AMRL-TR-75-14

DEVELOPMENT OF AN IMPROVED COMPUTER MODEL OF THE HUMAN BODY AND EXTREMITY DYNAMICS

CALSPAN CORPORATION
BUFFALO, NEW YORK 14221

JULY 1975

AEROSPACE MEDICAL RESEARCH LABORATORY
AEROSPACE MEDICAL DIVISION
Air Force Systems Command
Wright-Patterson Air Force Base, Ohio

NOTICES

FOR THE COMMANDER

HENNING E. VON GIERKE
Director
Biodynamics and Bionics Division
Aerospace Medical Research Laboratory

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER AMRL-TR-75-14	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) DEVELOPMENT OF AN IMPROVED COMPUTER MODEL OF THE HUMAN BODY AND EXTREMITY DYNAMICS		5. TYPE OF REPORT & PERIOD COVERED Technical Report 6. PERFORMING ORG. REPORT NUMBER Y11 series
7. AUTHOR(s) J. T. Fleck F. E. Butler		8. CONTRACT OR GRANT NUMBER(s) F33615-75-C-5002
9. PERFORMING ORGANIZATION NAME AND ADDRESS Calspan Corporation Buffalo, New York 14221		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62202F, 7231-05-19
11. CONTROLLING OFFICE NAME AND ADDRESS Aerospace Medical Research Laboratory Aerospace Medical Division Wright-Patterson Air Force Base, Ohio 45433		12. REPORT DATE July 1975 13. NUMBER OF PAGES 164
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) None 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Phase III Calspan Three-Dimensional Crash Victim Model Simulation was sponsored by the National Highway Safety Administration, Dept. of Transportation. The ground work for the simulation was performed in Phases I and II, both jointly sponsored by NHTSA and the Motor Vehicle Manufacturers Association.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computer Simulation - Three Dimension Dynamics
Mathematical Model - Vehicle Crash Victims
Harness Restraint System
Aerodynamic Forces

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Three principal modifications; namely, an improved joint formulation, an improved belt restraint formulation and inclusion of aerodynamic forces, were incorporated into the Phase III Calspan Three-Dimensional Crash Victim Model Simulation.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

AIR FORCE - 26-8-75 - 100

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

AD-A014816

PREFACE

This report describes the modifications which were incorporated into the Phase III* Calspan Three-Dimensional Crash Victim Simulation Program to satisfy current Air Force requirements.

Three principal modifications are:

1. Improved Joint Formulation
2. Improved Belt Formulation
3. Inclusion of Aerodynamic Forces.

The modifications have been made so that they may be used on the CDC6600 computer at the Mathematics and Analysis Branch of AMRL.

The research effort summarized in this report was performed for the Aerospace Medical Research Laboratory [FY8990] under Contract No. Calspan F33615-75-C-5002. Dr. John T. Fleck of the Computer Mathematics Department of Calspan served as principal investigator.

The authors wish to thank Ints Kaleps of the Aerospace Medical Research Laboratory for his suggestions and direction during the analytical development of the program.

---

*Phase III was sponsored by the National Highway Traffic Safety Administration, Department of Transportation. The ground work for the simulation was performed in Phases I and II, both jointly sponsored by NHTSA and the Motor Vehicle Manufacturers Association.

TABLE OF CONTENTS

LIST OF FIGURES

Section 1

INTRODUCTION

The Calspan 3-D Crash Victim Simulation Model was originally developed to study human body dynamics associated with automobile accidents. The formulation, however, is quite general, giving it great versatility and making it applicable to many studies involving human body dynamics. Reference 1 contains a complete description of this model.

To fit the specific needs of the Mathematics and Analysis Branch (BBM) of the Aerospace Medical Research Laboratory (AMRL), three principal modifications have been made to the program. These are: an improved joint formulation, an improved belt restraint formulation and the inclusion of aerodynamic forces.

The modifications are described in the following sections.

Section 2

JOINT ALGORITHM

The joint routine, subroutine VISPR, which is in the Calspan 3-D Crash Victim Model, has been modified to provide the option of computing the flexure torque as a function of both the flexure angle (elevation) and azimuth angle.

NOMENCLATURE

D_m	3 x 3 direction cosine matrix specifying the orientation of segment m's local reference with respect to the inertial reference.
$T_{m,n}$	3 x 3 direction cosine matrix specifying the relative orientation of joint n's local reference with respect to the local reference of segment m.
T	3 x 3 direction cosine matrix specifying the relative orientation of joint's local reference systems, T=I, the identity matrix, is the equilibrium position.
T_{ij}	ij^{th} element of matrix T.
$r_{m,n}$	3 x 1 matrix (vector) specifying the location of joint n as measured in segment m's local reference.
x,y,z x_m, y_m, z_m	used to designate axes of a right handed coordinate system may be regarded as 3 x 1 matrix (vector) which is of unit magnitude and is orthogonal.

θ flexure angle of joint.

ψ torsion angle of joint.

ϕ azimuth angle used to describe flexure torque asymmetrically.

μ 3 x 1 matrix (vector) of unit magnitude used to designate axis of flexure.

Joint Routine

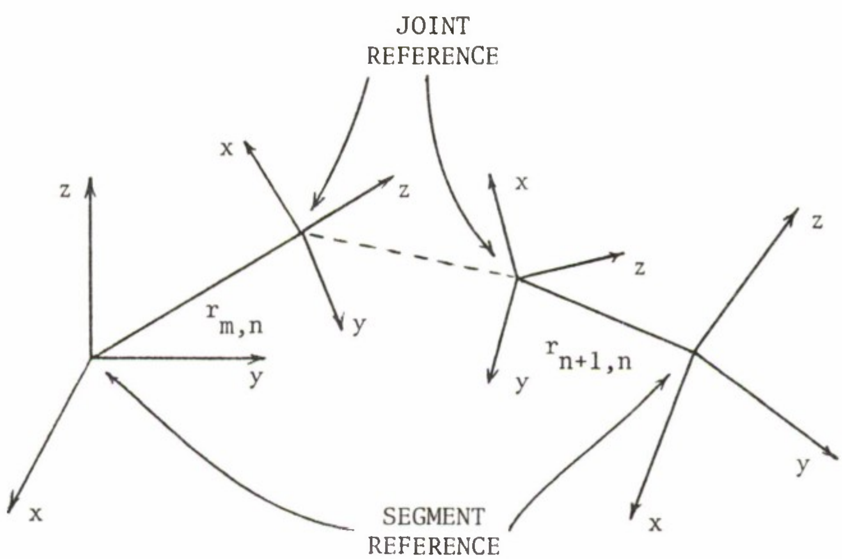


FIGURE 1 Joint Coordinate System

The position of joint n, which is fixed in segment m, is given by vector $r_{m,n}$ (see Figure 1). The orientation of the joint with respect to segment m's reference system is given by the direction cosine matrix $T_{m,n}$. The matrix $T_{m,n}$ is computed from the yaw (about Z), pitch (about Y), and roll (about X) angles, which are specified on input along with the vector $r_{m,n}$.

Joint n connects segments m and n+1. The vector $r_{n+1,n}$ and the matrix $T_{n+1,n}$ are determined from input as were $r_{m,n}$ and $T_{m,n}$.

For the relative orientation of the joint we have

$$TT_{m,n} D_m = T_{n+1,n} D_{n+1}$$

$$T = T_{n+1,n} D_{n+1} (T_{m,n} D_m)^{1}$$

where D_m, D_{n+1} are the direction cosine matrices specifying the orientation of the segments and T is the direction cosine matrix specifying the relative orientation of the joint, and where A^{1} is the transpose of A.

$T_{m,n}$ and $T_{n+1,n}$ are defined so that the equilibrium position of the joint occurs when T=I, the identity matrix.

Consider the following figure

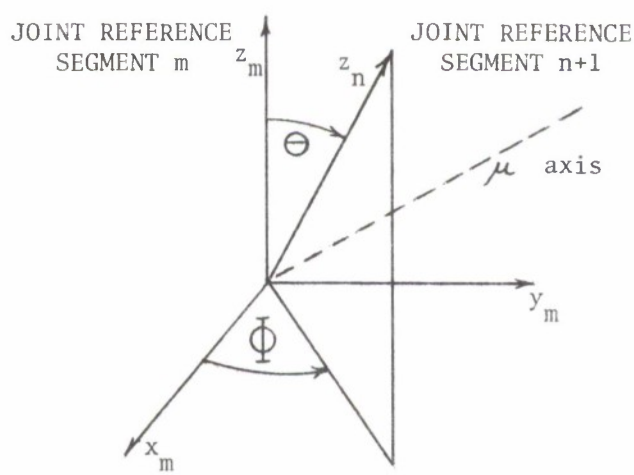


FIGURE 2 Joint Flexure

The angle, θ, between the Z axes of the joint reference systems is defined as the flexure angle of the joint. The angle, Φ, between the projection of Z_n in the $X_m Y_m$ plane and the X_m axis is defined as the azimuth angle. A twist (torsion) angle, ψ, may be defined as either a rotation about Z_m or a rotation about Z_n.

If the joint is pinned (hinge joint), the pin axis is taken as the Y axis, hence Y_m is parallel to Y_n for this pinned joint. In this case, Φ may either be 0 or π.

Flexure

The axis of flexure, μ, may be computed as the vector cross product of Z_m and Z_n.

That is $\mu = Z_m \otimes Z_n / \left| Z_m \otimes Z_n \right|$.

Note that when $\theta=0$ or $\theta=\pi$, μ is undefined. The case $\theta=\pi$ will not be considered. We will assume that $0 \leq \theta < \pi$. That is, any flexure angle equal or greater than π will never occur.

The direction cosines of a vector in the direction of Z_n with respect to joint reference m are given by the third row of the matrix T (i.e. the third row of T is a unit vector in the Z_n direction.)

We have

$$\sin\theta \cos\Phi = T_{31}$$

$$\sin\theta \sin\Phi = T_{32}$$

$$\cos 0 = T_{33}$$

Hence

$$\theta = \cos^{-1} T_{33}$$

$$\Phi = \tan^{-1}\left(\frac{T_{32}}{T_{31}}\right) \quad \text{if } \theta \neq 0$$

$$\mu = \begin{pmatrix} -T_{32} \\ T_{31} \\ 0 \end{pmatrix}$$

where T_{ij} is the i,j^{th} element of the matrix T.

Define the matrix T_μ which represents a rotation of θ about the axis μ. We have, in matrix form,*

$$T_\mu = \mu\mu^1 + (I - \mu\mu^1)\cos\theta - \sin\theta\,\mu\otimes$$

where $\mu\otimes$ is the matrix

$$\mu\otimes = \begin{pmatrix} 0 & -\mu_z & \mu_y \\ \mu_z & 0 & -\mu_x \\ -\mu_y & \mu_x & 0 \end{pmatrix}$$

In our case, $\mu_x = -T_{32}$, $\mu_y = T_{31}$ and $\mu_z = 0$.

We compute the restoring torque for flexure, $f(\theta,\Phi)$, as a function of the flexure angle, θ, and the azimuth, Φ. The torque, $+ f(\theta,\Phi)\mu$, will be applied to segment m and the torque, $-f(\theta,\Phi)\mu$, will be applied to segment n+1. We assume that when $\theta = 0$, $f(\theta,\Phi) = 0$, hence the fact that Φ and μ are undefined in this case will be of no consequence.

Representation of Flexure Torque

We use the following approximation for the torque function $f(\theta,\Phi)$.

The function, $f(\theta,\Phi)$, is represented as a continuous or tabular function of θ for discrete values of Φ.

That is, $f(\theta,\Phi_n) = g_n(\theta)$ n=1,N

and where $\Phi_1 = -\pi$, $\Phi_2, \ldots \Phi_N, \Phi_{N+1}$, ($\Phi_{N+1} = \pi$), are equally spaced between $-\pi$ and π and it is assumed that $f(\theta,\pi) = f(\theta,-\pi)$ therefore Φ_{N+1} is not required. (The range $-\pi$ to π is used to be consistent with the four quadrant arctan routines, which are used to evaluate Φ.)

* Reference 1, Volume 1, page 23.

The value of N will be restricted only by the storage one is willing to allocate and the computing time involved.

The function g_n may be defined as the m^{th} degree polynomial in $(\theta-\theta_\circ)$ or as a table. (They cannot be mixed, i.e. for a particular joint all g must be tabular or polynomial.)

In both cases, a deadband may be specified, i.e. a $\theta_{n_\circ}$ is given and $g_n(\theta)=0$ if $\theta<\theta_{n_\circ}$.

For intermediate values of Φ (i.e., $\Phi_n < \Phi < \Phi_{n+1}$), we evaluate f for g_n and for g_{n+1} and linearly interpolate on Φ. The wrap around if $\Phi > \Phi_N$ or $\Phi < \Phi_1$ is treated consistently (i.e., interpolate between g_N and g_1.)

Twist (Torsion)

When there is no flexure, i.e. $\theta=0$, the twist ψ is easily defined as the rotation about the Z_m axis. In this case

$$T=T_Z(\psi)=\begin{pmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

When θ is not equal to zero, we may think of first twisting about Z_m and then rotation about the axis μ

$$T=T_\mu(\theta)T_Z(\psi)$$

or we may first rotate about μ and then twist about the resultant Z, i.e. Z_n.

$$T=T_Z(\psi)T_\mu(\theta)$$

These definitions are equivalent and thus give a unique definition of the angle ψ. To show this, we expand T as

$$T=T_Z(\mu\mu^1+(I-\mu\mu^1)\cos\theta - \sin\theta\mu\otimes)$$

and as

$$T=(\mu_1\mu_1{}^1+(I-\mu_1\mu_1{}^1)\cos\theta-\sin\theta\mu_1\otimes)T_Z$$

where $\mu_1 = T_Z \mu$, i.e. the axis of rotation is fixed in the X_m, Y_m plane. Substituting for the value of μ_1 in the above expression shows the equivalence.

Although the angle ψ is well defined, it does not seem possible to uniquely define an axis, which may be used for the restoring torque. This problem exists because we are talking about a mathematical definition of twist and not a physical description of a joint.

The program has been coded to use the Z_m axis. The magnitude of the torque is computed by the standard spring function characteristics available in the program. This is done using subroutine EFUNCT.

The torque, q, is computed from the five parameters S_1, S_2, S_3, S_4 and S_5 by the following algorithms.

If $\psi \leq S_5$

$q = S_1 \psi$

If $\psi > S_5$ an additional torque q_s is computed as

$$q_s = S_2(\psi - S_5)^2 + S_3(\psi - S_5)^3$$

If $\dot{\psi} < 0$ (unloading) q_s is modified by $q_s = S_4 q_s$

(If S_5 is equal to zero, q_s is not computed but q is modified by S_4.)

For small values of $|\dot{\psi}|$, (10 radians/sec), the routine interpolates between the loading and unloading characteristics.

The total torque $q + q_s$ is returned as the function value.

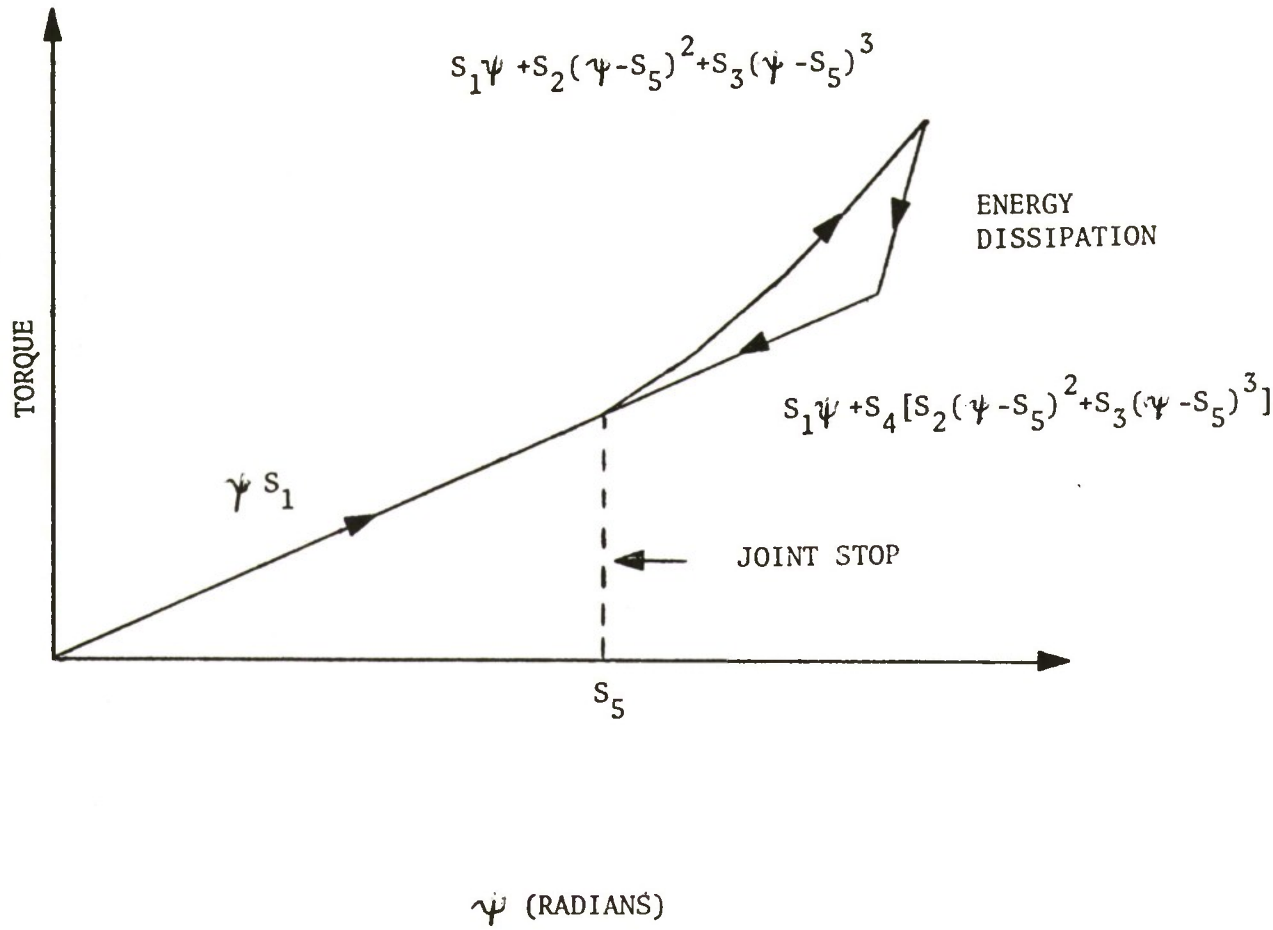
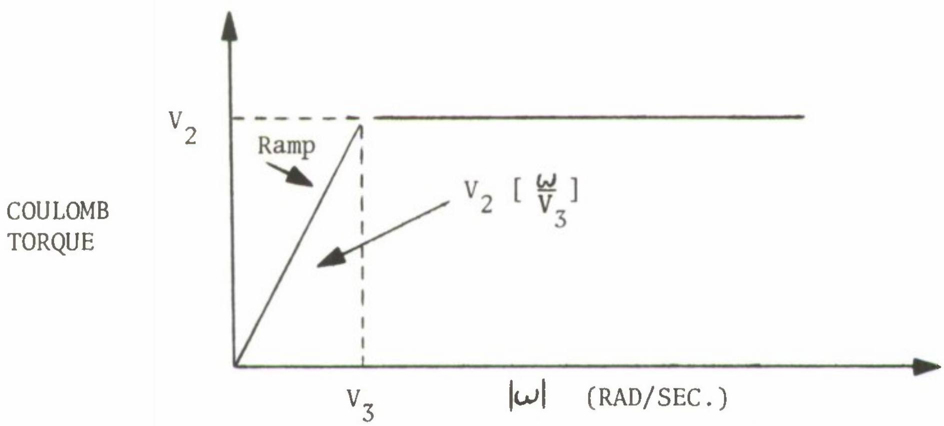


FIGURE 3 JOINT SPRING TORQUE

Viscous and Coulomb Torques

Let ω be the relative angular velocity. A torque is computed to oppose this velocity using the standard viscous function definition in the program. This is illustrated in Figure 4.



$$q/|\omega| = V_1 + V_2/(\max(|\omega|, V_3))$$

ω is the relative angular velocity.

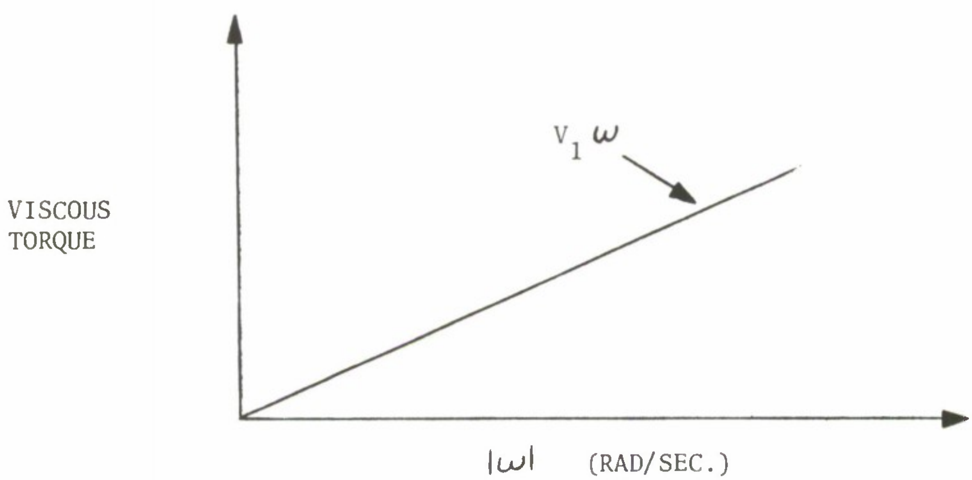


FIGURE 4 JOINT TORQUE DUE TO RELATIVE ANGULAR VELOCITY AT THE JOINT

Section 3

BELT ALGORITHM

The belt routine in Version III of the Calspan 3-D program is restricted to a simple belt passing around a single segment. Although several of these belts may be used, no provision for interaction of the belts was made.

To overcome this restriction and to satisfy the requirements of the current contract, an entirely new belt algorithm has been developed and incorporated into the program.

The current version of the algorithm assumes each belt lies essentially in a plane which may be described by a set of reference points rigidly attached to segments. Thus, its use should be restricted to harnesses (several belts connected at a common junction point) which constrain the segments involved from large relative motions.

The algorithm should lend itself to significant improvements in the modeling of harnesses.

The concept of a harness is introduced in this version of the program. A harness consists of from one to several belts. Each belt is defined as the set of straight line segments connecting prescribed reference points. One end of the belt is the anchor point and the other end the junction point. See Figure 5 below.

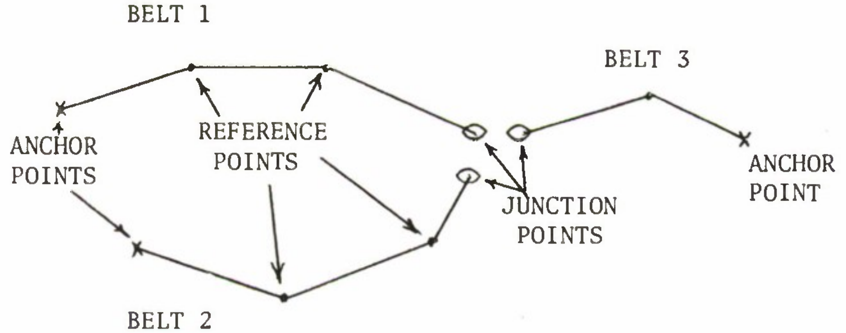


FIGURE 5 BELT HARNESS MODEL

Each reference point is fixed relative to a prescribed segment. An ellipsoid is assigned to each reference point. The ellipsoid is fixed to the same segment as the reference point. Ellipsoids associated with the anchor point and the junction point are ignored in the current version of the program hence may be specified as zero. The ellipsoid is used to determine an outward normal vector to the surface of the ellipsoid at the reference point. If the net force on the segment at this point has a positive component along this normal, the point will be ignored in computing the belt length. If no ellipsoid is specified for an interior point, this point will always be used in computing the belt length. For example, see Figure 6.

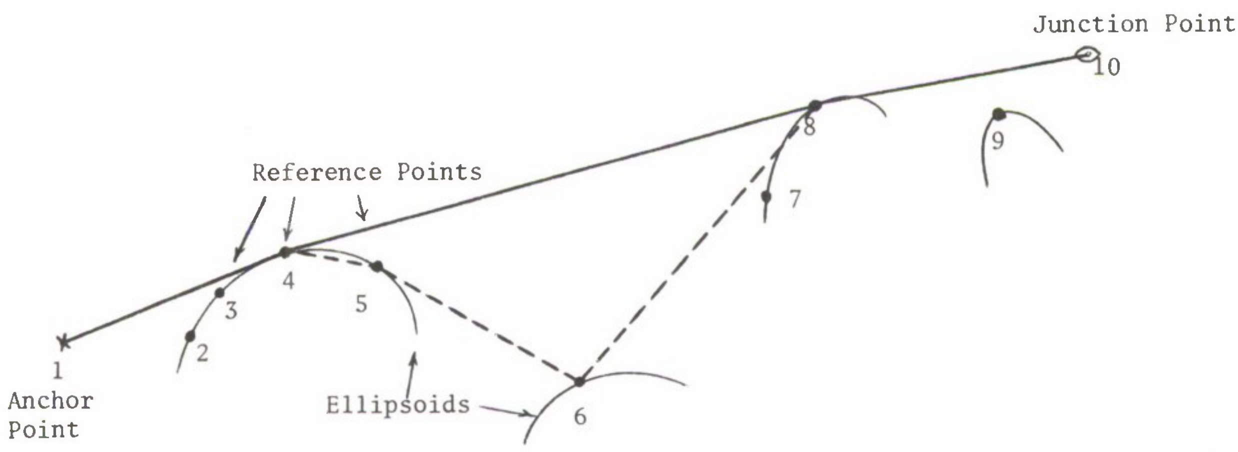


FIGURE 6 BELT LENGTH SPECIFICATION

In this example, the belt is defined by the 10 reference points illustrated. Ellipsoids associated with the interior points are represented by the curved lines. In the above configuration, the belt length would be computed as the sum of the length of the lines from 1 to 4, 4 to 8 and 8 to 10 as illustrated by the solid lines.

If no ellipsoid were specified for point 6, the belt would follow the dashed line, 1-4-5-6-8-10. The algorithm determines this belt configuration in the following manner:

1. The belt is first assumed to go from point 1 to point 2 to point 3.

2. The net force at point 2 (assuming constant tension) is found to be directed along the outward normal to the reference ellipsoid assigned to point 2 hence point 2 is dropped from consideration.

3. The belt is next assumed to go from point 1 to point 3 to point 4.

4. This is the same situation as was found in step 2 above hence point 3 is dropped.

5. The belt is next assumed to go from point 1 to point 4 to point 5.

6. The net force at point 4 is directed along the inward normal to the ellipsoid assigned to point 4 hence point 4 is accepted.

7. The belt is next assumed to go from point 4 (the last accepted point) to point 5 to point 6.

8. The net force at point 5 is found to be directed along the inward normal to the ellipsoid assigned to point 5 hence point 5 is accepted.

9. The belt is next assumed to go from point 5 to point 6 to point 7.

10. The net force at point 6 is found to be directed along the outward normal to the ellipsoid assigned to point 6 hence point 6 is dropped from consideration.

11. Point 5 was accepted because 4,5,6 were acceptable, since 6 is now rejected and the belt is assumed to go from 4 to 5 to 7.

This process is continued resulting in the following condensed steps.

12. 5 is rejected.

13. Try 4-7-8.

14. 7 rejected.

15. Try 4-8-9.

16. 8 accepted.

17. Try 8-9-10.

18. 9 rejected.

19. Since 10 is a junction point, the final belt is 1-4-8-10.

Computation of Belt Tension

The strain is computed as

$$\text{strain} = \frac{\text{calculated length - reference length}}{\text{reference length}}$$

The stress (tension) is computed by the standard force deflection routines available in the program using strain as the deflection parameter.

The current version of the program assumes that the tension is uniform in the belt.

Examples

The above technique provides considerable versatility in defining belt systems. For example:

SIMPLE LAP BELT

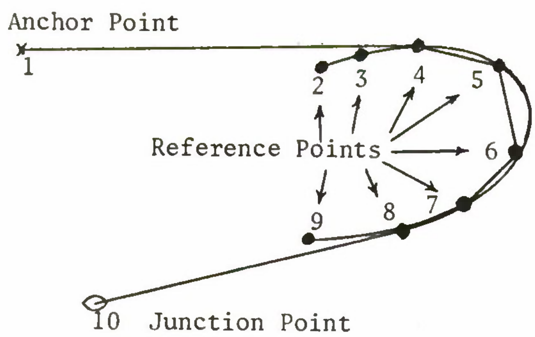


FIGURE 7 Simple Lap Belt Harness Configuration

Define a harness consisting of one belt. In this case, the junction point is actually an anchor point. All interior reference points are attached to the same segment and assigned the same reference ellipsoid.

SHOULDER BELT AND LAP BELT

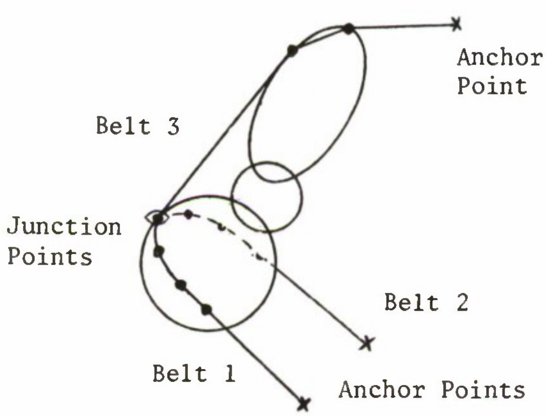


FIGURE 8 Shoulder Belt and Lap Belt Harness Configuration

Define a harness consisting of three belts with the junction points attached to a segment, which is disjoint (but need not be) from the body segments. This segment is assigned a mass and an inertia tensor and will move dynamically to achieve a force balance. (we recommend that the junction points all be located at the cg of the junction segment to prevent large angular accelerations.)

The Input Description of the program contains the details of inputting a belt system.

General Comments

Since there is no limitation (except storage) on the number of harnesses, belts or reference points, quite elaborate belt systems may be modeled.

The program is so written that it may be modified to include effects of belt friction and deformation of the surface at the reference points. For example, the reference points could be moved along the normal as a function of the normal force.

The belt is not constrained to lie in a plane. The algorithm, as illustrated in Figure 5 , was designed on the assumption that the interior reference points lie essentially in a plane. Highly, non planar sets of points may produce unexpected results. No study has been made of this potential problem or of other unusual configurations that would cause the algorithm to fail.

The computation of frictional effects and deformation is complicated by the fact that a change in belt position or tension at one point affect all the points. Thus, the problem would require the use of techniques such as finite element methods. However, a first approximation to the effects of deformation could be made by holding the reference points fixed during the course of an integration step and at the completion of a successful integration step the points could be moved along the normal (as defined by the reference ellipsoid) as a function of the normal force computed from the current belt configuration. Alternately, the point could be moved in the direction of the net force. Storage has been allocated in the program to store a fixed reference point and a modified reference point. Perhaps the effects of friction could be approximated by allowing the modified reference point to move in a direction other than along the normal. Future study should consider these possibilities.

Section 4

AERODYNAMIC FORCES

Routines have been added to the program to allow the application of a specified force to any segment. The method allows any force to act on any segment. In addition, for each segment a boundary plane is specified and the force is not applied until the segment penetrates the boundary plane.

An aerodynamic pressure, a boundary plane, and an ellipsoid are associated with each segment for which it is desired to compute an aerodynamic force. The aerodynamic pressure, as a function of time, is inputted as tabular data.

As the ellipsoid penetrates the boundary plane, an estimate of the projected area normal to the pressure is made and the force and torque are computed and applied to the segment. For partial penetration, the force is applied at a point in the ellipsoid. At full penetration, the force is applied at the center of the ellipsoid.

MATHEMATICAL FORMULATION

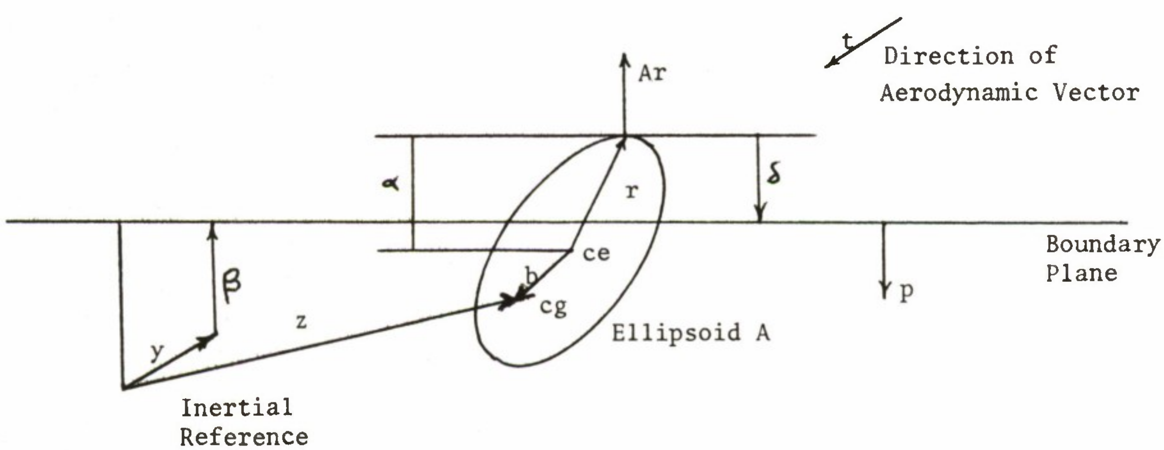


FIGURE 9 BOUNDARY PLANE PENETRATION BY ELLIPSOID

Let y location of reference point for plane

- z location of cg of segment
- b offset of center of ellipsoid from cg
- δ penetration distance
- p unit vector describing outward normal to boundary plane
- r vector from center of ellipsoid to point of maximum penetration
- α distance from center of ellipsoid to point of maximum penetration
- t vector describing wind (force per unit area)
- A matrix defining the ellipsoid (a 3x3 positive definite matrix.)
- β distance of plane from its reference point

We have the following equations:

$r \cdot Ar=1$ if r is on the ellipsoid

$p \cdot r=-\alpha$

$p \cdot [z+b+r]=-\beta+\delta+p \cdot y$

Ar vector normal to ellipsoid at r.

At a given instant in time we know y,z,b,p,t and A.

In the computer program, the ellipsoid matrix is a given constant in the reference system of the segment. All quantities are first converted to this reference system for ease of computation.

COMPUTATION OF PENETRATION DISTANCE

If r goes to the point of maximum penetration, then

$Ar=c\,p$, where c is some constant

since $r \cdot Ar=1$ if r is on the ellipsoid

$r=cA^{-1}p$

$r \cdot Ar=c^2 p \cdot A^{-1}p=1$

then $c = 1/\sqrt{p \cdot A^{-1} p}$

and $r = A^{-1}p/\sqrt{p \cdot A^{-1} p}$

The penetration distance δ may be computed as

$\delta = p \cdot [z+b+r-y] - \beta$,

and α may be computed as

$\alpha = p \cdot r = \sqrt{p \cdot A^{-1} p}$

If δ is less than zero, no penetration has occurred.

If δ is greater than 2α the ellipsoid has fully penetrated the plane.

COMPUTATION OF PROJECTED AREA

If penetration has occurred, we must distinguish between three cases in the computation of the shadow (projected) area of the ellipsoid onto the plane.

