CALL MAT31(D(1,1,M),VMN,VR)
  CALL CROSS(WMEG(1,M),RLM,WCM)
  CALL CROSS(DMNWN,RN,WCN)
  VRM = 0.0
  DO 16 I=1,3
    VR(I) = VR(I) + WCM(I) - WCN(I)
16 VRM = VRM + VR(I)*TM(I)
  VRT = 0.0
  DO 17 I=1,3
    VREL(I) = VR(I) - VRM*TM(I)
17 VRT = VRT + VREL(I)**2
  VRT = DSQRT(VRT)
  CF = EVALFD(P,NTAB(NT+5),1)
  LT = NTAB(NT)
  TAB(LT) = P
  FM = 1.0
  PDOT = -VRM
  ELOSS = 0.0
  IF (MCF.GT.0) CALL FRCDFL(P,PDOT,NT,1,FM,ELOSS)
  VRTS = VRT
  IF (VRT.LT.VRTEST) VRT = VRTEST/(2.0-VRT/VRTEST)
  FF = -DABS(FM)*CF/VRT
  IF (NCF.GT.0.AND.KQTYPE(NCF).EQ 6)  FF=0.0
  FS = (VRTS-VRT)/VRT
  IF (NCF.GT.0.AND.KQTYPE(NCF).EQ 6)  FS=0.0
  TF = 0.0
  L = LT+18

```

DO 18 I=1,3	PLSEGF
L = L+1	PLSEGF
FFM(I) = FM*TM(I) + FF*VREL(I) + FS*TAB(L)	PLSEGF
TF = TF + FFM(I)**2	PLSEGF
TTI(I) = T(I)	PLSEGF
R1I(I) = RLM(I)	PLSEGF
18 R2I(I) = RLN(I)	PLSEGF
TF = DSQRT(TF)	PLSEGF
MT = NTAB(NT+5)	PLSEGF
CREST = TAB(MT+3)	PLSEGF
CALL DOT31 (D(1,1,M),FFM,FR)	PLSEGF
IF (MCF.LE.0) GO TO 21	PLSEGF
CALL CROSS (RLM,FFM,TQM)	PLSEGF
CALL CROSS (RN,FFM,TQNT)	PLSEGF
CALL DOT31 (DMNT,TQNT,TQN)	PLSEGF
DO 19 I=1,3	PLSEGF
U1(I,M) = U1(I,M) + FR(I)	PLSEGF
U1(I,N) = U1(I,N) - FR(I)	PLSEGF
U2(I,M) = U2(I,M) + TQM(I)	PLSEGF
19 U2(I,N) = U2(I,N) - TQN(I)	PLSEGF
IF (NCF.LE.0) GO TO 23	PLSEGF
21 DO 22 I=1,3	PLSEGF
HQQ(I,NCF) = FR(I)/TF	PLSEGF
TQQ(I,NCF) = T(I)	PLSEGF
RK1(I,NCF) = RLM(I)	PLSEGF
22 RK2(I,NCF) = RLN(I)	PLSEGF
CFQQ(NCF) = CF	PLSEGF
MT = NTAB(NT+5)	PLSEGF
IF (KQTYPE(NCF).EQ.3) CFQQ(NCF) = TAB(MT+4)	PLSEGF
23 RETURN	PLSEGF
END	PLSEGF

```

SUBROUTINE PLTKYZ(P,C)                                PLTKYZ
C                                                      REV III.5 05/30/85VEHICL
C STORES PLOT CHARACTER (C) INTO PLOTYZ, PLOTXZ AND PLOTXY ARRAYS PLTKYZ
C IN VEHICLE REFERENCE FOR POINT (P) GIVEN IN INERTIAL REFERENCE. PLTKYZ
C                                                      PLTKYZ

  IMPLICIT REAL*8 (A-H,O-Z)                            PLTKYZ
  COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND, PLTKYZ
  *              NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG PAGE
  COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30), PLTKYZ
  *              SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30) PLTKYZ
  COMMON/VPOSTN/ ZPLT(3),SPLT(3),AXV(3.6),VATAB(6.501,6). VEHICL
  *              VTO(6),VDI(6),TIMEV(6),OMEGV(6),NVTAB(6),INDXV(6) PLTKYZ
  COMMON/TEMPVS/ DUM(101),PLOTYZ(96,55),PLOTXZ(96,55),PLOTXY(96,55) PLTKYZ
  *              ,DMY(33032) 80386
  LOGICAL*1 C,PLOTYZ,PLOTXZ,PLOTXY PLTKYZ
  DIMENSION P(3),TMP(3),XYZ(3) PLTKYZ
  DATA NPLTZ/96/ , NPLTX/55/ PLTKYZ

C                                                      PLTKYZ
C CONVERT P FROM INERTIAL TO VEHICLE REFERENCE BY PLTKYZ
C   XYZ = DVEH(P-XCOMP) PLTKYZ
C                                                      PLTKYZ

  DO 10 I=1,3 PLTKYZ
10  TMP(I) = P(I) - SEGLP(I,NVEH) PLTKYZ
  CALL MAT31(D(1,1,NVEH),TMP,XYZ) PLTKYZ

C                                                      PLTKYZ
C CONVERT XYZ INTO PLOT CORDINATES IX,IY,IZ AND PLTKYZ
C IF WITHIN PLOT LIMITS, STORE C IN PLOTYZ, PLOTXZ AND PLOTXY. PLTKYZ
C                                                      PLTKYZ

  IX = SPLT(1)*XYZ(1) + ZPLT(1) + 0.5 PLTKYZ
  IZ = SPLT(3)*XYZ(3) + ZPLT(3) + 0.5 PLTKYZ
  IF (IZ.LT.1 .OR. IZ.GT.NPLTZ) GO TO 11 PLTKYZ
  IY = SPLT(2)*XYZ(2) + ZPLT(2) + 0.5 PLTKYZ
  IF (IY.GE.1 .AND. IY.LE.NPLTX) PLOTYZ(IZ,IY) = C PLTKYZ
  IF (IX.GE.1 .AND. IX.LE.NPLTX) PLOTXZ(IZ,IX) = C PLTKYZ
11  IY = -SPLT(3)*XYZ(2) + ZPLT(2) + 0.5 PLTKYZ
  IF (IY.LT.1 .OR. IY.GT.NPLTZ) GO TO 99 PLTKYZ
  IF (IX.GE.1 .AND. IX.LE.NPLTX) PLOTXY(IY,IX) = C PLTKYZ
99  RETURN PLTKYZ
  END PLTKYZ

```

SUBROUTINE POSTPR(PRDT) REV IV 02/01/88MISDOT

CONTROLS GENERATION OF PRINTED TABULAR TIME HISTORIES
AND PLOTS BY THE VALUE OF NPRT(4) 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 PRINTED BY SUBROUTINE OUTPUT.

COMMON/CDINT/ JDTPPTS(18), ZZ(1000,3), DMMY(8068)
NOTE: THIS OVERWRITES COMMON /CDINT/.

COMMON/CONTRL/ TIME, NSEG, NJNT, NPL, NBLT, NBAG, NVEH, NGRND,
* NS, NQ, NSD, NFLX, NHRNSS, NWINDF, NJNTF, NPRT(36), NPG
REAL*8 TIME
COMMON/FORCES/ PSF(7,70), BSF(4,20), SSF(10,40), BAGSF(3,20),
* PRJNT(7,30), NPANEL(5), NPSF, NBSF, NSSF, NBGSF
REAL*8 PSF, BSF, SSF, BAGSF, PRJNT
COMMON/TITLES/ DATE(3), COMENT(40), VPSTTL(20), BDYTTL(5),
* BLTTTL(5,8), PLTTTL(5,30), BAGTTL(5,6), SEG(30),
* JOINT(30), CGS(30), JS(30)
REAL DATE, COMENT, VPSTTL, BDYTTL, BLTTTL, PLTTTL, BAGTTL, SEG, JOINT
LOGICAL*1 CGS, JS
COMMON/CNSNTS/ PI, RADIANT, G, THIRD, EPS(24),
* UNITL, UNITM, UNITT, GRAVITY(3), TWOPI
REAL*8 PI, RADIANT, G, THIRD, EPS, UNITL, UNITM, UNITT, GRAVITY, TWOPI
COMMON/JBARTZ/ MNPL(30), MNBLT(8), MNSEG(30), MNBAG(6),
* MPL(3,5,30), MBLT(3,5,8), MSEG(3,5,30), MBAG(3,10,6),
* NTPL(5,30), NTBLT(5,8), NTSEG(5,30)
COMMON / APSDM(3,20), APSDN(3,20), ASD(5,20), MSDM(20), MSDN(20)
REAL*8 .. 1, APSDN, ASD
COMMON/HRNESS/ 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)
REAL*8 BAR, BB, BBDOT, PLOSS, XLONG, HTIME
COMMON/RSAVE/ XSG(3,20,3), DPMI(3,3,30), LPMI(30),
* NSG(9), MSG(20,9), MCG, MCGIN(24,5), KREF(20,9)
REAL*8 XSG, DPMI, TDATA, UMSEC, PRDT, TEST1, TEST2, VDT1
REAL*8 VDT2, R30, R26
NOTE: SUBROUTINES POSTPR & HEDING SHARE THIS COMMON/TEMPVS/.
THE FIRST DIMENSION OF XLAB, YLAB, PLB1 AND PLB2 SHOULD BE THE SAME
AS THE VALUE ASSIGNED TO NW60 WHICH IS THE NUMBER OF WORDS THAT
IS NECESSARY TO CONTAIN 60 CONSECUTIVE CHARACTERS DEPENDING ON THE

```

C      COMPUTER SYSTEM THIS PROGRAM IS OPERATING ON. THE VALUE OF NW60      POSTPR
C      SHOULD BE 15 ON IBM 360 AND 370, 10 ON UNIVAC 1108, 6 ON CDC 6600. POSTPR
C      THE LAST TERM IN FORMAT 13 BELOW SHOULD BE 15A4(IBM), 10A6(UNIVAC) POSTPR
C      OR 6A10(CDC). ALSO, THE FIRST DIMENSION OF PLDATA IN SUBROUTINE      POSTPR
C      HEDING SHOULD BE 97(IBM), 77(UNIVAC) OR 61(CDC).                      REDIM2
C                                                                              POSTPR
      COMMON/TEMPVS/ TDATA(14,65),HEDATA(470),                              POSTPR
*          XO(20),XN(20),XL(20),XS(20),XLAB(15,20),PLB1(15,20),          POSTPR
*          YO(20),YN(20),YL(20),YS(20),YLAB(15,20),PLB2(15,20),          POSTPR
*          NYP(20),MX(2,20),MY(2,10,20),NX(20),NY(20),                  POSTPR
*          NXLAB(20),NYLAB(20),NPLB1(20),NPLB2(20),                      POSTPR
*          USEC(45),Z(1000,25),ZTTH(14,45,65),IDMMY                      80386
      CHARACTER*4 XLAB,YLAB,PLB1,PLB2                                     80386
      COMMON/DEVICE/ IDEF,IOPORT,MODEL                                   80386
      LOGICAL LTABH,LPLOT                                             POSTPR
      DATA LPP/45/ , NZD1/1000/, NZD2/25/                            PLTINC
      DATA NW60/15/                                                  POSTPR
      LTABH = .FALSE.                                                POSTPR
      LPLOT = .FALSE.                                                POSTPR
      NPRT4 = IABS(NPRT(4))                                           POSTPR
      LPLOT = NPRT4.EQ.1 .OR. NPRT4.EQ.3                               POSTPR
      LTABH = NPRT4.EQ.2 .OR. NPRT4.EQ.3                               POSTPR
      IF(NPRT(26).EQ.4) LTABH = .FALSE.                               TGMOD1
      IF(NPRT(26).GE.5) GO TO 99                                       TGMOD1
C                                                                              POSTPR
C      READ INPUT CARD H.11 TO CONTROL COMPUTATION OF HIC, HSI & CSI.     WINDOP
C                                                                              POSTPR
      READ (5,11) JDTPPTS                                             POSTPR
      WRITE(6,700) NPG                                               PAGE
      NPG=NPG+1                                                       PAGE
700 FORMAT(1H1,122X,'PAGE',15/,2X,                                    PAGE
*          'POSTPROCESSOR CONTROL PARAMETERS',/)                       PAGE
      WRITE(6,701)                                                    CHGIII
701 FORMAT(13X,'HIC & HSI POINT',7X,'CSI POINT')                    CHGIII
      WRITE(6,702) JDTPPTS(1),JDTPPTS(2)                             CHGIII
702 FORMAT(5X,'H.11',10X,I2,17X,I2,/)                                WINDOP
      NDPT = 0                                                         POSTPR
      IHIC = 0                                                         TGMOD1
      I26 = 0                                                         TGMOD1
      ITST1 = 0                                                        TGMOD1
      ITST2 = 0                                                        TGMOD1
      IF(NPRT(26).LT.0) I26 = IABS(NPRT(26))                          TGMOD1
      IF(JDTPPTS(1).GT.0.OR.JDTPPTS(2).GT.0) IHIC = 1                TGMOD1
      IF(NPRT(30).EQ.0.AND.NPRT(26).EQ.3) ITST1 = 1                  TGMOD1
      IF(NPRT(30).LT.I26) ITST2 = 1                                    TGMOD1
      IF(IHIC.EQ.1.AND.ITST1.EQ.1) WRITE(6,751)                       TGMOD1
      IF(IHIC.EQ.1.AND.ITST2.EQ.1) WRITE(6,752) NPRT(30),I26        TGMOD1
751 FORMAT(3X,'WARNING' LOGIC OF INPUT INDICATES USER ANTICIPATES HICTGMOD1
* , HSI AND CSI TO BE COMPUTED BASED ON DATA FOR EVERY SUCCESSFUL', TGMOD1
*/,10X,'INTEGRATION STEP, YET DATA WAS STORED (WRITTEN TO TAPE8) EVTGMOD1
*ERY DT. ')                                                         TGMOD1
752 FORMAT(3X,'WARNING! LOGIC OF INPUT INDICATES USER ANTICIPATES HICTGMOD1

```

```

*, HSI AND CSI TO BE COMPUTED BASED ON DATA FOR EVERY ',I2,/,10X,' TGMOD1
*INTEGER MULTIPLE OF DT, YET DATA WAS STORED (WRITTEN TO TAPE8) EVETGMOD1
*RY ',I2,' INTEGER MULTIPLE OF DT.') TGMOD1
IF(JDTPTS(1).GT.0.AND.NPRT(26).EQ.2.AND.NPRT(30).LT.1) STOP 91 TGMOD1
IF(JDTPTS(2).GT.0.AND.NPRT(26).EQ.2.AND.NPRT(30).LT.1) STOP 92 TGMOD1
IF (JDTPTS(1).NE.0) NDPT = NDPT + 1 POSTPR
IF (JDTPTS(2).NE.0) NDPT = NDPT + 1 POSTPR
IF (.NOT.LPLOT .AND. .NOT.LTABH .AND. NDPT.EQ.0) GO TO 99 POSTPR
CALL ELTIME (1,36) POSTPR
IF (.NOT.LPLOT) GO TO 20 POSTPR
C POSTPR
C READ INDICES OF VARIABLES TO BE PLOTTED AND POSTPR
C ARGUMENTS TO SUBROUTINE SLPLOT ON CARDS I. POSTPR
C POSTPR
C INPUT CARD I.1 POSTPR
C POSTPR
READ (5,11) NPLT , (NYP(K),K=1,NPLT) POSTPR
11 FORMAT (18I4) POSTPR
IF(NPLT.GT.0.AND.ITST1.EQ.1) WRITE(6,753) TGMOD1
IF(NPLT.GT.0.AND.ITST2.EQ.1) WRITE(6,754) NPRT(30),I26 TGMOD1
753 FORMAT(3X,'WARNING! LOGIC OF INPUT INDICATES USER ANTICIPATES PLOTGMOD1
*TS TO BE COMPUTED BASED ON DATA FOR EVERY SUCCESSFUL INTEGRATION STGMOD1
*TEP',/,10X,'YET DATA WAS STORED (WRITTEN TO TAPE8) EVERY DT.') TGMOD1
754 FORMAT(3X,'WARNING! LOGIC OF INPUT INDICATES USER ANTICIPATES PLOTGMOD1
*TS TO BE COMPUTED BASED ON DATA FOR EVERY ',I2,/,10X,'INTEGER MULTITGMOD1
*PLE OF DT, YET DATA WAS STORED (WRITTEN TO TAPE8) EVERY ',I2, TGMOD1
* ' INTEGER MULTIPLE OF DT.') TGMOD1
IF (NPLT.LE.0) LPLOT = .FALSE. POSTPR
IF (.NOT.LPLOT) GO TO 20 POSTPR
DO 15 K=1,NPLT POSTPR
NYPLT = NYP(K) POSTPR
C POSTPR
C INPUT CARD I.2.K POSTPR
C POSTPR
READ (5,11) MX(1,K), MX(2,K), (MY(1,J,K), MY(2,J,K), J=1,NYPLT) POSTPR
C POSTPR
C INPUT CARD I.3.K POSTPR
C POSTPR
READ (5,12) NX(K), X0(K), XN(K), XL(K), XS(K) POSTPR
12 FORMAT (I4 , 4X , 4F8.0 ) POSTPR
C POSTPR
C INPUT CARD I.4.K POSTPR
C POSTPR
READ (5,12) NY(K), Y0(K), YN(K), YL(K), YS(K) POSTPR
C POSTPR
C INPUT CARD I.5.K POSTPR
C POSTPR
READ (5,13) NXLAB(K), (XLAB(I,K),I=1,NW60) POSTPR
13 FORMAT (I4 , 4X , 15A4) POSTPR
C POSTPR
C NOTE - ABOVE FORMAT ASSUMES 4 ALPHANUMERIC CHARACTERS FOR SINGLE POSTPR
C PRECISION WORDS ON IBM 360 AND 370 COMPUTERS THE 15A4 TERM IN THEPOSTPR

```


C	FORMAT WILL HAVE TO BE CHANGED ON NON-IBM COMPUTERS TO PRODUCE A	POSTPR
C	CONTINUOUS STRING OF 60 CHARACTERS IN CORE MEMORY.	POSTPR
C		POSTPR
C	INPUT CARD I.6.K	POSTPR
C		POSTPR
C	READ (5,13) NYLAB(K), (YLAB(I,K), I=1, NW60)	POSTPR
C		POSTPR
C	INPUT CARD I.7.K	POSTPR
C		POSTPR
C	READ (5,13) NPLB1(K), (PLB1(I,K), I=1, NW60)	POSTPR
C		POSTPR
C	INPUT CARD I.8.K	POSTPR
C		POSTPR
C	15 READ (5,13) NPLB2(K), (PLB2(I,K), I=1, NW60)	POSTPR
C		CHGIII
C	WRITE OUT PLOTTING CONTROL DATA	CHGIII
C		CHGIII
	WRITE(6,703)	CHGIII
	703 FORMAT(4X, 'PLOTTING CONTROLS', /)	CHGIII
	WRITE(6,704)	CHGIII
	704 FORMAT(12X, 'NO. PLOTS', 11X, 'NO. OF Y VARIABLES PER PLOT')	CHGIII
	WRITE(6,705) NPLT, (NYP(JK), JK=1, NPLT)	CHGIII
	705 FORMAT(5X, 'I.1', 7X, I2, 7X, 20(I2, 2X))	CHGIII
	WRITE(6,706)	CHGIII
	706 FORMAT(12X, 'MX1 MX2 MY1A MY2A MY1B MY2B MY1C MY2C MY1D MY2D MY1E MCHGIII	CHGIII
	*Y2E MY1F MY2F MY1G MY2G MY1H MY2H MY1I MY2I MY1J MY2J')	CHGIII
	DO 730 IJ=1, NPLT	CHGIII
	WRITE(6,707) IJ, MX(1, IJ), MX(2, IJ).	CHGIII
	* (MY(1, L, IJ), MY(2, L, IJ), L=1, NYP(IJ))	CHGIII
	707 FORMAT(5X, 'I.2.', I2, 2X, I2, 2X, I2, 2X, 20(I2, 3X))	CHGIII
	730 CONTINUE	CHGIII
	WRITE(6,708)	CHGIII
	708 FORMAT(14X, 'NX', 8X, 'X0', 9X, 'XN', 8X, 'XL', 9X, 'XS')	CHGIII
	DO 731 IJ=1, NPLT	CHGIII
	WRITE(6,709) IJ, NX(IJ), X0(IJ), XN(IJ), XL(IJ), XS(IJ)	CHGIII
	709 FORMAT(5X, 'I.3.', I2, 2X, I3, 4X, 4(F8.3, 2X))	CHGIII
	731 CONTINUE	CHGIII
	WRITE(6,710)	CHGIII
	710 FORMAT(14X, 'NY', 8X, 'Y0', 9X, 'YN', 8X, 'YL', 9X, 'YS')	CHGIII
	DO 732 IJ=1, NPLT	CHGIII
	WRITE(6,711) IJ, NY(IJ), Y0(IJ), YN(IJ), YL(IJ), YS(IJ)	CHGIII
	711 FORMAT(5X, 'I.4.', I2, 2X, I3, 4X, 4(F8.3, 2X))	CHGIII
	732 CONTINUE	CHGIII
	WRITE(6,712)	CHGIII
	712 FORMAT(12X, 'NXLAB', 15X, 'XLAB')	CHGIII
	DO 733 IJ=1, NPLT	CHGIII
	WRITE(6,713) IJ, NXLAB(IJ), (XLAB(L, IJ), L=1, NW60)	CHGIII
	713 FORMAT(5X, 'I.5.', I2, 2X, I3, 5X, 15A4)	CHGIII
	733 CONTINUE	CHGIII
	WRITE(6,714)	CHGIII
	714 FORMAT(12X, 'NYLAB', 15X, 'YLAB')	CHGIII
	DO 734 IJ=1, NPLT	CHGIII

WRITE(6,715) IJ,NYLAB(IJ),(YLAB(L,IJ),L=1,NW60)	CHGIII
715 FORMAT(5X,'I.6.',I2,2X,I3,5X,15A4)	CHGIII
734 CONTINUE	CHGIII
WRITE(6,716)	CHGIII
716 FORMAT(12X,'NPLB1',15X,'PLB1')	CHGIII
DO 735 IJ=1,NPLT	CHGIII
WRITE(6,717) IJ,NPLB1(IJ),(PLB1(L,IJ),L=1,NW60)	CHGIII
717 FORMAT(5X,'I.7.',I2,2X,I3,5X,15A4)	CHGIII
735 CONTINUE	CHGIII
WRITE(6,718)	CHGIII
718 FORMAT(12X,'NPLB2',15X,'PLB2')	CHGIII
DO 736 IJ=1,NPLT	CHGIII
WRITE(6,719) IJ,NPLB2(IJ),(PLB2(L,IJ),L=1,NW60)	CHGIII
719 FORMAT(5X,'I.8.',I2,2X,I3,5X,15A4)	CHGIII
736 CONTINUE	CHGIII
C	POSTPR
C READ TIME HISTORY DATA FROM TAPE 8.	POSTPR
C	POSTPR
20 NPTS = 0	POSTPR
LINES = 0	POSTPR
IF (NPRT(4).GT.0) REWIND 8	POSTPR
READ (8,END=29) NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,NPANEL,	POSTPR
* MNPL,MNBLT,MNSEG,MNBAG,MPL,MBLT,MSEG,MBAG	POSTPR
READ (8,END=29) DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTTL,BAGTTL,	POSTPR
* SEG,JOINT,UNITL,UNITM,UNITT,NSG,MSG,XSG,MCG,	ATBIII
* MCGIN,KREF,NHRNSS,NBLTPH,NPTSPB,NSD,MSDM,MSDN	CHGIII
21 READ (8,END=29) NT,UMSEC,((TDATA(I,J),I=1,14),J=1,NT)	POSTPR
R30 = 1.0D0	TGMOD1
IF(NPRT(30).GT.0) R30 = NPRT(30)	TGMOD1
VDT1 = R30*PRDT	TGMOD1
TEST1 = DMOD(UMSEC,VDT1)	TGMOD1
TEST1 = DMIN1(TEST1,DABS(VDT1 - TEST1))	TGMOD1
IF(NPRT(30).GT.0.AND.TEST1.GT.EPS(4)) GO TO 25	TGMOD1
NPTS = NPTS + 1	POSTPR
IF (NPTS.GT.NZD1 .AND. (NDPT.NE.0 .OR. LPLOT)) STOP 52	ATBIII
ZZ(NPTS,1) = UMSEC	PLTINC
Z(NPTS,1) = UMSEC	PLTINC
IF (NDPT.EQ.0) GO TO 22	POSTPR
C	POSTPR
C STORE DATA FOR HIC, HSI AND CSI.	POSTPR
C	POSTPR
JJ = 1	POSTPR
DO 61 I=1,2	POSTPR
IF (JDTPTS(I).EQ.0) GO TO 61	POSTPR
JJ = JJ + 1	POSTPR
JD = JDTPTS(I) - 1	POSTPR
JE = 4*MOD(JD,3) + 4	POSTPR
JP = JD/3 + 1	POSTPR
ZZ(NPTS,JJ) = TDATA(JE,JP)	PLTINC
61 CONTINUE	POSTPR
22 IF (.NOT.LPLOT) GO TO 25	POSTPR
C	POSTPR

C	STORE DATA FOR PLOTTING	POSTPR
C		POSTPR
	JY = 1	PLTINC
	DO 24 K=1,NPLT	POSTPR
	JE = IABS(MX(2,K))	POSTPR
	IF (JE.EQ.0) GO TO 23	POSTPR
	JY = JY + 1	POSTPR
	IF (JY.GT.NZD2) STOP 53	ATBIII
	JP = MX(1,K) - 20	POSTPR
	Z(NPTS,JY) = TDATA(JE,JP)	POSTPR
23	NYPLT = NYP(K)	POSTPR
	DO 24 J=1,NYPLT	POSTPR
	JY = JY + 1	POSTPR
	JP = MY(1,J,K) - 20	POSTPR
	IF (JY.GT.NZD2) STOP 54	ATBIII
	JE = IABS(MY(2,J,K))	POSTPR
	Z(NPTS,JY) = UMSEC	POSTPR
24	IF (JE.NE.0) Z(NPTS,JY) = TDATA(JE,JP)	POSTPR
25	IF (.NOT.LTABH) GO TO 21	POSTPR
C		POSTPR
C	STORE DATA TO PRINT TABULAR TIME HISTORIES	POSTPR
C		POSTPR
	R26 = 1.0D0	TGMOD1
	IF(NPRT(26).LT.0) IFLG = 1	TGMOD1
	IF(IFLG.EQ.1) N26 = IABS(NPRT(26))	TGMOD1
	IF(IFLG.EQ.1) R26 = N26	TGMOD1
	VDT2 = R26*PRDT	TGMOD1
	TEST2 = DMOD(UMSEC,VDT2)	TGMOD1
	TEST2 = DMIN1(TEST2,DABS(VDT2 - TEST2))	TGMOD1
	IF (NPRT(26).LE.0 .AND. TEST2.GT.EPS(4)) GO TO 21	TGMOD1
	LINES = LINES + 1	POSTPR
	NTTH = MOD(LINES-1,LPP) + 1	POSTPR
	USEC(NTTH) = UMSEC	POSTPR
	DO 26 J=1,NT	POSTPR
	DO 26 I=1,14	POSTPR
26	ZTTH(I,NTTH,J) = TDATA(I,J)	POSTPR
	IF (NTTH.EQ.LPP) CALL HEDING (LINES,LPP)	POSTPR
	GO TO 21	POSTPR
29	IF (.NOT.LTABH .OR. LINES.EQ.0) GO TO 30	POSTPR
	IF (NTTH.NE.LPP) CALL HEDING (LINES,LPP)	POSTPR
30	IF (NDPT.NE.0) CALL HICCSI(NPTS)	POSTPR
	IF (.NOT.LPLOT) GO TO 98	POSTPR
C		POSTPR
C	PLOT DATA VIA SUBROUTINE SLPLOT.	POSTPR
C		POSTPR
C	INCLUDE ANY PROGRAM STATEMENTS HERE REQUIRED BY YOUR COMPUTER AND	POSTPR
C	PLOTTING SYSTEMS FOR PLOT INITIALIZATION (E.G., CALL PLOTS).	POSTPR
C		POSTPR
	CALL PLOTS (IDEF,IOPORT,MODEL)	80386
	JZ = 1	PLTINC
	DO 50 K=1,NPLT	POSTPR
	JX = 1	POSTPR

```

IF (MX(2,K).EQ.0) GO TO 42                                POSTPR
JZ = JZ + 1                                               POSTPR
JX = JZ                                                    POSTPR
IF (Z(1,JX).EQ.0.0 .OR. MX(2,K).GE.0) GO TO 42          POSTPR
DO 41 I=2,NPTS                                           POSTPR
41 Z(I,JX) = Z(I,JX) - Z(1,JX)                            POSTPR
Z(1,JX) = 0.0                                            POSTPR
42 NYPLT = NYP(K)                                         POSTPR
DO 44 J=1,NYPLT                                          POSTPR
JY = JZ + J                                              POSTPR
IF (Z(1,JY).EQ.0.0 .OR. MY(2,J,K).GE.0) GO TO 44        POSTPR
DO 43 I=2,NPTS                                           POSTPR
43 Z(I,JY) = Z(I,JY) - Z(1,JY)                            POSTPR
Z(1,JY) = 0.0                                            POSTPR
44 CONTINUE                                               POSTPR
NXK = NX(K)                                               POSTPR
NYK = NY(K)                                               POSTPR
XOK = XO(K)                                               POSTPR
YOK = YO(K)                                               POSTPR
XNK = XN(K)                                               POSTPR
YNK = YN(K)                                               POSTPR
XLK = XL(K)                                               POSTPR
YLK = YL(K)                                               POSTPR
XSK = XS(K)                                               POSTPR
YSK = YS(K)                                               POSTPR
NXLABK = NXLAB(K)                                         POSTPR
NYLABK = NYLAB(K)                                         POSTPR
NPLB1K = NPLB1(K)                                         POSTPR
NPLB2K = NPLB2(K)                                         POSTPR
CALL SLPLOT(Z(1,JX ), NXK, XOK, XNK, XLK, XSK, XLAB(1,K), NXLABK, POSTPR
*              Z(1,JZ+1), NYK, YOK, YNK, YLK, YSK, YLAB(1,K), NYLABK, POSTPR
*              NPTS,NYPLT,NZD1,PLB1(1,K),NPLB1K,PLB2(1,K),NPLB2K) POSTPR
C                                                         POSTPR
C   INSERT ANY CODE REQUIRED BY YOUR SYSTEM TO ADVANCE PLOT PAGES HERE POSTPR
C                                                         POSTPR
IF(NPRT(31).EQ.1) GO TO 444                               CHGIII
X00 = -0.5*(XSK-(XLK-0.5)) + XLK + 3.0                   FXPLOT
Y00 = -0.5*(YSK-(YLK-1.0))                               FXPLOT
CALL PLOT (X00,Y00,-3)                                    FXPLOT
50 JZ = JZ + NYPLT                                        POSTPR
444 CONTINUE                                             CHGIII
C                                                         POSTPR
C   INSERT ANY PLOT TERMINATION CODE REQUIRED BY YOUR SYSTEM HERE. POSTPR
C                                                         POSTPR
CALL PLOT(12.0,0.0,999)                                   PECONV
98 CALL ELTIME (2,36)                                     POSTPR
99 RETURN                                                POSTPR
END                                                        POSTPR

```



```

IF (LPMI(I).EQ.0) GO TO 19
CALL DOT33 (DPMI(1,1,I),D(1,1,I),T3)
CALL DOT31(DPMI(1,1,I),WMEG(1,I),T1)
CALL DOT31(DPMI(1,1,I),WMEGD(1,I),T2)
CALL YPRDEG (T3,YPR)
WRITE (6,31) I,SEG(I),YPR,(T1(K),K=1,3),(T2(K),K=1,3)
GO TO 20
19 CALL YPRDEG (D(1,1,I),YPR)
WRITE (6,31) I,SEG(I),YPR,(WMEG(K,I),K=1,3),(WMEGD(K,I),K=1,3)
20 CONTINUE
WRITE(6,770)
770 FORMAT(//,1X,23X,'(INERTIAL)',27X,'(INERTIAL)',32X,'(INERTIAL)')
WRITE (6,22) UNITL,UNITL,UNITT
22 FORMAT(18X,'LINEAR POSITION (' ,A4,')',
*      13X,'LINEAR VELOCITY (' ,A4,'/' ,A4,')',
*      16X,'LINEAR ACCELERATIONS (G''S)'/
*      ' SEGMENT',10X,'X',10X,'Y',10X,'Z',
*      13X,'X',11X,'Y',11X,'Z',15X,'X',13X,'Y',13X,'Z'/)
DO 30 I=1,MBAG
DO 29 K=1,3
29 T1(K) = SEGLA(K,I)/G
30 WRITE (6,31) I,SEG(I),(SEGLP(K,I),K=1,3),(SEGLV(K,I),K=1,3),T1
31 FORMAT(13,1X,A4,3X,3F11.4,3X,3F12.5,3X,3F14.6)
IF (NSEG.GT.6) WRITE (6,32) NPG
IF (NSEG.GT.6) NPG=NPG+1
32 FORMAT('1',122X,'PAGE',I5)
WRITE(6,775)
775 FORMAT(//,1X,23X,'(INERTIAL)',29X,'(LOCAL)')
WRITE (6,33) UNITL,UNITT,UNITT,UNITM,UNITL
33 FORMAT(18X,'U1 ARRAY (' ,A4,'/' ,A4,'**2)',
*      14X,'U2 ARRAY (RAD/' ,A4,'**2)',
*      25X,'KINETIC ENERGY'/
*      15X,'EXTERNAL LINEAR ACCELERATIONS',
*      8X,'EXTERNAL ANGULAR ACCELERATIONS',
*      22X,'(' ,A4,'-' ,A4,')'/
*      ' SEGMENT',10X,'X',10X,'Y',10X,'Z',13X,'X',11X,'Y',11X,'Z',
*      14X,'LINEAR',7X,'ANGULAR',7X,'TOTAL'/)
DO 80 J=1,3
80 TKE(J)=0.0
DO 34 I=1,NSEG
V=SEGLV(1,I)**2+SEGLV(2,I)**2+SEGLV(3,I)**2
SKE(1)=0.5*W(I)*V/G
SKE(2)=0.0
DO 81 J=1,3
81 SKE(2)=SKE(2)+0.5*PHI(J,I)*WMEG(J,I)**2
SKE(3)=SKE(1)+SKE(2)
DO 82 J=1,3
82 TKE(J)=TKE(J)+SKE(J)
IF (LPMI(I).EQ.0) GO TO 73
CALL DOT31 (DPMI(1,1,I),U2(1,I),T1)
WRITE (6,61) I,SEG(I),(U1(K,I),K=1,3),
*      (T1(K),K=1,3),(SKE(K),K=1,3)

```

	GO TO 34	PRINT
73	CONTINUE	PRINT
	WRITE (6,61) I,SEG(I),(U1(K,I),K=1,3),	KINETIC
	* (U2(K,I),K=1,3),(SKE(K),K=1,3)	KINETIC
61	FORMAT(I3,1X,A4,3X,3(D11.4,1X),3X,3(D12.5,1X),3X,3(D12.5,1X))	KINETIC
34	CONTINUE	FIXPRT
	WRITE(6,83) (TKE(K),K=1,3)	KINETIC
83	FORMAT(1X,98X,'TOTAL BODY KINETIC ENERGY'/	KINETIC
	* 1X,90X,3(1X,D12.5))	KINETIC
	IF (NJNT.LE.0) GO TO 39	PRINT
	WRITE(6,776)	CHGIII
776	FORMAT(/,1X,27X,'(INERTIAL)',27X,'(INERTIAL)')	CHGIII
	WRITE (6,35) UNITM,UNITL,UNITM,UNITT	PRINT
35	FORMAT(24X,'JOINT FORCES (' ,A4,')',	CHGIII
	* 15X,'JOINT TORQUES (' ,2A4,')',	PRINT
	* 9X,'RELATIVE ANGULAR'/	PRINT
	*' JOINT IPIN',9X,'X',10X,'Y',10X,'Z',13X,'X',11X,'Y',11X,'Z',	PRINT
	* 7X,'VELOCITY (RAD/',' ,A4,')'/)	PRINT
	DO 36 J=1,NJNT	PRINT
	IPINJ = IPIN(J)	PRINT
	IF (IABS(IPIN(J)).EQ.4) IPINJ = IEULER(J)	PRINT
	DO 137 II=1,3	MISC
137	T1(II)--TQ(II,J)	MISC
	WRITE (6,37) J,JOINT(J),IPINJ,(F(K,J),K=1,3),(T1(K),K=1,3),WJ(J)	MISC
37	FORMAT(I3,1X,A4,I4,7X,3(D10.3,1X),3X,3(D11.4,1X),3X,F13.3)	FIXPRT
36	CONTINUE	FIXPRT
39	IF (NQ.LE.0) GO TO 99	PRINT
	WRITE (6,41)	CHGIII
	WRITE (6,47)	CHGIII
47	FORMAT(1X,45X,'(INERTIAL)')	CHGIII
	WRITE (6,49) UNITM,UNITL	CHGIII
41	FORMAT(///' OTHER CONSTRAINT FORCES',/)	CHGIII
49	FORMAT(1X,' NO. TYPE SEG1 SEG2',	CHGIII
	* 15X,'CONSTRAINT FORCE (' ,A4,')',	PRINT
	* 16X,'DISTANCE (' ,A4,')'/)	PRINT
	ICH = 0	FIXPRT
	DO 50 J=1,NQ	PRINT
	IF (KQTYPE(J).NE.5) ICH = 0	FIXPRT
	IF (KQTYPE(J).LT.0) GO TO 50	PRINT
	IF (KQTYPE(J).EQ.5) ICH = ICH + 1	FIXPRT
	IF (ICH.EQ.2) GO TO 50	FIXPRT
	M = KQ1(J)	PRINT
	N = KQ2(J)	PRINT
	CALL DOT31(D(1,1,M),RK1(1,J),T1)	PRINT
	CALL DOT31(D(1,1,N),RK2(1,J),T2)	PRINT
	S1 = 0.0	PRINT
	DO 42 I=1,3	PRINT
	HH(I) = SEGLP(I,M)+T1(I) - SEGLP(I,N)-T2(I)	PRINT
42	S1 = S1 + HH(I)**2	PRINT
	SQS1 = DSQRT(S1)	PRINT
	WRITE (6,43) J,KQTYPE(J),SEG(M),SEG(N),(QQ(I,J),I=1,3),SQS1	PRINT
43	FORMAT(I4,I6,4X,A4,2X,A4,3X,3G15.7,6X,G15.7)	PRINT

```
50 CONTINUE
99 IF (NPRT(28).LE.0) NPRT(28) = -1
   RETURN
   END
```

```
PRINT
PRINT
PRINT
PRINT
```



```

SUBROUTINE PRIPLT                                     PRIPLT
C                                                     REV IV    07/24/86SLIP
C PRODUCES PRINTER PLOT OF Y-Z PLANE VIEW AND X-Z PLANE VIEW OF PRIPLT
C BODY SEGMENT CGS, JOINTS AND SELECTED POINTS OF VEHICLE COMPONENTSPRIPLT
C                                                     PRIPLT
C IMPLICIT REAL*8 (A-H,O-Z)                           PRIPLT
COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,  PRIPLT
* NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG     PAGE
COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),PRIPLT
* SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)         PRIPLT
COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60), SLIP
* RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),     PRIPLT
* JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)    PRIPLT
COMMON/JBARTZ/ MNPL( 30),MNBLT( 8),MNSEG( 30),MNBAG( 6), PRIPLT
* MPL(3,5,30),MBLT(3,5,8),MSEG(3,5,30),MBAG(3,10,6), PRIPLT
* NTPL( 5,30),NTBLT( 5,8),NTSEG( 5,30)               PRIPLT
COMMON/CNTRF/ PL(24,30),BELT(20,8),TPTS(6,8),BD(24,40)  EDGE
COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),  PRIPLT
* BLTTTL(5,8),PLTTL(5,30),BAGTTL(5,6),SEG(30),      PRIPLT
* JOINT(30),CGS(30),JS(30)                           PRIPLT
COMMON/HRNESS/ BAR(15,100),BB(100),BBDOT(100),PLOSS(2,100), PRIPLT
* XLONG(20),HTIME(2),IBAR(5,100),NL(2,100),         PRIPLT
* NPTSPB(20),NPTPLY(20),NTHRNS(20),NBLTPH(5)        PRIPLT
REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT PRIPLT
LOGICAL*1 CGS,JS                                       PRIPLT
COMMON/TEMPVS/ TEMP1(3),TEMP(3),TEMP2(3),CJOINT(3,30),BSN(2), PRIPLT
* PLOTYZ(96,55),PLOTXZ(96,55),PLOTXY(96,55)         PRIPLT
* ,DMMY(33032)                                         80386
LOGICAL*1 PLOTYZ,PLOTXZ,PLOTXY,CHARS(7),BLANK,BCHAR   PRIPLT
LOGICAL NPRT5,NPRT6,NPRT7                             PRIPLT
DATA CHARS/',' , '+' , '@' , '!' , ' _ ' , '- ' , '*' / , BLANK/' ' / PRIPLT
DATA ISTEP/0/ , NPLTI/96/ , NPLTJ/55/                PRIPLT
C                                                     PRIPLT
C DETERMINE IF PLOTTING IS TO BE DONE FOR THIS TIME STEP. PRIPLT
C                                                     PRIPLT
C ISTEP = ISTEP+1                                       PRIPLT
NPRT5 = (NPRT(5)).EQ.1)                               PRIPLT
IF (NPRT(5).GT.1) NPRT5 = (MOD(ISTEP,NPRT(5))).EQ.1) PRIPLT
NPRT6 = (NPRT(6)).EQ.1)                               PRIPLT
IF (NPRT(6).GT.1) NPRT6 = (MOD(ISTEP,NPRT(6))).EQ.1) PRIPLT
NPRT7 = (NPRT(7)).EQ.1)                               PRIPLT
IF (NPRT(7).GT.1) NPRT7 = (MOD(ISTEP,NPRT(7)) EQ 1)  PRIPLT
IF (.NOT.NPRT5 .AND. .NOT.NPRT6 .AND. NOT.NPRT7) GO TO 99 PRIPLT
CALL ELTIME(1, 4)                                       PRIPLT
C                                                     PRIPLT
C BLANK OUT PLOT ARRAYS.                               PRIPLT
C                                                     PRIPLT
C DO 10 J=1,NPLTJ                                       PRIPLT
PLOTYZ(1,J) = CHARS(6)                                PRIPLT
PLOTXZ(1,J) = CHARS(6)                                PRIPLT
PLOTXY(1,J) = CHARS(6)                                PRIPLT
DO 10 I=2,NPLTI                                       PRIPLT

```

	PLOTYZ(I,J) = BLANK	PRIPLT
	PLOTXZ(I,J) = BLANK	PRIPLT
10	PLOTXY(I,J) = BLANK	PRIPLT
C		PRIPLT
C	PLOT VEHICLE REFERENCE ORIGIN USING SYMBOL(*).	PRIPLT
C		PRIPLT
	CALL PLTXYZ (SEGLP(1,NVEH),CHARS(7))	PRIPLT
C		PRIPLT
C	PLOT CG OF BODY SEGMENTS USING SEGMENT SYMBOLS.	PRIPLT
C		PRIPLT
	DO 20 I=1,NSEG	PRIPLT
20	CALL PLTXYZ(SEGLP(1,I),CGS(I))	PRIPLT
C		PRIPLT
C	COMPUTE AND PLOT JOINT LOCATIONS USING JOINT SYMBOLS.	PRIPLT
C		PRIPLT
	IF (NJNT.EQ.0) GO TO 40	PRIPLT
	DO 31 J=1,NJNT	PRIPLT
	I = IABS(JNT(J))	PRIPLT
	IF (I.LE.0) GO TO 31	PRIPLT
	CALL DOT31(D(1,1,I),SR(1,2*J-1),TEMP)	PRIPLT
	DO 30 L=1,3	PRIPLT
30	CJOINT(L,J) = TEMP(L)+SEGLP(L,I)	PRIPLT
	CALL PLTXYZ (CJOINT(1,J),JS(J))	PRIPLT
31	CONTINUE	PRIPLT
	IF (NPRT(13).NE.0) WRITE (6,32) ((CJOINT(I,J),I=1,3),J=1,NJNT)	PRIPLT
32	FORMAT ('O JOINT POSITIONS'/(1X,9F14.4))	PRIPLT
C		PRIPLT
C	PLOT BELT ANCHOR, FIXED AND TANGENT POINTS USING SYMBOL(.).	PRIPLT
C		PRIPLT
40	IF (NBLT.LE.0) GO TO 50	PRIPLT
	DO 43 J=1,NBLT	PRIPLT
	IF (MNBLT(J).LE.0) GO TO 43	PRIPLT
	M1 = MBLT(1,1,J)	PRIPLT
	M2 = MBLT(2,1,J)	PRIPLT
	M3 = MBLT(3,1,J)	PRIPLT
	DO 41 I=1,3	PRIPLT
41	TEMP1(I) = BELT(I+6,J) + BD(I+3,M3)	PRIPLT
	CALL DOT31 (D(1,1,M2),TEMP1,TEMP)	PRIPLT
	CALL DOT31 (D(1,1,M1),BELT(1,J),TEMP1)	PRIPLT
	CALL DOT31 (D(1,1,M1),BELT(4,J),TEMP2)	PRIPLT
	DO 42 I=1,3	PRIPLT
	TEMP1(I) = TEMP1(I) + SEGLP(I,M1)	PRIPLT
	TEMP2(I) = TEMP2(I) + SEGLP(I,M1)	PRIPLT
42	TEMP (I) = TEMP (I) + SEGLP(I,M2)	PRIPLT
	CALL PLTXYZ (TEMP1,CHARS(1))	PRIPLT
	CALL PLTXYZ (TEMP2,CHARS(1))	PRIPLT
	CALL PLTXYZ (TEMP,CHARS(1))	PRIPLT
	CALL PLTXYZ (TPTS(1,J),CHARS(1))	PRIPLT
	CALL PLTXYZ (TPTS(4,J),CHARS(1))	PRIPLT
43	CONTINUE	PRIPLT
C		PRIPLT
C	PLOT POINTS IN PLAY ON HARNESS-BELT SYSTEMS USING SYMBOL(.).	PRIPLT

C	50 IF (NHRNSS.LE.0) GO TO 60	PRIPLT
	J1 = 1	PRIPLT
	K1 = 1	PRIPLT
	DO 54 NH=1,NHRNSS	PRIPLT
	IF (NBLTPH(NH).LE.J) GO TO 54	PRIPLT
	J2 = J1 + NBLTPH(NH) - 1	PRIPLT
	DO 53 NB=J1,J2	PRIPLT
	IF (NPTPLY(NB).LE.0) GO TO 53	PRIPLT
	K2 = K1 + NPTPLY(NB) - 1	PRIPLT
	DO 52 K=K1,K2	PRIPLT
	KI = NL(1,K)	PRIPLT
	KS = IABS(IBAR(1,KI))	PRIPLT
	IF (KS.GT.100) KS = MOD(KS,100)	PRIPLT
	CALL DOT31 (D(1,1,KS),BAR(4,KI),TEMP1)	PRIPLT
	CALL DOT31 (D(1,1,KS),BAR(7,KI),TEMP2)	PRIPLT
	DO 51 I=1,3	PRIPLT
	51 TEMP(I) = SEGLP(I,KS) + TEMP1(I) + TEMP2(I)	PRIPLT
	52 CALL PLTXYZ (TEMP,CHARS(1))	PRIPLT
	K1 = K2+1	PRIPLT
	53 CONTINUE	PRIPLT
	J1 = J2+1	PRIPLT
	54 CONTINUE	PRIPLT
C		PRIPLT
C	PLOT CENTER AND END OF AXES OF ELLIPSOIDAL TARGET USING SYMBOLS	PRIPLT
C	(@) FOR CENTER, (-) FOR ENDS OF Z AXIS,(!) FOR ENDS OF X,Y AXES.	PRIPLT
C		PRIPLT
	60 IF (NBAG.EQ.0) GO TO 80	PRIPLT
	BSN(1) = 1.0	PRIPLT
	BSN(2) = -1.0	PRIPLT
	DO 68 J=1,NBAG	PRIPLT
	IF (MNBAG(J).EQ.0) GO TO 68	PRIPLT
	JB = NVEH+J	PRIPLT
	BCHAR = CHARS(5)	PRIPLT
	L2 = 2	PRIPLT
	DO 67 I=1,4	PRIPLT
	IF (I.EQ.3) BCHAR = CHARS(4)	PRIPLT
	IF (I.EQ.4) BCHAR = CHARS(3)	PRIPLT
	IF (I.EQ.4) L2 = 1	PRIPLT
	DO 67 L=1,L2	PRIPLT
	DO 64 K=1,3	PRIPLT
	64 TEMP1(K) = BD(K,3,B)	PRIPLT
	IF (I.EQ.4) GO TO 65	PRIPLT
	TEMP1(I) = TEMP1(I) + BSN(L)*BD(I,JB)	PRIPLT
	65 CALL DOT31 (D(1,1,JB),TEMP1,TEMP2)	PRIPLT
	DO 66 K=1,3	PRIPLT
	66 TEMP2(K) = TEMP2(K) + SEGLP(K,JB)	PRIPLT
	67 CALL PLTXYZ (TEMP2,BCHAR)	PRIPLT
	68 CONTINUE	PRIPLT
C		PRIPLT
C	PRINT Y-Z , X-Z AND X-Y PLANE VIEW PLOTS.	PRIPLT
C		PRIPLT

```

8) TMSC = 1000.0*TIME                                PRIPLT
   IF (.NOT.NPRT5) GO TO 83                            PRIPLT
   WRITE (2,81) TMSC,SEGLP(2,NVEH),SEGLP(3,NVEH)      PRIPLT
81 FORMAT ('1 T=',F10.3,'   YO=',F10.5,'   ZO=',F10.5,'   Y-Z PLANE')PRIPLT
   WRITE (2,82) PLOTYZ                                PRIPLT
82 FORMAT (2X,96A1)                                    PRIPLT
83 IF (.NOT.NPRT6) GO TO 85                            PRIPLT
   WRITE (2,84) TMSC,SEGLP(1,NVEH),SEGLP(3,NVEH)      PRIPLT
84 FORMAT ('1 T=',F10.3,'   XO=',F10.5,'   ZO=',F10.5,'   X-Z PLANE')PRIPLT
   WRITE (2,82) PLOTXZ                                PRIPLT
85 IF (.NOT.NPRT7) GO TO 87                            PRIPLT
   WRITE (2,86) TMSC,SEGLP(1,NVEH),SEGLP(2,NVEH)      PRIPLT
86 FORMAT ('1 T=',F10.3,'   XO=',F10.5,'   YO=',F10.5,'   X-Y PLANE')PRIPLT
   WRITE (2,82) PLOTXY                                PRIPLT
87 CALL ELTIME(2, 4)                                   PRIPLT
99 RETURN                                             PRIPLT
   END                                                 PRIPLT

```

	SUBROUTINE QSET(F,Y,X,DER,N)	QSET
C		REV III.3 10/01/84REVIII
	IMPLICIT REAL*8(A-H,O-Z)	QSET
	DIMENSION F(5,3,80),Y(5,3,80),X(3,80),DER(3,80)	QSET
	DIMENSION T1(3),T2(3),T3(3),T4(3)	QSET
	DO 20 I=1,N	QSET
	E1=DSQRT(1.DO -X(1,I)**2-X(2,I)**2-X(3,I)**2)	QSET
	E1D=- (X(1,I)*DER(1,I)+X(2,I)*DER(2,I)+X(3,I)*DER(3,I))/E1	QSET
	E2=DSQRT(1.DO -Y(1,1,I)**2-Y(1,2,I)**2-Y(1,3,I)**2)	QSET
	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
	UHB=X(1,I)*F(3,1,I)+X(2,I)*F(3,2,I)+X(3,I)*F(3,3,I)	QSET
	UHC=X(1,I)*F(4,1,I)+X(2,I)*F(4,2,I)+X(3,I)*F(4,3,I)	QSET
	UDB=DER(1,I)*F(3,1,I)+DER(2,I)*F(3,2,I)+DER(3,I)*F(3,3,I)	QSET
	UDD=DER(1,I)**2+DER(2,I)**2+DER(3,I)**2	QSET
	EB=(E1D**2+UDD+UHB)/E1	QSET
	EC = (1.5*(UDB-E1D*EB)+UHC+F(5,1,I)*(E1D**2+UDD))/E1	QSET
	T1(1)=X(2,I)*F(3,3,I)-X(3,I)*F(3,2,I)	QSET
	T2(1)=X(2,I)*F(4,3,I)-X(3,I)*F(4,2,I)	QSET
	T3(1)=X(2,I)*Y(1,3,I)-X(3,I)*Y(1,2,I)	QSET
	T4(1)=X(2,I)*Y(2,3,I)-X(3,I)*Y(2,2,I)	QSET
	T1(2)=X(3,I)*F(3,1,I)-X(1,I)*F(3,3,I)	QSET
	T2(2)=X(3,I)*F(4,1,I)-X(1,I)*F(4,3,I)	QSET
	T3(2)=X(3,I)*Y(1,1,I)-X(1,I)*Y(1,3,I)	QSET
	T4(2)=X(3,I)*Y(2,1,I)-X(1,I)*Y(2,3,I)	QSET
	T1(3)=X(1,I)*F(3,2,I)-X(2,I)*F(3,1,I)	QSET
	T2(3)=X(1,I)*F(4,2,I)-X(2,I)*F(4,1,I)	QSET
	T3(3)=X(1,I)*Y(1,2,I)-X(2,I)*Y(1,1,I)	QSET
	T4(3)=X(1,I)*Y(2,2,I)-X(2,I)*Y(2,1,I)	QSET
	DO 20 J=1,3	QSET
	F(3,J,I)=E1*F(3,J,I)-T1(J)+EB*X(J,I)	JTF984
	F(4,J,I)=E1*F(4,J,I)-T2(J)+EC*X(J,I)	JTF984
	Y(3,J,I)=E1*Y(1,J,I)-T3(J)-E2*X(J,I)	JTF984
20	Y(4,J,I)=E1*Y(2,J,I)-T4(J)-E2D*X(J,I)	JTF984
	RETURN	QSET
	END	QSET

	SUBROUTINE QUAT(ANG,Q)		QUAT
		REV IV 07/23/86	TWOPI
C	COMPUTES QUATERNIONS FROM YAW, PITCH, ROLL ANGLES IN DEGREES		QUAT
C	IMPLICIT REAL *8(A-H,O-Z)		QUAT
	DIMENSION ANG(3),Q(4),R(4),T(3)		QUAT
	COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),		QUAT
	* UNITL,UNITM,UNITT,GRAVITY(3),TWOPI		TWOPI
	A = 0.5*ANG(1)*RADIAN		QUAT
	Q(1) = DCOS(A)		QUAT
	Q(2) = 0.0		QUAT
	Q(3) = 0.0		QUAT
	Q(4) = DSIN(A)		QUAT
	K = 3		QUAT
	DO 10 I = 2,3		QUAT
	A = 0.5*ANG(I)*RADIAN		QUAT
	R(1) = DCOS(A)		QUAT
	R(2) = 0.0		QUAT
	R(3) = 0.0		QUAT
	R(4) = 0.0		QUAT
	R(K) = DSIN(A)		QUAT
	DOT = Q(2)*R(2) + Q(3)*R(3) + Q(4)*R(4)		QUAT
	CALL CROSS(Q(2),R(2),T)		QUAT
	DO 5 J = 2,4		QUAT
5	Q(J) = Q(1)*R(J) + R(1)*Q(J) + T(J-1)		QUAT
	Q(1) = Q(1)*R(1) - DOT		QUAT
10	K = 2		QUAT
	SUM = DSQRT(Q(1)**2 + Q(2)**2 + Q(3)**2 + Q(4)**2)		QUAT
	DO 12 I = 1,4		QUAT
12	Q(I) = Q(I)/SUM		QUAT
	RETURN		QUAT
	END		QUAT

```

DOUBLE PRECISION FUNCTION RCRT(A,PL,Z,IP)                                RCRT
C                                                                                   REV 03 07/19/73RCRT
C COMPUTES THE RADIUS OF CURVATURE AT POINT Z OF ELLIPSOID A           RCRT
C IN THE PLANE PL(I,IP) WHERE                                         RCRT
C                                                                                   RCRT
C     A: 3X3 MATRIX DEFINING ELLIPSOID.                                RCRT
C     PL: 4X3 MATRIX CONTAINING THREE ORTHONORMAL VECTORS.           RCRT
C     Z: 3 COORDINATES OF POINT ON THE ELLIPSOID                    RCRT
C     AS MEASURED FROM CENTER OF ELLIPSOID                           RCRT
C     IP: IDENTIFIES THE NORMAL VECTOR OF PLANE IN WHICH THE         RCRT
C     RADIUS OF CURVATURE IS DESIRED.                                  RCRT
C                                                                                   RCRT
C     IMPLICIT REAL*8 (A-H,O-Z)                                        RCRT
C     DIMENSION A(3,3),PL(4,3),Z(3),T(5)                               RCRT
C     DO 10 I=1,5                                                       RCRT
10  T(I) = 0.0                                                           RCRT
C     M = IP+1                                                         RCRT
C     N = IP+2                                                         RCRT
C     IF(M.GT.3) M = M-3                                               RCRT
C     IF(N.GT.3) N = N-3                                               RCRT
C     DO 30 I=1,3                                                       RCRT
C     S1 = 0.                                                           RCRT
C     S2 = 0.                                                           RCRT
C     DO 20 J=1,3                                                       RCRT
C     S1 = S1+A(I,J)*PL(J,M)                                           RCRT
20  S2 = S2+A(I,J)*PL(J,N)                                           RCRT
C     T(1) = T(1)+S1*Z(I)                                              RCRT
C     T(2) = T(2)+S2*Z(I)                                              RCRT
C     T(3) = T(3)+S1*PL(I,M)                                           RCRT
C     T(4) = T(4)+S2*PL(I,N)                                           RCRT
30  T(5) = T(5)+S1*PL(I,N)                                           RCRT
C     W = DSQRT(T(1)**2+T(2)**2)                                         RCRT
C     T(1) = T(1)/W                                                     RCRT
C     T(2) = T(2)/W                                                     RCRT
C     RCRT = W/(T(3)*T(2)**2-2.0*T(1)*T(2)*T(5)+T(4)*T(1)**2)       RCRT
C     IF(RCRT.LT.0.0) RCRT = -RCRT                                     RCRT
C     RETURN                                                            RCRT
C     END                                                                RCRT

```

	SUBROUTINE ROT (A,L,TH)		ROT
C		REV IV	07/23/86TWOPI
C	COMPUTES ROTATION MATRIX A FOR ANGLE TH		ROT
C	ABOUT X,Y OR Z AXIS AS L = 1, 2, OR 3.		ROT
C			ROT
C	ARGUMENTS:		ROT
C	A: 3X3 ROTATION MATRIX TO BE COMPUTED.		ROT
C	L: 1,2 OR 3 TO ROTATE ABOUT X,Y OR Z AXIS.		ROT
C	TH: ANGLE OF ROTATION IN RADIAN.		ROT
C			ROT
	IMPLICIT REAL*8 (A-H,O-Z)		ROT
	DIMENSION A(3,3)		ROT
	COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),		ROT
	* UNITL,UNITM,UNITT,GRAVITY(3),TWOPI		TWOPI
	C=DCOS(TH)		ROT
	S=DSIN(TH)		ROT
	IF (DABS(C) .LT.EPS(8)) C=0.0		CONVER
	IF (DABS(S) .LT.EPS(8)) S=0.0		CONVER
	ONE = 1.0		ROT
	IF (1.0-DABS(C) .LT.EPS(8)) C = DSIGN(ONE,C)		CONVER
	IF (1.0-DABS(S) .LT.EPS(8)) S = DSIGN(ONE,S)		CONVER
	IF (L.EQ.2) S = -S		ROT
	DO 30 I=1,3		ROT
	IF(I.EQ.3)GO TO 20		ROT
	DO 10 J=I,2		ROT
	A(I,J+1)=0.0		ROT
	A(J+1,I)=0.0		ROT
	IF(I+J+L.NE.5)GO TO 10		ROT
	A(I,J+1)=S		ROT
	A(J+1,I)=-S		ROT
	10 CONTINUE		ROT
	20 A(I,I)= C		ROT
	IF(I.EQ.L)A(I,I)=1.0		ROT
	30 CONTINUE		ROT
	RETURN		ROT
	END		ROT

	SUBROUTINE ROTATE	REV IV 02/20/87	ROTATE
C			HYPER
C	THE PURPOSE OF THIS ROUTINE IS TO TRANSFORM THOSE VARIABLES THAT		ROTATE
C	HAVE BEEN SUPPLIED IN LOCAL GEOMETRIC COORDINATES TO PRINCIPAL		ROTATE
C	AXES COORDINATES AS INDICATED BY LPMI(I) # 0 FOR I = 1 TO NSEG.		ROTATE
C			ROTATE
	IMPLICIT REAL*8 (A-H,O-Z)		ROTATE
	COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,		ROTATE
*	NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG		PAGE
	COMMON/RSAVE/ XSG(3,20,3),DPMI(3,3,30),LPMI(30),		ATBIII
*	NSG(9),MSG(20,9),MCC,MCGIN(24,5),KREF(20,9)		TTHKREF
	COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60),		SLIP
*	RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),		ROTATE
*	JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)		ROTATE
	COMMON/CNTRF/ PL(24,30),BELT(20,8),TPTS(6,8),BD(24,40)		EDGE
	COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),		ROTATE
*	SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)		ROTATE
	COMMON/JBARTZ/ MNPL(30),MNBLT(8),MNSEG(30),MNBAG(6),		ROTATE
*	MPL(3,5,30),MBLT(3,5,8),MSEG(3,5,30),MBAG(3,10,6),		ROTATE
*	NTPL(5,30),NTBLT(5,8),NTSEG(5,30)		ROTATE
	COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),		ROTATE
*	HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12),		ROTATE
*	RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),		ROTATE
*	KQ1(12),KQ2(12),KQTYPE(12)		ROTATE
	COMMON/DAMPER/ APSDM(3,20),APSDN(3,20),ASD(5,20),MSDM(20),MSDN(20)		ROTATE
	COMMON/WINDFR/ !TIME(30),QFU(3,5),QFV(3,5),WF(3,30),IWIND(30),		WINDOP
*	MWSEG(7,30),NFVSEG(6),NFVNT(5),MOWSEG(30,30)		WINDOP
	COMMON/HRNESS/ BAR(15,100),BB(100),BBDOT(100),PLOSS(2,100),		ROTATE
*	XLONG(20),HTIME(2),IBAR(5,100),NL(2,100),		ROTATE
*	NPTSPB(20),NPTPLY(20),NTHRNS(20),NBLTPH(5)		ROTATE
	COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(1250),TAB(4500)		WINDROT
	COMMON/CEULER/ IEULER(30),HIR(3,3,90),ANG(3,30),ANGD(3,30),		FXHROT
*	FE(3,30),TQE(3,30),CONST(5,30)		FXHROT
	COMMON/TEMPVS/ T1(3),T3(3,3),LBD(40),T2(3),T4(3,3),DMY(35069)		80386
C			ROTATE
C	TRANSFORM DIRECTION COSINE MATRICES D FROM INPUT CARDS G.3.		ROTATE
C			ROTATE
	LTEST = 0		ROTATE
	DO 20 J=1,30		ROTATE
	IF (J.GT.NSEG) LPMI(J) = 0		ROTATE
	IF (LPMI(J).EQ.0) GO TO 20		ROTATE
	LTEST = 1		ROTATE
	DO 12 I=1,3		ROTATE
	T1(I) = WMEG(I,J)		ROTATE
	DO 12 K=1,3		ROTATE
12	T3(I,K) = D(I,K,J)		ROTATE
	CALL MAT33 (DPMI(1,1,J),T3,D(1,1,J))		ROTATE
	CALL MAT31 (DPMI(1,1,J),T1,WMEG(1,J))		ROTATE
20	CONTINUE		ROTATE
	IF (LTEST.EQ.0) GO TO 99		ROTATE
C			ROTATE
C	TRANSFORM SR,HT AND HB FROM INPUT CARDS B.3.		ROTATE

C	IF (NJNT.LE.0) GO TO 31	ROTATE
	DO 30 J=1,NJNT	ROTATE
	I = IABS(JNT(J))	ROTATE
	M = 2	SLIP
	IF (IABS(IPIN(J)).GT.4) M = 3	SLIP
	DO 24 K=1,2	ROTATE
	IF (I.EQ.0 .OR. LPMI(I).EQ.0) GO TO 24	ROTATE
	IJ = 2*J-2+K	ROTATE
	DO 22 LI=1,3	ROTATE
	T1(LI) = SR(LI,IJ)	ROTATE
	T2(LI) = HB(LI,IJ)	FXHROT
	DO 22 LJ=1,3	ROTATE
	T4(LI,LJ) = HIR(LI,LJ,IJ+30)	FXHROT
22	T3(LI,LJ) = HT(LI,LJ,IJ)	ROTATE
	CALL MAT31 (DPMI(1,1,I),T1,SR(1,IJ))	ROTATE
	CALL MAT31 (DPMI(1,1,I),T2,HB(1,IJ))	FXHROT
	CALL MAT33 (DPMI(1,1,I),T3,HT(1,1,IJ))	ROTATE
	CALL MAT33 (DPMI(1,1,I),T4,HIR(1,1,IJ+30))	FXHROT
24	I = J+1	ROTATE
30	CONTINUE	ROTATE
C		ROTATE
C	TRANSFORM RK1,RK2 FROM INPUT CARDS D.6.	ROTATE
C		ROTATE
31	IF (NQ.LE.0) GO TO 41	ROTATE
	K5 = 0	ROTATE
	DO 40 K=1,NQ	ROTATE
	IF (K5.EQ.1) GO TO 39	ROTATE
	KSEG = KQ1(K)	ROTATE
	IF (LPMI(KSEG).EQ.0) GO TO 36	ROTATE
	DO 35 I=1,3	ROTATE
35	T1(I) = RK1(I,K)	ROTATE
	CALL MAT31 (DPMI(1,1,KSEG),T1,RK1(1,K))	ROTATE
36	KSEG = KQ2(K)	ROTATE
	IF (LPMI(KSEG).EQ.0) GO TO 40	ROTATE
	DO 37 I=1,3	ROTATE
37	T1(I) = RK2(I,K)	ROTATE
	CALL MAT31 (DPMI(1,1,KSEG),T1,RK2(1,K))	ROTATE
39	IF (KQTYPE(K).EQ.5) K5 = 1-K5	ROTATE
40	CONTINUE	ROTATE
C		ROTATE
C	TRANSFORM APSDM,APSDN FROM INPUT CARDS D.8.	ROTATE
C		ROTATE
41	IF (NSD.LE.0) GO TO 151	FIXROT
	DO 50 J=1,NSD	ROTATE
	KSEG = MSDM(J)	ROTATE
	IF (LPMI(KSEG).EQ.0) GO TO 44	ROTATE
	DO 43 I=1,3	ROTATE
43	T1(I) = APSDM(I,J)	ROTATE
	CALL MAT31 (DPMI(1,1,KSEG),T1,APSDM(1,J))	ROTATE
44	KSEG = MSDN(J)	ROTATE
	IF (LPMI(KSEG).EQ.0) GO TO 50	ROTATE

DO 45 I=1,3	ROTATE
45 T1(I) = APSDN(I,J)	ROTATE
CALL MAT31 (DPMI(1,1,KSEG),T1,APSDN(1,J))	ROTATE
50 CONTINUE	ROTATE
C	FIXROT
C TRANSFORM QFU AND QFV FROM INPUT CARDS D.9.	FIXROT
C	FIXROT
151 NFORCE = NFVSEG(6)	FIXROT
IF (NFORCE.LE.0) GO TO 100	WINDROT
DO 152 J=1,NFORCE	FIXROT
KSEG = IABS(NFVSEG(J))	FIXROT
IF (LPMI(KSEG).EQ.0) GO TO 152	FIXROT
DO 143 I=1,3	FIXROT
T1(I) = QFU(I,J)	FIXROT
143 T2(I) = QFV(I,J)	FIXROT
CALL MAT31 (DPMI(1,1,KSEG),T1,QFU(1,J))	FIXROT
CALL MAT31 (DPMI(1,1,KSEG),T2,QFV(1,J))	FIXROT
152 CONTINUE	FIXROT
C	WINDROT
C ROTATE WIND FORCE FUNCTIONS	WINDROT
C	WINDROT
100 IF (NWINDF.EQ.0) GOTO 51	WINDROT
DO 101 I=1,NSEG	WINDROT
IF (MWSEG(1,I).EQ.0) GOTO 101	WINDROT
NT = MWSEG(5,I)	WINDROT
DO 102 J=1,I-1	WINDROT
IF (NT.EQ.MWSEG(5,J)) GOTO 101	WINDROT
102 CONTINUE	WINDROT
KT = NTI(NT)	WINDROT
RK = TAB(KT)	WINDROT
IF (RK.NE.0) GOTO 101	WINDROT
NSR = IDINT(TAB(KT+4))	WINDROT
IF (NSR.EQ.0 .OR. LPMI(NSR).EQ.0) GOTO 101	WINDROT
NENTRY = TAB(KT+5)	WINDROT
K1 = KT+6	WINDROT
K2 = 4*NENTRY+KT+2	WINDROT
DO 103 K=K1,K2,4	WINDROT
DO 104 J=1,3	WINDROT
104 T1(J) = TAB(K+J)	WINDROT
103 CALL MAT31(DPMI(1,1,NSR),T1,TAB(K+1))	WINDROT
101 CONTINUE	WINDROT
C	ROTATE
C CHECK PLANE AND ELLIPSOID ASSIGNMENTS ON INPUT CARDS F.1.	ROTATE
C TRANSFORM PLANE ARRAYS SET UP FROM INPUT CARD D.1.	ROTATE
C	ROTATE
51 DO 52 J=1,40	ROTATE
LBD(J) = 0	ROTATE
52 IF (J.LE.NSEG) LBD(J) = J	ROTATE
IF (NPL.LE.0) GO TO 61	ROTATE
DO 60 J=1,NPL	ROTATE
IF (MNPL(J).EQ.0) GO TO 60	ROTATE
LPL = 0	ROTATE

KPL = MNPL(J)	ROTATE
DO 56 I=1,KPL	ROTATE
M1 = MPL (1,I,J)	ROTATE
M2 = MPL (2,I,J)	ROTATE
M3 = MPL (3,I,J)	ROTATE
IF (LPL.EQ.M1 .OR. LPL.EQ.0) GO TO 54	ROTATE
WRITE (6,53) J,M1,LPL	ROTATE
53 FORMAT('O INPUT ERROR HAS BEEN DETECTED IN SUBROUTINE ROTATE.'/	ROTATE
* ' PLANE NO.',I3,' HAS BEEN ASSIGNED TO BOTH SEGMENTS NO.',	ROTATE
* I3,' AND NO.',I3,'.'/ PROGRAM IS BEING TERMINATED.')	ROTATE
STOP 43	ROTATE
54 LPL = M1	ROTATE
IF (LBD(M3).EQ.M2 .OR. LBD(M3).EQ.0) GO TO 55	ROTATE
WRITE (6,68) M3,M2,LBD(M3)	ROTATE
STOP 44	ROTATE
55 LBD(M3) = M2	ROTATE
56 CONTINUE	ROTATE
IF (LPMI(LPL).EQ.0) GO TO 60	ROTATE
L = 1	EDGE
DO 59 K=1,6	EDGE
IF((K.EQ.3).OR.(K.EQ.6)) L = L-1	EDGE
IF((K.EQ.4).OR.(K.EQ.5)) L = L+1	EDGE
DO 58 I=1,3	ROTATE
T1(I) = PL(L,J)	EDGE
58 L=L+1	EDGE
CALL MAT31 (DPMI(1,1,LPL),T1,PL(L-3,J))	EDGE
59 L=L+1	EDGE
60 CONTINUE	ROTATE
C	ROTATE
C CHECK ELLIPSOID ASSIGNMENTS ON INPUT CARDS F.2.	ROTATE
C TRANSFORM BELT(L,J) FOR L=1,9 FROM INPUT CARDS D.3.	ROTATE
C	ROTATE
61 IF (NBLT.LE.0) GO TO 66	ROTATE
DO 65 J=1,NBLT	ROTATE
IF (MNBLT(J).EQ.0) GO TO 65	ROTATE
KBLT = MNBLT(J)	ROTATE
DO 62 I=1,KBLT	ROTATE
M1 = MBLT(1,I,J)	ROTATE
M2 = MBLT(2,I,J)	ROTATE
M3 = MBLT(3,I,J)	ROTATE
IF (LBD(M3).EQ.M2 .OR. LBD(M3).EQ.0) GO TO 62	ROTATE
WRITE (6,68) M3,M2,LBD(M3)	ROTATE
STOP 45	ROTATE
62 LBD(M3) = M2	ROTATE
IF (LPMI(M1).EQ.0) GO TO 63	ROTATE
DO 57 I=1,3	ROTATE
T3(I,1) = BELT(I ,J)	ROTATE
57 T3(I,2) = BELT(I+3,J)	ROTATE
CALL MAT31 (DPMI(1,1,M1),T3(1,1),BELT(1,J))	ROTATE
CALL MAT31 (DPMI(1,1,M1).T3(1,2),BELT(4,J))	ROTATE
63 IF (LPMI(M2).EQ.0) GO TO 65	ROTATE
DO 64 I=1,3	ROTATE

64	T3(I,3) = BELT(I+6,J)	ROTATE
	CALL MAT31 (DPMI(1,1,M2),T3(1,3),BELT(7,J))	ROTATE
65	CONTINUE	ROTATE
C		ROTATE
C	CHECK ELLIPSOID ASSIGNMENTS ON INPUT CARDS F.3.	ROTATE
C		ROTATE
66	DO 70 J=1,NSEG	ROTATE
	IF (MNSEG(J).EQ.0) GO TO 70	ROTATE
	KSEG = MNSEG(J)	ROTATE
	DO 69 I=1,KSEG	ROTATE
	M1 = MSEG(1,I,J)	ROTATE
	M2 = MSEG(2,I,J)	ROTATE
	M3 = MSEG(3,I,J)	ROTATE
	IF (LBD(M1).EQ.J .OR. LBD(M1).EQ.0) GO TO 67	ROTATE
	WRITE (6,68) M1,J,LBD(M1)	ROTATE
	STOP 46	ROTATE
67	LBD(M1) = J	ROTATE
	IF (LBD(M3).EQ.M2 .OR. LBD(M3).EQ.0) GO TO 69	ROTATE
	WRITE (6,68) M3,M2,LBD(M3)	ROTATE
68	FORMAT('0 INPUT ERROR HAS BEEN DETECTED IN SUBROUTINE ROTATE.'/	ROTATE
	* ' ELLIPSOID NO.',I3,' HAS BEEN ASSIGNED TO BOTH SEGMENTS NO.'	ROTATE
	* I3,' AND NO.',I3,'.'/ PROGRAM IS BEING TERMINATED.')	ROTATE
	STOP 47	ROTATE
69	LBD(M3) = M2	ROTATE
70	CONTINUE	ROTATE
C		ROTATE
C	CHECK ELLIPSOID ASSIGNMENTS ON INPUT CARDS F.6.	ROTATE
C		ROTATE
	IF (NBAG.EQ.0) GO TO 174	TGMOD8
	DO 73 J=1,NBAG	ROTATE
	IF (MNBAG(J).EQ.0) GO TO 73	ROTATE
	KBAG = MNBAG(J)	ROTATE
	DO 72 I=1,KBAG	ROTATE
	M2 = MBAG(2,I,J)	ROTATE
	M3 = MBAG(3,I,J)	ROTATE
	IF (LBD(M3).EQ.M2 .OR. LBD(M3).EQ.0) GO TO 72	ROTATE
	WRITE (6,68) M3,M2,LBD(M3)	ROTATE
	STOP 50	ROTATE
72	LBD(M3) = M2	ROTATE
73	CONTINUE	ROTATE
C		TGMOD8
C	CHECK ELLIPSOID ASSIGNMENTS ON INPUT CARDS F.7.	TGMOD8
C		TGMOD8
174	IF(NWINDF.EQ.0) GO TO 74	TGMOD8
	DO 175 J=1,NSEG	TGMOD8
	M1 = IABS(MWSEG(1,J))	TGMOD8
	IF(M1.EQ.0) GO TO 175	TGMOD8
	M2 = MWSEG(2,J)	TGMOD8
	IF(LBD(M2).EQ.M1 .OR. LBD(M2).EQ.0) GO TO 172	TGMOD8
	WRITE(6,68) M2,M1,LBD(M2)	TGMOD8
	STOP 48	TGMOD8
172	LBD(M2) = M1	TGMOD8

175	CONTINUE	TGMOD8
C		ROTATE
C	CHECK ELLIPSOID ASSIGNMENTS ON INPUT CARDS F.8.	ROTATE
C	TRANSFORM BAR(L,K) FOR L=4,12 FROM INPUT CARDS F.8.D.	ROTATE
C		ROTATE
74	IF (NHRNSS.EQ.0) GO TO 81	ROTATE
	J1 = 1	ROTATE
	K1 = 1	ROTATE
	DO 80 II=1,NHRNSS	ROTATE
	IF (NBLTPH(II).LE.0) GO TO 80	ROTATE
	J2 = J1 + NBLTPH(II) - 1	ROTATE
	DO 79 JJ=J1,J2	ROTATE
	IF (NPTSPB(JJ).LE.0) GO TO 79	ROTATE
	K2 = K1 + NPTSPB(JJ) - 1	ROTATE
	DO 78 K=K1,K2	ROTATE
	M2 = MOD(IBAR(1,K),100)	ROTATE
	M3 = IBAR(2,K)	ROTATE
	IF (M3.EQ.0) GO TO 88	BUTLER1
	IF (LBD(M3).EQ.M2 .OR. LBD(M3).EQ.0) GO TO 75	ROTATE
	WRITE (6,68) M3,M2,LBD(M3)	ROTATE
	STOP 51	ROTATE
75	LBD(M3) = M2	ROTATE
88	IF (LPMI(M2).EQ.0) GO TO 78	BUTLER1
	DO 77 J=3,9,3	ROTATE
	DO 76 I=1,3	ROTATE
	IJ = I+J	ROTATE
76	T1(I) = BAR(IJ,K)	ROTATE
77	CALL MAT31 (DPMI(1,1,M2),T1,BAR(J+1,K))	ROTATE
73	CONTINUE	ROTATE
	K1 = K2+1	ROTATE
79	CONTINUE	ROTATE
	J1 = J2+1	ROTATE
80	CONTINUE	ROTATE
C		ROTATE
C	TRANSFORM DATA IN BD ARRAYS FOR ELLIPSOIDS THAT HAVE BEEN ASSIGNED	ROTATE
C		ROTATE
81	DO 90 J=1,40	ROTATE
	IF (LBD(J).EQ.0) GO TO 90	ROTATE
	KSEG = LBD(J)	ROTATE
	IF (LPMI(KSEG).EQ.0) GO TO 90	ROTATE
	L = 4	HYPER
	IF (BD(1,J).LT.0.0) L = 5	HYPER
	M = 8	HYPER
	DO 82 I=1,3	ROTATE
	T1(I) = BD(L,J)	HYPER
	L = L + 1	HYPER
	DO 82 K = 1,3	HYPER
	T3(K,I) = BD(M,J)	HYPER
82	M = M + 1	HYPER
	CALL MAT31 (DPMI(1,1,KSEG),T1,BD(L-3,J))	HYPER
	IF (BD(1,J).GT.0.0) GO TO 84	HYPER
	CALL MAT33 (DPMI(1,1,KSEG),T3,BD(8,J))	HYPER

GO TO 90
84 CALL DOTT33 (BD(7,J),DPMI(1,1,KSEG),T3)
CALL MAT33 (DPMI(1,1,KSEG),T3,BD(7,J))
CALL DOTT33 (BD(16,J),DPMI(1.1,KSEG),T3)
CALL MAT33 (DPMI(1,1,KSEG),T3,BD(16,J))
90 CONTINUE
99 RETURN
END

HYPER
HYPER
ROTATE
ROTATE
ROTATE
ROTATE
ROTATE

```

SUBROUTINE RSTART(IF,IT)
C
C THE FIVE FUNCTIONS OF SUBROUTINE RSTART ARE:
C 1. READ INPUT & INITIALIZATION RECORD FROM OLD RESTART TAPE.
C 2. WRITE INPUT & INITIALIZATION RECORD ONTO NEW RESTART TAPE.
C 3. READ TIME POINT RECORD FROM OLD RESTART TAPE.
C 4. READ NEW INPUT DATA FROM INPUT STREAM FOR RESTART.
C 5. WRITE TIME POINT RECORD ONTO NEW RESTART TAPE.
C
C IMPLICIT REAL*8(A-H,O-Z)
C
C ALL LABELED COMMON BLOCKS ARE INCLUDED HERE
C TO GIVE A COMPLETE SET FOR REFERENCE
C 1
COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,
* NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG
DIMENSION IC1(51)
EQUIVALENCE (IC1(1),NSEG)
C 2
COMMON/CNTRSF/ PL(24,30),BELT(20,8),TPTS(6,8),BD(24,40)
DIMENSION RC2(1888)
EQUIVALENCE (RC2(1),PL(1,1))
C 3
COMMON/VPOSTN/ ZPLT(3),SPLT(3),AXV(3,6),VATAB(6,501,6),
* VTO(6),VDT(6),TIMEV(6),OMEGV(6),NVTAB(6),INDXV(6)
DIMENSION RC3(18084),IC3(12)
EQUIVALENCE (RC3(1),ZPLT(1)),(IC3(1),NVTAB(1))
C 4
COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),
* SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)
DIMENSION RC4(900)
EQUIVALENCE (RC4(1),D(1,1,1))
C 5
COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60),
* F(3,30),TQ(3,30),WJ(30),A11(3,3,30)
DIMENSION RC5A(1296),RC5B(480)
EQUIVALENCE (RC5A(1),V1(1,1)),(RC5B(1),F(1,1))
C 6
COMMON/ABDATA/ ZDEF(3,5),DBR(3,3,5),DPVCTR(3,5),DEPLOY(3,5),
* AB(3,5),B(9,4,5),ZR(3,4,5),BFB(3,4,5),DRR(9,4,5),
* VBAGG(5),VSCS(5),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),TV2(3,10,5),
* SWITCH(5),PYMOUT(5),SCALE(5),PREVT,IFULL(6)
DIMENSION RC6A(610),RC6B(271)
EQUIVALENCE (RC6A(1),ZDEP(1,1)),(RC6B(1),CYMIN(1))
C 7
COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),
* BLTTTL(5,8),PLTTTL(5,30),BAGTTL(5,6),SEG(30),
* JOINT(30),CGS(30),JS(30)
REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTTL,BAGTTL,SEG,JOINT
LOGICAL*1 CGS,JS

```


	REAL RC7,RC7A,XDTE,XCMENT	RSTART
	DIMENSION RC7(305),RC7A(348),XDTE(3),XCMENT(40)	RSTART
	EQUIVALENCE (RC7(1),VPSTTL(1)),(RC7A(1),DATE(1))	RSTART
C 8	COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),	RSTART
	* UNITL,UNITM,UNITT,GRAVY(3),TWOPI	TWOPI
	DIMENSION RC8(35)	TWOPI
	EQUIVALENCE (RC8(1),PI)	RSTART
C 9	COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60),	SLIP
	* RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),	RSTART
	* JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)	RSTART
	DIMENSION RC9(2460),IC9(150)	SLIP
	EQUIVALENCE (RC9(1),PHI(1,1)),(IC9(1),JNT(1))	RSTART
C 10	COMMON/JBARTZ/ MNPL(30),MNBLT(8),MNSEG(30),MNBAG(6),	RSTART
	* MPL(3,5,30),MBLT(3,5,8),MSEG(3,5,30),MBAG(3,10,6),	RSTART
	* NTPL(5,30),NTBLT(5,8),NTSEG(5,30)	RSTART
	DIMENSION IC10(1614)	RSTART
	EQUIVALENCE (IC10(1),MNPL(1))	RSTART
C 11	COMMON/FORCES/PSF(7,70),BSF(4,20),SSF(10,40),BAGSF(3,20),	NCFORC
	* PRJNT(7,30),NPANEL(5),NPSF,NBSF,NSSF,NBGSF	RSTART
	DIMENSION RC11(1240),IC11(9)	NCFORC
	EQUIVALENCE (RC11(1),PSF(1,1)),(IC11(1),NPANEL(1))	RSTART
C 12	COMMON/INTEST/ SGTEST(3.4,30),XTEST(3,120),SEGT(120),REGT(120)	RSTART
	REAL SEGT	RSTART
	DIMENSION RC12(720)	RSTART
	EQUIVALENCE (RC12(1),SGTEST(1,1,1))	RSTART
C 13	COMMON/GSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),	RSTART
	* HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12),	RSTART
	* RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),	RSTART
	* KQ1(12),KQ2(12),KQTYPE(12)	RSTART
	DIMENSION RC13(72),IC13(36),RC13A(1212),RC13H(348)	RSTART
	EQUIVALENCE (RC13(1),RK1(1,1)),(IC13(1),KQ1(1)),	RSTART
	* (RC13A(1),A13(1,1,1)),(RC13H(1),HHT(1,1,1))	RSTART
C 14	COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(1250),TAB(4500)	DIMENB
	DIMENSION IC14(1304)	BUTLER2
	EQUIVALENCE (IC14(1),MXNTI)	RSTART
C 15	COMMON/COMAIN/VAR(240),DER(240),DT,H0,HMAX,HMIN,RSTIME,	RSTART
	* ISTEP,NSTEPS,NDINT,NEQ,IRSIN,IRSOUT	RSTART
	DIMENSION RC15(485),IC15(6)	RSTART
	EQUIVALENCE (RC15(1),VAR(1)),(IC15(1),ISTEP)	RSTART
C 16	COMMON/CDINT/ UU(4),GH(3,4),	RSTART
	* E(3,240),FF(5,240),GG(5,240),Y(5,240),U(5,240),	RSTART
	* H,HPRINT,HS,TPRINT,TSTART,ICNT,IDBL,IFLAG,IDMMY	80386
C	NOTE: FF REPLACES F FROM SUBROUTINE DINT.	RSTART