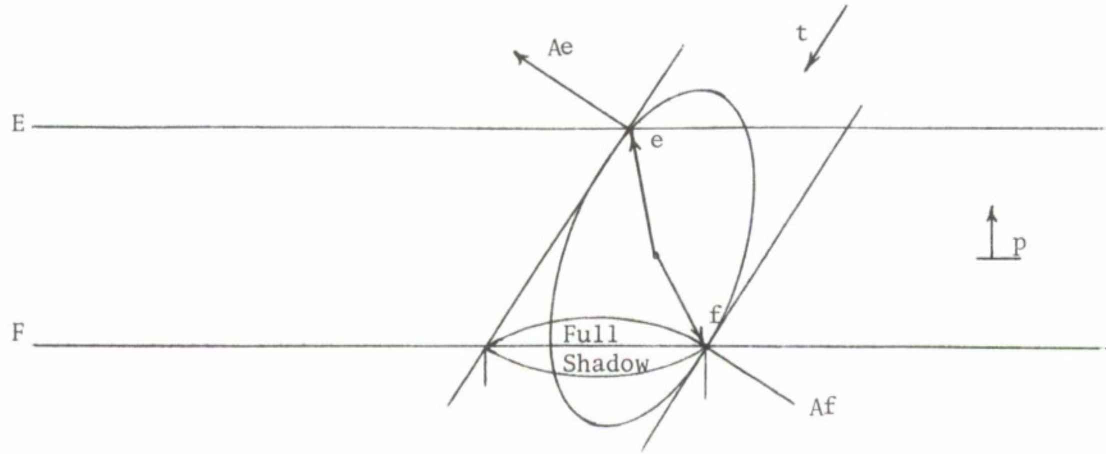


FIGURE 10 SHADOW (PROJECTED) AREA OF ELLIPSOID ONTO A PLANE

Consider the above figure where planes E and F are parallel to the boundary plane p and are such that at the points e and f which are on the ellipsoid and in the indicated planes we have $t \cdot Ae=0$, $t \cdot Af=0$ and the planes are positioned that $|p \cdot e|$ and $|p \cdot f|$ are at their maximum values (i.e. if the planes were moved further away from the center of the ellipsoid, no such points could be found.)

<u>Consider the Three Cases</u>

Case I The boundary plane is above plane E but still intersects the ellipsoid. In this case, the projected area is the area of the ellipsoid formed by the intersection with the plane p projected on a plane normal to t.

Case II The boundary plane is between planes E and F. This is a region where the projected area is made up of parts of two ellipses. One is the shadow ellipse and the other is the ellipse formed by the intersection of the ellipsoid and the plane p.

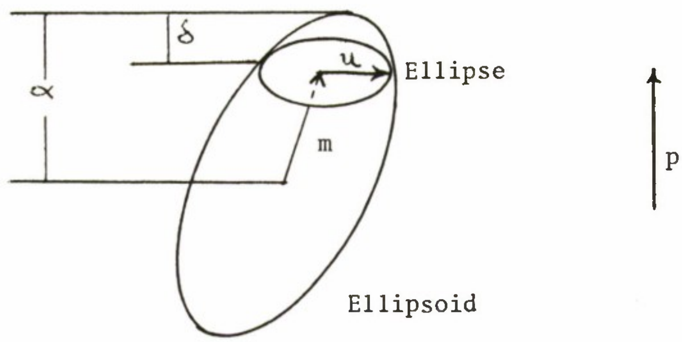
Case III The boundary plane is below plane F. In this case, the full shadow ellipse is produced and does not alter with the penetration distance or orientation of the boundary plane.

Computation in the Above Cases

It may be shown that

$$p \cdot e = (p \cdot A^{-1} p - (p \cdot t)^2 / t \cdot At)^{1/2} = -p \cdot f$$

Case I This case exists when $\delta > 0$ and $\alpha > p \cdot e + \delta$



The center of the ellipse of intersection is at m where

$$m = (\alpha - \delta) A^{-1} p / p \cdot A^{-1} p$$

FIGURE 11 Intersection Ellipsoid

The equation of a point u on this ellipse is

$$u \cdot Bu = 1$$

where u is measured from the center of the ellipse hence lies in the plane of intersection, and

$$B = (I - pp\cdot) A (I - pp\cdot) / (1 - m \cdot Am)$$

$$m \cdot Am = (\alpha - \delta)^2 / p \cdot A^{-1} p$$

The matrix B is singular (has a zero eigenvalue) but the product of the two non zero eigenvalues is the reciprocal of the square of the product of the major and minor axis of the ellipse. The area is the product of π times the product of the major and minor axes.

We have

P = product of eigenvalues = $[(tr(B))^2 - tr(B^2)]/2$

where tr(B) is the trace (sum of diagonal elements) of the matrix B. For a matrix such as B, the product of the eigenvalues can most readily be computed as the sum of the principle minors.

Hence,

$\text{Area} = \pi (1-m\cdot Am)/p^{1/2}$

The area normal to t is then equal to $|p\cdot t|$.

The point of force application will be taken as the center of the ellipse (i.e. At m)

Case III Full Shadow

This exists when $\delta > 0$ and $\delta > p\cdot e + \alpha = \alpha - p\cdot f$

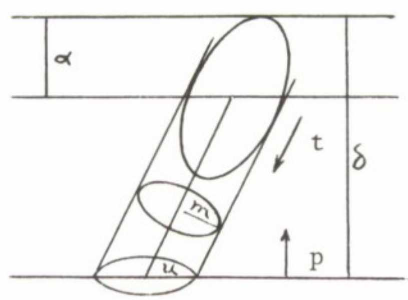


FIGURE 12 Full Shadow

At point u on the shadow ellipsoidal cone is measured from the center of the ellipse in a plane $\perp$ to t.

$u\cdot C\, u = 1$

where $C = A - At(At)\cdot/t\cdot At$

The product p of the nonzero eigenvalues is the sum of the principle minors of C. The projected area is then $\text{Area}=\pi/p^{1/2}$

The force is applied at center of ellipsoid.

Case II   Mixed Region

The exact calculation here is involved since it involves the computation of areas of partial ellipses. Since the method of aerodynamic force computation is only an approximation, we decided that it is reasonable to compute the projected area and the point of force application in the following manner:

1. Referring to Figure 9, if δ is less than zero, no penetration has occurred so no aerodynamic force will be applied.

2. If δ is positive, a scale factor is computed as (Figure 10)

   Scale = $(\alpha - \delta + |p \cdot e|)/(\alpha + |p \cdot e|)$

   If scale is greater than one, no penetration has occurred ($\delta < 0$). If scale is less than zero, the ellipsoid has penetrated enough that Case III applies. In this case, scale is set to zero.

3. The full projected Area is computed by the formula in Case III
   Area $= \pi / p^{1/2}$.

   The effective area is computed as
   Effective Area = (1-Scale^2)*Area

4. The point of force application is computed as
   q=Scale*r
   where r is the vector from the center of the ellipsoid to the point of maximum penetration (see Figure 9.)

5. The aerodynamic force is computed by interpolating the given table of pressure for the proper time and multiplying this pressure by the Area.

6. The force and torque are then applied to the segment.

Section 5

REFERENCES

1. Fleck, J.T., Butler, F.E., and Vogel, S.L., "An Improved Three-Dimensional Computer Simulation of Motor Vehicle Crash Victim," Calspan Technical Report No. ZQ-5180-L-1, July 1974. Vol. I Engineering Manual, Vol. II Model Validation, Vol. III User's Manual, Vol. IV Programmer's Manual.

2. Bartz, J.A., "A Three-Dimensional Computer Simulation of a Motor Vehicle Crash Victim' Phase 1 Development of the Computer Program" Calspan Technical Report No. VJ-2978-U-1, July 1971.

3. Bartz, J.A., and Butler, F.E., "A Three-Dimensional Computer Simulation of a Motor Vehicle Crash Victim' Phase 2 Validation of the Model, "Calspan Technical Report No. VJ-2978-V-2, December 1972.

APPENDIX A

REV 12 12/19/74

INPUT DESCRIPTION FOR THE CALSPAN 3-D CRASH VICTIM SIMULATION PROGRAM AS SUPPLIED TO WRIGHT PATTERSON A.F.B. (CONTRACT NO. F33615-75-C-5002)

NOTE: THIS REPORT IS SUPPLIED WITH 'I' IN COLUMN 1 FOR PAGE SKIP CONTROL TO ALLOW FOR PRINTING ON VARIOUS COMPUTER SYSTEMS.

THE FOLLOWING SPECIAL SYMBOLS MAY DIFFER ON OTHER SYSTEMS:

"#" IS USED TO INDICATE "NOT EQUAL".
"<" IS USED TO INDICATE "LESS THAN".
">" IS USED TO INDICATE "GREATER THAN".
"|" IS USED TO INDICATE "ABSOLUTE VALUE".

ANY LINE WITH A "|" AT THE RIGHT INDICATES A CHANGE MADE TO THIS INPUT DESCRIPTION INCLUDED IN CALSPAN REPORT NO. ZQ-5180-L-I ENTITLED "AN IMPROVED THREE DIMENSIONAL COMPUTER SIMULATION OF MOTOR VEHICLE CRASH VICTIMS" (JULY 1974).

OUTLINE OF INPUT TO THE PROGRAM :

CARDS A - DATE AND RUN DESCRIPTION, UNITS OF INPUT AND OUTPUT, CONTROL OF RESTART, INTEGRATOR AND OPTIONAL OUTPUT.

CARDS B - PHYSICAL CHARACTERISTICS OF THE SEGMENTS AND JOINTS.

CARDS C - DESCRIPTION OF THE VEHICLE MOTION.

CARDS D - CONTACT PLANES, BELTS, AIR BAGS, CONTACT ELLIPSOIDS, CONSTRAINTS, AND SYMMETRY OPTIONS.

CARDS E - FUNCTIONS DEFINING FORCE-DEFLECTIONS, INERITIAL SPIKE, ENERGY ABSORPTION FACTOR, AND FRICTION COEFFICIENTS.

CARDS F - ALLOWED CONTACTS AMONG SEGMENTS, PLANES, BELTS, AIR BAGS AND CONTACT ELLIPSOIDS.

CARDS G - INITIAL ORIENTATIONS AND VELOCITIES OF THE SEGMENTS.

CARDS H - CONTROL OF OUTPUT OF TIME HISTORY OF SELECTED SEGMENT MOTIONS AND JOINT PARAMETERS.

A. MAIN PROGRAM INPUT

CARD A.I.A	FORMAT (3A4,2I4,F8.0)
DATE(I),I=1,3	DATE OF THE RUN (12 CHARACTERS).
IRSIN	RESTART INPUT UNIT NO. IF BLANK OR ZERO, ALL INPUT TO BE SUPPLIED ON CARDS A.3 TO CARDS H.7. IF NONZERO (SUGGESTED VALUE =4) INPUT WILL BE SUPPLIED FROM A PREVIOUS RESTART TAPE AND CARDS A.1.B,C AND A.2.
IRSOUT	RESTART OUTPUT UNIT NO. IF NONZERO (SUGGESTED VALUE =3) RECORDS WILL BE WRITTEN ON THIS OUTPUT UNIT FOR FUTURE RESTART RUNS. AN INITIAL RECORD CONTAINING ALL INPUT AND INITIALIZATION DATA WILL BE WRITTEN PLUS A TIME POINT RECORD AT EVERY TIME INTERVAL AS SPECIFIED BY DT ON CARD A.4.
RSTIME	RESTART TIME (SEC.) REQUIRED IF IRSIN # 0. SHOULD BE NONZERO AND AN INTEGER MULTIPLE OF DT ON CARD A.4. PROGRAM WILL READ RECORDS FROM THE PREVIOUS RESTART TAPE UP TO AND INCLUDING THIS TIME, MAKE CHANGES PER CARD A.2 AND CONTINUE OPERATION FROM THERE.
CARDS A.1.B - A.1.C	FORMAT (20A4/20A4)
COMENT(I),I=I,40	DESCRIPTION OF THE RUN (160 CHARACTERS ON TWO CARDS).

THESE CARDS REQUIRED ONLY IF IRSIN > 0 , IN WHICH CASE ALL OTHER INPUT AS SPECIFIED ON CARDS A.3 TO H.7 ARE BYPASSED. TWO SETS OF A.2 (EACH TERMINATED WITH A BLANK CARD) ARE REQUIRED. THE FIRST SET IS PROCESSED AFTER THE INITIAL INPUT RECORD IS READ FROM INPUT UNIT IRSIN AND, IF IRSOUT # 0, BEFORE THE INPUT RECORD IS WRITTEN ON OUTPUT UNIT IRSOUT. THE SECOND SET IS PROCESSED AFTER THE TIME POINT RECORD FOR TIME = RSTIME HAS BEEN READ AND, IF IRSOUT # 0, AFTER THE SAME RECORD IS WRITTEN ON OUTPUT UNIT IRSOUT, BUT BEFORE THE PROGRAM RESUMES OPERATION.

CARDS A.2.A - A.2.N	FORMAT(A8, 4I4, 2(F8.0, I8, A8))
AVAR	ALPHANUMERIC NAME (LEFT ADJUSTED IN FIELD) OF VARIABLE TO BE REDEFINED FOR RESTART. PROGRAM IS CAPABLE OF CHANGING ANY VARIABLE IN THE LABELED COMMON BLOCKS AS USED AFTER ALL INITIALIZATION HAS BEEN PERFORMED. THE USER SHOULD ASCERTAIN THAT CHANGING THIS VARIABLE IS VALID FOR THE PROGRAM.
INDEX(I),I=1,3	THE ARRAY INDICES, IF ANY, OF THE VARIABLE. MUST AGREE IN NUMBER AND THE VALUES MUST BE LESS THAN OR EQUAL TO THE DIMENSIONS OF THE VARIABLE. BLANK OR ZERO FOR NO DIMENSION.
ITYPE	SUPPLY 1,2 OR 3 TO INDICATE THAT THE NEW VALUE IS TO BE REAL(RR), INTEGER(II) OR ALPHANUMERIC(AA). MUST AGREE WITH THE TYPE OF THE VARIABLE WITHIN THE PROGRAM.
RR,II OR AA	NEW VALUE OF THE VARIABLE AVAR TO BE SUPPLIED IN THE APPROPRIATE FIELD DETERMINED BY THE VALUE OF ITYPE.
RROLD,IIOLD OR AAOLD	THE PREVIOUS VALUE OF THE VARIABLE AVAR IN THE APPROPRIATE FIELD ACCORDING TO THE ITYPE VALUE. INTEGER OR ALPHANUMERIC DATA WILL BE TESTED EXACTLY, REAL DATA TO 5 SIGNIFICANT DIGITS. IF THE CURRENT VALUE IS DIFFERENT, THE PROGRAM WILL TERMINATE WITH AN ERROR MESSAGE. IF ZERO OR BLANK IS SUPPLIED,NO CHECK IS PERFORMED.

THESE A.2 CARDS WILL BE PROCESSED UNTIL A BLANK VALUE FOR AVAR IS ENCOUNTERED. NO FURTHER INPUT IS REQUIRED.

CARD A.3 FORMAT (3A4, 3F12.0)

UNITL	UNIT OF LENGTH (4 CHARACTERS)
UNITM	UNIT OF FORCE (MASS) (4 CHARACTERS)
UNITT	UNIT OF TIME (4 CHARACTERS).

NOTE : UNITL, UNITM AND UNITT SHOULD CORRESPOND TO THE USER'S INPUTS. THROUGHOUT THIS DESCRIPTION, INCHES, POUNDS AND SECONDS (IN,LBS,SEC) ARE USED AS SAMPLE UNITS.

GRAVTY(I),I=1,3	THE X, Y AND Z COMPONENTS OF GRAVITY (IN/SEC**2).

CARD A.4 FORMAT (2I4, 4F8.0)

NDINT	NUMBER OF ITERATIONS FOR FINAL CONVERGENCE TEST OF THE INTEGRATOR SUBROUTINE DINT (MINIMUM VALUE = 2, SUGGESTED VALUE = 4).
NSTEPS	NUMBER OF INTEGRATION STEPS (OR OUTPUT TIME POINTS) FOR THE INTEGRATOR ROUTINE. MAY BE ZERO TO OBTAIN INITIAL CONDITIONS.
DT	MAIN PROGRAM TIME INTERVAL FOR INTEGRATOR ROUTINE OUTPUT (SEC). TOTAL TIME OF RUN WILL BE NSTEPS*DT SECONDS WITH MAIN PROGRAM TAPE 1, PRINTER PLOT AND OPTIONAL OUTPUT PRODUCED EVERY DT SECONDS.
HO	INITIAL INTEGRATOR STEP SIZE (SEC).
HMAX	MAXIMUM INTEGRATOR STEP SIZE (SEC). FOR BEST EFFICIENCY DT SHOULD BE AN INTEGRAL MULTIPLE OF HMAX AND HMAX A POWER OF TWO MULTIPLE OF HO. (SUGGESTED VALUE = 0.001 SEC.)
HMIN	MINIMUM INTEGRATOR STEP SIZE (SEC). IF A FIXED STEP SIZE IS DESIRED, SET HMIN GREATER THAN HMAX, AND STEP SIZE WILL DOUBLE FROM HO UNTIL HMAX IS ACHIEVED.

CARD A.5 FORMAT (40I2)

NPRT(I),I=1,40 — AN ARRAY OF INDICATORS THAT CONTROL VARIOUS OPTIONAL DIAGNOSTIC OUTPUT FOR THE PROGRAM. A VALUE OF ZERO OR BLANK INDICATES NO OUTPUT FOR THAT PARTICULAR ITEM. IN GENERAL, A VALUE OF 1 MEANS THAT THE OUTPUT WILL BE PRODUCED EVERY TIME A PARTICULAR SUBROUTINE IS EXECUTED. HOWEVER, FOR ELEMENTS 1-6 THE VALUE INDICATES THE PRINT FREQUENCY, E.G., A VALUE OF 5 WILL PRODUCE OUTPUT EVERY 5TH EXECUTION OF THE SUBROUTINE. OUTPUT PRODUCED BY ELEMENTS 7-26 IS INTENDED FOR DIAGNOSTIC OR "CHECK OUT" PURPOSES AND IS NOT COMPLETELY LABELED. THE USER SHOULD CONSULT THE LISTING OF THE SUBROUTINE FOR A DESCRIPTION OF THE VARIABLES THAT ARE PRINTED.

THE NPRT ARRAY

ELEMENT NO.	SUBROUTINE	OUTPUT PRODUCED
1	MAIN	TAPE 1 OUTPUT
2	MAIN	ELTIME OUTPUT
3	MAIN	SUBROUTINE PRINT OUTPUT
4	NOT USED	
5	PRIPLT	Y-Z VIEW PRINTER PLOT
6	PRIPLT	X-Z VIEW PRINTER PLOT
7	BINPUT	HA, HB
8*	DAUX	IJK,RHS,C ARRAYS
9	DAUX	SUBROUTINE PRINT OUTPUT
10	IMPULS	DIAGNOSTIC OUTPUT
11	SETUPI	U2,VI ARRAYS
12	VISPR	DIAGNOSTIC OUTPUT
13	PRIPLT	CJOINT ARRAY
14	WINDY	WIND FORCES
15	BELTG	DIAGNOSTIC OUTPUT
16	HBELT	HARNESS-BELT FORCES
17	EDEPTH	DIAGNOSTIC OUTPUT
18	NOT USED	
19	NOT USED	
20	CHAIN	XCOMP,XVCOMP,SEGLP,SEGLV
21	AIRBAG	DIAGNOSTIC OUTPUT
22	AIRBG1	DIAGNOSTIC OUTPUT
23	AIRBG3	DIAGNOSTIC OUTPUT
24	UPDATE	ROLL-SLIDE TEST OUTPUT
25	DINT	CONVERGENCE TEST DATA
26	DINT	SUBROUTINE OUTPUT EVERY STEP

* A VALUE OF NPRT(8) = 2 WILL PRINT ARRAYS BEFORE AND AFTER THE FIRST CALL TO SUBROUTINE FSMSOL ONLY.

B. SUBROUTINE BINPUT

CARD B.1	FORMAT (2I6, 8X, 5A4)
NSEG	THE NUMBER OF SEGMENTS (MAXIMUM = 20). NOTE: THE VEHICLE AND GROUND WILL BE ASSIGNED SEGMENT NUMBERS NSEG+1 AND NSEG+2.
NJNT	THE NUMBER OF JOINTS (MAXIMUM = 21). NOTE: NORMALLY NJNT = NSEG-1, BUT JOINT NUMBERS NSEG AND NSEG+1 MAY BE USED TO CONNECT THE VEHICLE AND THE GROUND TO LOWER NUMBERED SEGMENTS.
BDYTTL(I),I=1,5	DESCRIPTION OF THE CRASH VICTIM (20 CHARACTERS).

CARDS B.2.A - B.2.I (NSEG CARDS)	FORMAT (A4, 1X, A1, 10F6.0)

EACH CARD (I) FOR I = 1, NSEG WILL CONTAIN INPUT DATA FOR THE ITH SEGMENT. THE SEGMENT IDENTIFYING NUMBERS (I) WILL BE REFERRED TO ON LATER INPUT CARDS.

SEG(I)	AN ABBREVIATION OF THE NOMENCLATURE OF THE ITH SEGMENT (4 CHARACTERS).
CGS(I)	THE PLOT SYMBOL OF THE SEGMENT C.G. (1 CHARACTER).
W(I)	THE WEIGHT OF THE SEGMENT (LBS).
PHI(J,I),J=1,3	THE PRINCIPAL MOMENTS OF INERTIA OF THE SEGMENT ABOUT THE X, Y, AND Z AXES OF THE SEGMENT (LBS-SEC**2-IN). THERE ARE NO RESTRICTIONS ON THE VALUES OF W(I) OR PHI(J,I), THEY MAY BE NEGATIVE OR ZERO. IF ANY COMPONENT IS ZERO, IT IS ASSUMED THAT THE SYSTEM IS SUITABLY CON-STRAINED SO THAT THE SYSTEM MATRIX IS NON-SINGULAR.
BD(J,I),J=1,3	THE X, Y, AND Z SEMIAXES OF THE SEGMENT CONTACT ELLIPSOID (IN).
BD(J,I),J=4,6	THE LOCATION OF THE CENTER OF THE SEGMENT CONTACT ELLIPSOID, WITH RESPECT TO THE CENTER OF GRAVITY OF THE SEGMENT, IN THE LOCAL BODY SEGMENT REFERENCE(IN). THESE PRIMARY CONTACT ELLIPSOIDS ARE GIVEN THE SAME INDENTIFYING NUMBER AS THE SEGMENT. THEY MAY BE REDEFINED WITH AN ARBITRARY ORIENTATION ON CARDS D.5.

CARDS B.3.A - B.3.J   FORMAT (A4, 1X, A1, 2I4, 6F6.0/ 14X, 6F6.0)
(2*NJNT CARDS - 2 CARDS FOR EACH JOINT)

EACH CARD (J) FOR J = 1, NJNT WILL CONTAIN INPUT DATA FOR THE JTH JOINT. THE JOINT IDENTIFYING NUMBERS (J) WILL BE REFERRED TO ON LATER INPUT CARDS.

JOINT(J) — AN ABBREVIATION OF THE NOMENCLATURE OF THE JTH JOINT (4 CHARACTERS).

JS(J) — PLOT SYMBOL OF THE JOINT LOCATION (1 CHARACTER).

JNT(J) — MAGNITUDE INDICATES THE NUMBER OF THE SEGMENT THAT IS CONNECTED TO SEGMENT J+1 BY JOINT J. IF NEGATIVE, JOINT J IS ASSOCIATED WITH A FLEXIBLE ELEMENT. IF ZERO, SEGMENT J+I IS THE REFERENCE SEGMENT OF ANOTHER BODY. (|JNT(J)| < J+1).

IPIN(J) — 0 - THERE ARE TO BE NO CONSTRAINTS ON JOINT J.
1 - JOINT J IS PINNED (HINGE).
2 - JOINT J IS NOT PINNED (BALL AND SOCKET).
4 - JOINT J IS AN EULER JOINT.
NON-ZERO VALUES FOR IPIN MAY BE SUPPLIED AS POSITIVE OR NEGATIVE TO INDICATE THAT THE INITIAL CONDITION OF THE JOINT IS UNLOCKED (POSITIVE) OR UNLOCKED (NEGATIVE).
THE INITIAL STATE OF AN EULER JOINT IS SET BY USE OF IPIN AS FOLLOWS

IPIN	IEULER	STATE	
4	8	FREE	
- 4	7	ALL AXES LOCKED	
- 5	6	SPIN	FREE, OTHERS LOCKED
- 6	5	NUTATION	FREE, OTHERS LOCKED
- 7	4	PRECESSION	FREE, OTHERS LOCKED
- 8	3	SPIN	LOCKED, OTHERS FREE
- 9	2	NUTATION	LOCKED, OTHERS FREE
-10	1	PRECESSION	LOCKED, OTHERS FREE

(PRECESSION ABOUT Z
NUTATION ABOUT RESULTANT X
SPIN ABOUT RESULTANT Z)
IF IPIN IS LESS THAN -3 PROGRAM WILL SET IEULER AS ABOVE AND THEN SET IPIN = -4.

SR(I,2*J-1),I=1,3 — COORDINATES OF LOCATION OF JOINT J (IN.) IN THE LOCAL REFERENCE SYSTEM OF SEGMENT JNT(J).

SR(I,2*J),I=1,3 — COORDINATES OF LOCATION OF JOINT J (IN.) IN THE LOCAL REFERENCE SYSTEM OF SEGMENT J+1.

FOLLOWING DATA IS ON 2ND CARD FOR EACH JOINT.

YPR1(I,J),I=1,3	THE YAW, PITCH AND ROLL ANGLES (DEGREES) SPECIFYING THE PRINCIPAL AXES OF JOINT J IN THE LOCAL REFERENCE SYSTEM OF SEGMENT JNT(J). YAW ABOUT Z AXIS PITCH ABOUT RESULTANT Y AXIS ROLL ABOUT RESULTANT X AXIS
YPR2(I,J),I=1,3	THE YAW, PITCH AND ROLL ANGLES (DEGREES) SPECIFYING THE PRINCIPAL AXES OF JOINT J IN THE LOCAL REFERENCE SYSTEM OF SEGMENT J+1. THE Z AXIS IS THE REFERENCE AXIS TO DEFINE FLEXURE. THE Y AXIS IS USED AS THE PIN AXIS EXCEPT FOR THE SPECIAL EULER JOINTS. THE XY PLANE IS USED FOR GLOBALGRAPHIC JOINTS WITH X AS THE REFERENCE AXIS.

CARDS B.4.A - B.4.J   FORMAT (2 (4F6.0, F12.0))

(NJNT SETS OF CARDS, ONE FOR EACH JOINT J. IF |IPIN(J)| # 4, EACH SET READS VALUES FOR 3*J-2 AND 3*J-1 ON ONE CARD ONLY. IF |IPIN(J)| = 4 , JOINT J IS AN EULER JOINT AND A SECOND CARD IS NECESSARY TO READ VALUES FOR 3*J)

SPRING(I,3*J-2), I=1,5 — THE FLEXURAL SPRING CHARACTERISTICS FOR JOINT J. IF J IS AN EULER JOINT, THE SPRING CHARACTERISTICS ABOUT THE PRECESSION AXIS. IF JOINTF(J) # 0 (ON CARD F.5.A), THESE VALUES ARE NOT USED AND SHOULD BE ZERO.

SPRING(I,3*J-1), I=I,5 — THE TORSIONAL SPRING CHARACTERISTICS FOR JOINT J. IF J IS AN EULER JOINT, THE SPRING CHARACTERISTICS ABOUT THE NUTATION AXIS.

SPRING(I,3*J), I=1,5 — SECOND CARD OF EACH SET IS REQUIRED ONLY IF J IS AN EULER JOINT, THE SPRING CHARACTERISTICS ABOUT THE SPIN AXIS.

I=1 — LINEAR SPRING COEFFICIENT (IN-LBS/DEG).

I=2 — QUADRATIC SPRING COEFFICIENT (IN-LBS/DEG**2).

I=3 — CUBIC SPRING COEFFICIENT (IN-LBS/DEG**3).

I=4 — ENERGY DISSIPATION COEFFICIENT (DIMENSIONLESS).
A VALUE OF 1. SPECIFIES NO LOSS
A VALUE OF 0. SPECIFIES MAXIMUM LOSS

I=5 — JOINT STOP LOCATION WITH RESPECT TO THE CENTER OF SYMMETRY (DEG).
FOR A VALUE OF ZERO THE ROUTINE WILL USE ONLY THE LINEAR SPRING COEFFICIENT AND WILL APPLY THE ENERGY DISSIPATION FACTOR

CARDS B.5.A - B.5.J FORMAT(5F6.0, 18X, 2F6.0)

(NJNT SETS OF CARDS, ONE FOR EACH JOINT J. IF |IPIN(J)| # 4, VALUES FOR 3*J-2 ARE ON ONE CARD ONLY. IF |IPIN(J)| = 4, J IS AN EULER JOINT AND VALUES FOR 3*J-1 AND 3*J ARE REQUIRED ON A SECOND AND THIRD CARD OF EACH SET.)

VISC(I,3*J-2), I=1,7 — THE VISCOUS CHARACTERISTICS FOR JOINT J. IF J IS AN EULER JOINT, THE VISCOUS CHARACTERISTICS ABOUT THE PRECESSION AXIS.

VISC(I,3*J-1), I=1,7 — THE SECOND CARD OF EACH SET IS REQUIRED ONLY IF J IS AN EULER JOINT, THE VISCOUS CHARACTERISTICS ABOUT THE NUTATION AXIS.

VISC(I,3*J) I=1,7 — THE THIRD CARD OF EACH SET IS REQUIRED ONLY IF J IS AN EULER JOINT, THE VISCOUS CHARACTERISTICS ABOUT THE SPIN AXIS.

I=1 VISCOUS COEFFICIENT (IN-LB-SEC/DEG).

I=2 COULOMB FRICTION COEFFICIENT (IN-LB).

I=3 RELATIVE ANGULAR VELOCITY OF JOINT AT WHICH FULL COULOMB FRICTION IS APPLIED (DEG/SEC). MUST BE GREATER THAN 0.

I=4 T1: THE MAXIMUM TORQUE (IN-LBS) ALLOWED FOR A LOCKED JOINT (OR EULER AXIS). IF EXCEEDED,THE JOINT WILL UNLOCK. IF T1 = 0, THE TEST WILL NOT BE PERFORMED.

I=5 T2: THE MINIMUM TORQUE (IN-LBS) ALLOWED FOR JOINT J TO REMAIN UNLOCKED. IF T2 = 0, THE TEST WILL NOT BE PERFORMED.

I=6 T3: THE MINIMUM ANGULAR VELOCITY (RAD/SEC) NECESSARY FOR JOINT J TO REMAIN UNLOCKED. IF T3 = 0, THE TEST WILL NOT BE PERFORMED.

I=7 E = (1+U)/2 WHERE U IS THE CLASSICAL COEFFICIENT OF RESTITUTION TO BE USED FOR THE IMPULSE OPTION IF THE JOINT HITS THE JOINT STOP (0 < E < 1 OR -1 < U < +1). A VALUE OF E = 0 MEANS THAT THE IMPULSE OPTION WILL NOT BE EXERCISED FOR THIS JOINT.

CARDS B.6.A - B.6.I (NSEG CARDS)	FORMAT (12F6.0)
SGTEST(1,1,I)	MAGNITUDE TEST FOR THE ANGULAR VELOCITY OF SEGMENT NO. I (RAD/SEC).
SGTEST(2,1,I)	ABSOLUTE ERROR TEST FOR THE ANGULAR VELOCITY OF SEGMENT NO. I (RAD/SEC).
SGTEST(3,1,I)	RELATIVE ERROR TEST FOR THE ANGULAR VELOCITY OF SEGMENT NO. I (DIMENSIONLESS).
SGTEST(1,2,I) (2,2,I) (3,2,I)	SAME AS ABOVE, BUT FOR THE LINEAR VELOCITY OF SEGMENT NO. I (IN/SEC).
SGTEST(1,3,I) (2,3,I) (3,3,I)	SAME AS ABOVE, BUT FOR THE ANGULAR ACCELERATION OF SEGMENT NO. I (RAD/SEC**2).
SGTEST(1,4,I) (2,4,I) (3,4,I)	SAME AS ABOVE BUT FOR THE LINEAR ACCELERATION OF SEGMENT NO. 1 (IN/SEC**2).

THESE CONVERGENCE TESTS ARE PERFORMED IN SUBROUTINE DINT ON THE RESULTANT OF THE DERIVATIVE VECTORS. THE LINEAR VELOCITIES AND ACCELERATIONS ARE COMPUTED ONLY FOR REFERENCE SEGMENTS (I.E. SEGMENT NO. 1 AND THOSE SEGMENTS I WHERE JNT(I-1) = 0), THEREFORE ANY TEST NUMBERS SUPPLIED FOR LINEAR VELOCITIES AND ACCELERATIONS OF OTHER SEGMENTS WILL BE IGNORED. THE TESTS FOR CONVERGENCE ARE PERFORMED IN THE FOLLOWING ORDER :

1) IF THE MAGNITUDE TEST IS ZERO, NO TESTING IS DONE FOR THAT VARIABLE.

2) IF THE MAGNITUDE OF THE RESULTANT VECTOR IS LESS THAN THE MAGNITUDE TEST,THE ROUTINE HAS PASSED THE CONVERGENCE TEST FOR THAT VARIABLE.

3) IF THE ABSOLUTE ERROR TEST IS GREATER THAN ZERO, AND THE MAGNITUDE OF THE ABSOLUTE ERROR (DIFFERENCE BETWEEN THE PREDICTED AND COMPUTED VECTOR) IS LESS THAN THE ABSOLUTE ERROR TEST, THE ROUTINE HAS PASSED THE CONVERGENCE TEST FOR THAT VARIABLE.

4) IF THE RELATIVE ERROR OF THE MAGNITUDE OF THE ABSOLUTE ERROR COMPARED TO THE MAGNITUDE OF THE COMPUTED VECTOR IS GREATER THAN THE RELATIVE ERROR TEST, THE CONVERGENCE TEST HAS FAILED.

IF NFLX # 0, CARDS B.7 ARE REQUIRED. EACH FLEXIBLE ELEMENT AS DEFINED ON CARDS B.3 CONTAINS AT LEAST THREE CONNECTED SEGMENTS CONSISTING OF A REFERENCE SEGMENT, ONE OR MORE INTERIOR SEGMENTS AND A TERMINATING SEGMENT. EACH JOINT IN THE ELEMENT SHOULD HAVE A NEGATIVE VALUE FOR JNT, AND THE NUMBER OF INTERIOR SEGMENTS WILL BE ONE LESS THAN THE NUMBER OF NEGATIVE VALUES OF JNT FOR EACH ELEMENT. NFLX IS THE TOTAL NUMBER OF INTERIOR SEGMENTS OF ALL FLEXIBLE ELEMENTS.

CARD B.7.A FORMAT (18I4)

NFX	THE NUMBER OF INTERIOR SEGMENTS FOR WHICH HF ARRAYS ARE TO BE SUPPLIED.
KNT(K),K=1,NFX	THE INTERIOR SEGMENT IDENTIFICATION NUMBERS IN THE ORDER OF THE HF ARRAYS TO BE SUPPLIED. IF THE VALUES OF NFX AND KNT ARE NOT CONSISTENT WITH THE NEGATIVE VALUES OF JNT ON CARDS B.3 THE PROGRAM WILL TERMINATE WITH AN APPROPRIATE ERROR MESSAGE.

CARDS B.7.B - B.7.N FORMAT (12F6.0)
(4*NFX CARDS, 4 CARDS FOR EACH SEGMENT IN THE ORDER AS THEY ARE DEFINED IN THE KNT VECTOR.)

(HF(I,J,K),J=1,12) ,I=1,4	THE COEFFICIENTS OF THE QUADRATIC FORM FUNCTION USED TO DEFINE THE ORIENTATION OF INTERIOR SEGMENT KNT(K) WITH RESPECT TO REFERENCE SEGMENT OF THE ELEMENT.

FORM THE COLUMN VECTOR V WITH FOUR COMPONENTS Y,P,R AND 1, WHERE Y,P,R ARE THE YAW, PITCH AND ROLL OF THE TERMINATING SEGMENT RELATIVE TO THE REFERENCE SEGMENT. LET H BE A SYMMETRIC 4X4 MATRIX SUCH THAT F(V) = 1/2 V.HV REPRESENTS A QUADRATIC SCALAR FUNCTION OF THE VARIABLES Y,P AND R IN RADIANS. THUS

```
  YAW OF SEGMENT KNT(K) = I/2 V.HF(I,J  ,K)V
PITCH OF SEGMENT KNT(K) = I/2 V.HF(I,J+4,K)V
 ROLL OF SEGMENT KNT(K) = I/2 V.HF(I,J+8,K)V      (I,J=I,4)
```

C. SUBROUTINE VINPUT

CARD C.1	FORMAT (20A4)
VPSTTL(I),I=1,20	DESCRIPTION OF THE CRASH VEHICLE DECELERATION (80 CHARACTERS).
CARD C.2	FORMAT (8F6.0, I6, 2F6.0)
ANGLE(I),I=1,3	FOR THE HALF SINE-WAVE DECELERATION (NATAB = 0) OR FOR THE UNIDIRECTIONAL DECELERATION TABULAR INPUT (NATAB > 0), ANGLE(1) AND ANGLE(2) REPRESENT THE AZIMUTH AND ELEVATION (OBLIQUE ANGLES) OF THE DIRECTION OF THE DECELERATION IMPULSE (DEG). ANGLE(3) IS NOT USED AND THE INITIAL YAW, PITCH AND ROLL OF THE VEHICLE ARE ASSUMED TO BE ZERO. FOR THE OMNIDIRECTIONAL TABULAR INPUT (NATAB < 0). THEY REPRESENT THE INITIAL YAW, PITCH AND ROLL OF THE VEHICLE (DEG).
VIPS	THE INITIAL VELOCITY OF THE CRASH VEHICLE. (IN/SEC - UNITS AS SPECIFIED ON CARD A.3) A NEGATIVE VALUE MAY BE SUPPLIED FOR NATAB=0 TO INDICATE THAT THE VEHICLE WILL ACCELERATE FROM A VELOCITY OF ZERO TO |VIPS|.
VTIME	THE TIME DURATION OF THE DECELERATION IMPULSE (SEC). REQUIRED ONLY IF NATAB = 0. A VALUE OF ZERO IS NOT PERMITTED IF NATAB=0.
XO(I),I=1,3	THE INITIAL X, Y, AND Z COORDINATES OF THE VEHICLE REFERENCE ORIGIN IN INERTIAL REFERENCE (IN).
NATAB	INTEGER NUMBER OF TIME POINTS FOR WHICH VEHICLE DECELERATION DATA IS TO BE SUPPLIED. THE ALGEBRAIC SIGN OF NATAB DETERMINES THE TYPE OF VEHICLE MOTION AS FOLLOWS: IF NATAB = 0, THE DIRECTION IMPULSE IS AN ANALYTICAL HALF-SINE WAVE FUNCTION THAT DECELERATES THE VEHICLE FROM AN INITIAL SPEED OF VMPH TO ZERO IN VTIME SECONDS. IF NATAB > 0, THE VEHICLE MOTION IS UNIDIRECTIONAL AND ONLY THE RESULTANT LINEAR DECELERATION IS INPUTTED IN TABULAR FORM ON CARDS C.3. (NATAB SHOULD BE ODD AND MAXIMUM VALUE IS 99.) IF NATAB < 0, THE VEHICLE MOTION IS ALSO ROTATIONAL, AND THE COMPONENTS OF BOTH LINEAR AND ANGULAR ACCELERATION ARE INPUTTED IN TABULAR FORM ON CARDS C.4. (MINIMUM VALUE OF NATAB IS -100.)

ATO — THE BEGINNING TIME POINT FOR THE DECELERATION TABLE INPUT (SEC).

ADT — FIXED TIME INTERVAL FOR THE DECELERATION TABLE INPUT (SEC).

CARDS C.3.A - C.3.N — FORMAT (12F6.0)

THESE CARDS ARE REQUIRED ONLY IF NATAB > 0.

ATAB(1,I),I=1,NATAB — THE NATAB VALUES OF DECELERATION (G'S) FOR THE CRASH VEHICLE FOR FIXED TIME INTERVALS

T(I) = ATO + (I-1)*ADT FOR I=1,NATAB.

SUPPLY 12 VALUES PER CARD, USE AS MANY CARDS AS NECESSARY. SINCE A SIMPSON'S INTEGRATION IS USED TO COMPUTE VELOCITY AND POSITION, THE VALUE OF NATAB MUST BE ODD. THE LAST VALUE, ATAB(1,NATAB) WILL BE USED TO INTEGRATE FOR ANY TIME GREATER THAN T(NATAB-1).

CARDS C.4.A - C.4.M — FORMAT (10X, 6F10.0)

MATAB CARDS ARE REQUIRED ONLY IF NATAB < 0 (MATAB = -NATAB)

EACH CARD (I) WILL CONTAIN THE LINEAR AND ANGULAR ACCELERATIONS FOR TIME T(I) = ATO + (I-1)*ADT FOR I = 1,MATAB.

ATAB(J,I),J=1,3 — THE VALUES OF THE X,Y AND Z COMPONENTS OF LINEAR DECELERATION (G'S) FOR TIME POINT T(I). THE PROGRAM WILL INTEGRATE FOR VELOCITY AND POSITION BEYOND THE LAST TIME POINT USING THE LAST VALUES SUPPLIED.

ATAB(J,I),J=4,6 — THE VALUES OF THE COMPONENTS OF ANGULAR ACCELERATION (DEG/SEC**2) FOR TIME POINT(I). THE VALUES FOR THE LAST TIME POINT MUST BE ZERO WHICH IS ASSUMED BY PROGRAM FOR INTEGRATING BEYOND THE LAST TIME POINT.

D. SUBROUTINE SINPUT

CARD D.1	FORMAT (6I6)
NPL	THE NUMBER OF PLANES DESCRIBING A CONTACT PANEL OF THE VEHICLE (20 MAXIMUM).
NBLT	THE NUMBER OF BELTS USED TO RESTRAIN THE CRASH VICTIM (8 MAXIMUM).
NBAG	THE NUMBER OF AIR BAGS USED TO RESTRAIN THE CRASH VICTIM (5 MAXIMUM).
NELP	THE NUMBER OF CONTACT ELLIPSOIDS TO BE SUPPLIED ON CARDS D.5.
NQ	THE NUMBER OF CONSTRAINTS TO BE SUPPLIED ON CARDS D.6. EACH CONSTRAINT TYPE 5 WILL BE CONSIDERED AS TWO CONSTRAINTS REQUIRING TWO SETS OF CARDS (NOTE: THE PROGRAM WILL LATER INCREMENT NQ BY I FOR EACH NF(1) = 0 ON CARDS F.I.B AND F.3.B AND THE FINAL MAXIMUM ON NQ IS 12).
NSD	THE NUMBER OF SPRING DAMPERS TO BE SUPPLIED ON CARDS D.8 (20 MAXIMUM).

IF NPL # 0, NPL SETS OF D.2 ARE REQUIRED.

CARD D.2.A	FORMAT (I4, 4X, 5A4)
J	THE NUMBER IDENTIFYING THE PLANE, MUST BE INPUTTED AS SUCCESSIVE INTEGERS 1, 2, 3,..., NPL.
PLTTL(I,J),I=1,5	A 20 CHARACTER DESCRIPTION OF THE JTH PANEL.
CARDS D.2.B - D.2.D	FORMAT (3F12.0)
P1(I),I=1,3	THE X,Y AND Z COORDINATES OF POINT P1 IN VEHICLE (OR SEGMENT TO WHICH PLANE IS ATTACHED) REFERENCE (IN).
P2(I),I=1,3	THE X,Y AND Z COORDINATES OF POINT P2 IN VEHICLE (OR SEGMENT TO WHICH PLANE IS ATTACHED) REFERENCE (IN).
P3(I),I=1,3	THE X,Y AND Z COORDINATES OF POINT P3 IN VEHICLE (OR SEGMENT TO WHICH PLANE IS ATTACHED) REFERENCE (IN).

WHERE P1, P2, AND P3 ARE 3 OF THE CORNERS OF A BOUNDED RECTANGULAR PLANE SUCH THAT THE EDGE P1P2 IS 90 DEGREES CLOCKWISE (AS VIEWED FROM THE EXTERNAL SURFACE) FROM THE EDGE P1P3.

IF NBLT # 0, NBLT SETS OF D.3 ARE REQUIRED.

CARD D.3.A	FORMAT (5A4)
BLTTTL(I,J),I=1,5	A 20 CHARACTER DESCRIPTION OF THE JTH BELT.

CARD D.3.B	FORMAT (6F12.0)
BELT(I,J),I=1,3	X,Y, AND Z COORDINATES, IN VEHICLE (OR SEGMENT TO WHICH BELT IS ANCHORED) REFERENCE, OF ANCHOR POINT A FOR THE JTH BELT (IN).
BELT(I,J),I=4,6	X,Y, AND Z COORDINATES, IN VEHICLE (OR SEGMENT TO WHICH BELT IS ANCHORED) REFERENCE, OF ANCHOR POINT B FOR THE JTH BELT (IN).

NOTE: THE PROGRAM MUST PASS A PLANE THROUGH THE THREE POINTS, ANCHOR POINT A, ANCHOR POINT B, AND A FIXED POINT ON THE CONTACTED BODY SEGMENT. IF ANCHOR POINTS A AND B COINCIDE, THEY MUST BE SEPARATED SLIGHTLY FOR INPUT SUCH THAT THE DESIRED BELT PLANE WILL BE DEFINED.

CARD D.3.C	FORMAT (5F12.0)
BELT(I,J),I=7,9	X, Y, AND Z COORDINATES, IN LOCAL BODY SEGMENT REFERENCE (BUT WITH RESPECT TO ELLIPSOID CENTER, NOT C.G.), OF THE FIXED CONTACT POINT ON THE BODY SEGMENT FOR THE JTH BELT (IN).
BELT(10,J)	CURRENTLY NOT USED BY THE PROGRAM.
BELT(11,J)	BELT SLACK (IN). THE SLACK, WHEN ADDED TO THE INITIAL GEOMETRIC LENGTH, RESULTS IN THE INITIAL BELT LENGTH. IF DESIRED, THE INITIAL BELT LENGTH MAY BE INPUTTED AS A NEGATIVE NUMBER AND THE PROGRAM WILL COMPUTE THE SLACK.

IF NBAG # 0, NBAG SETS OF D.4 ARE REQUIRED BY SUBROUTINE AIRBGI.

CARD D.4.A	FORMAT (5A4, I4)
BAGTTL(I,J),I=I,5	A 20 CHARACTER DESCRIPTION OF THE JTH AIR BAG.
NPANEL(J)	NUMBER OF VEHICLE CONTACT PANELS THAT ARE ALLOWED TO INTERACT WITH THE JTH AIR BAG (MAXIMUM = 4).
CARD D.4.B	FORMAT(6F12.0)
AB(I,J),I=I,3	THE X, Y AND Z SEMIAXES OF THE JTH AIR BAG WHEN FULLY INFLATED AND UNDEFORMED (IN).
BFA(I,J),I=1,3	THE X,Y AND Z COORDINATES OF THE CENTER OF THE AIR BAG CONTACT ELLIPSOID WITH RESPECT TO THE AIR BAG CENTER OF GRAVITY (IN).
CARD D.4.C	FORMAT (6F12.0)
YB,PB,RB	THE INITIAL ORIENTATION (YAW, PITCH, AND ROLL) OF THE JTH AIR BAG IN THE VEHICLE REFERENCE (DEG).
ZDEP(I,J),I=1,3	THE X, Y, AND Z COORDINATES OF THE DEPLOYMENT POINT OF THE JTH AIR BAG IN THE LOCAL REFERENCE OF THE 1ST PANEL ON CARD D.4.G (IN).
CARD D.4.D	FORMAT (6F12.0)
XBM(J)	WEIGHT OF AIR BAG MEMBRANE AND CONTENTS (LBS).
CYTD(J)	GAS SUPPLY ACTUATOR FIRING TIME AFTER THE START OF VEHICLE DECELERATION (SEC).
CYPA(J)	ATMOSPHERIC PRESSURE (PSIA).
CYSP(J)	INITIAL GAS SUPPLY PRESSURE (PSIG).
CYTO(J)	INITIAL GAS SUPPLY TEMPERATURE (DEG R).
CYVO(J)	GAS SUPPLY RESERVOIR VOLUME (IN**3).

CARD D.4.E	FORMAT (6F12.0)
CYCD(J)	SONIC THROAT DISCHARGE COEFFICIENT (DIMENSIONLESS).
CYK(J)	RATIO OF SPECIFIC HEATS OF SUPPLY GAS (DIMENSIONLESS).
CYR(J)	SPECIFIC GAS CONSTANT (IN/DEG R).
CYAT(J)	SONIC THROAT AREA (IN**2).
CYPV(J)	VENT PRESSURE OF THE EXHAUST ORIFICE (PSIG).
CYCDO(J)	EXHAUST ORIFICE DISCHARGE COEFFICIENT (DIMENSIONLESS).

CARD D.4.F	FORMAT (5F12.0)
CYAO(J)	EXHAUST ORIFICE AREA (IN**2).
SPRK(J)	SPRING CONSTANT OF A LINEAR SPRING USED TO SIMULATE ATTACHMENT OF THE BAG AT THE DEPLOYMENT POINT IN THE VEHICLE (LB/IN).
VSCS(J)	COEFFICIENT OF SLIDING FRICTION OF THE AIR BAG (DIMENSIONLESS)
CK(J)	PARAMETER USED TO STABILIZE AIR BAG NUMERICAL INTEGRATION (SEC**-1). SUGGESTED VALUE = 250.
CMASS(J)	MULTIPLIER TO INCREASE OR DECREASE THE MASS OF THE AIR BAG TO ARTIFICIALLY DAMPEN THE INTEGRATED AIR BAG MOTION.

NPANEL(J) SETS OF THE FOLLOWING TWO CARDS ARE REQUIRED TO DEFINE THE ELLIPSOIDS USED TO APPROXIMATE THE CONTACT PANELS FOR THE JTH AIR BAG. THE FIRST PANEL IS THE REACTION PANEL.

CARD D.4.G	FORMAT (6F12.0)
B(I,K,J),I=1,3	X, Y, AND Z SEMIAXES FOR THE KTH PANEL FOR THE JTH AIR BAG (IN).
BFB(I,K,J),I=1,3	THE LOCATION OF THE CENTER OF THE PANEL ELLIPSOID WITH RESPECT TO ITS CENTER OF GRAVITY (IN).

CARD D.4.H	FORMAT (6F12.0)
ZR(I,K,J),I=1,3	X, Y, AND Z COORDINATES IN VEHICLE REFERENCE OF THE CENTER OF GRAVITY OF THE KTH PANEL OF THE JTH AIR BAG (IN).
YP,PP,RP	THE ORIENTATION, YAW, PITCH, AND ROLL OF THE KTH PANEL (DEG).

IF NELP # 0, NELP D.5 CARDS ARE REQUIRED BY SUBROUTINE SINPUT.

NOTE: NELP IS THE NUMBER OF CONTACT ELLIPSOIDS TO BE SUPPLIED HERE, NOT THE NUMBER OF CONTACT ELLIPSOIDS IN THE PROGRAM. THE FIRST NSEG ELLIPSOIDS WERE SUPPLIED ON CARDS B.2.A - B.2.I WITH NO ANGULAR ROTATIONS. THEY MAY BE REPLACED HERE IF DESIRED.

CARDS D.5.A - D.5.J (NELP CARDS)	FORMAT (I6, 9F6.0)
M	CONTACT ELLIPSOID NUMBER. MAX = 24. IF M < NSEG + 1, DATA WILL REPLACE INPUT SUPPLIED ON CARDS B.2.A - B.2.I.
P1(I),I=1,3	THE X, Y, AND Z SEMIAXES OF THE CONTACT ELLIPSOID (IN).
P2(I),I=1,3	THE X, Y, AND Z COORDINATES OF THE ELLIPSOID OFFSET FROM THE SEGMENT CENTER OF GRAVITY.
P3(I),I=1,3	THE YAW, PITCH AND ROLL (DEGREES) OF THE CONTACT ELLIPSOID FROM THE PRINCIPAL AXIS OF THE SEGMENT.

IF NQ # 0, NQ D.6 CARDS ARE REQUIRED BY SUBROUTINE SINPUT.

CARDS D.6.A - D.6.J (NQ CARDS)	FORMAT (3I6, 6F6.0)
KQTYPE(J)	TYPE NO. OF THE JTH CONSTRAINT 1: POINT SPECIFIED BY RK1 ON SEGMENT KQ1 WILL BE CONSTRAINED TO BE THE SAME AS THE POINT SPECIFIED BY RK2 ON SEGMENT KQ2. 2: POINT SPECIFIED BY RK1 ON SEGMENT KQ1 WILL BE CONSTRAINED TO REMAIN AT AN EQUAL DISTANCE (D > 0) FROM THE POINT SPECIFIED BY RK2 ON SEGMENT KQ2. 5: TENSION ELEMENT CONSTRAINT CONNECTING POINT RK1 ON SEGMENT KQ1 TO POINT RK2 ON SEGMENT RK2 (REQUIRES TWO CARDS WITH KQTYPE, KQ1 AND KQ2 THE SAME ON BOTH).
KQ1(J)	SEGMENT IDENTIFICATION NUMBER OF THE 1ST SPECIFIED POINT.
KQ2(J)	SEGMENT IDENTIFICATION NUMBER OF THE 2ND SPECIFIED POINT.
RK1(I,J),I=I,3	COORDINATES OF SPECIFIED POINT ON SEGMENT KQ1 (IN). IF KQTYPE = 5, THE SECOND CARD WILL CONTAIN THE EFFECTIVE MASSES MA, MB AND MAB (LB.SEC**2/IN) IN PLACE OF RKI.
RK2(I,J),I=1,3	COORDINATES OF SPECIFIED POINT ON SEGMENT KQ2 (IN). IF KQTYPE = 5, THE SECOND CARD WILL CONTAIN THE SPRING CONSTANT K (LB/IN), THE VISCOUS DAMPING CONSTANT D (LB SEC/IN) AND THE REFERENCE LENGTH L (IN) IN PLACE OF RK2.

CARD D.7 IS ALWAYS REQUIRED. SUPPLY BLANK CARD FOR NORMAL 3D MOTION.

CARD D.7 — FORMAT (18I4) — IF NSEG>18, USE 2 CARDS.

NSYM(J),J=1,NSEG — CONTROLS SYMMETRY OPTION OF BODY SEGMENTS AS FOLLOWS :

NSYM(J) = 0 : NORMAL THREE DIMENSIONAL MOTION FOR BODY SEGMENT J.

NSYM(J) = J : MOTION OF BODY SEGMENT J WILL BE RESTRICTED TO THE X-Z PLANE WITH NO LATERAL MOTION, HENCE IT WILL BE TWO DIMENSIONAL.

NSYM(J) = K : BODY SEGMENTS J AND K ARE TO REMAIN SYMMETRICAL WITH NO LATERAL MOTION. THE MOTION OF EACH WILL BE REPLACED WITH THEIR AVERAGE AND RESTRICTED TO THE LOCAL X-Z PLANE. NSYM(K) MUST EQUAL J.

NSYM(J) = -K : BODY SEGMENTS J AND K ARE TO REMAIN MIRROR SYMMETRICAL WITH RESPECT TO THE X-Z PLANE. EQUAL BUT OPPOSITE LATERAL MOTION IS PERMITTED. NSYM(K) MUST EQUAL -J.

NOTE : IN THE ABOVE SYMMETRY OPTIONS, THE USER MUST TAKE EXTREME CARE THAT ALL INPUT WILL ALLOW THE SYMMETRY TO EXIST.

IF NSD # 0, NSD D.8 CARDS ARE REQUIRED BY SUBROUTINE SINPUT.

CARDS D.8.A - D.8.J (NSD CARDS) — FORMAT (2I3, 11F6.0)

MSDM(J)
MSDN(J) — SEGMENT IDENTIFICATION NUMBERS (M AND N) TO WHICH THE JTH SPRING DAMPER IS ATTACHED.

APSDM(I,J),I=1,3
APSDN(I,J),I=1,3 — COORDINATES OF ATTACHMENT POINTS IN LOCAL SEGMENT REFERENCE ON SEGMENTS M AND N FOR THE JTH SPRING DAMPER (IN.)

ASD(I,J),I=1,5 — COEFFICIENTS OF QUADRATIC FUNCTIONS TO COMPUTE THE SPRING FORCE (FS) AND THE VISCOUS FORCE (FD) FOR THE JTH SPRING DAMPER USING THE RELATIONSHIPS

I=1 : D0 (IN)
I=2 : A1 (LB/IN)
I=3 : A2 (LB/IN**2)
I=4 : B1 (LB SEC/IN)
I=5 : B2 (LB SEC**2/IN**2)

FS= (D-D0)*(|A1| + A2*|D-D0|)
FD= DV*(B1 + B2*|DV|)

WHERE D AND DV ARE THE DISTANCE AND ITS TIME DERIVATIVE BETWEEN THE POINTS APSDM AND APSDN.
IF A1 < 0. AND (D-D0) < 0.,
PROGRAM WILL SET FS= 0., I.E. THIS WILL ACT AS TENSION ELEMENT.

E. SUBROUTINE CINPUT (FUNCTIONS INPUT)

THESE FUNCTIONS ARE REFERRED TO BY NUMBER IN THE NF ARRAYS REQUIRED ON CARDS F.1.B, F.2.B, F.3.B AND F.4.B. THEY ARE USED TO DEFINE THE FORCE DEFLECTION, INERTIAL SPIKE, R (ENERGY ABSORPTION) FACTOR, G (DEFLECTION) FACTOR AND FRICTION COEFFICENT FUNCTIONS.

EACH FUNCTION MAY BE SUBDIVIDED, IF DESIRED, INTO TWO SEPARATE PARTS, F1 AND F2, WHERE

F1(D) IS DEFINED FOR 0 .LE. DO .LE. D .LE. |D1|

F2(D) IS DEFINED FOR |D1| .LE. D .LE. |D2|.

IN ADDITION, EACH PART OF EACH FUNCTION MAY BE DEFINED IN EITHER OF THREE FUNCTIONAL FORMS: CONSTANT VALUE, TABULAR DATA, OR A FIFTH DEGRE POLYNOMIAL. THE EXISTENCE AND FORM OF EACH FUNCTION PART IS DETERMINED B THE SUPPLIED VALUES OF DO, D1, AND D2 AS FOLLOWS:

F1	F2	DO	D1	D2
CONSTANT	-	0	0	F1 = D2
TABULAR	-	DO .GE. 0	D1 .LT. 0	0
POLYNOMIAL	-	DO .GE. 0	DI .GT. 0	0
TABULAR	POLYNOMIAL	DO .GE. 0	D1 .LT. 0	D2 .GT. 0
POLYNOMIAL	TABULAR	DO .GE. 0	D1 .GT. 0	D2 .LT. 0
POLYNOMIAL	POLYNOMIAL	DO .GE. 0	D1 .GT. 0	D2 .GT. 0

THE CONSTANT FORM IS APPLICABLE TO F1 ONLY BECAUSE THE ROUTINES ASSUME

IF D .GT. |D2| THEN F(D) = F(|D2|) FOR D2 .NE. 0 OR

IF D .GT. |D1| THEN F(D) = F(|D1|) FOR D2 = 0.

THE CASE OF BOTH F1 AND F2 BEING TABULAR IS UNNECESSARY.

A MAXIMUM OF 50 FUNCTIONS MAY BE SUPPLIED TO THE PROGRAM. THESE FUNCTIONS MAY BE OF THE TYPES DESCRIBED ON EITHER CARDS E.1-E.4, CARDS E.6 OR CARDS E.7.

CARD E.1	FORMAT (I4, 4X, 5A4)
I	THE FUNCTION IDENTIFYING NUMBER. THESE NUMBERS NEED NOT BE SUPPLIED IN NUMERIC ORDER. IF THE SAME NUMBER IS USED MORE THAN ONCE, A WARNING WILL BE PRINTED AND THE LAST ONE SUPPLIED WILL BE USED. THE END OF THE FUNCTION INPUT IS INDICATED BY SUPPLYING A SINGLE CARD WITH I > 50.
KTITLE	A 20 CHARACTER ALPHANUMERIC TITLE DESCRIBING THE FUNCTION.

CARD E.2 FORMAT (5F12.0)

D0 THE LOWER ABSCISSA VALUE OF THE FIRST PART OF THE FUNCTION, F1. D0 MUST BE NON-NEGATIVE (UNITS ARE IN. EXCEPT FOR THE BELT STRESS-STRAIN FUNCTIONS WHERE THEY ARE IN/IN).

D1 THE MAGNITUDE OF DI IS THE UPPER ABSCISSA VALUE OF F1 AND THE LOWER ABSCISSA VALUE OF F2, IF ANY. $D1 < 0$ INDICATES F1 IS TABULAR, $D1 > 0$ INDICATES F1 IS A POLYNOMIAL, AND $D1 = 0$ INDICATES $F1 = D2$, A CONSTANT.

D2 IF $D1 = 0$, D2 IS THE CONSTANT VALUE OF FI. OTHERWISE, THE MAGNITUDE OF D2 IS THE UPPER ABSCISSA VALUE OF F2. IF $D2 = 0$, F2 IS NOT DEFINED; IF D2 IS NEGATIVE, F2 IS TABULAR; AND IF D2 IS POSITIVE, F2 IS A POLYNOMIAL.

D3 IF THE FUNCTION IS TO BE USED FOR AN INERTIAL SPIKE, D3 REPRESENTS THE ABSCISSA VALUE FOR WHICH THE INERTIAL SPIKE IS TO BE IGNORED IF UNLOADING OCCURS AFTER DEFLECTION EXCEEDS D3. IF THE FUNCTION IS TO BE USED FOR A COEFFICIENT OF FRICTION, $D3 = (1+U)/2$ WHERE U IS THE COEFFICIENT OF RESTITUTION FOR THE IMPULSE OPTION ($0<D3<1$ OR $-1<U<+1$). A VALUE OF $D3 = 0$ MEANS THAT THE IMPULSE OPTION WILL NOT BE USED FOR THOSE CONTACTS USING THIS FUNCTION. WHEN THE GLOBALGRAPHIC OPTION IS USED, A FRICTION FUNCTION IS DEFINED AND THE VALUE OF D3 IS USED TO SPECIFY THE IMPULSE. (SEE CARD B.5.)

D4 IF THE FUNCTION IS TO BE USED AS A FORCE DEFLECTION FUNCTION BY SUBROUTINE PLELP, D4=RHO, THE SCALAR THAT DETERMINES THE POINT OF FORCE APPLICATION. SUPPLY ZERO FOR POINT OF MAXIMUM PENETRATION, ONE FOR CENTER OF INTERSECTION ELLIPSE. IF USED AS THE FRICTION FUNCTION FOR A ROLL-SLIDE CONSTRAINT, D4 IF THE COEFFICIENT OF STATIC FRICTION TO BE USED FOR THE ROLL CONSTRAINT.

THE DEFINITIONS OF F1 AND F2, IF THEY EXIST, ARE NOW SUPPLIED ON CARD E.3 FOR THE FIFTH DEGREE POLYNOMIAL DEFINITION, OR ON CARDS E.4 FOR THE TABULAR DEFINITION.

CARD E.3	FORMAT (6F12.0)
A0,A1,A2,A3,A4,A5	COEFFICIENTS OF FIFTH-DEGREE POLYNOMIAL F = A0 + A1*X + A2*X**2 + A3*X**3 + A4*X**4 + A5*X**5 (UNITS ARE DEPENDENT ON USE OF FUNCTION.)
CARD E.4.A	FORMAT (I6)
NPI	THE NUMBER OF DATA POINTS TO BE SUPPLIED TO IDENTIFY THE FUNCTION IF IT IS DEFINED IN TABULAR FORM.
CARDS E.4.B - E.4.N	FORMAT (6F12.0)
(X(I),Y(I),I=1,NPI)	THE ABSCISSA AND ORDINATE VALUES OF THE DATA POINTS USED TO DEFINE THE TABULAR FORM OF THE FUNCTION. THE PROGRAM WILL LINEARLY INTERPOLATE TO DETERMINE INTERMEDIATE VALUES. SUPPLY 3 POINTS PER CARD; USE AS MANY CARDS AS REQUIRED. (UNITS ARE DEPENDENT ON USE OF FUNCTION.)

SUBROUTINE KINPUT (WIND FORCE AND JOINT RESTORING FORCE FUNCTIONS)

CARD E.5 IS ALWAYS REQUIRED AFTER THE END-OF-DATA CARD E.1 (I > 50). MAY BE BLANK TO DESIGNATE NO FUNCTIONS ON CARDS E.6 OR E.7.

CARD E.5	FORMAT (2I6)
NWINDF	THE NUMBER OF WIND FORCE FUNCTIONS TO BE SUPPLIED ON CARDS E.6.A-E.6.N. MAY BE ZERO.
NJNTF	THE NUMBER OF JOINT RESTORING FORCE FUNCTIONS TO BE SUPPLIED ON CARDS E.7.A-E.7.N. MAY BE BLANK OR ZERO.

NWINDF SETS OF CARDS E.6.A - E.6.N ARE REQUIRED.

CARD E.6.A	FORMAT (I4, 4X, 5A4)
I,KTITLE	SAME AS CARD E.1 EXCEPT THAT EACH FUNCTION NUMBER (I) MUST BE LESS THAN 51 AND MUST BE DISTINCT FROM THOSE SUPPLIED ON CARDS E.1.
CARD E.6.B	FORMAT (5F12.0)
D0,D1,D2,D3,D4	CURRENTLY NOT USED BY PROGRAM.
CARD E.6.C	FORMAT (I6)
NTMPTS	THE NUMBER OF TIME POINTS OR CARDS REQUIRED TO DEFINE THIS FUNCTION ON CARDS E.6.D-E.6.N.
CARDS E.6.D - E.6.N (NTMPTS CARDS)	FORMAT (4F12.0)
T	TIME (SEC.) SINCE INITIAL PENETRATION OF BOUNDARY PLANE. VALUES SHOULD BE IN ASCENDING ORDER WITH FIRST VALUE EQUAL TO ZERO.
FX,FY,FZ	THE X,Y AND Z COMPONENTS OF FORCE PER UNIT AREA (LBS./IN.**2) IN INERTIAL REFERENCE DUE TO THE WIND BLAST FORCE AT TIME T. THE PROGRAM WILL USE LINEAR INTERPOLATION ON T. IF LAST VALUE OF T IS EXCEEDED, THE LAST VALUES OF FX,FY AND FZ WILL BE USED.

NJNTF (FROM CARD E.5) SETS OF CARDS E.7.A - E.7.N ARE REQUIRED.

CARD E.7.A	FORMAT (I4, 4X, 5A4)
I,KTITLE	SAME AS CARD E.1 EXCEPT THAT EACH FUNCTION NUMBER (I) MUST BE LESS THAN 51 AND MUST BE DISTINCT FROM THOSE SUPPLIED ON CARDS E.1 OR CARDS E.6.A.
CARD E.7.B	FORMAT (5F12.0)
D0,D1,D2,D3,D4	CURRENTLY NOT USED BY PROGRAM.
CARD E.7.C	FORMAT (2I6)
NTHETA	MAGNITUDE INDICATES THE NUMBER OF COLUMNS IN THE TWO DIMENSIONAL INPUT DATA MATRIX TO BE SUPPLIED ON CARDS E.7.D-E.7.N. THE MINIMUM VALUE IS 2. IF POSITIVE, THE NTHETA ENTRIES IN EACH ROW WILL BE TABULAR DATA FOR EQUALLY SPACED VALUES OF THE JOINT FLEXURE ANGLE (THETA) BETWEEN 0 AND 180 DEGREES. IF NEGATIVE, THE ENTRIES WILL REPRESENT THE COEFFICIENTS OF A (-NTHETA-1) ORDER POLYNOMIAL IN (THETA-THETAO)
NPHI	NUMBER OF ROWS OF MATRIX OF DATA TO BE SUPPLIED ON CARDS E.7.D-E.7.N. EACH ROW REPRESENTS EQUALLY SPACED VALUES OF THE JOINT AZIMUTH ANGLE (PHI) BETWEEN -180 AND +180 DEGREES, BUT DOES NOT INCLUDE THE LAST ROW SINCE THE PROGRAM ASSUMES DATA FOR PHI(NPHI+1)=180 ARE THE SAME AS FOR PHI(1)=-180. MINIMUM = 1.

CARDS E.7.D - E.7.N FORMAT (5F12.0)
(NPHI SETS OF CARDS. USE EXTRA CARDS PER SET IF |NTHETA| > 5.)

THETAO	THE VALUE OF THE "DEAD BAND" ZONE FOR THIS VALUE OF PHI (DEGREES). IF THE FLEXURE ANGLE (THETA) IS LESS THAN THETAO, THE JOINT RESTORING FORCE WILL BE ZERO.
F(J),J=2,NTHETA	FOR NTHETA POSITIVE, TABULAR VALUES OF THE JOINT RESTORING FORCE FOR FLEXURE ANGLES THETA(J) = (J-1)*180/(NTHETA-I) DEGREES VALUES OF ZERO SHOULD BE SUPPLIED FOR THETA < THETAO. FOR NTHETA NEGATIVE, THE COEFFICIENTS OF A POLYNOMIAL IN (THETA-THETAO) OF ORDER ONE LESS THAN THE MAGNITUDE OF NTHETA. F(J) IS THE COEFFICIENT OF (THETA-THETAO)**(J-1) WHERE (THETA-THETAO) IS EXPRESSED IN RADIANS. F(I) IS ASSUMED TO BE ZERO.

F. SUBROUTINE FINPUT (ALLOWED CONTACTS AND ASSOCIATED FUNCTIONS)

IF NPL # 0, F.1 IS REQUIRED.

CARD F.1.A	FORMAT (18I4)	IF NPL>18, USE 2 CARDS.
MNPL(J),J=1,NPL	FOR PLANE J, THE NUMBER OF SEGMENTS FOR WHICH SEGMENT-PLANE CONTACT IS ALLOWED. NPL IS THE NUMBER OF PLANES FROM CARD D.1. THE VALUE OF ANY MNPL FOR PLANE J MAY BE ZERO AND THE MAXIMUM VALUE IS 5. HOWEVER IF IT IS REQUIRED TO HAVE MORE THAN 5 SEGMENTS CONTACT THE SAME PLANE, SET UP TWO OR MORE IDENTICAL PLANES AND PERMIT A MAXIMUM OF 5 SEGMENTS TO CONTACT EACH PLANE.	

FOR EACH PLANE J, MNPL(J) CARDS OF THE FOLLOWING MUST BE SUPPLIED.

CARDS F.1.B - F.1.N	FORMAT (9I4)
NJ	THE PLANE NUMBER FOR WHICH CONTACT IS ALLOWED. NJ MUST CORRESPOND TO J ABOVE. THERE MUST BE MNPL(J) CARDS WITH THIS SAME NJ. IF MNPL(J) = 0, NO NJ = J SHOULD BE PRESENT.
NS(1)	THE SEGMENT NUMBER TO WHICH PLANE J IS ATTACHED. IF VEHICLE, SUPPLY NSEG+1, IF GROUND, SUPPLY NSEG+2.
NS(2)	THE SEGMENT NUMBER (DETERMINED BY THE CARD NUMBER I UNDER CARD B.2.A FOR WHICH CONTACT WITH THE NJTH PLANE IS ALLOWED.
NS(3)	THE NUMBER OF THE CONTACT ELLIPSOID ASSOCIATED WITH THE SEGMENT NS(2).