```

DIMENSION RC16(5541),IC16(3)                                RSTART
EQUIVALENCE (RC16(1),UU(1)),(IC16(1),ICNT)                RSTART
C 17                                                         RSTART
COMMON/DAMPER/ APSDM(3,20),APSDN(3,20),ASD(5,20),MSDM(20),MSDN(20)RSTART
DIMENSION RC17(220),IC17(40)                                RSTART
EQUIVALENCE (RC17(1),APSDM(1,1)),(IC17(1),MSDM(1))        RSTART
C 18                                                         RSTART
COMMON/CEULER/ IEULER(30),HIR(3,3,90),ANG(3,30),ANGD(3,30),
* FE(3,30),TQE(3,30),CONST(5,30)                            JDRIFT
DIMENSION RC18(1320)                                        JDRIFT
EQUIVALENCE (RC18(1),HIR(1,1,1))                            RSTART
C 19                                                         RSTART
COMMON/TEMPVI/ CREST,TTI(3),RII(3),R2I(3),JSTOP(4,2,30)   RSTART
DIMENSION RC19(10),IC19(180)                               RSTART
EQUIVALENCE (RC19(1),CREST),(IC19(1),JSTOP(1,1,1))        RSTART
C 20                                                         RSTART
COMMON/CYDATA/ CYTD(5),CYPA(5),CYSP(5),CYT0(5),CYV0(5),CYCD(5),
* CYK(5),CYR(5),CYAT(5),CYPV(5),CYCDO(5),CYA0(5),          RSTART
* CYP0(5),CYSS(5),CYLO(5),CYC(5),CYRH00(5),CYVMAX(5),      RSTART
* CYORFC(5),CYRHO(5),CYT(5),CYP(5),CYV(5)                  RSTART
DIMENSION RC20A(95),RC20B(20)                               RSTART
EQUIVALENCE (RC20A(1),CYTD(1)),(RC20B(1),CYRHO(1))        RSTART
C 21                                                         RSTART
COMMON/RSAGE/ XSG(3,20,3),DPMI(3,3,30),LPMI(30),           ATBIII
* NSG(9),MSG(20,9),MCG,MCGIN(24,5),KREF(20,9)              TTHKREF
DIMENSION RC21(450),IC21(520)                              TTHKREF
EQUIVALENCE (RC21(1),XSG(1,1,1)),(IC21(1),LPMI(1))        RSTART
C 22                                                         RSTART
COMMON/FLXBLE/ HF(4,12,8),B42(3,3,24),V4(3,8),NFLEX(3,8)  RSTART
DIMENSION RC22(624),IC22(24)                               RSTART
EQUIVALENCE (RC22(1),HF(1,1,1)),(IC22(1),NFLEX(1,1))     RSTART
C 23                                                         RSTART
COMMON/HRNESS/ BAR(15,100),BB(100),BBDOT(100),PLOSS(2,100),
* XLONG(20),HTIME(2),IBAR(5,100),NL(2,100),                RSTART
* NPTSPB(20),NPTPLY(20),NTHRNS(20),NBLTPH(5)              RSTART
DIMENSION RC23(1922),IC23(765)                             RSTART
EQUIVALENCE (RC23(1),BAR(1,1)),(IC23(1),IBAR(1,1))        RSTART
C 24                                                         RSTART
COMMON/WINDFR/ WTIME(30),QFU(3,5),QFV(3,5),WF(3,30),IWIND(30),
* MWSEG(7,30),NFVSEG(6),NFVNT(5),MOWSEG(30,30)            WINDOP
DIMENSION RC24(150),IC24(1151)                             WINDOP
EQUIVALENCE (RC24(1),WTIME(1)),(IC24(1),IWIND(1))         WINDOP
C                                                         RSTART
REAL AOLD4,AAOLD4                                           RSTART
DIMENSION COMMON(24),INDEX(3)                               RSTART
DATA COMMON /8HCONTRL ,8HCNTRF ,8HVPOSTN ,8HSGMNTS ,      RSTART
* 8HCMATRX ,8HABDATA ,8HTITLES ,8HCNSNTS ,                RSTART
* 8HDESCRP ,8HJBARTZ ,8HFORCES ,8HINTEST ,                RSTART
* 8HCSTRNT ,8HTABLES ,8HCOMAIN ,8HCDINT ,                  RSTART
* 8HDAMPER ,8HCEULER ,8HTEMPVI ,8HCYDATA ,                RSTART
* 8HRSAGE ,8HFLXBLE ,8HHRNESS ,8HWINDFR /                 RSTART
DATA BLANK/8H /                                             RSTART

```


1	IF (ITEM.GT.1)	GO TO 401	RSTART
	IF (ITYPE.NE.1)	GO TO 490	RSTART
	ROLD - TIME		RSTART
	TIME - RR		RSTART
	GO TO 492		RSTART
401	IF (ITEM.GT.52)	GO TO 490	PAGE
	IF (ITYPE.NE.2)	GO TO 490	RSTART
	IOLD - IC1(ITEM-1)		RSTART
	IC1(ITEM-1) - II		RSTART
	GO TO 494		RSTART
C	COMMON /CNTSRF/		RSTART
2	IF (ITEM.GT.1888)	GO TO 490	EDGE
	IF (ITYPE.NE.1)	GO TO 490	RSTART
	ROLD - RC2(ITEM)		RSTART
	RC2(ITEM) - RR		RSTART
	GO TO 492		RSTART
C	COMMON /VPOSTN/		RSTART
3	IF (ITEM.GT.18084)	GO TO 403	VEHICL
	IF (ITYPE.NE.1)	GO TO 490	RSTART
	ROLD - RC3(ITEM)		RSTART
	RC3(ITEM) - RR		RSTART
	GO TO 492		RSTART
403	IF (ITEM.GT.18096)	GO TO 490	VEHICL
	IF (ITYPE.NE.2)	GO TO 490	RSTART
	IOLD - IC3(ITEM-18084)		VEHICL
	IC3(ITEM-18084) - II		VEHICL
	GO TO 494		RSTART
C	COMMON /SGMNTS/		RSTART
4	IF (ITEM.GT.900)	GO TO 404	RSTART
	IF (ITYPE.NE.1)	GO TO 490	RSTART
	ROLD - RC4(ITEM)		RSTART
	RC4(ITEM) - RR		RSTART
	GO TO 492		RSTART
404	IF (ITEM.GT.930)	GO TO 490	RSTART
	IF (ITYPE.NE.2)	GO TO 490	RSTART
	IOLD - NSYM(ITEM-900)		RSTART
	NSYM(ITEM-900) - II		RSTART
	GO TO 494		RSTART
C	COMMON /CMATRX/		RSTART
5	IF (ITEM.GT.1776)	GO TO 490	SLIP
	IF (ITYPE.NE.1)	GO TO 490	RSTART
	ROLD - RC5A(ITEM)		RSTART
	RC5A(ITEM) - RR		RSTART
	GO TO 492		RSTART
C	COMMON /ABDATA/		RSTART
6	IF (ITEM.GT.881)	GO TO 406	RSTART
	IF (ITYPE.NE.1)	GO TO 490	RSTART
	ROLD - RC6A(ITEM)		RSTART
	RC6A(ITEM) - RR		RSTART
	GO TO 492		RSTART
406	IF (ITEM.GT.887)	GO TO 490	RSTART
	IF (ITYPE.NE.2)	GO TO 490	RSTART

	IOLD = IFULL(ITEM-881)	RSTART
	IFULL(ITEM-881) = II	RSTART
	GO TO 494	RSTART
C	COMMON /TITLES/ NOTE: NO PROVISION FOR CGS OR JS.	RSTART
	7 IF (ITEM.GT.348) GO TO 490	RSTART
	IF (ITYPE.NE.3) GO TO 490	RSTART
	AOLD = RC7A(ITEM)	RSTART
	RC7A(ITEM) = AA	RSTART
	GO TO 496	RSTART
C	COMMON /CNSNTS/	RSTART
	8 IF (ITEM.GT.35) GO TO 490	TWOPI
	IF (ITEM.GT.31) GO TO 408	RSTART
	IF (ITEM.LE.28) GO TO 408	RSTART
	IF (ITYPE.NE.3) GO TO 490	RSTART
	AOLD = RC8(ITEM)	RSTART
	RC8(ITEM) = AA	RSTART
	GO TO 496	RSTART
	408 IF (ITYPE.NE.1) GO TO 490	RSTART
	ROLD = RC8(ITEM)	RSTART
	RC8(ITEM) = RR	RSTART
	GO TO 492	RSTART
C	COMMON /DESCRP/	RSTART
	9 IF (ITEM.GT.2460) GO TO 409	SLIP
	IF (ITYPE.NE.1) GO TO 490	RSTART
	ROLD = RC9(ITEM)	RSTART
	RC9(ITEM) = RR	RSTART
	GO TO 492	RSTART
	409 IF (ITEM.GT.2610) GO TO 490	SLIP
	IF (ITYPE.NE.2) GO TO 490	RSTART
	IOLD = IC9(ITEM-2460)	SLIP
	IC9(ITEM-2460) = II	SLIPRT
	GO TO 494	RSTART
C	COMMON /JBARTZ/	RSTART
	10 IF (ITEM.GT.1614) GO TO 490	RSTART
	IF (ITYPE.NE.2) GO TO 490	RSTART
	IOLD = IC10(ITEM)	RSTART
	IC10(ITEM) = II	RSTART
	GO TO 494	RSTART
C	COMMON /FORCES/	RSTART
	11 IF (ITEM.GT.1240) GO TO 411	NCFORC
	IF (ITYPE.NE.1) GO TO 490	RSTART
	ROLD = RC11(ITEM)	RSTART
	RC11(ITEM) = RR	RSTART
	GO TO 492	RSTART
	411 IF (ITEM.GT.1249) GO TO 490	NCFORC
	IF (ITYPE.NE.2) GO TO 490	RSTART
	IOLD = IC11(ITEM-1240)	NCFORC
	IC11(ITEM-1240) = II	NCFORC
	GO TO 494	RSTART
C	COMMON /INTEST/	RSTART
	12 IF (ITEM.GT.720) GO TO 412	RSTART
	IF (ITYPE.NE.1) GO TO 490	RSTART

ROLD = RC12(ITEM)	RSTART
RC12(ITEM) = RR	RSTART
GO TO 492	RSTART
412 IF (ITEM.GT.840) GO TO 512	RSTART
IF (ITYPE.NE.3) GO TO 490	RSTART
AOLD = SEGT(ITEM-720)	RSTART
SEGT(ITEM-720) = AA	RSTART
GO TO 496	RSTART
512 IF (ITEM.GT.960) GO TO 490	RSTART
IF (ITYPE.NE.3) GO TO 490	RSTART
AOLD = REGT(ITEM-840)	RSTART
REGT(ITEM-840) = AA	RSTART
GO TO 496	RSTART
C COMMON /CSTRNT/	RSTART
13 IF (ITEM.GT.1212) GO TO 413	RSTART
IF (ITYPE.NE.1) GO TO 490	RSTART
ROLD = RC13A(ITEM)	RSTART
RC13A(ITEM) = RR	RSTART
GO TO 492	RSTART
413 IF (ITEM.GT.1248) GO TO 490	RSTART
IF (ITYPE.NE.2) GO TO 490	RSTART
IOLD = IC13(ITEM-1212)	RSTART
IC13(ITEM-1212) = II	RSTART
GO TO 494	RSTART
C COMMON /TABLES/	RSTART
14 IF (ITEM.GT.1304) GO TO 414	RSTART
IF (ITYPE.NE.2) GO TO 490	BUTLER2
IOLD = IC14(ITEM)	RSTART
IC14(ITEM) = II	RSTART
GO TO 494	RSTART
414 IF (ITEM.GT.5804) GO TO 490	MISC
IF (ITYPE.NE.1) GO TO 490	RSTART
ROLD = TAB(ITEM-1304)	BUTLER2
TAB(ITEM-1304) = RR	BUTLER2
GO TO 492	RSTART
C COMMON /COMAIN/	RSTART
15 IF (ITEM.GT.485) GO TO 415	RSTART
IF (ITYPE.NE.1) GO TO 490	RSTART
ROLD = RC15(ITEM)	RSTART
RC15(ITEM) = RR	RSTART
GO TO 492	RSTART
415 IF (ITEM.GT.491) GO TO 490	RSTART
IF (ITYPE.NE.2) GO TO 490	RSTART
IOLD = IC15(ITEM-485)	RSTART
IC15(ITEM-485) = II	RSTART
GO TO 494	RSTART
C COMMON /CDINT /	RSTART
16 IF (ITEM.GT.5541) GO TO 416	RSTART
IF (ITYPE.NE.1) GO TO 490	RSTART
ROLD = RC16(ITEM)	RSTART
RC16(ITEM) = RR	RSTART
GO TO 492	RSTART

416	IF (ITEM.GT.5544) GO TO 490	RSTART
	IF (ITYPE.NE.2) GO TO 490	RSTART
	IOLD = IC16(ITEM-5541)	RSTART
	IC16(ITEM-5541) = II	RSTART
	GO TO 494	RSTART
C	COMMON /DAMPER/	RSTART
17	IF (ITEM.GT.220) GO TO 417	RSTART
	IF (ITYPE.NE.1) GO TO 490	RSTART
	ROLD = RC17(ITEM)	RSTART
	RC17(ITEM) = RR	RSTART
	GO TO 492	RSTART
417	IF (ITEM.GT.260) GO TO 490	RSTART
	IF (ITYPE.NE.2) GO TO 490	RSTART
	IOLD = IC17(ITEM-220)	RSTART
	IC17(ITEM-220) = II	RSTART
	GO TO 494	RSTART
C	COMMON /CEULER/	RSTART
18	IF (ITEM.GT.30) GO TO 418	RSTART
	IF (ITYPE.NE.2) GO TO 490	RSTART
	IOLD = IEULER(ITEM)	RSTART
	IEULER(ITEM) = II	RSTART
	GO TO 494	RSTART
418	IF (ITEM.GT.1350) GO TO 490	JDRIFT
	IF (ITYPE.NE.1) GO TO 490	RSTART
	ROLD = RC18(ITEM-30)	RSTART
	RC18(ITEM-30) = RR	RSTART
	GO TO 492	RSTART
C	COMMON /TEMPVI/	RSTART
19	IF (ITEM.GT.10) GO TO 419	RSTART
	IF (ITYPE.NE.1) GO TO 490	RSTART
	ROLD = RC19(ITEM)	RSTART
	RC19(ITEM) = RR	RSTART
	GO TO 492	RSTART
419	IF (ITEM.GT.190) GO TO 490	RSTART
	IF (ITYPE.NE.2) GO TO 490	RSTART
	IOLD = IC19(ITEM-10)	RSTART
	IC19(ITEM-10) = II	RSTART
	GO TO 494	RSTART
C	COMMON /CYDATA/	RSTART
20	IF (ITEM.GT.115) GO TO 490	RSTART
	IF (ITYPE.NE.1) GO TO 490	RSTART
	ROLD = RC20A(ITEM)	RSTART
	RC20A(ITEM) = RR	RSTART
	GO TO 492	RSTART
C	COMMON /RSAVE/	RSTART
21	IF (ITEM.GT.450) GO TO 421	RSTART
	IF (ITYPE.NE.1) GO TO 490	RSTART
	ROLD = RC21(ITEM)	RSTART
	RC21(ITEM) = RR	RSTART
	GO TO 492	RSTART
421	IF (ITEM.GT.970) GO TO 490	TTHKREF
	IF (ITYPE.NE.2) GO TO 490	RSTART

	IOLD = IC21(ITEM-450)	RSTART
	IC21(ITEM-450) = II	RSTART
	GO TO 494	RSTART
C	COMMON /FLXBLE/	RSTART
	22 IF (ITEM.GT.624) GO TO 422	RSTART
	IF (ITYPE.NE.1) GO TO 490	RSTART
	ROLD = RC22(ITEM)	RSTART
	RC22(ITEM) = RR	RSTART
	GO TO 492	RSTART
	422 IF (ITEM.GT.648) GO TO 490	RSTART
	IF (ITYPE.NE.2) GO TO 490	RSTART
	IOLD = IC22(ITEM-624)	RSTART
	IC22(ITEM-624) = II	RSTART
	GO TO 494	RSTART
C	COMMON /HRNESS/	RSTART
	23 IF (ITEM.GT.1922) GO TO 423	RSTART
	IF (ITYPE.NE.1) GO TO 490	RSTART
	ROLD = RC23(ITEM)	RSTART
	RC23(ITEM) = RR	RSTART
	GO TO 492	RSTART
	423 IF (ITEM.GT.2687) GO TO 490	RSTART
	IF (ITYPE.NE.2) GO TO 490	RSTART
	IOLD = IC23(ITEM-1922)	RSTART
	IC23(ITEM-1922) = II	RSTART
	GO TO 494	RSTART
C	COMMON /WINDFR/	RSTART
	24 IF (ITEM.GT.150) GO TO 424	WINDOP
	IF (ITYPE.NE.1) GO TO 490	RSTART
	ROLD = RC24(ITEM)	RSTART
	RC24(ITEM) = RR	RSTART
	GO TO 492	RSTART
	424 IF (ITEM.GT.1301) GO TO 490	WINDOP
	IF (ITYPE.NE.2) GO TO 490	RSTART
	IOLD = IC24(ITEM-150)	WINDOP
	IC24(ITEM-150) = II	WINDOP
	GO TO 494	RSTART
C		RSTART
C	ERROR MESSAGE - TERMINATE PROGRAM.	RSTART
C		RSTART
	490 WRITE (6,491) AVAR,INDEX,NCOM,ITEM,ITYPE,RR,II,AA	RSTART
	491 FORMAT('0 SUBROUTINE RSTART INPUT ERROR'//	RSTART
	* ' AVAR= ',A8,' INDEX= ',3I6,' NCOM= ',I6,' ITEM= ',I6,	RSTART
	* ' ITYPE= ',I6,' RR= ',G15.8,' II= ',I8,' AA= ',A8//	RSTART
	* ' PROGRAM IS BEING TERMINATED.'	RSTART
	STOP 2	RSTART
C		RSTART
C	PRINT MESSAGE FOR REAL VARIABLES.	RSTART
C		RSTART
	492 WRITE (6,493) AVAR,INDEX,COMMON(NCOM),ROLD,RR	RSTART
	493 FORMAT('0',A6,'(',I4,',',I4,',',I4,') OF COMMON/',A6,'/',	RSTART
	* ' HAS BEEN CHANGED FROM ',G15.8,' TO ',G15.8)	RSTART
	IF (RROLD.EQ.0.0) GO TO 400	RSTART

IF (DABS(RROLD-ROLD).LE.0.00001*RROLD) GO TO 400	RSTART
WRITE (6,383) RROLD	RSTART
383 FORMAT(' INPUT VALUE FOR RROLD WAS ',G15.8//)	RSTART
GO TO 490	RSTART
C	RSTART
C PRINT MESSAGE FOR INTEGER VARIABLES.	RSTART
C	RSTART
494 WRITE (6,495) AVAR,INDEX,COMMON(NCOM),IOLD,II	RSTART
495 FORMAT('0',A6,'(',I4,',',I4,',',I4,') OF COMMON',A6,'/',	RSTART
* ' HAS BEEN CHANGED FROM ', I8,' TO ', I8)	RSTART
IF (IIOLD.EQ.0) GO TO 400	RSTART
IF (IOLD.EQ.IIOLD) GO TO 400	RSTART
WRITE (6,385) IIOLD	RSTART
385 FORMAT(' INPUT VALUE FOR IIOLD WAS ',I8//)	RSTART
GO TO 490	RSTART
C	RSTART
C PRINT MESSAGE FOR ALPHANUMERIC VARIABLES.	RSTART
C	RSTART
496 WRITE (6,497) AVAR,INDEX,COMMON(NCOM),AOLD,AA	RSTART
497 FORMAT('0',A6,'(',I4,',',I4,',',I4,') OF COMMON',A6,'/',	RSTART
* ' HAS BEEN CHANGED FROM ', A8,' TO ', A8)	RSTART
IF (AAOLD.EQ.BLANK) GO TO 400	RSTART
AAOLD4 = AAOLD	RSTART
AOLD4 = AOLD	RSTART
IF (AOLD4.EQ.AAOLD4) GO TO 400	RSTART
WRITE (6,387) AAOLD	RSTART
387 FORMAT(' INPUT VALUE FOR AAOLD WAS ',A8//)	RSTART
GO TO 490	RSTART
999 CALL ELTIME(2,25)	RSTART
RETURN	RSTART
END	RSTART

	SUBROUTINE SEARCH(AVAR, INDEX, NCOM, ITEM)		SEARCH
C		REV IV 07/24/86	SLIP
C	CALL BY SUBROUTINE RSTART TO COMPUTE NCOM & ITEM FROM AVAR &		SEARCH
C	INDEX. RETURNS NCOM=0 FOR ERROR AND NCOM=50 FOR BLANK.		SEARCH
C			SEARCH
	IMPLICIT REAL*8(A-H,O-Z)		SEARCH
	DIMENSION BVAR(264), KOUNT(25), NDIM(3, 264), NJ(3), NK(3), INDEX(3)		SLIP
	DIMENSION C1 (17) , NC1 (51)		PAGE
	DIMENSION C2 (4) , NC2 (12)		SEARCH
	DIMENSION C3 (10) , NC3 (30)		SEARCH
	DIMENSION C4 (9) , NC4 (27)		SEARCH
	DIMENSION C5 (9) , NC5 (27)		SLIP
	DIMENSION C6 (30) , NC6 (90)		SEARCH
	DIMENSION C7 (11) , NC7 (33)		SEARCH
	DIMENSION C8 (10) , NC8 (30)		TWOPI
	DIMENSION C9 (15) , NC9 (45)		SEARCH
	DIMENSION C10(11) , NC10(33)		SEARCH
	DIMENSION C11(10) , NC11(30)		SEARCH
	DIMENSION C12(4) , NC12(12)		SEARCH
	DIMENSION C13(16) , NC13(48)		SEARCH
	DIMENSION C14(7) , NC14(21)		SEARCH
	DIMENSION C15(13) , NC15(39)		SEARCH
	DIMENSION C16(15) , NC16(45)		SEARCH
	DIMENSION C17(5) , NC17(15)		SEARCH
	DIMENSION C18(7) , NC18(21)		SEARCH
	DIMENSION C19(5) , NC19(15)		SEARCH
	DIMENSION C20(23) , NC20(69)		SEARCH
	DIMENSION C21(8) , NC21(24)		CHGII
	DIMENSION C22(4) , NC22(12)		SEARCH
	DIMENSION C23(12) , NC23(36)		SEARCH
	DIMENSION C24(9) , NC24(27)		WINDOP
C			SEARCH
	EQUIVALENCE (C1 (1), BVAR(1)) , (NC1 (1), NDIM(1, 1))		SEARCH
	EQUIVALENCE (C2 (1), BVAR(18)) , (NC2 (1), NDIM(1, 18))		PAGE
	EQUIVALENCE (C3 (1), BVAR(22)) , (NC3 (1), NDIM(1, 22))		PAGE
	EQUIVALENCE (C4 (1), BVAR(32)) , (NC4 (1), NDIM(1, 32))		PAGE
	EQUIVALENCE (C5 (1), BVAR(41)) , (NC5 (1), NDIM(1, 41))		PAGE
	EQUIVALENCE (C6 (1), BVAR(50)) , (NC6 (1), NDIM(1, 50))		SLIP
	EQUIVALENCE (C7 (1), BVAR(80)) , (NC7 (1), NDIM(1, 80))		SLIP
	EQUIVALENCE (C8 (1), BVAR(91)) , (NC8 (1), NDIM(1, 91))		SLIP
	EQUIVALENCE (C9 (1), BVAR(101)) , (NC9 (1), NDIM(1, 101))		SLIP
	EQUIVALENCE (C10(1), BVAR(116)) , (NC10(1), NDIM(1, 116))		SLIP
	EQUIVALENCE (C11(1), BVAR(127)) , (NC11(1), NDIM(1, 127))		SLIP
	EQUIVALENCE (C12(1), BVAR(137)) , (NC12(1), NDIM(1, 137))		SLIP
	EQUIVALENCE (C13(1), BVAR(141)) , (NC13(1), NDIM(1, 141))		SLIP
	EQUIVALENCE (C14(1), BVAR(157)) , (NC14(1), NDIM(1, 157))		SLIP
	EQUIVALENCE (C15(1), BVAR(164)) , (NC15(1), NDIM(1, 164))		SLIP
	EQUIVALENCE (C16(1), BVAR(177)) , (NC16(1), NDIM(1, 177))		SLIP
	EQUIVALENCE (C17(1), BVAR(192)) , (NC17(1), NDIM(1, 192))		SLIP
	EQUIVALENCE (C18(1), BVAR(197)) , (NC18(1), NDIM(1, 197))		SLIP
	EQUIVALENCE (C19(1), BVAR(204)) , (NC19(1), NDIM(1, 204))		SLIP
	EQUIVALENCE (C20(1), BVAR(209)) , (NC20(1), NDIM(1, 209))		SLIP

	EQUIVALENCE (C21(1),BVAR(232)) , (NC21(1),NDIM(1,232))	SLIP
	EQUIVALENCE (C22(1),BVAR(240)) , (NC22(1),NDIM(1,240))	SLIP
	EQUIVALENCE (C23(1),BVAR(244)) , (NC23(1),NDIM(1,244))	SLIP
	EQUIVALENCE (C24(1),BVAR(256)) , (NC24(1),NDIM(1,256))	SLIP
C		SEARCH
	DATA NVAR/264/ , KOM/24/ , BLANK/8H /	SLIP
	DATA KOUNT/1,18,22,32,41,50,80,91,101,116,127,137,141,157,	SLIP
	* 164,177,192,197,204,209,232,240,244,256,265/	SLIP
C		SEARCH
C	1 COMMON/CONTRL/	SEARCH
C		SEARCH
	DATA C1 / 8HTIME ,8HNSEG ,8HNJNT ,8HNPL ,8HNBLT ,	SEARCH
*	8HNBAG ,8HNVEH ,8HNGRND ,8HNS ,8HNQ ,	SEARCH
*	8HNSD ,8HNFLX ,8HNNRNS ,8HNWINDF ,8HNJNTF ,	SEARCH
*	8HNPRT ,8HNPG /	PAGE
	DATA NC1 / 0,0,0 , 0,0,0 , 0,0,0 , 0,0,0 , 0,0,0 ,	SEARCH
*	0,0,0 , 0,0,0 , 0,0,0 , 0,0,0 , 0,0,0 ,	SEARCH
*	0,0,0 , 0,0,0 , 0,0,0 , 0,0,0 , 0,0,0 ,	SEARCH
*	36,0,0 , 0,0,0 /	PAGE
C		SEARCH
C	2 COMMON/CNTRSF/	SEARCH
C		SEARCH
	DATA C2 / 8HPL ,8HBELT ,8HTPTS ,8HBD /	SEARCH
	DATA NC2 / 24,30,0 , 20,8,0 , 6,8,0 , 24,40,0 /	EDGE
C		SEARCH
C	3 COMMON/VPOSTN/	SEARCH
C		SEARCH
	DATA C3 / 8HZPLT ,8HSPLT ,8HAXV ,8HVATAB ,8HVTO ,	SEARCH
*	8HVDT ,8HTIMEV ,8HOMEGV ,8HNVTAB ,8HINDXV /	SEARCH
	DATA NC3 / 3,0,0 , 3,0,0 , 3,6,0 , 6,501,6 , 6,0,0 ,	VEHICL
*	6,0,0 , 6,0,0 , 6,0,0 , 6,0,0 , 6,0,0 /	SEARCH
C		SEARCH
C	4 COMMON/SGMNTS/	SEARCH
C		SEARCH
	DATA C4 / 8HD ,8HWMEG ,8HWMEGD ,8HU1 ,8HU2 ,	SEARCH
*	8HSEGLP ,8HSEGLV ,8HSEGLA ,8HNSYM /	SEARCH
	DATA NC4 / 3,3,30 , 3,30,0 , 3,30,0 , 3,30,0 , 3,30,0 ,	SEARCH
*	3,30,0 , 3,30,0 , 3,30,0 , 30,0,0 /	SEARCH
C		SEARCH
C	5 COMMON/CMATRX/	SEARCH
C		SEARCH
	DATA C5 / 8HV1 ,8HV2 ,8HV3 ,8HB12 ,8HA22 ,	SEARCH
*	8HF ,8HTQ ,8HWJ ,8HA11 /	SLIP
	DATA NC5 / 3,30,0 , 3,30,0 , 3,12,0 , 3,3,60 , 3,3,60 ,	SEARCH
*	3,30,0 , 3,30,0 , 30,0,0 , 3,3,60 /	SLIP
C		SEARCH
C	6 COMMON/ABDATA/	SEARCH
C		SEARCH
	DATA C6 / 8HZDEP ,8HDBR ,8HDPVCTR ,8HDEPLOY ,8HAB ,	SEARCH
*	8HB ,8HZR ,8HBFB ,8HDRR ,8HVBAGG ,	SEARCH
*	8HVSCS ,8HSPRK ,8HCK ,8HCMASS ,8HCYMIN ,	SEARCH
*	8HCYMOUT ,8HBAGPV ,8HPD ,8HVBAG ,8HVOLBP ,	SEARCH

	*	8HPCYV	,8HPCYMIN	,8HPVBAG	,8HTV1	,8HTV2	, SEARCH
	*	8H SWITCH	,8HPYMOUT	,8HSCALE	,8HPREVT	,8HIFULL	/ SEARCH
		DATA NC6 /	3,5,0	, 3,3,5	, 3,5,0	, 3,5,0	, SEARCH
	*		9,4,5	, 3,4,5	, 3,4,5	, 9,4,5	, 5,0,0
	*		5,0,0	, 5,0,0	, 5,0,0	, 5,0,0	, SEARCH
	*		5,0,0	, 5,0,0	, 5,0,0	, 5,0,0	, SEARCH
	*		5,0,0	, 5,0,0	, 5,0,0	, 5,0,0	, SEARCH
	*		5,0,0	, 5,0,0	, 5,0,0	, 3,4,5	, 3,10,5
	*		5,0,0	, 5,0,0	, 5,0,0	, 0,0,0	, 6,0,0
C							/ SEARCH
C	7	COMMON/TITLES/					SEARCH
C							SEARCH
		DATA C7 /	8HDATE	,8HCOMMENT	,8HVPSTTL	,8HBDYTTL	,8HBLTTTL
	*		8HPLTTL	,8HBAGTTL	,8HSEG	,8HJOINT	,8HCGS
	*		8HJS	/			SEARCH
		DATA NC7 /	3,0,0	, 40,0,0	, 20,0,0	, 5,0,0	, 5,8,0
	*		5,30,0	, 5,6,0	, 30,0,0	, 30,0,0	, 30,0,0
	*		30,0,0	/			SEARCH
C							SEARCH
C	8	COMMON/CNSNTS/					SEARCH
C							SEARCH
		DATA C8 /	8HPI	,8HRADIAN	,8HG	,8HTHIRD	,8HEPS
	*		8HUNITL	,8HUNITM	,8HUNITT	,8HGRAVY	,8HTWOPI
		DATA NC8 /	0,0,0	, 0,0,0	, 0,0,0	, 0,0,0	, 24,0,0
	*		0,0,0	, 0,0,0	, 0,0,0	, 3,0,0	, 0,0,0
							/ TWOPI
C							SEARCH
C	9	COMMON/DESCRP/					SEARCH
C							SEARCH
		DATA C9 /	8HPI	,8HW	,8HRW	,8HSR	,8HHA
	*		8HHB	,8HRPHI	,8HHT	,8HSPRING	,8HVISC
	*		8HJNT	,8HJPIN	,8HJSING	,8HJGLOB	,8HJOINTF
		DATA NC9 /	3,30,0	, 30,0,0	, 30,0,0	, 4,50,0	, 3,60,0
	*		3,60,0	, 3,30,0	, 3,3,60	, 5,90,0	, 7,90,0
	*		30,0,0	, 30,0,0	, 30,0,0	, 30,0,0	, 30,0,0
							/ SLIP
							SEARCH
C							SEARCH
C	10	COMMON/JBARTZ/					SEARCH
C							SEARCH
		DATA C10/	8HMNPL	,8HMNBLT	,8HMNSEG	,8HMNBAG	,8HMPL
	*		8HMBLT	,8HMSEG	,8HMBAG	,8HNTPL	,8HNTBLT
	*		8HNTSEG	/			SEARCH
		DATA NC10/	30,0,0	, 8,0,0	, 30,0,0	, 6,0,0	, 3,5,30
	*		3,5,8	, 3,5,30	, 3,10,6	, 5,30,0	, 5,8,0
	*		5,30,0	/			SEARCH
C							SEARCH
C	11	COMMON/FORCES/					SEARCH
C							SEARCH
		DATA C11/	8HPSF	,8HBSF	,8HSSF	,8HBAGSF	,8HPRJNT
	*		8HNPANEL	,8HNPSF	,8HNBSF	,8HNSSF	,8HNBGSF
		DATA NC11/	7,70,0	, 4,20,0	, 10,40,0	, 3,20,0	, 7,30,0
	*		5,0,0	, 0,0,0	, 0,0,0	, 0,0,0	, 0,0,0
							/ NCFORC
							SEARCH
C							SEARCH
C	12	COMMON/INTEST/					SEARCH
C							SEARCH

	DATA C12/ 8HSGTEST	,8HXTEST	,8HSEGT	,8HRECT	/	SEARCH
	DATA NC12/ 3,4,30	, 3,120,0	, 120,0,0	, 120,0,0	/	SEARCH
C						SEARCH
C	C 13 COMMON/CSTRNT/					SEARCH
C						SEARCH
	DATA C13/ 8HA13	,8HA23	,8HB31	,8HB32	,8HHHT	, SEARCH
*	8HRK1	,8HRK2	,8HQQ	,8HTQQ	,8HRQQ	, SEARCH
*	8HHQQ	,8HSQQ	,8HCFQQ	,8HKQ1	,8HKQ2	, SEARCH
*	8HKQTYPE	/				SEARCH
	DATA NC13/ 3,3,24	, 3,3,24	, 3,3,24	, 3,3,24	, 3,3,12	, SEARCH
*	3,12,0	, 3,12,0	, 3,12,0	, 3,12,0	, 3,12,0	, SEARCH
*	3,12,0	, 12,0,0	, 12,0,0	, 12,0,0	, 12,0,0	, SEARCH
*	12,0,0	/				SEARCH
C						SEARCH
C	C 14 COMMON/TABLES/					SEARCH
C						SEARCH
	DATA C14/ 8HMXNTI	,8HMXNTB	,8HMXTB1	,8HMXTB2	,8HNTI	, SEARCH
*	8HNTAB	,8HTAB	/			SEARCH
	DATA NC14/ 0,0,0	, 0,0,0	, 0,0,0	, 0,0,0	, 50,0,0	, SEARCH
*	1250,0,0	, 4500,0,0/				BUTLER2
C						SEARCH
C	C 15 COMMON/COMAIN/					SEARCH
C						SEARCH
	DATA C15/ 8HVAR	,8HDER	,8HDT	,8HHO	,8HHMAX	, SEARCH
*	8HHMIN	,8HRSTIME	,8HISTEP	,8HNSTEPS	,8HNDINT	, SEARCH
*	8HNEQ	,8HIRSIN	,8HIRSOUT	/		SEARCH
	DATA NC15/ 240,0,0	, 240,0,0	, 0,0,0	, 0,0,0	, 0,0,0	, SEARCH
*	0,0,0	, 0,0,0	, 0,0,0	, 0,0,0	, 0,0,0	, SEARCH
*	0,0,0	, 0,0,0	, 0,0,0	/		SEARCH
C						SEARCH
C	C 16 COMMON/CDINT /					SEARCH
C						SEARCH
	DATA C16/ 8HUU	,8HGH	,8HE	,8HFF	,8HGG	, SEARCH
*	8HY	,8HU	,8HH	,8HHPRINT	,8HHS	, SEARCH
*	8HTPRINT	,8HTSTART	,8HICNT	,8HIDBL	,8HIFLAG	/ SEARCH
	DATA NC16/ 4,0,0	, 3,4,0	, 3,240,0	, 5,240,0	, 5,240,0	, SEARCH
*	5,240,0	, 5,240,0	, 0,0,0	, 0,0,0	, 0,0,0	, SEARCH
*	0,0,0	, 0,0,0	, 0,0,0	, 0,0,0	, 0,0,0	/ SEARCH
C						SEARCH
C	C 17 COMMON/DAMPER/					SEARCH
C						SEARCH
	DATA C17/ 8HAPSDM	,8HAPSDN	,8HASD	,8HMSDM	,8HMSDN	/ SEARCH
	DATA NC17/ 3,20,0	, 3,20,0	, 5,20,0	, 20,0,0	, 20,0,0	/ SEARCH
C						SEARCH
C	C 18 COMMON/CEULER/					SEARCH
C						SEARCH
	DATA C18/ 8HIEULER	,8HHIR	,8HANG	,8HANGD	,8HFE	, SEARCH
*	8HTQE	,8HCONST	/			SEARCH
	DATA NC18/ 30,0,0	, 3,3,90	, 3,30,0	, 3,30,0	, 3,30,0	, JDRIIFT
*	3,30,0	, 5,30,0	/			JDRIIFT
C						SEARCH
C	C 19 COMMON/TEMPVI/					SEARCH

C	DATA C19/ 8HCREST	,8HTTI	,8HR1I	,8HR2I	,8HJSTOP	SEARCH
	DATA NC19/ 0,0,0	, 3,0,0	, 3,0,0	, 3,0,0	, 4,2,30	/ SEARCH
C	COMMON/CYDATA/					SEARCH
C	DATA C20/ 8HCYTD	,8HCYPA	,8HCYSP	,8HCYTO	,8HCYVO	SEARCH
*	8HCYCD	,8HCYK	,8HCYR	,8HCYAT	,8HCYPV	, SEARCH
*	8HCYCD0	,8HCYAO	,8HCYPO	,8HCYSS	,8HCYLO	, SEARCH
*	8HCYC	,8HCYRHO0	,8HCYVMAX	,8HCYORFC	,8HCYRHO	, SEARCH
*	8HCYT	,8HCYRHP	,8HCYV	/		SEARCH
	DATA NC20/ 5,0,0	, 5,0,0	, 5,0,0	, 5,0,0	, 5,0,0	, SEARCH
*	5,0,0	, 5,0,0	, 5,0,0	, 5,0,0	, 5,0,0	, SEARCH
*	5,0,0	, 5,0,0	, 5,0,0	, 5,0,0	, 5,0,0	, SEARCH
*	5,0,0	, 5,0,0	, 5,0,0	, 5,0,0	, 5,0,0	, SEARCH
±	5,0,0	, 5,0,0	, 5,0,0	/		SEARCH
C	COMMON/RSAVE/					SEARCH
C	DATA C21/ 8HXSG	,8HDPMI	,8HLPMI	,8HNSG	,8HMSG	SEARCH
*	8HMCG	,8HMCGIN	,8HKREF	/		CHGIII
	DATA NC21/ 3,20,3	, 3,3,30	, 30,0,0	, 9,0,0	, 20,9,0	, WINDOP
*	1,0,0	, 24,5,0	, 20,9,0	/		TTHKREF
C	COMMON/FLXBLE/					SEARCH
C	DATA C22/ 8HHF	,8HB42	,8HV4	,8HNFLEX	/	SEARCH
	DATA NC22/ 4,12,8	, 3,3,24	, 3,8,0	, 3,8,0	/	SEARCH
C	COMMON/HRNESS/					SEARCH
C	DATA C23/ 8HBBAR	,8HBB	,8HBB0GT	,8HPLOSS	,8HXLONG	SEARCH
*	8HHTIME	,8HIBAR	,8HNL	,8HNPTSPB	,8HNPTPLY	, SEARCH
*	8HNTHRNS	,8HNBLTPH	/			SEARCH
	DATA NC23/ 15,100,0	, 100,0,0	, 100,0,0	, 2,100,0	, 20,0,0	, SEARCH
*	2,0,0	, 5,100,0	, 2,100,0	, 20,0,0	, 20,0,0	, SEARCH
*	20,0,0	, 5,0,0	/			SEARCH
C	COMMON/WINDFR/					SEARCH
C	DATA C24/ 8HWTIME	,8HQFU	,8HQFV	,8HWF	,8HIWIND	SEARCH
*	8HMWSEG	,8HNFVSEG	,8HNFVNT	,8HMOWSEG	/	WINDOP
	DATA NC24/ 30,0,0	, 3, 5,0	, 3, 5,0	, 3,30,0	, 30,0,0	, WINDOP
*	7,30,0	, 6,0,0	, 5,0,0	, 30,30,0	/	WINDOP
	NCOM = 50					SEARCH
	IF (AVAR.EQ.BLANK) GO TO 99					SEARCH
C	SEARCH FOR VARIABLE NO. IV.					SEARCH
C						SEARCH
C	NCOM = 0					SEARCH
	DO 10 IV=1,NVAR					SEARCH
	IF (AVAR.EQ.BVAR(IV)) GO TO 12					SEARCH
	10 CONTINUE					SEARCH

	GO TO 99	SEARCH
C		SEARCH
C	SEARCH FOR COMMON NO. IC.	SEARCH
C		SEARCH
	12 DO 20 IC=1,KOM	SEARCH
	IF (IV.GE.KOUNT(IC).AND.IV.LT.KOUNT(IC+1)) GO TO 22	SEARCH
	20 CONTINUE	SEARCH
	GO TO 99	SEARCH
C		SEARCH
C	COMPUTE ITEM NO. FOR VARIABLE IV IN COMMON IC.	SEARCH
C		SEARCH
	22 K1 = KOUNT(IC)	SEARCH
	K2 = IV-1	SEARCH
	ITEM = 1	SEARCH
	IF (K1.EQ.IV) GO TO 25	SEARCH
	DO 24 K=K1,K2	SEARCH
	NI = 1	SEARCH
	DO 23 I=1,3	SEARCH
	IF (NDIM(I,K).NE.0) NI=NI*NDIM(I,K)	SEARCH
	23 CONTINUE	SEARCH
	24 ITEM = ITEM+NI	SEARCH
	25 DO 26 I=1,3	SEARCH
	IF (INDEX(I).EQ.0 .AND. NDIM(I,IV).NE.0) GO TO 99	SEARCH
	IF (NDIM(I,IV).EQ.0 .AND. INDEX(I).GT.1) GO TO 99	SEARCH
	NJ(I) = MAXO(INDEX(I)-1,0)	SEARCH
	NK(I) = MAXO(NDIM(I,IV),1)	SEARCH
	IF (NJ(I).GE.NK(I)) GO TO 99	SEARCH
	26 CONTINUE	SEARCH
	ITEM = ITEM+NJ(1)+NJ(2)*NK(1)+NJ(3)*NK(2)*NK(1)	SEARCH
	NCOM = IC	SEARCH
	99 RETURN	SEARCH
	END	SEARCH

	SUBROUTINE SEGSEG(M,MM,N,NS,NT)	REV IV 02/07/87HYPER	SEGSEG
C	IMPLICIT REAL*8(A-H,O-Z)		SEGSEG
	COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(1250),TAB(4500)		SEGSEG
	COMMON/CNTRSF/ PL(24,30),BELT(20,8),TPTS(6,8),BD(24,40)		EDGE
	COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),		SEGSEG
*	SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)		SEGSEG
	COMMON/FORCES/PSF(7,70),BSF(4,20),SSF(10,40),BAGSF(3,20),		NCFORC
*	PRJNT(7,30),NPANEL(5),NPSF,NBSF,NSSF,NBGSF		SEGSEG
	COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),		SEGSEG
*	HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12),		SEGSEG
*	RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),		SEGSEG
*	KQ1(12),KQ2(12),KQTYPE(12)		SEGSEG
	COMMON/RSAVE/ XSG(3,20,3),DPMI(3,3,30),LPMI(30),		TGMOD7
*	NSG(9),MSG(20,9),MCG,MCGIN(24,5),KREF(20,9)		TGMOD7
	COMMON/TEMPVS/DMNT(3,3),TEMP(3,3),B(3,3),XMN(3),RLN(3),XMM(3),		SEGSEG
*	TM(3),R(3),RM(3),DMNWN(3),RLM(3),RN(3),VMN(3),VR(3),		SEGSEG
*	WNM(3),WCM(3),WCN(3),VREL(3),FFM(3),FR(3),TQM(3),		SEGSEG
*	TQN(3),TQNT(3),T(3),H(3),T1(3),T2(3),RMD(3),RND(3),		SEGSEG
*	TD(3),TT4(3,4),TT5(3,4),T3(3),T4(3),P,AMR,FM,CF,		SEGSEG
*	VRM,VRT,VRTS,VRTEST,TF,ELOSS,MCF,NCF,T5(3),T6(3)		TGMOD7
*	,DMMY(34958)		80386
	CALL ELTIME(1,23)		SEGSEG
C			EDGE
C	COMPUTATIONS ARE DONE IN M'S REFERENCE SYSTEM		EDGE
	NN = IABS(NS)		SEGSEG
	CALL DOTT33(D(1,1,M),D(1,1,N),DMNT)		SEGSEG
	DO 10 I = 1,3		SEGSEG
10	XMN(I) = SEGLP(I,M) - SEGLP(I,N)		SEGSEG
	CALL MAT31(D(1,1,M),XMN,XMM)		SEGSEG
	J = 3		HYPER
	IF(BD(1,NN).LT.0.0)J = 4		HYPER
	CALL MAT31(DMNT,BD(J+1,NN),RLN)		HYPER
	J = 3		HYPER
	IF(BD(1,MM).LT.0.0)J = 4		HYPER
	DO 15 I = 1,3		HYPER
	J = J + 1		EDGE
15	R(I) = RLN(I) - XMM(I) - BD(J,MM)		HYPER
	LT = NTAB(NT)		HYPER
	TB = 1.0		SEGSEG
	IF((BD(1,MM).GT.0.0).AND.(BD(1,NN).GT.0.0))GO TO 20		EDGE
C NEW	HYPERELLIPSOID - AT LEAST ONE SURFACE IS A HYPERELLIPSOID		HYPER
	IF (BD(1,MM).LT.0.0.AND.BD(23,MM).NE.0.0) STOP 23		HYPER
	IF (BD(1,NN).LT.0.0.AND.BD(23,NN).NE.0.0) STOP 23		HYPER
C A	HYPERELLIPSOID MUST HAVE IDENTICAL POWERS.		HYPER
C	IF(NS.LT.0) STOP - INTERIOR INTERSECTION NOT OPERATIONAL		HYPER
	IF(NS.LT.0) STOP 38		HYPER
	IF(TAB(LT+23).LE.1.0) CALL HYEST(BD(1,MM),BD(1,NN),TAB(LT+22))		HYPER
	IF(TAB(LT+23).GT.1.0) CALL HYNTR(BD(1,MM),BD(1,NN),TAB(LT+22))		HYPER
	BET = TAB(LT+23)		HYPER
	IF(BET.GT.1.0)TB = 1.0/BET		HYPER
	GO TO 25		HYPER

C	OLD ELLIPSOIDS	HYPER
	20 IF(NS.LT.0.0)TB = -TB	HYPER
	CALL DOT33(BD(7,NN),DMNT,TEMP)	EDGE
	CALL MAT33(DMNT,TEMP,B)	EDGE
	CALL INTERS(BD(7,MM),B,R,TB,RM,TAB(LT+22),TM)	SEGSEG
C		EDGE
	A B R Z C AZ	EDGE
C	INTERS SOLVES (CA + B)Z - BR, TB - SQRT(Z.AZ)	EDGE
C		EDGE
	25 MCF = NTAB(NT+1)	HYPER
	NCF = -MCF	SEGSEG
	IF(NCF.GT.0)CFQQ(NCF) = -999.	SEGSEG
C		EDGE
C	CHECK FOR INTERSECTION	EDGE
C		EDGE
	IF(TB.GE.1.0)GO TO 75	HYPER
	S1 = 0.0	SEGSEG
	S2 = 0.0	SEGSEG
	DO 30 I = 1,3	HYPER
	RI = R(I)	SEGSEG
	IF(NS.LT.0)RI = RM(I) + TB*(RM(I) - R(I))	SEGSEG
	S1 = S1 + RI**2	SEGSEG
30	S2 = S2 + TM(I)**2	HYPER
	AMR = DSQRT(S2)	SEGSEG
	P = (1.0/TB - 1.0)*DSQRT(S1)	SEGSEG
	J = 3	HYPER
	IF(BD(1,MM).LT.0.0)J = 4	HYPER
	DO 35 I = 1,3	HYPER
	J = J + 1	HYPER
	IF((BD(1,MM).LT.0.0).OR.(BD(1,NN).LT.0.0))RM(I) = TB*RM(I)	HYPER
	TM(I) = -TM(I)/AMR	SEGSEG
	T2(I) = RM(I) - R(I)	SEGSEG
	RN(I) = T2(I) + RLN(I)	SEGSEG
35	RLM(I) = RM(I) + BD(J,MM)	HYPER
	CALL DOT31(DMNT,RN,RLN)	SEGSEG
	CALL PLSEGF(M,N,NT)	SEGSEG
C		EDGE
C	STORE PRINT DATA	EDGE
C		EDGE
	SSF(1,NSSF) = P	SEGSEG
	DO 40 I = 1,3	HYPER
	SSF(I+4,NSSF) = RLM(I)	EDGE
40	SSF(I+7,NSSF) = RLN(I)	HYPER
	IF(LPMI(M).NE.0) CALL DOT31(DPMI(1,1,M),RLM,SSF(5,NSSF))	EDGE
	IF(LPMI(N).NE.0) CALL DOT31(DPMI(1,1,N),RLN,SSF(8,NSSF))	EDGE
	IF(MCF.LT.0)GO TO 45	HYPER
	SSF(2,NSSF) = FM	SEGSEG
	TF2FM2 = TF**2 - FM**2	SEGSEG
	IF(TF2FM2.LT.0.0)TF2FM2 = 0.0	SEGSEG
	SSF(3,NSSF) = DSQRT(TF2FM2)	SEGSEG
	SSF(4,NSSF) = TF	SEGSEG
	GO TO 75	HYPER
C		EDGE

C	ROLL-SLIDE	
45	DO 50 I = 1,3	EDGE
50	SSF(I+1,NSSF) = T(I)	HYPER
	IF((BD(1,MM).LT.0.0).OR.(BD(1,NN).LT.0.0)) STOP 29	HYPER
	ANR = XDY(TM,B,T2)	HYPER
	CALL CROSS(TM,WNM,T2)	SEGSEG
	CALL MAT31(B,VR,T1)	SEGSEG
	TB = TM(1)*T1(1) + TM(2)*T1(2) + TM(3)*T1(3)	SEGSEG
	DO 60 I = 1,3	EDGE
	DO 55 J = 1,3	HYPER
	K = I + 3*(J+1)	HYPER
	TT4(I,J) = BD(K,MM)/AMR + B(I,J)/ANR	SEGSEG
55	TT5(I,J) = TT4(I,J)	SEGSEG
	TT4(I,4) = T2(I) - (T1(I) - TB*TM(I))/ANR	HYPER
60	TT5(I,4) = TM(I)	EDGE
	CALL DSMSOL(TT4,3,3)	HYPER
	CALL DSMSOL(TT5,3,3)	SEGSEG
	S1 = TM(1)*TT4(1,4) + TM(2)*TT4(2,4) + TM(3)*TT4(3,4)	SEGSEG
	S2 = (TM(1)*TT5(1,4) + TM(2)*TT5(2,4) + TM(3)*TT5(3,4))/S1	EDGE
	DO 65 I = 1,3	EDGE
	RMD(I) = TT4(I,4) - S2*TT5(I,4)	HYPER
65	RND(I) = RND(I) + VR(I)	EDGE
	CALL CROSS(DMNWN,RND,T1)	HYPER
	CALL CROSS(WMEG(1,MM),RMD,T2)	EDGE
	CALL MAT31(B,RND,T3)	EDGE
	CALL CROSS(DMNWN,TM,T4)	EDGE
	S1 = TM(1)*T3(1) + TM(2)*T3(2) + TM(3)*T3(3)	EDGE
	SQQ(NCF) = 0.0	EDGE
	DO 70 I = 1,3	SEGSEG
	T1(I) = T1(I) - T2(I)	HYPER
70	SQQ(NCF)=SQQ(NCF)+TM(I)*T1(I)-VR(I)*(T4(I)+(T3(I)-S1*TM(I))/ANR)	EDGE
	CALL DOT31(D(1,1,M),T1,RQQ(1,NCF))	HYPER
75	CALL ELTIME(2,23)	EDGE
	RETURN	HYPER
	END	SEGSEG
		SEGSEG

```

SUBROUTINE SETUP1                                SETUP1
C                                                    REV IV   07/24/86SLIP
C FOR KK=1 (BEFORE CONTACT ROUTINE IN DAUX)        SETUP1
C SET UP INITIAL VALUES OF A2 AND B2 ARRAYS FOR THIS TIME POINT.  SETUP1
C SET UP INITIAL VALUES OF ARRAYS U1,U2 AND V1.   SETUP1
C                                                    SETUP1
C IMPLICIT REAL*8(A-H,O-Z)                        SETUP1
COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,  SETUP1
* NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG      PAGE
COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30), SETUP1
* SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)         SETUP1
COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60), SLIP
* RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),      SETUP1
* JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)     SETUP1
COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60), SETUP1
* F(3,30),TQ(3,30),WJ(30),A11(3,3,30)                SLIP
COMMON/CEULER/ IEULER(30),HIR(3,3,90),ANG(3,30),ANGD(3,30), SLIP
* FE(3,30),TQE(3,30),CONST(5,30)                     SLIP
COMMON/TEMPVS/T(3),S(3),T1(3),T2(3),T3(3),T4(3),T5(3),T6(3),  SETUP1
* T7(3),T8(3),T9(3),T10(3),T11(3),T12(3),HH(3),      SETUP1
* TT1(3,3),TT2(3,3),S1,SQS1,S2,S3,S4,V1T,SR2        SLIP
* ,DDMY(35043)                                         80386
DATA IFIRST/1/                                        SLIP
C                                                    SETUP1
CALL ELTIME(1,10)                                    SETUP1
IF (IFIRST.EQ.0) GO TO 15                            SLIP
IF (NJNT.EQ.0) GO TO 15                             SLIP
DO 10 I = 1,NJNT                                    SLIP
DO 10 J = 1,3                                       SLIP
DO 8 K = 1,3                                         SLIP
8 A11(J,K,I) = 0.0                                   SLIP
10 A11(J,J,I) = 1.0                                 SLIP
IFIRST = 0                                          SLIP
15 DO 20 I=1,NGRND                                  SLIP
C                                                    SETUP1
C SET EACH U1N = 0                                  SETUP1
C                                                    SETUP1
C U1(1,I) = 0.0                                     SETUP1
C U1(2,I) = 0.0                                     SETUP1
C U1(3,I) = 0.0                                     SETUP1
C                                                    SETUP1
C SET EACH U2N = WNX(PHIN*WN)                      SETUP1
C                                                    SETUP1
C U2(1,I) = WMEG(2,I)*WMEG(3,I) * (PHI(2,I)-PHI(3,I)) SETUP1
C U2(2,I) = WMEG(1,I)*WMEG(3,I) * (PHI(3,I)-PHI(1,I)) SETUP1
20 U2(3,I) = WMEG(1,I)*WMEG(2,I) * (PHI(1,I)-PHI(2,I)) SETUP1
IF (NPRT(11).NE.0) WRITE (6,21) ((U2(I,J),I=1,3),J=1,NSEG) SETUP1
21 FORMAT(' U2 ARRAY'/(1X,1P9D14.4))              SETUP1
IF (NJNT.LE.0) GO TO 98                             SETUP1
DO 40 J=1,NJNT                                     SETUP1
DO 31 K=1,3                                         SETUP1
T1(K) = SR(K,2*J-1)                                SLIP

```

```

T2(K) = SR(K,2*J )
IF (IABS(IPIN(J)).LT.5) GO TO 31
IF (IEULER(J).EQ.-1) GO TO 31
T1(K) = T1(K) + SR(4,2*J-1)*HT(K,3,2*J-1)
31 V1(K,J) = 0.0
I = IABS(JNT(J))
IF (I.LE.0) GO TO 40
C
C FOR EACH JOINT SET
C B12(2J-1) = B12(J,I ) = -D(I)' * SR(2J-1) X
C B12(2J ) = B12(J,J+1) = D(J+1)' * SR(2J) X
C
C B12(1,1,2*J-1) = D(3,1,I)*T1(2) - D(2,1,I)*T1(3)
C B12(2,1,2*J-1) = D(3,2,I)*T1(2) - D(2,2,I)*T1(3)
C B12(3,1,2*J-1) = D(3,3,I)*T1(2) - D(2,3,I)*T1(3)
C B12(1,2,2*J-1) = D(1,1,I)*T1(3) - D(3,1,I)*T1(1)
C B12(2,2,2*J-1) = D(1,2,I)*T1(3) - D(3,2,I)*T1(1)
C B12(3,2,2*J-1) = D(1,3,I)*T1(3) - D(3,3,I)*T1(1)
C B12(1,3,2*J-1) = D(2,1,I)*T1(1) - D(1,1,I)*T1(2)
C B12(2,3,2*J-1) = D(2,2,I)*T1(1) - D(1,2,I)*T1(2)
C B12(3,3,2*J-1) = D(2,3,I)*T1(1) - D(1,3,I)*T1(2)
C
C B12(1,1,2*J ) = D(2,1,J+1)*T2(3) - D(3,1,J+1)*T2(2)
C B12(2,1,2*J ) = D(2,2,J+1)*T2(3) - D(3,2,J+1)*T2(2)
C B12(3,1,2*J ) = D(2,3,J+1)*T2(3) - D(3,3,J+1)*T2(2)
C B12(1,2,2*J ) = D(3,1,J+1)*T2(1) - D(1,1,J+1)*T2(3)
C B12(2,2,2*J ) = D(3,2,J+1)*T2(1) - D(1,2,J+1)*T2(3)
C B12(3,2,2*J ) = D(3,3,J+1)*T2(1) - D(1,3,J+1)*T2(3)
C B12(1,3,2*J ) = D(1,1,J+1)*T2(2) - D(2,1,J+1)*T2(1)
C B12(2,3,2*J ) = D(1,2,J+1)*T2(2) - D(2,2,J+1)*T2(1)
C B12(3,3,2*J ) = D(1,3,J+1)*T2(2) - D(2,3,J+1)*T2(1)
C
C NOTE THAT FOR EACH JOINT
C A21(M,N) = B12(N,M)
C
C FOR EACH JOINT SET
C V1(J) = -D(I)'*W(I)X( W(I)XSR(2J-1) )
C +D(J+1)'*W(J+1)X( W(J+1)XSR(2J) )
C
C CALL CROSS(WMEG(1,I),T1,T)
C CALL CROSS(WMEG(1,I),T,S)
C CALL DOT31(D(1,1,I),S,V1(1,J))
C CALL CROSS(WMEG(1,J+1),T2,T)
C CALL CROSS(WMEG(1,J+1),T,S)
C CALL DOT31(D(1,1,J+1),S,T)
C DO 32 K=1,3
32 V1(K,J) = T(K) - V1(K,J)
IF (IABS(IPIN(J)).LT.5) GO TO 40
IF (IEULER(J).EQ.-1) GO TO 40
CALL DOT31(D(1,1,I),HT(1,3,2*J-1),T4)
CALL CROSS(WMEG(1,I),HT(1,3,2*J-1),T5)
CALL DOT31(D(1,1,I),T5,T6)

```

```

SLIP
SLIP
SLIP
SLIP
SETUP1
SETUP1
SETUP1
SETUP1
SETUP1
SETUP1
SETUP1
SLIP
SLIP
SLIP
SLIP
SLIP
SLIP
SLIP
SLIP
SLIP
SLIP
SETUP1
SLIP
SLIP
SLIP
SLIP
SLIP
SLIP
SLIP
SETUP1
SETUP1
SETUP1
SETUP1
SETUP1
SETUP1
SETUP1
SETUP1
SLIP
SETUP1
SETUP1
SETUP1
SETUP1
SETUP1
SETUP1
SETUP1
SLIP
SLIP
SLIP
SLIP
SLIP
SLIP

```

```

V1T = V1(1,J)*T4(1) + V1(2,J)*T4(2) + V1(3,J)*T4(3)      SLIP
SR2 = 2.0*SR(4,2*J)                                         SLIP
DO 34 K = 1,3                                                SLIP
V1(K,J) = V1(K,J) - V1T*T4(K) - SR2*T6(K)                 SLIP
S1=T4(1)*B12(1,K,2*J-1)+T4(2)*B12(2,K,2*J-1)+T4(3)*B12(3,K,2*J-1) SLIP
S2=T4(1)*B12(1,K,2*J )+T4(2)*B12(2,K,2*J )+T4(3)*B12(3,K,2*J ) SLIP
DO 33 L = 1,3                                                SLIP
A11(K,L,J) = -T4(K)*T4(L)                                   SLIP
B12(L,K,2*J-1) = B12(L,K,2*J-1) - S1*T4(L)               SLIP
33 B12(L,K,2*J ) = B12(L,K,2*J ) - S2*T4(L)              SLIP
34 A11(K,K,J) = 1.0 + A11(K,K,J)                            SLIP
40 CONTINUE                                                  SLIP
IF (NPRT(11).NE.0) WRITE (6,41) ((V1(I,J),I=1,3),J=1,NJNT) SETUP1
41 FORMAT(' V1 ARRAY'/(1X,1P9D14.4))                       SETUP1
C                                                            SETUP1
C IF IPIN(M)=1, SET V2(M)=(WN.HN-WM.HM)DN'WNXHN           SETUP1
C                                                            SETUP1
DO 50 J=1,NJNT                                             SETUP1
DO 43 K=1,3                                                SETUP1
43 V2(K,J) = 0.0                                           SETUP1
IF (IPIN(J).LT.1) GO TO 50                                SLIP
IF (IPIN(J).GT.1.AND.IPIN(J).LT.6) GOTO 50              SLIP
I = IABS(JNT(J))                                          SETUP1
CALL CROSS (WMEG(1,I ),HB(1,2*J-1),T)                   SETUP1
CALL DOT31 (D(1,1,I ),T,T1)                              SETUP1
C CALL CROSS (WMEG(1,J+1),HB(1,2*J ),T)                 SETUP1
C CALL DOT31 (D(1,1,J+1),T,T2)                           SETUP1
S1 = WMEG(1,I)*HB(1,2*J-1)                                SETUP1
* + WMEG(2,I)*HB(2,2*J-1)                                SETUP1
* + WMEG(3,I)*HB(3,2*J-1)                                SETUP1
S2 = WMEG(1,J+1)*HB(1,2*J)                                SETUP1
* + WMEG(2,J+1)*HB(2,2*J)                                SETUP1
* + WMEG(3,J+1)*HB(3,2*J)                                SETUP1
DO 44 K=1,3                                                SETUP1
C 44 V2(K,J) = S1*T1(K) - S2*T2(K)                       SETUP1
44 V2(K,J) = (S1-S2)*T1(K)                                SETUP1
50 CONTINUE                                               SETUP1
98 CALL ELTIME(2,10)                                       SETUP1
RETURN                                                    SETUP1
END                                                       SETUP1

```

	SUBROUTINE SETUP2	REV IV 07/24/86	SLIP
C			
C	CALL BY DAUX AFTER CONTACT ROUTINES AND BY UPDATE PRIOR TO		SETUP2
C	DAUX TO SET UP A2 ARRAY AND (FOR NQ#0) THE A13,A23 AND V3 ARRAYS.		SETUP2
C			SETUP2
	IMPLICIT REAL*8(A-H,O-Z)		SETUP2
	COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,		SETUP2
*	NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG		PAGE
	COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),		SETUP2
*	SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)		SETUP2
	COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60),		SLIP
*	RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),		SETUP2
*	JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)		SETUP2
	COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60),		SETUP2
*	F(3,30),TQ(3,30),WJ(30),A11(3,3,30)		SLIP
	COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),		SETUP2
*	HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12),		SETUP2
*	RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),		SETUP2
*	KQ1(12),KQ2(12),KQTYPE(12)		SETUP2
	COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),		SETUP2
*	UNITL,UNITM,UNITT,GRAVTY(3),TWOPI		TWOPI
	LOGICAL*1 FREE,LDDMY		80386
	COMMON/TEMPVS/T(3),S(3),T1(3),T2(3),T3(3),T4(3),T5(3),T6(3),		SETUP2
*	T7(3),T8(3),T9(3),T10(3),T11(3),T12(3),HH(3),		SETUP2
*	TT1(3,3),TT2(3,3),S1,SQS1,S2,S3,S4		SETUP2
*	,WCRM(3),RM(3),WCM(3),WWCM(3),WWM(3),RBA(3),BA		SETUP2
*	,WCRN(3),RN(3),WCN(3),WWCN(3),WWN(3),RBAD(3)		SETUP2
*	,IDUM(14290),FREE(30),LDDMY(2),DDMY(27859)		80386
C			SETUP2
	CALL ELTIME(1,26)		SETUP2
C			SETUP2
C	COMPUTE A22 ARRAY VIA DHHPIN FOR DAUX2 ROUTINES.		SETUP2
C			SETUP2
	IF (NJNT.EQ.0) GO TO 50		SETUP2
	DO 49 M=1,NJNT		SETUP2
	FREE(M) = .TRUE.		SLIP
	N = IABS(JNT(M))		SETUP2
	IF (N.EQ.0) GO TO 49		SETUP2
	IF (IPIN(M).EQ.0) GOTO 49		SLIP
	IF (IPIN(M).GE.2.AND. IPIN(M).LE.5) GO TO 49		SLIP
	FREE(M) = .FALSE.		SLIP
	CALL DHHPIN(A22(1,1,2*M-1),T,N,M,2*M-1)		SETUP2
	CALL DHHPIN(A22(1,1,2*M),T,M+1,M,2*M)		SETUP2
49	CONTINUE		SETUP2
C			PECONV
C	THIS STATEMENT IS NECESSARY FOR THE PROGRAM TO RUN ON THE		PECONV
C	P&E FORTRAN VII O (REV 4) COMPILER		PECONV
C			PECONV
	NNNET = IPIN(M)		PECONV
C			SETUP2
C	SET UP A13,A23 AND V3 ARRAYS FOR DAUX33.		SETUP2
C			SETUP2

50	IF (NQ.EQ.0) GO TO 98	SETUP2
	DO 70 K=1,NQ	SETUP2
	IF (KQTYPE(K).LT.0) GO TO 70	SETUP2
	IF (KQTYPE(K).EQ.5) GO TO 70	SETUP2
	M = KQ1(K)	SETUP2
	N = KQ2(K)	SETUP2
	IF (KQTYPE(K).EQ.2 .OR. KQTYPE(K).EQ.4) GO TO 53	SETUP2
C		SETUP2
C	FOR KQTYPE = 1 OR 3, SET HHT = I	SETUP2
C		SETUP2
	DO 52 J=1,3	SETUP2
	DO 51 I=1,3	SETUP2
51	HHT(I,J,K) = 0.0	SETUP2
52	HHT(J,J,K) = 1.0	SETUP2
	IF (KQTYPE(K).NE.6) GO TO 61	SETUP2
C		SETUP2
C	FOR KQTYPE=6, SET HHT= I-TT'	SETUP2
C		SETUP2
	DO 60 J=1,3	SETUP2
	DO 60 I=1,3	SETUP2
60	HHT(I,J,K) = HHT(I,J,K) - TQQ(I,K)*TQQ(J,K)	SETUP2
	GO TO 61	SETUP2
53	IF (KQTYPE(K).NE.2) GO TO 56	SETUP2
C		SETUP2
C	FOR KQTYPE=2, COMPUTE HH AND HHT.	SETUP2
C		SETUP2
	CALL DOT31(D(1,1,M),RK1(1,K),T1)	SETUP2
	CALL DOT31(D(1,1,N),RK2(1,K),T2)	SETUP2
	S1 = 0.0	SETUP2
	DO 54 I=1,3	SETUP2
	HH(I) = SEGLP(I,M)+T1(I) - SEGLP(I,N)-T2(I)	SETUP2
54	S1 = S1 + HH(I)**2	SETUP2
	SQS1 = DSQRT(S1)	SETUP2
	DO 55 I=1,3	SETUP2
	HH(I) = HH(I)/SQS1	SETUP2
55	IF (DABS(HH(I)).LE.EPS(12)) HH(I) = 0.0	SETUP2
	CALL DOT31(HH,HH,HHT(1,1,K))	SETUP2
56	IF (KQTYPE(K).NE.4) GO TO 61	SETUP2
C		SETUP2
C	FOR KQTYPE = 4, SET HHT = HHT	SETUP2
C		SETUP2
	CALL DOT31(HQQ(1,K),HQQ(1,K),HHT(1,1,K))	SETUP2
C		SETUP2
C	SET A13(2K-1) = HHT	SETUP2
C	AND A13(2K) = -HHT	SETUP2
C		SETUP2
61	DO 62 J=1,3	SETUP2
	DO 62 I=1,3	SETUP2
	A13(I,J,2*K-1) = HHT(I,J,K)	SETUP2
62	A13(I,J,2*K) = -HHT(I,J,K)	SETUP2
C		SETUP2
C	SET A23(2K-1) = (R1X)(D1)A13(2K-1)	SETUP2

C	AND A23(2K) = (R2X)(D2)A13(2K)	SETUP2
C		SETUP2
	CALL MAT33(D(1,1,M),A13(1,1,2*K-1),TT1)	SETUP2
	CALL MAT33(D(1,1,N),A13(1,1,2*K),TT2)	SETUP2
	DO 63 J=1,3	SETUP2
	CALL CROSS(RK1(1,K),TT1(1,J),A23(1,J,2*K-1))	SETUP2
63	CALL CROSS(RK2(1,K),TT2(1,J),A23(1,J,2*K))	SETUP2
	IF (KQTYPE(K).EQ.4) GO TO 72	SETUP2
C		SETUP2
C	FOR KQTYPE = 1,2 OR 3, SET B31 = A13' AND B32 = A23'	SETUP2
C		SETUP2
	DO 71 I=1,3	SETUP2
	DO 71 J=1,3	SETUP2
	B31(I,J,2*K-1) = A13(J,I,2*K-1)	SETUP2
	B31(I,J,2*K) = A13(J,I,2*K)	SETUP2
	B32(I,J,2*K-1) = A23(J,I,2*K-1)	SETUP2
71	B32(I,J,2*K) = A23(J,I,2*K)	SETUP2
	GO TO 76	SETUP2
C		SETUP2
C	FOR KQTYPE = 4, SET B31(2K-1) = HTT	SETUP2
C	B31(2K) = -HTT	SETUP2
C	B32 = (B31)(D')(RX)'	SETUP2
C		SETUP2
	72 CALL DOT31(HQQ(1,K),TQQ(1,K),B31(1,1,2*K-1))	SETUP2
	DO 73 I=1,3	SETUP2
	DO 73 J=1,3	SETUP2
	73 B31(I,J,2*K) = -B31(I,J,2*K-1)	SETUP2
	CALL DOT33(D(1,1,M),B31(1,1,2*K-1),B32(1,1,2*K-1))	SETUP2
	CALL DOT33(D(1,1,N),B31(1,1,2*K),B32(1,1,2*K))	SETUP2
	DO 74 J=1,3	SETUP2
	CALL CROSS(RK1(1,K),B32(1,J,2*K-1),TT1(1,J))	SETUP2
74	CALL CROSS(RK2(1,K),B32(1,J,2*K),TT2(1,J))	SETUP2
	DO 75 I=1,3	SETUP2
	DO 75 J=1,3	SETUP2
	B32(I,J,2*K-1) = TT1(J,I)	SETUP2
75	B32(I,J,2*K) = TT2(J,I)	SETUP2
C		SETUP2
C	COMPUTE V3 = D2'(W2X(W2XR2)) - D1'(W1X(W1XR1))	SETUP2
C		SETUP2
	76 CALL CROSS(WMEG(1,M),RK1(1,K),T3)	SETUP2
	CALL CROSS (WMEG(1,M),T3,T4)	SETUP2
	CALL DOT31 (D(1,1,M),T4,T5)	SETUP2
	CALL CROSS (WMEG(1,N),RK2(1,K),T6)	SETUP2
	CALL CROSS (WMEG(1,N),T6,T7)	SETUP2
	CALL DOT31 (D(1,1,N),T7,T8)	SETUP2
	DO 64 I=1,3	SETUP2
64	V3(I,K) = T8(I) - T5(I)	SETUP2
	IF (KQTYPE(K).NE.2) GO TO 67	SETUP2
C		SETUP2
C	RECOMPUTE V3 FOR KQTYPE=2.	SETUP2
C		SETUP2
	CALL DOT31 (D(1,1,M),T3,T9)	SETUP2

	CALL DOT31 (D(1,1,N),T6,T10)	SETUP2
	S2 = 0.0	SETUP2
	DO 65 I=1,3	SETUP2
	T11(I) = SEGLV(I,M)+T9(I) - SEGLV(I,N)-T10(I)	SETUP2
65	S2 = S2 + T11(I)**2	SETUP2
	S3 = HH(1)*V3(1,K) + HH(2)*V3(2,K) + HH(3)*V3(3,K)	SETUP2
	S4 = S3-S2/SQS1	SETUP2
	DO 66 I=1,3	SETUP2
66	V3(I,K) = S4*HH(I)	SETUP2
67	IF (KQTYPE(K).NE.3.AND.KQTYPE(K).NE.6) GO TO 77	SETUP2
C		SETUP2
C	FOR KQTYPE=3 OR 6, ADD R DOT TERM FROM PLELP OR SEGSEG TO V3.	SETUP2
C		SETUP2
	DO 68 I=1,3	SETUP2
68	V3(I,K) = V3(I,K) + RQQ(I,K)	SETUP2
	IF (KQTYPE(K).NE.6) GO TO 70	SETUP2
C		SETUP2
C	FOR KQTYPE=6, SET V3 = (I-TT')(V3+RQQ)	SETUP2
C		SETUP2
	VQQ = V3(1,K)*TQQ(1,K) + V3(2,K)*TQQ(2,K) + V3(3,K)*TQQ(3,K)	SETUP2
	DO 69 I=1,3	SETUP2
69	V3(I,K) = V3(I,K) - VQQ*TQQ(I,K)	SETUP2
77	IF (KQTYPE(K).NE.4) GO TO 70	SETUP2
C		SETUP2
C	FOR KQTYPE = 4, ADD R TERM FROM PLELP OR SEGSEG TO V3.	SETUP2
C		SETUP2
	S3 = TQQ(1,K)*V3(1,K) + TQQ(2,K)*V3(2,K) + TQQ(3,K)*V3(3,K)	SETUP2
	S4 = S3+SQQ(K)	SETUP2
	DO 78 I=1,3	SETUP2
78	V3(I,K) = S4*HQQ(I,K)	SETUP2
70	CONTINUE	SETUP2
C		SETUP2
C	SPECIAL SETUP FOR TENSION ELEMENTS (KQTYPE = 5).	SETUP2
C		SETUP2
	N = 0	SETUP2
79	N = N+1	SETUP2
	IF (N.GE.NQ) GO TO 98	SETUP2
	IF (KQTYPE(N).NE.5) GO TO 79	SETUP2
	DO 81 I=1,3	SETUP2
	DO 80 J=1,3	SETUP2
	A13(I,J,2*N-1) = 0.0	SETUP2
	A13(I,J,2*N) = 0.0	SETUP2
	A23(I,J,2*N) = 0.0	SETUP2
	B31(I,J,2*N-1) = 0.0	SETUP2
	B31(I,J,2*N) = 0.0	SETUP2
	A13(I,J,2*N+1) = 0.0	SETUP2
	A13(I,J,2*N+2) = 0.0	SETUP2
	A23(I,J,2*N+1) = 0.0	SETUP2
	B31(I,J,2*N+1) = 0.0	SETUP2
	B31(I,J,2*N+2) = 0.0	SETUP2
	HHT(I,J,N) = 0.0	SETUP2
80	HHT(I,J,N+1) = 0.0	SETUP2