NF(1)	THE FUNCTION NUMBER FROM CARD E.1 TO DEFINE THE FORCE-DEFLECTION FUNCTION FOR THIS CONTACT. IF NF(1) = 0, A ROLLING - SLIDING CONSTRAINT OPTION WILL BE EXERCISED BY THE PROGRAM FOR THIS CONTACT WHICH DOES NOT REQUIRE NF(2), NF(3) OR NF(4) BUT DOES REQUIRE A FRICTION COEFFICIENT FUNCTION TO BE DEFINED BY NF(5). THE VALUE OF D3 ON CARD E.2 OF THIS FUNCTION SHOULD BE 0.5 (NON-ZERO TO ACTIVATE THE IMPULSE AND TO SET THE NORMAL COMPONENT OF RELATIVE VELOCITY TO ZERO AFTER THE IMPULSE HAS BEEN APPLIED). ALSO THE INITIAL POSITIONS ON CARDS G.2 MUST BE SUCH THAT CONTACT DOES NOT EXIST AT TIME = 0.
NF(2)	THE FUNCTION NUMBER FROM CARD E.1 TO DEFINE THE INERTIAL SPIKE FUNCTION FOR THIS CONTACT. IF NF(2) = 0, NO INERTIAL SPIKE EXISTS.
NF(3)	THE FUNCTION NUMBER FROM CARD E.1 TO DEFINE THE R (ENERGY-ABSORPTION) FACTOR FUNCTION. IF NF(3) = 0, A DEFAULT VALUE OF R = 1 IS ASSUMED.
NF(4)	THE FUNCTION NUMBER FROM CARD E.1 TO DEFINE THE G (DEFLECTION) FACTOR FUNCTION. IF NF(4) = 0, A DEFAULT VALUE OF G = 0 IS ASSUMED.
NF(5)	THE FUNCTION NUMBER FROM CARD E.1 TO DEFINE THE FRICTION COEFFICIENT FUNCTION.

IF NBLT # 0, F.2 IS REQUIRED.

CARD F.2.A	FORMAT (8I4)
MNBLT(J),J=1,NBLT	FOR BELT J, THE NUMBER OF SEGMENTS FOR WHICH SEGMENT-BELT INTERACTION IS ALLOWED. NBLT IS THE NUMBER OF BELTS FROM CARD D.1. EACH MNBLT MAY HAVE A VALUE OF 0 OR 1 ONLY.

FOR EACH BELT J, MNBLT(J) CARDS OF THE FOLLOWING MUST BE SUPPLIED.

CARDS F.2.B - F.2.N	FORMAT (9I4)
NJ	THE BELT NUMBER TO BE CONTACTED, MUST CORRESPOND TO J ABOVE. THERE MUST BE MNBLT(J) CARDS WITH THE SAME NJ. IF MNBLT(J) = 0, NO NJ = J SHOULD BE PRESENT.
NS(1)	THE SEGMENT NUMBER TO WHICH BELT NJ IS ATTACHED. IF VEHICLE, SUPPLY NSEG+1, IF GROUND, SUPPLY NSEG+2.
NS(2)	THE SEGMENT NUMBER (DETERMINED BY THE CARD NUMBER I UNDER CARD B.2.A) FOR WHICH INTERACTION WITH THE NJTH BELT IS ALLOWED.
NS(3)	THE NUMBER OF THE CONTACT ELLIPSOID ASSOCIATED WITH THE SEGMENT NS(2).
NF(1)	THE FUNCTION NUMBER FROM CARD E.1 TO DEFINE THE FORCE-DEFLECTION FUNCTION FOR THIS CONTACT. THE ABSCISSA FOR THIS FUNCTION SHOULD BE STRAIN (IN/IN).
NF(I),I=2,4	SAME DEFINITION AS ON CARD F.1.B ABOVE.
NF(5)	IF NON-ZERO, FULL BELT FRICTION IS ASSUMED, I.E., FORCES ARE COMPUTED FOR EACH HALF OF THE BELT SEPARATELY. IF ZERO, ZERO BELT FRICTION IS ASSUMED, I.E., BELT TENSION IS IS THE SAME AT BOTH BELT ANCHOR POINTS.

A BLANK F.3.A CARD IS REQUIRED FOR NO SEGMENT-SEGMENT CONTACTS.

CARD F.3.A		FORMAT (18I4) IF NSEG>18, USE TWO CARDS.
MNSEG(J),J=1,NSEG		FOR SEGMENT J, THE NUMBER OF SEGMENTS FOR WHICH SEGMENT-SEGMENT CONTACT IS ALLOWED. NSEG IS THE NUMBER OF SEGMENTS FROM CARD B.1. EACH SEGMENT CONTACT, A VERSUS B, MAY BE INPUTTED EITHER WAY EXCEPT WHERE AN INTERIOR CONTACT IS DESIRED (SEE NS(3)). ANY OR ALL VALUES OF MNSEG MAY BE ZERO. THE MAXIMUM VALUE FOR EACH MNSEG IS 5.

FOR EACH SEGMENT J, MNSEG(J) CARDS OF THE FOLLOWING MUST BE SUPPLIED.

CARDS F.3.B - F.3.N		FORMAT (9I4)
	NJ	THE SEGMENT NUMBER TO BE CONTACTED, MUST CORRESPOND TO J ABOVE. THERE MUST BE MNSEG(J) CARDS WITH THIS SAME NJ. IF MNSEG(J) = 0, NO NJ = J SHOULD BE PRESENT.
	NS(1)	THE NUMBER OF THE CONTACT ELLIPSOID ASSOCIATED WITH SEGMENT NJ.
	NS(2)	THE SEGMENT NUMBER (DETERMINED BY THE CARD NUMBER I UNDER CARD B.2.A) FOR WHICH CONTACT WITH THE NJTH SEGMENT IS ALLOWED.
	NS(3)	THE NUMBER OF THE CONTACT ELLIPSOID ASSOCIATED WITH THE SEGMENT NS(2). IF NEGATIVE, AN INTERIOR CONTACT WILL BE ASSUMED WITH ELLIPSOID NS(1) INSIDE NS(3).
	NF(I),I=1,5	SAME DEFINITIONS AS ON CARD F.1.B ABOVE.

IF NJNT > 0, F.4.A IS REQUIRED.
SUPPLY IGLOB=1 FOR GLOBALGRAPHIC OPTION, OTHERWISE SUPPLY 0 OR BLANK

CARD F.4.A	FORMAT (18I4) IF NJNT>18, USE TWO CARDS.
IGLOB(J),J=1,NJNT	FOR EACH JOINT J, SUPPLY 1 FOR IGLOB(J) IF IPIN(J) IS +3 OR -3 ON CARDS B.3.A - B.3.J; OTHERWISE SUPPLY ZERO OR BLANK. ONE CARD F.4.J MUST BE SUPPLIED BELOW FOR EACH J FOR WHICH IGLOB(J) =1.
CARDS F.4.B - F.4.J	FORMAT (9I4)
NJ	THE IDENTIFICATION NUMBER FOR A GLOBALGRAPHIC JOINT, MUST CORRESPOND TO J ABOVE AND CARDS MUST BE SUPPLIED IN ASCENDING ORDER ON NJ.
NS(I),I=1,3	CURRENTLY NOT USED BY PROGRAM.
NF(1)	THE FUNCTION NUMBER FROM CARD E.1 TO DEFINE THE TORQUE-DEFLECTION FOR THIS GLOBALGRAPHIC JOINT. THE ORDINATE FOR THIS FUNCTION SHOULD BE TORQUE (IN. LB.) AND THE ABSCISSA IS THE ANGULAR DEFLECTION (RADIANS) INTO THE STOP.
NF(2)	THE FUNCTION NUMBER FROM CARD E.1 TO DEFINE THE HERRON FORMULAS FOR T (JOINT STOP ANGLE IN RADIANS) AND ITS DERIVATIVE TP WITH RESPECT TO PHI BOTH AS FUNCTIONS OF PHI (THE JOINT ANGLE FROM THE REFERENCE AXIS IN RADIANS). NORMALLY THEY WILL BE COMPUTED BY

```
 T = P1 + SP*P2
TP = P1' + CP*P2 + SP*P2'
```

WHERE P1,P2 ARE THE 5TH DEGREE POLYNOMIAL EVALUATIONS OF COS(PHI) USING THE TWO POLYNOMIALS F1 AND F2 OBTAINED BY SETTING BOTH D1,D2 > 0 ON CARD E.2;

P1',P2' ARE THEIR DERIVATIVES WITH RESPECT TO PHI;

AND CP,SP ARE COS(PHI) AND SIN(PHI).

IF D1,D2 ARE NOT BOTH POSITIVE, T AND TP WILL BE EVALUATED AS FUNCTIONS OF PHI IN RADIANS (0 < PHI < 2*PI) AS SPECIFIED ON CARDS E.1 - E.4 FOR FUNCTION NF(2).

SAME DEFINITIONS AS ON CARD F.1.B ABOVE

IF NJNT > 0, CARD F.5.A IS ALWAYS REQUIRED BUT MAY BE BLANK.

CARD F.5.A	FORMAT (I8I4) USE TWO CARDS IF NJNT > I8.
JOINTF(J),J=I,NJNT	FOR EACH JOINT (J), THE FUNCTION IDENTIFIC-ATION NUMBER AS SUPPLIED ON CARDS E.7.A TO BE USED BY SUBROUTINE VISPR TO COMPUTE THE JOINT RESTORING FORCE BY FUNCTION FNTERP. IF ZERO, THE VALUES OF SPRING(1,3*J-2) AS SUPPLIED ON CARDS B.4.A WILL BE USED USING FUNCTION EJOINT.

IF NBAG # 0, NBAG CARDS OF THE FOLLOWING MUST BE SUPPLIED. SINCE THE AIR BAG ROUTINES DO NOT USE THE FORCE-DEFLECTION FUNCTIONS, THIS INPUT HAS DIFFERENT FORMATS THAN THE ABOVE ALLOWED CONTACTS.

CARDS F.6.A - F.6.N	FORMAT (2I4, 20I2)
K	THE AIR BAG NUMBER CORRESPONDING TO THE INDEX J UNDER CARDS D.4 ABOVE. K MUST BE IN NUMERIC ORDER K = 1 TO NBAG, WHERE NBAG IS THE NUMBER OF AIR BAGS DEFINED ON CARD D.I.
NK	THE NUMBER OF SEGMENTS ALLOWED TO CONTACT THE KTH AIR BAG. THE MAXIMUM VALUE IS 10. IF NK = 0, THE REMAINDER OF THE CARD IS BLANK.
MBAG(2,I,K), MBAG(3,I,K),I=1,NK	THE SEGMENT NUMBERS (DETERMINED BY THE CARD NUMBER I UNDER CARD B.2.A) EACH FOLLOWED BY THE NUMBER OF THE ASSOCIATED CONTACT ELLIPSOID FOR WHICH CONTACT FORCES WITH THE KTH AIR BAG WILL BE COMPUTED.

CARD F.7.A IS ALWAYS REQUIRED. INSERT A BLANK CARD IF NO WIND FORCE CALCULATIONS ARE TO BE PERFORMED.

CARD F.7.A — FORMAT (18I4) USE TWO CARDS IF NSEG > 18.

MWSEG(1,J),J=1,NSEG	FOR EACH SEGMENT J, SUPPLY ZERO IF NO WIND FORCE CALCULATIONS ARE TO BE PERFORMED. OTHERWISE, SUPPLY A VALUE OF ONE TO INDICATE WIND FORCES ARE TO BE PERFORMED.

SUPPLY CARD F.7.B FOR EACH SEGMENT (J) WHERE MWSEG(1,J) = 1.

CARD F.7.B — FORMAT (5I4)

JJ	THE SEGMENT IDENTIFICATION NUMBER FROM CARDS B.2.A FOR WHICH WIND FORCE CALCULATIONS ARE TO BE PERFORMED. MUST CORRESPOND TO J FROM CARD F.7.A AND BE SUPPLIED IN ASCENDING ORDER.
MWSEG(2,J)	THE NUMBER OF THE CONTACT ELLIPSOID TO BE ASSOCIATED WITH SEGMENT NUMBER JJ.
MWSEG(3,J)	THE SEGMENT IDENTIFICATION NUMBER (NSEG+1 FOR THE VEHICLE, NSEG+2 FOR THE GROUND) ASSOCIATED WITH PLANE NUMBER MWSEG (4,J).
MWSEG(4,J)	THE PLANE IDENTIFICATION NUMBER FROM CARD D.2.A THROUGH WHICH IF SEGMENT J PASSES, WIND FORCE CALCULATIONS WILL BE PERFORMED.
MWSEG(5,J)	THE FUNCTION NUMBER FROM CARD E.6.A FOR THE WIND FORCE FUNCTION TO BE USED.

F.8 SUBROUTINE HINPUT - CARD INPUT FOR HARNESS-BELT SYSTEMS.

CARD F.8.A IS ALWAYS REQUIRED. INSERT BLANK CARD IF NO HARNESS-BELT SYSTEMS ARE DESIRED.

CARD F.8.A	FORMAT (6I4)
NHRNSS	NUMBER OF HARNESS-BELT SYSTEMS TO BE SUPPLIED ON CARDS F.8.B-F.8.D. MAY BE ZERO OR BLANK. MAXIMUM VALUE = 5.
NBLTPH(I), I=I,NHRNSS	NUMBER OF INDIVIDUAL BELTS FOR EACH HARNESS NO. I. MAY BE ZERO OR BLANK. MAXIMUM VALUE OF SUM OF ALL NBLTPH IS 20.

CARD F.8.A IS FOLLOWED BY NHRNSS SETS OF CARDS F.8.B - F.8.D.

CARD F.8.B	FORMAT (18I4) USE TWO CARDS IF NBLTPH(I)>I8.
NPTSPB(J), J=I,NBLTPH(I)	THE NUMBER OF REFERENCE POINTS INCLUDING ANCHOR POINTS FOR BELT NO. J OF HARNESS NO. I. MAY BE ZERO OR BLANK. THE MAXIMUM VALUE OF THE SUM OF ALL NPTSPB FOR ALL HARNESS-BELT SYSTEMS IS 100.

EACH CARD F.8.B IS FOLLOWED BY NBLTPH(I) SETS OF CARDS F.8.C - F.8.D.

CARD F.8.C	FORMAT (5I4, F12.6)
NFBLT(L,J),L=1,5	THE IDENTIFICATION NUMBERS OF THE 5 FUNCTIONS TO BE USED FOR BELT NO. J. THESE CORRESPOND TO I AS SUPPLIED ON CARDS E.1 OF THE FUNCTION DEFINITIONS. THESE FUNCTIONS ARE IDENTICAL TO THOSE DEFINED BY NF(I) - NF(5) ON CARDS F.2.B EXCEPT THAT THE COEFFICIENT OF FRICTION AS SPECIFIED BY NF(5) IS NOT CURRENTLY USED.
XLONG(J)	BELT SLACK (IN). THE SLACK. WHEN ADDED TO THE INITIAL GEOMETRIC LENGTH, RESULTS IN THE INITIAL BELT LENGTH. IF DESIRED, THE INITIAL BELT LENGTH MAY BE SUPPLIED AS A NEGATIVE NUMBER AND THE PROGRAM WILL COMPUTE THE SLACK.

EACH CARD F.8.C IS FOLLOWED BY NPTSPB(J) CARDS F.8.D SPECIFYING THE REFERENCE POINTS (K) TO BE USED FOR BELT NO. J OF HARNESS NO. I.

CARD F.8.D	FORMAT (2I6, 3F12.6)
IBAR(1,K)	THE IDENTIFICATION NUMBER OF THE SEGMENT ASSOCIATED WITH REFERENCE POINT K.
IBAR(2,K)	THE IDENTIFICATION NUMBER OF THE CONTACT ELLIPSOID ASSOCIATED WITH POINT K. IF ZERO, PROGRAM WILL ASSUME BELT IS RIGIDLY ATTACHED TO THAT POINT (AS FOR ANCHOR POINTS ATTACHED TO THE VEHICLE).
BAR(L,K),L=1,3	THE X,Y AND Z COORDINATES OF REFERENCE POINT K IN THE LOCAL COORDINATE SYSTEM OF SEGMENT NO. IBAR(1,K). THE PROGRAM WILL ASSUME THAT BELT J WILL RUN THROUGH THE POINTS IN THE ORDER THEY ARE SUPPLIED. HOWEVER IF A CONTACT ELLIPSOID IS SPECIFYED BY IBAR(2,K) AND THE THE FORCES ARE SUCH AS TO PULL THE BELT AWAY FROM THE SURFACE, THIS POINT WILL BE IGNORED THAT TIME POINT.

G. SUBROUTINE INITAL

CARD G.I	FORMAT (3F10.0, 5I4)
ZPLT(I),I=I,3	THE X, Y, AND Z PLOT COORDINATES (FOR SUBROUTINE PRIPLT) OF THE ORIGIN OF THE VEHICLE REFERENCE SYSTEM. 0 < X < 61 0 < Y < 6I 0 < Z < I2I
I1,J1,I2,J2	NOT USED BY THE CURRENT PROGRAM.
I3	IF ZERO, SEGMENT AND ANGULAR VELOCITIES ARE NOT SUPPLIED ON THE FOLLOWING CARDS BUT ARE SET EQUAL TO THE INITIAL VEHICLE VELOCITY. IF I3 # 0, SEGLV AND WMGDEG MUST BE SUPPLIED.

ONE G.2 CARD MUST BE SUPPLIED FOR EACH REFERENCE SEGMENT (I.E., SEGMENT NO. 1 AND FOR EACH SEGMENT J+I WHERE JNT(J) = 0 ON CARDS B.3) IN ASCENDING SEGMENT NUMBER SEQUENCE.

CARDS G.2.A - G.2.M	FORMAT (6F10.0)
SEGLP(I,J),I=1,3	THE INITIAL X, Y, AND Z COORDINATES OF THE JTH BODY SEGMENT IN INERTIAL REFERENCE (IN).
SEGLV(I,J),I=1,3	THE INITIAL X, Y, AND Z COMPONENTS OF VELOCITY OF THE JTH BODY SEGMENT IN INERTIAL REFERENCE (IN/SEC). THESE FIELDS MAY BE LEFT BLANK IF I3 = 0 ON CARD G.I IN WHICH CASE THE INITIAL VELOCITY OF THE VEHICLE WILL BE USED.

CARDS G.3.A - G.3.N (NSEG CARDS)	FORMAT (6F10.0)
YPR(I,J),I=I,3	THE INITIAL YAW, PITCH AND ROLL ANGLES OF THE JTH BODY SEGMENT (DEGREES).

NOTE: THE DIRECTION COSINE MATRICES OF THE BODY SEGMENTS ARE INITIALLY COMPUTED BY ASSUMING THE ORDER OF THE ROTATING ANGLES IS REVERSED, I.E., ROLL,PITCH,YAW.
(ROLL ABOUT X, PITCH ABOUT Y, AND YAW ABOUT Z.)

WMGDEG(I,J),I=I,3	THE INITIAL COMPONENTS OF ANGULAR VELOCITY ABOUT THE LOCAL X,Y AND Z AXES OF THE JTH BODY SEGMENT (DEG/SEC). IF I3 = 0 ON CARD G.I, THE INITIAL ANGULAR VELOCITY OF THE VEHICLE WILL BE CONVERTED TO THE SEGMENT REFERENCE AND WILL BE USED.

H. SUBROUTINE HEDING

THIS SUBROUTINE PROVIDES INPUT TO CONTROL THE DESIRED TIME HISTORY OUTPUT OF SELECTED SEGMENT LINEAR AND ANGULAR ACCELERATIONS, VELOCITIES, AND DISPLACEMENTS, AND JOINT PARAMETERS.

H.1 SEGMENT LINEAR ACCELERATIONS (K = 1)

CARD H.1.A	FORMAT (2I6, 3F12.6)
NSG(K)	THE NUMBER OF SELECTED POINTS ON THE VARIOUS BODY SEGMENTS FOR WHICH TIME HISTORIES ARE DESIRED. THE MAXIMUM VALUE FOR NSG(K) IS 20. IF NSG(K) IS 0, INSERT 2 BLANK CARDS. IF NSG(K) IS 1, A SINGLE BLANK CARD SHOULD FOLLOW CARD H.1.K.
MSG(1,K)	THE SEGMENT NUMBER AS DETERMINED BY INDEX I ON CARDS B.2.A - B.2.N OF THE FIRST POINT.
XSG(I,1,K),I=1,3	THE X, Y, AND Z COORDINATES IN SEGMENT REFERENCE OF THE FIRST POINT (INCHES).

FOLLOWED BY NSG(K)-1 CARDS OF THE FOLLOWING (J = 2, NSG(K))

CARDS H.1.B - H.1.N	FORMAT (I12, 3F12.6)
MSG(J,K)	SAME AS ABOVE BUT FOR THE JTH POINT.
XSG(I,J,K),I=1,3	SAME AS ABOVE BUT FOR THE JTH POINT.

H.2 SEGMENT LINEAR VELOCITIES (K = 2)

CARDS H.2.A - H.2.N FORMAT (2I6, 3F12.6/(I12, 3F12.6))

DESCRIPTION SAME AS FOR H.1.

H.3 SEGMENT LINEAR DISPLACEMENTS (K = 3)

CARDS H.3.A - H.3.N FORMAT (2I6, 3F12.6/(I12, 3F12.6))

DESCRIPTION SAME AS FOR H.1.

H.4 SEGMENT ANGULAR ACCELERATIONS (K = 4)

CARD H.4	FORMAT (12I6/I12, 3I6)
NSG(K)	THE NUMBER OF SELECTED BODY SEGMENTS FOR WHICH TIME HISTORIES ARE DESIRED. INSERT BLANK CARD IF NONE ARE DESIRED (NSEG MAXIMUM).
MSG(J,K),J=1,KSG WHERE KSG=NSG(K)	THE SEGMENT NUMBERS AS DETERMINED BY INDEX I ON CARDS B.2.A - B.2.N. IF NSG(K) > 11, USE THE SECOND CARD, LEAVING THE FIRST FIELD OF 6 COLUMNS BLANK. IF NSG(K) = 11, A SECOND CARD, COMPLETELY BLANK, SHOULD FOLLOW THIS CARD.

H.5 SEGMENT ANGULAR VELOCITIES (K = 5)

CARD H.5	FORMAT (12I6/I12, 3I6)

DESCRIPTION SAME AS FOR H.4.

H.6 SEGMENT ANGULAR DISPLACEMENTS (K = 6)

CARD H.6	FORMAT (12I6/I12, 3I6)

DESCRIPTION SAME AS FOR H.4.

H.7 JOINT PARAMETERS (K = 7)

CARD H.7	FORMAT (12I6/I12, 2I6)
NSG(K)	THE NUMBER OF SELECTED JOINTS FOR WHICH TIME HISTORIES ARE DESIRED. INSERT BLANK CARD IF NONE ARE DESIRED (NJNT MAXIMUM).
MSG(J,K),J=I,KSG WHERE KSG=NSG(K)	THE JOINT NUMBERS AS DETERMINED BY INDEX J ON CARDS B.3.A - B.3.J. IF NSG(K) > 11, USE A SECOND CARD LEAVING THE FIRST FIELD OF 6 COLUMNS BLANK. IF NSG(K) = 11, A SECOND CARD, COMPLETELY BLANK, SHOULD FOLLOW THIS CARD.

APPENDIX B

The listing of 21 subroutines that follow represent the changes or additions that were made to the computer program contained in Volume IV, Programmer's Manual of Calspan Report No. ZQ-5180-L-1, "An Improved Three Dimensional Computer Simulation of Motor Vehicle Crash Victims," July 1974 to fulfill the requirements of Wright Patterson AFB Contract No. F33615-75-C-5002. Any subroutine not contained herein remains unchanged from the above mentioned report.

The following is a list of the included subroutines and a summary of the changes that have been made to them.

1. SUBROUTINE CINPUT: Statement numbers have been renumbered for readability and previous version has been subdivided into new SUBROUTINE CINPUT and SUBROUTINE FINPUT. Calls to new subroutines KINPUT and HINPUT have been added.

2. SUBROUTINE CONTCT: Card No.'s 140-160 and 780-1050 have been added to control the calling of SUBROUTINE WINDY and SUBROUTINE HBELT.

3. SUBROUTINE DINT: Card No.'s 530 and 2150-2170 have been modified to simplify program logic and are equivalent to previous version.

4. SUBROUTINE ELTIME: Card No.'s 210 and 220 have been modified to include SUBROUTINE WINDY and SUBROUTINE HBELT for N=35 and 36.

5. FUNCTION EVALFD: Extensive modifications have been made to accomodate abscissas that exceed the range of tabular function definitions.

6. SUBROUTINE FINPUT: New subroutine that is actually the second half of previous SUBROUINTE CINPUT that controlled the input specifying allowed contacts between body segments with vehicle panels, belts, airbags and other body segments. New code has been inserted at card no.'s 2100-2280 for new input card F.5 defining joint functions to be used. Old input cards F.5 have been renamed F.6 and new code has been inserted at card no.'s 2540-2800 for input cards F.7 controlling the new wind force calculations.

APPENDIX B (Continued)

7. SUBROUTINE FLXSEG: Card No.'s 130 and 140 have been interchanged to properly control the call to SUBROUTINE ELTIME.

8. FUNCTION FNTERP: New subroutine that computes the restoring torque of a joint by double linear interpolation on the flexure angle (theta) and azimuth angle (phi).

9. SUBROUTINE HBELT: New subroutine that computes the forces and torques of individual belt sections of the harness-belt systems.

10. SUBROUTINE HINPUT: New subroutine that controls the input of cards F.8.A-F.8.D containing the setup and control of the harness-belt system.

11. SUBROUTINE IMPULS: Card No.'s 1810 and 1820 have been modified to properly control call to SUBROUTINE ELTIME.

12. SUBROUTINE KINPUT: New subroutine that controls the input of cards E.5, E.6 and E.7 containing the definitions of the wind force and joint restoring force functions.

13. SUBROUTINE OUTPUT: Card No.'s 1280-1300 have been modified to print the joint angles in degrees for the new joint functions.

14. SUBROUTINE PLELP: Comments in card no.'s 50 and 60 have been corrected.

15. SUBROUTINE RSTART: Card No.'s 630-640, 1480-1580, 1630-1640, 1770, 1890, 1970, 2060, 2150, 2170-2180, 2930 and 4410-4620 have been modified or added to insert JOINTF in COMMON/DESCRP/ and to include COMMON/HARNESS/ and COMMON/KALEPS/.

16. SUBROUTINE SEARCH: Several additions and modifications have been made to accomodate the changes made to SUBROUTINE RSTART.

APPENDIX B (Continued)

17. SUBROUTINE UPDATE: Card No.'s 360-370, 1020-1210 have been added to call SUBROUTINE UPDFDC for each belt of harness-belt systems and card no. 2400 has been modified to set initial state of rolling-sliding constraint.

18. SUBROUTINE VEHPOS: Card No.'s 410-440 and 1100-1110 have been modified to extrapolate beyond last entry in vehicle position input tables.

19. SUBROUTINE VINPUT: Card No.'s 560, 600 and 750-810 have been modified to permit input of negative VIPS on input card C.2 and delete the restriction that the last acceleration on card C.3 be zero.

20. SUBROUTINE VISPR: Card No.'s 190, 310, 560-570, 600-870, 930, 1090-1100, 1190-1230, 1350-1410, 1450, 1470-1480, 1500, 1560, 1580, 1650-1670, 1800, 1830-1840, 1990-2000 and 2020 have been added or modified to include the necessary logic for the new joint functions.

21. SUBROUTINE WINDY: New subroutine that computes the forces and torques of a wind blast acting on specified body segments.

```
      SUBROUTINE CINPUT                                                 CINP0010
C                                                  REV 12 12/18/74CINP0020
C     CONTROLS THE CARD INPUT OF THE FORCE-DEFLECTION, INERTIAL SPIKE,  CINP0030
C     R FACTOR, G FACTOR AND FRICTION COEFFICIENT FUNCTION DEFINITIONS  CINP0040
C                                                                       CINP0050
      IMPLICIT REAL*8(A-H,O-Z)                                          CINP0060
      COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(500),TAB(2000) CINP0070
      COMMON/TEMPVS/JTITLE(5,51),NF(5),NS(3),KTITLE(31)                 CINP0080
      REAL  JTITLE,KTITLE                                               CINP0090
      IS = 0                                                            CINP0100
      DO 10 I = 1,50                                                    CINP0110
   10 NTI(I) = 0                                                        CINP0120
      J1 = 1                                                            CINP0130
C                                                                       CINP0140
C     INPUT CARD E.1 - FUNCTION NO. AND TITLE, IF NO. > 50 SKIP OUT.    CINP0150
C                                                                       CINP0160
   11 READ (5,12) I,(KTITLE(J  ),J = 1,5)                               CINP0170
   12 FORMAT (I4,4X,5A4)                                                CINP0180
      IF (I.GT.50) GO TO 30                                             CINP0190
      DO 13 J = 1,5                                                     CINP0200
   13 JTITLE(J,I) = KTITLE(J)                                           CINP0210
C                                                                       CINP0220
C     HAS FUNCTION NO. BEEN ALREADY USED?                               CINP0230
C                                                                       CINP0240
      IF  (NTI(I).NE.0)  WRITE(6,14) I                                  CINP0250
   14 FORMAT('0 FUNCTION NO.',I4,' HAS ALREADY BEEN INPUTTED AND WILL BECINP0260
     *REPLACED BY NEXT FUNCTION')                                       CINP0270
      NTI(I) = J1                                                       CINP0280
      J2 = J1+4                                                         CINP0290
C                                                                       CINP0300
C     INPUT CARD E.2                                                    CINP0310
C                                                                       CINP0320
      READ (5,15) (TAB(J),J = J1,J2)                                    CINP0330
   15 FORMAT (6F12.0)                                                   CINP0340
      IS = 1-IS                                                         CINP0350
      IF (IS.EQ.0) WRITE (6,16)                                         CINP0360
   16 FORMAT(/////)                                                     CINP0370
      WRITE (6,17) IS,I,(JTITLE(J,I),J=1,5),I,NTI(I),(TAB(J),J=J1,J2)   CINP0380
   17 FORMAT(I1,'FUNCTION NO.',I4,4X,5A4,20X,'NTI(',I2,') =',I5,45X     CINP0390
     * 'CARDS E'//10X,'D0',13X,'D1',13X,'D2',13X,'D3',13X,'D4'/5F15.4//)CINP0400
      D0 = TAB(J1)                                                      CINP0410
      D1 = TAB(JI+1)                                                    CINP0420
      D2 = TAB(J1+2)                                                    CINP0430
      JI = J2+1                                                         CINP0440
      IF (D1)  22,18,20                                                 CINP0450
C                                                                       CINP0460
C     FUNCTION IS CONSTANT  D2 FOR ALL D.                               CINP0470
C                                                                       CINP0480
   18 WRITE (6,19)  D2                                                  CINP0490
   19 FORMAT(7X,'FUNCTION IS CONSTANT',F12.6)                           CINP0500
```

```
      GO TO 11                                                          CINP0510
C                                                                       CINP0520
C        5TH ORDER POLYNOMIAL ...  1ST FUNCTION                         CINP0530
C        INPUT CARD E.3                                                 CINP0540
C                                                                       CINP0550
   20 J2 = J1+5                                                         CINP0560
      READ(5,15)(TAB(J),J = J1,J2)                                      CINP0570
      WRITE(6,21)  (TAB(J),J = J1,J2)                                   CINP0580
   21 FORMAT(7X,'FIRST PART OF FUNCTION - 5TH DEGREE POLYNOMIAL'//      CINP0590
     *         8X,'A0',13X,'A1',13X,'A2',13X,'A3',13X,'A4',13X,'A5',13X/CINP0600
     *         6F15.6//)                                                CINP0610
      J1 = J2+1                                                         CINP0620
      GO TO 25                                                          CINP0630
C                                                                       CINP0640
C         TABLE LOAD ...  1ST FUNCTION                                  CINP0650
C         INPUT CARDS E.4.A-E.4.N                                       CINP0660
C                                                                       CINP0670
   22 READ(5,23)  NPI                                                   CINP0680
   23 FORMAT (12I6)                                                     CINP0690
      TAB(J1) = NPI                                                     CINP0700
      J1 = J1+1                                                         CINP0710
      J2 = J1+2*NPI-1                                                   CINP0720
      READ(5,15)(TAB(J),J = J1,J2)                                      CINP0730
      WRITE (6,24)  NPI, (TAB(J)    ,J = J1, J2)                        CINP0740
   24 FORMAT(7X,'FIRST PART OF FUNCTION - ',I4,' TABULAR POINTS'//      CINP0750
     *         8X,'D',16X,'F(D)'  /(F15.6,F15.4))                       CINP0760
      J1 = J2+1                                                         CINP0770
C                                                                       CINP0780
C        CHECK FOR SECOND FUNCTION                                      CINP0790
C                                                                       CINP0800
   25 IF(D2)  28,11,26                                                  CINP0810
C                                                                       CINP0820
C        SECOND FUNCTION ...  5TH ORDER POLYNOMIAL                      CINP0830
C        INPUT CARD E.3                                                 CINP0840
C                                                                       CINP0850
   26 J2 = J1+5                                                         CINP0860
      READ(5,15)(TAB(J),J = J1,J2)                                      CINP0870
      WRITE (6,27)  (TAB(J),J = J1,J2)                                  CINP0880
   27 FORMAT(7X,'SECOND PART OF FUNCTION - 5TH DEGREE POLYNOMIAL'//     CINP0890
     *         8X,'B0',13X,'B1',13X,'B2',13X,'B3',13X,'B4',13X,'B5',13X/CINP0900
     *         6F15.6//)                                                CINP0910
      J1 = J2+1                                                         CINP0920
      GO TO 11                                                          CINP0930
C                                                                       CINP0940
C        SECOND FUNCTION ...  TABLE LOAD                                CINP0950
C        INPUT CARDS E.4.A-E.4.N                                        CINP0960
C                                                                       CINP0970
   28 READ(5,23)  NPI                                                   CINP0980
      TAB(J1) = NPI                                                     CINP0990
      J1 = J1+1                                                         CINP1000
```

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

```
      SUBROUTINE CONTCT
C                                                       REV 12 12/19/74
C
C     CONTROLS THE CALLING OF SUBROUTINES REQUIRED TO COMPUTE THOSE
C     EXTERNAL FORCES AND TORQUES ACTING ON THE BODY SEGMENTS.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/CONTRL/NSEG,NJNT,NS3,NJ3,NPL,NBLT,NBAG,NVEH,NGRND,NPRT(40)
      COMMON/JBARTZ/ MNPL(  20),MNBLT(  8),MNSEG(  22),MNBAG(   6),
     *              MPL(3,5,20),MBLT(3,5,8),MSEG(3,5,22),MBAG(3,10,6),
     *                NTPL(5,20),NTBLT(5,8),NTSEG(5,22)
      COMMON/FORCES/PSF(7,20),BSF(4,20),SSF(10,20),BAGSF(3,20),
     *               NPSF,NBSF,NSSF,NBGSF,NPANEL(6),PRJNT(6,21)
      COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NT1(50),NTAB(500),TAB(2000)
      COMMON/KALEPS/WTIME(30),IWIND(30),MWSEG(5,22)
      COMMON/HRNESS/ BAR(6,100) , XLONG(20), IBAR(2,100), NTHRNS(20),
     *                NHRNSS, NBLTPH(5), NFBLT(5,20), NPTSPB(20)
      CALL ELTIME(1,12)
      NPSF = 0
      NBSF = 0
      NSSF = 0
      1F (NPL.LE.0) GO TO 21
C
C     CALL PLELP ROUTINE FOR EACH ALLOWED PLANE-SEGMENT CONTACT.
C
      DO 20 J=1,NPL
      IF(MNPL(J).EQ.0) GO TO 20
      KPL = MNPL(J)
      DO 19 I=1,KPL
      NPSF = NPSF+1
      M1 = MPL(1,1,J)
      M2 = MPL(2,1,J)
      M3 = MPL(3,I,J)
      NT = NTPL(I,J)
      JT = NTAB(NT)
      TAB(JT) = 0.0
   19 CALL PLELP(M2,M3,M1,J,NT)
   20 CONTINUE
   21 1F(NBLT.LE.0) GO TO 41
C
C     CALL BELTRT ROUTINE FOR EACH ALLOWED BELT-SEGMENT CONTACT.
C
      DO 30 J=1,NBLT
      1F(MNBLT(J).EQ.0) GO TO 30
      KBLT = MNBLT(J)
      DO 29 I=1,KBLT
      NBSF = NBSF+1
      M1 = MBLT(1,I,J)
      M2 = MBLT(2,1,J)
      M3 = MBLT(3,I,J)
      NT = NTBLT(1,J)
```

```
      JT = NTAB(NT)
      TAB(JT) = 0.0
      NF = NTAB(NT+5)
      IF (NF.NE.0) JT = NTAB(NT+6)
      IF (NF.NE.0) TAB(JT) = 0.0
   29 CALL BELTRT(M2,M3,M1,J,NT)
   30 CONTINUE
C
C     CALL SEGSEG ROUTINE FOR EACH ALLOWED SEGMENT-SEGMENT CONTACT.
C
   41 DO 50 J=1,NSEG
      IF(MNSEG(J).EQ.0) GO TO 50
      KSEG = MNSEG(J)
      DO 49 I=1,KSEG
      NSSF = NSSF+1
      M1 = MSEG(1,I,J)
      M2 = MSEG(2,I,J)
      M3 = MSEG(3,I,J)
      NT = NTSEG(I,J)
      JT = NTAB(NT)
      TAB(JT) = 0.0
   49 CALL SEGSEG(J,M1,M2,M3,NT)
   50 CONTINUE
C
C     CALL AIRBAG ROUTINE FOR ALLOWED BAG-SEGMENT CONTACTS, IF ANY.
C
      IF (NBAG.NE.0) CALL AIRBAG
C
C     CALL WINDY ROUTINE FOR WIND FORCES ON EACH SEGMENT.
C
      DO 60 J=1,NSEG
      IF (MWSEG(1,J).EQ.0) GO TO 60
      M1 = MWSEG(2,J)
      M2 = MWSEG(3,J)
      M3 = MWSEG(4,J)
      NT = MWSEG(5,J)
      CALL WINDY (J,M1,M2,M3,NT)
   60 CONTINUE
C
C     CALL HBELT ROUTINE FOR EACH HARNESS-BELT SYSTEM.
C
      IF (NHRNSS.LE.0) GO TO 99
      J1 = 1
      K1 = 1
      DO 70 I=1,NHRNSS
      IF (NBLTPH(I).LE.0) GO TO 70
      J2 = J1 + NBLTPH(I) -1
      DO 69 J=J1,J2
      IF (NPTSPB(J).LE.0) GO TO 69
      K2 = K1 + NPTSPB(J) -1
```

```
      CALL HBELT(NPTSPB(J),IBAR(1,K1),BAR(1,K1),NTHRNS(J),XLONG(J))
      K1 = K2+1
   69 CONTINUE
      J1 = J2+1
   70 CONTINUE
   99 CALL ELTIME(2,12)
      RETURN
      END
```

```
      SUBROUTINE DINT(IN,N,DTPR,H0,HMAX,HMIN,/T/,X,DER,NOINT)
C                                                 REV 12 10/25/74
C     EXECUTIVE ROUTINE USED FOR PERFORMING AN INTEGRATION
C     STEP BETWEEN PRINT TIME POINTS.
C
C     ARGUMENTS
C         IN: INTEGRATION STEP NUMBER
C          N: NO OF VARIABLES TO BE SUPPLIED AS INPUT TO ROUTINE
C             OR COMPUTED BY SUBROUTINE PDAUX WHEN K=0 (MAX=120).
C       DTPR: PRINT TIME INTERVAL DESIRED
C         H0: INITIAL INTEGRATION STEP SIZE
C       HMAX: MAXIMUM INTEGRATION STEP SIZE
C       HMIN: MINIMUM STEP SIZE
C          T: TIME
C          X: ARRAY OF STATE VARIABLES
C        DER: ARRAY OF DERIVATIVES OF STATE VARIABLES
C      NOINT: NUMBER OF ESTIMATES OF INTEGRATION PARAMETERS
C             TO BE MADE AT THE END OF ANY INTERMEDIATE TIME STEP.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/CONTRL/ NSEG,NJNT,NS3,NJ3,NPL,NBLT,NBAG,NVEH,NGRND,NPRT(40)
      COMMON/INTEST/ SGTEST(3,4,22),XTEST(264)
      COMMON/CNSNTS/ PI, RADIAN,G,THIRD,EPS1,EPS4,EPS6,EPS8,
     *               EPS12,EPS15,EPS20,EPS24,UNITL,UNITM,UNITT,GRAVTY(3)
      COMMON/COINT/ E(3,120),F(5,120),GG(5,120),Y(5,120),U(5,120)
     *             ,H,HPRINT,TSAVE,TPRINT,TSTART,ICNT,IDBL,IFLAG
      DIMENSION      X(120),DER(120)
C
C
      CALL ELTIME(1,3)
       IF (IN.NE.0) GO TO 3
C
C     FIRST TIME IN ROUTINE, PERFORM INITIALIZATION STEP .
C
      H = H0
      HPRINT = H
       IDBL=3
      ICNT = 0
      TPRINT = T
      CALL OUTPUT(0)
      K = 0
      CALL PDAUX(X,DER,N,K)
      IF (N.GT.120) WRITE (6,9) N
    9 FORMAT('0 NUMBER OF VARIABLES IN SUBROUTINE DINT IS',I6,
     *       ' AND EXCEEDS THE ARRAY SIZES OF 120. PROGRAM TERMINATED.')
      IF (N.GT.120) STOP
      DO 1  I=1,N
      F(1,I) = X(I)
      F(2,I) = DER(I)
      F(3,I) = 0.
```

```
      F(4,I) = 0.
    1 F(5,I) = 0.
      GO TO 65
C
C     START OF NEW PRINT POINT INTERVAL.
C
    3 TPRINT = TPRINT+DTPR
      H = HPRINT
C
C     ENTRY TO ADVANCE INTEGRATOR.
C
    4 K = 1
      CALL UPDATE(K)
      IF (K.EQ.1) GO TO 2
C
C     RECALL PDAUX FOR IMPULSE IF K = -1
C
      IF (NPRT(26).NE.0) CALL OUTPUT(0)
      CALL PDAUX(X,DER,N,K)
      IF (NPRT(26).NE.0) CALL OUTPUT(1)
      H = HO
      ICNT = 0
      K = 1
      DO 6 I=1,N
      F(1,I) = X(I)
      F(2,I) = DER(I)
      F(3,I) = 0.0
      F(4,I) = 0.0
    6 F(5,I) = 0.0
    2 HPRINT = H
      IF (T+H+EPS8.GE.TPRINT) H = TPRINT-T
C
C     ENTRY TO BACKUP INTEGRATOR, CONVERGENCE TEST FAILED.
C
    5  D1 = 0.5*H
       D12=D1+D1
       D123=H-D1
       TSTART=T
       T=TSTART+D1
      DO 10  I=1,N
      DO 10  J=1,5
       U(J,I)=0.
       Y(J,I)=0.
   10 GG(J,I) = F(J,I)
       CALL DZP(N,X,GG,E,D1,1)
      IF (NPRT(26).NE.0) CALL OUTPUT(0)
      CALL PDAUX(X,DER,N,K)
      IF (NPRT(26).NE.0) CALL OUTPUT(1)
      DO 20  I=1,N
       W=X(I)-GG(1,I)
```

```
      Z=DER(I)-GG(2,I)
       Y(1,I)=Y(1,I)+W
       Y(2,I)=Y(2,I)+Z
       Y(3,I)=Y(3,I)+W**2
       Y(4,I)=Y(4,I)+Z*W
   20  GG(2,I)=DER(I)
       CALL DZP(N,X,GG,E,D1,0)
      K = 2
      IF (NPRT(26).NE.0) CALL OUTPUT(0)
      CALL PDAUX(X,DER,N,K)
      IF (NPRT(26).NE.0) CALL OUTPUT(1)
       T=TSTART+H
      H1 = EPS1/H
      DO 30  I=1,N
       GG(2,I)=F(2,I)
       W=X(I)-GG(1,I)
       Z=DER(I)-GG(2,I)
       Y(1,I)=Y(1,I)+W
       Y(2,I)=Y(2,I)+Z
       Y(3,I)=Y(3,I)+W**2
       Y(4,I)=Y(4,I)+Z*W
        Y(5,I)=Y(3,I)-.5*Y(1,I)**2
       U(5,I)=Y(4,I)-.5*Y(1,I)*Y(2,I)
       Z=0.
        IF(Y(5,I).NE.0.)Z=U(5,I)/Y(5,I)
      IF (Z.GT.H1) Z = H1
       GG(5,I)=Z
      ZYZ = (Y(2,I)-Z*Y(1,I))/D12
      GG(4,I) = 0.5*GG(4,I)
   30 GG(3,I) = ZYZ - D1*GG(4,I)
       CALL DZP(N,X,GG,E,H ,1)
      K = 3
      IF (NPRT(26).NE.0) CALL OUTPUT(0)
      CALL PDAUX(X,DER,N,K)
      IF (NPRT(26).NE.0) CALL OUTPUT(1)
      DO 44 L=1,NDINT
       ZL=L
       ZH=ZL*H
      DO 40  I=1,N
       W=X(I)-GG(1,I)
       Z=DER(I)-GG(2,I)
      IF (DABS(W).LT.EPS24) W=0.0
      IF (DABS(Z).LT.EPS24) Z=0.0
       U(1,I)=U(1,I)+W
       U(2,I)=U(2,I)+Z
       U(3,I)=U(3,I)+W**2
       U(4,I)=U(4,I)+W*Z
       Z=GG(5,I)
       IF(L.EQ.1)GO TO 35
       Z=0.
```

```
      FX=Y(5,I)+U(3,I)-U(1,I)**2/ZL
      IF(FX.NE.0.)Z=(U(5,I)+U(4,I)-U(1,I)*U(2,I)/ZL)/FX
      IF (Z.GT.H1) Z = H1
   35 GG(5,I)=Z
      W=(Y(2,I)-Z*Y(1,I))/D12
      Z=(U(2,I)-Z*U(1,I))/ZH
      GG(3,I)=(H*W-D1*Z)/D123
   40 GG(4,I)=(Z-W)/D123
      M=1
      IF(L.EQ.1)M=0
      CALL DZP(N,X,GG,E,H,M)
      IF (L.EQ.NDINT.OR.NPRT(26).NE.0) CALL OUTPUT(0)
      IF (L.EQ.NDINT) K = 4
      CALL PDAUX(X,DER,M,K)
   44 IF (L.NE.NDINT.AND.NPRT(26).NE.0) CALL OUTPUT(1)
C
C     TEST FOR CONVERGENCE
C
      IF (K.LT.0) GO TO 47
      DO 46 II=1,N,3
      IF (XTEST(II).LE.0.0) GO TO 46
      TE = 0.0
      TT = 0.0
      I2 = II+2
      DO 45 I=II,I2
      Z=GG(5,I)*(X(I)-GG(1,I))+GG(2,I)+H*(GG(3,I)+H*GG(4,I))
      TE = TE+(DER(I)-Z)**2
   45 TT = TT+DER(I)**2
      IF (NPRT(25).NE.0) WRITE (6,48) T,II,TT,TE,(XTEST(I),I=II,I2)
      IF (TT.LT.XTEST(II)) GO TO 46
      IF (XTEST(II+1).GT.0.0 .AND. TE.LT.XTEST(II+1)) GO TO 46
      IF (TE.GE.XTEST(II+2)*TT) GO TO 47
   46 CONTINUE
C
C     CONVERGENCE SUCCESSFUL
C
      GO TO 60
C
C     CONVERGENCE FAILED, TEST TO DIVIDE H.
C
   47 IF (NPRT(25).EQ.0) WRITE (6,48) T,II,TT,TE,(XTEST(I),I=II,I2)
   48 FORMAT('0 DINT CONV. TEST',F10.6,I6,5G16.8)
      WRITE (6,49) T,H
   49 FORMAT('0 TEST FAILED AT TIME =',F10.6,' FOR H =',F10.6)
      ICNT = 0
      IF (H.LE.HMIN) GO TO 61
      IF (NPRT(26).NE.0) CALL OUTPUT(1)
      T = T-H
      H = H*0.5
      K = 2
```

```
      GO TO 5                                                           DINT2010
   60 IF (H.GT.0.74*HPRINT) ICNT = ICNT+1                               DINT2020
   61 DO 63 I=1,N                                                       DINT2030
      F(1,I) = X(I)                                                     DINT2040
      F(2,I) = DER(I)                                                   DINT2050
      F(3,I) = GG(3,I) +2.0*H*GG(4,I)                                   DINT2060
      F(4,I) = GG(4,I)                                                  DINT2070
   63 F(5,I) = GG(5,I)                                                  DINT2080
      IF(ICNT.LT.IDBL) GO TO 65                                         DINT2090
      ICNT = 0                                                          DINT2100
      H = H*2.0                                                         DINT2110
      IF (H.GT.HMAX) H=HMAX                                             DINT2120
      HPRINT = 2.0*HPRINT                                               DINT2130
      IF (HPRINT.GT.HMAX) HPRINT = HMAX                                 DINT2140
   65 CALL UPDATE(2)                                                    DINT2150
      CALL OUTPUT(1)                                                    DINT2160
      IF (TPRINT-T.GE.EPS8) GO TO 4                                     DINT2170
      CALL ELTIME(2,3)                                                  DINT2180
      RETURN                                                            DINT2190
      END                                                               DINT2200
```

```
      SUBROUTINE ELTIME(L,N)                                            ELTI0010
C                                                  REV 12 12/19/74ELTI0020
C     COUNTS THE NUMBER OF TIMES CERTAIN BASIC SUBROUTINES ARE CALLED   ELTI0030
C     AND ACCOUNTS FOR ALL COMPUTER CPU TIME USED BY THESE ROUTINES.    ELTI0040
C                                                                       ELTI0050
C       ARGUMENTS L: 1 INDICATES CALL IS AT START OF ROUTINE            ELTI0060
C                    2 INDICATES CALL IS AT END OF ROUTINE.             ELTI0070
C                 N: THE SUBROUTINE IDENTIFICATION NUMBER.              ELTI0080
C                                                                       ELTI0090
C       ASSUMES FUNCTION LTIME(I) IS GIVING ELAPSED CPU TIME IN INTEGER ELTI0100
C       UNITS OF 0.01 SECONDS SINCE FUNCTION LTIME(0) WAS CALLED.       ELTI0110
C                                                                       ELTI0120
      COMMON/GBTIME/NT(40),MTIN(40),NC(40),IND(40),NSUB                 ELTI0130
      REAL*8 SUB(40)/                                                   ELTI0140
     *   8H MAIN3D ,8H INPUT  ,8H DINT   ,8H PRIPLT ,8H DZP    ,        ELTI0150
     *   8H PDAUX  ,8H UPDATE ,8H OUTPUT ,8H DAUX   ,8H SETUP1 ,        ELTI0160
     *   8H CHAIN  ,8H CONTCT ,8H VISPR  ,8H DAUX11 ,8H DAUX12 ,        ELTI0170
     *   8H DAUX22 ,8H DAUX31 ,8H DAUX32 ,8H DAUX33 ,8H FSMSOL ,        ELTI0180
     *   8H PLELP  ,8H BELTRT ,8H SEGSEG ,8H AIRBAG ,8H RSTART ,        ELTI0190
     *   8H SETUP2 ,8H IMPULS ,8H IMPLS2 ,8H AIRBG3 ,8H DAUX55 ,        ELTI0200
     *   8H EJOINT ,8H SPDAMP ,8H DAUX44 ,8H FLXSEG ,8H WINDY  ,        ELTI0210
     *   8H HBELT  ,8H        ,8H        ,8H        ,8H        /        ELTI0220
      IF (N.GT.1) GO TO 20                                              ELTI0230
      IF (L.GT.1) GO TO 40                                              ELTI0240
C                                                                       ELTI0250
C     INITIAL CALL AT BEGINNING OF MAIN PROGRAM.                        ELTI0260
C                                                                       ELTI0270
      MTIN(1) = LTIME(0)                                                ELTI0280
      DO 11 I=1,40                                                      ELTI0290
      IND(I)  =  0                                                      ELTI0300
      NC(I)   =  0                                                      ELTI0310
      MTIN(I) = -1                                                      ELTI0320
   11 NT(I)   =  0                                                      ELTI0330
      NSUB    =  1                                                      ELTI0340
      IND(1)  =  1                                                      ELTI0350
      NC(1)   =  1                                                      ELTI0360
      MTIN(1) =  0                                                      ELTI0370
      GO TO 99                                                          ELTI0380
C                                                                       ELTI0390
C      CALL AT BEGINNING OF NTH SUBROUTINE.                             ELTI0400
C                                                                       ELTI0410
   20 IF (L.GT.1) GO TO 30                                              ELTI0420
      MTIN(N) = LTIME(1)                                                ELTI0430
      IF (NC(N).NE.0) GO TO 21                                          ELTI0440
      NSUB = NSUB+1                                                     ELTI0450
      IND(NSUB) = N                                                     ELTI0460
   21 NC(N) = NC(N)+1                                                   ELTI0470
      GO TO 99                                                          ELTI0480
C                                                                       ELTI0490
C     CALL AT END OF NTH SUBROUTINE.                                    ELTI0500
```

```
C
   30 MTOUT = LTIME(1)
      NDIFF = MTOUT-MTIN(N)
      MTIN(N) = -1
      IF (NDIFF.EQ.0) GO TO 32
      NT(N) = NT(N) + NDIFF
      DO 31 I=1,40
      IF (MTIN(I).NE.-1) MTIN(1) = MTIN(I) + NDIFF
   31 CONTINUE
   32 GO TO 99
C
C     SUBSEQUENT CALLS FROM MAIN PROGRAM, PRINT SUMMARY TABLE.
C
   40 NTSUM = LTIME(1)
      NT(1) = NTSUM - MTIN(1)
      TIME = FLOAT(NTSUM)/100.0
      WRITE (6,41) TIME
   41 FORMAT('1 ELAPSED CPU TIME =',F10.2,' SECONDS'//
     *    '     SUB          CALLS       TIME        %    '//)
      PCSUM = 0.0
      NTSUM = 0
      DO 42 I=1,NSUB
      J = IND(I)
      PC = FLOAT(NT(J))/TIME
      PCSUM = PCSUM + PC
      NTSUM = NTSUM + NT(J)
   42 WRITE (6,43) SUB(J),NC(J),NT(J),PC
   43 FORMAT(A10,2I10,F10.2)
      WRITE (6,44) NTSUM,PCSUM
   44 FORMAT('0TOTAL',14X,I10,F10.2)
   99 RETURN
      END
```

```
      DOUBLE PRECISION FUNCTION EVALFD (D,N,L)
C                                                     REV 10 09/26/74
C     EVALUATE FUNCTION THAT IS DEFINED AT LOCATION N OF TAB ARRAY
C     FOR ABSCISSA VALUE D.  EVALUATES DERIVATIVE, FUNCTION OR INTEGRAL
C     AS L EQUALS 0, 1, OR 2. TAB ARRAY IS DEFINED AS FOLLOWS:
C        TAB(N)     -  DO (DO MUST BE NON-NEGATIVE)
C        TAB(N+1)   -  D1 (F1 DEFINED FOR DO < D < D1)
C        TAB(N+2)   -  D2 (F2 DEFINED FOR D1 < D < D2)
C        TAB(N+3)   -  (NOT CURRENTLY USED)
C        TAB(N+4)   -  (NOT CURRENTLY USED)
C        TAB(N+5)   -  START OF DEFINITION OF 1ST PART OF FUNCTION (F1)
C        WHICH IS FOLLOWED BY DEFINITION OF 2ND PART OF FUNCTION (F2),
C                                       IF ANY.
C        2ND PART OF FUNCTION EXISTS IF D2 IS NON-ZERO.
C        SIGN OF D1 DETERMINES FORM OF DEFINITION FOR 1ST PART OF
C                                     THE FUNCTION.
C
C              D1 ZERO INDICATES THAT FUNCTION IS CONSTANT D2 FOR ALL D.
C
C              D1 POSITIVE INDICATES THAT TAB(N+5)-TAB(N+10) CONTAINS
C              A0,A1,...A5.   THE COEFFICIENTS OF A 5TH ORDER POLYNOMIAL.
C
C              D1 NEGATIVE INDICATES THAT TAB(N+5) CONTAINS NP (REAL)
C              FOLLOWED BY   D(1), F(1), D(2), F(2) ..., D(NP), F(NP)
C
C              WARNING- TABULAR FUNCTION MUST BE DEFINED FOR WHOLE RANGE,
C              THAT IS, FROM DO TO D1 INCLUSIVE,OR D1 TO D2 INCLUSIVE.
C
C
C        SIMILARLY, THE SIGN OF D2 (IF NON-ZERO) DETERMINES FORM OF
C        DEFINITION OF 2ND PART OF FUNCTION, IF ANY.
C
C
C     IF D < DO                  FUNCTION = 0
C     IF D > |D1|  AND  D2=0     FUNCTION = F1(|D1|)
C     IF D > |D2|  AND  D2#0     FUNCTION = F2(|D2|)
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(500),TAB(2000)
      F = 0.0
      IOUTR = 0
      DO = TAB(N)
      IF (D.LT.DO) GO TO 40
      D1 = TAB(N+1)
      D2 = TAB(N+2)
      IF (D1.NE.0.0)  GO TO 26
      IF (L-1) 40,24,25
   24 F = D2
      GO TO 40
   25 F= (D-DO)*D2
```

```
      GO TO 40
C
C     COMPUTE INDEX OF F1 DEFINITION
C
   26 NP = N+5
      IF (L.EQ.2) GO TO 41
C
C    DERIVATIVES AND FUNCTIONS HERE, INTEGRALS HAVE OTHER LOGIC
C
      IF (D.LT.DABS(D1)) GO TO 31
      IF (D2.NE.0.0) GO TO 32
C
C     D .GE.|D1| , D2 = 0
C
  30  IF (D1.LE.0.0) GO TO 33
C
C     IOUTR.EQ.1 INDICATES D BEYOND RANGE. DERIVATIVE = 0.
C     IOUTR.EQ.0 INDICATES D.LE.|D1|. COMPUTE POLY DERIVATIVE
C
      IF (D.GT.DABS(D1)) IOUTR = 1
      X = D1
      GO TO 37
C
C     D0 < D < |D1|
C
   31 IF (D1.LT.0.0) GO TO 35
      X = D
      GO TO 37
C
C     D .GE. |D1|, D2 NON-ZERO, USE F2
C
   32 MP = 6
C
C     COMPUTE INDEX OF F2 DEFINITION
C
      IF (D1.LT.0.0) MP = 2.0 * TAB(NP)+1.0
      NP = NP+MP
      IF (D.LT.DABS(D2)) GO TO 34
  29  IF (D2.LT.0.0) GO TO 33
C
C     IOUTR.EQ.1 INDICATES D BEYOND RANGE. DERIVATIVE = 0.
C     IOUTR.EQ.C INDICATES D.LE.|D2|. COMPUTE POLY DERIVATIVE
C
      IF (D.GT.DABS(D2)) IOUTR = 1
C
C     D .GE. D2 (POSITIVE), EVALUATE F2 FOR D2
C
      X = D2
      GO TO 37
C
```

```
C      D EXCEEDS TABULAR DEFINITION, SET F = F(NP)                      EVAL1010
C      IF TABLE DEFINITION EXTENDS BEYOND RANGE, USE TABLE VALUES       EVAL1020
C                                                                       EVAL1030
   33 MB = TAB(NP)                                                      EVAL1040
      NB = NP+MB+MB                                                     EVAL1050
      IF (D .LE. TAB(NB-1)) GO TO 35                                    EVAL1060
      IF (L.EQ.1) F=TAB(NB)                                             EVAL1070
      GO TO 40                                                          EVAL1080
C                                                                       EVAL1090
C      |D1| .LE. D < |D2|                                               EVAL1100
C                                                                       EVAL1110
   34 IF (D2.LT.0.0) GO TO 35                                           EVAL1120
      X = D                                                             EVAL1130
      GO TO 37                                                          EVAL1140
C                                                                       EVAL1150
C      EVALUATE F FROM TABULAR DEFINITION                               EVAL1160
C                                                                       EVAL1170
   35 MB = TAB(NP)                                                      EVAL1180
      K1 = NP+3                                                         EVAL1190
      K2 = NP+MB+MB                                                     EVAL1200
      DO 36 K=K1,K2,2                                                   EVAL1210
      IF (D.GT.TAB(K)) GO TO 36                                         EVAL1220
      IF (L-1) 28,27,40                                                 EVAL1230
C                                                                       EVAL1240
C      EVALUATE DERIVATIVE FROM TABLE                                   EVAL1250
C                                                                       EVAL1260
  28   F = (TAB(K+1)-TAB(K-1))/(TAB(K)-TAB(K-2))                        EVAL1270
      GO TO 40                                                          EVAL1280
C                                                                       EVAL1290
C      EVALUATE FUNCTION FROM TABLE                                     EVAL1300
C                                                                       EVAL1310
  27  R2 = TAB(K)-TAB(K-2)                                              EVAL1320
      R1 = (D-TAB(K-2))/R2                                              EVAL1330
      R2 = (TAB(K)-D )/R2                                               EVAL1340
      F = R1*TAB(K+1)+R2*TAB(K-1)                                       EVAL1350
      GO TO 40                                                          EVAL1360
   36 CONTINUE                                                          EVAL1370
      IF (L.EQ.1) F = TAB(K2)                                           EVAL1380
      GO TO 40                                                          EVAL1390
   37 IF (IOUTR.EQ.1 .AND. L.EQ.0 ) GO TO 40                            EVAL1400
      IF (L-1) 38,39,40                                                 EVAL1410
C                                                                       EVAL1420
C      EVALUATE DERIVATIVE OF 5TH DEGREE POLYNOMIAL                     EVAL1430
C                                                                       EVAL1440
  38  F = TAB(NP+1)+X*(2.0*TAB(NP+2)+X*(3.0*TAB(NP+3)+X*(4.0*TAB(NP+4)+ EVAL1450
     *        X*5.0*TAB(NP+5))))                                        EVAL1460
      GO TO 40                                                          EVAL1470
C                                                                       EVAL1480
C      EVALUATE 5TH DEGREE POLYNOMIAL                                   EVAL1490
C                                                                       EVAL1500
```

```
   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
C
C     L=2: COMPUTE INTEGRAL OF FUNCTION FROM DO TO D.
C
   41 IF (D.EQ.DO) GO TO 40
      XO = DO
      X1 = D1
      DO 50 I=1,2
      IF (X1) 43,49,42
   42 A0 = TAB(NP   )
      A1 = TAB(NP+1)/2.0
      A2 = TAB(NP+2)/3.0
      A3 = TAB(NP+3)/4.0
      A4 = TAB(NP+4)/5.0
      A5 = TAB(NP+5)/6.0
      NP = NP+6
      X = XO
      IF (X.NE.0.0) F=F-X*(A0+X*(A1+X*(A2+X*(A3+X*(A4+X*A5)))))
      X = DMIN1(D,X1)
      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
      IF(I.EQ.1.AND.D2.NE.0.0)  GO TO 49
C
C     NOTE -  NP WAS UPDATED NP=NP+6 BEFORE THIS, READY FOR SECOND PASS
C
      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
   43 MB = TAB(NP)
      K1 = NP+3
      K2 = NP+MB+MB
      NP = K2+1
      DL = DMIN1(D,DABS(X1))
      DO 44 K=K1,K2,2
      IF (XO.GE.TAB(K)) GO TO 44
      Z1 = DMAX1(XO,TAB(K-2))
      Z2 = DMIN1(DL,TAB(K))
      FYX = TAB(K-1)*TAB(K) - TAB(K+1)*TAB(K-2)
      FY = TAB(K+1) - TAB(K-1)
      F = F +(FYX  + 0.5*FY*(Z1+Z2)) *(Z2-Z1)/ (TAB(K)-TAB(K-2))
      IF (Z2.NE.DL) GO TO 44
      IF(I.EQ.1.AND.D2.NE.0.0)  GO TO 49
      IF(Z2. EQ. D) GO TO 40
      F = F +(D-Z2)*(FYX+Z2*FY)/ (TAB(K)-TAB(K-2))
      GO TO 40
   44 CONTINUE
   49 XO = DABS(D1)
   50 X1 = D2
```

```
40 EVALFD = F                                                     EVAL2010
   RETURN                                                         EVAL2020
   END                                                            EVAL2030
```

```
      SUBROUTINE FINPUT
C                                                  REV 12 12/18/74
C     CONTROLS CARD INPUT SPECIFYING THE ALLOWED CONTACTS OF THE CRASH
C     VICTIM BODY SEGMENTS WITH VEHICLE PANELS, BELTS, AIRBAGS AND OTHER
C     BODY SEGMENTS ALONG WITH THE ASSOCIATED FUNCTIONS TO BE USED FOR
C     EACH CONTACT.
C     ALSO SETS UP TABLES TO CONTROL TIME HISTORY INFORMATION FOR
C     EACH FUNCTION FOR EACH ALLOWED CONTACT.
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/CONTRL/NSEG,NJNT,NS3,NJ3,NPL,NBLT,NBAG,NVEH,NGRND,NPRT(40)
      COMMON/JBARTZ/ MNPL(  20),MNBLT(  8),MNSEG(  22),MNBAG(   6),
     *             MPL(3,5,20),MBLT(3,5,8),MSEG(3,5,22),MBAG(3,10,6),
     *               NTPL(5,20),NTBLT(5,8),NTSEG(5,22)
      COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(500),TAB(2000)
      COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),BLTTTL(5,8)
     *              ,PLTTL(5,20),BAGTTL(5,6),SEG(22),JOINT(21)
     *              ,CGS(21),JS(21)
      REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT
      LOGICAL*1 CGS,JS
      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)
     *             ,NQ,KQ1(12),KQ2(12),KQTYPE(12)
      COMMON/DESCRP/ PHI(3,22),W(22),SR(3,42),HA(3,42),HB(3,42)
     *              ,HT(3,3,42),RPHI(3,22),RW(22),SPRING(5,63)
     *              ,VISC(7,63),JNT(21),IPIN(21),MS,ISING(22)
     *              ,IGLOB(21),JOINTF(21)
      COMMON/CEULER/ IEULER(22),HIR(3,3,21),ANG(3,21),ANGD(3,21),
     *               FE(3,21),TQE(3,31),CONST(3,21)
      COMMON/KALEPS/WTIME(30),IWIND(30),MWSEG(5,22)
      COMMON/TEMPVS/JTITLE(5,51),NF(5),NS(3),KTITLE(31)
      REAL BLANK /'    '/,JTITLE,KTITLE
      REAL SURFCE(2,3)/'  PL','ANE ','  BE','LT  ',' SEG','MENT'/
      MXNTI = 50
      J1 = MXTB1+1
C
C     INPUT ALLOWED CONTACTS AND FUNCTIONS BY REF. NO.
C
      NT = 1
      WRITE (6,31)
   31 FORMAT('1 ALLOWED CONTACTS AND ASSOCIATED FUNCTIONS')
      DO 61 I=1,4
      IJK = 0
      GO TO (32,34,35,36),I
   32 IF (NPL.LE.0) GO TO 61
C
C     INPUT NO. OF SEGMENTS TO CONTACT EACH PLANE.
C     INPUT CARD F.1.A
C
```

```
      READ  (5,33) (MNPL(J),J=1,NPL)                                    FINP0510
   33 FORMAT(18I4)                                                      FINP0520
      NJJ = NPL                                                         FINP0530
      GO TO 37                                                          FINP0540
   34 IF (NBLT.LE.0) GO TO 61                                           FINP0550
C                                                                       FINP0560
C     INPUT NO. OF SEGMENTS TO CONTACT EACH BELT.                       FINP0570
C     INPUT CARD F.2.A                                                  FINP0580
C                                                                       FINP0590
      READ  (5,33) (MNBLT(J),J=1,NBLT)                                  FINP0600
      NJJ = NBLT                                                        FINP0610
      GO TO 37                                                          FINP0620
   35 IF (NSEG.LE.0) GO TO 61                                           FINP0630
C                                                                       FINP0640
C     INPUT NO. OF SEGMENTS TO CONTACT EACH SEGMENT.                    FINP0650
C     INPUT CARD F.3.A                                                  FINP0660
C                                                                       FINP0670
      READ  (5,33) (MNSEG(J),J=1,NSEG)                                  FINP0680
      NJJ = NSEG                                                        FINP0690
      GO TO 37                                                          FINP0700