```

A13(I,I,2*N-1) = 1.0
B31(I,I,2*N-1) = RK1(1,N+1)
B31(I,I,2*N ) = RK1(3,N+1)
A13(I,I,2*N+2) = 1.0
B31(I,I,2*N+1) = RK1(3,N+1)
81 B31(I,I,2*N+2) = RK1(2,N+1)
N1 = KQ1(N)
N2 = KQ2(N)
DO 82 K=1,3
CALL CROSS(RK1(1,N),D(1,K,N1),A23(1,K,2*N-1))
82 CALL CROSS(RK2(1,N),D(1,K,N2),A23(1,K,2*N+2))
DO 83 I=1,3
DO 83 J=1,3
B32(I,J,2*N-1) = RK1(1,N+1)*A23(J,I,2*N-1)
B32(I,J,2*N ) = RK1(3,N+1)*A23(J,I,2*N+2)
B32(I,J,2*N+1) = RK1(3,N+1)*A23(J,I,2*N-1)
83 B32(I,J,2*N+2) = RK1(2,N+1)*A23(J,I,2*N+2)
CALL CROSS(WMEG(1,N1),RK1(1,N),WCRM)
CALL CROSS(WMEG(1,N2),RK2(1,N),WCRN)
CALL DOT31(D(1,1,N1),RK1(1,N),RM)
CALL DOT31(D(1,1,N2),RK2(1,N),RN)
CALL DOT31(D(1,1,N1),WCRM,WCM)
CALL DOT31(D(1,1,N2),WCRN,WCN)
BA = 0.0
DO 84 I=1,3
RBA (I) = SEGLP(I,N2) + RN (I) - SEGLP(I,N1) - RM (I)
RBA(I) = SEGLV(I,N2) + WCN(I) - SEGLV(I,N1) - WCM(I)
84 BA = BA + RBA(I)**2
BA = DSQRT(BA)
FORCE = 0.0
IF (BA.GT.RK2(3,N+1)) FORCE = RK2(1,N+1)*(1.0-RK2(3,N+1)/BA)
DO 85 I=1,3
V3(I,N) = RK2(2,N+1)*RBA(I) + FORCE*RBA(I)
85 V3(I,N+1) = -V3(I,N)
CALL CROSS(WMEG(1,N1),WCRM,WWCM)
CALL CROSS(WMEG(1,N2),WCRN,WWCN)
CALL DOT31(D(1,1,N1),WWCM,WWM)
CALL DOT31(D(1,1,N2),WWCN,WWN)
DO 86 I=1,3
V3(I,N ) = V3(I,N ) - RK1(1,N+1)*WWM(I) - RK1(3,N+1)*WWN(I)
86 V3(I,N+1) = V3(I,N+1) - RK1(3,N+1)*WWM(I) - RK1(2,N+1)*WWN(I)
N = N+1
GO TO 79
98 CALL ELTIME(2,26)
RETURN
END

```

```

SUBROUTINE SINPUT                                SINPUT
C                                                    REV IV 02/20/87HYPER
C READS AND PRINTS THE INPUT CARDS THAT DESCRIBE THE PHYSICAL SINPUT
C DIMENSIONS OF THE PLANES REPRESENTING THE VEHICLE PANELS AND OF SINPUT
C THE RESTRAINT BELTS. ALSO PROCESSES THOSE DATA CARDS THAT DESCRIBESINPUT
C ADDITIONAL CONTACT ELLIPSOIDS, CONSTRAINTS, BODY SEGMENT SYMMETRY SINPUT
C OPTIONS AND SPRING DAMPER FUNCTIONS. SINPUT
C SINPUT
C IMPLICIT REAL*8 (A-H,O-Z) SINPUT
COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,XGRND, SINPUT
* NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG PAGE
COMMON/CNTRF/ PL(24,30),BELT(20,8),TPTS(6,8),BD(24,40) EDGE
COMMON/SGMNTS/ D(3,3,36),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30), SINPUT
* SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30) SINPUT
COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24), SINPUT
* HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TOQ(3,12), SINPUT
* RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12), SINPUT
* KQ1(12),KQ2(12),KQTYPE(12) SINPUT
COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5), SINPUT
* BLTTL(5,8),PLTTL(5,30),BAGTTL(5,6),SEG(30), SINPUT
* JOINT(30),CGS(30),JS(30) SINPUT
REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTL,PLTTL,BAGTTL,SEG,JOINT SINPUT
LOGICAL*1 CGS,JS,LP4 HYPER
COMMON/DAMPER/ APSDM(3,20),APSDN(3,20),ASD(5,20),MSDM(20),MSDN(20) SINPUT
COMMON/WINDFR/ WTIME(30),QFU(3,5),QFV(3,5),WF(3,30),IWIND(30), WINDOP
* MWSEG(7,30),NFVSEG(6),NFVNT(5),MOWSEG(30,30) WINDOP
COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24), SINPUT
* UNITL,UNITM,UNITT,GRAVITY(3),TWOPI TWOPI
COMMON/TEMPVS/ P1(3),P2(3),P3(3),P4(3),DE(3,3),DDMY(35092) 80386
DIMENSION IDYPR(3) SINPUT
DATA IDYPR/3,2,1/ SINPUT
DATA MAXBD/40/ CHGIII
DATA NPLMAX/30/,NBLTMX/8/,NBAGMX/5/,NELPMX/40/,NQMAX/12/, MISC
* NSDMAX/20/,NHRNSM/5/,NWINDM/50/,NJNTFM/50/,NFORCM/5/ MISC
C SINPUT
C INPUT CARD D.1 SINPUT
C SINPUT
READ (5,11) NPL,NBLT,NBAG,NELP,NQ,NSD,NHRNSS,NWINDF,NJNTF,NFORCESINPUT
11 FORMAT(12I6) SINPUT
WRITE (6,16) NPG,NPL,NBLT,NBAG,NELP,NQ,NSD,NHRNSS,NWINDF,NJNTF, PAGE
* NFORCE PAGE
NPG=NPG+1 PAGE
16 FORMAT('1 NPL NBLT NBAG NELP NQ NSD NHRNSS',PAGE
* ' NWINDF NJNTF NFORCE',43X,'PAGE',I5/10I8,40X,'CARD D.1')PAGE
IF (NPL.GT.NPLMAX) STOP 65 CHGIII
IF (NBLT.GT.NBLTMX) STOP 66 MISC
IF (NBAG.GT.NBAGMX) STOP 67 MISC
IF (NELP.GT.NELPMX) STOP 68 MISC
IF (NQ.GT.NQMAX) STOP 69 CHGIII
IF (NSD.GT.NSDMAX) STOP 70 CHGIII
IF (NHRNSS.GT.NHRNSM) STOP 71 MISC
IF (NWINDF.GT.NWINDM) STOP 72 MISC

```

	IF (MINTF.GT.NJNTFM) STOP 73	MISC
	IF (NFORCE.GT.NFORCM) STOP 74	MISC
	IF (NPL.EQ.0) GO TO 15	SINPUT
	IPAGE = 0	SINPUT
	DO 20 J=1,NPL	SINPUT
C		SINPUT
C	READ AND PRINT CARDS D.2.A,D.2.B AND D.2.C FOR THE JTH PLANE.	SINPUT
C		SINPUT
	READ (5,23) JJ,(PLTTL(I,J),I = 1,5),P1,P2,P3	SINPUT
	23 FORMAT (I4,4X,5A4/(3F12.0))	SINPUT
	IF (JJ.NE.J) WRITE (6,24) JJ,J	SINPUT
	24 FORMAT (' PLANE INDEX INPUT ERROR,' ,2I4)	SINPUT
	IF (JJ.NE.J) STOP 10	SINPUT
	IF (MOD(J,7).EQ.1.AND.IPAGE.EQ.0) WRITE (6,12) IPAGE	PAGE
	IF (MOD(J,7).EQ.1.AND.IPAGE.EQ.1) WRITE (6,112) IPAGE,NPG	PAGE
	IF (MOD(J,7).EQ.1.AND.IPAGE.EQ.1) NPG=NPG+1	PAGE
	112 FORMAT(11,' PLANE INPUTS',109X,'PAGE',I5/120X,'CARDS D.2')	PAGE
	12 FORMAT(11,' PLANE INPUTS',106X,'CARDS D.2')	SINPUT
	IPAGE = 1	SINPUT
	WRITE (6,25) J, (PLTTL(I,J),I = 1,5),P1,P2,P3	SINPUT
	25 FORMAT('0 PLANE NO.',I4,4X,5A4//17X,'X',11X,'Y',11X,'Z'/	SINPUT
	* ' POINT 1 ' ,3F12.4/	SINPUT
	* ' POINT 2 ' ,3F12.4/	SINPUT
	* ' POINT 3 ' ,3F12.4)	SINPUT
C		SINPUT
C	PROGRAM NOW ASSUMES THE FINITE PLANE IS A PARALLELOGRAM IN SHAPE	SINPUT
C	WHERE THE INPUT POINTS P1,P2,P3 ARE 3 OF THE CORNERS SUCH THAT	SINPUT
C	EDGE P1-P2 IS LESS THAN 180 DEGREES CLOCKWISE (AS VIEWED BY THE	SINPUT
C	OCCUPANT) FROM THE EDGE P1-P3.	SINPUT
C		SINPUT
C	SET UP PL ARRAY AS REQUIRED BY SUBROUTINE PLELP	SINPUT
C		SINPUT
C	PL(1,J) = A0 NORMAL EQUATION OF JTH PLACE	SINPUT
C	PL(2,J) = B0 A0*X + B0*Y + C0*Z = D0	SINPUT
C	PL(3,J) = C0	SINPUT
C	PL(4,J) = D0	SINPUT
C		SINPUT
C	PL(5,J)	SINPUT
C	PL(6,J) POINT 1	EDGE
C	PL(7,J)	SINPUT
C		SINPUT
C	PL(8,J) =A1	SINPUT
C	PL(9,J) =B1 NORMAL EQUATION OF 1ST BOUNDARY PLANE	SINPUT
C	PL(10,J)=C1 A1*X + B1*Y + C1*Z = D1	SINPUT
C	PL(11,J)=D1 AND E1 IS LENGTH OF PLANE FROM BOUNDARY.	SINPUT
C	PL(12,J)=E1	SINPUT
C		SINPUT
C	PL(13,J)=A2	SINPUT
C	PL(14,J)=B2 NORMAL EQUATION OF 2ND BOUNDARY PLANE	SINPUT
C	PL(15,J)=C2 A2*X + B2*Y + C2*Z = D2	SINPUT
C	PL(16,J)=D2 AND E2 IS LENGTH OF PLANE FROM BOUNDARY.	SINPUT
C	PL(17,J)=E2	SINPUT

C			SINPUT
C	PL(18,J)		EDGE
C	PL(19,J)	POINT 2 - POINT 1	EDGE
C	PL(20,J)		EDGE
C			EDGE
C	PL(21,J)		EDGE
C	PL(22,J)	POINT 3 - POINT 1	EDGE
C	PL(23,J)		EDGE
C			EDGE
C	PL(24,J)	NOT CURRENTLY USED	EDGE
	S22 = 0.0		SINPUT
	S23 = 0.0		SINPUT
	S33 = 0.0		SINPUT
	DO 26 I =1,3		SINPUT
	P2(I) = P2(I)-P1(I)		SINPUT
	P3(I) = P3(I)-P1(I)		SINPUT
	PL(I+ 4,J) = P1(I)		EDGE
	PL(I+17,J) = P2(I)		EDGE
	PL(I+20,J) = P3(I)		EDGE
	S22 = S22 + P2(I)*P2(I)		SINPUT
	S23 = S23 + P2(I)*P3(I)		SINPUT
26	S33 = S33 + P3(I)*P3(I)		SINPUT
	S2 = DSQRT(S22)		SINPUT
	S3 = DSQRT(S33)		SINPUT
	CALL CROSS(P2,P3,PL(1,J))		SINPUT
	S1 = 0.0		SINPUT
	DO 27 I=1,3		SINPUT
27	S1 = S1 + PL(I,J)**2		SINPUT
	S1 = DSQRT(S1)		SINPUT
	DO 28 I=1,3		SINPUT
	PL(I,J) = PL(I,J)/S1		SINPUT
	PL(I+7 ,J) = (S33*P2(I) - S23*P3(I)) / (S1*S3)		SINPUT
28	PL(I+12,J) = (S22*P3(I) - S23*P2(I)) / (S1*S2)		SINPUT
	PL(4,J) = P1(1)*PL(1,J) + P1(2)*PL(2,J) + P1(3)*PL(3,J)		SINPUT
	PL(11,J) = P1(1)*PL(8,J) + P1(2)*PL(9,J) + P1(3)*PL(10,J)		SINPUT
	PL(12,J) = P2(1)*PL(8,J) + P2(2)*PL(9,J) + P2(3)*PL(10,J)		SINPUT
	PL(16,J) = P1(1)*PL(13,J) + P1(2)*PL(14,J) + P1(3)*PL(15,J)		SINPUT
20	PL(17,J) = P3(1)*PL(13,J) + P3(2)*PL(14,J) + P3(3)*PL(15,J)		SINPUT
15	IF (NBLT.EQ.0) GO TO 35		SINPUT
	DO 30 J=1,NBLT		SINPUT
C			SINPUT
C	READ AND PRINT CARDS D.3.A, D.3.B AND D.3.C FOR THE JTH BELT.		SINPUT
C			SINPUT
	READ (5,13) (BLTTTL(I,J),I - 1,5),(BELT(I,J),I - 1,11)		SINPUT
13	FORMAT (5A4/(6F12.0))		SINPUT
	IF (MOD(J,5).EQ.1) WRITE (6,21) NPG		PAGE
	IF (MOD(J,5).EQ.1) NPG=NPG+1		PAGE
21	FORMAT('1 BELT INPUTS',110X,'PAGE',15/120X,'CARDS D.3')		PAGE
30	WRITE (6,14) J,(BLTTTL(I,J),I - 1,5),(BELT(I,J),I - 1,11)		SINPUT
14	FORMAT('0 BELT NO.',I4,4X,5A4//		SINPUT
	* 30X,'ANCHOR POINT A',46X,'ANCHOR POINT B'//		SINPUT
	* 2(16X,'X',19X,'Y',19X,'Z',3X)/6F20.3//		SINPUT

	* 26X, 'FIXED POINT ON SEGMENT', 45X, 'SLACK(+)' /	SINPUT
	* 16X, 'X', 19X, 'Y', 19X, 'Z', 17X, 'BLANK', 13X, 'LENGTH(-)' / 5F20.3)	SINPUT
C		SINPUT
C	CALL AIRBG1 ROUTINE IF REQUIRED FOR AIRBAG INPUT	SINPUT
C		SINPUT
	35 IF (NBAG.NE.0) CALL AIRBG1	SINPUT
	IF (NELP.LE.0) GO TO 51	SINPUT
C		SINPUT
C	READ AND PRINT CARDS D.5 FOR ELLIPSOID INPUT, IF ANY.	SINPUT
C	NOTE: NELP IS THE NO. OF ELLIPSOIDS TO BE SUPPLIED HERE, NOT THE	SINPUT
C	NO. OF ELLIPSOIDS IN THE PROGRAM, SINCE THE FIRST NSEG	SINPUT
C	ELLIPSOIDS WERE SUPPLIED ON CARDS B.2.A - B.2.1. HOWEVER	SINPUT
C	THEY MAY BE REPLACED HERE IF DESIRED.	SINPUT
C		SINPUT
	WRITE (6,41) NPG,UNITL,UNITL	PAGE
	NPG=NPG+1	PAGE
	41 FORMAT('1 ADDITIONAL ELLIPSOID INPUT', 95X, 'PAGE', I5/120X,	PAGE
	* 'CARDS D.5' / 17X, 'SEMIAXES (' , A4, ')', 18X, 'OFFSET (' , A4, ')',	PAGE
	* 20X, 'ROTATION (DEG)', 15X, 'POWER' /	HYPER
	* 3X, 'NO.', 2(8X, 'X', 8X, 'Y', 8X, 'Z', 6X), 7X, 'YAW', 7X, 'PITCH', 5X,	SINPUT
	* 'ROLL' //)	SINPUT
	DO 50 MM=1, NELP	SINPUT
	READ (5,42) M, P1, P2, P3, P4	HYPER
	42 FORMAT(I6, 9F6.0, 3F4.0)	HYPER
	IF (M.GT.MAXBD) STOP 63	CHGIII
C		CHGIII
C	PREVENT EXTRA ELLIPSOIDS FROM CHANGING AIRBAG ELLIPSOIDS	CHGIII
C		CHGIII
	IF (M.GT.NVEH.AND.M.LT.NGRND) WRITE (6,330)	CHGIII
	330 FORMAT(3X, 'THE EXTRA CONTACT ELLIPSOID NUMBER IS THE SAME AS AN AICHGIII	CHGIII
	*RBAG ELLIPSOID')	CHGIII
	IF (M.GT.NVEH.AND.M.LT.NGRND) STOP 64	CHGIII
	WRITE (6,43) M, P1, P2, P3, P4	HYPER
	43 FORMAT(I6, 3(3X, 3F9.3, 3X), 3F6.0)	HYPER
	CALL DRCYPR (DE, P3, IDYPR)	SINPUT
	N = 1	HYPER
	LP4 = .FALSE.	HYPER
	DO 39 J = 1, 3	HYPER
	39 IF (P4(J).GT.2.0) LP4 = .TRUE.	HYPER
	IF (LP4) N = 2	HYPER
	DO 46 I = 1, 3	HYPER
	BD(N ,M) = P1(I)	HYPER
	BD(N+3,M) = P2(I)	HYPER
	IF (LP4) GO TO 46	HYPER
	DO 45 J=1, 3	SINPUT
	SUM1 = 0.0	SINPUT
	SUM2 = 0.0	SINPUT
	DO 44 L=1, 3	SINPUT
	SUM1 = SUM1 + DE(L,I)/P1(L)**2*DE(I, J)	SINPUT
	44 SUM2 = SUM2 + DE(L,I)*P1(L)**2*DE(L, J)	SINPUT
	K = 3*I + J + 3	SINPUT
	BD(K ,M) = SUM1	SINPUT

45	BD(K+9,M) = SUM2	SINPUT
46	N = N + 1	HYPER
	IF (.NOT.LP4) GO TO 50	HYPER
	BD(1,M) = -P4(1)	HYPER
	N = 8	HYPER
	DO 48 J = 1,3	HYPER
	BD(J+19,M) = P4(J)	HYPER
	IF (BD(J+19,M).EQ.0.0) BD(J+19,M) = BD(20,M)	HYPER
	BD(J+16,M) = 1.0/BD(J+1,M)**2	HYPER
	DO 48 I = 1,3	HYPER
	BD(N,M) = DE(I,J)	HYPER
48	N = N + 1	HYPER
	BD(23,M) = 0.0	HYPER
	IF (BD(20,M).NE.BD(21,M)) BD(23,M) = 1.0	HYPER
	IF (BD(21,M).NE.BD(22,M)) BD(23,M) = 1.0	HYPER
	IF (BD(22,M).NE.BD(20,M)) BD(23,M) = 1.0	HYPER
50	CONTINUE	SINPUT
C		SINPUT
C	READ AND PRINT CARDS D.6 FOR CONSTRAINT INPUT, IF ANY.	SINPUT
C		SINPUT
51	IF (NQ.LE.0) GO TO 70	SINPUT
	DO 60 K=1,NQ	SINPUT
	READ (5,52) KQTYPE(K),KQ1(K),KQ2(K), (RK1(I,K),I=1,3)	SINPUT
	* (RK2(I,K),I=1,3)	SINPUT
52	FORMAT(3I6,6F6.0)	SINPUT
	IF (K.EQ.1) WRITE (6,53) NPG,UNITL,UNITL	PAGE
	IF (K.EQ.1) NPG=NPG+1	PAGE
53	FORMAT('1 CONSTRAINT INPUT',105X,'PAGE',I5/120X,'CARDS D.6'/	PAGE
	* ' TYPE SEGMENT SEGMENT POINT ON 1ST SEGMENT ('	SINPUT
	* A4,')', ' POINT ON 2ND SEGMENT (' ,A4,')'/'	SINPUT
	* ' NO. NO. 1 NO. 2 X Y Z	SINPUT
	* X Y Z'//)	SINPUT
	WRITE (6,54) KQTYPE(K),KQ1(K),KQ2(K), (RK1(I,K),I=1,3)	SINPUT
	* (RK2(I,K),I=1,3)	SINPUT
54	FORMAT(I6,2I9,2(6X,3F9.3))	SINPUT
60	CONTINUE	SINPUT
C		SINPUT
C	CARD D.7 BODY SEGMENT SYMMETRY INPUT	SINPUT
C		SINPUT
70	READ (5,71) (NSYM(J),J=1,NSEG)	SINPUT
71	FORMAT(18I4)	SINPUT
	DO 103 J=1,NSEG	TGMOD2
	LJ = NSYM(J)	TGMOD2
	IF(IABS(LJ).GT.NSEG) GO TO 107	TGMOD2
	IF(LJ) 104,103,105	TGMOD2
105	LK = NSYM(LJ)	TGMOD2
	IF(IABS(LK).GT.NSEG) GO TO 107	TGMOD2
	IF(LK.NE.J) GO TO 106	TGMOD2
	GO TO 103	TGMOD2
104	JJ = -J	TGMOD2
	LJ = -LJ	TGMOD2
	LK = NSYM(LJ)	TGMOD2

IF(IABS(LK).GT.NSEG) GO TO 107	TGMOD2
IF((LK.NE.JJ).OR.(NSYM(J).EQ.JJ)) GO TO 106	TGMOD2
GO TO 103	TGMOD2
106 STOP 96	TGMOD2
107 STOP 97	TGMOD2
103 CONTINUE	TGMOD2
WRITE(6,72) (J,J-1,NSEG)	SINPUT
WRITE(6,73) (NSYM(J),J-1,NSEG)	SINPUT
72 FORMAT('0 BODY SEGMENT SYMMETRY INPUT',91X,'CARD D.7'//	SINPUT
* ' SEG NO.',30I4)	SINPUT
73 FORMAT('0 NSYM(J)',30I4)	SINPUT
NSEG1 = NSEG+1	SINPUT
DO 74 J=NSEG1,NGRND	SINPUT
74 NSYM(J) = 0	SINPUT
IF (NSD.LE.0) GO TO 90	SINPUT
C	SINPUT
C CARD D.8 SPRING DAMPERS FUNCTION INPUT.	SINPUT
C	SINPUT
DO 79 J=1,NSD	SINPUT
79 READ (5,80) MSDM(J),MSDN(J),(APSDM(I,J),I=1,3),	SINPUT
* (APSDN(I,J),I=1,3),(ASD(I,J),I=1,5)	SINPUT
80 FORMAT(2I3,11F6.0)	SINPUT
WRITE (6,81) UNITL	SINPUT
81 FORMAT('0',5X,'SPRING DAMPERS FUNCTION INPUT',82X,'CARDS D.8'//	SINPUT
* 18X,'COORDINATES OF ATTACHMENT POINTS (' ,A4,')'//	SINPUT
* 5X,'SEGMENT',9X,'SEGMENT M',16X,'SEGMENT N',15X,	SINPUT
* 'SPRING FORCE FUNCTION',12X,'DAMPING FORCE FUNCTION'//	AFREVS
* ' NO. M N',2(6X,'X',7X,'Y',7X,'Z',2X),7X,'DO',9X,'A1',11X,	SINPUT
* 'A2',13X,'B1',10X,'B2' //)	SINPUT
DO 82 J=1,NSD	SINPUT
82 WRITE (6,83) J,MSDM(J),MSDN(J),(APSDM(I,J),I=1,3),	SINPUT
* (APSDN(I,J),I=1,3),(ASD(I,J),I=1,5)	SINPUT
83 FORMAT(I3,2I4,2(1X,3F8.2),F11.2,2F12.3,F15.3,F12.3)	SINPUT
C	SINPUT
C CARDS D.9 FORCE AND/OR TORQUE FUNCTIONS.	CHGIII
C	SINPUT
90 NFVSEG(6)= NFORCE	SINPUT
IF (NFORCE.LE.0) GO TO 99	SINPUT
WRITE (6,91)	SINPUT
91 FORMAT ('0',6X,'FORCE AND/OR TORQUE FUNCTION INPUTS',78X,'CARDS D.	CHGIII
*9'//, 5X,'NO.', 5X,'SEG', 5X,'FCN', 13X,'X', 9X,'Y', 9X,'Z',	CHGIII
* 13X,'YAW', 6X,'PITCH', 6X,'ROLL' //)	SINPUT
DO 95 J=1,NFORCE	SINPUT
READ (5,92) NFVSEG(J),NFVNT(J),P1,P2	SINPUT
92 FORMAT (2I6,6F10.0)	SINPUT
WRITE (6,93) J,NFVSEG(J),NFVNT(J),P1,P2	SINPUT
93 FORMAT (3I8,6X,3F10.3,6X,3F10.3)	SINPUT
CALL DRCYPR (DE,P2,IDYPR)	SINPUT
DO 94 I=1,3	SINPUT
94 QFU(I,J) = DE(1,I)	FIXSPT
95 CALL CROSS (P1,QFU(1,J),QFV(1,J))	SINPUT
99 RETURN	SINPUT

END

SINPUT

```

SUBROUTINE SLPLOT (X, NX, XO, XN, XL, XSIZE, XLAB, NXLB, SLPLOT
*                Y, NY, YO, YN, YL, YSIZE, YLAB, NYLB, SLPLOT
*                NPTS, NYY, NDY, PLAB1, NPLB1, PLAB2, NPLB2) SLPLOT
C                REV III.2 08/08/84REVIII
C
C ARGUMENTS: SLPLOT
C X(NPTS) - ARRAY OF NPTS ABCISSAS TO BE PLOTTED. SLPLOT
C Y(NDY,NYY) - ARRAY OF NPTS*NY Y ORDINATES TO BE PLOTTED. SLPLOT
C NX,NY - POSITIVE - NO. OF LINEAR SUBDIVISIONS. SLPLOT
C          NEGATIVE - NO. OF LOGARITHMIC DECADES. SLPLOT
C XO,YO - AXES ORIGINS (POWER OF TEN IF NX,NY NEGATIVE). SLPLOT
C XN,YN - AXES END VALUES (REQUIRED IF NX,NY POSITIVE). SLPLOT
C XL,YL - LENGTH (INCHES) OF X,Y AXES. SLPLOT
C XSIZE,YSIZE - PAPER SIZE (INCHES) IN X,Y DIRECTIONS. SLPLOT
C XLAB,YLAB - X,Y AXES LABELS (ALPHANUMERIC ARRAYS). SLPLOT
C NXLB,NYLB - NO. OF CHARACTERS IN X,Y LABELS. SLPLOT
C NPTS - NO. OF POINTS IN X ARRAY AND EACH Y ARRAY. SLPLOT
C NYY - NO. OF Y ARRAYS TO BE PLOTTED VS. X ARRAY. SLPLOT
C NDY - FIRST DIMENSION OF Y ARRAY IN CALLING ROUTINE. SLPLOT
C          (NDY MUST BE .GE. NPTS) SLPLOT
C PLAB1,PLAB2 - 1ST & 2ND LINES OF PLOT ID LABELS (ALPHANUMERIC). SLPLOT
C NPLB1,NPLB2 - NO. OF CHARACTERS IN PLOT ID LABELS. SLPLOT
C
C NOTE: PLOTS WILL BE TRUNCATED AS FOLLOWS: SLPLOT
C NX,NY POSITIVE - XO,YO .LE. X,Y .LE. XN,YN SLPLOT
C NX,NY NEGATIVE - XO,YO .LE. X,Y .LE. XN*10**(-NX),YO*10**(-NY) SLPLOT
C
C DIMENSION X(NPTS),Y(NDY,NYY) 80386
C CHARACTER*4 XLAB(1),YLAB(1),PLAB1(1),PLAB2(1) 80386
C
C NOTE: THIS ROUTINE HAS BEEN WRITTEN FOR THE PLOTTING FACILITIES SLPLOT
C AT CALSPAN. THE FOLLOWING ITEMS ARE KNOWN TO BE CONTRARY TO THE SLPLOT
C NORMAL CALCOMP PROCEDURES AND SHOULD BE EXAMINED BY USERS AT OTHERS L PLOT
C COMPUTER SYSTEMS AND CHANGES MADE ACCORDINGLY. SLPLOT
C
C 1. AT CALSPAN THE PLOTTED CHARACTERS GENERATED BY SUBROUTINE SLPLOT
C SYMBOL HAVE A WIDTH OF 6/7 TIMES THE HEIGHT. FOR THE CALCOMP SLPLOT
C ROUTINES THE WIDTH IS EQUAL TO THE HEIGHT. THE STATEMENT SLPLOT
C 'WIDTHF = 6.0/7.0' SHOULD BE CHANGED TO 'WIDTHF = 1.0'. SLPLOT
C
C 2. THE ONLY INITIALIZATION REQUIRED AT CALSPAN IS THE STATEMENT SLPLOT
C 'CALL PLOT (0.0,0.0,0)' TO ESTABLISH A NEW PAGE, INCLUDING SLPLOT
C THE FIRST PAGE. THIS IS FOLLOWED BY 'CALL PLOT (XO,YO,-3)' TO SLPLOT
C SET THE PLOT ORIGIN ON THE PAGE. PROPER PLOT INITIALIZATION SLPLOT
C SHOULD BE DONE HERE AND IN SUBROUTINE POSTPR (AFTER STATEMENTS SLPLOT
C NO. 30) AS REQUIRED BY THE USER'S PLOTTING FACILITY. SLPLOT
C
C 3. THE STATEMENT 'CALL NEWPEN(2)' SHOULD BE EXAMINED OR DELETED. SLPLOT
C
C 4. THE STATEMENT 'CALL EF PLOT' AFTER STATEMENT NO. 50 IN POSTPR SLPLOT
C IS REQUIRED AT CALSPAN TO CLOSE OUT THE PLOT FILES. THIS SLPLOT
C SHOULD BE CHANGED TO CONFORM TO THE REQUIREMENTS OF THE SLPLOT

```

C	USER'S PLOTTING FACILITIES.	SLPLOT
C		SLPLOT
C	5. THE NECESSARY JOB CONTROL LANGUAGE FOR PLOTTING IS NECESSARY.	SLPLOT
C		SLPLOT
C	6. THE ONLY CALCOMP ROUTINES NEEDED ARE SYMBOL, NUMBER AND PLOT.	SLPLOT
C		SLPLOT
	LOGICAL NXPOS, NXNEG, NYPOS, NYNEG	SLPLOT
	DATA HN/0.07/, HL/0.105/	SLPLOT
	WIDTHF = 1.0	SLPLOT
	WN = WIDTHF*HN	REDIMN
	WL = WIDTHF*HL	SLPLOT
C	** PLOT PAGE INITIALIZATION **	SLPLOT
	CALL PLOT (0.0,0.0,-3)	CHANGE
	XP = 0.5*(XSIZE-(XL-0.5))	SLPLOT
	YP = 0.5*(YSIZE-(YL-1.0))	SLPLOT
	CALL PLOT (XP,YP,-3)	SLPLOT
	NXPOS = NX.GT.0	SLPLOT
	NXNEG = NX.LT.0	SLPLOT
	NYPOS = NY.GT.0	SLPLOT
	NYNEG = NY.LT.0	SLPLOT
C	** PLOT AXES AND ID LABELS. **	SLPLOT
	XP = 0.0	SLPLOT
	YP = 0.0	SLPLOT
	IF (.NOT.NXPOS) GO TO 12	SLPLOT
C	** LINEAR X AXIS **	SLPLOT
	CALL LINAXS (XP, YP, 0.0, NX, XL)	SLPLOT
	XB = XL/(XN-XO)	SLPLOT
C	** LINEAR X AXIS NUMERICS **	SLPLOT
	DX = XL/FLOAT(NX)	SLPLOT
	EX = XO	SLPLOT
	DD = (XN-XO)/FLOAT(NX)	SLPLOT
	ND = 0.99 - ALOG10(ABS(DD))	SLPLOT
	IF (ND.LE.0) ND = -1	SLPLOT
	IX = 0	SLPLOT
	YC = YP - 2.0*HN	SLPLOT
11	AX = ABS(EX)	SLPLOT
	NF = 0	SLPLOT
	IF (AX.GE.10.0) NF = ALOG10(AX)	SLPLOT
	NS = 0	SLPLOT
	IF (EX.LT.0.0) NS = 1	SLPLOT
	SP = NS+NF+2+ND	SLPLOT
	XC = XP - 0.5*SP*WN	SLPLOT
	CALL NUMBER (XC, YC, HN, EX, 0.0, ND)	SLPLOT
	XP = XP + DX	SLPLOT
	EX = EX + DD	SLPLOT
	IX = IX + 1	SLPLOT
	IF (ABS(EX).GT.ABS(0.1*DD)) GO TO 18	SLPLOT
	IF (IX.GT.NX) GO TO 12	SLPLOT
	CALL PLOT (XP, YP+YL,3)	SLPLOT
	CALL PLOT (XP, YP, 2)	SLPLOT
18	IF (IX.LE.NX) GO TO 11	SLPLOT
12	IF (.NOT.NXNEG) GO TO 14	SLPLOT

C	**	LOG X AXIS	**	SLPLOT
		CALL LOGAXS (XP, YP, 0.0, -NX, XL)		SLPLOT
		XB = XL/ALOG(10.0**(-NX))		SLPLOT
		XA = -XB*ALOG(XO)		SLPLOT
C	**	LOG X AXIS NUMERICS	**	SLPLOT
		DX = XL/FLOAT(-NX)		SLPLOT
		EX = ALOG10(XO)		SLPLOT
		IX = 0		SLPLOT
13		CALL NUMBER (XP-1.0*WN, YP-2.5*HN, HN, 10.0, 0.0, -1)		SLPLOT
		CALL NUMBER (XP+1.0*WN, YP-2.0*HN, HN, EX, 0.0, -1)		SLPLOT
		XP = XP + DX		SLPLOT
		EX = EX + 1.0		SLPLOT
		IX = IX - 1		SLPLOT
		IF (IX.GE.NX) GO TO 13		SLPLOT
14		IF (NXLB.LE.0) GO TO 15		SLPLOT
C	**	X AXIS LABEL	**	SLPLOT
		XPX = (XL-FLOAT(NXLB)*WL)/2.0		SLPLOT
		YPX = YP-4.0*HN-HL		SLPLOT
		CALL SYMBOL(XPX, YPX, HL, XLAB, 0.0, NXLB)		SLPLOT
15		IF (NPLB1.LE.0) GO TO 16		SLPLOT
C	**	PLOT LABEL - 1ST LINE	**	SLPLOT
		XP1 = (XL-FLOAT(NPLB1)*WL)/2.0		SLPLOT
		YP1 = YP-4.0*HN-4.0*HL		SLPLOT
		CALL SYMBOL(XP1, YP1, HL, PLAB1, 0.0, NPLB1)		SLPLOT
16		IF (NPLB2.LE.0) GO TO 20		SLPLOT
C	**	PLOT LABEL - 2ND LINE	**	SLPLOT
		XP2 = (XL-FLOAT(NPLB2)*WL)/2.0		SLPLOT
		YP2 = YP-4.0*HN-6.0*HL		SLPLOT
		CALL SYMBOL(XP2, YP2, HL, PLAB2, 0.0, NPLB2)		SLPLOT
20		XP = 0.0		SLPLOT
C	**	COMPLETE AXIS GRID	**	SLPLOT
		IF (NYPOS) CALL LINAXS (XL, YP, 90.0, NY, YL)		SLPLOT
		IF (NYNEG) CALL LOGAXS (XL, YP, 90.0, -NY, YL)		SLPLOT
		IF (NXPOS) CALL LINAXS (XL, YL, 180.0, NX, XL)		SLPLOT
		IF (NXNEG) CALL LOGAXS (XL, YL, 180.0, -NX, -XL)		SLPLOT
		IF (.NOT.NYPOS) GO TO 22		SLPLOT
C	**	LINEAR Y AXIS	**	SLPLOT
		CALL LINAXS (XP, YL, -90.0, NY, YL)		SLPLOT
		YB = YL/(YN-YO)		SLPLOT
C	**	LINEAR Y AXIS NUMERICS	**	SLPLOT
		DY = YL/FLOAT(NY)		SLPLOT
		EY = YO		SLPLOT
		DD = (YN-YO)/FLOAT(NY)		SLPLOT
		ND = 0.99 - ALOG10(ABS(DD))		SLPLOT
		IF (ND.LE.0) ND = -1		SLPLOT
		IY = 0		SLPLOT
		XC = XP - 1.0*HN		SLPLOT
21		AY = ABS(EY)		SLPLOT
		NF = 0		SLPLOT
		IF (AY.GE.10.0) NF = ALOG10(AY)		SLPLOT
		NS = 0		SLPLOT
		IF (EY.LT.0.0) NS = 1		SLPLOT

	SP = NS+NF+2+ND	SLPLOT
	YC = YP - 0.5*SP*WN	SLPLOT
	CALL NUMBER (XC, YC, HN, EY, 90.0, ND)	SLPLOT
	YP = YP + DY	SLPLOT
	EY = EY + DD	SLPLOT
	IY = IY + 1	SLPLOT
	IF (ABS(EY).GT.ABS(0.1*DD)) GO TO 19	SLPLOT
	IF (IY.GT.NY) GO TO 22	SLPLOT
	CALL PLOT (XP+XL, YP, 3)	SLPLOT
	CALL PLOT (XP, YP, 2)	SLPLOT
19	IF (IY.LE.NY) GO TO 21	SLPLOT
22	IF (.NOT.NYNEG) GO TO 24	SLPLOT
C	** LOG Y AXIS **	SLPLOT
	CALL LOGAXS (XP, YL, -90.0, -NY, -YL)	SLPLOT
	YB = YL/ALOG(10.0**(-NY))	SLPLOT
	YA = -YB*ALOG(YO)	SLPLOT
C	** LOG Y AXIS NUMERICS **	SLPLOT
	DY = YL/FLOAT(-NY)	SLPLOT
	EY = ALOG10(YO)	SLPLOT
	IY = 0	SLPLOT
23	CALL NUMBER (XP-1.0*HN, YP-1.0*WN, HN, 10.0, 90.0, -1)	SLPLOT
	CALL NUMBER (XP-1.5*HN, YP+1.0*WN, HN, EY, 90.0, -1)	SLPLOT
	YP = YP + DY	SLPLOT
	EY = EY + 1.0	SLPLOT
	IY = IY - 1	SLPLOT
	IF (IY.GE.NY) GO TO 23	SLPLOT
24	IF (NYLB.LE.0) GO TO 25	SLPLOT
C	** Y AXIS LABEL **	SLPLOT
	XPY = XP-4.0*HN	SLPLOT
	YPY = (YL-FLOAT(NYLB)*WL)/2.0	SLPLOT
	CALL SYMBOL(XPY, YPY, HL, YLAB, 90.0, NYLB)	SLPLOT
25	CONTINUE	SLPLOT
C	** PLOT DATA ARRAYS **	SLPLOT
	NSYM = 24	SLPLOT
	IS = NPTS/NSYM	SLPLOT
	IF (IS.EQ.0) IS = 1	VARTTH
	XOMIN = XO/1000.0	SLPLOT
	YOMIN = YO/1000.0	SLPLOT
	DO 40 J=1, NYY	SLPLOT
	IPEN = 3	SLPLOT
	DO 39 I=1, NPTS	SLPLOT
	X1 = X2	SLPLOT
	Y1 = Y2	SLPLOT
	IF (NXPOS) X2 = XB*(X(I) -XO)	SLPLOT
	IF (NYPOS) Y2 = YB*(Y(I,J)-YO)	SLPLOT
	IF (NXNEG) X2 = XA + XB*ALOG(AMAX1(X(I), XOMIN))	SLPLOT
	IF (NYNEG) Y2 = YA + YB*ALOG(AMAX1(Y(I,J), YOMIN,))	SLPLOT
	IF (Y2.LT.0.0 .OR. Y2.GT.YL) GO TO 33	SLPLOT
	IF (X2.LT.0.0 .OR. X2.GT.XL) GO TO 33	SLPLOT
	IF (IPEN.EQ.3) GO TO 33	SLPLOT
	CALL PLOT (X2, Y2, IPEN)	SLPLOT
C	** PLOT NYSM SYMBOLS **	SLPLOT

	IF (NYY.EQ.1 .OR. MOD(I,IS).NE.0) GO TO 39	SLPLOT
	IF (MOD((I/IS)-1,NYY)+1.EQ.J)	80386
	* CALL SYMBOL (X2,Y2,0.14,CHAR(J),0.0,-2)	80386
	GO TO 39	SLPLOT
33	IF (I.EQ.1) GO TO 39	SLPLOT
	DX = X2 - X1	SLPLOT
	IF (DX.NE.0.0) GO TO 34	SLPLOT
	AX0 = 1.0	SLPLOT
	AXL = 0.0	SLPLOT
	IF (X1.GE.0.0) AX0 = 0.0	SLPLOT
	IF (X1.LE.XL) AXL = 1.0	SLPLOT
	GO TO 35	SLPLOT
34	AX0 = -X1 /DX	SLPLOT
	AXL = (XL-X1)/DX	SLPLOT
35	AX1 = AMIN1(AX0,AXL)	SLPLOT
	AX2 = AMAX1(AX0,AXL)	SLPLOT
	DY = Y2 - Y1	SLPLOT
	IF (DY.NE.0.0) GO TO 36	SLPLOT
	AY0 = 1.0	SLPLOT
	AYL = 0.0	SLPLOT
	IF (Y1.GE.0.0) AYO = 0.0	SLPLOT
	IF (Y1.LE.YL) AYL = 1.0	SLPLOT
	GO TO 37	SLPLOT
36	AY0 = -Y1 /DY	SLPLOT
	AYL = (YL-Y1)/DY	SLPLOT
37	AY1 = AMIN1(AY0,AYL)	SLPLOT
	AY2 = AMAX1(AY0,AYL)	SLPLOT
	A1 = AMAX1(AX1,AY1,0.0)	SLPLOT
	A2 = AMIN1(AX2,AY2,1.0)	SLPLOT
	IF (A1.GE.A2) GO TO 39	SLPLOT
	XP = X1 + A1*DX	SLPLOT
	YP = Y1 + A1*DY	SLPLOT
	CALL PLOT(XP,YP,IPEN)	SLPLOT
	IPEN = 2	SLPLOT
	XP = X1 + A2*DX	SLPLOT
	YP = Y1 + A2*DY	SLPLOT
	CALL PLOT(XP,YP,IPEN)	SLPLOT
	IF (A2.NE.1.0) IPEN = 3	SLPLOT
39	CONTINUE	SLPLOT
40	CONTINUE	SLPLOT
	RETURN	SLPLOT
	END	SLPLOT

C	SUBROUTINE SOLVA(R,AA11,AA22,AA12)	SOLVA
	IMPLICIT REAL*8 (A-H,O-Z)	REV III.2 08/08/84REVIII SOLVA
	DIMENSION R(2,3)	SOLVA
	A11=R(1,1)**2	SOLVA
	A12=2.0*R(2,1)*R(1,1)	SOLVA
	A13=R(2,1)**2	SOLVA
	A21=R(1,2)**2	SOLVA
	A22=2.0*R(2,2)*R(1,2)	SOLVA
	A23=R(2,2)**2	SOLVA
	A31=R(1,3)**2	SOLVA
	A32=2.0*R(2,3)*R(1,3)	SOLVA
	A33=R(2,3)**2	SOLVA
	DEL=A11*(A22*A33-A23*A32)-A12*(A21*A33-A23*A31)+	SOLVA
	* A13*(A21*A32-A22*A31)	SOLVA
	AA11=((A22-A12)*(A33-A23)-(A23-A13)*(A32-A22))/DEL	SOLVA
	AA12=((A23-A13)*(A31-A21)-(A21-A11)*(A33-A23))/DEL	SOLVA
	AA22=((A21-A11)*(A32-A22)-(A22-A12)*(A31-A21))/DEL	SOLVA
	RETURN	SOLVA
	END	SOLVA

```

SUBROUTINE SOLVR(A1,A2,A3,A4,A5,A6,A7,A8,P,RX,RZ)          SOLVR
C                                                            REV III.2 08/08/84REVI III
IMPLICIT REAL*8 (A-H,O-Z)                                SOLVR
C                                                            SOLVR
C *****                                                    SOLVR
C                                                            SOLVR
C THIS SUBROUTINE WILL SOLVE A SET OF SIMULTANEOUS EQUATIONS SOLVR
C TO FIND COMPONENTS OF VECTOR R THAT SATISFY THE PROPERTIES NEEDED SOLVR
C TO DETERMINE THE EQUATION OF THE PROJECTED ELLIPSE.      SOLVR
C                                                            SOLVR
C SEE WRITEUP.                                            SOLVR
C                                                            SOLVR
C *****                                                    SOLVR
DIMENSION P(3)                                           SOLVR
B=A1*P(1)+A2*P(2)+A3*P(3)                                 SOLVR
D=A4*P(1)+A5*P(2)+A6*P(3)                                 SOLVR
T1=A7*(D/B)**2+A6-2.0*A8*D/B                               SOLVR
T2=2.0*A7*D/(B)**2-2.0*A8/B                               SOLVR
T3=A7*(1/B)**2-1                                          SOLVR
RX=(-T2+DSQRT(T2**2-4.0*T1*T3))/(2.0*T1)                 SOLVR
RZ=-D*RZ/B-1.0/B                                         SOLVR
RETURN                                                    SOLVR
END                                                       SOLVR

```


	SUBROUTINE SPDAMP	REV IV 07/24/86SLIP	SPDAMP
C			
C	COMPUTES THE SPRING AND VISCOUS FORCE OF A SPRING DAMPER BETWEEN		SPDAMP
C	SPECIFIED POINTS ON SELECTED SEGMENTS AND ADDS THE RESULTING		SPDAMP
C	FORCE AND TORQUE TO THE U1 AND U2 ARRAYS.		SPDAMP
C			SPDAMP
	IMPLICIT REAL*8(A-H,O-Z)		SPDAMP
	COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,		SPDAMP
*	NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG		PAGE
	COMMON/SCMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),		SPDAMP
*	SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)		SPDAMP
	COMMON/DAMPER/ APSDM(3,20),APSDN(3,20),ASD(5,20),MSDM(20),MSDN(20)		SPDAMP
	COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(1250),TAB(4500)		BUTLER2
	COMMON/FORCES/PSF(7,70),BSF(4,20),SSF(10,40),BAGSF(3,20),		NGFORC
*	PRJNT(7,30),NPANEL(5),NPSF,NBSF,NSSF,NBSGF		SPDAMP
	COMMON/TEMPVS/DELM(3),DELN(3),DD(3),DEL,T1(3),T2(3),T3(3),T4(3),		SPDAMP
*	DUNIT(3),DV(3),DMV,DDO,FS,FD,TOTF(3),		SPDAMP
*	T5(3),T6(3),T7(3),T8(3),DMY(35066)		80386
	CALL ELTIME(1,32)		SPDAMP
	NBSFO = NBSF		SPDAMP
	DO 90 I=1,NSD		SPDAMP
	M = MSDM(I)		SPDAMP
	N = MSDN(I)		SPDAMP
C			SPDAMP
C	COMPUTE VECTOR AND ITS MAGNITUDE BETWEEN THE SPECIFIED POINTS.		SPDAMP
C			SPDAMP
	CALL DOT31 (D(1,1,M),APSDM(1,I),DELM)		SPDAMP
	CALL DOT31 (D(1,1,N),APSDN(1,I),DELN)		SPDAMP
	DEL = 0.0		SPDAMP
	DO 10 K=1,3		SPDAMP
	DD(K) = SEGLP(K,M)+DELM(K)-SEGLP(K,N)-DELN(K)		SPDAMP
10	DEL = DEL+DD(K)**2		SPDAMP
	IF (DEL.LE.0.0) GO TO 90		SPDAMP
	DEL = DSQRT(DEL)		SPDAMP
C			SPDAMP
C	COMPUTE RELATIVE VELOCITY AND ITS COMPONENT ON VECTOR LINE.		SPDAMP
C			SPDAMP
	CALL CROSS(WMEG(1,M),APSDM(1,I),T1)		SPDAMP
	CALL CROSS(WMEG(1,N),APSDN(1,I),T2)		SPDAMP
	CALL DOT31 (D(1,1,M),T1,T3)		SPDAMP
	CALL DOT31 (D(1,1,N),T2,T4)		SPDAMP
	DO 20 K=1,3		SPDAMP
	DUNIT(K) = DD(K)/DEL		SPDAMP
20	DV(K) = SEGLV(K,M)+T3(K)-SEGLV(K,N)-T4(K)		SPDAMP
	DMV = DUNIT(1)*DV(1)+DUNIT(2)*DV(2)+DUNIT(3)*DV(3)		SPDAMP
C			SPDAMP
C	COMPUTE SPRING AND VISCOUS FORCE AND THE COMPONENTS		SPDAMP
C	ALONG THE UNIT VECTOR		SPDAMP
C			SPDAMP
	FS = 0.0		SPDAMP
	FD = 0.0		SPDAMP
	IF (ASD(1,I).LT.0.0) GO TO 21		SLIP

DDO = DEL-ASD(1,I)	SPDAMP
IF (DDO.LE.0.0 .AND. ASD(2,I).LE.0.0) GO TO 41	SPDAMP
FS = DDO*(DABS(ASD(2,I)) + DABS(DDO)*ASD(3,I))	SPDAMP
FD = DMV*(ASD(4,I)+DABS(DMV)*ASD(5,I))	SPDAMP
GO TO 29	SPDAMP
21 DDO = DEL+ASD(1,I)	SPDAMP
JF1 = ASD(2,I)	SPDAMP
IF (JF1.EQ.0) GO TO 22	SPDAMP
JF2 = NTI(JF1)	SPDAMP
IF (DDO.GT.0.0 .OR. ASD(3,I).EQ.0.0) FS = EVALFD(DDO,JF2,1)	SPDAMP
22 JF3 = ASD(4,I)	SPDAMP
IF (JF3.EQ.0) GO TO 29	SPDAMP
JF4 = NTI(JF3)	SPDAMP
IF (DDO.GT.0.0 .OR. ASD(3,I).EQ.0.0) FD = EVALFD(DMV,JF4,1)	SLIP
29 DO 30 K=1,3	SPDAMP
30 TOTF(K) = (FS+FD)*DUNIT(K)	SPDAMP
C	SPDAMP
C AND ADD THE RESULTING FORCE AND TORQUE TO THE U1 AND U2 ARRAYS.	SPDAMP
C	SPDAMP
CALL MAT31(D(1,1,M),TOTF,T5)	SPDAMP
CALL MAT31(D(1,1,N),TOTF,T6)	SPDAMP
CALL CROSS(APSDM(1,I),T5,T7)	SPDAMP
CALL CROSS(APSDN(1,I),T6,T8)	SPDAMP
DO 40 K=1,3	SPDAMP
U1(K,M) = U1(K,M) - TOTF(K)	SPDAMP
U1(K,N) = U1(K,N) + TOTF(K)	SPDAMP
U2(K,M) = U2(K,M) - T7(K)	SPDAMP
40 U2(K,N) = U2(K,N) + T8(K)	SPDAMP
41 IBSF = 3-2*MOD(I,2)	SPDAMP
NBSF = NBSFO + (I+1)/2	SPDAMP
BSF(IBSF ,NBSF) = DEL	SPDAMP
BSF(IBSF+1,NBSF) = FD + FS	SPDAMP
90 CONTINUE	SPDAMP
CALL ELTIME(2,32)	SPDAMP
RETURN	SPDAMP
END	SPDAMP

```

SUBROUTINE SPLINE (X,Y,F,N,L)                                       SPLINE
C                                                                 REV 19   05/14/79SPLINE
C                                                                 SPLINE
C ROUTINE TO FIT A SET OF POLYNOMIALS OF DEGREE L                 SPLINE
C TO A SET OF GIVEN DATA POINTS (X(I),Y(I),I=1,N)               SPLINE
C                                                                 SPLINE
C FUNCTION IS OF FORM:                                           SPLINE
C                                                                 SPLINE
C   Y = F(2,K) + F(3,K)*DX + F(4,K)*DX**2 + F(5,K)*DX**3        SPLINE
C                                                                 SPLINE
C WHERE: DX = XX - F(1,K)                                         SPLINE
C       F(1,K) .LE. XX .LT. F(1,K+1) ; (SETS K)                 SPLINE
C       IF (XX.GT.F(1,N)) ; USE K=N, CONSTANT FIT TO Y(N)      SPLINE
C       IF (XX.LT.F(1,1)) ; EXTRAPOLATED FIT FOR K=1           SPLINE
C                                                                 SPLINE
C       F(1,I) = X(I) ,                                           I=1,N                 SPLINE
C       F(2,I) = Y(I) ,                                           I=1,N                 SPLINE
C                                                                 SPLINE
C   DEGREE L                                                       CONTINUITY                       SPLINE
C     0 F(3,I) = F(4,I) = F(5,I) = 0 , I=1,N                       NONE                    SPLINE
C     1 F(4,I) = F(5,I) = 0 , I=1,N                               Y                        SPLINE
C     2 F(5,I) = 0 , I=1,N                                         Y,Y'                      SPLINE
C     3 CUBIC SPLINE                                               Y,Y',Y''                 SPLINE
C                                                                 SPLINE
C       F(K,N)=0 FOR K=3,5 IN ALL CASES                          SPLINE
C                                                                 SPLINE
C FOR L=2 AND L=3 THE CHANGES IN THE L'TH DERIVATIVES ARE MINIMIZED SPLINE
C                                                                 SPLINE
C SPECIAL CASES:                                                 SPLINE
C   N=1 ; TREATED AS L=0                                         SPLINE
C   N=2 ; TREATED AS L=MIN(L,1)                                  SPLINE
C   L<0 ; TREATED AS L=0                                         SPLINE
C   L>3 ; TREATED AS L=3                                         SPLINE
C                                                                 SPLINE
C STORAGE REQUIRED X(N),Y(N),F(5,N); SET BY CALLING PROGRAM     SPLINE
C                                                                 SPLINE
C USAGE:                                                         SPLINE
C   ALL COMPUTATIONS AND REAL VARIABLES ARE DOUBLE PRECISION   SPLINE
C   GIVEN: L,N, (X(I),Y(I),I=1,N)                               SPLINE
C   CALL SPLINE (X,Y,F,N,L) ; SETS F                            SPLINE
C                                                                 SPLINE
C TO EVALUATE FUNCTION AND DERIVATIVES AT POINT XX             SPLINE
C                                                                 SPLINE
C   DO 10 K=1,N                                                  SPLINE
C   IF (K.EQ.N) GO TO 11                                         SPLINE
C   IF (XX.LT.F(1,K+1)) GO TO 11                                  SPLINE
C   10 CONTINUE                                                  SPLINE
C   11 DX = XX - F(1,K)                                           SPLINE
C       YY = F(2,K) + DX*(F(3,K)+DX*(F(4,K)+DX*(F(5,K))))        SPLINE
C       YD = F(3,K) + DX*(2.0*F(4,K)+3.0*DX*(F(5,K)))            SPLINE
C       YDD = 2.0*F(4,K) + 6.0*DX*(F(5,K))                        SPLINE

```

C	YDED = 6.0*F(5,K)	SPLINE
C	YDDDD = 0.0	SPLINE
CC		SPLINE
CC	FUNCTIONAL VALUE IN YY, DERIVATIVES IN YD'S	SPLINE
CC	REPEAT FOR NEXT VALUE OF XX	SPLINE
C		SPLINE
C	AUTHOR: DR. JOHN T. FLECK	SPLINE
C		SPLINE
	IMPLICIT REAL*8 (A-H,O-Z)	SPLINE
	DIMENSION X(N),Y(N),F(5,N),C(2,3)	SPLINE
	DO 20 I=1,N	SPLINE
	F(1,I) = X(I)	SPLINE
	DO 10 K=2,5	SPLINE
10	F(K,I) = 0.0	SPLINE
	IF (L.LT.3) F(2,I) = Y(I)	SPLINE
20	IF (L.GT.0 .AND. I.LT.N) F(3,I) = (Y(I+1)-Y(I))/(X(I+1)-X(I))	SPLINE
	IF (L.LT.2 .OR. N.LT.3) GO TO 99	SPLINE
	IF (L.GE.3) GO TO 50	SPLINE
	D1 = X(2) - X(1)	SPLINE
	SS = 0.0	SPLINE
	DS = 0.0	SPLINE
	DO 30 I=3,N	SPLINE
	F(4,I-1) = F(3,I-1) - F(3,I-2) - F(4,I-2)	SPLINE
	DX1 = X(I) - X(I-1)	SPLINE
	DX2 = X(I-1) - X(I-2)	SPLINE
	DD = D1/DX1 + D1/DX2	SPLINE
	SS = SS + DD*DD	SPLINE
	DS = DS + DD*(F(4,I-1)/DX1 - F(4,I-2)/DX2)	SPLINE
30	D1 = -D1	SPLINE
	F(4,1) = DS/SS	SPLINE
	DX = (X(2)-X(1))*F(4,1)	SPLINE
	F(3,1) = F(3,1) - DX	SPLINE
	DO 40 I=3,N	SPLINE
	XX = F(4,I-1) - DX	SPLINE
	F(3,I-1) = F(3,I-1) - XX	SPLINE
	F(4,I-1) = XX/(X(I)-X(I-1))	SPLINE
40	DX = -DX	SPLINE
	GO TO 99	SPLINE
C		SPLINE
C	CUBIC SPLINE	SPLINE
C		SPLINE
50	DO 51 I=2,N	SPLINE
	IF (I.EQ.N) GO TO 51	SPLINE
	F(4,I) = 3.0*(F(3,I)-F(3,I-1))	SPLINE
	F(5,I) = 2.0*(X(I+1)-X(I-1))	SPLINE
51	F(3,I-1) = 0.0	SPLINE
	F(2,N) = -1.0	SPLINE
	F(3,1) = -1.0	SPLINE
	DO 60 I=3,N	SPLINE
	DX = X(I-1) - X(I-2)	SPLINE
	IF (I.GT.3) DX = DX/F(5,I-2)	SPLINE
	DO 60 K=3,5	SPLINE

```

60 F(K,I-1) = F(K,I-1) - F(K,I-2)*DX**((K-1)/2)          SPLINE
   DO 70 I=3,N                                             SPLINE
   NI = N-I                                               SPLINE
   DX = X(NI+3) - X(NI+2)                                 SPLINE
   DO 70 K=2,4                                           SPLINE
70 F(K,NI+2) = (F(K,NI+2) - DX*F(K,NI+3))/F(5,NI+2)     SPLINE
   DO 71 J=1,2                                           SPLINE
   DO 71 K=J,3                                           SPLINE
   C(J,K) = 0.0                                          SPLINE
   DO 71 I=3,N                                           SPLINE
   DX1 = X(I) - X(I-1)                                   SPLINE
   DX2 = X(I-1) - X(I-2)                                SPLINE
71 C(J,K) = C(J,K) + ( (F(J+1,I) - F(J+1,I-1))/DX1     SPLINE
*                      - (F(J+1,I-1) - F(J+1,I-2))/DX2) SPLINE
*                      * ( (F(K+1,I) - F(K+1,I-1))/DX1   SPLINE
*                      - (F(K+1,I-1) - F(K+1,I-2))/DX2) SPLINE
   DEN = C(1,1)*C(2,2) - C(1,2)*C(1,2)                 SPLINE
   F(4,1) = (C(1,1)*C(2,3) - C(1,2)*C(1,3))/DEN        SPLINE
   F(4,N) = (C(2,2)*C(1,3) - C(1,2)*C(2,3))/DEN        SPLINE
   DO 72 I=3,N                                           SPLINE
72 F(4,I-1) = F(4,I-1) - F(4,1)*F(3,I-1) - F(4,N)*F(2,I-1) SPLINE
   D1 = X(2) - X(1)                                       SPLINE
   F(3,1) = (Y(2)-Y(1))/D1 - (2.0*F(4,1)+F(4,2))*D1/3.0 SPLINE
   F(2,1) = Y(1)                                          SPLINE
   DO 80 I=2,N                                           SPLINE
   F(2,I) = Y(I)                                          SPLINE
   DX = X(I) - X(I-1)                                   SPLINE
   IF (I.LT.N) F(3,I) = F(3,I-1) + (F(4,I)+F(4,I-1))*DX SPLINE
80 F(5,I-1) = (F(4,I)-F(4,I-1))/(3.0*DX)               SPLINE
   F(4,N) = 0.0                                          SPLINE
99 RETURN                                               SPLINE
   END                                                    SPLINE

```

```

DOUBLE PRECISION FUNCTION SPRNGF(T,D,ZD,SPR,JSTOP)          SPRNGF
C                                                     REV IV    07/23/86TWOPI
C COMPUTES NONLINEAR SPRING TORQUE FOR JOINTS AS A FUNCTION OF ANGLESPRNGF
C ACTUALLY ROUTINE RETURNS TORQUE/ABS(SIN THETA)          SPRNGF
C                                                     SPRNGF
C ARGUMENTS:                                             SPRNGF
C     T      : COS THETA WHERE THETA IS ANGLE OF JOINT (0<THETA<PI) SPRNGF
C     D      : ABS(SIN THETA)                             SPRNGF
C     ZD     : -THETA DOT * SIN THETA                    SPRNGF
C     SPR    : ARRAY OF 5 VALUES DESCRIBING FUNCTION EVALUATION SPRNGF
C     JSTOP  : INDICATOR TO BE SET TO ONE IF JOINT IS IN STOP SPRNGF
C                                                     SPRNGF
IMPLICIT REAL*8 (A-H,O-Z)                                SPRNGF
COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),              SPRNGF
*          UNITL,UNITM,UNITT,GRAVITY(3),TWOPI          SPRNGF
DIMENSION SPR(5)                                         SPRNGF
C                                                     SPRNGF
C     RESET T=1 IF T>1 (HAD & HBD IN VISPR)            SPRNGF
C                                                     SPRNGF
C                                                     SPRNGF
IF (T.GT.1.0) T = 1.0                                    SPRNGF
IF (T.LT.-1.0) T = -1.0                                  SPRNGF
Z = DACOS(T)                                             SPRNGF
U = EPS(1)*D                                             SPRNGF
Q = 0.0                                                  SPRNGF
IF (D.NE.0.0) Q = -ZD/U                                  SPRNGF
IF (Q.GT.1.0) Q = 1.0                                    SPRNGF
IF (Q.LT.-1.0) Q = -1.0                                  SPRNGF
X = 0.5*(1.0+SPR(4) + Q*(1.0-SPR(4))) )                 SPRNGF
Y = 0.0                                                  SPRNGF
IF (D.NE.0.0) Y = Z/D                                    SPRNGF
Q = 1.0                                                  SPRNGF
IF (DABS(Z).LT.EPS(4)) Y = DSIGN(Q,Z)                   SPRNGF
SPRNGF = Y*SPR(1)                                        SPRNGF
JSTOP = 0                                                SPRNGF
IF (SPR(5).GT.0.0) GO TO 10                              SPRNGF
SPRNGF = X*SPRNGF                                        SPRNGF
GO TO 11                                                 SPRNGF
10 IF (Z.LT.SPR(5)) GO TO 11                              SPRNGF
JSTOP = 1                                                SPRNGF
Z = Z-SPR(5)                                             SPRNGF
SPRNGF = SPRNGF + X/D*(SPR(2)+Z*SPR(3))*Z**2           SPRNGF
11 CONTINUE                                             SPRNGF
RETURN                                                  SPRNGF
END                                                    SPRNGF

```

C

SUBROUTINE TRIGFS	REV 19	08/05/78	TRIGFS
IMPLICIT REAL*8 (A-H,O-Z)			TRIGFS
COMMON/CDINT/ UU(4),GH(3,4),			TRIGFS
* E(3,240), F(5,240),GG(5,240),Y(5,240),U(5,240),			TRIGFS
* H,HPRINT,HS,TPRINT,TSTART,ICNT,IDBL,IFLAG,IDMMY			80386
BETA = 0.0			TRIGFS
IF (HS.NE.0.0) BETA = (H/HS)**2			TRIGFS
R1 = HS/H			TRIGFS
R2 = 1.0+BETA*R1			TRIGFS
GH(3,1) = 2.0/(H*R2)			TRIGFS
GH(2,1) = GH(3,1)*(BETA-1.0)			TRIGFS
GH(1,1) = GH(3,1)* BETA			TRIGFS
GH(1,2) = 4.0*BETA/(R2*H**2)			TRIGFS
GH(3,2) = GH(1,2)* R1			TRIGFS
GH(2,2) = GH(1,2)*(R1+1.0)			TRIGFS
GH(3,3) = 1.0/H			TRIGFS
GH(2,3) = 4.0*GH(3,3)			TRIGFS
GH(1,3) = 3.0*GH(3,3)			TRIGFS
GH(3,4) = 2.0/H**2			TRIGFS
GH(2,4) = 2.0*GH(3,4)			TRIGFS
GH(1,4) = GH(3,4)			TRIGFS
UU(1) = 2.0/H			TRIGFS
UU(2) = 0.0			TRIGFS
UU(3) = 0.5*H			TRIGFS
UU(4) = 0.25*H**2			TRIGFS
IF (HS.EQ.0.0) GO TO 99			TRIGFS
UU(1) = BETA*(4.25+2.25/R1)			TRIGFS
UU(2) = BETA*(2.25+1.25/R1)/R1			TRIGFS
UAU = 1.0+UU(1)+UU(2)			TRIGFS
UU(1) = 2.0*UU(1)/(UAU*H)			TRIGFS
UU(2) = 4.0*UU(2)/(UAU*H**2)			TRIGFS
99 RETURN			TRIGFS
END			TRIGFS

	SUBROUTINE UNIT1(IND)	UNIT1
		REV IV 02/20/87HYPER
C		
C	THIS SUBROUTINE REPLACES THE PROGRAM CODE THAT PREVIOUSLY WAS	UNIT1
C	NEAR THE END OF THE MAIN PROGRAM TO WRITE ON UNIT 1 THAT DATA	UNIT1
C	USED FOR VARIOUS PLOTTING PROGRAMS (E.G. BUBBLE MAN PLOT).	UNIT1
C		UNIT1
C	THIS SUBROUTINE IS WRITTEN TO GENERATE UNIT 1 IN SUCH A MANNER	UNIT1
C	TO BE COMPATIBLE WITH THE INPUT REQUIREMENTS FOR THE AAMRL VIEW	UNIT1
C	PROGRAM THAT IS NOW BEING DISTRIBUTED ON THE CVS PROGRAM TAPES.	UNIT1
C		UNIT1
C	ARGUMENTS:	UNIT1
C	IND = 0: CALL IS FROM THE MAIN PROGRAM	UNIT1
C	# 0: CALL IS FROM SUBROUTINE EQUILB	UNIT1
C		UNIT1
C	IMPLICIT REAL*8 (A-H,O-Z)	UNIT1
C	COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,	UNIT1
*	NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG	PAGE
C	COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),	UNIT1
*	SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)	UNIT1
C	COMMON/CNTRSF/ PL(24,30),BELT(20,8),TPTS(6,8),BD(24,40)	EDGE
C	COMMON/JBARTZ/ MNPL(30),MNBLT(8),MNSEG(30),MNBAG(6),	UNIT1
*	MPL(3,5,30),MBLT(3,5,8),MSEG(3,5,30),MBAG(3,10,6),	UNIT1
*	NTPL(5,30),NTBLT(5,8),NTSEG(5,30)	UNIT1
C	COMMON/RSAVE/ XSG(3,20,3),DPMI(3,3,30),LPMI(30),	UNIT1
*	NSG(9),MSG(20,9),MCG,MCGIN(24,5),KREF(20,9)	TTHKREF
C	COMMON/TEMPVS/ XD(3,3,30),XSEGLP(3,30),XPL(17,30),XBD(24,40),	UNIT1
*	T1(3),T3(3,3),DMMY(34186)	80386
C	REAL XTIME,XD,XSEGLP,XPL,XBD	UNIT1
C	DATA IFIRST/0/	UNIT1
C	IF (NPRT(1).EQ.0) GO TO 99	UNIT1
C	IF (IFIRST.NE.0) GO TO 20	UNIT1
C	IFIRST = 1	UNIT1
C		UNIT1
C	FIRST TIME IN ROUTINE, WRITE STATIC DATA ON OUTPUT UNIT 1.	UNIT1
C	DATA MUST BE CONVERTED TO SINGLE PRECISION FOR VIEW PROGRAM.	UNIT1
C		UNIT1
C	DO 11 J=1,30	UNIT1
C	DO 11 I=1,17	FIXWBS
C	11 XPL(I,J) = PL(I,J)	FIXWBS
C	DO 12 J=1,40	UNIT1
C	K = 1	HYPER
C	IF (BD(1,J).LT.0.0) K = 2	HYPER
C	DO 12 I=1,24	UNIT1
C	XBD(I,J) = BD(K,J)	HYPER
C	12 K = K + 1	HYPER
C	WRITE (1) NSEG,NPL,XPL,XBD,MPL	UNIT1
C	GOTO 99	EDGE
C		UNIT1
C	WRITE TIME POINT DATA ON OUTPUT UNIT 1.	UNIT1
C	DATA MUST BE CONVERTED TO SINGLE PRECISION FOR VIEW PROGRAM.	UNIT1
C		UNIT1
C	20 XTIME = TIME	UNIT1

DO 22 K=1,30	UNIT1
DO 22 J=1,3	UNIT1
DO 21 I=1,3	UNIT1
21 XD(I,J,K) = D(I,J,K)	UNIT1
22 XSEGLP(J,K) = SEGLP(J,K)	UNIT1
DO 25 K=1,NSEG	UNIT1
IF (LPMI(K).EQ.0) GO TO 25	UNIT1
CALL DOT33 (DPMI(1,1,K),D(1,1,K),T3)	UNIT1
DO 24 I=1,3	UNIT1
DO 24 J=1,3	UNIT1
24 XD(I,J,K) = T3(I,J)	UNIT1
25 CONTINUE	UNIT1
WRITE (1) XTIME,XSEGLP,XD	UNIT1
99 RETURN	UNIT1
END	UNIT1

```

SUBROUTINE UPDATE(I)                                UPDATE
C                                                     REV IV   07/24/86SLIP
C CALLED BY SUBROUTINE DINT                          UPDATE
C                                                     UPDATE
C   (I=1) AT THE START OF A NEW STEP TO SETUP ANY NEW CONDITIONS UPDATE
C         TO BE VALID FOR ENTIRE INTEGRATION STEP    UPDATE
C           A. UPDATE FORCE DEFLECTION FUNCTIONS(SUBROUTINE UPDFDC)UPDATE
C           B. TEST FOR LOCKED JOINTS                UPDATE
C   NOTE: ARGUMENT I WILL BE SET TO -1 TO RESET INTEGRATOR. UPDATE
C                                                     UPDATE
C   (I=2) AT THE END OF EACH SUCCESSFUL INTEGRATION STEP TO UPDATE
C         COMPLETE CALCULATIONS FOR OUTPUT (SUBROUTINE AIRBG3). UPDATE
C                                                     UPDATE
      IMPLICIT REAL*8(A-H,O-Z)                       UPDATE
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND, UPDATE
*          NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG    PAGE
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30), UPDATE
*          SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)      UPDATE
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60), SLIP
*          RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),    UPDATE
*          JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)  UPDATE
      COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60), UPDATE
*          F(3,30),TQ(3,30),WJ(30),A11(3,3,30)              'IIP
      COMMON/JBARTZ/ MNPL( 30),MNBLT( 8),MNSEG( 30),MNBAG( 6)   ATE
*          MPL(3,5,30),MBLT(3,5,8),MSEG(3,5,30),MBAG(3,10,6), UPDATE
*          NTPL( 5,30),NTBLT( 5,8),NTSEG( 5,30)             UPDATE
      COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(1250),TAB(4500)UPDATE
      COMMON/FORCES/PSF(7,70),BSF(4,20),SSF(10,40),BAGSF(3,20), NCFORC
*          PRJNT(7,30),NPANEL(5),NPSF,NBSF,NSSF,NBGSF      UPDATE
      COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24), UPDATE
*          HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12), UPDATE
*          RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),           UPDATE
*          KQ1(12),KQ2(12),KQTYPE(12)                     UPDATE
      COMMON/TEMPVI/ CREST,TTI(3),R1I(3),R2I(3),JSTOP(4,2,30) UPDATE
      COMMON/CEULER/ IEULER(30),HIR(3,3,90),ANG(3,30),ANGD(3,30), JDRIFT
*          FE(3,30),TQE(3,30),CONST(5,30)                 JDRIFT
      COMMON/HRNESS/ BAR(15,100),BB(100),BBDOT(100),PLOSS(2,100), UPDATE
*          XLONG(20),HTIME(2),IBAR(5,100),NL(2,100),       UPDATE
*          NPTSPB(20),NPTPLY(20),NTHRNS(20),NBLTPH(5)     UPDATE
      DIMENSION TQTEST(3),LOCK(8,3) ,T(3)                UPDATE
      DATA LOCK,-8, 6, 5, 7,-3,-2,-4, 1,               UPDATE
*          6,-8, 4,-3, 7,-1,-5, 2,                       UPDATE
*          5, 4,-8,-2,-1, 7,-6, 3/                       UPDATE
C                                                     UPDATE
C   CALL AIRBG3 FOR AIRBAG, IF ANY.                   UPDATE
C                                                     UPDATE
      IF (NBAG.NE.0) CALL AIRBG3(1)                     UPDATE
      IF (I.EQ.2) GO TO 42                              UPDATE
      CALL ELTIME (1,7)                                 UPDATE
      IF (NPL.LE.0) GO TO 13                            UPDATE
C                                                     UPDATE
C   CALL UPDFDC FOR EACH ALLOWED PLANE-SEGMENT CONTACT. UPDATE

```

C	NPSF = 0	UPDATE
	DO 12 J=1,NPL	UPDATE
	NK = MNPL(J)	UPDATE
	IF (NK.LE.0) GO TO 12	UPDATE
	DO 11 K = 1, NK	UPDATE
	NPSF = NPSF+1	UPDATE
	NT = NTPL(K,J)	UPDATE
	NF = NTAB(NT+5)	UPDATE
	CALL UPDFDC(NT)	UPDATE
	IF (NT.GT.0.OR.TAB(NF+3).EQ.0.0) GO TO 11	UPDATE
	CALL IMPULS(1,K,J)	UPDATE
	I = -1	UPDATE
	11 CONTINUE	UPDATE
	12 CONTINUE	UPDATE
	13 IF (NBLT.LE.0) GO TO 16	UPDATE
C		UPDATE
C	CALL UPDFDC FOR EACH ALLOWED BELT-SEGMENT CONTACT.	UPDATE
C		UPDATE
	DO 15 J=1,NBLT	UPDATE
	NK = MNBLT(J)	UPDATE
	IF (NK.LE.0) GO TO 15	UPDATE
	DO 14 K = 1,NK	UPDATE
	NT = NTBLT(K,J)	UPDATE
	NF = NTAB(NT+5)	UPDATE
	NT6 = NT+6	UPDATE
	CALL UPDFDC(NT)	UPDATE
C		UPDATE
C	AND FOR 2ND FUNCTION, IF FULL BELT FRICTION.	UPDATE
C		UPDATE
	14 IF (NF.NE.0) CALL UPDFDC(NT6)	UPDATE
	15 CONTINUE	UPDATE
C		UPDATE
C	CALL UPDFDC FOR EACH ALLOWED SEGMENT-SEGMENT CONTACT.	UPDATE
C		UPDATE
	16 NSSF = 0	UPDATE
	DO 18 J=1,NSEG	UPDATE
	NK = MNSEG(J)	UPDATE
	IF (NK.LE.0) GO TO 18	UPDATE
	DO 17 K = 1,NK	UPDATE
	NSSF = NSSF+1	UPDATE
	NT = NTSEG(K,J)	UPDATE
	NF = NTAB(NT+5)	UPDATE
	CALL UPDFDC(NT)	UPDATE
	IF (NT.GT.0.OR.TAB(NF+3).EQ.0.0) GO TO 17	UPDATE
	CALL IMPULS(3,K,J)	UPDATE
	I = -1	UPDATE
	17 CONTINUE	UPDATE
	18 CONTINUE	UPDATE
	IF (NHRNSS.LE.0) GO TO 71	UPDATE
C		UPDATE
C	CALL UPDFDC FOR EACH BELT OF HARNESS-BELT SYSTEMS.	UPDATE

C	TEST TO LOCK OR UNLOCK JOINTS	UPDATE
C		UPDATE
C		UPDATE
C	CONDITIONS TO CHANGE SIGN OF IPIN(J)	UPDATE
C		UPDATE
C	PINNED UNPINNED	UPDATE
C	LOCKED (-1) !H.TQ! > T1 (-2) !TQ! > T1	UPDATE
C		UPDATE
C	UNLOCKED (+1) !H.TQ! < T2 (+2) !TQ! < T2	UPDATE
C	OR OR	UPDATE
C	WJ < T3 WJ < T3	UPDATE
C		UPDATE
	DO 28 J=1,NJNT	UPDATE
	IF (IABS(IPIN(J)).EQ.4) GO TO 28	UPDATE
	IF (IPIN(J)) 22,28,23	UPDATE
22	T1 = VISC(4,3*J-2)	UPDATE
	IF (T1.EQ.0.0) GO TO 28	UPDATE
	IF (IPIN(J).GT.-1) GOTO 51	SLIP
	IF (IPIN(J).GT.-6.AND.IPIN(J).LT.-1) GOTO 51	SLIP
	TQM = XDY(HB(1,2*J),D(1,1,J+1),TQ(1,J))	UPDATE
	ABSTQM = DABS(TQM)	UPDATE
	IF (ABSTQM.GT.T1) HA(2,2*J-1) = TQM	UPDATE
	TQM = ABSTQM	UPDATE
	GO TO 52	UPDATE
51	TQM = DSQRT(TQ(1,J)**2 + TQ(2,J)**2 + TQ(3,J)**2)	UPDATE
	IF (TQM.GT.T1) CALL DOT31(HIR(1,1,J),TQ(1,J),HA(1,2*J-1))	UPDATE
52	IF (TQM-T1) 28,28,26	UPDATE
23	T2 = VISC(5,3*J-2)	UPDATE
	IF (HA(2,2*J).NE.0.0) GO TO 54	UPDATE
	DO 53 K=1,3	UPDATE
53	HA(K,2*J-1) = 0.0	UPDATE
54	IF (T2.EQ.0.0) GO TO 24	UPDATE
	IF (IPIN(J).GE.2.AND.IPIN(J).LE.5)	SLIP
	* TQM = DSQRT(TQ(1,J)**2+TQ(2,J)**2+TQ(3,J)**2)	SLIP
	IF (IPIN(J).EQ.1.OR.IPIN(J).EQ.6.OR.IPIN(J).EQ.7)	SLIP
	* TQM = DABS(XDY(HB(1,2*J),D(1,1,J+1),TQ(1,J)))	SLIP
	IF (TQM-T2) 25,28,28	UPDATE
24	T3 = VISC(6,3*J-2)	UPDATE
	IF (T3.EQ.0.0) GO TO 28	UPDATE
	IF (WJ(J)-T3) 25,28,28	UPDATE
25	CALL IMPLS2(0,J,HB(1,2*J))	UPDATE
	I = -1	UPDATE
26	IPIN(J) = -IPIN(J)	UPDATE
	TMSEC = 1000.0*TIME	UPDATE
	IPINJ = -IPIN(J)	UPDATE
	WRITE (6,27) TMSEC,J,IPINJ,IPIN(J)	UPDATE
27	FORMAT('0 AT TIME =',F9.3,' MSEC, IPIN(',I2,	BUTLER1
	* ') HAS BEEN CHANGED FROM',I3,' TO',I3)	BUTLER1
28	CONTINUE	UPDATE
C		UPDATE
C	TEST TO LOCK OR UNLOCK EULER JOINTS AXES.	UPDATE
C	USE SAME TEST AS ABOVE BUT ON EACH AXIS SERARATELY.	UPDATE

C		UPDATE
C	IF LOCK(IEULER,K) IS NEGATIVE, AXIS K IS LOCKED;	UPDATE
C	TO UNLOCK AXIS SET IEULER TO -LOCK(IEULER,K).	UPDATE
C		UPDATE
C	IF LOCK(IEULER,K) IS POSITIVE, AXIS K IS UNLOCKED;	UPDATE
C	TO LOCK AXIS SET IEULER TO LOCK(IEULER,K).	UPDATE
C		UPDATE
	DO 36 J=1,NJNT	UPDATE
	IF (IABS(IPIN(J)).NE.4) GO TO 36	UPDATE
	JEULER = IEULER(J)	UPDATE
	CALL DOT31(HIR(1,1,J),TQ(1,J),TQTEST)	UPDATE
	DO 31 K=1,3	UPDATE
	K3J = 3*J-3+K	UPDATE
	NLOCK = LOCK(JEULER,K)	UPDATE
	IF (NLOCK.GT.0) GO TO 29	UPDATE
	IF (VISC(4,K3J).EQ.0.0) GO TO 31	UPDATE
	IF (DABS(TQTEST(K)).LE.VISC(4,K3J)) GO TO 31	UPDATE
	JEULER = -NLOCK	UPDATE
	HA(K,2*J-1) = TQTEST(K)	UPDATE
	GO TO 31	UPDATE
	29 IF (HA(K,2*J).EQ.0.0) HA(K,2*J-1) = 0.0	UPDATE
	IF (VISC(5,K3J).EQ.0.0) GO TO 30	UPDATE
	IF (DABS(TQTEST(K)).LT.VISC(5,K3J)) JEULER = NLOCK	UPDATE
	GO TO 31	UPDATE
	30 IF (VISC(6,K3J).EQ.0.0) GO TO 31	UPDATE
	IF (DABS(ANGD(K,J)).LT.VISC(6,K3J)) JEULER = NLOCK	UPDATE
	31 CONTINUE	UPDATE
	IF (JEULER.EQ.IEULER(J)) GO TO 36	UPDATE
	TMSEC = 1000.0*TIME	UPDATE
	WRITE (6,2) TMSEC,J,IEULER(J),JEULER	UPDATE
	32 FORMAT('0 AT TIME =',F9.3,' MSEC, IEULER(',I2,	BUTLER1
	* ') HAS BEEN CHANGED FROM',I3,' TO',I3)	BUTLER1
	IF (JEULER.EQ.8) GO TO 35	UPDATE
	IF (IEULER(J).EQ.7) GO TO 35	UPDATE
	IF (IEULER(J).EQ.6 .AND. (JEULER.EQ.2.OR.JEULER.EQ.1)) GO TO 35	UPDATE
	IF (IEULER(J).EQ.5 .AND. (JEULER.EQ.3.OR.JEULER.EQ.1)) GO TO 35	UPDATE
	IF (IEULER(J).EQ.4 .AND. (JEULER.EQ.3.OR.JEULER.EQ.2)) GO TO 35	UPDATE
	MODE = -1	UPDATE
	K = JEULER	UPDATE
	IF (K.GT.3) GO TO 33	UPDATE
	IF (K.EQ.2) GO TO 34	UPDATE
	K4 = 4-K	UPDATE
	CALL CROSS (HIR(1,K4,J),HIR(1,2,J),T)	UPDATE
	IEULER(J) = 8	UPDATE
	IPIN(J) = 4	UPDATE
	CALL IMPLS2(MODE,J,T)	UPDATE
	I = -1	UPDATE
	GO TO 35	UPDATE
	33 MODE = 1	UPDATE
	K = K-3	UPDATE
	IF (K.GT.3) MODE=0	UPDATE
	34 IEULER(J) = 8	UPDATE

IPIN(J) - 4	UPDATE
CALL IMPLS2(MODE,J,HIR(1,K,J))	UPDATE
I - -1	UPDATE
35 IEULER(J) = JEULER	UPDATE
IPIN(J) = 4	UPDATE
IF (IEULER(J).NE.8) IPIN(J) = -4	UPDATE
C GET SINE AND COSINE OF NUTATION: IF IEULER GOES TO STATE 2	JDRIFT
CALL EJOINT(-1,J)	JDRIFT
IF(JEULER.NE.2) GOTO 36	JDRIFT
TQM=ANG(2,J)+CONST(2,J)	JDRIFT
CONST(4,J) = DCOS(TQM)	JDRIFT
CONST(5,J) = DSIN(TQM)	JDRIFT
36 CONTINUE	UPDATE
DO 90 J = 1,NJNT	SLIP
IF (IABS(IPIN(J)).LE.4) GO TO 90	SLIP
IF (IEULER(J).GE.0) GO TO 90	SLIP
IF (CONST(1,J).EQ.0.0.AND.CONST(2,J).EQ.0.0) GO TO 90	SLIP
M = JNT(J)	SLIP
FTEST = XDY(HT(1,3,2*J-1),D(1,1,M),F(1,J))	SLIP
IF (FTEST.GE.CONST(1,J).AND.FTEST.LE.CONST(2,J)) GO TO 90	SLIP
IEULER(J) = 0	SLIP
TMSEC = 1000.0*TIME	SLIP
WRITE (6,88) TMSEC,J	SLIP
88 FORMAT('/O AT TIME =',F9.3,' MSEC, JOINT ',I3,' HAS BEEN',	SLIP
* ' UNLOCKED AND ALLOWED TO SLIP.'/)	SLIP
90 CONTINUE	SLIP
C F IS THE FORCE ON SEGMENT J+1, - F IS ON SEGMENT M	SLIP
C	UPDATE
37 IF (NQ.LE.0) GO TO 41	UPDATE
DO 40 K=1,NQ	UPDATE
IF (KQTYPE(K).LT.3) GO TO 40	UPDATE
IF (KQTYPE(K).GT.4) GO TO 40	UPDATE
IF (CFQQ(K).LT.0.0) KQTYPE(K) = -KQTYPE(K)	UPDATE
IF (CFQQ(K).LT.0.0) GO TO 39	UPDATE
C	UPDATE
C TEST IF ROLLING CONSTRAINT SHOULD BE SLIDING AND VICE VERSA.	UPDATE
C	UPDATE
QN = -XDY(TQQ(1,K),HHT(1,1,K),QQ(1,K))	UPDATE
IF (NPRT(24).NE.0) WRITE (6,38) KQTYPE(K),KQ1(K),KQ2(K),	UPDATE
* (RK1(II,K),II-1,3), (RK2(II,K),II-1,3),	UPDATE
* ((HHT(II,J,K),J-1,3), II-1,3),	UPDATE
* (QQ(II,K),II-1,3), (TQQ(II,K),II-1,3), (RQQ(II,K),II-1,3),	UPDATE
* (HQQ(II,K),II-1,3), SQQ(K),CFQQ(K),QN	UPDATE
38 FORMAT('O UPDATE ROLL-SLIDE TEST'/(2X,9G14.6))	UPDATE
IF (QN.LT.0.0) KQTYPE(K) = -4	UPDATE
IF (QN.LT.0.0) GO TO 39	UPDATE
QDOTQ = QQ(1,K)**2 + QQ(2,K)**2 + QQ(3,K)**2	UPDATE
QT = DSQRT(QDOTQ-QN**2)	UPDATE
IF (KQTYPE(K).EQ.3 .AND. QT.LE.CFQQ(K)*QN) GO TO 40	UPDATE
IF (KQTYPE(K).EQ.4 .AND. QT.GE.0.9*CFQQ(K)*QN) GO TO 40	UPDATE
KQTYPE(K) = 7-KQTYPE(K)	UPDATE
39 CALL OUTPUT(0)	UPDATE

```
CALL SETUP2
CALL DAUX(K)
IF (NPRT(24).NE.0) CALL OUTPUT(1)
IF (NPRT( 3).NE.0) CALL PRINT (6HUPDATE)
I = -1
40 CONTINUE
41 CALL ELTIME(2,7)
42 RETURN
END
```

```
UPDATE
UPDATE
UPDATE
UPDATE
UPDATE
UPDATE
UPDATE
UPDATE
UPDATE
```



```

SUBROUTINE UPDFDC (M)                                UPDFDC
C                                                    REV III.2 08/08/84REVIII
C UPDATE FORCE DEFLECTION CURVE DEFINITION THAT IS DEFINED UPDFDC
C IN LOCATION M OF NTAB ARRAY. SUBROUTINE ASSUMES THAT UPDFDC
C A SUCCESSFUL INTEGRATION STEP HAS JUST BEEN COMPLETED AND UPDFDC
C WILL COMPUTE ENTIRE CURVE DEFINITION TO BE VALID FOR NEXT STEP. UPDFDC
C                                                    UPDFDC
  IMPLICIT REAL*8(A-H,O-Z)                            UPDFDC
  COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(1250),TAB(4500)DIMENB
  L      = NTAB(M)                                    UPDFDC
  IF (L.EQ.0) GO TO 99                                UPDFDC
  D      = TAB(L)                                     UPDFDC
  IF (D.LT.0.0) D = 0.0                               UPDFDC
  DLAST = TAB(L+1)                                    UPDFDC
  IF (D.EQ.DLAST) GO TO 99                            UPDFDC
  DCUBIC = TAB(L+6)                                   UPDFDC
  IF (D.EQ.DCUBIC) GO TO 98                           UPDFDC
  AREA   = TAB(L+2)                                   UPDFDC
  RLAST  = TAB(L+3)                                   UPDFDC
  GLAST  = TAB(L+4)                                   UPDFDC
  DG     = TAB(L+5)                                   UPDFDC
  DGO    = DG                                         UPDFDC
  DREF   = TAB(L+7)                                   UPDFDC
  DMAX   = TAB(L+8)                                   UPDFDC
  DINER  = TAB(L+9)                                   UPDFDC
  FDMAX  = TAB(L+10)                                  UPDFDC
  DCO    = TAB(L+18)                                  UPDFDC
  LQ     = L+11                                       UPDFDC
  LC     = L+14                                       UPDFDC
  IF (NTAB(M+1).LT.0) GO TO 98                         UPDFDC
  IF (D-DCUBIC) 10,98,20                               UPDFDC
C                                                    UPDFDC
C D < DCUBIC, DEFINE NEW CUBIC                        UPDFDC
C  $Y(X) = A0 + A1*(X-X1) + A2*(X-X1)**2 + A3*(X-X1)**3$  UPDFDC
C WHOSE DERIVATIVE IS                                UPDFDC
C  $Y'(X) = A1 + 2*A2*(X-X1) + 3*A3*(X-X1)**2$  UPDFDC
C                                                    UPDFDC
10 X1 = DMAX1 (D,DG)                                  UPDFDC
   X2 = DREF                                           UPDFDC
C                                                    UPDFDC
C IF INERTIAL SPIKE EXISTS AND IF DIMAX < DREF , DROP INERTIAL SPIKEUPDFDC
  NI = NTAB(M+2)                                       UPDFDC
  IF (NI.GT.0.AND.TAB(NI+3).GT.0.0.AND.DREF.GT.TAB(NI+3))NTAB(M+2)=0UPDFDC
  DX = X2-X1                                           UPDFDC
  X  = X1-DG                                           UPDFDC
  Y1 = TAB(LQ) +X *(TAB(LQ+1)+X *TAB(LQ+2))          UPDFDC
  Y1P = TAB(LQ+1)+2.0*X *TAB(LQ+2)                  UPDFDC
  X2DOT = 0.0                                          UPDFDC
  CALL FRCDFL (X2,X2DOT,M,0,Y2P,ELOSS)                UPDFDC
  CALL FRCDFL (X2,X2DOT,M,1,Y2 ,ELOSS)                UPDFDC
  DCUBIC = X1                                          UPDFDC
  DCO    = DCUBIC                                     UPDFDC

```

C		UPDFDC
C	A0 = Y(X1) (THE VALUE OF THE QUADRATIC AT X1)	UPDFDC
C	A1 = Y'(X1) (THE DERIVATIVE OF THE QUADRATIC AT X1)	UPDFDC
C		UPDFDC
	A0 = Y1	UPDFDC
	A1 = Y1P	UPDFDC
C		UPDFDC
C	SOLVE SIMULTANEOUSLY FOR A2 AND A3	UPDFDC
C	A2*(X2-X1)**2 + A3*(X2-X1)**3 = Y(X2)-A0-A1*(X2-X1)	UPDFDC
C	2*A2*(X2-X1) + 3*A3*(X2-X1)**2 = Y'(X2)-A1	UPDFDC
C		UPDFDC
	R13 = (Y2 - Y1 - Y1P*DX)/DX**2	UPDFDC
	R23 = (Y2P - Y1P)/DX	UPDFDC
	A2 = 3.0*R13 - R23	UPDFDC
	A3 = (R23 - 2.0*R13)/DX	UPDFDC
C		UPDFDC
C	IF LOCAL MINIMUM OF CUBIC (ABSCISSA VALUE WHERE Y'(X) = 0)	UPDFDC
C	LIES BETWEEN DCUBIC AND DREF AND IS NEGATIVE, THEN REPLACE	UPDFDC
C	CUBIC DEFINITION WITH STRAIGHT LINE BETWEEN (X1,Y1) AND (X2,Y2).	UPDFDC
C		UPDFDC
	IF (A3.NE.0.0) GO TO 14	UPDFDC
	R2 = -0.5*A1/A2	UPDFDC
	GO TO 15	UPDFDC
14	A33 = 3.0*A3	UPDFDC
	DISC = A2**2-A1*A33	UPDFDC
	IF (DISC.LT.0.0) GO TO 13	UPDFDC
	SQDISC = DSQRT(DISC)	UPDFDC
	R1 = (-A2+SQDISC)/A33	UPDFDC
	IF (R1.LE.0.0.OR.R1.GE.DX) GO TO 11	UPDFDC
	FR1 = A0+R1*(A1+R1*(A2+R1*A3))	UPDFDC
	IF (FR1.LT.0.0) GO TO 12	UPDFDC
11	R2 = (-A2-SQDISC)/A33	UPDFDC
15	IF (R2.LE.0.0.OR.R2.GE.DX) GO TO 13	UPDFDC
	FR2 = A0+R2*(A1+R2*(A2+R2*A3))	UPDFDC
	IF (FR2.GE.0.0) GO TO 13	UPDFDC
12	A0 = Y1	UPDFDC
	A1 = (Y2-Y1)/DX	UPDFDC
	A2 = 0.0	UPDFDC
	A3 = 0.0	UPDFDC
13	TAB(LC) = A0	UPDFDC
	TAB(LC+1) = A1	UPDFDC
	TAB(LC+2) = A2	UPDFDC
	TAB(LC+3) = A3	UPDFDC
	TAB(L+6) = DCUBIC	UPDFDC
	TAB(L+18) = DCO	UPDFDC
	GO TO 98	UPDFDC
20	IF (D-DREF) 21,21,30	UPDFDC
C		UPDFDC
C	DCUBIC < D < DREF, DEFINE NEW QUADRATIC FROM CUBIC CURVE.	UPDFDC
C		UPDFDC
21	X = D-DCO	UPDFDC
	Y2 = TAB(LC)+X*(TAB(LC+1)+X*(TAB(LC+2)+X*TAB(LC+3)))	UPDFDC

```

X1 = DCUBIC - DG
AREA = X1*(TAB(LQ)+X1*(TAB(LQ+1)/2.0+X1*TAB(LQ+2)/3.0))
* + X*(TAB(LC)+X*(TAB(LC+1)/2.0+X*(TAB(LC+2)/3.0+X*TAB(LC+3)/4.0))
X = DCUBIC - DCO
IF (X.NE.0.0) AREA = AREA
* - X*(TAB(LC)+X*(TAB(LC+1)/2.0+X*(TAB(LC+2)/3.0+X*TAB(LC+3)/4.0))
GO TO 31

C
C DREF < D, DEFINE NEW QUADRATIC FROM BASE CURVE.
C
C IF DINER < D , REMOVE INERTIAL SPIKE
C
30 IF (NTAB(M+2).GT.0 .AND. D.GE.DINER) NTAB(M+2) = 0
NR = NTAB(M+3)
RLAST = 1.0
IF (NR.GT.0 ) RLAST = EVALFD(D,NR,1)
IF (RLAST.NE.1.0) GO TO 39

C
C R = 1, USE BASE CURVE FOR UNLOADING
C
DG = 0.0
DCUBIC = 0.0
DREF = 0.0
A0 = 0.0
A1 = 0.0
A2 = 0.0
GO TO 32
39 NG = NTAB(M+4)
GLAST = 0.0
IF (NG.GT.0 ) GLAST = EVALFD(D,NG,1)
NB = NTAB(M+1)
D0 = TAB(NB)
DG = D0 + GLAST*(D-D0)
Y2 = EVALFD(D, NB,1)
NI = NTAB(M+2)
IF (NI.GT.0) Y2 = Y2+EVALFD(D,NI,1)
AREA = EVALFD(D,NB,2)
DREF = D
31 DCUBIC = D
X1 = DG
X2 = D
DX = X2-X1
Y1 = 0.0
RAREA = RLAST*AREA

C
C COMPUTE UNLOADING QUADRATIC COEFFICIENTS SUCH THAT
C ENDPOINT DERIVATES ARE NON-NEGATIVE.
C
A0 = 0.0
A1 = 2.0/DX*(3.0*RAREA/DX-Y2)
IF (A1.LT.0.0) A1 = 0.0
A2 = (Y2/DX-A1)/DX

```

	IF (A2.GE.0.0) GO TO 32	UPDFDC
	A1 = Y2/DX	UPDFDC
	A2 = 0.0	UPDFDC
C		UPDFDC
C	RESTORE TAB VALUES THAT MAY HAVE BEEN CHANGED	UPDFDC
C		UPDFDC
	. 32 TAB(L+2) = AREA	UPDFDC
	TAB(L+3) = RLAST	UPDFDC
	TAB(L+4) = GLAST	UPDFDC
	TAB(L+5) = DG	UPDFDC
	TAB(L+6) = DCUBIC	UPDFDC
	TAB(L+7) = DREF	UPDFDC
	TAB(LQ) = A0	UPDFDC
	TAB(LQ+1) = A1	UPDFDC
	TAB(LQ+2) = A2	UPDFDC
	98 TAB(L+1) = D	UPDFDC
	IF (D.GT.DGO .AND. DLAST.LE.DGO) M--M	UPDFDC
	99 RETURN	UPDFDC
	END	UPDFDC

```

SUBROUTINE VEHPOS                                VEHPOS
C                                                    REV IV    07/23/86TWOPI
C COMPUTES COMPONENTS OF VEHICLE ACCELERATIONS ONLY AS A FUNCTION VEHPOS
C OF TIME USING DATA AND TABLES PRODUCED BY SUBROUTINE VINPUT. VEHPOS
C                                                    VEHPOS
C IMPLICIT REAL*8 (A-H,O-Z) VEHPOS
COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND, VEHPOS
* NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG PAGE
COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30), VEHPOS
* SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30) VEHPOS
COMMON/VPOSTN/ ZPLT(3),SPLT(3),AXV(3,6),VATAB(6,501,6), VEHICL
* VTO(6),VDT(6),TIMEV(6),OMEGV(6),NVTAB(6),INDXV(6) VEHPOS
COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24), VEHPOS
* UNITL,UNITM,UNITT,GRAVTY(3),TWOPI TWOPI
DIMENSION AX(3) VEHPOS
T = TIME VEHPOS
M = 1 VEHPOS
15 DO 16 I=1,3 VEHPOS
16 AX(I) = AXV(I,M) VEHPOS
ATO = VTO(M) VEHPOS
ADT = VDT(M) VEHPOS
VTIME = TIMEV(M) VEHPOS
OMEG = OMEGV(M) VEHPOS
NATAB = NVTAB(M) VEHPOS
K = INDXV(M) VEHPOS
IF(NATAB.NE.0) GO TO 20 VEHPOS
C VEHPOS
C HALF-SINE WAVE DECELERATION VEHPOS
C VEHPOS
IF(T.GT.VTIME) T=VTIME VEHPOS
WT = OMEG*T VEHPOS
SWT = DSIN(WT) VEHPOS
DO 10 I=1,3 VEHPOS
AW = AX(I)*OMEG VEHPOS
SEGLA(I,K) = -AW*OMEG*SWT VEHPOS
10 WMEGD(I,K) = 0.0 VEHPOS
GO TO 99 VEHPOS
20 IF (NATAB.LT.0) GO TO 30 VEHPOS
C VEHPOS
C UNIDIRECTIONAL DECELERATION VEHPOS
C VEHPOS
IF (T.LT.VTIME) GO TO 21 VEHPOS
C VEHPOS
C TIME POINT EXCEEDS TABLE, USE LAST VALUES OF ACCELERATION. VEHPOS
C VEHPOS
ACO = VATAB(1,NATAB,M) VEHPOS
GO TO 25 VEHPOS
C VEHPOS
C USE QUADRATIC INTERPOLATION FROM TABLES FOR CURRENT VALUE OF VEHPOS
C TIME TO BE CONSISTENT WITH SIMPSON INTEGRATION OF TABLES. VEHPOS
C VEHPOS
21 J= 0.5*(T-ATO)/ADT +1.0 VEHPOS

```

	XK = T/ADT -DFLOAT(2*J-1)	VEHPOS
	X1 = XK+1.0	VEHPOS
	X3 = XK-1.0	VEHPOS
	ACO = 0.5*XK*X3*VATAB(1,2*J-1,M)	VEHPOS
	* - X3*X1*VATAB(1,2*J ,M)	VEHPOS
	* + 0.5*XK*X1*VATAB(1,2*J+1,M)	VEHPOS
C		VEHPOS
C	COMPONENTS OF VEHICLE ACCELERATION.	VEHPOS
C		VEHPOS
	25 DO 29 I=1,3	VEHPOS
	SEGLA(I,K) = -G*AX(I)*ACO	VEHPOS
	29 WMEGD(I,K) = 0.0	VEHPOS
	GO TO 99	VEHPOS
C		VEHPOS
C	OMNIDIRECTIONAL DECELERATION	VEHPOS
C		VEHPOS
	30 J = (TIME-ATO)/ADT + 1.0	VEHPOS
	IF (J.GE.-NATAB) GO TO 32	VEHPOS
C		VEHPOS
C	INTERPOLATION FROM VINPUT TABLES OF COMPONENTS OF VEHICLE	VEHPOS
C	LINEAR AND ANGULAR ACCELERATION.	VEHPOS
C		VEHPOS
	TJ = ATO + DFLOAT(J-1)*ADT	VEHPOS
	DLT = TIME-TJ	VEHPOS
	R1 = DLT/ADT	VEHPOS
	R2 = 1.0-R1	VEHPOS
	DO 31 I=1,3	VEHPOS
	SEGLA(I,K) = -G*(VATAB(I ,J+1,M)*R1 + VATAB(I ,J,M)*R2)	VEHPOS
	31 WMEGD(I,K) = RADIAN*(VATAB(I+3,J+1,M)*R1 + VATAB(I+3,J,M)*R2)	VEHPOS
	GO TO 99	VEHPOS
C		VEHPOS
C	TIME POINT EXCEEDS TABLE, USE LAST VALUES OF ACCELERATION.	VEHPOS
C		VEHPOS
	32 J = - NATAB	VEHPOS
	DO 33 I=1,3	VEHPOS
	SEGLA(I,K) = -G*VATAB(I ,J,M)	VEHPOS
	33 WMEGD(I,K) = RADIAN*VATAB(I+3,J,M)	VEHPOS
	99 M = M+1	VEHPOS
	IF (M.LE.6 .AND. INDXV(M).NE.0) GO TO 15	VEHPOS
	RETURN	VEHPOS
	END	VEHPOS

	SUBROUTINE VINPUB	REV IV 07/24/86	VINPUT
C		SLIP	
C	PERFORMS CARD INPUT AND COMPUTES DATA AND TABLES REQUIRED BY		VINPUT
C	SUBROUTINE VEHPOS TO INTEGRATE THE CRASH VEHICLE MOTION FOR ONE OF		VINPUT
C	THREE PERMISSABLE OPTIONS:		VINPUT
C	(1) HALF SINE-WAVE LINEAR DECELERATION IMPULSE		VINPUT
C	(2) UNIDIRECTIONAL LINEAR DECELERATION TABULAR INPUT		VINPUT
C	(3) OMNIDIRECTIONAL LINEAR AND ANGULAR ACCELERATION TABULAR		VINPUT
C	INPUT (6 DEGREES OF FREEDOM VEHICLE MOTION)		VINPUT
C			VINPUT
	IMPLICIT REAL*8 (A-H,O-Z)		VINPUT
	COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,		VINPUT
*	NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG		PAGE
	COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),		VINPUT
*	SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)		VINPUT
	COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60),		SLIP
*	RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),		VINPUT
*	JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)		VINPUT
	COMMON/VPOSTN/ ZPLT(3),SPLT(3),AXV(3,6),VATAB(6,501,6),		VEHICL
*	VTO(6),VDT(6),TIMEV(6),OMEGV(6),NVTAB(6),INDXV(6)		VINPUT
	COMMON/TEMPVS/ XO(3),XDOTO(3),XCOMP(3),XVCOMP(3),ANGLE(3),		VINPUT
*	ATAB(15,501),DVEH(3,3),VMEG(3),VMEGD(3),		VEHICL
*	XACOMP(3),THET(3),AX(3),F(5,101),XYZ(103,6),TT(103),		CHGIII
*	VIPS,VMPH,ATO,ADT,VTIME,OMEG,NATAB		VINPUT
*	,SP(5,101,4),Q1(101,4),A1(3),W1(4),QD(4),QC(4)		JTF984
*	,IDMMY,DDMY(23887)		80386
	COMMON/INTEST/ SGTEST(3,4,30),XTEST(3,120),SEGT(120),REGT(120)		VINPUT
	REAL SEGT		VINPUT
	COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),		VINPUT
*	UNITL,UNITM,UNITT,GRAVTY(3),TWOPI		TWOPI
	COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),		VINPUT
*	BLTTTL(5,8),PLTTTL(5,30),BAGTTL(5,6),SEG(30),		VINPUT
*	JOINT(30),CGS(30),JS(30)		VINPUT
	REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTTL,BAGTTL,SEG,JOINT		VINPUT
	LOGICAL*1 CGS,JS		VINPUT
	DIMENSION IDYPR(3)		VINPUT
	REAL VEH(6),GRND		VINPUT
	DATA VEH/4HVEH1,4HVEH2,4HVEH3,4HVEH4,4HVEH5,4HVEH /,GRND/4HGRND/		VINPUT
	DATA IDYPR/3,2,1/		VINPUT
	DATA MXTAB2/99/,MXTAB3/501/,MXTAB4/101/		MISC
C			VINPUT
C	READ AND PRINT CONTENTS OF CARDS C.1 AND C.2		VINPUT
C			VINPUT
	NVEH = NSEG		VINPUT
	NVH = 0		VINPUT
	DO 11 I=1,6		VINPUT
11	INDXV(I) = 0		VINPUT
12	READ (5,13) VPSTTL		VINPUT
13	FORMAT (20A4)		VINPUT
	READ(5,14) ANGLE,VIPS,VTIME,XO,NATAB,ATO,ADT,MSEG		VINPUT
14	FORMAT(8F6.0,I6,2F6.0,I6)		VINPUT
	INTAB = IABS(NATAB)		CHGIII

```

IF (NATAB.GT.0.AND.INTAB.GT.MXTAB2) STOP 79
WRITE (6,15) NPG,VPSTTL,ANGLE,VIPS,VTIME,XO,NATAB,ATO,ADT,MSEG
NPG = NPG+1
15 FORMAT('1 VEHICLE DECELERATION INPUTS',94X,'PAGE',I5/120X,
*      'CARDS C'/3X,20A4//
*      7X,'YAW',9X,'PITCH',7X,'ROLL',8X,'VIPS',8X,'VTIME',7X,'XO(X)',VININPUT
*      7X,'XO(Y)',7X,'XO(Z)',2X,'NATAB',6X,'ATO',9X,'ADT',4X,'MSEG'/VININPUT
*      8F12.3,I5,2X,2F12.6,I5)VININPUT
DA1 = ANGLE(1)*RADIANVININPUT
DA2 = ANGLE(2)*RADIANVININPUT
AX(3) = DCOS(DA2)VININPUT
AX(1) = DCOS(DA1)*AX(3)VININPUT
AX(2) = DSIN(DA1)*AX(3)VININPUT
AX(3) = DSIN(DA2)VININPUT
IF(NATAB.NE.0) GO TO 18VININPUT
C
C HALF-SINE WAVE DECELERATIONVININPUT
C
OMEG = PI/VTIMEVININPUT
AT = 0.5*VIPS/OMEGVININPUT
IF (VIPS.LT.0.0) VIPS = 0.0VININPUT
DO 16 I=1,3VININPUT
XACOMP(I) = 0.0VININPUT
XDOTO(I) = VIPS*AX(I)VININPUT
16 AX(I) = AT*AX(I)VININPUT
WRITE (6,17) VIPS,UNITL,UNITT,ANGLE,VTIME,UNITTVININPUT
17 FORMAT('0 PASSENGER COMPARTMENT DISPLACEMENT HISTORY'/VININPUT
*      ' ANALYTICAL HALF-SINE WAVE DECELERATION'/VININPUT
*      ' VO-',F8.3,1X,A4,'/',A4,' , OBLIQUE ANGLES -',3F7.2,VININPUT
*      ' DEGREES, TIME DURATION -',F7.3,1X,A4//)VININPUT
GO TO 28VININPUT
18 IF (NATAB.LT.0) GO TO 31VININPUT
C
C FOR UNIDIRECTIONAL VEHICLE MOTIONVININPUT
C READ LINEAR DECELERATION TABLES FROM CARDS C.3VININPUT
C
READ (5,19) (ATAB(1,I),I=1,NATAB)VININPUT
19 FORMAT (12F6.0)VININPUT
C
C EXTEND TABLE IF NECESSARY SUCH THAT NATAB IS ODD ANDVININPUT
C LAST ENTRY NEED NOT BE ZERO. IF TABLE SIZE IS EXCEEDED ON TIME,VININPUT
C VALUE OF LAST ENTRY WILL BE USED.VININPUT
C
IF (MOD(NATAB,2).EQ.1) GO TO 20VININPUT
ATAB(1,NATAB+1) = ATAB(1,NATAB)VININPUT
NATAB = NATAB+1VININPUT
20 VTIME = ADT * DFLOAT(NATAB-1)VININPUT
C
C USING SIMPSON'S INTEGRATION, COMPUTE VELOCITY AND DISPLACEMENTVININPUT
C TABLE FOR NATAB EQUALLY SPACED (ADT) TIME POINTS.VININPUT
C FOR I=1,NATABVININPUT
C ATAB(1,I) = LINEAR DECELERATION (G'S)VININPUT

```