   36  IF (NJNT.LE.0) GO TO 61                                          FINP0710
C                                                                       FINP0720
C     INPUT CARD F.4.A                                                  FINP0730
C     SUPPLY IGLOB(J)=1 FOR EACH GLOBALGRAPHIC JOINT J=1,NJNT           FINP0740
C                                                                       FINP0750
       READ (5,33) (IGLOB(J),J=1,NJNT)                                  FINP0760
      NJJ = NJNT                                                        FINP0770
C                                                                       FINP0780
C     START OF LOOP TO READ CONTACTS FOR PLANES (I=1), BELTS (I=2),     FINP0790
C     SEGMENTS (I=3) AND FUNCTIONS FOR GLOBALGRAPHIC JOINTS (I=4).      FINP0800
C                                                                       FINP0810
   37 DO 60 J=1,NJJ                                                     FINP0820
      IF (I.EQ.1) NK = MNPL(J)                                          FINP0830
      IF (I.EQ.2) NK = MNBLT(J)                                         FINP0840
      IF (I.EQ.3) NK = MNSEG(J)                                         FINP0850
      IF (I.EQ.4) NK = IGLOB(J)                                         FINP0860
      IF (NK.LE.0) GO TO 60                                             FINP0870
      DO 59 K=1,NK                                                      FINP0880
      IF (IJK.EQ.0) WRITE (6,38) I                                      FINP0890
   38 FORMAT('0',119X,'CARDS F.',I1)                                    FINP0900
      IF (IJK.EQ.0 .AND. I.NE.4) WRITE (6,39) SURFCE(1,I),SURFCE(2,I)   FINP0910
   39 FORMAT('0',3X,2A4,8X,'SEGMENT',2X,'FORCE DEFLECTION',6X,'INERTIAL FINP0920
     *SPIKE',10X,'R FACTOR',13X,'G FACTOR',10X,'FRICTION COEF.')        FINP0930
      IF (IJK.EQ.0 .AND. I.EQ.4) WRITE (6,40)                           FINP0940
   40 FORMAT('0',5X,'JOINT (GLOBALGRAPHIC)',2X,'TORQUE DEFLECTION',6X,'HFINP0950
     *ERRON FORMULA',10X,'R FACTOR',13X,'G FACTOR',10X,'FRICTION COEF.')FINP0960
      IJK = 1                                                           FINP0970
C                                                                       FINP0980
C     INPUT CONTACT SURFACE NO., SEGMENT NO., AND FUNCTION NOS.         FINP0990
C     INPUT CARD F.(I).(K)                                              FINP1000
```

```
C
      READ  (5,33) NJ,NS,NF
      WRITE (6,41) NJ,NS,NF
   41 FORMAT('0',I7,'-',I3,I11,'-',13,18,4I21)
      IF (NJ.NE.J) WRITE (6,42)
   42 FORMAT(' FINPUT INPUT ERROR. PROGRAM TERMINATED.')
      IF (NJ.NE.J) STOP
      NLT = 1
      DO 43 JJ = 1,31
   43 KTITLE(JJ) = BLANK
      GO TO (44,46,48,49),I
C
C     PLACE SEGMENT NO. AND INDEX TO NTAB ARRAY INTO M- AND NT- ARRAYS.
C
   44 MPL(1,K,J) = NS(1)
      MPL(2,K,J) = NS(2)
      MPL(3,K,J) = NS(3)
      NTPL(K,J) = NT
      DO 45 JJ = 1,5
   45 KTITLE(JJ) = PLTTL (JJ,J)
      GO TO 50
   46 MBLT(1,K,J) = NS(1)
      MBLT(2,K,J) = NS(2)
      MBLT(3,K,J) = NS(3)
      NTBLT(K,J) = NT
      DO 47 JJ = 1,5
   47 KTITLE(JJ) = BLTTTL (JJ,J)
C
C     SET UP TWO TABLES FOR FULL BELT FRICTION
C
      IF (NF(5).NE.0)  NLT = 2
      GO TO 50
   48 MSEG(1,K,J) = NS(1)
      MSEG(2,K,J) = NS(2)
      MSEG(3,K,J) = NS(3)
      NTSEG(K,J) = NT
      KTITLE (3) = SEG(J)
      GO TU 50
C
C       NOTE:  GLOBALGRAPHIC JOINT WILL SAVE NT IN IGLOB ARRAY
C
   49  IGLOB(J) = NT
      KTITLE(2) = JOINT(J)
C
C     SET UP POINTERS TO TAB ARRAY IN NTAB ARRAY.
C
   50 NFJ = NS(2)
      IF (NFJ.GT.0) KTITLE(6) = SEG(NFJ)
      DO 52 JJ = 1,5
      IF (NF(JJ).LE.0) GO TO 52
```

```
      NFJ = NF(JJ)
      DO 51 KK = 1,5
      KJ = 5*JJ+KK+1
   51 KTITLE(KJ) = JTITLE(KK,NFJ)
   52 CONTINUE
      WRITE (6,53) KTITLE
   53 FORMAT(1X,5A4,1X,A4,5(1X,5A4))
      DO 58 NL = 1,NLT
      NTAB(NT) = J1
      NT = NT+1
      DO 56 L=1,5
      NX = NF(L)
      NTAB(NT) = 0
      IF (NX.EQ.0) GO TO 55
      NTAB(NT) = NTI(NX)
      IF (NTI(NX).NE.0) GO TO 56
      WRITE(6,54) NX
   54 FORMAT ('0 FUNCTION NO.',I4,' HAS NOT BEEN DEFINED.  PROGRAM TERMI
     *NATED.')
      STOP
   55 IF (L.NE.1) GO TO 56

      IF FORCE DEFLECTION FUNCTION NO. IS ZERO,
      SET UP FOR ROLLING CONSTRAINT

      NQ = NQ+1
      NTAB(NT) = -NQ
      KQTYPE(NQ) = -4
      KQ1(NQ) = NS(2)
      KQ2(NQ) = NS(1)
      IF (I.NE.3) GO TO 56
      KQ1(NQ) = J
      KQ2(NQ) = NS(2)
   56 NT = NT+1

      INITIALIZE TAB ARRAY TO ZERO EXCEPT FOR DMAX, DINER, FDMAX.

      J2 = J1+19
      DO 57 JJ=J1,J2
   57 TAB(JJ) = 0.0
      NX = NTAB(NT-5)
      IF (NX.LT.0) GO TO 58
      TAB(J1+8) = DABS(TAB(NX+1))
      IF (TAB(NX+2).NE.0.0) TAB(J1+8) = DABS(TAB(NX+2))
      TAB(J1+10) = EVALFD(TAB(J1+8),NX,1)
      NX = NTAB(NT-4)
      IF (NX.LE.0) GO TO 58
      TAB(J1+9) = DABS(TAB(NX+1))
      IF (TAB(NX+2).NE.0.0) TAB(J1+9) = DABS(TAB(NX+2))
   58 J1 = J2+1
```

```
   59 CONTINUE
   60 CONTINUE
   61 CONTINUE
      MXNT8 = NT-I
      MXTB2 = J1-1
      IF (MXTB2.GT.2000) WRITE (6,62) MXTB2
   62 FORMAT('0 ERROR IN SUBROUTINE FINPUT, SIZE OF TA8 ARRAY =',I8//
     *        '  PROGRAM TERMINATED')
      IF (MXTB2.GT.2000) STOP
C
C     INPUT CARD F.5 - JOINT FUNCTIONS TO BE USED.
C
      IF (NJNT.LE.0) GO TO 81
      READ (5,33) (JOINTF(J),J=I,NJNT)
      IJK = 0
      DO 80 J=1,NJNT
      IF (JOINTF(J).EQ.0) GO TO 80
      IF (IJK.EQ.0) WRITE (6,77)
   77 FORMAT('1',119X,'CARD F.5'/
     *        ' THE FOLLOWING JOINT RESTORING FORCE FUNCTIONS AS DEFINED
     *ON CARDS E.7 WILL BE USED.'//4X,'JOINT',IOX,'FUNCTION'//)
      JF = JOINTF(J)
      IJK = 1
      WRITE (6,78) J,JOINT(J),JF,(JTITLE(I,JF),I=1,5)
   78 FORMAT(I6,'-',A4,I10,'-',5A4)
      IF (NTI(JF).EQ.0) WRITE (6,42)
      IF (NTI(JF).EQ.0) STOP
   80 CONTINUE
C
C     INPUT CONTACT SEGMENTS FOR AIRBAG, IF ANY.
C
   81 IF (NBAG.LE.0) GO TO 69
      IJK = 0
      DO 68 J=1,NBAG
C
C     INPUT CARD F.6.(J)
C
      READ (5,63) K,NK,(MBAG(2,I,J),MBAG(3,I,J),I=I,NK)
   63 FORMAT(2I4,20I2)
      MNBAG(J) = NK
      IF (NK.EQ.0) GO TO 68
      IF (IJK.EQ.0) WRITE (6,64)
   64 FORMAT(////5X,'AIRBAG',4X,'VS.',4X,'SEGMENTS',90X,'CARDS F.6')
      IF (K.NE.J) WRITE (6,42)
      IF (K.NE.J) STOP
      WRITE (6,65) J,(MBAG(2,I,J),MBAG(3,I,J),I=1,NK)
   65 FORMAT('0     NO.',I2,12X,IO(I3,'-',I3))
      DO 66 I=1,NK
      K = MBAG(2,I,J)
   66 KTITLE(I) = SEG(K)
```

```
      WRITE (6,67) (BAGTTL(I,J),I=1,5),(KTITLE(I),I=1,NK)               FINP2510
   67 FORMAT(1X,5A4,10(3X,A4))                                          FINP2520
   68 CONTINUE                                                          FINP2530
C                                                                       FINP2540
C     INPUT CARDS F.7.A-F.7.B FOR SUBROUTINE WINDY.                     FINP2550
C                                                                       FINP2560
   69 READ  (5,33) (MWSEG(1,J),J=1,NSEG)                                FINP2570
      IPAGE = 0                                                         FINP2580
      DO 73 J=1,NSEG                                                    FINP2590
      IWIND(J) = 0                                                      FINP2600
      WTIME(J) = 0.0                                                    FINP2610
      IF (MWSEG(1,J).EQ.0) GO TO 73                                     FINP2620
      IF (IPAGE.EQ.0) WRITE (6,70)                                      FINP2630
   70 FORMAT('1 SEGMENT WIND FORCES',99X,'CARDS F.7'//                  FINP2640
     *       '  SEGMENT-ELLIPSOID  SEGMENT-PLANE',                      FINP2650
     *       17X,'WIND FORCE FUNCTION')                                 FINP2660
      IPAGE = 1                                                         FINP2670
      READ  (5,33) (MWSEG(I,J),I=1,5)                                   FINP2680
      WRITE (6,71) (MWSEG(I,J),I=1,5)                                   FINP2690
   71 FORMAT('0',I7,' -',I3,I14,' -',I3,I30)                            FINP2700
      IF (MWSEG(1,J).NE.J) WRITE (6,42)                                 FINP2710
      IF (MWSEG(1,J).NE.J) STOP                                         FINP2720
      M3 = MWSEG(3,J)                                                   FINP2730
      M4 = MWSEG(4,J)                                                   FINP2740
      M5 = MWSEG(5,J)                                                   FINP2750
      WRITE (6,72) SEG(J),SEG(M3),(PLTTL(I,M4),I=1,5)                   FINP2760
     *                           ,(JTITLE(I,M5),I=1,5)                  FINP2770
   72 FORMAT(5X,A4,15X,A4,'-',5A4,2X,5A4)                               FINP2780
   73 CONTINUE                                                          FINP2790
      RETURN                                                            FINP2800
      END                                                               FINP2810
```

```
      SUBROUTINE FLXSEG
C                                                         REV 12 10/25/74
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/SGMNTS/D(3,3,22),WMEG(3,22),WMEGD(3,22),U1(3,22),U2(3,22)
     *              ,SEGLP(3,22),SEGLV(3,22),SEGLA(3,22),NSYM(22)
      COMMON/FLXBLE/ HF(4,12,8),B42(3,3,24),V4(3,8),NFLEX(3,8),NFLX
      COMMON/CNSNTS/ PI, RADIAN,G,THIRD,EPS1,EPS4,EPS6,EPS8,
     *               EPS12,EPS15,EPS20,EPS24,UNITL,UNITM,UNITT,GRAVTY(3)
      COMMON/TEMPVS/ TT(3,3), THN(4), CN1(3,3), CN(3,3), WNM1(3),
     *               THND(4), PTD(3), WCSN(3), RHSN(3), RHS1(3),
     *               RHS2(3), GF(3,4), GC(3,3), CGC(3,3), THA(3),
     *              THAD(3), THADEG(3), DN2N1(3,3), RMG(3)
      IF (NFLX.EQ.0) GO TO 99
      CALL ELTIME(1,34)
      IFX = 1
   11 N1 = NFLEX(1,IFX)
      N3 = NFLEX(3,IFX)
      CALL DOTT(D(1,1,N3),D(1,1,N1),TT,3,3,3)
      THN(1) =  DATAN2(TT(1,2),TT(1,1))
      THN(2) = -DARSIN(TT(1,3))
      THN(3) =  DATAN2(TT(2,3),TT(3,3))
      THN(4) =  1.0
      CT22 = 1.0-TT(1,3)**2
      CT2  = DSQRT(CT22)
      ST2  = -TT(1,3)
      CT1  =  TT(1,1)/CT2
      ST1  =  TT(1,2)/CT2
      CN1(1,1) = -TT(1,1)*TT(1,3)/CT22
      CN1(1,2) = -TT(1,2)*TT(1,3)/CT22
      CN1(1,3) =  1.0
      CN1(2,1) = -ST1
      CN1(2,2) =  CT1
      CN1(2,3) =  0.0
      CN1(3,1) =  TT(1,1)/CT22
      CN1(3,2) =  TT(1,2)/CT22
      CN1(3,3) =  0.0
      CALL DOT(TT,WMEG(1,N3),WNM1,3,1,3)
      DO 12 I=1,3
   12 WNM1(I) = WNM1(I) - WMEG(I,N1)
      CALL MAT(CN1,WNM1,THND,3,3,1,3,3,3)
      THND(4) =  0.0
      CALL CROSS(WMEG(1,N1),WNM1,WCSN)
      RHSN(1) = ( (-THND(1)*ST1*ST2 + THND(2)*CT1/CT2)*WNM1(1)
     *           +( THND(1)*CT1*ST2 + THND(2)*ST1/CT2)*WNM1(2) )/CT2
      RHSN(2) =  -THND(1)*(CT1*WNM1(1) + ST1*WNM1(2))
      RHSN(3) = ( (-THND(1)*ST1 + THND(2)*CT1*ST2/CT2)*WNM1(1)
     *           +( THND(1)*CT1 + THND(2)*ST1*ST2/CT2)*WNM1(2) )/CT2
   13 N2 = NFLEX(2,IFX)
      M = 0
      DO 15 I=1,3
```

```
      DO 14 J=1,4
      JM = J+M
      GF(I,J) = 0.0
      DO 14 K=1,4
   14 GF(I,J) = GF(I,J) + HF(K,JM,IFX)*THN(K)
   15 M = M+4
      DO 17 I=1,3
      THA(I)  = 0.0
      THAD(I) = 0.0
      DO 16 J=1,4
      THA (I) = THA (I) + GF(I,J)*THN (J)
   16 THAD(I) = THAD(I) + GF(I,J)*THND(J)
      THA (I) = 0.5*THA(I)
   17 THADEG(I) = THA(I)/RADIAN
      CALL DRCYPR(DN2N1,THADEG,3,2,1)
      CALL MAT(DN2N1,D(1,1,N1),D(1,1,N2),3,3,3,3,3,3)
      CSC = DCOS(THA(2))
      CSS = DSIN(THA(2))
      CN(1,1) =   0.0
      CN(2,1) =   0.0
      CN(3,1) =   1.0
      CN(1,2) =  -DSIN(THA(1))
      CN(2,2) =   DCOS(THA(1))
      CN(3,2) =   0.0
      CN(1,3) =   CSC*CN(2,2)
      CN(2,3) =  -CSC*CN(1,2)
      CN(3,3) =  -CSS
      CALL MAT(GF, CN1, GC, 3,3,3,3,3,3)
      CALL MAT(CN, GC, CGC, 3,3,3,3,3,3)
      CALL DOT (D(1,1,N1),CGC,B42(1,1,3*IFX-2),3,3,3)
      CALL DOTT(B42(1,1,3*IFX-2),TT,B42(1,1,3*IFX),3,3,3)
      DO 20 I=1,3
      DO 20 J=1,3
      B42(I,J,3*IFX-2) = B42(I,J,3*IFX-2) - D(J,I,N1)
      B42(I,J,3*IFX-1) = D(J,I,N2)
   20 B42(I,J,3*IFX  ) = -B42(I,J,3*IFX)
C
C     COMPUTE V4
C
      CALL MAT(CGC,WNM1,RHS1,3,3,1,3,3,3)
      DO 21 I=1,3
   21 RMG(I) = RHS1(I) + WMEG(I,N1)
      CALL MAT(DN2N1,RMG,WMEG(1,N2),3,3,1,3,3,3)
      CALL CROSS(WMEG(1,N1),RHS1,RHS2)
      CALL MAT(CGC,WCSN,RHS1,3,3,1,3,3,3)
      DO 25 I=1,3
   25 RHS1(I) = RHS2(I) - RHS1(I)
      CALL MAT(GC,WNM1,RHS2,3,3,1,3,3,3)
      RHS1(1) = RHS1(1) - THAD(1)*(CN(2,2)*RHS2(2)-CN(1,2)*CSC*RHS2(3))
     *                  - THAD(2)*CN(2,2)*CSS*RHS2(3)
```

```
      RHS1(2) = RHS1(2) + THAD(1)*(CN(1,2)*RHS2(2)+CN(2,2)*CSC*RHS2(3))
     *                  + THAD(2)*CN(1,2)*CSS*RHS2(3)
      RHS1(3) = RHS1(3) - THAD(2)*CSC*RHS2(3)
      CALL MAT(GF, RHSN, RHS2, 3,3,1,3,3,3)
      M = 1
      DO 30 I=1,3
      CALL MAT(HF(1,M,IFX), THND, PTD, 3,3,1,4,3,3)
      RHS2(I) = RHS2(I) + XDY(PTD,CN1,WNM1)
   30 M = M+4
      CALL MAT(CN, RHS2, PTD, 3,3,1,3,3,3)
      DO 35 I=1,3
   35 RHS1(I) = RHS1(I) + PTD(I)
      CALL DOT(D(1,1,N1),RHS1,V4(1,IFX),3,1,3)
      IF (IFX.EQ.NFLX) GO TO 98
      IFX = IFX+1
      IF (NFLEX(1,IFX).EQ.N1 .AND. NFLEX(3,IFX).EQ.N3) GO TO 13
      GO TO 11
   98 CALL ELTIME(2,34)
   99 RETURN
      END
```

```
      DOUBLE PRECISION FUNCTION FNTERP(THETA,PHI,NT)
C                                                       REV 12 12/06/74
C     COMPUTES THE RESTORING TORQUE OF A JOINT AS A FUNCTION OF THE
C     FLEXURE ANGLE (THETA) AND THE AZIMUTH ANGLE (PHI) AS DEFINED BY
C     FUNCTION NO. NT
C
C     ASSUMES  0  < THETA < PI
C             -PI <  PHI  < PI
C              DATA IN TAB ARRAY CONTAINS NTHETA,NPHI FOLLOWED BY
C              TWO DIMENSIONAL ARRAY OF FUNCTIONAL VALUES (NTHETA > 0)
C              OR POLYNOMIAL COEFFICIENTS (NTHETA < 0) FOR EQUALLY
C              SPACED VALUES OF PHI.
C
C                   THETA(I) = (I-1)*PI/(NTHETA-1) FOR I=1,NTHETA
C                   PHI(J) = -PI + (J-1)*2*PI/NPHI FOR J=1,NPHI
C                       F(THETA,PI) = F(THETA,-PI)
C
C     SUBROUTINE EVALUATES G1(THETA) = F(THETA,PHI(J)   )
C                          G2(THETA) = F(THETA,PHI(J+1))
C                FOR  PHI(J) < PHI < PHI(J+1)
C     BY LINEAR INTERPOLATION OR POLYNOMIAL EVALUATION AND THEN LINEAR
C     INTERPOLATES BETWEEN G1 AND G2 TO OBTAIN F(THETA,PHI).
C     IF F < 0, F IS SET TO ZERO, THEREFORE A DEAD BAND IS OBTAINED
C     BY NEGATIVE VALUES IN THE TABLE.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS1,EPS4,EPS6,EPS8,
     *               EPS12,EPS15,EPS20,EPS24,UNITL,UNITM,UNITT,GRAVTY(3)
      COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NT1(50),NTAB(500),TAB(2000)
      IERROR = 0
      IF (PHI.LT.-PI)   IERROR = 1
      IF (PHI.GT. PI)   IERROR = 2
      IF (THETA.LT.0.0) IERROR = 3
      IF (THETA.GT.PI ) IERROR = 4
      IF (IERROR.NE.0)  WRITE (6,11) IERROR,THETA,PHI,NT
   11 FORMAT('0 IMPROPER ARGUMENTS TO FUNCTION FNTERP. ERROR CODE =',I4/
     *       '0 THETA =',G25.15, ' PHI =',G25.15,' NT =',I6)
      IF (IERROR.NE.0) STOP
      NF = NT1(NT) + 5
      NTHETA = TAB(NF)
      NPHI   = TAB(NF+1)
C
C     DETERMINE INDEX AND INTERPOLATION PARAMETERS FOR PHI.
C
      XNP = (PHI+PI)/(2.0*PI)*TAB(NF+1)
      NP1 = XNP
      NP2 = NP1+1
      IF (NP2.GE.NPHI) NP2 = 0
      RP2 = XNP - DFLOAT(NP1)
      RP1 = 1.0 - RP2
```

```
      NTH = IABS(NTHETA)
      IP1 = NF+1+NP1*NTH
      IP2 = NF+1+NP2*NTH
C
C     DETERMINE INDEX AND INTERPOLATION PARAMETERS FOR THETA.
C
      IF (NTHETA.LT.0) GO TO 20
      XNT = THETA/PI*(TAB(NF)-1.0)
      NT1 = XNT
      RT2 = XNT - DFLOAT(NT1)
      RT1 = 1.0 - RT2
      IT1 = IP1 + NT1
      IT2 = IP2 + NT1
      G1  = RT1*TAB(IT1+1) + RT2*TAB(IT1+2)
      G2  = RT1*TAB(IT2+1) + RT2*TAB(IT2+2)
      GO TO 23
C
C     COMPUTE FOR POLYNOMIALS IN THETA FOR FIXED PHI.
C
   20 NPOLY = -NTHETA-1
      IT1 = IP1 + NPOLY + 2
      IT2 = IP2 + NPOLY + 2
      THETA1 = THETA - TAB(IP1+1)
      THETA2 = THETA - TAB(IP2+1)
      G1 = 0.0
      G2 = 0.0
      DO 21 I=1,NPOLY
      IT1 = IT1-1
      IT2 = IT2-1
      G1 = THETA1*(TAB(IT1)+G1)
   21 G2 = THETA2*(TAB(IT2)+G2)
   23 FNTERP = RP1*G1 + RP2*G2
      IF (FNTERP.LT.0.0) FNTERP = 0.0
      RETURN
      END
```

```
      SUBROUTINE HBELT(NSECT,IBAR,BAR,NT,XLONG)
C                                                        REV 12 12/19/74
C
C     COMPUTES THE FORCES AND TORQUES OF INDIVIDUAL BELT SECTIONS AND
C     ADDS THEM TO THE U1 AND U2 ARRAYS FOR CONTACTING SEGMENTS.
C       ARGUMENTS:
C         NSECT     - NO. OF REFERENCE POINTS ON BELT
C         IBAR(1,J) - SEGMENT NO. ASSOCIATED WITH PUINT J
C             (2,J) - ELLIPSOID NO. (J=1,NSECT)
C         BAR (1,J) - INPUT REFERENCE POINT
C             (4,J) - CONTACT POINT
C         NT        - INDEX TO FORCE DEFLECTION FUNCTION
C         XLONG     - REFERENCE LENGTH
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION BAR(6,10),IBAR(2,10)
      COMMON/CONTRL/NSEG,NJNT,NS3,NJ3,NPL,NBLT,NBAG,NVEH,NGRND,NPRT(40)
      COMMON/VPOSTN/TIME
      COMMON/SGMNTS/D(3,3,22),WMEG(3,22),WMEGD(3,22),U1(3,22),U2(3,22)
     *             ,SEGLP(3,22),SEGLV(3,22),SEGLA(3,22),NSYM(22)
      COMMON/CNTSRF/ PL(17,20),GAB(8,3),BELT(20,8),TPTS(6,8),BD(24,25)
      COMMON/TEMPVS/T1(3),T2(3),T3(3),T4(3),T5(3),T6(3),T7(3),T8(3),
     *             ZNR(3),ZW(3,101),ZY(3,100),DS(100),JF(101)
      CALL ELTIME(1,36)
       LL = 0
       DO 20 K=1,NSECT
C
C     COMPUTE ZW(K) - THE LOCATION OF POINT(K) IN INERTIAL REFERENCE.
C
       KK = IBAR(1,K)
       CALL DOT(D(1,1,KK),BAR(4,K),T1,3,1,3)
       DO 11 J=1,3
   11  ZW(J,K) = SEGLP(J,KK) + T1(J)
       IF (K.EQ.1)        GO TO 20
       LL = LL+1
   12  JJ = JF(LL)
C
C     COMPUTE VECTOR ZY(LL) AND LENGTH DS(LL) FOR BELT SECTION LL
C     BETWEEN POINTS K=JF(LL+1) AND JJ=JF(LL).
C
       DSS = 0.0
       DO 13 J=1,3
       ZY(J,LL) = ZW(J,K) - ZW(J,JJ)
   13  DSS = DSS + ZY(J,LL)**2
       DS(LL) = DSQRT(DSS)
       IF (LL.EQ.1)       GO TO 20
C
C     COMPUTE DPR - DOT PRODUCT BETWEEN ZY(LL) AND ZY(LL-1)
C
       KK = IBAR(1,JJ)
       MM = IBAR(2,JJ)
```

```
      IF (MM.EQ.0)       GO TO 20
      DO 14 J=1,3
   14 T2(J) = BAR(J+3,JJ) - BD(J+3,MM)
      CALL MAT(BD( 7,MM),T2,T3,3,3,1,3,3,3)
      CALL DOT(D(1,1,KK),T3,ZNR,3,1,3)
      DPR = 0.0
      DO 15 J=1,3
   15 DPR = DPR + ZNR(J)*(ZY(J,LL)/DS(LL) - ZY(J,LL-1)/DS(LL-1))
      IF (DPR.LT.0.0)    GO TO 20
C
C    POSITIVE DPR INDICATES BELT IS PULLING AWAY FROM POINT JJ. REMOVE
C    POINT FROM FUTURE CONSIDERATION AND DECREASE LL - NO. OF LENGTHS.
C
      LL = LL-1
      GO TO 12
   20 JF(LL+1) = K
C
C    COMPUTE XLG - TOTAL LENGTH OF THE LL BELT SECTIONS.
C
      XLG = 0.0
      DO 30 L=1,LL
   30 XLG = XLG + DS(L)
C
C     NEGATIVE XLONG INDICATES INITIAL SLACK IN BELT.
C
     IF (XLONG.LT.0.0) XLONG = XLG-XLONG
C
C    COMPUTE FRC - TOTAL FORCE OF BELT AND APPLY IT TO ALL SEGMENTS.
C
     STRAIN = (XLG-XLONG)/XLONG
     FRC = FRCDFL(STRAIN,NT,1)
      IF (FRC.LE.0.0)    GO TO 99
     IF (NPRT(16).NE.0) WRITE (6,31) TIME,XLG,STRAIN,FRC,LL
   31 FORMAT('0 SUB HBELT',F13.6,3G18.7,I6)
      DO 40 L=1,LL
      L1 = JF(L)
      L2 = JF(L+1)
      K1 = IBAR(1,L1)
      K2 = IBAR(1,L2)
      CALL MAT(D(1,1,K1),ZY(1,L),T4,3,3,1,3,3,3)
      CALL MAT(D(1,1,K2),ZY(1,L),T5,3,3,1,3,3,3)
      CALL CROSS(BAR(4,L1),T4,T6)
      CALL CROSS(BAR(4,L2),T5,T7)
      FR = FRC/DS(L)
     IF (NPRT(16).NE.0) WRITE (6,32) L1,K1,K2,DS(L),FR,(ZY(J,L),J=1,3)
   32 FORMAT(6X,3I6,5G18.7)
      DO 40 J=1,3
      U1(J,K1) = U1(J,K1) + FR*ZY(J,L)
      U2(J,K1) = U2(J,K1) + FR*T6(J)
      U1(J,K2) = U1(J,K2) - FR*ZY(J,L)
```

```
40  U2(J,K2) = U2(J,K2) - FR*T7(J)                                    HBEL1010
99 CALL ELTIME(2,36)                                                  HBEL1020
   RETURN                                                             HBEL1030
    END                                                               HBEL1040
```

```
      SUBROUTINE HINPUT
C                                                        REV 12 12/19/74
C
C     CONTROLS THE INPUT OF CARDS F.8.A - F.8.D CONTAINING THE SETUP AND
C     CONTROL OF THE HARNESS BELT SYSTEM.
C
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/HRNESS/ BAR(6,100) , XLONG(20), IBAR(2,100), NTHRNS(20),
     *               NHRNSS, NBLTPH(5), NFBLT(5,20), NPTSPB(20)
      COMMON/TEMPVS/JTITLE(5,51),NF(5),NS(3),KTITLE(31)
      COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(500),TAB(2000)
      COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),BLTTTL(5,8)
     *              ,PLTTL(5,20),BAGTTL(5,6),SEG(22),JOINT(21)
     *              ,CGS(21),JS(21)
      REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT
      LOGICAL*1 CGS,JS
C
C     INPUT CARD F.8.A
C         NHRNSS - NO. OF HARNESS-BELT SYSTEMS
C         NBLTPH - NO. OF BELTS PER HARNESS
C
      READ (5,11) NHRNSS,(NBLTPH(I),I=1,NHRNSS)
   11 FORMAT(18I4)
      IF (NHRNSS.LE.0) GO TO 99
      WRITE (6,12) NHRNSS,(NBLTPH(I),I=1,NHRNSS)
   12 FORMAT('1 HARNESS-BELT SYSTEM INPUT',93X,'CARDS F.8'//
     *       ' NO. OF HARNESSES =',I4//
     *       ' NO. OF BELTS PER HARNESS =',5I6)
      J1 = 1
      K1 = 1
      JJ1 = MXTB2 + 1
      NT  = MXNTB + 1
      DO 90 I=1,NHRNSS
      IF (NBLTPH(I).LE.0) GO TO 90
      J2 = J1 + NBLTPH(I) -1
C
C     INPUT CARD F.8.B - NPTSPB - NO. OF POINTS PER BELT.
C
      READ (5,11) (NPTSPB(J),J=J1,J2)
      WRITE (6,13) I,(NPTSPB(J),J=J1,J2)
   13 FORMAT('0 FOR HARNESS NO.',I3,' NO. OF POINTS PER BELT =',20I4)
      DO 80 J=J1,J2
      IF (NPTSPB(J).EQ.0) GO TO 80
C
C     INPUT CARD F.8.C - 5 FUNCTION NOS AND LENGTH OF EACH BELT.
C
      READ (5,14)       (NFBLT(L,J),L=1,5),XLONG(J)
   14 FORMAT(5I4,F12.6)
      WRITE (6,15) I,J,(NFBLT(L,J),L=1,5),XLONG(J)
   15 FORMAT('0 HARNESS NO.',I3,' BELT NO.',I3,' FUNCTION NOS.',5I6,
```

```
     *        '  REFERENCE LENGTH =',G16.8///
     *        ' HARNESS  BELT  POINT  SEGMENT ELLIPSOID  ',
     *          9X,'REFERENCE POINT'    /
     *        3X,'NO.',5X,'NO.',3X,'NO.',5X,'NO.',5X,'NO.',6X,
     *          6X,'X',9X,'Y',9X,'Z',3X / )
C
C      CHANGE SIGN OF XLONG FOR INITIAL CALL TO HBELT.
C
      XLONG(J) = -XLONG(J)
C
C     SET UP POINTERS IN NTAB AND INITIAL VALUES OF TAB FOR BELT J
C         AS WAS DONE FOR OTHER CONTACTS IN SUBROUTINE FINPUT.
C
      NTHRNS(J) = NT
      NTAB(NT)  = JJ1
      NT = NT+1
      DO 17 L=1,5
      NTAB(NT) = 0
      NX = NFBLT(L,J)
      IF (NX.EQ.0) GO TO 17
      NTAB(NT) = NTI(NX)
      IF (NTI(NX).GT.0) GO TO 17
      WRITE (6,16) NX
   16 FORMAT('0 FUNCTION NO.',I4,' HAS NOT BEEN DEFINED.',
     *        ' PROGRAM TERMINATED.')
      STOP
   17 NT = NT+1
      JJ2 = JJ1+19
      DO 18 JJ=JJ1,JJ2
   18 TAB(JJ) = 0.0
      NX = NTAB(NT-5)
      IF (NX.LT.0) GO TO 19
      TAB(JJ1+8) = DABS(TAB(NX+1))
      IF (TAB(NX+2).NE.0.0) TAB(JJ1+8) = DABS(TAB(NX+2))
      TAB(JJ1+10) = EVALFD(TAB(JJ1+8),NX,1)
      NX = NTAB(NT-4)
      IF (NX.LE.0) GO TO 19
      TAB(JJ1+9) = DABS(TAB(NX+1))
      IF (TAB(NX+2).NE.0.0) TAB(JJ1+9) = DABS(TAB(NX+2))
   19 JJ1 = JJ2+1
      K2 = K1 + NPTSPB(J) - 1
      DO 70 K=K1,K2
C
C     INPUT CARD F.8.D
C
      READ  (5,21)       (IBAR(L,K),L=1,2),(BAR(L,K),L=1,3)
      WRITE (6,22) I,J,K,(IBAR(L,K),L=1,2),(BAR(L,K),L=1,3)
   21 FORMAT(2I6,3F12.6)
   22 FORMAT(I5,I8,I6,2I8,7X,3F10.3)
      DO 23  L=1,3
```

```
23 BAR(L+3,K) = BAR(L,K)
70 CONTINUE
   K1 = K2+1
80 CONTINUE
   J1 = J2+1
90 CONTINUE
   MXNTB = NT-1
   MXTB2 = JJ1-1
   IF (MXTB2.GT.2000) WRITE (6,62) MXTB2
62 FORMAT('0 ERROR IN SUBROUTINE HINPUT, SIZE OF TAB ARRAY =',I8//
  *       '  PROGRAM TERMINATED.')
   IF (MXTB2.GT.2000) STOP
99 RETURN
   END
```

```
      SUBROUTINE IMPULS(I1,I2,I3)
C                                                    REV 12 10/25/74
C         ARGUMENTS: I1 = 1 - IMPULS FOR PLELP.
C                         3 - IMPULS FOR SEGSEG.
C                         4 - IMPULS FOR VISPR OR EJOINT
C                    I2 = INDEX OF CONTACTING SEGMENT OR JOINT AXIS
C                    I3 = PLANE, SEGMENT,OR JOINT NUMBER
C
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/CONTRL/NSEG,NJNT,NS3,NJ3,NPL,NBLT,NBAG,NVEH,NGRND,NPRT(40)
      COMMON/VPOSTN/ TIME,X0(3),XDOT0(3),XCOMP(3),XVCOMP(3),AX(3),
     *              ANGLE(3),VMPH,VTIME,ATAB(15,100),AT0,ADT,OMEGA,
     *              NATAB,NACLR,DVEH(3,3),VMEG(3),VMEGD(3),XACOMP(3),
     *              THET(3),ZPLT(3)
      COMMON/SGMNTS/D(3,3,22),WMEG(3,22),WMEGD(3,22),U1(3,22),U2(3,22)
     *              ,SEGLP(3,22),SEGLV(3,22),SEGLA(3,22),NSYM(22)
      COMMON/CMATRX/V1(3,21),V2(3,21),V3(3,12),B12(3,3,42),A22(3,3,42)
     *              ,F(3,21),TQ(3,21),WJ(21)
      COMMON/DESCRP/ PHI(3,22),W(22),SR(3,42),HA(3,42),HB(3,42)
     *              ,HT(3,3,42),RPHI(3,22),RW(22),SPRING(5,63)
     *              ,VISC(7,63),JNT(21),IPIN(21),NS,ISING(22)
     *              ,IGLOB(21)
      COMMON/JBARTZ/ MNPL(  20),MNBLT(  8),MNSEG(   22),MNBAG(    6),
     *             MPL(3,5,20),MBLT(3,5,8),MSEG(3,5,22),MBAG(3,10,6),
     *              NTPL(5,20),NTBLT(5,8),NTSEG(5,22)
      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)
     *             ,NQ,KQ1(12),KQ2(12),KQTYPE(12)
      COMMON/FLXBLE/ HF(4,12,8),B42(3,3,24),V4(3,8),NFLEX(3,8),NFLX
      COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(500),TAB(2000)
      COMMON/TEMPVI/ TT1(3),R1I(3),R2I(3),CREST,JSTOP(4,2,21)
      DIMENSION TEMP(3),DWR1(3),DWR2(3),DWR3(3),DWR4(3),VREL(3),DV(3)
      IF (TIME.EQ.0.0) GO TO 99
C
C     SPECIAL SETUP FOR CALL TO SUBROUTINE DAUX
C     REPLACE SETUP WITH U1,U2,V1,V2,V3 = 0.
C     ASSUME OTHER ARRAYS FROM PREVIOUS CALL TO DAUX.
C
      CALL ELTIME(1,27)
      CALL OUTPUT(0)
      KQTEST = 0
      NT = 0
      IF (I1.EQ.1) NT = NTPL (I2,I3)
      IF (I1.EQ.3) NT = NTSEG(I2,I3)
      IF (NT.EQ.0) GO TO 29
      KQ = -NTAB(NT+1)
      IF (KQ.LE.0) GO TO 29
      KQTYPE(KQ) = IABS(KQTYPE(KQ))
```

```
      CALL DAUX(0)
   29 IF (NQ.LE.0) GO TO 31
      DO 30 J=1,NQ
      DO 30 I=1,3
   30 V3(I,J) = 0.0
   31 DO 32 J=1,NSEG
      DO 32 I=1,3
      U1(I,J) = 0.0
   32 U2(I,J) = 0.0
      IF (NJNT.LE.0) GO TO 21
      DO 33 J=1,NJNT
      DO 33 I=1,3
      V1(I,J) = 0.0
   33 V2(I,J) = 0.0
   21 IF (NFLX.EQ.0) GO TO 23
      DO 22 J=1,NFLX
      DO 22 I=1,3
   22 V4(I,J) = 0.0
C
C     REPLACE CALLS TO CONTACT AND VISPR WITH SINGLE CALL
C     AT FIRST CONTACT IF NOT CONSTRAINT.
C
   23 IF (I1.NE.1) GO TO 34
      NT = NTPL(I2,I3)
      M1 = MPL(1,I2,I3)
      M2 = MPL(2,I2,I3)
      M3 = MPL(3,I2,I3)
      CALL PLELP(M2,M3,M1,I3,NT)
      IF (NTAB(NT+1).LT.0) GO TO 37
      K1 = M2
      K2 = M1
      GO TO 39
   34 IF (I1.NE.3) GO TO 35
      NT = NTSEG(I2,I3)
      M1 = MSEG(1,I2,I3)
      M2 = MSEG(2,I2,I3)
      M3 = MSEG(3,I2,I3)
      CALL SEGSEG(I3,M1,M2,M3,NT)
      IF (NTAB(NT+1).LT.0) GO TO 37
      K1 = I3
      K2 = M2
      GO TO 39
   35 IF (I1.NE.4) WRITE (6,36) I1,I2,I3
   36 FORMAT('0 IMPROPER ARGUMENTS TO SUBROUTINE IMPULS'/
     *         '  ARGUMENTS = ', 3I6 /
     *         '  PROGRAM TERMINATED' )
      IF (I1.NE.4) STOP
C
C     RECALL VISPR FOR JOINT STOP.
C
```

```
      IF (IABS(IPIN(I3)).NE.4) GO TO 25
      CALL EJOINT(I2,I3)
      GO TO 26
   25 CALL VISPR(I2,I3)
   26 K1 = IABS(JNT(I3))
      K2 = I3+1
      GO TO 39
C
C     SET UP SPECIAL U1,U2 FOR FIRST CONTACT OF CONSTRAINT.
C
   37 KQ = -NTAB(NT+1)
      KQTEST = 1
      KQTYPE(KQ) = -IABS(KQTYPE(KQ))
      K1 = KQ1(KQ)
      K2 = KQ2(KQ)
      IF (K1.GT.NSEG) GO TO 38
      CALL MAT(A13(1,1,2*KQ-1),QQ(1,KQ),U1(1,K1),3,3,1,3,3,3)
      CALL MAT(A23(1,1,2*KQ-1),QQ(1,KQ),U2(1,K1),3,3,1,3,3,3)
   38 IF (K2.GT.NSEG) GO TO 39
      CALL MAT(A13(1,1,2*KQ  ),QQ(1,KQ),U1(1,K2),3,3,1,3,3,3)
      CALL MAT(A23(1,1,2*KQ  ),QQ(1,KQ),U2(1,K2),3,3,1,3,3,3)