```

C      ATAB(2,I) = LINEAR VELOCITY (L UNITS/T UNITS)          VINPUT
C      ATAB(3,I) = LINEAR DISPLACEMENT (L UNITS)             VINPUT
C
C      ATAB(2,1) = VIPS                                       VINPUT
C      ATAB(3,1) = 0.0                                       VINPUT
C      DA1 = ADT/3.0                                         VINPUT
C      DA2 = ADT/12.0                                        VINPUT
C      UNITS = -G                                           VINPUT
C      DO 22 J=2,3                                           VINPUT
C      DO 21 I=2,NATAB,2                                     VINPUT
C      F1 = ATAB(J-1,I-1) * UNITS                           VINPUT
C      F2 = ATAB(J-1,I ) * UNITS                            VINPUT
C      F3 = ATAB(J-1,I+1) * UNITS                           VINPUT
C      ATAB(J,I ) = ATAB(J,I-1) + DA2*(5.0*F1+8.0*F2-F3)   VINPUT
21 ATAB(J,I+1) = ATAB(J,I-1) + DA1*( F1+4.0*F2+F3)         VINPUT
22 UNITS = 1.0                                             VINPUT
C
C      PRINT TABLES                                       VINPUT
C
C      WRITE (6,23) (UNITL,UNITT,UNITL,I-1,2)              VINPUT
23 FORMAT('O UNIDIRECTIONAL VEHICLE POSITION TABLES'//      VINPUT
*      2('      TIME      ACC      VELOCITY      POSITION  ')/ VINPUT
*      2('      (MSEC)      (G)      (' ,A4, '/' ,A4, '),' ,5X, '(' ,A4, '),' ,4X)// VINPUT
C      DO 26 J=1,50                                         VINPUT
C      IF (J.GT.NATAB) GO TO 26                             VINPUT
C      T1 = (ATO + DFLOAT(J-1)*ADT)*1000.0                 VINPUT
C      IF (J+50.LE.NATAB) GO TO 25                         VINPUT
C      WRITE (6,24) T1,(ATAB(I,J),I-1,3)                   VINPUT
24 FORMAT(2(F11.5,F10.2,F13.4,F13.5,3X))                   VINPUT
C      GO TO 26                                             VINPUT
25 T2 = (ATO + DFLOAT(J+49)*ADT)*1000.0                   VINPUT
C      WRITE (6,24) T1,(ATAB(I,J),I-1,3),T2,(ATAB(I,J+50),I-1,3) VINPUT
26 CONTINUE                                               VINPUT
C
C      INITIALIZATION                                       VINPUT
C
C      DO 27 I=1,3                                          VINPUT
C      XACOMP(I) = -G*AX(I)*ATAB(1,1)                      VINPUT
27 XDOT0(I) = VIPS*AX(I)                                  VINPUT
28 DO 30 I=1,3                                          VINPUT
C      DO 29 J=1,3                                          VINPUT
29 DVEH(I,J) = 0.0                                       VINPUT
C      DVEH(I,I) = 1.0                                       VINPUT
C      VMEGD(I) = 0.0                                       VINPUT
30 VMEG(I) = 0.0                                       VINPUT
C      GO TO 64                                             VINPUT
C
C      FOR OMNIDIRECTIONAL (6 DEGREES OF FREEDOM) VEHICLE MOTION VINPUT
C      READ LINEAR DECELERATION AND ANGULAR ACCELERATION TABLES VINPUT
C      FROM CARDS C.2.B AND C.4.                            CHGIII
C
31 MATAB = -NATAB                                         VINPUT

```

```

      READ (5,32) LTYPE,LFIT,NPTS,(VMEG(I),I=1,3)
32  FORMAT (3I6,22X,3F10.0)
      IF (MATAB.GT.MXTAB3) STOP 80
      IF (LTYPE.EQ.2.AND.LFIT.LT.1) STOP 82
      IF (LTYPE.EQ.1.AND.LFIT.LT.2) STOP 83
      IF (LTYPE.GT.0) GO TO 34
      READ (5,33) ((ATAB(I,J),I=1,3),(ATAB(I,J),I=10,12),J=1,MATAB)
33  FORMAT (10X,6F10.0)
      ISKIP = 0
      GO TO 46
C
C   FOR SPLINE FIT VEHICLE MOTION
C   READ DATA FROM CARDS C.5.
C
34  LPTS = LTYPE-1 + NPTS
      IF (NPTS.GT.MXTAB3) STOP 81
      READ (5,35) (TT(I),(XYZ(I,J),J=1,6),I=1,LPTS)
35  FORMAT (7F10.0)
      WRITE (6,36) LTYPE,LFIT,NPTS
36  FORMAT ('O SPLINE FIT TABULAR INPUT'//
*      3X,'LTYPE =',I6,' LFIT =',I6,' NPTS =',I6/)
      IF (LTYPE.EQ.2) WRITE(6,701) UNITL,UNITT,TT(1),(XYZ(1,J),J=1,6)
      IF (LTYPE.EQ.3) WRITE(6,702) UNITL,UNITT,TT(1),(XYZ(1,J),J=1,6),
* UNITL,UNITT,UNITT,UNITT,TT(2),(XYZ(2,J),J=1,6)
701  FORMAT(32X,'INITIAL LINEAR POSITION (' ,A4,')',17X,'INITIAL ANGULACHGIII
*R POSITION (DEG)',/,3X,'TIME(' ,A4,')-',F9.4,3X,2('X-',F10.3,2X,
* 'Y-',F10.3,2X,'Z-',F10.3,8X),/)
702  FORMAT(32X,'INITIAL LINEAR POSITION (' ,A4,')',17X,'INITIAL ANGULACHGIII
*R POSITION (DEG)',/,3X,'TIME(' ,A4,')-',F9.4,3X,2('X-',F10.3,2X,
* 'Y-',F10.3,2X,'Z-',F10.3,8X),/,30X,'INITIAL LINEAR VELOCITY (' ,CHGIII
* A4,'/' ,A4,')',12X,'INITIAL ANGULAR VELOCITY (DEG/' ,A4,')',
*/,3X,'TIME(' ,A4,')-',F9.4,3X,2('X-',F10.2,2X,'Y-',
* F10.2,2X,'Z-',F10.2,8X),/)
      IF (LTYPE.EQ.1) WRITE(6,703) UNITL,UNITT
      IF (LTYPE.EQ.2) WRITE(6,704) UNITL,UNITT,UNITT,UNITT
      IF (LTYPE.EQ.3) WRITE(6,705) UNITT,UNITT
703  FORMAT(29X,'LINEAR POSITION (' ,A4,')',21X,'ANGULAR POSITION (DECHGIII
*G)',/,5X,'TIME(' ,A4,')',11X,'X',11X,'Y',11X,'Z',18X,'YAW',8X,
* 'PITCH',8X,'ROLL')
704  FORMAT(26X,'LINEAR VELOCITY (' ,A4,'/' ,A4,')',16X,
* 'ANGULAR VELOCITY (DEG/' ,A4,')',/,5X,'TIME(' ,A4,')',
* 11X,2('X',11X,'Y',11X,'Z',19X))
705  FORMAT(26X,'LINEAR DECELERATION (G'S)',15X,
* 'ANGULAR ACCELERATION (DEG/' ,A4,'**2)',/,5X,'TIME(' ,A4,')',
* 11X,2('X',11X,'Y',11X,'Z',19X))
      IF (LTYPE.EQ.1) WRITE(6,706) (TT(I),(XYZ(I,J),J=1,6),I=1,LPTS)
      IF (LTYPE.EQ.2) WRITE(6,706) (TT(I),(XYZ(I,J),J=1,6),I=2,LPTS)
      IF (LTYPE.EQ.3) WRITE(6,706) (TT(I),(XYZ(I,J),J=1,6),I=3,LPTS)
706  FORMAT(1X,F12.5,6X,3F12.3,8X,3F12.3)
      DO 37 I=1,3
      XO(I) = XYZ(1,I)
      XDOTO(I) = XYZ(2,I)

```

VMEG(I) = XYZ(2,I+3)	VINPUT
37 ANGLE(I) = XYZ(1,I+3)	JTF984
IMJ = 6	JTF984
IF(LTYPE.EQ.1)IMJ = 3	JTF984
DO 45 II=1,IMJ	JTF984
CALL SPLINE (TT(LTYPE),XYZ(LTYPE,II),F,NPTS,LFIT)	VINPUT
I = II	VINPUT
IF (II.GT.3) I = II + 6	VINPUT
IF(LTYPE.EQ.1) XDOTO(I) = F(3,1)	JTF984
UNITS = 1.0	JTF984
IF (LTYPE.LT.3 .AND. II.LE.3) UNITS = -1.0/G	VINPUT
K1 = 1	VINPUT
DO 45 J=1,MATAB	VINPUT
TTT = ATO + DFLOAT(J-1)*ADT	VINPUT
DO 39 L=K1,NPTS	JTF984
K = L	JTF984
IF (TTT.LT.F(1,L+1)) GO TO 40	VINPUT
39 CONTINUE	VINPUT
40 K1 = K	VINPUT
DX = TTT - F(1,K)	VINPUT
IF (LTYPE-2) 41,42,43	BUTLER1
41 ACC = 2.0*F(4,K) + 6.0*DX*F(5,K)	VINPUT
GO TO 44	VINPUT
42 ACC = F(3,K) + DX*(2.0*F(4,K)+3.0*DX*F(5,K))	VINPUT
GO TO 44	VINPUT
43 ACC = F(2,K) + DX*(F(3,K)+DX*(F(4,K)+DX*F(5,K)))	VINPUT
44 ATAB(I,J) = ACC*UNITS	VINPUT
45 CONTINUE	VINPUT
ISKIP = 1	VINPUT
IF(LTYPE.NE.1)GO TO 46	JTF984
C CODE FOR OMEGA ROUTINE: COMPUTE ATAB(I,J),I=10,11,12 J = 1,MATAB	JTF984
DO 80 I = 1,NPTS	JTF984
DO 91 K = 1,3	JTF984
91 A1(K) = XYZ(I,K+3)	JTF984
CALL QUAT(A1,W1)	JTF984
DO 76 K = 1,4	JTF984
76 Q1(I,K) = W1(K)	JTF984
IF(I.EQ.1)GO TO 80	JTF984
TA = 0.0	JTF984
TB = 0.0	JTF984
DO 77 K = 1,4	JTF984
TA = TA + DABS(Q1(I,K) - Q1(I-1,K))	JTF984
77 TB = TB + DABS(Q1(I,K) + Q1(I-1,K))	JTF984
IF(TA.LE.TB)GO TO 80	JTF984
DO 78 K = 1,4	JTF984
78 Q1(I,K) = -Q1(I,K)	JTF984
80 CONTINUE	JTF984
DO 82 K = 1,4	JTF984
82 CALL SPLINE(TT,Q1(1,K),SP(1,1,K),NPTS,LFIT)	JTF984
DO 90 J = 1,MATAB	JTF984
TTT = ATO + DFLOAT(J-1)*ADT	JTF984
K1 = 1	JTF984

	DO 83 L = K1,NPTS	JTF984
	K = L	JTF984
83	IF(TTT.LT.SP(1,L+1,1))GO TO 84	JTF984
84	K1 = K	JTF984
	DX = TTT - SP(1,K,1)	JTF984
	DO 85 L = 1,4	JTF984
	W1(L) = SP(2,K,L)+DX*(SP(3,K,L)+DX*(SP(4,K,L)+DX*SP(5,K,L)))	JTF984
	QD(L) = 2.0*SP(4,K,L) + 6.0*DX*SP(5,K,L)	JTF984
85	IF(J.EQ.1)QC(L) = SP(3,K,L)+DX*(2.0*SP(4,K,L)+DX*3.0*SP(5,K,L))	MISC
	CCC = 2.0/RADIAN	JTF984
	IF(J.GT.1)GO TO 88	JTF984
	CALL CROSS(QC(2),W1(2),A1)	JTF984
	DO 86 K = 1,3	JTF984
86	VMEG(K) = CCC*(W1(1)*QC(K+1) - QC(1)*W1(K+1) + A1(K))	JTF984
	CALL DRCQUA(DVEH,W1)	JTF984
	CALL YPRDEG(DVEH,ANGLE)	JTF984
88	CALL CROSS(QD(2),W1(2),QC(2))	JTF984
	DO 89 K = 2,4	JTF984
89	ATAB(K+8,J) = CCC*(W1(1)*QD(K)-QD(1)*W1(K) + QC(K))	JTF984
90	CONTINUE	JTF984
46	DO 55 J=1,MATAB	VINPUT
	IF (MOD(J,45).NE.1) GO TO 49	VINPUT
C		VINPUT
C	PRINT PAGE HEADING AT START OF EACH 45 TIME POINTS.	VINPUT
C		VINPUT
	IPAGE = (J-1)/45 + 1	VINPUT
	IF (ISKIP.EQ.1) WRITE (6,75) NPG	PAGE
	IF (ISKIP.EQ.1) NPG=NPG+1	PAGE
75	FORMAT('1',122X,'PAGE',I5)	PAGE
	WRITE (6,48) VPSTTL,IPAGE,UNITL,UNITT,UNITL	PAGE
48	FORMAT('0 VEHICLE LINEAR TIME HISTORY',3X,20A4,3X,	PAGE
	* 'PAGE NO.',I3//	VINPUT
	* 4X,'TIME',12X,'LINEAR DECELERATIONS (G'S)',	VINPUT
	* 11X,'LINEAR VELOCITIES ('A4','/'A4,')',	VINPUT
	* 11X,'LINEAR DISPLACEMENTS ('A4,')' /	VINPUT
	* 3X,'(MSEC)',3(11X,'X',11X,'Y',11X,'Z',3X) /)	VINPUT
	ISKIP = 1	VINPUT
49	IF (J.GT.1) GO TO 52	VINPUT
C		VINPUT
C	INTEGRATION INITIALIZATION FOR TIME = 0.	VINPUT
C		VINPUT
	DO 50 I=1,3	VINPUT
	ATAB(I+6,J) = XO(I)	VINPUT
50	ATAB(I+12,J) = VMEG(I)	JTF984
	CALL DRCYPR (DVEH,ANGLE, IDYPR)	VINPUT
	DO 51 I=1,3	VINPUT
	IF (LTYPE.EQ.0) XDOT0(I) = VIPS*DVEH(1,I)	VINPUT
51	ATAB(I+3,J) = XDOT0(I)	VINPUT
	GO TO 54	VINPUT
52	DO 53 I=1,3	VINPUT
C		VINPUT
C	INTEGRATE LINEAR VELOCITY AND DISPLACEMENT.	VINPUT

C	ATAB(I+3,J) = ATAB(I+3,J-1) - G*ADT/2.0*(ATAB(I,J-1)+ATAB(I,J))	VINPUT
	53 ATAB(I+6,J) = ATAB(I+6,J-1)	VINPUT
	* +ADT*(ATAB(I+3,J-1) - G*ADT/6.0*(2.0*ATAB(I,J-1)+ATAB(I,J)))	VINPUT
	54 T1 = (ATO + DFLOAT(J-1)*ADT)*1000.0	VINPUT
	55 WRITE(6,56) T1,(ATAB(I,J),I-1,9)	VINPUT
	56 FORMAT(F9.3,3(3X,3F12.3))	VINPUT
	DO 61 J=1,MA*AB	VINPUT
	IF (MOD(J,45).NE.1) GO TO 58	VINPUT
C		VINPUT
C	PRINT PAGE HEADING AT START OF EACH 45 TIME POINTS.	VINPUT
C		VINPUT
	IPAGE = (J-1)/45 + 1	VINPUT
	WRITE (6,57) VPSTTL,NPG,IPAGE,UNITT,UNITT	PAGE
	NPG=NPG+1	PAGE
	57 FORMAT('1 VEHICLE ANGULAR TIME HISTORY',3X,20A4,10X,'PAGE',I5/	PAGE
	* 116X,'PAGE NO.',I3/	PAGE
	* 4X,'TIME',7X,'ANGULAR ACCELERATIONS (DEG/','A4,'**2)',	VINPUT
	* 7X,'ANGULAR VELOCITIES (DEG/','A4,')',	VINPUT
	* 11X,'ANGULAR DISPLACEMENTS (DEG)' /	VINPUT
	* 3X,'(MSEC)',2(11X,'X',11X,'Y',11X,'Z',3X),	VINPUT
	* 10X,'YAW',8X,'PITCH',8X,'ROLL' /)	VINPUT
	58 IF(J.EQ.1) GO TO 60	VINPUT
C		VINPUT
C	INTEGRATE ANGULAR VELOCITY AND DISPLACEMENT.	VINPUT
C		VINPUT
	DO 59 I=1,3	VINPUT
	ATAB(I+12,J) = ATAB(I+12,J-1)+(ATAB(I+9,J-1)+ATAB(I+9,J))*ADT/2.0	VINPUT
	59 THET(I) = ADT*(ATAB(I+12,J-1)+(2.0*ATAB(I+9,J-1)+ATAB(I+9,J))*ADT	VINPUT
	*/6.0)*RADIAN	VINPUT
	CALL DSETD(DVEH,THET,THT)	VINPUT
	60 CALL YPRDEG(DVEH,THET)	VINPUT
	T1 = (ATO + DFLOAT(J-1)*ADT)*1000.0	VINPUT
	61 WRITE (6,56) T1,(ATAB(I,J),I-10,15),THET	VINPUT
C		VINPUT
C	PROGRAM INITIALIZATION FOR TIME = 0.	VINPUT
C		VINPUT
	CALL DRCYPR (DVEH,ANGLE,IDYPR)	VINPUT
	DO 63 I=1,3	VINPUT
	XACOMP(I) = -G*ATAB(I,1)	VINPUT
	VMEG(I) = ATAB(I+12,1)*RADIAN	VINPUT
	63 VMEGD(I) = ATAB(I+9,1)*RADIAN	VINPUT
	64 J = MSEG	VINPUT
	IF (MSEG.EQ.0) GO TO 65	VINPUT
	IF (MSEG.LE.NSEG) GO TO 66	VINPUT
	IF (MSEG.NE.NVEH+1) STOP 6	VINPUT
	65 NVEH = NVEH+1	VINPUT
	J = NVEH	VINPUT
C		VINPUT
C	SETUP FOR ALL PRESCRIBED SEGMENT MOTION.	VINPUT
C		VINPUT
	66 NVH = NVH+1	VINPUT

ISING(J) = -1	VINPUT
IF (MSEG.GT.NSEG) SEG(J) = VEH(NVH)	VINPUT
RW(J) = 0.0	VINPUT
DO 67 I=1,3	VINPUT
RPHI (I,J) = 0.0	VINPUT
SEGLA(I,J) = VMEGD(I)	VINPUT
WMEGD(I,J) = XACOMP(I)	VINPUT
. 67 AXV(I,NVH) = AX(I)	VINPUT
VTO(NVH) = ATO	VINPUT
VDT(NVH) = ADT	VINPUT
OMEGV(NVH) = OMEG	VINPUT
TIMEV(NVH) = VTIME	VINPUT
NVTAB(NVH) = NATAB	VINPUT
INDXV(NVH) = J	VINPUT
NJ = IABS(NATAB)	VINPUT
IF (NJ.LE.0) GO TO 69	VINPUT
DO 68 K=1,NJ	VINPUT
DO 68 I=1,3	VINPUT
VATAB(I ,K,NVH) = ATAB(I,K)	VINPUT
68 VATAB(I+3,K,NVH) = ATAB(I+9,K)	VINPUT
69 IF (J.LE.NSEG) GO TO 72	VINPUT
C	VINPUT
C	VINPUT
C	VINPUT
SETUP FOR NEW VEHICLE (SEGMENT) MOTION.	VINPUT
W(J) = 0.0	VINPUT
RW(J) = 0.0	VINPUT
DO 71 I=1,3	VINPUT
DO 70 K=1,3	VINPUT
D(I,K,J) = DVEH(I,K)	VINPUT
70 SGTEST(I,K,J) = 0.0	VINPUT
SGTEST(I,4,J) = 0.0	VINPUT
SEGLP(I,J) = XO(I)	VINPUT
SEGLV(I,J) = XDOTO(I)	VINPUT
WMEG (I,J) = VMEG(I)	VINPUT
PHI (I,J) = 0.0	VINPUT
71 RPHI (I,J) = 0.0	VINPUT
72 IF (MSEG.NE.0) GO TO 12	VINPUT
SEG(NVEH) = VEH(6)	VINPUT
C	VINPUT
C	VINPUT
C	VINPUT
SET UP SEGMENT DATA FOR GROUND	VINPUT
NGRND = NVEH+1	VINPUT
IF (NGRND.GT.30 .OR. NVH.GT.6) STOP 7	VINPUT
SEG(NGRND) = GRND	VINPUT
J = NGRND	VINPUT
ISING(J) = -1	VINPUT
W(J) = 0.0	VINPUT
RW(J) = 0.0	VINPUT
DO 74 I=1,3	VINPUT
DO 73 K=1,3	VINPUT
D(I,K,J) = 0.0	VINPUT
73 SGTEST(I,K,J) = 0.0	VINPUT

D(I,I,J) = 1.0
SGTEST(I,4,J) = 0.0
SEGLP(I,J) = 0.0
SEGLV(I,J) = 0.0
SEGLA(I,J) = 0.0
WMEG (I,J) = 0.0
WMEGD(I,J) = 0.0
PHI (I,J) = 0.0
74 RPHI (I,J) = 0.0
RETURN
END

VINPUT
VINPUT
VINPUT
VINPUT
VINPUT
VINPUT
VINPUT
VINPUT
VINPUT
VINPUT
VINPUT
VINPUT

	DOUBLE PRECISION FUNCTION VISCOS(ZD,VISC,HA)		VISCOS
C		REV 19	10/23/78VISCOS
C	COMPUTES SUM OF COULOMB AND VISCOUS TORQUES		VISCOS
C	AT JOINTS AS A FUNCTION OF THETA DOT.		VISCOS
C	ACTUALLY ROUTINE RETURNS SUM/ZD.		VISCOS
C			VISCOS
C	ARGUMENTS:		VISCOS
C	ZD : !THETA DOT! WHERE THETA IS THE ANGLE OF THE JOINT.		VISCOS
C	VISC: ARRAY OF 5 VALUES DESCRIBING FUNCTION EVALUATION.		VISCOS
C			VISCOS
	IMPLICIT REAL*8 (A-H,O-Z)		VISCOS
	DIMENSION VISC(5)		VISCOS
	Z = ZD		VISCOS
	IF (ZD.LT.VISC(3)) Z = VISC(3)/(2.0-ZD/VISC(3))		VISCOS
	HA = (Z-ZD)/Z		VISCOS
	VISCOS = VISC(1)+VISC(2)/Z		VISCOS
	RETURN		VISCOS
	END		VISCOS


```

SUBROUTINE VISPR(IJ,NJ)
C
C                                     REV IV   02/01/88MISDOT
C
C   COMPUTES VISCOS AND SPRING TORQUES AT THE JOINTS
C   AND ADDS THEM TO THE U2 ARRAY.
C
C   ARGUMENTS:
C     NJ = 0 - REGULAR COMPUTATION FOR ALL JOINTS
C     # 0 - COMPUTE ONLY FOR JOINT NJ IMPULSE
C
C     IJ = 1 IMPULSE FOR FLEXURE ONLY
C     = 2 IMPULSE FOR TORSION ONLY
C     = 4 IMPULSE FOR GLOBALGRAPHIC ONLY
C
C   IMPLICIT REAL*8 (A-H,O-Z)
COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,
*   NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG
COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),
*   SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)
COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(4,60),HA(3,60),HB(3,60),
*   RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),
*   JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)
COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60),
*   F(3,30),TQ(3,30),WJ(30),A11(3,3,30)
COMMON/FORCES/PSF(7,70),BSF(4,20),SSF(10,40),BAGSF(3,20),
*   PRJNT(7,30),NPANEL(5),NPSF,NBSF,NSSF,NBGSF
COMMON/CEULER/ IEULER(30),HIR(3,3,90),ANG(3,30),ANGD(3,30),
*   FE(3,30),TQE(3,30),CONST(5,30)
COMMON/TEMPVI/ CREST,TTI(3),R1I(3),R2I(3),JSTOP(4,2,30)
COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),
*   UNITL,UNITM,UNITT,GRAVTY(3),TWOPI
COMMON/TEMPVS/ T3(3),T6(3),T7(3),T8(3),T9(3),
*   WIJ(3),ANGL(3),DH1(3,3),HD3(3,3),
*   HAD,HBD,WIJM,CV,CSA,CSB,TQC,DMMY(35067)
IF (NJNT.LE.0) GO TO 99
CALL ELTIME(1,13)
IF (NPRT(12).NE.0) WRITE (6,11)  TIME,NPG
IF (NPRT(12).NE.0) NPG=NPG+1
11 FORMAT('1 VISPR COMPUTATIONS FOR TIME -',F12.6,80X,'PAGE',I5)
J1 = 1
J2 = NJNT
IF (NJ.EQ.0) GO TO 13
J1 = NJ
J2 = NJ
13 DO 90 J=J1,J2
DO 12 L=1,3
T3(L) = 0.0
T6(L) = 0.0
ANGL(L) = 0.0
12 TQ(L,J) = 0.0
WJ(J) = 0.0
C
C   DO NOT COMPUTE TORQUES FOR NULL, LOCKED OR EUILER JOINTS.

```

C	I = IABS(JNT(J))	VISPR
	IF (I.LE.0) GO TO 90	VISPR
	CALL DOT33 (D(1,1,J+1),HT(1,1,2*J),HIR(1,1,J))	VISPR
	IF (IABS(IPIN(J)).EQ.4) GO TO 90	VISPR
C		SLIP
C	ZERO T1-T9 ARRAYS AND HAD,HBD,WIJM,CV,CS4,CSB AND TQC.	VISPR
C		VISPR
	WIJM = 0.0	VISPR
	HAC = 0.0	BUTLER1
	CV = 0.0	VISPR
	CSA = 0.0	VISPR
	CSB = 0.0	VISPR
	TQC = 0.0	VISPR
	CALL DOT33 (D(1,1,I),HT(1,1,2*J-1),DH1)	VISPR
	CALL DOT33 (DH1,HIR(1,1,J),HD3)	VISPR
	DO 220 L=1,3	TGMOD6
	DO 220 K=1,3	TGMOD6
	IF(DABS(HD3(L,K)).LT.EPS(10)) HD3(L,K) = 0.DO	TGMOD6
220	CONTINUE	TGMOD6
	HAD = HD3(3,3)	VISPR
	IF (HAD.GT. 1.0) HAD = 1.0	VISPR
	IF (HAD.LT. -1.0) HAD = -1.0	VISPR
	ANGL(1) = DACOS(HAD)	VISPR
	IF ((HD3(2,3).NE.0.0 .OR. HD3(1,3).NE.0.0).AND. IABS(IPIN(J)).NE.7)SLIP	VISPR
	*ANGL(2) = DATAN2(HD3(2,3),HD3(1,3))	VISPR
	ANGL(3) = DATAN2(HD3(2,1)-HD3(1,2),HD3(1,1)+HD3(2,2))	VISPR
	IF(NPRT(12).NE.0.AND. IPIN(J).LT.0) WRITE (6,739) J,I,ANGL,	TGMOD6
	*((D(L,K,J+1),K-1,3),(HT(L,K,2*J),K-1,3),(HIR(L,K,J),K-1,3),L-1,3),	TGMOD6
	*((D(L,K,I),K-1,3),(HT(L,K,2*J-1),K-1,3),(DH1(L,K),K-1,3),L-1,3),	TGMOD6
	* ((HD3(L,K),K-1,3),L-1,3)	TGMOD6
739	FORMAT(1H0,'J= ',I2,1X,'I= ',I2,3(2X,D14.7),/,	TGMOD6
	* 2(3(9(1X,D13.6),/),/),3(3(2X,D18.12),/))	TGMOD6
	IF (IPIN(J).LT.0) GO TO 41	VISPR
	IF (NJ.NE.0.AND. IJ.EQ.4) GO TO 27	VISPR
C		VISPR
C	CONVERT TO INERTIAL REFERENCE SYSTEM	VISPR
C	T1= D(I)*HA(NJ) T4=D(J+1)*HA(MJ)	VISPR
C	T3= D(I)*WMEG(I) T6=D(J+1)*WMEG(J+1)	VISPR
C		VISPR
C	HAD = COS TA = T1.T4	VISPR
C	WIJ = T3-T6	VISPR
C	WJ = !WIJ!	VISPR
C		VISPR
	DO 20 L=1,3	VISPR
	DO 15 M=1,3	VISPR
	T3(L) = T3(L)+ D(M,L,I)* WMEG(M,I)	VISPR
15	T6(L) = T6(L)+ D(M,L,J+1)* WMEG(M,J+1)	VISPR
	WIJ(L)= T3(L)-T6(L)	VISPR
20	WIJM = WIJM + WIJ(L)**2	VISPR
	WIJM = DSQRT(WIJM)	VISPR
	IF (WIJM.LE.EPS(12)) WIJM = 0.0	MISDOT

```

WJ(J) = WIJM                                VISPR
C                                             VISPR
C T7 = T1 X T4                                VISPR
C HAC = !T7!                                  VISPR
C                                             VISPR
CALL CROSS (DH1(1,3),HIR(1,3,J),T7)         VISPR
HACC = T7(1)**2 + T7(2)**2 + T7(3)**2       VISPR
HAC = DSQRT(HACC)                             VISPR
C                                             VISPR
C COMPUTE CV, THE MAGNITUDE OF VISCOUS AND COULOMB TORQUE/WIJM VISPR
C RA = +SGN TA DOT = -WIJ.T7                 VISPR
C AND CSA, THE MAGNITUDE OF FLEXURE TORQUE/HAC VISPR
C                                             VISPR
CV = VISCOS(WIJM,VISC(1,3*J-2),HA2)         VISPR
IF (NJ.EQ.0) HA(2,2*J) = HA2                 VISPR
CREST = VISC(7,3*J-2)                       VISPR
RA = -(WIJ(1)*T7(1) + WIJ(2)*T7(2) + WIJ(3)*T7(3)) VISPR
IF (HAC.LT.EPS(12)) RA=0.0                  MISDOT
IF (HAC.GE.EPS(12)) RA=RA/HAC               MISDOT
JSTP = 0                                      VISPR
IF (IPIN(J).EQ.7) GOTO 25                    SLIP
IF (JOINTF(J).EQ.0) CSA = EFUNCT(ANGL(1),RA,SPRING(1,3*J-2),JSTP) VISPR
IF (JOINTF(J).NE.0) CSA = FINTERP(ANGL(1),ANGL(2),JOINTF(J)) VISPR
IF (HAC.LT.EPS(12)) CSA=0.0                 MISDOT
IF (HAC.GE.EPS(12)) CSA=CSA/HAC             MISDOT
25 IF (NJ.EQ.0) JSTOP(1,1,J) = JSTP          SLIP
IF (IPIN(J).EQ.1) GO TO 34                  VISPR
IF (IPIN(J).EQ.6) GOTO 34                  SLIP
C                                             VISPR
C RB = +SGN TB DOT = -WIJ.T8                 VISPR
C COMPUTE CSB, THE MAGNITUDE OF TORSIONAL TORQUE/HBC VISPR
C                                             VISPR
RB = -(WIJ(1)*HIR(1,3,J) + WIJ(2)*HIR(2,3,J) + WIJ(3)*HIR(3,3,J)) VISPR
CSB = EFUNCT(ANGL(3),RB,SPRING(1,3*J-1),JSTP) VISPR
IF (NJ.EQ.0) JSTOP(2,1,J) = JSTP           VISPR
IF (NJ.GT.0) GO TO 34                      VISPR
C                                             VISPR
C COMPUTE EFFECT OF GLOBALGRAPHIC JOINT STOP (IPIN=3) VISPR
C                                             VISPR
27 IF (IPIN(J).NE.3) GO TO 34              VISPR
CALL GLOBAL (J,HD3(1,3),DH1,TQC,T9,ANGL)   VISPR
C                                             VISPR
C COMPUTE TOTAL TORQUE IN INERTIAL REFERENCE BY VISPR
C TQ = -CV*WIJ + CSA*T7 + CSB*T8 + TQC*T9   VISPR
C                                             VISPR
34 IF (NJ.EQ.0) GO TO 35                    JDRIFT
CV = 0.0                                    VISPR
IF (IJ.NE.1) CSA = 0.0                     VISPR
IF (IJ.NE.2) CSB = 0.0                     VISPR
IF (IJ.NE.4) TQC = 0.0                     VISPR
35 IF (HA(2,2*J).EQ.0.0) GO TO 36          JDRIFT
CALL MAT31 (HIR(1,1,J),HA(1,2*J-1),TQ(1,J)) VISPR

```

```

DO 38 L=1,3                                VISPR
38 TQ(L,J) = HA(2,2*J)*TQ(L,J)             VISPR
36 DO 37 L=1,3                                VISPR
   TQ(L,J) = TQ(L,J) - CV*WIJ(L) + CSA*T7(L) + CSB*HIR(L,3,J) + TQC*T9(L) VISPR
37 TTI(L) = TQ(L,J)                          VISPR
   IF (NPRT(12).NE.0) WRITE (6,39)         VISPR
   *           J,CV,CSA,CSB,HAC,RA,RB,(TQ(L,J),L-1,3), VISPR
   *           WIJ,T7,ANGL, DH1 , HD3,     VISPR
   *           ((HIR(L,K,J),L-1,3),K-1,3)  VISPR
39 FORMAT (1H0,I3,3F14.3,6F14.6/(4X,9F14.6)) VISPR
C
C   ADD TORQUE CONVERTED TO LOCAL REFERENCE BY VISPR
C   U2I = U2I + DI*TQ                        VISPR
C   U2J = U2J - DJ*TQ                        VISPR
C
DO 40 L=1,3                                VISPR
DO 40 M=1,3                                VISPR
   U2(L,I ) = U2(L,I ) + D(L,M,I )*TQ(M,J) VISPR
40 U2(L,J+1) = U2(L,J+1) - D(L,M,J+1)*TQ(M,J) VISPR
C
C   STORE DATA FOR OUTPUT ROUTINE INTO PRJNT ARRAY. VISPR
C
41 PRJNT(1,J) = IPIN(J)                     VISPR
   PRJNT(2,J) = ANGL(1)                     VISPR
   PRJNT(3,J) = ANGL(2)                     VISPR
   PRJNT(4,J) = ANGL(3)                     VISPR
   PRJNT(5,J) = (CSA*HAC)**2 + CSB**2      VISPR
   PRJNT(6,J) = (CV*WIJM)**2               VISPR
   PRJNT(7,J) = TQ(1,J)**2 + TQ(2,J)**2 + TQ(3,J)**2 VISPR
90 CONTINUE                                 VISPR
   CALL ELTIME(2,13)                         VISPR
99 RETURN                                   VISPR
   END                                       VISPR

```

```

SUBROUTINE WINDY(MMM,MM,N,NN,NT)
C
C
C REV IV 07/23/86TWOPI
C COMPUTES FORCES AND TORQUES ADDING THEM TO THE U1 AND U2 ARRAYS
C OF WIND BLAST FORCES DETERMINED BY FUNCTION STORED IN TAB(NT)
C ON ELLIPSOID (MM) ATTACHED TO BODY SEGMENT (M) WHICH EXTENDS
C THROUGH THE INTERSECTING PLANE (NN) ATTACHED TO SEGMENT (N).
C
C
C IMPLICIT REAL*8 (A-H,O-Z)
C COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,
* NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36),NPG
C COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),
* SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)
C COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(1250),TAB(4500)DIMENB
C COMMON/WINDFR/ WTIME(30),QFU(3,5),QFV(3,5),WF(3,30),IWIND(30),
* MWSEG(7,30),NFVSEG(6),NFVNT(5),MOWSEG(30,30)
C COMMON/CNTRSFR/ PL(24,30),BELT(20,8),TPTS(6,8),BD(24,40)
C COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),
* UNITL,UNITM,UNITT,GRAVTY(3),TWOPI
C COMMON/TEMPVS/ DMNT(3,3),XMN(3),XMM(3),TM(3),BET,BTS,P,FT(3),
* FF(3),AF(3),FAF,TF,BREF,SCALE,TRACER,AREA,RLM(3),
* TQM(3),RM(3),DD(3,3),DDD(3,3),R(3,3),DVP(3,3),
* SI(3,15),R2(2,3),TTF(3),FFT(3),AM(3,3),VP(3),
* SS(3),SM(3),SN1(3),AS(3),BTE,XNORM,TEMP,
* X,Y,AI(3,3,15),RYC,AMDA1,AMDA2,B1,B2,RXC
* ,DMMY(34805)
C
C
C MMM=0 CALCULATE NFORCE
C MMM>0 WIND FORCE CALCULATED USING ENTIRE AREA METHOD
C MMM<0 WIND FORCE CALCULATED USING GRID METHOD
C (ALLOWS BLOCKING SEGMENTS)
C
C DATA NSTEPS/4/
C CALL ELTIME(1,37)
C M=IABS(MMM)
C IF (MMM.EQ.0) GO TO 50
C
C COMPUTE PENETRATION DISTANCE; IF NEGATIVE, RETURN.
C
C CALL DOTT33 (D(1,1,M),D(1,1,N),DMNT)
C DO 10 I=1,3
10 XMN(I) = SEGLP(I,M) - SEGLP(I,N)
C CALL MAT31 (D(1,1,M),XMN,XMM)
C CALL MAT31 (DMNT,PL(1,NN),TM)
C BET = PL(4,NN)
C DO 11 I=1,3
11 BET = BET - TM(I)*(BD(I+3,MM)+XMM(I))
C CALL MAT31 (BD(16,MM),TM,RM)
C BTS = TM(1)*RM(1) + TM(2)*RM(2) + TM(3)*RM(3)
C BTE = -DSQRT(BTS)
C P = BET - BTE
C IF (P.LT.0.0) GO TO 99
C
C
C

```

C	FETCH OR STORE INITIAL PENETRATION TIME.	WINDY
C		WINDY
	IWIND(M) = M	WINDY
	IF (TIME.LE.WTIME(M)) WTIME(M) = TIME	WINDY
	FTIME = TIME - WTIME(M)	WINDY
C		WINDY
C	GET DRAG COEFFICIENT CD FROM TABLE NTC FOR TIME = FTIME.	WINDOP
C		WINDOP
	CD=1.0	WINDOP
	NTC=MWSEG(6,M)	WINDOP
	IF (NTC.EQ.0) GOTO 20	WINDOP
	KT=NTI(NTC)	WINDOP
	NENTRY=TAB(KT+5)	WINDOP
	K1=KT+10	WINDOP
	K2=4*NENTRY+KT+2	WINDOP
	IF (NENTRY.EQ.1) GOTO 18	WINDOP
	DO 17 K=K1,K2,4	WINDOP
	IF (FTIME.GT.TAB(K)) GOTO 17	WINDOP
	KK=K	WINDOP
	R1=(TAB(K)-FTIME)/(TAB(K)-TAB(K-4))	WINDOP
	GOTO 19	WINDOP
17	CONTINUE	WINDOP
18	KK=K2	WINDOP
	R1=0.0	WINDOP
19	R22=1.0-R1	WINDOP
	K=KK+1	WINDOP
	CD=R22*TAB(K)+R1*TAB(K-4)	WINDOP
C		WINDOP
C	GET FORCE VECTOR FT	WINDOP
C		WINDOP
C	RK=0 TIME DEPENDENT WIND FORCE FROM TABLE	WINDOP
C	RK#0 VELOCITY DEPENDENT WIND FORCE	WINDOP
C		WINDOP
20	KT = NTI(NT)	WINDOP
	RK=TAB(KT)	WINDOP
	IF (RK.EQ.0.0) GOTO 13	WINDOP
	C=TAB(KT+1)	WINDOP
	PR=TAB(KT+2)	WINDOP
	NSV=IDINT(TAB(KT+3))	WINDOP
	NSR=IDINT(TAB(KT+4))	WINDOP
	DO 12 I=1,3	WINDOP
	V=SEGLV(I,NSV)-SEGLV(I,NSR)	WINDOP
12	FT(I)=DSIGN(0.5DO,-V)*CD*RK*PR*V**2/C**2	WINDOP
	GOTO 14	WINDOP
13	NSR=IDINT(TAB(KT+4))	WINDOP
	NENTRY = TAB(KT+5)	WINDY
	K1 = KT+10	WINDY
	K2 = 4*NENTRY + KT+2	WINDY
	IF (NENTRY.EQ.1) GO TO 31	WINDY
	DO 30 K=K1,K2,4	WINDY
	IF (FTIME.GT.TAB(K)) GO TO 30	WINDY
	KK = K	WINDY

	R1 = (TAB(K)-FTIME)/(TAB(K)-TAB(K-4))	WINDY
	GO TO 32	WINDY
30	CONTINUE	WINDY
31	KK = K2	WINDY
	R1 = 0.0	WINDY
32	R22= 1.0 - R1	WINDOP
	DO 33 I=1,3	WINDY
	K= KK+I	WINDY
33	FT(I)=(R22*TAB(K) + R1*TAB(K-4))*CD	WINDOP
	IF (NSR.EQ.0) GOTO 14	WINDOP
	CALL DOT31(D(1,1,NSR),FT,FF)	WINDOP
	DO 21 I=1,3	WINDOP
21	FT(I)=FF(I)	WINDOP
14	IF (MMM.LT.0) GOTO 15	WINDOP
C		WINDY
C	COMPUTE PRESENTED AREA TO WIND FORCE.	WINDY
C		WINDY
	CALL MAT31 (D(1,1,M),FT,FF)	WINDY
	CALL MAT31 (BD(7,MM),FF,AF)	WINDY
	FAF = FF(1)*AF(1) + FF(2)*AF(2) + FF(3)*AF(3)	WINDY
	IF (FAF.LE.0.0) GO TO 99	WINDY
	TF = TM(1)*FF(1) + TM(2)*FF(2) + TM(3)*FF(3)	WINDY
	BREF=0.0	CCWIND
	TEMP=BTS-TF*TF/FAF	CCWIND
	IF(TEMP.GT.0.0) BREF =DSQRT(TEMP)	CCWIND
	SCALE = (-BET+BREF)/(-BTE+BREF)	WINDY
	IF (SCALE.GE.1.0) GO TO 99	WINDY
	IF (SCALE.LT.0.0) SCALE = 0.0	WINDY
	TRACER = (BD(7,MM)-AF(1)**2/FAF)*(BD(11,MM)-AF(2)**2/FAF)	WINDY
*	+ (BD(7,MM)-AF(1)**2/FAF)*(BD(15,MM)-AF(3)**2/FAF)	WINDY
*	+ (BD(11,MM)-AF(2)**2/FAF)*(BD(15,MM)-AF(3)**2/FAF)	WINDY
*	- (BD(8,MM)-AF(1)*AF(2)/FAF)**2	WINDY
*	- (BD(9,MM)-AF(1)*AF(3)/FAF)**2	WINDY
*	- (BD(12,MM)-AF(2)*AF(3)/FAF)**2	WINDY
	AREA = (1.0-SCALE**2) * PI / DSQRT(TRACER)	WINDY
C		WINDY
C	ADD FORCE AND TORQUES TO U1 AND U2 ARRAYS FOR SEGMENT M.	WINDY
C		WINDY
	SCALE = SCALE/BTE	WINDY
	DO 36 I=1,3	WINDY
	RLM(I) = RM(I)*SCALE + BD(I+3,MM)	WINDY
	FT (I) = FT(I)*AREA	WINDY
36	FF (I) = FF(I)*AREA	WINDY
	CALL CROSS (RLM,FF,TQM)	WINDY
	DO 39 I=1,3	WINDY
	WF(I,M)=FT(I)	WINDOP
	U1(I,M) = U1(I,M) + FT(I)	WINDY
39	U2(I,M) = U2(I,M) + TQM(I)	WINDY
	IF (NPRT(14).NE.0) WRITE (6,41) TIME,M,P,AREA,FT,TQM	WINDY
41	FORMAT(' WIND FORCE' ,F14.6,I6,2F10.3,3X,3F12.5,3X,3F12.5)	WINDY
	GO TO 99	WINDY
C		WINDY

C	USE GRID TO CALCULATE WIND FORCE	WINDOP
C	VP - ORIGIN OF WIND	WINDOP
C		WINDOP
15	AREAT=0.0	WINDOP
	DO 16 I=1,3	WINDOP
	TTF(I)=0.0	WINDOP
	TQM(I)=0.0	WINDOP
16	VP(I) = -FT(I)*10000.0	WINDOP
	TEMP=FT(1)**2+FT(2)**2+FT(3)**2	WINDOP
	IF (TEMP.EQ.0.0) GOTO 99	WINDOP
	CALL MAT31(D(1,1,M),FT,FF)	WINDOP
	TEMP = 0.0	WINDOP
	IF (FT(1).NE.0.0.OR.FT(2).NE.0.0) GOTO 150	WINDOP
C		WINDOP
C	CALCULATE DIRECTION COSINE MATRIX FOR VP COORD. SYS.	WINDOP
C		WINDOP
	DO 140 I=1,3	WINDOP
	DO 140 J=1,3	WINDOP
140	DVP(I,J)=0.0	WINDOP
	DVP(1,2)=1.0	WINDOP
	DVP(2,1)=1.0	WINDOP
	DVP(3,3)=-1.0	WINDOP
	GO TO 141	WINDOP
150	CONTINUE	WINDOP
	DO 110 I=1,3	WINDOP
110	TEMP=TEMP+FT(I)*FT(I)	WINDOP
	TEMP = DSQRT(TEMP)	WINDOP
	XNORM = DSQRT(FT(1)*FT(1)/TEMP**2+FT(2)*FT(2)/TEMP**2)	WINDOP
	DVP(1,1) = FT(2)/(XNORM*TEMP)	WINDOP
	DVP(1,2) = -FT(1)/(XNORM*TEMP)	WINDOP
	DVP(1,3) = 0.0	WINDOP
	DVP(2,1) = FT(1)*FT(3)/(XNORM*TEMP*TEMP)	WINDOP
	DVP(2,2) = FT(2)*FT(3)/(XNORM*TEMP*TEMP)	WINDOP
	DVP(2,3) = -XNORM	WINDOP
	DO 130 I=1,3	WINDOP
130	DVP(3,I) = FT(I)/TEMP	WINDOP
141	CONTINUE	WINDOP
	MOELP = MWSEG(7,M)	WINDOP
C		WINDOP
C	PROJECT MM ELLIPSOID UNTO VP-PLANE	WINDOP
C	AS - PROJECTED ELLIPSE MATRIX	WINDOP
C		WINDOP
	CALL DOTT33(D(1,1,M),DVP,DD)	WINDOP
	CALL MAT33(BD(7,MM),DD,DDD)	WINDOP
	CALL DOT33(D(1,1,M),DDD,DD)	WINDOP
	CALL MAT33(DVP,DD,AM)	WINDOP
	DO 101 K=1,3	WINDOP
101	SS(K)=SEGLP(K,M)+BD(K+3,MM)-VP(K)	WINDOP
	CALL MAT31(DVP,SS,SM)	WINDOP
	DO 114 K=1,3	WINDOP
	IF (DABS(SM(K)).LT.EPS(5)) SM(K)=DSIGN(EPS(5),SM(K))	WINDOP
114	CONTINUE	WINDOP


```

CALL SOLVR(AM(1,1),AM(2,1),AM(3,1),AM(1,3),AM(2,3),AM(3,3),      WINDOP
*      AM(1,1),AM(1,3),SM,R(1,1),R(3,1))                          WINDOP
CALL SOLVR(AM(1,2),AM(2,2),AM(3,2),AM(1,3),AM(2,3),AM(3,3),      WINDOP
*      AM(2,2),AM(2,3),SM,R(2,2),R(3,2))                          WINDOP
CALL SOLVR(AM(1,1)+AM(1,2),AM(2,1)+AM(2,2),AM(3,1)+AM(3,2),      WINDOP
*      AM(1,3),AM(2,3),AM(3,3),AM(1,1)+2.0*AM(1,2)+AM(2,2),      WINDOP
*      AM(1,3)+AM(2,3),SM,R(1,3),R(3,3))                          WINDOP
R(2,1)=0.0                                                         WINDOP
R(1,2)=0.0                                                         WINDOP
R(2,3)=R(1,3)                                                     WINDOP
DO 102 K=1,3                                                       WINDOP
DO 102 J=1,2                                                       WINDOP
102  R2(J,K)=R(J,K)                                               WINDOP
CALL SOLVA(R2,AS(1),AS(2),AS(3))                                  WINDOP
C                                                                    WINDOP
C  GET MAJOR & MINOR AXES OF PROJECTED ELLIPSE                    WINDOP
C                                                                    WINDOP
TEMP=(AS(1)+AS(2))**2-4.0*(AS(1)*AS(2)-AS(3)**2)                 WINDOP
IF (TEMP.LT.0.0) TEMP=0.0                                         WINDOP
TEMP = DSQRT(TEMP)                                                WINDOP
AMDA1=(AS(1)+AS(2)+TEMP)/2.0                                       WINDOP
AMDA2=(AS(1)+AS(2)-TEMP)/2.0                                       WINDOP
R2(1,1)=AS(3)                                                       WINDOP
R2(2,1)=AMDA1-AS(1)                                                 WINDOP
R2(1,2)=AMDA2-AS(2)                                                 WINDOP
R2(2,2)=AS(3)                                                       WINDOP
AMDA1=DABS(AMDA1)                                                  WINDOP
AMDA2=DABS(AMDA2)                                                  WINDOP
B1=DSQRT(1.0/(AMDA1*(R2(1,1)**2+R2(1,2)**2)))                       WINDOP
B2=DSQRT(1.0/(AMDA2*(R2(2,1)**2+R2(2,2)**2)))                       WINDOP
R2(1,1)=R2(1,1)*B1                                                 WINDOP
R2(1,2)=R2(1,2)*B2                                                 WINDOP
R2(2,1)=R2(2,1)*B1                                                 WINDOP
R2(2,2)=R2(2,2)*B2                                                 WINDOP
C                                                                    WINDOP
C  GET BLOCKING ELLIPSOIDS IN VP COORD. SYS.                      WINDOP
C                                                                    WINDOP
DO 103 MI=1,MOELP                                                  WINDOP
I=MOWSEG(M,MI*2-1)                                                 WINDOP
II=MOWSEG(M,MI*2)                                                  WINDOP
CALL DOT33(D(1,1,I),DVP,DD)                                         WINDOP
CALL MAT33(BD(7,II),DD,DDD)                                         WINDOP
CALL DOT33(D(1,1,I),DDD,DD)                                         WINDOP
CALL MAT33(DVP,DD,AI(1,1,MI))                                       WINDOP
DO 104 K=1,3                                                       WINDOP
104  SS(K)=SEGLP(K,I)+BD(K+3,II)-VP(K)                             WINDOP
CALL MAT31(DVP,SS,SI(1,MI))                                         WINDOP
DO 115 K=1,3                                                       WINDOP
IF (DABS(SI(K,MI)).LT.EPS(6)) SI(K,MI)=DSIGN(EPS(6),SI(K,MI))    WINDOP
115  CONTINUE                                                       WINDOP
103  CONTINUE                                                       WINDOP
C                                                                    WINDOP

```

```

C      SET-UP GRID AND CHECK EACH RECTANGLE CENTER POINT      WINDOP
C
AREA=DSQRT((R2(1,1)**2+R2(2,1)**2)*(R2(1,2)**2+R2(2,2)**2))    WINDOP
AREA=AREA/NSTEPS**2                                          WINDOP
IN=2*NSTEPS+1                                               WINDOP
DO 105 I=1, IN                                             WINDOP
RXC=R2(1,1)-R2(1,1)*(I-1)/NSTEPS                            WINDOP
RYC=R2(2,1)-R2(2,1)*(I-1)/NSTEPS                            WINDOP
DO 106 J=1, IN                                             WINDOP
RM(1)=(RXC-R2(1,2)*(NSTEPS-J+1)/NSTEPS)*0.9999             WINDOP
RM(2)=(RYC-R2(2,2)*(NSTEPS-J+1)/NSTEPS)*0.9999             WINDOP
TM(1)=AM(3,3)                                               WINDOP
TM(2)=2.0*(RM(1)*AM(1,3)+RM(2)*AM(2,3))                     WINDOP
TM(3)=RM(1)**2*AM(1,1)+RM(2)**2*AM(2,2)+2.0*RM(1)*RM(2)*AM(1,2)-1. WINDOP
TEMP=TM(2)**2-4.0*TM(1)*TM(3)                                WINDOP
IF (TEMP.LT.0.0) GOTO 106                                    WINDOP
B1=(DSQRT(TEMP)-TM(2))/(2.0*TM(1))                           WINDOP
B2=- (DSQRT(TEMP)+TM(2))/(2.0*TM(1))                         WINDOP
RM(3)=B1                                                      WINDOP
IF (B2.LT.B1) RM(3)=B2                                        WINDOP
SN1(1)=RM(1)+SM(1)                                           WINDOP
SN1(2)=RM(2)+SM(2)                                           WINDOP
SN1(3)=RM(3)+SM(3)                                           WINDOP
CALL DOT31(DVP,SN1,XMM)                                       WINDOP
C
C      CHECK FOR PENETRATION                                  WINDOP
C
DO 107 K=1,3                                                 WINDOP
107  XMN(K)=VP(K)-SEGLP(K,N)+XMM(K)                           WINDOP
CALL MAT31(D(1,1,N),XMN,XMM)                                  WINDOP
BET=PL(4,NN)                                                  WINDOP
BTS=PL(1,NN)*XMM(1)+PL(2,NN)*XMM(2)+PL(3,NN)*XMM(3)        WINDOP
IF (BTS.GT.BET) GOTO 106                                     WINDOP
C
C      CHECK FOR BLOCKING ELLIPSOIDS                          WINDOP
C
DO 109 IM=1,MOELP                                           WINDOP
X=SN1(1)-SI(1,IM)                                           WINDOP
Y=SN1(2)-SI(2,IM)                                           WINDOP
TM(1)=AI(3,3,IM)                                             WINDOP
TM(2)=2.0*(AI(1,3,IM)*X+AI(2,3,IM)*Y)                         WINDOP
TM(3)=AI(1,1,IM)*X**2+AI(2,2,IM)*Y**2+2.0*AI(1,2,IM)*X*Y-1.0 WINDOP
TEMP=TM(2)**2-4.0*TM(1)*TM(3)                                WINDOP
IF (TEMP.LT.0.0) GOTO 109                                    WINDOP
B1=(-TM(2)+DSQRT(TEMP))/(2.0*TM(1))                           WINDOP
B2=(-TM(2)-DSQRT(TEMP))/(2.0*TM(1))                           WINDOP
IF (B2.LT.B1) B1=B2                                          WINDOP
SNZ=B1+SI(3,IM)                                              WINDOP
IF (SNZ.LT.SN1(3)) GOTO 106                                  WINDOP
109  CONTINUE                                                 WINDOP
CALL DOT31(DVP,RM,SS)                                         WINDOP
CALL MAT31(D(1,1,M),SS,RM)                                    WINDOP

```

C		WINDOP
C	SUM FORCES & TORQUES	WINDOP
C		WINDOP
	AREAT=AREAT+AREA	WINDOP
	DO 111 K=1,3	WINDOP
	TTF(K)=FT(K)*AREA+TTF(K)	WINDOP
	RM(K)=RM(K)+BD(K+3,MM)	WINDOP
111	FFT(K)=FF(K)*AREA	WINDOP
	CALL CROSS(RM,FFT,TM)	WINDOP
	DO 112 K=1,3	WINDOP
112	TQM(K)=TQM(K)+TM(K)	WINDOP
106	CONTINUE	WINDOP
105	CONTINUE	WINDOP
C		WINDOP
C	ADD FORCE & TORQUE TO U1 & U2 ARRAYS FOR SEGMENT M	WINDOP
C		WINDOP
	IF (NPRT(14).NE.0) WRITE(6,200) TIME,M,AREAT,TTF,TQM	WINDOP
200	FORMAT(' WIND FORCE',F14.6,I6,I3X,F10.3,3F12.5,3X,3F12.5)	WINDOP
	DO 113 I=1,3	WINDOP
	WF(I,M)=TTF(I)	WINDOP
	U1(I,M)=U1(I,M)+TTF(I)	WINDOP
113	U2(I,M)=U2(I,M)+TQM(I)	WINDOP
	GO TO 99	WINDOP
C		WINDOP
C	M = 0: CALCULATE FORCE FUNCTIONS.	WINDOP
C		WINDY
50	NFORCE = NRVSEG(6)	WINDY
	DO 60 J=1,NFORCE	WINDY
	NFS = IABS(NRVSEG(J))	WINDY
	NFT = IABS(NRVNT(J))	WINDY
	KFT = NTV(NFT)	WINDY
	FRCE = EVALFD(TIME,KFT,1)	WINDY
	IF (NVRSEG(J).GT.0) GO TO 52	WINDY
	DO 51 I=1,3	WINDY
51	U2(I,NFS) = U2(I,NFS) + FRCE*QFU(I,J)	WINDY
	GO TO 60	WINDY
52	CALL DOT31 (D(1,1,NFS),QFU(1,J),TM)	WINDY
	DO 53 I=1,3	WINDY
	U1(I,NFS) = U1(I,NFS) + FRCE*TM(I)	WINDY
53	U2(I,NFS) = U2(I,NFS) + FRCE*QFV(I,J)	WINDY
60	CONTINUE	WINDY
99	CALL ELTIME (2,37)	WINDY
	RETURN	WINDY
	END	WINDY

	DOUBLE PRECISION FUNCTION XDY(X,D,Y)		XDY
C		REV IV	07/23/86JTF786
C	FUNCTION ROUTINE TO COMPUTE X.DY OR Y.D'X		XDY
C			XDY
	IMPLICIT REAL*8(A-H,O-Z)		XDY
	DIMENSION X(3),D(3,3),Y(3)		XDY
	XDY = 0.0		XDY
	DO 10 I=1,3		XDY
10	XDY = XDY + X(I)*(D(I,1)*Y(1)+D(I,2)*Y(2)+D(I,3)*Y(3))		JTF786
	RETURN		XDY
	END		XDY

	SUBROUTINE YPRDEG(D,A)		REV IV	11/26/86	YPRDEG
C					YPRFIX
C	COMPUTES YAW PITCH AND ROLL IN DEGREES AND PLACES THEM				YPRDEG
C	INTO THE A ARRAY FOR A GIVEN DIRECTION COSINE MATRIX D.				YPRDEG
C					YPRDEG
C	ASSUMES D - D(R)D(P)D(Y) , WHERE				YPRDEG
C					YPRDEG
C	1 0 0 CP 0 -SP CY SY 0				YPRDEG
C	D(R) - 0 CR SR ,D(P) - 0 1 0 AND D(Y) - -SY CY 0				YPRDEG
C	0 -SR CR SP 0 CP 0 0 1				YPRDEG
C					YPRDEG
	IMPLICIT REAL*8(A-H,O-Z)				YPRDEG
	DIMENSION A(3),D(3,3)				YPRDEG
	COMMON/CNSNTS/ PI,RADIAN,G,THIRD, EPS(24),				YPRDEG
	* UNITL,UNITM,UNITT,GRAVTY(3),TWOPI				TWOPI
	IF (DABS(D(1,1)).LE.EPS(15).AND.DABS(D(1,2)).LE.EPS(15))GOTO10				YPRFIX
	IF (DABS(D(2,3)).LE.EPS(15).AND.DABS(D(3,3)).LE.EPS(15))GOTO10				YPRFIX
	YAW = DATAN2(D(1,2),D(1,1))				YPRDEG
	ROLL = DATAN2(D(2,3),D(3,3))				YPRDEG
	GO TO 11				YPRDEG
10	YAW = DATAN2(-D(2,1),D(2,2))				YPRDEG
	ROLL = 0.0				YPRDEG
11	PITCH = -DASIN(D(1,3))				YPRDEG
	IF (DABS(ROLL).LE.0.5*PI) GO TO 20				YPRDEG
	IF (DABS(YAW).LE.0.5*PI) GO TO 20				YPRDEG
	PITCH = DSIGN(PI-DABS(PITCH),PITCH)				YPRDEG
	YAW = DATAN2(-D(1,2),-D(1,1))				YPRDEG
	ROLL = DATAN2(-D(2,3),-D(3,3))				YPRDEG
20	A(1) = YAW/RADIAN				YPRDEG
	A(2) = PITCH/RADIAN				YPRDEG
	A(3) = ROLL/RADIAN				YPRDEG
	RETURN				YPRDEG
	END				YPRDEG

APPENDIX J

386-VIEW Program Listing

```

PROGRAM VIEW V1.1
COMMON/PLTT/SFACTR, INT, TIME, ICOLOR(91), OFSETX, OFSETY, ZTIME V1.1
COMMON/INTERS/ NIE(90), IE(90, 90) V1.1
COMMON/ELLIPSE/NSTEPS(90), IELP, A(3, 3, 30), SEGLP(3, 90), VP(3), V1.1
*D(3, 3, 90), DVP(3, 3), RA(3), NSEG, SEGLPO(3, 90) V1.2
COMMON/POLYGON/NPLANE, IFLAG, NPPP(90), PO(3, 4, 60), P(3, 4, 60), V1.1
*CONVEC(2, 4, 90), POS(2, 90), SIGN(90) V1.1
COMMON/ATB/PL(17, 30) V1.1
COMMON/VIEWP/VP0(3), DVPO(3, 3), IVP, VP2(3) V1.1
COMMON/DEBUG/IDEBUG(80), NISG, DEVFLG, ONLINE, TERM, BDRS, OFFLINE V1.1
COMMON/DEVICE/IDEF, IOPORT, MODEL 80386
DIMENSION WORK(10000) V1.1
CHARACTER*4 ID(10) 80386
INTEGER DEVFLG, ONLINE, TERM, BDRS, OFFLINE V1.1
DOUBLE PRECISION ZTIME, CTIME, STIME, DTIME, ETIME V1.1
CHARACTER*12 FILE1, FILE5, FILE6 80386
C V1.1
C PROGRAM VIEW VERSION 1.1 LAST MOD. - MARCH 9, 1983 V1.1
C VERSION 1.2 LAST MOD. - DECEMBER 18, 1984 V1.2
C WRITTEN BY SRL IN SUPPORT OF THE V1.1
C MODELING AND ANALYSIS BRANCH OF AMRL V1.1
C AT WPAFB. V1.1
C 80386
C 80386 IMPLEMENTATION BY KETRON, INC. - OCTOBER 31, 1989 80386
C V1.1
C STORED ON TAPE1 THAT IS OUTPUT FROM THE ATBM VERSION V5D. V1.1
C V1.1
C THIS PROGRAM USES CONTOUR LINES TO REPRESENT THE 3-D PROPERTIES OF V1.1
C THE DATA ON TAPE1. THE CONTOUR LINES ARE PLOTTED ON PAPER THAT V1.1
C REPRESENTS THE PROJECTION PLANE. THE POINTS THAT COMPOSE A CONTOUR V1.1
C LINE IN 3-SPACE ARE PROJECTED THROUGH A POINT ON TO THE PROJECTION V1.1
C PLANE. V1.1
C V1.1
C CURRENTLY THERE ARE TWO CLASSES OF OBJECTS THAT ARE PLOTTED USING V1.1
C CONTOUR LINES. V1.1
C CLASS 1 - ELLIPSOIDS V1.1
C ELLIPSOIDS ARE USED TO REPRESENT BODY SEGMENTS. THIS PROGRAM ALLOWS V1.1
C ELLIPSOIDS TO BE IMBEDDED IN OTHER ELLIPSOIDS. V1.1
C V1.1
C CLASS 2 - CONVEX POLYGONS V1.1
C CONVEX POLYGONS ARE USED TO REPRESENT OBJECTS THAT CAN BE DEFINED V1.1
C BY A SET OF PLANES. ALL POLYGONS DEFINED BY THE INPUT MUST BE V1.1
C CONVEX POLYGONS; CONCAVE POLYGONS CAN BE OBTAINED USING A V1.1
C COMBINATION OF CONVEX POLYGONS. V1.1
C V1.1
C THE HIDDEN LINE ROUTINES CHECK FOR POINTS HIDDEN BY V1.1
C ELLIPSOIDS OR POLYGONS. THESE ROUTINES MUST CHECK V1.1
C FOR ANY POSSIBLE OBJECT THAT MAY BE BLOCKING THE V1.1
C CURRENT POINT AS SEEN FROM THE VIEWPOINT. V1.1
C IN ORDER TO ELIMINATE CHECKING ALL OBJECTS FOR V1.1
C EACH POINT, SUBROUTINES ARE INCLUDED IN THE V1.1
C VIEW PROGRAM THAT DETECT OBJECT OVERLAP ON THE V1.1

```

```

C      PROJECTION PLANE. BEFORE THE PLOTTING PHASE OF THE          V1.1
C      VIEW PROGRAM, OBJECTS WHICH OVERLAP EACH OTHER ON          V1.1
C      THE PROJECTION PLANE ARE RECORDED IN THE IE ARRAY.        V1.1
C      DURING THE PLOTTING PHASE OF THE VIEW PROGRAM, THE        V1.1
C      IE ARRAY IS REORDERED TO DECREASE THE SEARCH TIME        V1.1
C      FOR OBJECTS THAT MAY BE BLOCKING THE CURRENT POINT        V1.1
C      BEING PLOTTED. THE ASSUMPTION USED HERE IS - IF AN        V1.1
C      OBJECT BLOCKED THE PREVIOUS POINT ON A CONTOUR LINE        V1.1
C      THEN THAT OBJECT PROBABLY BLOCKS THE NEXT POINT          V1.1
C      ON THE CONTOUR LINE.                                       V1.1
C                                                                    V1.1
      OPEN (UNIT=4,FILE='VIEWPE.DIR',STATUS='OLD')                80386
      READ (4, '(A12/A12/A12)') FILE1, FILE5, FILE6                80386
      READ (4, '(I4/I4/I4)') IDEF, IOPORT, MODEL                  80386
      OPEN (UNIT=1,FILE=FILE1,STATUS='OLD',FORM='UNFORMATTED')    80386
      OPEN (UNIT=5,FILE=FILE5,STATUS='OLD')                        80386
      OPEN (UNIT=6,FILE=FILE6,STATUS='NEW',CARRIAGE CONTROL='FORTRAN') 80386
C      LUPLOT=8                                                    80386
      ONLINE=1                                                     V1.1
      TERM=2                                                         V1.1
      NF=0                                                            V1.1
      OFFLINE=3                                                      V1.1
      BDRS=4                                                         V1.1
      READ(5,130) DEVFLG                                           V1.1
      DEVFLG=BDRS                                                  80386
      WRITE(6,130) DEVFLG                                          V1.1
130     FORMAT(I1)                                                 V1.1
C                                                                    V1.1
      CALL PLOTS(IDEF, IOPORT, MODEL)                               80386
      CALL FACTOR(.75)                                             80386
C                                                                    V1.1
      IFLAG = 0                                                     V1.1
      CALL TITLE                                                     V1.1
      CALL FACTOR(1.0)                                              80386
      READ(5,200) (ID(I), I=1,10)                                  V1.1
      WRITE(6,200) (ID(I), I=1,10)                                  V1.1
200     FORMAT(10A4)                                               V1.1
      READ(5,150) STIME, DTIME, ETIME                               V1.1
150     FORMAT(3D10.0)                                             V1.1
      CTIME=STIME-DTIME                                           V1.1
      ITIME=CTIME*1000000.DO                                       V1.1
      CTIME=ITIME/1000000.DO                                       V1.1
      READ(5,125) IDEBUG                                           V1.1
      WRITE(6,125) IDEBUG                                          V1.1
125     FORMAT(80I1)                                              V1.1
100     CONTINUE                                                  V1.1
      IFLAG = IFLAG + 1                                           V1.1
      CTIME=CTIME+DTIME                                           V1.1
      ITIME=CTIME*1000000.DO                                       V1.1
      CTIME=ITIME/1000000.DO                                       V1.1
      IF(CTIME.GT.ETIME) CALL PLOT(0.,0.,999)                     V1.1
      IF(CTIME.GT.ETIME) STOP                                       V1.1

```


CALL INPUT(CTIME)	V1.1
IF(IFLAG.EQ.10) GO TO 100	V1.1
IF(DEVFLG.EQ.OFFLINE.OR.DEVFLG.EQ.BDRS)	80386
* CALL COLOR(ICOLOR(91),IERR)	80386
XTIME=ZTIME*1000.0	V1.1
NF=NF+1	V1.1
WRITE(6,1000)NF	V1.1
1000 FORMAT(' MAIN - PROCESSING FRAME #',I4)	V1.1
CALL PLOT(0.,0.,-3)	V1.1
CALL SYMBOL(.5, 8.0,.335,ID,0.,35)	80386
CALL SYMBOL(.5,7.0,.335,'TIME(MSEC)',0.,10)	80386
CALL NUMBER(3.85,7.0,.335,XTIME,0.,-1)	80386
CALL MAT(DVPO,D(1,1,IVP),DVP,3,3,3,3,3,3)	V1.1
CALL DOT(D(1,1,IVP),VPO,VP,3,1,3)	V1.1
DO 10 K=1,3	V1.1
10 VP(K)=VP(K)+SEGLP(K,IVP)	V1.1
CALL MAT(DVP,VP,VP2,3,3,1,3,3,3)	V1.1
CALL CONVREC	V1.1
CALL PRJPLY	V1.1
CALL PRJELR	V1.1
CALL POLYD	V1.1
CALL BUILDIE	V1.1
IF(IDEBUG(1).EQ.1) WRITE(6,350) (NIE(I),I=1,90)	V1.1
IF(IDEBUG(2).EQ.1) WRITE(6,350) ((IE(I,J),I=1,90),J=1,90)	V1.1
350 FORMAT(270(30(1X,I2)/))	V1.1
DO 30 IK=1,NSEG	V1.1
IF(DEVFLG.EQ.OFFLINE.OR.DEVFLG.EQ.BDRS) CALL COLOR(ICOLOR(IK),IERR)	80386
IELP=IK	V1.1
INDEX=NSTEPS(IK)+1	V1.1
INDEX2=4*INDEX-3	V1.1
IX1=1	V1.1
IIN=3*INDEX*INDEX+IX1	V1.1
ISEG=IIN+INDEX	V1.1
CALL ELIPSN(INDEX,WORK(IX1),WORK(IIN))	V1.1
CALL PSE(WORK(IX1),WORK(IIN),WORK(ISEG),INDEX,INDEX2,1)	V1.1
CALL PSE(WORK(IX1),WORK(IIN),WORK(ISEG),INDEX,INDEX2,2)	V1.1
30 CONTINUE	V1.1
CALL PLPLN(WORK,INDEX2)	V1.1
IF(DEVFLG.EQ.BDRS) CALL NFRAME	V1.1
IF(DEVFLG.EQ.OFFLINE.OR.DEVFLG.EQ.ONLINE) CALL PLOT (12.,0.0,-3)	V1.1
IF(DEVFLG.EQ.TERM) CALL PLOT(0.,0.,-3)	V1.1
GO TO 100	V1.1
END	V1.1

	SUBROUTINE BUILDIE	V1.1
C		V1.1
C	ONCE THIS SUBROUTINE IS CALLED ALL OBJECTS ARE REPRESENTED BY	V1.1
C	POLYGONS PROJECTED ON THE VIEWPOINT PROJECTION PLANE.	V1.1
C	THIS SUBROUTINE WILL BUILD THE IE AND NIE ARRAYS.	V1.1
C	NEE(K) REPRESENTS THE NUMBER OF ENTRIES IN THE IE(I,K) ARRAY FOR	V1.1
C	OBJECT K.	V1.1
C	THE IE(I,K) ARRAY CONTAINS OBJECT NUMBERS FOR OBJECTS THAT OVERLAP	V1.1
C	IN THE PROJECTION PLANE.	V1.1
C	FOR EXAMPLE, IE(1,2) MIGHT CONTAIN A 3 WHICH MEANS OBJECT 3	V1.1
C	OVERLAPS WITH OBJECT 2.	V1.1
	COMMON/POLYGON/NPLANE, IFLAG, NPPP(90), PO(3,4,60), P(3,4,60),	V1.1
	*CONVEC(2,4,90), POS(2,90), SIGN(90)	V1.1
	COMMON/ELLIPSE/NSTEPS(90), IELP, A(3,3,30), SEGLP(3,90), VP(3),	V1.1
	*D(3,3,90), DVP(3,3), RA(3), NSEG, SEGLPO(3,90)	V1.2
	COMMON/INTERS/ NIE(90), IE(90,90)	V1.1
	COMMON/REMOVE/NPREM, IREMOV(30)	V1.1
	DO 5 I=1,90	V1.1
	NIE(I) = 0	V1.1
	DO 5 K=1,90	V1.1
5	IE(I,K) = 0	V1.1
	IF(NSEG .EQ. 0) GO TO 60	V1.1
	DO 55 I=1,NSEG	V1.1
	IF(NIE(I) .NE. 0) GO TO 10	V1.1
	NIE(I) = 1	V1.1
	IE(1,I) = I	V1.1
10	IOBJ = I + 1	V1.1
	IF(I.EQ.NSEG) GO TO 31	V1.1
	DO 30 K=IOBJ,NSEG	V1.1
	CALL OVERLAP(I,K,MFLAG)	V1.1
	IF(MFLAG .EQ. 0) GO TO 30	V1.1
C		V1.1
C	YES, THERE IS OVERLAP BETWEEN I AND K	V1.1
C		V1.1
	IF(NIE(K) .NE. 0) GO TO 20	V1.1
	NIE(K) = 1	V1.1
	IE(1,K) = K	V1.1
20	NIE(K) = NIE(K) + 1	V1.1
	NIE(I) = NIE(I) + 1	V1.1
	IE(NIE(I),I) = K	V1.1
	IE(NIE(K),K) = I	V1.1
30	CONTINUE	V1.1
31	CONTINUE	V1.1
	IF(NPLANE.EQ.0) GO TO 55	V1.1
	IOBJ = NPLANE + 30	V1.1
	DO 50 K=31,IOBJ	V1.1
	DO 200 LT=1,NPREM	V1.1
	IF(K-30.EQ.IREMOV(LT)) GO TO 50	V1.1
200	CONTINUE	V1.1
	CALL OVERLAP(I,K,MFLAG)	V1.1
	IF(MFLAG .EQ. 0) GO TO 50	V1.1
C		V1.1

C	YES, THERE IS OVERLAP BETWEEN I AND K	V1.1
C		V1.1
	NIE(K) = NIE(K) + 1	V1.1
	NIE(I) = NIE(I) + 1	V1.1
	IE(NIE(I),I) = K	V1.1
	IE(NIE(K),K) = I	V1.1
50	CONTINUE	V1.1
55	CONTINUE	V1.1
C		V1.1
C	NOW CHECK PLANE AGAINST PLANE	V1.1
C		V1.1
60	IF(NPLANE .LE. 1) RETURN	V1.1
	MPLANE=NPLANE+29	V1.1
	KPLANE=NPLANE+30	V1.1
	DO 100 I=31,MPLANE	V1.1
	DO 300 LT=1,NPREM	V1.1
	IF(I-30.EQ.IREMOV(LT)) GO TO 100	V1.2
300	CONTINUE	V1.1
	IOBJ = I + 1	V1.1
	DO 400 K=IOBJ,KPLANE	V1.1
	CALL OVERLAP(I,K,MFLAG)	V1.1
	IF(MFLAG .EQ. 0) GO TO 400	V1.1
C		V1.1
C	YES, THERE IS OVERLAP BETWEEN I AND K	V1.1
C		V1.1
	DO 500 LT=1,NPREM	V1.2
	IF(K-30.EQ.IREMOV(LT)) GO TO 400	V1.2
500	CONTINUE	V1.2
	NIE(K) = NIE(K) + 1	V1.1
	NIE(I) = NIE(I) + 1	V1.1
	IE(NIE(I),I) = K	V1.1
	IE(NIE(K),K) = I	V1.1
400	CONTINUE	V1.1
100	CONTINUE	V1.1
	RETURN	V1.1
	END	V1.1

```

SUBROUTINE CLIP(X,Y,XSAV,YSAV,XMIN,XMAX,YMIN,YMAX,IPEN,IPLT)      V1.1
C *****                                                    V1.1
C                                                                 V1.1
C THIS SUBROUTINE CLIPS PLOTTING OFF BOTH ENDS OF THE CALCOMP DRUM V1.1
C                                                                 V1.1
C *****                                                    V1.1
C                                                                 V1.1
C LOGICAL XOFF,YOFF                                              V1.1
C DATA LCALL/0/                                               V1.1
C                                                                 V1.1
C DETERMINE IF X AND/OR Y CLIPPING AND IF OFF TOP,BOTTOM,LEFT, V1.1
C OR RIGHT OF PLOTTER                                          V1.1
C                                                                 V1.1
C XOFF=.FALSE.                                                 V1.1
C YOFF=.FALSE.                                                 V1.1
C IF (X.LT.XMIN.OR.X.GT.XMAX) XOFF=.TRUE.                     V1.1
C IF (Y.LT.YMIN.OR.Y.GT.YMAX) YOFF=.TRUE.                     V1.1
C IF (X.LT.XMIN) XLIMIT=XMIN                                   V1.1
C IF (X.GT.XMAX) XLIMIT=XMAX                                   V1.1
C IF (Y.LT.YMIN) YLIMIT=YMIN                                   V1.1
C IF (Y.GT.YMAX) YLIMIT=YMAX                                   V1.1
C                                                                 V1.1
C IF PREVIOUS CALL TO CLIP WAS A PEN UP TO 1ST POINT IN SEGMENT, V1.1
C INTERPOLATE USING NEW AND SAVED COORD.'S, MOVE PEN, RESET SAVE V1.1
C VALUES FOR X+Y POINTS, AND CONTINUE                         V1.1
C                                                                 V1.1
C IF (LCALL.EQ.0) GO TO 10                                     V1.1
C   IF (XOFF) YLTEMP=YINTCP(X,Y,XSAV,YSAV,XLIMIT)             V1.1
C   IF (.NOT.XOFF) YLTEMP=YLIMIT                               V1.1
C   IF (YOFF) XLTEMP=XINTCP(X,Y,XSAV,YSAV,YLIMIT)             V1.1
C   IF (.NOT.YOFF) XLTEMP=XLIMIT                               V1.1
C   IF (XLTEMP.LT.XMIN) XLTEMP=XMIN                             V1.1
C   IF (XLTEMP.GT.XMAX) XLTEMP=XMAX                             V1.1
C   IF (YLTEMP.LT.YMIN) YLTEMP=YMIN                             V1.1
C   IF (YLTEMP.GT.YMAX) YLTEMP=YMAX                             V1.1
C   CALL PLOT(XLTEMP,YLTEMP,3)                                  V1.1
C   XSAV=XLTEMP                                                V1.1
C   YSAV=YLTEMP                                                V1.1
C   LCALL=0                                                    V1.1
C                                                                 V1.1
C IF 1ST POINT OF SEGMENT AND PEN UP, SAVE THESE COORD.'S, SET V1.1
C FLAG, AND EXIT                                              V1.1
C                                                                 V1.1
C 10 CONTINUE                                                  V1.1
C   IF (IPLT.NE.1.OR.IPEN.NE.3) GO TO 20                       V1.1
C   XLSAV=X                                                    V1.1
C   YLSAV=Y                                                    V1.1
C   LCALL=1                                                    V1.1
C   IPLT=0                                                      V1.1
C   RETURN                                                       V1.1
C                                                                 V1.1
C DO WE WANT TO PLOT?                                          V1.1

```

C		V1.1
20	CONTINUE	V1.1
	IF (IPLOT.NE.1) GO TO 30	V1.1
C		V1.1
C	DETERMINE X AND Y COORDINATES TO PLOT TO	V1.1
C		V1.1
	IF (XOFF) YTEMP=YINTCP(X,Y,XSAV,YSAV,XLIMIT)	V1.1
	IF (.NOT.XOFF) YTEMP=YLIMIT	V1.1
	IF (YOFF) XTEMP=XINTCP(X,Y,XSAV,YSAV,YLIMIT)	V1.1
	IF (.NOT.YOFF) XTEMP=XLIMIT	V1.1
C		V1.1
C	PLOT ONLY THE FIRST SEGMENT AFTER CLIPPING DETERMINED, IGNORING	V1.1
C	ALL SEGMENTS AFTER UNLESS PEN TO BE LIFTED.	V1.1
C		V1.1
	CALL PLOT(XTEMP,YTEMP,IPEN)	V1.1
30	CONTINUE	V1.1
	IPLOT=0	V1.1
	RETURN	V1.1
	END	V1.1

	SUBROUTINE CONVREC	V1.1
C		V1.1
C	THIS SUBROUTINE CONVERTS RECTANGLES IN THE ATB	V1.1
C	SIMULATION FORMAT TO POLYGONS IN THE VIEW PLOTTING FORMAT.	V1.1
C		V1.1
	COMMON/ATB/PL(17,30)	V1.1
	DIMENSION R(3)	V1.1
	DIMENSION DX(3)	V1.1
	COMMON/POLYGON/NPLANE, IFLAG, NPPP(90), PO(3,4,60), P(3,4,60),	V1.1
	*CONVEC(2,4,90), POS(2,90), SIGN(90)	V1.1
	COMMON/CONNECT/ NP, MPL(3,5,60)	V1.1
	COMMON/DEBUG/IDEBUG(80), NISG, DEVFLG, ONLINE, TERM, BDRS, OFFLINE	V1.1
	COMMON/ELLIPSE/NSTEPS(90), IELP, A(3,3,30), SEGLP(3,90), VP(3),	V1.1
	*D(3,3,90), DVP(3,3), RA(3), NSEG, SEGLPO(3,90)	V1.2
	INTEGER DEVFLG, ONLINE, TERM, BDRS, OFFLINE	V1.1
	IF(NPLANE.EQ.0) RETURN	V1.1
	IF(IDEBUG(4).NE.0) WRITE(6,50)	V1.1
50	FORMAT(1H1, 'PLANE INFORMATION',/,1H ,17(1H*))	V1.1
	DO 100 J=1,NPLANE	V1.1
	IF(J.GT.NP) GO TO 15	V1.1
	IF(IFLAG.NE.1) GO TO 15	V1.1
	DDD=DET(PL(1,J),PL(2,J),PL(3,J),PL(8,J),PL(9,J),PL(10,J),	V1.1
	- PL(13,J),PL(14,J),PL(15,J))	V1.1
	DX(1) = DET(PL(4,J),PL(2,J),PL(3,J),PL(11,J),PL(9,J),PL(10,J),	V1.1
	- PL(16,J),PL(14,J),PL(15,J))	V1.1
	DX(2) = DET(PL(1,J),PL(4,J),PL(3,J),PL(8,J),PL(11,J),PL(10,J),	V1.1
	- PL(13,J),PL(16,J),PL(15,J))	V1.1
	DX(3) = DET(PL(1,J),PL(2,J),PL(4,J),PL(8,J),PL(9,J),PL(11,J),	V1.1
	- PL(13,J),PL(14,J),PL(16,J))	V1.1
	DO 10 I=1,3	V1.1
	TEMP=DX(I)/DDD	V1.1
	PO(I,1,J)=TEMP	V1.1
	PO(I,2,J)=PL(I+7,J)*PL(12,J)+TEMP	V1.1
	PO(I,3,J)=PL(I+12,J)*PL(17,J)+PO(I,2,J)	V1.1
10	PO(I,4,J)=PL(I+12,J)*PL(17,J)+TEMP	V1.1
15	CONTINUE	V1.1
	ISG=MPL(1,1,J)	V1.1
	IF(ISG.EQ.0) ISG=NISG	V1.1
	IF(IDEBUG(4).NE.0) WRITE(6,200) ISG	V1.1
200	FORMAT(1X, 'ISG=', I2)	V1.1
	DO 20 L=1,4	V1.1
	CALL DOT(D(1,1,ISG),PO(1,L,J),R,3,1,3)	V1.1
	DO 20 I=1,3	V1.1
	P(I,L,J)=R(I)+SEGLP(I,ISG)	V1.1
20	CONTINUE	V1.1
	IF(J.LE.NP) NPPP(J+30)=4	V1.1
	IF(IDEBUG(4).NE.0) WRITE(6,3000)	V1.1
3000	FORMAT(1X,30(1H*))	V1.1
	IF(IDEBUG(4).NE.0) WRITE(6,2000) J	V1.1
2000	FORMAT(3X, 'PLANE NUMBER = ', I3)	V1.1
	JJ=J	V1.1
	NPPPP=NPPP(J+30)	V1.1

IF(IDEBUG(4).NE.0) WRITE(6,1000)((P(I,K,JJ),I=1,3),K=1,NPPPP)	V1.1
1000 FORMAT(3X,F7.2,3X,F7.2,3X,F7.2)	V1.1
100 CONTINUE	V1.1
RETURN	V1.1
END	V1.1

	SUBROUTINE CROSS(A,B,C)	V1.1
C	COMPUTES VECTOR CROSS PRODUCT C=AXB	V1.1
	DIMENSION A(3),B(3),C(3)	V1.1
	C(1)=A(2)*B(3)-A(3)*B(2)	V1.1
	C(2)=A(3)*B(1)-A(1)*B(3)	V1.1
	C(3)=A(1)*B(2)-A(2)*B(1)	V1.1
	RETURN	V1.1
	END	V1.1

FUNCTION DET(A11,A12,A13,A21,A22,A23,A31,A32,A33)	V1.1
DET=A11*(A22*A33-A23*A32)-A12*(A21*A33-A23*A31)	V1.1
1+A13*(A21*A32-A22*A31)	V1.1
RETURN	V1.1
END	V1.1

	SUBROUTINE DOT(A,B,C,N,M,L)		V1.1
C		REV 03 05/31/73	V1.1
C	PERFORMS MATRIX MULTIPLICATION C = A*B.		V1.1
C	IF A AND B ARE VECTORS, C IS THE DOT PRODUCT A.B		V1.1
C			V1.1
C	ARGUMENTS:		V1.1
C	A: MATRIX OF SIZE (L,N).		V1.1
C	B: MATRIX OF SIZE (L,M).		V1.1
C	C: PRODUCT MATRIX OF SIZE (N,M).		V1.1
C	N,M,L: SIZES OF MATRICES A,B,C.		V1.1
C			V1.1
C	(NOTE: SUBROUTINE ASSUMES THAT THE FIRST DIMENSION		V1.1
C	OF A,B AND C IN THE CALLING PROGRAM IS L,L AND N.)		V1.1
C			V1.1
	DIMENSION A(L,1),B(L,1),C(N,1)		V1.1
	DO 10 I=1,N		V1.1
	DO 10 J=1,M		V1.1
	C(I,J) = 0.0		V1.1
	DO 10 K=1,L		V1.1
10	C(I,J) = C(I,J) + A(K,I)*B(K,J)		V1.1
	RETURN		V1.1
	END		V1.1

	SUBROUTINE DOTT(A,B,C,N,M,L)		V1.1
C		REV 01 11/20/72	V1.1
C	PERFORMS MATRIX MULTIPLICATION C = AB'		V1.1
C	WHERE DIMENSIONS ARE A(N,L) , B(M,L) AND C(N,M).		V1.1
C			V1.1
	DIMENSION A(N,1),B(M,1),C(N,1)		V1.1
	DO 10 I=1,N		V1.1
	DO 10 J=1,M		V1.1
	C(I,J)=0.		V1.1
	DO 5 K=1,L		V1.1
	5 C(I,J)= A(I,K)*B(J,K)+C(I,J)		V1.1
10	CONTINUE		V1.1
	RETURN		V1.1
	END		V1.1

	SUBROUTINE DRCYPR (D,A,I1,I2,I3)	V1.1
C		V1.1
	REV 03 07/08/74	V1.1
C	SETS UP 3X3 DIRECTION COSINE MATRIX FOR GIVEN YAW,PITCH AND ROLL.	V1.1
C		V1.1
C	ARGUMENTS:	V1.1
C	D: 3X3 DIRECTION COSINE MATRIX TO BE COMPUTED.	V1.1
C	A: ARRAY OF LENGTH 3 CONTAINING ROTATION ANGLES (DEGREES).	V1.1
C	I1: AXIS OF ROTATION FOR 1ST ANGLE (1,2,3 = X,Y,Z)	V1.1
C	I2: AXIS OF ROTATION FOR 2ND ANGLE (1,2,3 = X,Y,Z)	V1.1
C	I3: AXIS OF ROTATION FOR 3RD ANGLE (1,2,3 = X,Y,Z)	V1.1
C		V1.1
	DIMENSION D(3,3),A(3),T(6,3)	V1.1
	RADIAN=.0174532925199433	V1.1
	Y = A(1)*RADIAN	V1.1
	P = A(2)*RADIAN	V1.1
	R = A(3)*RADIAN	V1.1
	M = 6	V1.1
	N = 3	V1.1
	DO 10 I=1,3	V1.1
	DO 5 J=1,3	V1.1
	D(I,J)=0.	V1.1
	5 T(I,J)=0.	V1.1
	T(I,I)=1.	V1.1
	10 D(I,I)=1.	V1.1
	IF(Y.EQ.0.)GO TO 20	V1.1
	CALL ROT(T,I1,Y,M)	V1.1
	DO 15 I=1,3	V1.1
	DO 15 J=1,3	V1.1
	15 D(I,J)=T(I,J)	V1.1
	20 IF(ABS(P).LT.1.E-11)GO TO 30	V1.2
	CALL ROT(T(4,1),I2,P,M)	V1.1
	CALL MAT(T(4,1),T(1,1),D(1,1),3,3,3,M,M,N)	V1.1
	DO 25 I=1,3	V1.1
	DO 25 J=1,3	V1.1
	25 T(I,J)=D(I,J)	V1.1
	30 IF(ABS(R).LT.1.E-11) GO TO 40	V1.2
	CALL ROT(T(4,1),I3,R,M)	V1.1
	CALL MAT(T(4,1),T(1,1),D(1,1),3,3,3,M,M,N)	V1.1
	40 CONTINUE	V1.1
	RETURN	V1.1
	END	V1.1

	SUBROUTINE ELIPSN(INDEX,X1,IN)	V1.1
C	*****	V1.1
C		V1.1
C	THIS SUBROUTINE GENERATES 1/4 OF THE CONTOUR LINES FOR AN ELLIPSOID	V1.1
C		V1.1
C	*****	V1.1
	DIMENSION X1(3,INDEX,INDEX),IN(INDEX)	V1.1
	COMMON/ELLIPSE/NSTEPS(90),IELP,A(3,3,30),SEGLP(3,90),VP(3),	V1.1
	*D(3,3,90),DVP(3,3),RA(3),NSEG,SEGLPO(3,90)	V1.2
	DELTA X=SQRT(1/A(1,1,IELP))/NSTEPS(IELP)	V1.1
	DELTA Y=SQRT(1/A(2,2,IELP))/NSTEPS(IELP)	V1.1
	SIMP1 = A(1,1,IELP)	V1.1
	SIMP2 = A(2,2,IELP)	V1.1
	SIMP3 = A(3,3,IELP)	V1.1
	DO 101 L=1,INDEX	V1.1
	X=(L-1)*DELTA X	V1.1
	N=0	V1.1
	DO 50 K=1,INDEX	V1.1
	Y=(K-1)*DELTA Y	V1.1
	TEMP = 1-X*X*SIMP1	V1.1
	IF(TEMP.LT.0) TEMP=0	80386
	TEST = TEMP - Y*Y*SIMP2	V1.1
	N=N+1	V1.1
	IF(TEST.LT.0.0) GO TO 100	V1.1
	Z = SQRT(TEST/SIMP3)	V1.1
	X1(1,N,L)=X	V1.1
	X1(2,N,L)=Y	V1.1
	X1(3,N,L)=Z	V1.1
50	CONTINUE	V1.1
	GO TO 101	V1.1
100	Y = SQRT(TEMP/SIMP2)	V1.1
	X1(1,N,L)=X	V1.1
	X1(2,N,L)=Y	V1.1
	X1(3,N,L)=0.0	V1.1
101	IN(L)=N	V1.1
	RETURN	V1.1
	END	V1.1

	SUBROUTINE EXTEND(P,I,J)	V1.1
	COMMON/INTERS/ NIE(90),IE(90,90)	V1.1
	COMMON/ELLIPSE/NSTEPS(90),IELP,A(3,3,30),SEGLP(3,90),VP(3),	V1.1
	*D(3,3,90),DVP(3,3),RA(3),NSEG,SEGLPO(3,90)	V1.2
	DIMENSION P(3,2),P3(3)	V1.1
	COMMON/PLTT/SFACTR,INT,TIME,ICOLOR(91),OFSETX,OFSETY,ZTIME	V1.1
	DOUBLE PRECISION ZTIME	80386
C	*****	V1.1
C		V1.1
C	THIS SUBROUTINE FINDS A MIDPOINT FOR A LINE THAT	V1.1
C	BEGINS ON P(1) AND ENDS ON P(2).	V1.1
C	THIS NEW POINT IS CHECKED BY SUBROUTINE HYDE.	V1.1
C	IF IT IS HIDDEN THEN P(I)=P3	V1.1
C	IF IT IS NOT HIDDEN THEN P(J)=P3	V1.1
C	THIS ALGORITHM IS ITERATED INT TIMES.	V1.1
C	UPON LEAVING EXTEND- P(I) WILL CONTAIN THE RESULT.	V1.1
C		V1.1
C	NOTE: P ARRAY IS CHANGED BY THIS SUBROUTINE.	V1.1
C		V1.1
C	*****	V1.1
	DO 3 IN=1,INT	V1.1
	DO 1 L=1,3	V1.1
1	P3(L)=(P(L,2)+P(L,1))/2.0	V1.1
	NUM=NIE(IELP)	V1.1
	DO 2 IM=1,NUM	V1.1
	KK=IE(IM,IELP)	V1.1
	IF(KK.LE.30) CALL HYDE(KK,P3,IFLAG)	V1.1
	IF(KK.GT.30) CALL HIDE(KK,P3,IFLAG)	V1.1
	IF(IFLAG.EQ.1) GO TO 10	V1.1
2	CONTINUE	V1.1
10	N=I	V1.1
	IF(IFLAG.EQ.1) N=J	V1.1
	DO 3 L=1,3	V1.1
3	P(L,N)=P3(L)	V1.1
	RETURN	V1.1
	END	V1.1

	SUBROUTINE GENDCM(CAMERA, FOCUS, D, ICODE)	V1.2
C	*****	V1.1
C	THIS SUBROUTINE GENERATES A DIRECTION COSINE MATRIX.	V1.1
C		V1.1
	DIMENSION CAMERA(3), FOCUS(3), Z(3), D(3,3)	V1.1
	SUM = 0.0	V1.1
	SI=1.0	V1.2
	IF (FOCUS(1).NE.CAMERA(1).OR.FOCUS(2).NE.CAMERA(2)) GOTO 50	V1.2
	SI=SIGN(1., CAMERA(3) - FOCUS(3))	V1.2
	DO 40 I=1,3	V1.1
	DO 40 J=1,3	V1.1
	D(I,J)=0.0	V1.1
40	CONTINUE	V1.1
	D(1,1)=SI	V1.2
	D(2,2)=1.	V1.2
	D(3,3)=SI	V1.2
	GO TO 999	V1.1
50	IF (ICODE.LT.0) SI=-1.0	V1.2
C		V1.1
	DO 100 I=1,3	V1.1
	Z(I) = FOCUS(I) - CAMERA(I)	V1.1
100	SUM = SUM + Z(I)*Z(I)	V1.1
	SUM = SQRT(SUM)	V1.1
	DO 200 I=1,3	V1.1
200	Z(I) = Z(I)/SUM	V1.1
C		V1.1
	XNORM = SQRT(Z(1)*Z(1) + Z(2)*Z(2))	V1.1
C		V1.1
C	FILL IN FIRST ROW OF D	V1.1
C		V1.1
	D(1,1) = Z(2)/XNORM*SI	V1.2
	D(1,2) = -Z(1)/XNORM*SI	V1.2
	D(1,3) = 0.0	V1.1
C		V1.1
C	FILL IN SECOND ROW OF D	V1.1
C		V1.1
	D(2,1) = Z(1)*Z(3)/XNORM*SI	V1.2
	D(2,2) = Z(2)*Z(3)/XNORM*SI	V1.2
	D(2,3) = -XNORM*SI	V1.2
C		V1.1
C	FILL IN THIRD ROW OF D	V1.1
C		V1.1
	DO 300 I=1,3	V1.1
300	D(3,I) = Z(I)	V1.1
999	CONTINUE	V1.1
	RETURN	V1.1
	END	V1.1

	SUBROUTINE HIDE(KK,P3,IFLAG)	V1.1
	COMMON/POLYGON/NPLANE,KFLAG,NPPP(90),PO(3,4,60),P(3,4,60),	V1.1
	*CONVEC(2,4,90),POS(2,90),SIGN(90)	V1.1
	COMMON/ELLIPSE/NSTEPS(90),IELP,A(3,3,30),SEGLP(3,90),VP(3),	V1.1
	*D(3,3,90),DVP(3,3),RA(3),NSEG,SEGLPO(3,90)	V1.2
	DIMENSION P7(2),PP(3)	V1.1
	DIMENSION P3(3),P4(3)	V1.1
	REAL PPRIME(3),NPRIME(3)	V1.1
	CALL TRANS1(P3,P4)	V1.1
	P7(1)=P4(1)/P4(3)	V1.1
	P7(2)=P4(2)/P4(3)	V1.1
	CALL TPOINT(P7,KK,IFLAG)	V1.1
	IF(IFLAG.EQ.2) RETURN	V1.1
C		V1.1
C	POINT IS INSIDE POLYGON CHECK TO SEE	V1.1
C	IF POLYGON OR POINT IS CLOSER TO VIEWPOINT.	V1.1
C	CALCULATE TAU.	V1.1
C		V1.1
	DO 5 I=1,3	V1.1
	PPRIME(I)=D(1,I,KK)	V1.1
	IF (ABS(PPRIME(I)).LT.1.E-11) PPRIME(I)=.000001	V1.2
5	CONTINUE	V1.1
	CALL MAT(DVP,PPRIME,NPRIME,3,3,1,3,3,3)	V1.1
	DO 10 J=1,3	V1.1
10	PP(J)=P(J,1,KK-30)-VP(J)	V1.1
	CALL MAT(DVP,PP,PPRIME,3,3,1,3,3,3)	V1.1
	CALL DOT(NPRIME,PPRIME,P5,1,1,3)	V1.1
	CALL DOT(NPRIME,P4,P6,1,1,3)	V1.1
	IFLAG=2	V1.1
	IF(P5/P6.GE..99999999) RETURN	V1.1
	IFLAG=1	V1.1
	RETURN	V1.1
	END	V1.1


```

SUBROUTINE HYDE(N,R,IFLAG)
C
C *****
C
C SUBROUTINE HYDE DETERMINES IF A POINT IS HIDDEN BY
C ANOTHER ELLIPSOID.
C
C *****
C *****
C N= POSSIBLE HIDING ELLIPSOID NUMBER.
C R= VECTOR TO PLOTTING POINT.
C IFLAG= FLAG THAT INDICATES HIDDEN LINE OR NOT.
C IFLAG = 2 = NOT HIDDEN
C IFLAG = 1 = HIDDEN
C *****
COMMON/ELLIPSE/NSTEPS(90),IELP,AA(3,3,30),SEGLP(3,90),VP(3),
*D(3,3,90),DVP(3,3),RA(3),NSEG,SEGLPO(3,90)
DIMENSION P1(3),P2(3),R2(3),S(3),V(3)
DIMENSION MU(3),M(3,2),P(3),SON(3),SOI(3)
DIMENSION DD(3,3),VP1(3)
DIMENSION R(3)
REAL M , MU , MAG
C ASSUME NOT HIDDEN.
IFLAG=2
C
C *****
C
C PUT SEGLP(N) IN N'S FRAME.
C PUT SEGLP(M) IN N'S FRAME.
C PUT R2 IN N'S FRAME.
C PUT VIEW POINT IN N'S FRAME.
C
C *****
DO 4 I=1,3
SON(I) = SEGLP(I,N) + SEGLPO(I,N)
4 SOI(I) = SEGLP(I,IELP) + SEGLPO(I,IELP)
CALL MAT(D(1,1,N),SON,P1,3,3,1,3,3,3)
CALL MAT(D(1,1,N),SOI,P2,3,3,1,3,3,3)
IF(IELP .LE. 30) GO TO 55
DO 56 I=1,3
DO 56 J=1,3
56 DD(I,J) = D(I,J,N)
GO TO 57
55 CONTINUE
CALL DOT(D(1,1,N),D(1,1,IELP),DD,3,3,3)
57 CONTINUE
CALL MAT(DD,R,R2,3,3,1,3,3,3)
CALL MAT(D(1,1,N),VP,VP1,3,3,1,3,3,3)
MAG = 0.0
C
C ** *****
C

```

```

C          FIND VECTORS S,V,AND MU.                                V1.1
C          MU WILL BECOME A UNIT VECTOR IN VECTOR M'S DIRECTION   V1.1
C          OR IN M'S OPPOSITE DIRECTION.                          V1.1
C                                                                 V1.1
C          *****                                               V1.1
C                                                                 V1.1
C          DO 1 I=1,3                                             V1.1
C          S(I) = P2(I) + R2(I) - P1(I)                          V1.1
C          V(I) = VP1(I) - P1(I)                                 V1.1
C          MU(I) = S(I) - V(I)                                   V1.1
C          1 MAG = MAG + MU(I)**2                                V1.1
C          MAG = SQRT(MAG)                                       V1.1
C                                                                 V1.1
C          *****                                               V1.1
C                                                                 V1.1
C          MAKE MU A UNIT VECTOR.                                 V1.1
C                                                                 V1.1
C          *****                                               V1.1
C                                                                 V1.1
C          DO 2 I=1,3                                             V1.1
C          2 MU(I) = MU(I) / MAG                                  V1.1
C          A = AA(1,1,N)                                         V1.1
C          B = AA(2,2,N)                                         V1.1
C          C = AA(3,3,N)                                         V1.1
C          IF(ABS(MU(1)).GT..000000001) GO TO 10                 V1.1
C          IF(ABS(MU(2)).GT..000000001) GO TO 20                 V1.1
C          CALL Z(MU,A,B,C,S,M,JFLAG)                            V1.1
C          30 IF(JFLAG.EQ.1) RETURN                               V1.1
C                                                                 V1.1
C          *****                                               V1.1
C                                                                 V1.1
C          FIND P AND COMPARE M TO P TO DETERMINE WHAT POINT     V1.1
C          IS CLOSER TO THE VIEW POINT.                          V1.1
C                                                                 V1.1
C          *****                                               V1.1
C                                                                 V1.1
C          DO 3 I=1,3                                             V1.1
C          3 P(I)= S(I) - V(I) - M(I,1)                          V1.1
C          CALL DOT(P,M(1,1),RESULT1,1,1,3)                     V1.1
C          CALL DOT(P,M(1,2),RESULT2,1,1,3)                     V1.1
C          IF(N.EQ.IELP) GO TO 400                               V1.1
C          IF(RESULT1.GT.0.000000001) IFLAG=1                    V1.1
C          41 IF(RESULT2.GT.0.000000001) IFLAG=1                 V1.1
C          RETURN                                                V1.1
C          10 CALL XYZ(MU,A,B,C,S,M,JFLAG)                       V1.1
C          GO TO 30                                              V1.1
C          20 CALL YZ(MU,A,B,C,S,M,JFLAG)                       V1.1
C          GO TO 30                                              V1.1
C          400 IF(ABS(RESULT2).GT.ABS(RESULT1)) GO TO 41         V1.1
C          RESULT2=RESULT1                                       V1.1
C          GO TO 41                                              V1.1
C          RETURN                                                V1.1

```

END

V1.1

	SUBROUTINE INPUT(CTIME)	V1.1
	COMMON/PLTT/SFACTR, INT, TIME, ICOLOR(91), OFSETX, OFSETY, ZTIME	V1.1
	COMMON/INTERS/ NIE(90), IE(90, 90)	V1.1
	COMMON/ELLIPSE/NSTEPS(90), IELP, A(3, 3, 30), SEGLP(3, 90), VP(3),	V1.1
	*D(3, 3, 90), DVP(3, 3), RA(3), NSEG, SEGLPO(3, 90)	V1.2
	COMMON/POLYGON/NPLANE, IFLAG, NPPP(90), PO(3, 4, 60), P(3, 4, 60),	V1.1
	*CONVEC(2, 4, 90), POS(2, 90), SIGN(90)	V1.1
	COMMON/ATB/PL(17, 30)	V1.1
	DIMENSION BD(24, 40)	V1.1
	DIMENSION DD(3)	V1.1
	COMMON/DEBUG/IDEBUG(80), NISG, DEVFLG, ONLINE, TERM, BDRS, OFFLINE	V1.1
	COMMON/VIEWP/VPO(3), DVPO(3, 3), IVP, VP2(3)	V1.1
	COMMON/CONECT/ NP, MPL(3, 5, 60)	V1.1
	COMMON /REMOVE/NPREM, IREMOV(30)	V1.1
	DOUBLE PRECISION CTIME, ZTIME	V1.1
	INTEGER DEVFLG, ONLINE, TERM, BDRS, OFFLINE	80386
	IF(IFLAG.NE.1) GO TO 600	V1.1
	READ(5, 70) NFAST, NPREM, NISG	V1.1
	WRITE(6, 70) NFAST, NPREM, NISG	V1.1
	READ(5, 72) (IREMOV(I), I=1, NPREM)	V1.1
	WRITE(6, 72) (IREMOV(I), I=1, NPREM)	V1.1
72	FORMAT(3(40I2/))	V1.1
	READ(1, END=800) NSEG, NP, PL, BD, (((MPL(I, J, K), I=1, 3), J=1, 5), K	V1
	*=1, 30)	V1.1
C		V1.1
	39 READ(1, END=700) TIME, ((SEGLP(I, J), I=1, 3), J=1, 30),	V1.1
	* ((D(I, J, K), I=1, 3), J=1, 3), K=1, 30)	V1.1
	ITIME=TIME*1000000.+5	V1.1
	ZTIME=ITIME/1000000.DO	V1.1
C		V1.1
70	FORMAT(3I2)	V1.1
	IF(ZTIME.LT.CTIME) GO TO 39	V1.1
	READ(5, 40) NSP	V1.1
40	FORMAT(I2)	V1.1
	IF(NSP.EQ.0) GO TO 46	V1.1
	DO 45 L=1, NSP	V1.1
	K=NP+L	V1.1
	II=30+NP+L	V1.1
	READ(5, 41) NPPP(II), MPL(1, 1, K)	V1.1
41	FORMAT(I1, I2)	V1.1
	NSIDES=NPPP(II)	V1.1
	DO 45 J=1, NSIDES	V1.1
	READ(5, 42) (PO(I, J, K), I=1, 3)	V1.1
42	FORMAT(3F10.0)	V1.1
45	CONTINUE	V1.1
46	NPLANE=NP+NSP	V1.1
	DO 100 J=1, NSEG	V1.1
	DO 100 I=1, 3	V1.1
100	A(I, I, J)=1.0/BD(I, J)**2	V1.1
	DO 200 J=1, NSEG	V1.1
	CALL DOT(D(1, 1, J), BD(4, J), DD, 3, 1, 3)	V1.1
	DO 200 I=1, 3	V1.1