C
C     FINAL SETUP OF U1 AND U2
C
   39 DO 40 J=1,NSEG
      DO 40 I=1,3
      U1(I,J) = U1(I,J)*RW(J)
   40 U2(I,J) = U2(I,J)*RPHI(I,J)
      DO 41 I=1,3
      SEGLA(I,NVEH) = 0.0
   41 WMEGD(I,NVEH) = 0.0
      CALL DAUX(I1)
      IF (KQTEST.EQ.1) KQTYPE(KQ) = IABS(KQTYPE(KQ))
      IF (NPRT(10).NE.0) CALL PRINT(6HPREIMP)
      IF (I1.GT.3) GO TO 51
      IF (NPRT(10).NE.0) WRITE (6,42) R11,R21
   42 FORMAT ('0'/(6G20.8))
      CALL CROSS(WMEG (1,K1),R11(1),TEMP)
      CALL DOT(D(1,1,K1),TEMP,DWR1(1),3,1,3)
      CALL CROSS(WMEG (1,K2),R21(1),TEMP)
      CALL DOT(D(1,1,K2),TEMP,DWR2(1),3,1,3)
      CALL CROSS(WMEGD(1,K1),R11(1),TEMP)
      CALL DOT(D(1,1,K1),TEMP,DWR3(1),3,1,3)
      CALL CROSS(WMEGD(1,K2),R21(1),TEMP)
      CALL DOT(D(1,1,K2),TEMP,DWR4(1),3,1,3)
      TVREL = 0.0
      TDV   = 0.0
      DO 50 I=1,3
      VREL(I) = SEGLV(I,K1)+DWR1(I) - SEGLV(I,K2)-DWR2(I)
      DV  (I) = SEGLA(I,K1)+DWR3(I) - SEGLA(I,K2)-DWR4(I)
```

```
      TVREL = TVREL + TTI(I)*VREL(I)
   50 TDV   = TDV   + TTI(I)*DV  (I)
      GO TO 53
   51 CALL DOT(D(1,1,K1),WMEG (1,K1),DWR1(1),3,1,3)
      CALL DOT(D(1,1,K2),WMEG (1,K2),DWR2(1),3,1,3)
      CALL DOT(D(1,1,K1),WMEGD(1,K1),DWR3(1),3,1,3)
      CALL DOT(D(1,1,K2),WMEGD(1,K2),DWR4(1),3,1,3)
      TVREL = 0.0
      TDV   = 0.0
      DO 52 I=1,3
      VREL(I) = DWRI(1) - DWR2(1)
      DV  (I) = DWR3(I) - DWR4(I)
      TVREL = TVREL + TTI(1)*VREL(I)
   52 TDV   = TDV   + TTI(I)*DV  (1)
   53 ALPHA = 0.0
C
C     NOTE: CREST IS SUPPLIED AS (1+E)/2 WHERE E IS THE CLASSICAL
C     COEFFICIENT OF RESTITUTION BUT WITH A RANGE OF -1 TO +1.
C     CREST HAS A RANGE OF 0 TO +1 WHERE 0 (E=-1) REPRESENTS NO IMPULSE.
C
      IF (TDV.NE.0.0) ALPHA = -2.0*CREST*TVREL/TDV
      IF (NPRT(10).NE.0) WRITE (6,42) DWR1,DWR2,DWR3,DWR4,
     *                                TT1,VREL,DV,
     *                                TVREL,TDV,CREST,ALPHA
      DO 60 J=1,NSEG
      DO 60 I=1,3
      SEGLV(I,J) = SEGLV(I,J) + ALPHA*SEGLA(I,J)
   60 WMEG (I,J) = WMEG (I,J) + ALPHA*WMEGD(I,J)
      CALL OUTPUT(1)
      CALL PRINT(6HIMPULS)
      CALL ELTIME(2,27)
   99 RETURN
      END
```

```
      SUBROUTINE KINPUT                                                 KINP0010
C                                                 REV 12 12/II/74KINP0020
C     PERFORMS THE FOLLOWING CARD INPUT AFTER CARDS E.1-E.4 (SUBROUTINE KINP003C
C     CINPUT) AND BEFORE CARDS F.I-F.5 (SUBROUTINE FINPUT).             KINP0040
C         CARD E.5 - NWINDF: NO. OF WIND FORCE FUNCTIONS ON CARDS E.6   KINP0050
C                  - NJNTF : NO. OF JOINT FORCE FUNCTIONS ON CARDS E.7  KINP0060
C         CARDS E.6 - DEFINITIONS OF WIND FORCE FUNCTIONS               KINP0070
C         CARDS E.7 - DEFINITIONS OF JOINT RESTORING FORCE FUNCTIONS    KINP0080
C                                                                       KINP0090
      IMPLICIT REAL*8(A-H,O-Z)                                          KINPOI00
      COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(500),TAB(2000) KINP0110
      COMMON/TEMPVS/JTITLE(5,5I),NF(5),NS(3),KTITLE(31),TH(50)          KINPOI20
C     NOTE: TEMPVS IS SHARED HERE WITH SUBROUTINES CINPUT AND FINPUT.   KINP0130
      REAL BLANK/'    '/,JTITLE,KTITLE                                  KINPOI40
      COMMON/CNSNTS/ PI, RADIAN,G,THIRD,EPS1,EPS4,EPS6,EPS8,            KINPOI50
     *               EPSI2,EPSI5,EPS20,EPS24,UNITL,UNITM,UNITT,GRAVTY(3)KINPOI60
C                                                                       KINPOI70
C     INPUT CARD E.5 - NWINDF AND NJNTF                                 KINP0180
C                                                                       KINP0190
      READ (5,11) NWINDF,NJNTF                                          KINP0200
   I1 FORMAT(2I6)                                                       KINP0210
      JI = MXTB1+1                                                      KINP0220
      IF (NWINDF.LE.0) GO TO 3I                                         KINP0230
      DO 30 K=I,NWINDF                                                  KINP0240
C                                                                       KINP0250
C     INPUT CARD E.6.A - FUNCTION NO. AND TITLE                         KINP0260
C                                                                       KINP0270
      READ (5,12) I,(KTITLE(J),J=1,5)                                   KINP0280
   I2 FORMAT(I4,4X,5A4)                                                 KINP0290
      WRITE (6,13) I,(KTITLE(J),J=1,5),I,J1                             KINP0300
   13 FORMAT('1 WIND FORCE FUNCTION NO.',I4,4X,5A4,10X,'NTI(',I2,') =', KINP0310
     *       I5,43X,'CARDS E.6'//)                                      KINP0320
      IF (I.LE.0.OR.I.GT.50) WRITE (6,I4)                               KINP0330
   I4 FORMAT('0 IMPROPER FUNCTION NO. PROGRAM TERMINATED.')             KINP0340
      IF (I.LE.0.OR.I.GT.50) STOP                                       KINP0350
      IF (NTI(I).NE.0) WRITE (6,15) I                                   KINP0360
   15 FORMAT('0 FUNCTION NO.',I4,' HAS ALREADY BEEN INPUTTED AND WILL BEKINP0370
     * REPLACED BY THIS FUNCTION.')                                     KINP0380
      NTI(I) = J1                                                       KINP0390
      DO I6 J=I,5                                                       KINP0400
   I6 JTITLE(J,I) = KTITLE(J)                                           KINP0410
      J2 = JI+4                                                         KINP0420
C                                                                       KINP0430
C     INPUT CARD E.6.B - DO THRU D4 (FOR NOW A BLANK CARD)              KINP0440
C                                                                       KINP0450
      READ  (5,I7) (TAB(J),J=J1,J2)                                     KINP0460
      WRITE (6,18) (TAB(J),J=J1,J2)                                     KINP0470
   17 FORMAT(6F12.0)                                                    KINP0480
   I8 FORMAT(IOX,'DO',13X,'DI',I3X,'D2',I3X,'D3',13X,'D4'/5FI5.4//)     KINP0490
      J1 = J2+I                                                         KINP0500
```

```
C                                                                         KINP0510
C     INPUT CARD E.6.C - NTMPTS                                           KINP0520
C                                                                         KINP0530
      READ  (5,11) NTMPTS                                                 KINP0540
      WRITE (6,19) NTMPTS                                                 KINP0550
   19 FORMAT('0 WIND FORCE TABLES FOR ',I6,' TIME POINTS.'//              KINP0560
     *        11X,'T',14X,'FX(T)',15X,'FY(T)',15X,'FZ(T)'  /)             KINP0570
      TAB(J1) = NTMPTS                                                    KINP0580
      J1 = J1+1                                                           KINP0590
      J2 = J1+4*NTMPTS-1                                                  KINP0600
C                                                                         KINP0610
C     INPUT CARDS E.6.D-E.6.N - NTMPTS CARDS OF T,FX(T),FY(T),FZ(T)       KINP0620
C                                                                         KINP0630
      READ  (5,20) (TAB(J),J=J1,J2)                                       KINP0640
      WRITE (6,21) (TAB(J),J=J1,J2)                                       KINP0650
   20 FORMAT(4F12.0)                                                      KINP0660
   21 FORMAT(3X,F12.6,3G20.6)                                             KINP0670
      J1 = J2+1                                                           KINP0680
   30 CONTINUE                                                            KINP0690
   31 IF (NJNTF.LE.0) GO TO 51                                            KINP0700
      DO 50 K=1,NJNTF                                                     KINP0710
C                                                                         KINP0720
C     INPUT CARD E.7.A - FUNCTION NO. AND TITLE                           KINP0730
C                                                                         KINP0740
      READ  (5,12) I,(KTITLE(J),J=1,5)                                    KINP0750
      WRITE (6,32) I,(KTITLE(J),J=1,5),I,J1                               KINP0760
   32 FORMAT('1 JOINT FORCE FUNCTION NO.',I4,4X,5A4,10X,'NTI(',I2,') =',KINP0770
     *         I5,42X,'CARDS E.7'//)                                      KINP0780
      IF (I.LE.0.OR.I.GT.50) WRITE (6,14)                                 KINP0790
      IF (I.LE.0.OR.I.GT.50) STOP                                         KINP0800
      IF (NTI(I).NE.0) WRITE (6,15) I                                     KINP0810
      NTI(I) = J1                                                         KINP0820
      DO 33 J=1,5                                                         KINP0830
   33 JTITLE(J,I) = KTITLE(J)                                             KINP0840
C                                                                         KINP0850
C     INPUT CARD E.7.B - D0,D1,D2,D3,D4 (FOR NOW A BLANK CARD).           KINP0860
C                                                                         KINP0870
      J2 = J1+4                                                           KINP0880
      READ  (5,17) (TAB(J),J=J1,J2)                                       KINP0890
      WRITE (6,18) (TAB(J),J=J1,J2)                                       KINP0900
      J1 = J2+1                                                           KINP0910
C                                                                         KINP0920
C     INPUT CARD E.7.C - NTHETA,NPHI                                      KINP0930
C                                                                         KINP0940
      READ  (5,11) NTHETA,NPHI                                            KINP0950
      TAB(J1  ) = NTHETA                                                  KINP0960
      TAB(J1+1) = NPHI                                                    KINP0970
      J1 = J1+2                                                           KINP0980
      IF (NTHETA.LT.0) GO TO 38                                           KINP0990
      DO 35 J=1,NTHETA                                                    KINP1000
```

```
   35 TH(J) = DFLOAT(J-1)*180.0/DFLOAT(NTHETA-1)
      WRITE (6,36) NTHETA,NPHI,(TH(J),J=2,NTHETA)
   36 FORMAT('0 FUNCTION IS TABULAR FOR' ,I3,' X',I3,' VALUES OF THETA A
     *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('0 FUNCTION IS COEFFICIENTS OF' ,I3,' ORDER POLYNOMIALS IN
     *(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,
     *  A4,'N =',I2,11X,A4,'N =',I2,11X,A4,'N =',I2,11X,A4,'N =',I2)  )
   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.
C        EACH SET I IS FOR PHI(I) = -180 +(I-1)*360/NPHI DEGREES AND
C        ASSUMES DATA FOR PHI(NPHI+1) = 180 IS SAME AS PHI(1) = -180.
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
C     VALUES. OVERWRITE VALUE IN FIRST COLUMN (SUPPLIED AS THETA0) WITH
C     VALUE FOR THETA = 0 AND ALL OTHER ZERO VALUES.
C
      THETA0 = TAB(J1)
      IF (THETA0.EQ.0.0) GO TO 49
      JJ = THETA0*DFLOAT(NTHETA-1)/180.0 + 1.0 + EPS6
      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
      DO 45 J=1,JJ
      J1J = J1+J-1
      IF (J.NE.1.AND.TAB(J1J).GT.0.0) IERROR = 3
   45 TAB(J1J) = TAB(JJ1)*(TH(J)-THETA0)/(TH(JJ+1)-THETA0)
   46 IF (IERROR.NE.0) WRITE (6,47) IERROR
   47 FORMAT('0 INPUT ERROR. INCONSISTENT VALUE OF THETA0. IERROR =',I2,
     *       ' PROGRAM TERMINATED.')
      IF (IERROR.NE.0) STOP
   49 J1 = J2+1
   50 CONTINUE
```

```
51 MXTB1 = J1-1                                                        KINP1510
   RETURN                                                              KINP1520
   END                                                                 KINP1530
```

```
      SUBROUTINE OUTPUT(IJK)
C                                                      REV 12 12/17/74
C
C     CONTROLS TABULATED OUTPUT ON FORTRAN UNITS (STARTING WITH NO. 21)
C     OF SELECTED OPTIONAL SEGMENT LINEAR AND ANGULAR ACCELERATIONS,
C     VELOCITIES AND DISPLACEMENTS, JOINT PARAMETERS AND SELECTED DATA
C     FROM ALL ALLOWED CONTACT FORCE COMPUTATIONS BETWEEN BODY SEGMENTS
C     AND VEHICLE COMPONENTS.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/CONTRL/NSEG,NJNT,NS3,NJ3,NPL,NBLT,NBAG,NVEH,NGRND,NPRT(40)
      COMMON/JBARTZ/ MNPL(  20),MNBLT(  8),MNSEG(  22),MNBAG(   6),
     *             MPL(3,5,20),MBLT(3,5,8),MSEG(3,5,22),MBAG(3,10,6),
     *               NTPL(5,20),NTBLT(5,8),NTSEG(5,22)
      COMMON/SGMNTS/D(3,3,22),WMEG(3,22),WMEGD(3,22),U1(3,22),U2(3,22)
     *              ,SEGLP(3,22),SEGLV(3,22),SEGLA(3,22),NSYM(22)
      COMMON/DESCRP/ PHI(3,22),W(22),SR(3,42),HA(3,42),HB(3,42)
     *              ,HT(3,3,42),RPHI(3,22),RW(22),SPRING(5,63)
     *              ,VISC(7,63),JNT(21),IPIN(21),NS,ISING(22)
      COMMON/CNSNTS/ PI, RADIAN,G,THIRD,EPS1,EPS4,EPS6,EPS8,
     *               EPS12,EPS15,EPS20,EPS24,UNITL,UNITM,UNITT,GRAVTY(3)
      COMMON/VPOSTN/ TIME,X0(3),XDOT0(3),XCOMP(3),XVCOMP(3),AX(3),
     *              ANGLE(3),VMPH,VTIME,ATAB(15,100),AT0,ADT,OMEGA,
     *              NATAB,NACLR,DVEH(3,3),VMEG(3),VMEGD(3),XACOMP(3),
     *              THET(3),ZPLT(3)
      COMMON/FORCES/PSF(7,20),BSF(4,20),SSF(10,20),BAGSF(3,20),
     *              NPSF,NBSF,NSSF,NBGSF,NPANEL(6),PRJNT(6,21)
      COMMON/TEMPVS/ ACC(6,20),T1(3),T2(3),T3(3),T4(3)
      COMMON/RSAVE/ XSG(3,20,3),NSG(7),MSG(20,7)
      DATA LINES/0/,LPP/45/
C
      IF (IJK.NE.0) GO TO 9
C
C     SET ALL FORCE ARRAYS TO ZERO.
C
      DO  8 I=1,480
    8 PSF(I,1) = 0.0
      DO 7 J=1,NJNT
      PRJNT(1,J) = 1.0
      PRJNT(2,J) = 1.0
      IF  (IABS(IPIN(J)).EQ.4)  PRJNT(1,J)  = 0.0
      IF  (IABS(IPIN(J)).EQ.4)  PRJNT(2,J)  = 0.0
      PRJNT(3,J)  = 0.0
      PRJNT(4,J)  = 0.0
      PRJNT(5,J)  = 0.0
    7 PRJNT(6,J)  = 0.0
      RETURN
C
    9 CALL ELTIME(1,8)
C
C     INCREMENT LINE COUNT AND TEST FOR START OF NEW PAGE ON EACH UNIT.
```

```
C
      LINES = LINES+1
      IF (MOD(LINES,LPP).NE.1) GO TO 51
      CALL HEDING(LINES,LPP,MPSF,MBSF,MSSF)
C
C     PRINT LINE OF DATA FOR THIS TIME POINT ON EACH OUTPUT UNIT (NT).
C
   51 USEC = 1000.0*TIME
      NT = 20
C
C     COMPUTE AND PRINT DATA FOR 7 TYPES OF OUTPUT ABOVE
C
      DO 68 K=1,7
      IF (NSG(K).LE.0) GO TO 68
      KSG = NSG(K)
      J3  = 3
      IF (K.EQ.7) J3 = 2
      DO 67 J1=1,KSG,J3
      J2 = MIN0(J1+J3-1,KSG)
      NT = NT+1
      DO 66 J=J1,J2
      L  = MSG(J,K)
      GO TO (52,54,56,59,61,64,65),K
C
C     1. SEGMENT LINEAR ACCELERATIONS IN LOCAL REFERENCE
C
   52 CALL CROSS (WMEG(1,L),XSG(1,J,K),T1)
      CALL CROSS (WMEG(1,L),T1,T2)
      CALL CROSS (WMEGD(1,L),XSG(1,J,K),T3)
      CALL MAT(D(1,1,L),SEGLA(1,L),T4,3,3,1,3,3,3)
      DO 53 I=1,3
   53 ACC(I,J) = (T4(I)+T3(I)+T2(I))/G
      GO TO 63
C
C     2. SEGMENT LINEAR VELOCITIES IN VEHICLE REFERENCE
C
   54 CALL CROSS (WMEG(1,L),XSG(1,J,K),T1)
      CALL DOT(D(1,1,L),T1,T2,3,1,3)
      DO 55 I=1,3
   55 T3(I) = SEGLV(I,L)+T2(I)-XVCOMP(I)
      GO TO 58
C
C     3. SEGMENT LINEAR DISPLACEMENTS IN VEHICLE REFERENCE
C
   56 CALL DOT(D(1,1,L),XSG(1,J,K),T1,3,1,3)
      DO 57 I=1,3
   57 T3(I) = SEGLP(I,L)+T1(I)-XCOMP(I)
   58 CALL MAT (DVEH,T3,ACC(1,J),3,3,1,3,3,6)
      GO TO 63
C
```

```
C     4. SEGMENT ANGULAR ACCELERATIONS IN LOCAL REFERENCE
C
   59 DO 60 I=1,3
   60 ACC(I,J) = WMEGD(I,L)/(2.0*PI)
      GO TO 63
C
C     5. SEGMENT ANGULAR VELOCITIES IN VEHICLE REFERENCE
C
   61 CALL DOT (D(1,1,L),WMEG(1,L),T1,3,1,3)
      CALL MAT (DVEH,T1,T2,3,3,1,3,3,3)
      DO 62 I=1,3
   62 ACC(I,J) = (T2(I)-VMEG(I))/(2.0*PI)
   63 ACC(4,J) = DSQRT(ACC(1,J)**2+ACC(2,J)**2+ACC(3,J)**2)
      GO TO 66
C
C     6. SEGMENT ANGULAR DISPLACEMENTS IN VEHICLE REFERENCE
C
   64 CALL DOTT (D(1,1,L),DVEH,T1,3,3,3)
      CALL YPRDEG(T1,ACC(1,J))
      TRACE = 0.5*(T1(1)+T2(2)+T3(3)-1.0)
      IF (TRACE.GT. 1.0) TRACE =  1.0
      IF (TRACE.LT.-1.0) TRACE = -1.0
      ACC(4,J) = DARCOS(TRACE)/RADIAN
      GO TO 66
C
C     7. JOINT PARAMETERS
C
   65 ACC(1,J) = PRJNT(1,L)/RADIAN
      ACC(2,J) = PRJNT(2,L)/RADIAN
      ACC(3,J) = PRJNT(3,L)
      ACC(4,J) = PRJNT(4,L)
      ACC(5,J) = PRJNT(5,L)
      ACC(6,J) = PRJNT(6,L)
   66 CONTINUE
      IF (K.LE.6) WRITE (NT,121) USEC,((ACC(I,J),I=1,4),J=J1,J2)
  121 FORMAT(F9.3,3(3X,4F9.2) )
   67 IF (K.EQ.7) WRITE (NT,123) USEC,((ACC(I,J),I=1,6),J=J1,J2)
  123 FORMAT(F9.3,2(2X,0P2F7.2,1P4D11.4)  )
   68 CONTINUE
C
C     PRINT PLANE FORCES
C
      IF (MPSF.EQ.0) GO TO 77
      DO 76 J1=1,MPSF,2
      J2 = MIN0(J1+1,MPSF)
      NT = NT+1
   76 WRITE (NT,129) USEC,((PSF(I,J),I=1,7),J=J1,J2)
  129 FORMAT(F9.3,2(F9.3,3F9.2,3F8.2) )
C
C     PRINT BELT FORCES
```

```
C
   77 IF (MBSF.EQ.0) GO TO 79
      DO 78 J1=1,MBSF,2
      J2 = MINO(J1+1,MBSF)
      NT = NT+1
   78 WRITE (NT,135) USEC,((BSF(I,J),I=1,4),J=J1,J2)
  135 FORMAT(F9.3,4(F15.6,F12.2,3X) )
C
C     PRINT SEGMENT CONTACT FORCES
C
   79 IF (MSSF.EQ.0) GO TO 81
      DO 80 J=1,MSSF
      NT = NT+1
   80 WRITE (NT,37) USEC,(SSF(I,J),I=1,10)
   37 FORMAT(2F9.3,3F9.2,3F8.2,2X,3F8.2)
C
C     PRINT AIRBAG FORCES
C
   81 IF (NBAG.EQ.0) GO TO 91
      K1 = 1
      DO 83 J=1,NBAG
      IF (MNBAG(J).EQ.0) GO TO 83
      KBAG = MNBAG(J)+NPANEL(J)+5
      DO 82 J1=1,KBAG,4
      J2 = MINO(J1+3,KBAG)
      K2 = K1+J2-J1
      NT = NT+1
      WRITE (NT,21) USEC,((BAGSF(I,K),I=1,3),K=K1,K2)
   21 FORMAT(F9.3,4(3X,3F9.2))
   82 K1 = K2+1
   83 CONTINUE
   91 CONTINUE
C
      CALL ELTIME(2,8)
      RETURN
      END
```

```
      SUBROUTINE PLELP(M,MM,N,NN,NT)
C                                                      REV 12 11/25/74
C     COMPUTES FORCES (WHICH ARE ADDED TO U1 ARRAY)
C      AND TORQUES    (WHICH ARE ADDED TO U2 ARRAY)
C     OF ELLIPSOID (MM) ATTACHED TO BODY SEGMENT (M)
C     INTERSECTING PLANE (NN) ATTACHED TO SEGMENT (N).
C
      IMPLICIT REAL *8(A-H,O-Z)
      COMMON/TABLES/MXNT1,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(500),TAB(2000)
      COMMON/SGMNTS/D(3,3,22),WMEG(3,22),WMEGD(3,22),U1(3,22),U2(3,22)
     *              ,SEGLP(3,22),SEGLV(3,22),SEGLA(3,22),NSYM(22)
      COMMON/FORCES/PSF(7,20),BSF(4,20),SSF(10,20),BAGSF(3,20),
     *              NPSF,NBSF,NSSF,NBGSF,NPANEL(6),PRJNT(6,21)
      COMMON/CNTSRF/ PL(17,20),GAB(8,3),BELT(20,8),TPTS(6,8),BD(24,25)
      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)
     *             ,NQ,KQ1(12),KQ2(12),KQTYPE(12)
      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,
     *              VRT,VRTS,TF,MCF,NCF
      CALL ELTIME(1,21)
C
C     COMPUTE PENETRATION DISTANCE, IF NEGATIVE, RETURN.
C
      CALL DOTT(D(1,1,M),D(1,1,N),DMNT,3,3,3)
      DO 10 I=1,3
   10 XMN(I) = SEGLP(I,M) - SEGLP(I,N)
      CALL MAT(D(1,1,M),XMN,XMM,3,3,1,3,3,3)
      CALL MAT(DMNT,PL(1,NN),TM,3,3,1,3,3,3)
      BET = PL(4,NN)
      DO 11 I=1,3
   11 BET = BET - TM(I)*(BD(I+3,MM)+XMM(I))
      CALL MAT(BD(16,MM),TM,RM,3,3,1,3,3,3)
      BTS = TM(1)*RM(1) + TM(2)*RM(2) + TM(3)*RM(3)
      BTE = -DSQRT(BTS)
      P = BET - BTE
      MCF = NTAB(NT+1)
      NCF = -MCF
      IF (NCF.GT.0) CFQQ(NCF) = -999.
      IF (P.LE.0.0) GO TO 99
C
C     IF COMPLETE PENETRATION, RETURN
C
      IF (BET+BTE.GT.0.0) GO TO 99
C
C     COMPUTE TG - THE POINT IN SEGMENT REFERENCE AT WHICH THE CONTACT
```

```
C                     FORCES ARE TO BE APPLIED WHICH LIES ON THE SCALED
C                     LINE BETWEEN THE POINT OF MAXIMUM PENETRATION (RHO=0)
C                     AND THE CENTER OF THE INTERSECTION ELLIPSE (RHO=1).
C         AND TEMP  - THE SAME POINT IN VEHICLE REFERENCE.
C
      RHO = 0.0
      IF (MCF.GT.0) RHO = TAB(MCF+4)
      BETE = (1.0+RHO*P/BTE)/BTE
      AMR  = -1.0/BTE
      DO 13 I=1,3
      RM(I) = BETE*RM(I)
      RLM(I) = RM(I) + BD(I+3,MM)
   13 RN(I)  = RLM(I) + XMM(I)
      CALL DOT(DMNT,RN,RLN,3,1,3)
C
C     IF BOUNDARY PLANE IS GIVEN, COMPUTE DISTANCE FROM POINT TO PLANE,
C         IF NEGATIVE OR > LIMIT, RETURN.
C
      DO 14 I=8,13,5
      IF (PL(I+4,NN).LE.0.0) GO TO 14
      DIST = RLN(1)*PL(I  ,NN)
     *     + RLN(2)*PL(I+1,NN)
     *     + RLN(3)*PL(I+2,NN) - PL(I+3,NN)
      IF (DIST.LE.0.0 .OR. DIST.GT.PL(I+4,NN)) GO TO 99
   14 CONTINUE
      CALL PLSEGF(M,N,NT)
      IF (MCF.LT.0) GO TO 30
C
C     STORE RESULTS FOR OUTPUT ROUTINE.
C
      PSF(1,NPSF) = P
      PSF(2,NPSF) = FM
      PSF(3,NPSF) = FM*CF
      IF (VRT.EQ.1.0) PSF(3,NPSF) = FM*CF*VRTS
      PSF(4,NPSF) = TF
      DO 24 I=1,3
   24 PSF(I+4,NPSF) = RLN(I)
      GO TO 99
   30 PSF(1,NPSF) = P
      DO 31 I=1,3
      PSF(I+1,NPSF) = T(I)
   31 PSF(I+4,NPSF) = RLN(I)
      CALL CROSS(WMN,TM,T1)
      CALL MAT(BD(16,MM),T1,T2,3,3,1,3,3,3)
      TMT = TM(1)*T2(1) + TM(2)*T2(2) + TM(3)*T2(3)
      TMT = TMT/BTE
      DO 32 I=1,3
   32 RMD(I) = (T2(I)-TMT*RM(I))*BETE
      CALL CROSS(DMNWN,VREL,T1)
      CALL CROSS(WMN,RMD,T3)
```

```
      CALL DOT(D(1,1,M),T3,RQQ(1,NCF),3,1,3)                           PLEL1010
      SQQ(NCF) = 0.0                                                   PLEL1020
      DO 36 I=1,3                                                      PLEL1030
   36 SQQ(NCF) = SQQ(NCF) + TM(I)*(T3(I)+2.0*T1(I))                    PLEL1040
   99 CALL ELTIME(2,21)                                                PLEL1050
      RETURN                                                           PLEL1060
      END                                                              PLEL1070
```

```
      SUBROUTINE RSTART(IF,IT)
C                                                 REV 12 12/19/74
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
      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/NSEG,NJNT,NS3,NJ3,NPL,NBLT,NBAG,NVEH,NGRND,NPRT(40)
      DIMENSION IC1(49)
      EQUIVALENCE (IC1(1),NSEG)
C  2
      COMMON/CNTSRF/ PL(17,20),GAB(8,3),BELT(20,8),TPTS(6,8),BD(24,25)
      DIMENSION RC2(1172)
      EQUIVALENCE (RC2(1),PL(1,1))
C  3
      COMMON/VPOSTN/ TIME,X0(3),XDOT0(3),XCOMP(3),XVCOMP(3),AX(3),
     *              ANGLE(3),VMPH,VTIME,ATAB(15,100),AT0,ADT,OMEGA,
     *              NATAB,NACLR,DVEH(3,3),VMEG(3),VMEGD(3),XACOMP(3),
     *              THET(3),ZPLT(3)
      DIMENSION RC3(1527),RC3A(1511),IC3(2),RC3B(18)
      EQUIVALENCE (RC3(1),TIME),(RC3A(1),AX(1)),
     *            (IC3(1),NATAB),(RC3B(1),DVEH(1,1))
C  4
      COMMON/SGMNTS/D(3,3,22),WMEG(3,22),WMEGD(3,22),U1(3,22),U2(3,22)
     *              ,SEGLP(3,22),SEGLV(3,22),SEGLA(3,22),NSYM(22)
      DIMENSION RC4(660)
      EQUIVALENCE (RC4(1),D(1,1,1))
C  5
      COMMON/CMATRX/V1(3,21),V2(3,21),V3(3,12),B12(3,3,42),A22(3,3,42)
     *              ,F(3,21),TQ(3,21),WJ(21)
      DIMENSION RC5A(918),RC5B(147)
      EQUIVALENCE (RC5A(1),V1(1,1)),(RC5B(1),F(1,1))
C  6
      COMMON /ABDATA/ ABC(3,5), ZA(3,5), DA(3,3,5), BFA(3,5)
     *              ,BCGV(3,5),BMEG(3,5)
      DIMENSION RC6(120)
      EQUIVALENCE (RC6(1),ABC(1,1))
C  7
      COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),BLTTTL(5,8)
     *              ,PLTTL(5,20),BAGTTL(5,6),SEG(22),JOINT(21)
     *              ,CGS(21),JS(21)
      REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT
      LOGICAL*1 CGS,JS
```

```
      REAL   RC7,RC7A,XDTE,XCMENT                                       RSTA0510
      DIMENSION RC7(238),RC7A(281),XDTE(3),XCMENT(40)                   RSTA0520
      EQUIVALENCE (RC7(1),VPSTTL(1)),(RC7A(1),DATE(1))                  RSTA0530
C  8                                                                    RSTA0540
      COMMON/CNSNTS/ PI, RADIAN,G,THIRD,EPS1,EPS4,EPS6,EPS8,            RSTA0550
     *               EPS12,EPS15,EPS20,EPS24,UNITL,UNITM,UNITT,GRAVTY(3)RSTA0560
      DIMENSION RC8(18)                                                 RSTA0570
      EQUIVALENCE (RC8(1),PI)                                           RSTA0580
C  9                                                                    RSTA0590
      COMMON/DESCRP/ PHI(3,22),W(22),SR(3,42),HA(3,42),HB(3,42)         RSTA0600
     *              ,HT(3,3,42),RPHI(3,22),RW(22),SPRING(5,63)          RSTA0610
     *              ,VISC(7,63),JNT(21),IPIN(21),NS,ISING(22)           RSTA0620
     *              ,IGL08(21),JOINTF(21)                               RSTA0630
      DIMENSION RC9(1688),IC9(107)                                      RSTA0640
      EQUIVALENCE (RC9(1),PHI(1,1)),(IC9(1),JNT(1))                     RSTA0650
C 10                                                                    RSTA0660
      COMMON/J8ARTZ/ MNPL(   20),MN8LT(   8),MNSEG(   22),MN8AG(    6), RSTA0670
     *             MPL(3,5,20),M8LT(3,5,8),MSEG(3,5,22),MBAG(3,10,6),   RSTA0680
     *               NTPL(5,20),NT8LT(5,8),NTSEG(5,22)                  RSTA0690
      DIMENSION IC10(1236)                                              RSTA0700
      EQUIVALENCE (IC10(1),MNPL(1))                                     RSTA0710
C 11                                                                    RSTA0720
      COMMON/FORCES/PSF(7,20),BSF(4,20),SSF(10,20),BAGSF(3,20),         RSTA0730
     *              NPSF,NBSF,NSSF,NBGSF,NPANEL(6),PRJNT(6,21)          RSTA0740
      DIMENSION RC11(480),IC11(4),RC11A(126)                            RSTA0750
      EQUIVALENCE (RC11(1),PSF(1,1)),(IC11(1),NPSF),                    RSTA0760
     *            (RC11A(1),PRJNT(1,1))                                 RSTA0770
C 12                                                                    RSTA0780
      COMMON/INTEST/ SGTEST(3,4,22),XTEST(3,88)                         RSTA0790
      DIMENSION RC12(528)                                               RSTA0800
      EQUIVALENCE (RC12(1),SGTEST(1,1,1))                               RSTA0810
C 13                                                                    RSTA0820
      COMMON/CSTRNT/A13(3,3,24),A23(3,3,24),831(3,3,24),B32(3,3,24)     RSTA0830
     *             ,HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12)  RSTA0840
     *             ,RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12)                RSTA0850
     *             ,NQ,KQ1(12),KQ2(12),KQTYPE(12)                       RSTA0860
      DIMENSION RC13(72),IC13(37),RC13A(1212),RC13H(348)                RSTA0870
      EQUIVALENCE (RC13(1),RK1(1,1)),(IC13(1),NQ),(RC13A(1),A13(1,1,1)) RSTA0880
     *           ,(RC13H(1),HHT(1,1,1))                                 RSTA0890
C 14                                                                    RSTA0900
      COMMON/TA8LES/MXNTI,MXNTB,MXT81,MXT82,NTI(50),NTA8(500),TAB(2000) RSTA0910
      DIMENSION IC14(554)                                               RSTA0920
      EQUIVALENCE (IC14(1),MXNTI)                                       RSTA0930
C 15                                                                    RSTA0940
      COMMON/COMAIN/VAR(120),DER(120),DT,HO,HMAX,HMIN,RSTIME,           RSTA0950
     *             ISTEP,NSTEPS,NDINT,NEQ,IRSIN,IRSOUT                  RSTA0960
      DIMENSION RC15(245),IC15(6)                                       RSTA0970
      EQUIVALENCE (RC15(1),VAR(1)),(IC15(1),ISTEP)                      RSTA0980
C 16                                                                    RSTA0990
      COMMON/CDINT/ E(3,120),FF(5,120),GG(5,120),Y(5,120),U(5,120)      RSTA1000
```

```
     *              ,H,HPRINT,TSAVE,TPRINT,TSTART,ICNT,IDBL,IFLAG           RSTA1010
C       NOTE: FF REPLACES F FROM SUBROUTINE DINT.                           RSTA1020
      DIMENSION RC16(2765),IC16(3)                                          RSTA1030