```

200 SEGLPO(I,J)= DD(I)                                V1.2
    IF(IDEBUG(3).EQ.1) WRITE(6,6) NSEG,NPLANE         V1.1
    6 FORMAT(1H1,'NUMBER OF SEGMENTS = ',I2,', NUMBER OF PLANES = ',I2) V1.1
    NSEG = NSEG-NFAST                                  V1.1
    READ(5,301) (ICOLOR(I),I=1,30)                   V1.1
301 FORMAT(8(5X,I5))                                   V1.1
    II=NPLANE+30                                       V1.1
    READ(5,301) (ICOLOR(I),I=31,90)                  V1.1
    READ(5,301) ICOLOR(91)                             V1.1
    IF(IDEBUG(3).EQ.1) WRITE(6,71) NSEG              V1.1
    71 FORMAT(1X,'THE NUMBER OF SEGMENTS TO BE PLOTTED = ',I2) V1.1
    READ(5,1) (NSTEPS(IPP),IPP=1,NSEG)                V1.1
    READ(5,1) (NSTEPS(IPP+30),IPP=1,NPLANE)           V1.1
    1 FORMAT(30I2)                                       V1.1
    IF(IDEBUG(3).EQ.1) WRITE(6,2) (NSTEPS(IPP),IPP=1,NSEG) V1.1
    2 FORMAT(10X,'NUMBER OF DIVISIONS ALONG A RADIUS',/2X,30I3) V1.1
    IF(IDEBUG(3).EQ.1) WRITE(6,55) (NSTEPS(IPP+30),IPP=1,NPLANE) V1.1
    55 FORMAT(10X,'NUMBER OF DIVISIONS ALONG A SIDE ',/2X,30I3) V1.1
    READ(5,11) INT,SFACTR                               V1.1
    11 FORMAT(I3,7X,F10.2)                              V1.1
    WRITE(6,11) INT,SFACTR                              V1.1
    READ(5,901) OFSETX,OFSETY                           V1.1
    WRITE(6,901) OFSETX,OFSETY                          V1.1
901  FORMAT(2F10.0)                                     V1.1
    IF(IDEBUG(3).EQ.1) WRITE(6,902) OFSETX,OFSETY     V1.1
902  FORMAT(1X,'OFSETX= ',F10.3,4X,'OFSETY= ',F10.3)  V1.1
    IF(IDEBUG(3).EQ.1) WRITE(6,12) SFACTR,INT         V1.1
    12 FORMAT(1X,'SCALE FACTOR = ',F10.2,2X,' ITERATION NUMBER = ',I3) V1.1
    READ(5,13) VP,RA,IVP,ICODE                         V1.1
    13 FORMAT(6F10.0,2I10)                             V1.1
C                                                     V1.1
C ICODE = 0 : ROLL, PITCH, AND YAW ANGLES ARE SUPPLIED IN RA ARRAY. V1.1
C                                                     V1.1
C ICODE = 1 : DIRECTION COSINE MATRIX SUPPLIED AS INPUT.RA ARRAY IS 1ST V1.1
C              ROW OF MATRIX.THE NEXT CARD CONTAINS THE 2ND AND 3RD ROWS V1.1
C                                                     V1.1
C ICODE = 2 : POINT AT WHICH VIEWPOINT Z-AXIS IS TO AIM IS SUPPLIED V1.1
C              IN RA ARRAY.                             V1.1
C                                                     V1.1
C ICODE =-2 : SAME AS ICODE=2, EXCEPT THE POSITIVE Z-AXIS IN THE ATB V1.2
C              SIMULATION IS UP.                       V1.2
C                                                     V1.1
C                                                     V1.1
    IF(ICODE .NE. 0) GO TO 500                          V1.1
    IF(IDEBUG(3).EQ.1) WRITE(6,4) VP,RA                V1.1
    4  FORMAT(1X,'VIEWPOINT VECTOR (' ,F10.1,',',F10.1,',',F10.1,')'/1X V1.1
    1  'ROTATION OF VIEWPOINT RELATIVE TO INERTIAL COORDINATE SYSTEM'/1X V1.1
    2  'THESE ROTATIONS MUST BE DETERMINED BY PERFORMING ROLL MOTION FIRS V1.1
    3  'THEN A PITCH MOTION AND THEN THE YAW MOTION'/10X V1.1
    4  'ROLL = ',F10.1,1X,'DEG.',5X,'PITCH = ',F10.1,1X,'DEG.', V1.1
    55X,'YAW = ',F10.1,1X,'DEG. ') V1.1
    CALL DRCYPR(DVP,RA,1,2,3)                          V1.1
    GO TO 550                                           V1.1

```

```

500 IF (IABS(ICODE).EQ.2) CALL GENDCM(VP,RA,DVP,ICODE)          V1.2
    IF (IABS(ICODE).EQ.2) GO TO 550                            V1.2
    DO 501 JJJ=1,3                                             V1.1
501 DVP(1,JJJ) = RA(JJJ)                                       V1.1
    READ(5,13) ((DVP(I,J),J=1,3),I=2,3)                       V1.1
    WRITE(6,13)((DVP(I,J),J=1,3),I=2,3)                       V1.1
    IF(IDEBUG(3).EQ.1) WRITE(6,15) (VP(I),I=1,3)             V1.1
15  FORMAT(' VIEW POINT VECTOR  (' ,F10.1,',',F10.1,',',F10.1,')',/,
- ' VIEW POINT ORIENTATION DEFINED IN DIRECTION COSINE MATRIX FORM
- ')                                                           V1.1
550 CONTINUE                                                  V1.1
    DO 80 J=1,3                                               V1.1
    VPO(J)=VP(J)                                             V1.1
    DO 80 I=1,3                                               V1.1
80  DVPO(J,I)=DVP(J,I)                                       V1.1
    IF(IFLAG.NE.1) RETURN                                     V1.1
    IF(IDEBUG(3).EQ.1) WRITE(6,60)                            V1.1
60  FORMAT(1X,'*****'/,1X,
1  'VIEWPOINT DIRECTION COSINE MATRIX')                       V1.1
    IF(IDEBUG(3).EQ.1) WRITE(6,53)((DVP(I,J),J=1,3),I=1,3)  V1.1
    IF(IDEBUG(3).EQ.1) WRITE(6,54)                            V1.1
    DO 20 IKE=1,NSEG                                          V1.1
    IF(IDEBUG(3).EQ.1) WRITE(6,50) IKE,(SEGLP(I,IKE),I=1,3)  V1.1
50  FORMAT(1X,'*****'/1X'SEGMENT # ',I3,5X,'SEGLP = ('
- F10.3,2(' ,F10.3),',',//,1X,'A MATRIX',T50,'DIRECTION COSINE MATRIX
- '/')                                                       V1.1
    DO 400 III=1,3                                           V1.1
    IF(IDEBUG(3).EQ.1) WRITE(6,51) (A(III,J,IKE),J=1,3),(D(III,J,IKE),
+J=1,3)                                                       V1.1
400 CONTINUE                                                  V1.1
'1  FORMAT(3(2X,F8.5),T50,3(2X,F9.6))                       V1.1
53  FORMAT(3(2X,F9.6))                                       V1.1
    IF(IDEBUG(3).EQ.1) WRITE(6,54)                            V1.1
54  FORMAT(1X,/,1X'*****')                                  V1.1
20  CONTINUE                                                  V1.1
    RETURN                                                    V1.1
600 CONTINUE                                                  V1.1
    IF(ZTIME.LT.CTIME) GO TO 675                              V1.1
    IF(ISW1.EQ.0) GO TO 680                                   V1.1
    DO 650 J=1,NSEG                                          V1.1
    CALL DOT(D(1,1,J),BD(4,J),DD,3,1,3)                       V1.1
    DO 650 I=1,3                                             V1.1
650 SEGLPO(I,J)= DD(I)                                       V1.2
    ISW1=0                                                    V1.1
    IFLAG=5                                                  V1.1
    RETURN                                                    V1.1
675 READ(1,END=700) TIME,((SEGLP(I,J),I=1,3),J=1,30),
* ((D(I,J,K),I=1,3),J=1,3),K=1,30)                          V1.1
    ITIME=TIME*1000000.+5                                     V1.1
    ZTIME=ITIME/1000000.DO                                   V1.1
C                                                                 V1.1
    ISW1=1                                                    V1.1

```

GO TO 600	V1.1
700 WRITE(6,720)	V1.1
720 FORMAT(1X,'END OF DATA REACHED.')	V1.1
ISW1=1	V1.1
STOP	V1.1
800 WRITE(6,820)	V1.1
820 FORMAT(1X,'NO DATA ON TAPE.')	V1.1
STOP	V1.1
680 IFLAG=10	V1.1
RETURN	V1.1
END	V1.1

```

SUBROUTINE LSEGINT(P1,P2,R1,R2,IFLAG)
C
C THIS SUBROUTINE DETERMINES IF TWO LINE SEGMENTS, P1P2 AND R1R2,
C INTERSECT.
C ALL PARALLEL LINE SEGMENTS,WHETHER COINCIDENT OR NOT, ARE
C CONSIDERED TO BE NON-INTERSECTING.
C CASE 1 IS CONSIDERED TO BE THE REGULAR CONFIGURATION.
C THE SPECIAL CASES ARE AS FOLLOWS:
C 2) ONE LINE IS VERTICAL
C 3) BOTH LINES ARE VERTICAL
C 4) BOTH LINES ARE HORIZONTAL
C 5) BOTH LINES HAVE THE SAME NON-ZERO SLOPE
C 6) ONE LINE IS VERTICAL, THE OTHER IS HORIZONTAL
C
C IFLAG = 1 INDICATES INTERSECTION; IFLAG = 0 INDICATES NO INTERSECTION.
C
C DIMENSION P1(2),P2(2),R1(2),R2(2),P(4,2),T(2)
C REAL M(2)
C IFLAG=0
C
C SET UP ARRAYS
C DO 1 I=1,2
C P(1,I) = P1(I)
C P(2,I) = P2(I)
C P(3,I) = R1(I)
C 1 P(4,I) = R2(I)
C
C DETERMINE IF CASE 3
C IF(ABS(P(1,1)-P(2,1)).LT.1.E-11.AND.ABS(P(3,1)-P(4,1)).LT.
C +1.E-11) RETURN
C
C DETERMINE IF CASE 4
C IF(ABS(P(1,2)-P(2,2)).LT.1.E-11.AND.ABS(P(3,2)-P(4,2)).LT.
C +1.E-11) RETURN
C
C DETERMINE IF CASE 6
C DO 2 I=1,2
C J = 3-I
C IF(ABS(P(1,I)-P(2,I)).LT.1.E-11.AND.ABS(P(3,J)-P(4,J)).LT.
C +1.E-11) GO TO 10
C 2 CONTINUE
C
C DETERMINE IF CASE 2
C IF(ABS(P(1,1)-P(2,1)).LT.1.E-11.OR.ABS(P(3,1)-P(4,1)).LT.
C +1.E-11) GO TO 6
C GO TO 5
C 6 DO 3 I=1,4
C TEMP = P(I,1)
C P(I,1) = P(I,2)
C 3 P(I,2) = TEMP
C

```


C	REGULAR PROCEDURE	V1.1
	5 DO 4 I=1,2	V1.1
	I1 = 2*I	V1.1
	I2 = 2*I - 1	V1.1
	4 M(I) = (P(I1,2) - P(I2,2))/(P(I1,1) - P(I2,1))	V1.1
C		V1.1
C	CHECK FOR CASE 5	V1.1
	IF(ABS(M(1)-M(2)).LT.1.E-11) RETURN	V1.1
	X = (P(3,2) - P(1,2) + M(1)*P(1,1) - M(2)*P(3,1))/(M(1) - M(2))	V1.1
	DO 7 I=1,2	V1.1
	7 T(I) = (X - P(2*I-1,1))/(P(2*I,1) - P(2*I-1,1))	V1.1
	20 IF(T(1).GT.0 .AND. T(2).GT.0 .AND. T(1).LT.1 .AND. T(2).LT.1)	V1.1
	- IFLAG=1	V1.1
	RETURN	V1.1
C		V1.1
C	CASE 6 PROCEDURE	V1.1
	10 IF(ABS(P(1,1)-P(2,1)).LT.1.E-11) GO TO 11	V1.1
	J=1	V1.1
	I=2	V1.1
	GO TO 12	V1.1
	11 I=1	V1.1
	J=2	V1.1
	12 T(1) = (P(3,J) - P(1,J))/(P(2,J) - P(1,J))	V1.1
	T(2) = (P(1,I) - P(3,I))/(P(4,I) - P(3,I))	V1.1
	GO TO 20	V1.1
	END	V1.1

	SUBROUTINE MAT(A,B,C,LL,MM,NN,JA,JB,JC)		V1.1
C		REV 03 05/31/73	V1.1
C	PERFORMS MATRIX MULTIPLICATION C = AB.		V1.1
C			V1.1
C	ARGUMENTS:		V1.1
C	A: MATRIX OF SIZE (L,M).		V1.1
C	B: MATRIX OF SIZE (M,N).		V1.1
C	C: PRODUCT MATRIX OF SIZE (L,N).		V1.1
C	L,M,N: SIZES OF MATRICES A,B,C.		V1.1
C	LA, LB, LC: 1ST DIMENSION OF A,B,C IN CALLING PROGRAM.		V1.1
C			V1.1
	DIMENSION A(JA,1),B(JB,1),C(JC,1)		V1.1
	DO 20 L=1,LL		V1.1
	DO 10 N=1,NN		V1.1
	S = 0.0		V1.1
	DO 5 M=1,MM		V1.1
	5 S=S+A(L,M)*B(M,N)		V1.1
	C(L,N)=S		V1.1
10	CONTINUE		V1.1
20	CONTINUE		V1.1
	RETURN		V1.1
	END		V1.1

	SUBROUTINE NFRAME	V1.1
C		V1.1
C	THIS ROUTINE PERFORMS THE END OF FRAME HANDLING FOR THE BDRS	V1.1
C		V1.1
	COMMON/DEVICE/IDEF, IOPORT, MODEL	80386
	INTEGER*2 ENDFRA, MASK, STATUS	V1.1
	DATA ENDFRA, MASK/Z'FFFF', Z'FFFF' /	80386
C	CALL DOLWH(8, 1, ENDFRA, MASK, STATUS)	80386
C	CALL PLOTS(M, N, LU)	80386
	CALL PLOT(0.0, 0.0, -999)	80386
	RETURN	V1.1
	END	V1.1

```

SUBROUTINE OVERLAP(III, KKK, MFLAG)                                V1.1
DIMENSION P1(2), P2(2), R1(2), R2(2)                            V1.1
DIMENSION PP2(2)                                                V1.1
COMMON/POLYGON/NPLANE, IFLAG, NPPP(90), PO(3,4,60), P(3,4,60),  V1.1
*CONVEC(2,4,90), POS(2,90), SIGN(90)                            V1.1
C                                                                V1.1
C OVERLAP TAKES OBJECTS I AND K AND TESTS FOR ANY OVERLAP ON THE V1.1
C PROJECTION PLANE.                                             V1.1
C MFLAG WILL BE RETURNED TO INDICATE IF OVERLAP OR NOT.        V1.1
C     MFLAG=0 MEANS NO OVERLAP                                   V1.1
C     MFLAG=1 MEANS OVERLAP                                     V1.1
C                                                                V1.1
C     I = III                                                  V1.1
C     K = KKK                                                  V1.1
C     5 CONTINUE                                               V1.1
C     DO 10 J=1,2                                              V1.1
10 PP2(J) = POS(J,K)                                           V1.1
C                                                                V1.1
C GO AROUND THE RINGS                                           V1.1
C                                                                V1.1
C     NPTS1 = NPPP(I)                                          V1.1
C     NPTS2 = NPPP(K)                                          V1.1
C     DO 200 J=1, NPTS2                                        V1.1
C     CALL TPOINT(PP2, I, MFLAG)                               V1.1
C     IF(MFLAG.EQ.1) RETURN                                    V1.1
C     DO 200 N=1,2                                             V1.1
200 PP2(N) = PP2(N) + CONVEC(N, J, K)                           V1.1
C                                                                V1.1
C     CHECKED ALL POINTS AND FOUND THEY WERE ALL OUTSIDE.     V1.1
C                                                                V1.1
C NEXT, CHECK FOR INTERSECTING LINE SEGMENTS.                  V1.1
C                                                                V1.1
C     DO 60 II=1,2                                             V1.1
C     P1(II) = POS(II, I)                                       V1.1
60 R1(II) = POS(II, K)                                          V1.1
C     DO 61 L=1, NPTS1                                         V1.1
C     DO 62 II=1,2                                             V1.1
62 P2(II) = P1(II) + CONVEC(II, L, I)                           V1.1
C     DO 63 J=1, NPTS2                                         V1.1
C     DO 64 II=1,2                                             V1.1
64 R2(II) = R1(II) + CONVEC(II, J, K)                           V1.1
C     CALL LSEGINT(P1, P2, R1, R2, MFLAG)                       V1.1
C     IF(MFLAG.EQ. 1) RETURN                                    V1.1
C     R1(1) = R2(1)                                           V1.1
63 R1(2) = R2(2)                                               V1.1
C     P1(1) = P2(1)                                           V1.1
61 P1(2) = P2(2)                                               V1.1
C     IF(I.NE. III) RETURN                                     V1.1
C     I = KKK                                                  V1.1
C     K = III                                                  V1.1
C     GO TO 5                                                  V1.1
C     END                                                       V1.1

```

```

SUBROUTINE PLPLN(SEG, INDEX2)
C *****
C THIS SUBROUTINE PLOTS THE PLANES.
C *****
COMMON/ELLIPSE/NSTEPS(90), IELP, AA(3,3,30), SEGLP(3,90), VP(3),
*D(3,3,90), DVP(3,3), RA(3), NSEG, SEGLPO(3,90)
COMMON/POLYGON/NPLANE, IFLAG, NPPP(90), PO(3,4,60), P(3,4,60),
*CONVEC(2,4,90), POS(2,90), SIGN(90)
COMMON/DEBUG/IDEBUG(80), NISG, DEVFLG, ONLINE, TERM, BDRS, OFFLINE
DIMENSION SEG(3,3333)
INTEGER DEVFLG, ONLINE, TERM, BDRS, OFFLINE
COMMON/PLTT/SFACTR, INT, TIME, ICOLOR(91), OFSETX, OFSETY, ZTIME
COMMON /REMOVE/NPREM, IREMOV(30)
DOUBLE PRECISION ZTIME
IF(NPLANE .EQ. 0) RETURN
SEG(1,1) = 0.0
SEG(2,1) = 0.0
SEG(3,1) = 0.0
DO 500 LL=1, NPLANE
DO 100 I=1, NPREM
IF (LL.EQ.IREMOV(I)) GO TO 500
CONTINUE
L = LL + 30
IF(DEVFLG.EQ.OFFLINE.OR.DEVFLG.EQ.BDRS) CALL COLOR(ICOLOR(L), IERR)
NUM = NPPP(L)*NSTEPS(L) + 1
A = 1./NSTEPS(L)
NSIDES = NPPP(L)
DO 400 K=1, NSIDES
KK = K + 1
IF(K .EQ. NSIDES) KK=1
I1 = (K-1)*NSTEPS(L) + 2
I2 = I1 + NSTEPS(L) - 1
DO 400 I=I1, I2
DO 400 J=1, 3
400 SEG(J, I) = SEG(J, I-1) + A*(P(J, KK, LL) - P(J, K, LL))
IPEN = 3
IELP = L
CALL PNTPLT(SEG(1,1), IPEN, INDEX2, NUM)
500 CONTINUE
RETURN
END

```

```

SUBROUTINE PNTPLT(SEG, IPEN, INDEX2, NPTS)
C *****
C
C POINT PLOT SUBROUTINE.
C *****
COMMON/PLTT/SFACTR, INT, TIME, ICOLOR(91), OFSETX, OFSETY, ZTIME
COMMON/ELLIPSE/NSTEPS(90), IELP, A(3,3,30), SEGLP(3,90), VP(3),
*D(3,3,90), DVP(3,3), RA(3), NSEG, SEGLPO(3,90)
COMMON/INTERS/ NIE(90), IE(90,90)
DOUBLE PRECISION ZTIME
DIMENSION P(3), PP(3,2), PPP(3)
DIMENSION SEG(3,3333)
DATA YMIN/0.0/, YMAX/11.0/, IPLOT/1/
DATA IFIRST/0/
DATA ITWO/2/, ITHREE/3/
IF (IFIRST.EQ.0) READ(5,1000)XMIN,XMAX
1000 FORMAT(2F10.2)
IF (IFIRST.EQ.0) WRITE(6,1001)XMIN,XMAX
1001 FORMAT(' XMIN,XMAX=',2(1X,F10.3))
IFIRST=1
LFLAG=2
IFLAG=2
NEWPEN=0
DO 100 IPNT=1,NPTS
INUM=NIE(IELP)
IF(INUM.EQ.0) GO TO 61
DO 60 K=1, INUM
KK=IE(K, IELP)
IF(KK.LE.30) CALL HYDE(KK, SEG(1, IPNT), IFLAG)
IF(KK.GT.30) CALL HIDE(KK, SEG(1, IPNT), IFLAG)
IF(IFLAG.EQ.1) GO TO 61
60 CONTINUE
61 IF(K.GT.INUM .OR. K.EQ.1) GO TO 62
ITEMP = IE(1, IELP)
IE(1, IELP) = IE(K, IELP)
IE(K, IELP) = ITEMP
62 IF(IFLAG.NE. LFLAG) THEN
IF(IPNT.EQ.1) GO TO 70
DO 250 IJ=1,3
PP(IJ,1)=SEG(IJ, IPNT-1)
250 PP(IJ,2)=SEG(IJ, IPNT)
CALL EXTEND(PP, IFLAG, LFLAG)
DO 260 IJ=1,3
260 P(IJ)=PP(IJ, IFLAG)
CALL TRANS1(P, PPP)
X=-PPP(1)*SFACTR/PPP(3)+OFSETX
Y=PPP(2)*SFACTR/PPP(3)+OFSETY
IF(LFLAG.EQ.1) GO TO 350
IF (X.GE.XMIN.AND.X.LE.XMAX.AND.
1 Y.GE.YMIN.AND.Y.LE.YMAX) GO TO 261
CALL CLIP(X, Y, XSAV, YSAV, XMIN, XMAX, YMIN, YMAX, ITWO, IPLOT)

```

		NEWPEN--3	80386
		GO TO 265	80386
261		CONTINUE	80386
		IF (IPNT.NE.1)	80386
	1	CALL PREPLT(X,Y,XSAV,YSAV,XMIN,XMAX,YMIN,YMAX,IPEN,NEWPEN)	80386
		CALL PLOT(X,Y,2)	80386
		IPLOT=1	80386
		NEWPEN=3	80386
265		CONTINUE	80386
		IPEN=3	80386
		XSAV=X	80386
		YSAV=Y	80386
		LFLAG=1	80386
		GO TO 100	80386
350		CONTINUE	80386
		IF (X.GE.XMIN.AND.X.LE.XMAX.AND.	80386
	1	Y.GE.YMIN.AND.Y.LE.YMAX) GO TO 351	80386
		CALL CLIP(X,Y,XSAV,YSAV,XMIN,XMAX,YMIN,YMAX,ITHREE,IPLOT)	80386
		NEWPEN--2	80386
		IPEN=3	80386
		GO TO 355	80386
351		CONTINUE	80386
		IF (IPNT.NE.1)	80386
	1	CALL PREPLT(X,Y,XSAV,YSAV,XMIN,XMAX,YMIN,YMAX,IPEN,NEWPEN)	80386
		CALL PLOT(X,Y,3)	80386
		IPLOT=1	80386
		NEWPEN=2	80386
		IPEN=2	80386
355		CONTINUE	80386
		XSAV=X	80386
		YSAV=Y	80386
		ENDIF	80386
		70 IF(IFLAG.EQ.1) THEN	80386
		IPEN=3	80386
		LFLAG=1	80386
		GO TO 100	80386
		ENDIF	80386
		LFLAG=2	V1.1
		CALL TRANS1(SEG(1,IPNT),PPP)	V1.1
		X=-PPP(1)*SFACTR/PPP(3)+OFSETX	V1.1
		Y=PPP(2)*SFACTR/PPP(3)+OFSETY	V1.1
		IF (X.GE.XMIN.AND.X.LE.XMAX.AND.	V1.1
	1	Y.GE.YMIN.AND.Y.LE.YMAX) GO TO 71	V1.1
		CALL CLIP(X,Y,XSAV,YSAV,XMIN,XMAX,YMIN,YMAX,IPEN,IPLOT)	V1.1
		NEWPEN--2	V1.1
		IPEN=3	V1.1
		GO TO 75	V1.1
71		CONTINUE	V1.1
		IF (IPNT.NE.1)	V1.1
	1	CALL PREPLT(X,Y,XSAV,YSAV,XMIN,XMAX,YMIN,YMAX,IPEN,NEWPEN)	V1.1
		CALL PLOT(X,Y,IPEN)	V1.1
		IPLOT=1	V1.1

	NEWPEN=2	V1.1
	IPEN=2	V1.1
75	CONTINUE	V1.1
	XSAV=X	V1.1
	YSAV=Y	V1.1
100	CONTINUE	V1.1
	RETURN	V1.1
	END	V1.1

	SUBROUTINE POLYD	V1.1
C		V1.1
C	POLYD GENERATES DIRECTION COSINE MATRICIES FOR THE POLYGONS.	V1.1
C	THE X AXIS OF THE POLYGON COORDINATE SYSTEM IS THE	V1.1
C	NORMAL VECTOR TO THE POLYGON SURFACE.	V1.1
C	Y VECTOR IS ALIGNED WITH ONE OF THE POLYGON SIDES.	V1.1
C		V1.1
	COMMON/ELLIPSE/NSTEPS(90), IELP, A(3,3,30), SEGLP(3,90), VP(3),	V1.1
	*D(3,3,90), DVP(3,3), RA(3), NSEG, SEGLPO(3,90)	V1.2
	COMMON/POLYGON/NPLANE, IFLAG, NPPP(90), PO(3,4,60), P(3,4,60),	V1.1
	*CONVEC(2,4,90), POS(2,90), SIGN(90)	V1.1
	DIMENSION INDEX(6), D1(810)	V1.1
	EQUIVALENCE (D, D1)	V1.1
	DATA INDEX/3,6,7,1,2,5/	V1.1
	DO 100 L=1, NPLANE	V1.1
	J=9*L+262	V1.1
	DO 20 I=1, 3	V1.1
	D1(J+I+2)=P(I,2,L)-P(I,1,L)	V1.1
20	D1(J+I+5)=P(I,3,L)-P(I,1,L)	V1.1
	CALL CROSS(D1(J+3), D1(J+6), D1(J))	V1.1
	SUMD1=0.0	V1.1
	SUMD2=0.0	V1.1
	DO 30 I=1, 3	V1.1
	SUMD1=SUMD1+D1(J+I-1)**2	V1.1
30	SUMD2=SUMD2+D1(J+I+2)**2	V1.1
	SUMD1=SQRT(SUMD1)	V1.1
	SUMD2=SQRT(SUMD2)	V1.1
	DO 40 I=1, 3	V1.1
	D1(J+I-1)=D1(J+I-1)/SUMD1	V1.1
40	D1(J+I+2)=D1(J+I+2)/SUMD2	V1.1
	CALL CROSS(D1(J), D1(J+3), D1(J+6))	V1.1
	DO 50 I=1, 3	V1.1
	TEMP=D1(INDEX(I)+J)	V1.1
	D1(INDEX(I)+J)=D1(INDEX(I+3)+J)	V1.1
50	D1(INDEX(I+3)+J)=TEMP	V1.1
100	CONTINUE	V1.1
	DO 600 J=1, NPLANE	V1.1
	NUM = J + 30	V1.1
	DO 600 K=1, 3	V1.1
600	SEGLP(K, NUM) = P(K, 1, J)	V1.1
	RETURN	V1.1
	END	V1.1

```

SUBROUTINE PREPLT(X,Y,XSAV,YSAV,XMIN,XMAX,YMIN,YMAX,IPEN,NEWPEN)      V1.1
C*****                                                                    V1.1
C                                                                    V1.1
C THIS SUBROUTINE PERFORMS NECESSARY PEN MOVES FOR 1ST PLOT          V1.1
C AFTER CLIPPING                                                       V1.1
C                                                                    V1.1
C*****                                                                    V1.1
      LOGICAL XOFF,YOFF                                                V1.1
      IF (NEWPEN.NE.-2) RETURN                                          V1.1
C                                                                    V1.1
C OUTSIDE X AND/OR Y BOUNDARIES???                                     V1.1
C                                                                    V1.1
      XOFF=.FALSE.                                                    V1.1
      YOFF=.FALSE.                                                    V1.1
      IF (X.LT.XMIN.OR.XSAV.LT.XMIN.OR.                                V1.1
1      X.GT.XMAX.OR.XSAV.GT.XMAX) XOFF=.TRUE.                          V1.1
      IF (Y.LT.YMIN.OR.YSAV.LT.YMIN.OR.                                V1.1
1      Y.GT.YMAX.OR.YSAV.GT.YMAX) YOFF=.TRUE.                          V1.1
C                                                                    V1.1
C DETERMINE IF OFF TOP,BOTTOM,RIGHT,OR LEFT SIDE OF PLOTTER          V1.1
C                                                                    V1.1
      IF (X.LT.XMIN.OR.XSAV.LT.XMIN) XLIMIT=XMIN                       V1.1
      IF (X.GT.XMAX.OR.XSAV.GT.XMAX) XLIMIT=XMAX                       V1.1
      IF (Y.LT.YMIN.OR.YSAV.LT.YMIN) YLIMIT=YMIN                       V1.1
      IF (Y.GT.YMAX.OR.YSAV.GT.YMAX) YLIMIT=YMAX                       V1.1
C                                                                    V1.1
C GET X AND Y POINTS DEPENDING ON WHAT BOUNDARIES OVER                V1.1
C                                                                    V1.1
      IF (XOFF) YTEMP=YINTCP(X,Y,XSAV,YSAV,XLIMIT)                    V1.1
      IF (.NOT.XOFF) YTEMP=YLIMIT                                       V1.1
      IF (YOFF) XTEMP=XINTCP(X,Y,XSAV,YSAV,YLIMIT)                    V1.1
      IF (.NOT.YOFF) XTEMP=XLIMIT                                         V1.1
C                                                                    V1.1
C MOVE PEN UP TO THAT SPOT                                             V1.1
C                                                                    V1.1
      CALL PLOT(XTEMP,YTEMP,3)                                          V1.1
      IPEN=2                                                            V1.1
      RETURN                                                            V1.1
      END                                                                V1.1

```

```

SUBROUTINE PRJELR
C *****
C THIS SUBROUTINE PROJECTS ELLIPSOIDS ONTO THE PROJECTION PLANE.
C *****
COMMON/ELLIPSE/NSTEPS(90), IELP, A(3,3,30), SEGLP(3,90), VP(3),
*D(3,3,90), DVP(3,3), RA(3), NSEG, SEGLPO(3,90)
COMMON/POLYGON/NPLANE, IFLAG, NPPP(90), PO(3,4,60), P(3,4,60),
*CONVEC(2,4,90), POS(2,90), SIGN(90)
DIMENSION DD(3,3), DDD(3,3), SS(3), S(3)
DIMENSION R(3,3), R2(2,3)
REAL LAMDA1, LAMDA2, M1, M2
IF(NSEG .EQ. 0) RETURN
DO 100 I=1, NSEG
CALL DOT(D(1,1,I), DVP, DD, 3,3,3)
CALL MAT(A(1,1,I), DD, DDD, 3,3,3,3,3,3)
CALL DOT(D(1,1,I), DDD, DD, 3,3,3)
CALL MAT(DVP, DD, DDD, 3,3,3,3,3,3)
DO 10 K=1,3
10 SS(K) = SEGLP(K,I) - VP(K) + SEGLPO(K,I)
CALL MAT(DVP, SS, S, 3,3,1,3,3,3)
DO 30 II=1,3
IF (ABS(S(II)).LT.1.E-11) S(II)=-1.0
30 CONTINUE
C
CALL SOLVR(DDD(1,1), DDD(2,1), DDD(3,1), DDD(1,3), DDD(2,3), DDD(3,3),
*DDD(1,1), DDD(1,3), S, R(1,1), R(3,1))
CALL SOLVR(DDD(1,2), DDD(2,2), DDD(3,2), DDD(1,3), DDD(2,3), DDD(3,3),
*DDD(2,2), DDD(2,3), S, R(2,2), R(3,2))
CALL SOLVR(DDD(1,1)+DDD(1,2), DDD(2,1)+DDD(2,2), DDD(3,1)+DDD(3,2),
*DDD(1,3), DDD(2,3), DDD(3,3), DDD(1,1)+2.0*DDD(1,2)+DDD(2,2),
*DDD(1,3)+DDD(2,3), S, R(1,3), R(3,3))
R(2,1)=-0.0
R(1,2)=-0.0
R(2,3)=R(1,3)
DO 15 IK=1,3
DO 15 IJ=1,2
15 R2(IJ, IK)=(S(IJ)+R(IJ, IK))/(S(3)+R(3, IK))-S(IJ)/S(3)
CALL SOLVA(R2, A11, A22, A12)
TEMP=(A11+A22)**2-4.0*(A11*A22-A12**2)
IF(TEMP.LT.0.0) TEMP=0.0
TEMP=SQRT(TEMP)
LAMDA1=(A11+A22+TEMP)/2.0
LAMDA2=(A11+A22-TEMP)/2.0
RX1=A12
RY1=LAMDA1-A11
RX2=LAMDA2-A22
RY2=A12
LAMDA1=ABS(LAMDA1)
LAMDA2=ABS(LAMDA2)
M1=SQRT(1.0/(LAMDA1*(RX1**2+RY1**2)))

```

M2=SQRT(1.0/(LAMDA2*(RX2**2+RY2**2)))	V1.1
RX1=RX1*M1	V1.1
RX2=RX2*M2	V1.1
RY1=RY1*M1	V1.1
RY2=RY2*M2	V1.1
CONVEC(1,1,I)=-2.0*RX1	V1.1
CONVEC(2,1,I)=-2.0*RY1	V1.1
CONVEC(1,2,I)=-2.0*RX2	V1.1
CONVEC(2,2,I)=-2.0*RY2	V1.1
DO 20 IJ=1,2	V1.1
CONVEC(IJ,3,I) = -CONVEC(IJ,1,I)	V1.1
CONVEC(IJ,4,I) = -CONVEC(IJ,2,I)	V1.1
20 POS(IJ,I) = CONVEC(IJ,3,I)/2.0 + CONVEC(IJ,4,I)/2.0+S(IJ)/S(3)	V1.1
NPPP(I)=4	V1.1
SIGN(I)=CONVEC(1,1,I)*CONVEC(2,2,I) -	V1.1
1CONVEC(2,1,I)*CONVEC(1,2,I)	V1.1
100 CONTINUE	V1.1
RETURN	V1.1
END	V1.1

	SUBROUTINE PRJPLY	V1.1
C		V1.1
C	*****	V1.1
C		V1.1
C	THIS SUBROUTINE WILL SETUP THE CONVEC ARRAY.	V1.1
C	IT ALSO SETS UP THE POS AND SIGN ARRAYS.	V1.1
C		V1.1
C	ARRAY P IS THE ORIGINAL POSITION VECTORS FOR THE POLYGONS	V1.1
C	IN THE INERTIAL REFERENCE SYSTEM.	V1.1
C	ARRAY CONVEC WILL CONTAIN THE CONTOUR VECTORS	V1.1
C	FOR PROJECTED POLYGONS.	V1.1
C	ARRAY POS WILL CONTAIN POSITION VECTORS FROM THE	V1.1
C	PROJECTION PLANE ORIGIN TO POLYGON POINT # 1.	V1.1
C	ARRAY SIGN WILL CONTAIN THE SIGN THAT RESULTS FROM	V1.1
C	THE CROSS PRODUCT (P2-P1)X(P3-P2).	V1.1
C		V1.1
C	*****	V1.1
	COMMON/POLYGON/NPLANE, IFLAG, NPPP(90), PO(3,4,60), P(3,4,60),	V1.1
	*CONVEC(2,4,90), POS(2,90), SIGN(90)	V1.1
	COMMON/ELLIPSE/NSTEPS(90), IELP, A(3,3,30), SEGLP(3,90), VP(3),	V1.1
	*D(3,3,90), DVP(3,3), RA(3), NSEG, SEGLPO(3,90)	V1.2
	DIMENSION PPP2(3), PP2(3), PP1(3)	V1.1
	IF(NPLANE.EQ.0) RETURN	V1.1
	DO 40 I=1, NPLANE	V1.1
	II=I+30	V1.1
	NPTS=NPPP(II)	V1.1
	DO 35 K=1, NPTS	V1.1
	DO 10 J=1, 3	V1.1
	PPP2(J)=P(J,K,I)-VP(J)	V1.1
10	CONTINUE	V1.1
	CALL MAT(DVP, PPP2, PP2, 3, 3, 1, 3, 3, 3)	V1.1
	IF(K.NE.1) GO TO 16	V1.1
	DO 15 J=1, 2	V1.1
	POS(J, II)=PP2(J)/PP2(3)	V1.1
15	CONTINUE	V1.1
	GO TO 25	V1.1
16	DO 20 J=1, 2	V1.1
20	CONVEC(J,K-1, II)=PP2(J)/PP2(3)-PP1(J)/PP1(3)	V1.1
25	DO 30 J=1, 3	V1.1
30	PP1(J)=PP2(J)	V1.1
	IF(K.EQ.3) SIGN(II)=CONVEC(1,1, II)*CONVEC(2,2, II)-	V1.1
	1CONVEC(2,1, II)*CONVEC(1,2, II)	V1.1
35	CONTINUE	V1.1
	DO 40 J=1, 2	V1.1
	CONVEC(J, NPTS, II)=POS(J, II)-PP2(J)/PP2(3)	V1.1
40	CONTINUE	V1.1
	RETURN	V1.1
	END	V1.1

	SUBROUTINE PSE(X1,IN,SEG,INDEX,INDEX2,IHALF)	V1.1
	DIMENSION X1(3,INDEX,INDEX),IN(INDEX),SEG(3,3333)	V1.1
C		V1.1
C	THIS SUBROUTINE PLOTS A SEMIELLIPSOID.	V1.1
C	THE HALF OF THE ELLIPSOID PLOTTED DEPENDS UPON IHALF.	V1.1
C	IF IHALF = 1 X .GE. 0 IS PLOTTED.	V1.1
C	IF IHALF = 2 X .LT. 0 IS PLOTTED.	V1.1
	DO 100 I=IHALF,INDEX	V1.1
	LINE=INDEX-I+1	V1.1
	IF(IHALF.EQ.2) LINE=I	V1.1
	NPTS=IN(LINE)	V1.1
	DO 50 K=1,NPTS	V1.1
	DO 50 J=1,3	V1.1
	SEG(J,K)=X1(J,K,LINE)	V1.1
	IF(IHALF.EQ.2.AND.J.EQ.1) SEG(J,K)--SEG(J,K)	V1.1
50	CONTINUE	V1.1
	N=NPTS	V1.1
	IF(LINE.EQ.INDEX) GO TO 71	V1.1
	NPT=NPTS-1	V1.1
	DO 60 K=1,NPT	V1.1
	KK=NPT-K+1	V1.1
	N=N+1	V1.1
	SEG(1,N)=SEG(1,KK)	V1.1
	SEG(2,N)=SEG(2,KK)	V1.1
60	SEG(3,N)--SEG(3,KK)	V1.1
	NPT=N-1	V1.1
	DO 70 K=1,NPT	V1.1
	KK=NPT-K+1	V1.1
	N=N+1	V1.1
	SEG(1,N)=SEG(1,KK)	V1.1
	SEG(2,N)--SEG(2,KK)	V1.1
70	SEG(3,N)=SEG(3,KK)	V1.1
71	IPEN=3	V1.1
	CALL PNTPLT(SEG,IPEN,INDEX2,N)	V1.1
100	CONTINUE	V1.1
	RETURN	V1.1
	END	V1.1

	SUBROUTINE ROT(A,L,TH,M)		V1.1
C		REV 01 08/10/72	V1.1
C	COMPUTES ROTATION MATRIX A FOR ANGLE TH		V1.1
C	ABOUT X,Y OR Z AXIS AS L = 1,2, OR 3.		V1.1
C			V1.1
C	ARGUMENTS:		V1.1
C	A: 3X3 ROTATION MATRIX TO BE COMPUTED.		V1.1
C	L: 1,2 OR 3 TO ROTATE ABOUT X,Y OR Z AXIS.		V1.1
C	TH: ANGLE OF ROTATION IN RADIANS.		V1.1
C	M: 1ST DIMENSION OF A IN CALLING PROGRAM.		V1.1
C			V1.1
	DIMENSION A(M,3)		V1.1
	C= COS(TH)		V1.1
	S= SIN(TH)		V1.1
	IF (L.EQ.2) S = -S		V1.1
	DO 30 I=1,3		V1.1
	IF(I.EQ.3)GO TO 20		V1.1
	DO 10 J=1,2		V1.1
	A(I,J+1)=0.0		V1.1
	A(J+1,I)=0.0		V1.1
	IF(I+J+L.NE.5)GO TO 10		V1.1
	A(I,J+1)=S		V1.1
	A(J+1,I)=-S		V1.1
10	CONTINUE		V1.1
20	A(I,I)= C		V1.1
	IF(I.EQ.L)A(I,I)=1.0		V1.1
30	CONTINUE		V1.1
	RETURN		V1.1
	END		V1.1

SUBROUTINE SOLVA(R,AA11,AA22,AA12)	V1.1
DIMENSION R(2,3)	V1.1
A11=R(1,1)**2	V1.1
A12=2.0*R(2,1)*R(1,1)	V1.1
A13=R(2,1)**2	V1.1
A21=R(1,2)**2	V1.1
A22=2.0*R(2,2)*R(1,2)	V1.1
A23=R(2,2)**2	V1.1
A31=R(1,3)**2	V1.1
A32=2.0*R(2,3)*R(1,3)	V1.1
A33=R(2,3)**2	V1.1
DEL=A11*(A22*A33-A23*A32)-A12*(A21*A33-A23*A31)+	V1.1
1A13*(A21*A32-A22*A31)	V1.1
AA11=((A22-A12)*(A33-A23)-(A23-A13)*(A32-A22))/DEL	V1.1
AA12=((A23-A13)*(A31-A21)-(A21-A11)*(A33-A23))/DEL	V1.1
AA22=((A21-A11)*(A32-A22)-(A22-A12)*(A31-A21))/DEL	V1.1
IF(ABS(AA12).LT.1.E-11) AA12=.000001	V1.2
RETURN	V1.1
END	V1.1

	SUBROUTINE SOLVR(A1,A2,A3,A4,A5,A6,A7,A8,P,RX,RZ)	V1.1
C		V1.1
C	*****	V1.1
C		V1.1
C	THIS SUBROUTINE WILL SOLVE A SET OF SIMULTANEOUS EQUATIONS	V1.1
C	TO FIND COMPONETS OF VECTOR R THAT SATISFY THE PROPERTIES NEEDED	V1.1
C	TO DETURMINE THE EQUATION OF THE PROJECTED ELLIPSE.	V1.1
C		V1.1
C	SEE WRITEUP.	V1.1
C		V1.1
C	*****	V1.1
	DIMENSION P(3)	V1.1
	B=A1*P(1)+A2*P(2)+A3*P(3)	V1.1
	D=A4*P(1)+A5*P(2)+A6*P(3)	V1.1
	T1=A7*(D/B)**2+A6-2.0*A8*D/B	V1.1
	T2=2.0*A7*D/(B)**2-2.0*A8/B	V1.1
	T3=A7*(1/B)**2-1	V1.1
	RZ=(-T2+SQRT(T2**2-4.0*T1*T3))/(2.0*T1)	V1.1
	RX=-D*RZ/B-1.0/B	V1.1
	RETURN	V1.1
	END	V1.1

	SUBROUTINE TITLE	V1.1
	DIMENSION ICOLOR(21)	80386
	CHARACTER*4 ID(10,20)	80386
	COMMON/DEUG/IDEBUG(80),NISG,DEVFLG,ONLINE,TERM,BDRS,OFLINE	V1.1
	INTEGER ONLINE,DEVFLG,TERM,BDRS,OFLINE	V1.1
	NLINE=20	V1.1
	SIZE=.335	V1.1
	X=1.375	V1.1
	Y=10.0	V1.1
C		V1.1
C	INITIALIZE PLOTTING PACKAGE	V1.1
C		V1.1
	CALL PLOT(0.0,0.0,-3)	V1.1
	READ(5,1) NFRME,ICOLOR(21)	V1.1
1	FORMAT(I2,I2)	V1.1
	IF(NFRME.EQ.0) RETURN	V1.1
	DO 300 K=1,NFRME	V1.1
	DO 50 I=1,NLINE	V1.1
	READ(5,200) (ID(J,I),J=1,8),ICOLOR(I)	V1.1
	WRITE(6,200) (ID(J,I),J=1,8),ICOLOR(I)	V1.1
200	FORMAT(7A4,A2,I2)	V1.1
50	CONTINUE	V1.1
	DO 100 I=1,NLINE	V1.1
	Y=Y-.5	V1.1
	IF(DEVFLG.EQ.OFLINE.OR.DEVFLG.EQ.BDRS) CALL COLOR(ICOLOR(I),IERR) 80386	V1.1
	CALL SYMBOL(X,Y,SIZE,ID(1,I),0.,30)	V1.1
100	CONTINUE	V1.1
	IF(DEVFLG.EQ.ONLINE) CALL PLOT(12.,0.,-3)	V1.1
	IF(DEVFLG.EQ.TERM) CALL PLOT(0.0,0.0,-3)	V1.1
	IF(DEVFLG.EQ.BDRS) CALL NFRAME	V1.1
	IF(DEVFLG.EQ.OFLINE) CALL PLOT (14.,0.,-3)	V1.1
300	CONTINUE	V1.1
	RETURN	V1.1
	END	V1.1

	SUBROUTINE TPOINT(PP2,I,IN)	V1.1
	DIMENSION PP2(3),R(3),PP1(3)	V1.1
	COMMON/POLYGON/NPLANE,IFLAG,NPPP(90),PO(3,4,60),P(3,4,60),	V1.1
	*CONVEC(2,4,90),POS(2,90),SIGN(90)	V1.1
C		V1.1
C	THIS SUBROUTINE TESTS A POINT ON THE PROJECTION PLANE	V1.1
C	DEFINED BY PP2 AGAINST A POLYGON ON THE PROJECTION PLANE	V1.1
C	DEFINED BY I. A FLAG 'IN' IS RETURNED TO INDICATE IF THE	V1.1
C	POINT WAS INSIDE OR OUTSIDE THE POLYGON.	V1.1
C		V1.1
C	IN = 1 POINT WAS INSIDE POLYGON	V1.1
C		V1.1
C	IN = 2 POINT WAS OUTSIDE POLYGON	V1.1
C		V1.1
	IN=1	V1.1
	NPTS1=NPPP(I)	V1.1
	DO 20 JJ=1,2	V1.1
20	PP1(JJ)=POS(JJ,I)	V1.1
	DO 100 L=1,NPTS1	V1.1
	DO 30 N=1,2	V1.1
30	R(N)=PP2(N)-PP1(N)	V1.1
	SIGN2=CONVEC(1,L,I)*R(2)-CONVEC(2,L,I)*R(1)	V1.1
	IF(SIGN2*SIGN(I) .LT. 0.) GO TO 150	V1.1
	IF(ABS(SIGN2*SIGN(I)).LT.1.E-11) GO TO 150	V1.1
	DO 100 N=1,2	V1.1
100	PP1(N)=PP1(N)+CONVEC(N,L,I)	V1.1
	RETURN	V1.1
150	IN=2	V1.1
	RETURN	V1.1
	END	V1.1

SUBROUTINE TRANS1(R,P)	V1.1
COMMON/ELLIPSE/NSTEPS(90), IELP,A(3,3,30),SEGLP(3,90),VP(3),	V1.1
*D(3,3,90),DVP(3,3),RA(3),NSEG,SEGLPO(3,90)	V1.2
COMMON/VIEWP/VP0(3),DVPO(3,3),IVP,VP2(3)	V1.1
DIMENSION DD(3,3),P(3),R(3),R2(3),SEGLP2(3),S(3)	V1.2
IF(IELP .GT. 30) GO TO 10	V1.1
CALL DOTT(DVP,D(1,1,IELP),DD,3,3,3)	V1.1
20 CONTINUE	V1.1
CALL MAT(DD,R,R2,3,3,1,3,3,3)	V1.1
DO 2 I=1,3	V1.2
2 S(I) = SEGLP(I,IELP) + SEGLPO(I,IELP)	V1.2
CALL MAT(DVP,S,SEGLP2,3,3,1,3,3,3)	V1.2
DO 1 I=1,3	V1.1
1 P(I)=SEGLP2(I)+R2(I)-VP2(I)	V1.1
RETURN	V1.1
10 DO 11 I=1,3	V1.1
DO 11 J=1,3	V1.1
11 DD(I,J) = DVP(I,J)	V1.1
GO TO 20	V1.1
END	V1.1

	REAL FUNCTION XINTCP(X, Y, XSAV, YSAV, YTEMP)	V1.1
C	*****	V1.1
C		V1.1
C	THIS FUNCTION CALCULATES THE X INTERCEPT AT YTEMP	V1.1
C		V1.1
C	*****	V1.1
	X1=X-XSAV	V1.1
	Y1=Y-YSAV	V1.1
	IF (Y1.NE.0.0) PFACTR=X1/Y1	V1.1
	IF (ABS(Y1).LT.1.E-11) PFACTR=0.0	V1.2
	Y2=YTEMP-YSAV	V1.1
	XINTCP=Y2*PFACTR+XSAV	V1.1
	RETURN	V1.1
	END	V1.1

```

SUBROUTINE XYZ(MU,A,B,C,S,M,JFLAG)
DIMENSION MU(3),S(3),M(3,2)
REAL M,MU
C
C *****
C
C     ALPHA AND BETA RELATE THE Y AND Z COMPONENTS OF M TO
C     THE X COMPONENT OF M.
C *****
C
C     ALPHA = MU(2) / MU(1)
C     BETA = MU(3) / MU(1)
C
C *****
C
C     SOLVE THE EQUATION FOR ELLIPSE 'N' TO FIND A POINT ON
C     THE ELLIPSE THAT WILL DETERMINE A VECTOR M.
C *****
C
C     T1=A+B*ALPHA*ALPHA+C*BETA*BETA
C     T2=2*A*S(1)+2*B*ALPHA*S(2)+2*C*BETA*S(3)
C     T3=A*S(1)**2+B*S(2)**2+C*S(3)**2-1
C     TEMP= T2 * T2 - 4 * T1 * T3
C
C *****
C
C     NO SOLUTION MEANS ELLIPSOID 'N' NOT TOUCHED BY LINE OF
C     SIGHT RAY.
C *****
C
C     IF(TEMP.LT.0.0) GO TO 2
C     TEMP=SQRT(TEMP)
C
C *****
C
C     FIND TWO POSSIBLE M VECTORS BECAUSE A RAY ENTERING A
C     SOLID MUST ALSO LEAVE.
C *****
C
C     M(1,1)=(T2+TEMP)/(2*T1)
C     M(2,1)=ALPHA*M(1,1)
C     M(3,1)=BETA*M(1,1)
C     M(1,2)=(T2-TEMP)/(2*T1)
C     M(2,2)=ALPHA*M(1,2)
C     M(3,2)=BETA*M(1,2)
C     JFLAG=0
C     RETURN
C 2 JFLAG=1
C     RETURN

```

END

V1.1

	REAL FUNCTION YINTCP(X,Y,XSAV,YSAV,XTEMP)	V1.1
C	*****	V1.1
C		V1.1
C	THIS FUNCTION CALCULATES THE Y INTERCEPT AT XTEMP	V1.1
C		V1.1
C	*****8	V1.1
	X1=X-XSAV	V1.1
	Y1=Y-YSAV	V1.1
	IF (X1.NE.0.0) PFACTR=Y1/X1	V1.1
	IF (ABS(X1).LT.1.E-11) PFACTR=0.0	V1.2
	X2=XTEMP-XSAV	V1.1
	YINTCP=X2*PFACTR+YSAV	V1.1
	RETURN	V1.1
	END	V1.1


```

SUBROUTINE YZ(MU,A,B,C,S,M,JFLAG)                                V1.1
DIMENSION MU(3),S(3),M(3,2)                                    V1.1
REAL MU,M                                                        V1.1
C      *****                                                    V1.1
C      ALPHA RELATES THE Z COMPONET TO THE Y COMPONET          V1.1
C      *****                                                    V1.1
C      ALPHA=MU(3)/MU(2)                                         V1.1
C      *****                                                    V1.1
C      SOLVE THE EQUATION FOR ELLIPSE 'N' WHEN X=0 TO          V1.1
C      FIND A POINT ON THE ELLIPSE THAT WILL DETERMINE A VECTOR M. V1.1
C      *****                                                    V1.1
C      T1=B+C*ALPHA*ALPHA                                       V1.1
C      T2=2*B*S(2)+2*C*ALPHA*S(3)                               V1.1
C      T3=A*S(1)**2+B*S(2)**2+C*S(3)**2-1                      V1.1
C      TEMP=T2*T2-4*T1*T3                                       V1.1
C      *****                                                    V1.1
C      NO SOLUTION MEANS ELLIPSOID 'N' NOT TOUCHED BY LINE    V1.1
C      OF SIGHT RAY.                                           V1.1
C      *****                                                    V1.1
C      IF(TEMP.LT.0.0) GO TO 2                                   V1.1
C      TEMP=SQRT(TEMP)                                          V1.1
C      *****                                                    V1.1
C      FIND TWO POSSIBLE M VECTORS BECAUSE A RAY ENTERING      V1.1
C      A SOLID MUST ALSO LEAVE.                                 V1.1
C      *****                                                    V1.1
C      M(2,1)=(T2+TEMP)/(2*T1)                                   V1.1
C      M(1,1)=0.0                                               V1.1
C      M(3,1)=ALPHA*M(2,1)                                       V1.1
C      M(2,2)=(T2-TEMP)/(2*T1)                                   V1.1
C      M(1,2)=0.0                                               V1.1
C      M(3,2)=ALPHA*M(2,2)                                       V1.1
C      JFLAG=0                                                  V1.1
C      RETURN                                                    V1.1
2 JFLAG=1                                                       V1.1
C      RETURN                                                    V1.1
C      END                                                        V1.1

```

	SUBROUTINE Z(MU,A,B,C,S,M,JFLAG)	V1.1
	DIMENSION MU(3),S(3),M(3,2)	V1.1
	REAL M,MU	V1.1
C	*****	V1.1
C		V1.1
C	SOLVE THE EQUATION FOR ELLIPSE 'N' WHEN X=0 AND	V1.1
C	Y=0 TO FIND A POINT ON THE ELLIPSE THAT	V1.1
C	WILL DETERMINE A VECTOR M.	V1.1
C		V1.1
C	*****	V1.1
	T1=C	V1.1
	T2=2*C*S(3)	V1.1
	T3=A*S(1)**2+B*S(2)**2+C*S(3)**2-1	V1.1
	TEMP=T2*T2-4*T1*T3	V1.1
C	*****	V1.1
C		V1.1
C	NO SOLUTION MEANS ELLIPSOID 'N' NOT TOUCHED BY LINE	V1.1
C	OF SIGHT RAY.	V1.1
C		V1.1
C	*****	V1.1
	IF(TEMP.LT.0.0) GO TO 2	V1.1
	TEMP=SQRT(TEMP)	V1.1
C	*****	V1.1
C		V1.1
C	FIND TWO POSSIBLE M VECTORS BECAUSE A RAY ENTERING	V1.1
C	A SOLID MUST ALSO LEAVE.	V1.1
C		V1.1
C	*****	V1.1
	M(1,1)=0.0	V1.1
	M(2,1)=0.0	V1.1
	M(3,1)=(T2+TEMP)/(2*T1)	V1.1
	M(1,2)=0.0	V1.1
	M(2,2)=0.0	V1.1
	M(3,2)=(T2-TEMP)/(2*T1)	V1.1
	JFLAG=0	V1.1
	RETURN	V1.1
2	JFLAG=1	V1.1
	RETURN	V1.1
	END	V1.1