      EQUIVALENCE (RC16(1),E(1,1)),(IC16(1),ICNT)                           RSTA1040
C 17                                                                        RSTA1050
      COMMON/DAMPER/APSOM(3,20),APSON(3,20),ASD(5,20),                      RSTA1060
     *              NSO,MSDM(20),MSON(20)                                   RSTA1070
      DIMENSION RC17(220),IC17(41)                                          RSTA1080
      EQUIVALENCE (RC17(1),APSDM(1,1)),(IC17(1),NSO)                        RSTA1090
C 18                                                                        RSTA1100
      COMMON/CEULER/ IEULER(22),HIR(3,3,21),ANG(3,21),ANGO(3,21),           RSTA1110
     *               FE(3,21),TQE(3,21),CONST(3,21)                         RSTA1120
      DIMENSION RC18(504)                                                   RSTA1130
      EQUIVALENCE (RC18(1),HIR(1,1,1))                                      RSTA1140
C 19                                                                        RSTA1150
      COMMON/TEMPVI/ TT1(3),R11(3),R2I(3),CREST,JSTOP(4,2,21)               RSTA1160
      DIMENSION RC19(10),IC19(168)                                          RSTA1170
      EQUIVALENCE (RC19(1),TTI(1)),(IC19(1),JSTOP(1,1,1))                   RSTA1180
C 20                                                                        RSTA1190
      COMMON /WJONES/                                                       RSTA1200
     *          FORCE(3,5),TORA(3,5),XBM(5),ZOEP(3,5),VBAGG(5),VSCS(5),     RSTA1210
     *  BPHI(3,5),DBR(3,3,5),DPVCTR(3,5),DEPLOY(3,5),AB(3,5),SPRK(5),       RSTA1220
     *          CYTD(5),CYPA(5),CYSP(5),CYTO(5),                            RSTA1230
     *  CYVO(5), CYCO(5), CYK(5), CYR(5), CYAT(5), CYPV(5), CYCDO(5),       RSTA1240
     *  CYAO(5), CYPO(5), CYSS(5), CYLO(5), CYC(5), CYRHOO(5),CYVMAX(5),RSTA1250
     *  CYORFC(5),CYRHO(5),CYT(5),CYP(5),CYMIN(5),CYMOUT(5),                RSTA1260
     *          BAGPV(5),PD(5),VBAG(5),VOLBP(5),SWITCH(5),IFULL(6),         RSTA1270
     *          TMP(18),TMP1(3),A(3,3),PF(3),TORQ(3) ,                      RSTA1280
     *          TQB(3,10),FRB(3,10),VOL(10),OELF(3),                        RSTA1290
     *          B(9,4,5),ZB(3,4,5),ZR(3,4,5),BFB(9,4,5),DRR(9,4,5),         RSTA1300
     *  OB(9,4,5),PCGV(3,4,5),PMEG(3,4,5),VOLP(4,5),FRA(3,4),PREVT          RSTA1310
     *      , CK(5),CMASS(5)                                                RSTA1320
      DIMENSION RC20A(30),RC20B(235),RC20C(50),RC20D(109),RC20E(180),       RSTA1330
     *          RC20F(60),RC20G(420),RC20H(320),RC20I(10)                   RSTA1340
      EQUIVALENCE (RC20A(1),FORCE(1,1)),(RC20B(1),XBM(1)),                  RSTA1350
     *             (RC20C(1),CYRHO(1))   ,(RC20D(1),TMP(1)),                RSTA1360
     *             (RC20E(1),B(1,1,1))   ,(RC20F(1),ZB(1,1,1)),             RSTA1370
     *             (RC20G(1),ZR(1,1,1))  ,(RC20H(1),DB(1,1,1)),             RSTA1380
     *             (RC20I(1),CK(1))                                         RSTA1390
C 21                                                                        RSTA1400
      COMMON/RSAVE/ XSG(3,20,3),NSG(7),MSG(20,7)                            RSTA1410
      DIMENSION RC21(180),IC21(147)                                         RSTA1420
      EQUIVALENCE (RC21(1),XSG(1,1,1)) , (IC21(1),NSG(1))                   RSTA1430
C 22                                                                        RSTA1440
      COMMON/FLXBLE/ HF(4,12,8),B42(3,3,24),V4(3,8),NFLEX(3,8),NFLX         RSTA1450
      DIMENSION RC22(624),IC22(25)                                          RSTA1460
      EQUIVALENCE (RC22(1),HF(1,1,1)) , (IC22(1),NFLEX(1,1))                RSTA1470
C 23                                                                        RSTA1480
      COMMON/HRNESS/ BAR(6,100) , XLONG(20) , IBAR(2,100) , NTHRNS(20) ,RSTA1490
     *               NHRNSS , NBLTPH(5) , NFNBL(5,20) , NPTSPB(20)          RSTA1500
```

```
      DIMENSION RC23(620) , IC23(346)
      EQUIVALENCE (RC23(1),BAR(1,1)) , (IC23(1),IBAR(1,1))
C 24
      COMMON/KALEPS/ WTIME(30),IWIND(30),MWSEG(5,22)
      DIMENSION RC24(30) , IC24(140)
      EQUIVALENCE (RC24(1),WTIME(1)) , (IC24(1),IWIND(1))
C
      DIMENSION COMMON(24)
      DATA COMMON /8HCONTRL  ,8HCNTSRF  ,8HVPOSTN  ,8HSGMNTS  ,
     *             8HCMATRX  ,8HABDATA  ,8HTITLES  ,8HCNSNTS  ,
     *             8HDESCRP  ,8HJBARTZ  ,8HFORCES  ,8HINTEST  ,
     *             8HCSTRNT  ,8HTABLES  ,8HCOMAIN  ,8HCDINT   ,
     *             8HDAMPER  ,8HCEULER  ,8HTEMPVI  ,8HWJONES  ,
     *             8HRSAVE   ,8HFLXBLE  ,8HHRNESS  ,8HKALEPS  /
      REAL AOLD4,AAOLD4
      DATA BLANK/8H        /
      DIMENSION INDEX(3)
      CALL ELTIME(1,25)
      GO TO (100,200,300,400,500),IF
C
C        1. READ INPUT & INITIALIZATION RECORD FROM OLD RESTART TAPE.
C
  100 READ   (IT) IC1,PL,BD,XO,XDOTO,RC3A,NATAB,ZPLT,NSYM,XDTE,XCMENT,
     *            RC7,CGS,JS,RC8,RC9,JNT,IC10,NPANEL,SGTEST,RC13,IC13,
     *            IC14,DT,HO,HMAX,HMIN,NSTEPS,NDINT,RC17,IC17,IEULER,
     *            RC20B,IFULL,RC20E,RC20G,RC20I,RC21,IC21,NS,ISING
     *            ,HF,NFLEX,NFLX,IC19,IGLOB,JOINTF,RC23,IC23,RC24,IC24
      WRITE (6,101) IT,XDTE,XCMENT
  101 FORMAT('0 INPUT DATA HAS BEEN READ IN FROM UNIT NO.',I4//
     *       10X,3A4//10X,20A4/10X,20A4)
      GO TO 999
C
C        2. WRITE INPUT & INITIALIZATION RECORD ONTO NEW RESTART TAPE.
C
  200 WRITE  (IT) IC1,PL,BD,XO,XDOTO,RC3A,NATAB,ZPLT,NSYM,DATE,COMENT,
     *            RC7,CGS,JS,RC8,RC9,JNT,IC10,NPANEL,SGTEST,RC13,IC13,
     *            IC14,DT,HO,HMAX,HMIN,NSTEPS,NDINT,RC17,IC17,IEULER,
     *            RC20B,IFULL,RC20E,RC20G,RC20I,RC21,IC21,NS,ISING
     *            ,HF,NFLEX,NFLX,IC19,IGLOB,JOINTF,RC23,IC23,RC24,IC24
      GO TO 999
C
C        3. READ TIME POINT RECORD FROM OLD RESTART TAPE.
C
  300 READ   (IT) TIME,BELT,TPTS,XCOMP,XVCOMP,RC3B,RC4,RC5B,RC6,IPIN,RC11
     *            ,IC11,PRJNT,TAB,RC16,IC16,RC20A,RC20C,IFULL,RC20H,PREVT
     *            ,RC21,IC21,VAR,DER,NEQ,XTEST,V4,IC19,RC13H,KQTYPE
     *            ,IEULER,RC23,WTIME,IWIND
      CALL OUTPUT(1)
      GO TO 999
C
```

```
C         5. WRITE TIME POINT RECORD ONTO NEW RESTART TAPE.
C
  500 WRITE (IT) TIME,BELT,TPTS,XCOMP,XVCOMP,RC3B,RC4,RC5B,RC6,IPIN,RC11
     *           ,IC11,PRJNT,TAB,RC16,IC16,RC20A,RC20C,IFULL,RC20H,PREVTR
     *          ,RC21,IC21,VAR,DER,NEQ,XTEST,V4,IC19,RC13H,KQTYPE
     *           ,IEULER,RC23,WTIME,IWIND
      GO TO 999
C
C         4. READ NEW INPUT DATA FROM INPUT STREAM FOR RESTART.
C
  400 READ (5,401) AVAR,INDEX,ITYPE,RR,II,AA,RROLD,IIOLD,AAOLD
  401 FORMAT(A8,4I4,2(F8.0,I8,A8))
      CALL SEARCH(AVAR,INDEX,NCOM,ITEM)
      IF (NCOM.LE.0)    GO TO 490
      IF (NCOM.GT.24) GO TO 999
      IF (ITYPE.GT.3)   GO TO 490
      GO TO ( 1, 2, 3, 4, 5, 6, 7, 8, 9,10,11,12,
     *        13,14,15,16,17,18,19,20,21,22,23,24),NCOM
C     COMMON /CONTRL/
    1 IF (ITEM.GT.49)   GO TO 490
      IF (ITYPE.NE.2)   GO TO 490
      IOLD = IC1(ITEM)
      IC1(ITEM) = II
      GO TO 494
C     COMMON /CNTSRF/
    2 IF (ITEM.GT.1172) GO TO 490
      IF (ITYPE.NE.1)   GO TO 490
      ROLD = RC2(ITEM)
      RC2(ITEM) = RR
      GO TO 492
C     COMMON /VPOSTN/
    3 IF (ITEM.GT.1527) GO TO 402
      IF (NTYPE.NE.1)   GO TO 490
      ROLD = RC3(ITEM)
      RC3(ITEM) = RR
      GO TO 492
  402 IF (ITEM.GT.1529) GO TO 403
      IF (NTYPE.NE.2)   GO TO 490
      IOLD = IC3(ITEM-1527)
      IC3(ITEM-1527) = II
      GO TO 494
  403 IF (ITEM.GT.1553) GO TO 490
      IF (NTYPE.NE.1)   GO TO 490
      ROLD = RC3B(ITEM-1529)
      RC3B(ITEM-1529) = RR
      GO TO 492
C     COMMON /SGMNTS/
    4 IF (ITEM.GT.660 ) GO TO 404
      IF (ITYPE.NE.1)   GO TO 490
      ROLD = RC4(ITEM)
```

```
      RC4(ITEM) = RR
      GO TO 492
  404 IF (ITEM.GT.682 ) GO TO 490
      IF (ITYPE.NE.I)   GO TO 490
      IOLD = NSYM(ITEM-660)
      NSYM(ITEM-660) = II
      GO TO 494
C     COMMON /CMATRX/
    5 IF (ITEM.GT.1065) GO TO 490
      IF (ITYPE.NE.1)   GO TO 490
      ROLD = RC5A(ITEM)
      RC5A(ITEM) = RR
      GO TO 492
C     COMMON /ABDATA/
    6 IF (ITEM.GT.120 ) GO TO 490
      IF (ITYPE.NE.1)   GO TO 490
      ROLD = RC6(ITEM)
      RC6(ITEM) = RR
      GO TO 492
C     COMMON /TITLES/ NOTE: NO PROVISION FOR CGS OR JS.
    7 IF (ITEM.GT.28I ) GO TO 490
      IF (ITYPE.NE.3)   GO TO 490
      AOLD = RC7A(ITEM)
      RC7A(ITEM) = AA
      GO TO 496
C     COMMON /CNSNTS/
    8 IF (ITEM.GT.I5   ) GO TO 408
      IF (ITEM.LE.I2   ) GO TO 408
      IF (ITYPE.NE.3)   GO TO 490
      AOLD = RC8(ITEM)
      RC8(ITEM) = AA
      GO TO 496
  408 IF (ITYPE.NE.I)   GO TO 490
      ROLD = RC8(ITEM)
      RC8(ITEM) = RR
      GO TO 492
C     COMMON /DESCRP/
    9 IF (ITEM.GT.I688) GO TO 409
      IF (ITYPE.NE.1)   GO TO 490
      ROLD = RC9(ITEM)
      RC9(ITEM) = RR
      GO TO 492
  409 IF (ITEM.GT.I795) GO TO 490
      IF (ITYPE.NE.2)   GO TO 490
      IOLD = IC9(ITEM-I688)
      IC9(ITEM-1688) = II
      GO TO 494
C     COMMON /JBARTZ/
   10 IF (ITEM.GT.1236) GO TO 490
      IF (ITYPE.NE.2)   GO TO 490
```

```
      IOLD = IC10(ITEM)                                                 RSTA3010
      IC10(ITEM) = II                                                   RSTA3020
      GO TO 494                                                         RSTA3030
C     COMMON /FORCES/                                                   RSTA3040
   11 IF (ITEM.GT.480)  GO TO 411                                       RSTA3050
      IF (NTYPE.NE.1)   GO TO 490                                       RSTA3060
      ROLD = RC11(ITEM)                                                 RSTA3070
      RC11(ITEM) = RR                                                   RSTA3080
      GO TO 492                                                         RSTA3090
  411 IF (ITEM.GT.490)  GO TO 412                                       RSTA3100
      IF (NTYPE.NE.2)   GO TO 490                                       RSTA3110
      IOLD = IC11(ITEM-480)                                             RSTA3120
      IC11(ITEM-480) = II                                               RSTA3130
      GO TO 494                                                         RSTA3140
  412 IF (ITEM.GT.616)  GO TO 490                                       RSTA3150
      IF (NTYPE.NE.1)   GO TO 490                                       RSTA3160
      ROLD = RC11A(ITEM-490)                                            RSTA3170
      RC11A(ITEM-490) = RR                                              RSTA3180
      GO TO 492                                                         RSTA3190
C     COMMON /INTEST/                                                   RSTA3200
   12 IF (ITEM.GT.528 ) GO TO 490                                       RSTA3210
      IF (ITYPE.NE.1)   GO TO 490                                       RSTA3220
      ROLD = RC12(ITEM)                                                 RSTA3230
      RC12(ITEM) = RR                                                   RSTA3240
      GO TO 492                                                         RSTA3250
C     COMMON /CSTRNT/                                                   RSTA3260
   13 IF (ITEM.GT.1212) GO TO 413                                       RSTA3270
      IF (ITYPE.NE.1)   GO TO 490                                       RSTA3280
      ROLD = RC13A(ITEM)                                                RSTA3290
      RC13A(ITEM) = RR                                                  RSTA3300
      GO TO 492                                                         RSTA3310
  413 IF (ITEM.GT.1249) GO TO 490                                       RSTA3320
      IF (ITYPE.NE.2)   GO TO 490                                       RSTA3330
      IOLD = IC13(ITEM-1212)                                            RSTA3340
      IC13(ITEM-1212) = II                                              RSTA3350
      GO TO 494                                                         RSTA3360
C     COMMON /TABLES/                                                   RSTA3370
   14 IF (ITEM.GT.554 ) GO TO 414                                       RSTA3380
      IF (ITYPE.NE.2)   GO TO 490                                       RSTA3390
      IOLD = IC14(ITEM)                                                 RSTA3400
      IC14(ITEM) = II                                                   RSTA3410
      GO TO 494                                                         RSTA3420
  414 IF (ITEM.GT.2554) GO TO 490                                       RSTA3430
      IF (ITYPE.NE.1)   GO TO 490                                       RSTA3440
      ROLD = TAB(ITEM-554)                                              RSTA3450
      TAB(ITEM-554) = RR                                                RSTA3460
      GO TO 492                                                         RSTA3470
C     COMMON /COMAIN/                                                   RSTA3480
   15 IF (ITEM.GT.245 ) GO TO 415                                       RSTA3490
      IF (ITYPE.NE.1)   GO TO 490                                       RSTA3500
```

```
      ROLD = RC15(ITEM)                                                 RSTA3510
      RC15(ITEM) = RR                                                   RSTA3520
      GO TO 492                                                         RSTA3530
  415 IF (ITEM.GT.251 ) GO TO 490                                       RSTA3540
      IF (ITYPE.NE.2)   GO TO 490                                       RSTA3550
      IOLD = IC15(ITEM-245)                                             RSTA3560
      IC15(ITEM-245) = II                                               RSTA3570
      GO TO 494                                                         RSTA3580
C     COMMON /CDINT /                                                   RSTA3590
   16 IF (ITEM.GT.2765) GO TO 416                                       RSTA3600
      IF (ITYPE.NE.1)   GO TO 490                                       RSTA3610
      ROLD = RC16(ITEM)                                                 RSTA3620
      RC16(ITEM) = RR                                                   RSTA3630
      GO TO 492                                                         RSTA3640
  416 IF (ITEM.GT.2768) GO TO 490                                       RSTA3650
      IF (ITYPE.NE.2)   GO TO 490                                       RSTA3660
      IOLD = IC16(ITEM-2765)                                            RSTA3670
      IC16(ITEM-2765) = II                                              RSTA3680
      GO TO 494                                                         RSTA3690
C     COMMON /DAMPER/                                                   RSTA3700
   17 IF (ITEM.GT.220 ) GO TO 417                                       RSTA3710
      IF (ITYPE.NE.1)   GO TO 490                                       RSTA3720
      ROLD = RC17(ITEM)                                                 RSTA3730
      RC17(ITEM) = RR                                                   RSTA3740
      GO TO 492                                                         RSTA3750
  417 IF (ITEM.GT.261 ) GO TO 490                                       RSTA3760
      IF (ITYPE.NE.2)   GO TO 490                                       RSTA3770
      IOLD = IC17(ITEM-220)                                             RSTA3780
      IC17(ITEM-220) = II                                               RSTA3790
      GO TO 494                                                         RSTA3800
C     COMMON /CEULER/                                                   RSTA3810
   18 IF (ITEM.GT.22   ) GO TO 418                                      RSTA3820
      IF (ITYPE.NE.2)   GO TO 490                                       RSTA3830
      IOLD = IEULER(ITEM)                                               RSTA3840
      IEULER(ITEM) = II                                                 RSTA3850
      GO TO 494                                                         RSTA3860
  418 IF (ITEM.GT.526 ) GO TO 490                                       RSTA3870
      IF (ITYPE.NE.1)   GO TO 490                                       RSTA3880
      ROLD = RC18(ITEM-22)                                              RSTA3890
      RC18(ITEM-22) = RR                                                RSTA3900
      GO TO 492                                                         RSTA3910
C     COMMON /TEMPVI/                                                   RSTA3920
   19 IF (ITEM.GT.10   ) GO TO 419                                      RSTA3930
      IF (ITYPE.NE.1)   GO TO 490                                       RSTA3940
      ROLD = RC19(ITEM)                                                 RSTA3950
      RC19(ITEM) = RR                                                   RSTA3960
      GO TO 492                                                         RSTA3970
  419 IF (ITEM.GT.178 ) GO TO 490                                       RSTA3980
      IF (ITYPE.NE.2)   GO TO 490                                       RSTA3990
      IOLD = IC19(ITEM-10)                                              RSTA4000
```

```
      IC16(ITEM-10) = II
      GO TO 494
C     COMMON /WJONES/
   20 IF (ITEM.GT.315 ) GO TO 420
      IF (ITYPE.NE.1)   GO TO 490
      ROLD = RC20A(ITEM)
      RC20A(ITEM) = RR
      GO TO 492
  420 IF (ITEM.GT.321 ) GO TO 320
      IF (ITYPE.NE.2)   GO TO 490
      IOLD = IFULL(ITEM-315)
      IFULL(ITEM-315) = II
      GO TO 494
  320 IF (ITEM.GT.1433) GO TO 490
      IF (ITYPE.NE.1)   GO TO 490
      ROLD = RC200(ITEM-321)
      RC200(ITEM-321) = RR
      GO TO 492
C     COMMON /RSAVE/
   21 IF (ITEM.GT.180 ) GO TO 421
      IF (ITYPE.NE.1  ) GO TO 490
      ROLD = RC21(ITEM)
      RC21(ITEM) = RR
      GO TO 492
  421 IF (ITEM.GT.327 ) GO TO 490
      IF (ITYPE.NE.2  ) GO TO 490
      IOLD = IC21(ITEM-180)
      IC21(ITEM-180) = II
      GO TO 494
C     COMMON /FLXBLE/
   22 IF (ITEM.GT.624 ) GO TO 422
      IF (ITYPE.NE.1  ) GO TO 490
      ROLD = RC22(ITEM)
      RC22(ITEM) = RR
      GO TO 492
  422 IF (ITEM.GT.649 ) GO TO 490
      IF (ITYPE.NE.2  ) GO TO 490
      IOLD = IC22(ITEM-624)
      IC22(ITEM-624) = II
      GO TO 494
C     COMMON/HRNESS/
   23 IF (ITEM.GT.620 ) GO TO 423
      IF (ITYPE.NE.1)   GO TO 490
      ROLD = RC23(ITEM)
      RC23(ITEM) = RR
      GO TO 492
  423 IF (ITEM.GT.966 ) GO TO 490
      IF (ITYPE.NE.2)   GO TO 490
      IOLD = IC23(ITEM-620)
      IC23(ITEM-620) = II
```

```
      GO TO 494
C     COMMON/KALEPS/
   24 IF (ITEM.GT.30   ) GO TO 424
      IF (ITYPE.NE.1)   GO TO 490
      ROLD = RC24(ITEM)
      RC24(ITEM) = RR
      GO TO 492
  424 IF (ITEM.GT.170 ) GO TO 490
      IF (ITYPE.NE.2)   GO TO 490
      IOLD = IC24(ITEM-30 )
      IC24(ITEM-30 ) = II
      GO TO 494
C
C     ERROR MESSAGE - TERMINATE PROGRAM.
C
  490 WRITE (6,491) AVAR,INDEX,NCOM,ITEM,ITYPE,RR,II,AA
  491 FORMAT('0 SUBROUTINE RSTART INPUT ERROR'//
     *        ' AVAR= ',A8,'INDEX=',3I6,' NCOM=',I6,' ITEM=',I6,
     * ' ITYPE=',I6,' RR=',G15.8,' II=',I8,' AA= ',A8//
     * ' PROGRAM IS BEING TERMINATED.')
      STOP
C
C     PRINT MESSAGE FOR REAL VARIABLES.
C
  492 WRITE (6,493) AVAR,INDEX,COMMON(NCOM),ROLD,RR
  493 FORMAT('0',A6,'(',I4,',',I4,',',I4,') OF COMMON/',A6,'/',
     *        ' HAS BEEN CHANGED FROM ',G15.8,' TO ',G15.8)
      IF (RROLD.EQ.0.0) GO TO 400
      IF (DABS(RROLD-ROLD).LE.0.00001*RROLD) GO TO 400
      WRITE (6,383) RROLD
  383 FORMAT(' INPUT VALUE FOR RROLD WAS ',G15.8//)
      GO TO 490
C
C     PRINT MESSAGE FOR INTEGER VARIABLES.
C
  494 WRITE (6,495) AVAR,INDEX,COMMON(NCOM),IOLD,II
  495 FORMAT('0',A6,'(',I4,',',I4,',',I4,') OF COMMON/',A6,'/',
     *        ' HAS BEEN CHANGED FROM ',    I8,' TO ',    I8)
      IF (IIOLD.EQ.0) GO TO 400
      IF (IOLD.EQ.IIOLD) GO TO 400
      WRITE (6,385) IIOLD
  385 FORMAT(' INPUT VALUE FOR IIOLD WAS ',I8//)
      GO TO 490
C
C     PRINT MESSAGE FOR ALPHANUMERIC VARIABLES.
C
  496 WRITE (6,497) AVAR,INDEX,COMMON(NCOM),AOLD,AA
  497 FORMAT('0',A6,'(',I4,',',I4,',',I4,') OF COMMON/',A6,'/',
     *        ' HAS BEEN CHANGED FROM ',    A8,' TO ',    A8)
      IF (AAOLD.EQ.BLANK) GO TO 400
```

```
      AAOLD4 = AAOLD                                                    RSTA5010
      AOLD4  = AOLD                                                     RSTA5020
      IF (AOLD4.EQ.AAOLD4) GO TO 400                                    RSTA5030
      WRITE (6,387) AAOLD                                               RSTA5040
  387 FORMAT(' INPUT VALUE FOR AAOLD WAS ',A8//)                        RSTA5050
      GO TO 490                                                         RSTA5060
  999 CALL ELTIME(2,25)                                                 RSTA5070
      RETURN                                                            RSTA5080
      END                                                               RSTA5090
```

```
      SUBROUTINE SEARCH(AVAR,INDEX,NCOM,ITEM)                           SEAR0010
C                                                      REV 12 12/19/74SEAR0020
C     CALLED BY SUBROUTINE RSTART TO COMPUTE NCOM & ITEM FROM AVAR &    SEAR0030
C     INDEX. RETURNS NCOM=0 FOR ERROR AND NCOM=50 FOR BLANK.            SEAR0040
C                                                                       SEAR0050
      IMPLICIT REAL*8(A-H,O-Z)                                          SEAR0060
      DIMENSION BVAR(300),KOUNT(25),NDIM(3,300),NJ(3),NK(3),INDEX(3)    SEAR0070
      DATA NVAR/277/, KOM/24/                                           SEAR0080
      DATA KOUNT/  1, 11, 16, 37, 46, 54, 60, 71, 87,103,114,124,126,   SEAR0090
     *           143,150,163,176,182,189,194,258,261,266,274,278/       SEAR0100
      DATA BLANK/8H        /                                            SEAR0110
C                                                                       SEAR0120
C   1  COMMON/CONTRL/                                                   SEAR0130
C                                                                       SEAR0140
      DIMENSION C1 (  10) , NC1 (  30)                                  SEAR0150
      EQUIVALENCE (C1 (1),BVAR(  1)) , (NC1 (1),NDIM(1,  1))            SEAR0160
      DATA C1 / 8HNSEG     ,8HNJNT     ,8HNS3     ,8HNJ3     ,8HNPL     , SEAR0170
     *          8HNBLT     ,8HNBAG     ,8HNVEH    ,8HNGRND   ,8HNPRT    / SEAR0180
      DATA NC1 /  0,0,0    ,  0,0,0    ,  0,0,0   ,  0,0,0   ,  0,0,0   , SEAR0190
     *            0,0,0    ,  0,0,0    ,  0,0,0   ,  0,0,0   ,  40,0,0  / SEAR0200
C                                                                       SEAR0210
C   2  COMMON/CNTSRF/                                                   SEAR0220
C                                                                       SEAR0230
      DIMENSION C2 (   5) , NC2 (  15)                                  SEAR0240
      EQUIVALENCE (C2 (1),BVAR( 11)) , (NC2 (1),NDIM(1, 11))            SEAR0250
      DATA C2 / 8HPL       ,8HGAB      ,8HBELT    ,8HTPTS    ,8HBD      / SEAR0260
      DATA NC2 /  17,20,0  ,  8,3,0    ,  20,8,0  ,  6,8,0   ,  24,25,0 / SEAR0270
C                                                                       SEAR0280
C   3  COMMON/VPDSTN/                                                   SEAR0290
C                                                                       SEAR0300
      DIMENSION C3 (  21) , NC3 (  63)                                  SEAR0310
      EQUIVALENCE (C3 (1),BVAR( 16)) , (NC3 (1),NDIM(1, 16))            SEAR0320
      DATA C3 / 8HTIME     ,8HXO       ,8HXDOTO   ,8HXCOMP   ,8HXVCOMP  , SEAR0330
     *          8HAX       ,8HANGLE    ,8HVMPH    ,8HVTIME   ,8HATAB    , SEAR0340
     *          8HATC      ,8HADT      ,8HOMEGA   ,8HNATAB   ,8HNACLR   , SEAR0350
     *          8HDVEH     ,8HVMEG     ,8HVMEGD   ,8HXACOMP  ,8HTHET    , SEAR0360
     *          8HZPLT     /                                            SEAR0370
      DATA NC3 /  0,0,0    ,  3,0,0    ,  3,0,0   ,  3,0,0   ,  3,0,0   , SEAR0380
     *            3,0,0    ,  3,0,0    ,  0,0,0   ,  0,0,0   ,  15,100,0, SEAR0390
     *            0,0,0    ,  0,0,0    ,  0,0,0   ,  0,0,0   ,  0,0,0   , SEAR0400
     *            3,3,0    ,  3,0,0    ,  3,0,0   ,  3,0,0   ,  3,0,0   , SEAR0410
     *            3,0,0    /                                            SEAR0420
C                                                                       SEAR0430
C   4  COMMON/SGMNTS/                                                   SEAR0440
C                                                                       SEAR0450
      DIMENSION C4 (   9) , NC4 (  27)                                  SEAR0460
      EQUIVALENCE (C4 (1),BVAR( 37)) , (NC4 (1),NDIM(1, 37))            SEAR0470
      DATA C4 / 8HD        ,8HWMEG     ,8HWMEGD   ,8HU1      ,8HU2      , SEAR0480
     *          8HSEGLP    ,8HSEGLV    ,8HSEGLA   ,8HNSYM    /          SEAR0490
      DATA NC4 /  3,3,22   ,  3,22,0   ,  3,22,0  ,  3,22,0  ,  3,22,0  , SEAR0500
```

```
     *            3,22,0   ,  3,22,0   ,  3,22,0   ,  22,0,0  /
C
C  5  COMMON/CMATRX/
C
      DIMENSION C5 (    8) , NC5 (  24)
      EQUIVALENCE (C5 (1),BVAR( 46)) , (NC5 (1),NDIM(1, 46))
      DATA C5 / 8HV1       ,8HV2       ,8HV3       ,8HB12      ,8HA22      ,
     *          8HF        ,8HTQ       ,8HWJ       /
      DATA NC5 / 3,21,0    , 3,21,0    , 3,12,0    , 3,3,42    , 3,3,42    ,
     *           3,21,0    , 3,21,0    , 21,0,0    /
C
C  6  COMMON/ABDATA/
C
      DIMENSION C6 (    6) , NC6 (  18)
      EQUIVALENCE (C6 (1),BVAR( 54)) , (NC6 (1),NDIM(1, 54))
      DATA C6 / 8HABC      ,8HZA       ,8HDA       ,8HBFA      ,8HBCGV     ,
     *          8HBMEG     /
      DATA NC6 /  3,5,0    ,  3,5,0    ,  3,3,5    ,  3,5,0    ,  3,5,0    ,
     *            3,5,0    /
C
C  7  COMMON/TITLES/
C
      DIMENSION C7 (   11) , NC7 (  33)
      EQUIVALENCE (C7 (1),BVAR( 60)) , (NC7 (1),NDIM(1, 60))
      DATA C7 / 8HDATE     ,8HCOMENT   ,8HVPSTTL   ,8HBDYTTL   ,8HBLTTTL   ,
     *          8HPLTTL    ,8HBAGTTL   ,8HSEG      ,8HJOINT    ,8HCGS      ,
     *          8HJS       /
      DATA NC7 /  3,0,0    ,  40,0,0   ,  20,0,0   ,  5,0,0    ,  5,8,0    ,
     *            5,20,0   ,  5,6,0    ,  22,0,0   ,  21,0,0   ,  22,0,0   ,
     *            21,0,0   /
C
C  8  COMMON/CNSNTS/
C
      DIMENSION C8 (   16) , NC8 (  48)
      EQUIVALENCE (C8 (1),BVAR( 71)) , (NC8 (1),NDIM(1, 71))
      DATA C8 / 8HPI       ,8HRADIAN   ,8HG        ,8HTHIRD    ,8HEPS1     ,
     *          8HEPS4     ,8HEPS6     ,8HEPS8     ,8HEPS12    ,8HEPS15    ,
     *          8HEPS20    ,8HEPS24    ,8HUNITL    ,8HUNITM    ,8HUNITT    ,
     *          8HGRAVTY   /
      DATA NC8 /  0,0,0    ,  0,0,0    ,  0,0,0    ,  0,0,0    ,  0,0,0    ,
     *            0,0,0    ,  0,0,0    ,  0,0,0    ,  0,0,0    ,  0,0,0    ,
     *            0,0,0    ,  0,0,0    ,  0,0,0    ,  0,0,0    ,  0,0,0    ,
     *            3,0,0    /
C
C  9  COMMON/DESCRP/
C
      DIMENSION C9 (   16) , NC9 (  48)
      EQUIVALENCE (C9 (1),BVAR( 87)) , (NC9 (1),NDIM(1, 87))
      DATA C9 / 8HPHI      ,8HW        ,8HSR       ,8HHA       ,8HHB       ,
     *          8HHT       ,8HRPHI     ,8HRW       ,8HSPRING   ,8HVISC     ,
```

```
     *             8HJNT      ,8HIPIN     ,8HNS        ,8HISING    ,8HIGLOB     ,
     *             8HJOINTF   /
      DATA NC9 / 3,22,0       , 22,0,0    , 3,42,0     , 3,42,0    , 3,42,0     ,
     *            3,3,42      , 3,22,0    , 22,0,0     , 5,63,0    , 7,63,0     ,
     *            21,0,0      , 21,0,0    , 0,0,0      , 22,0,0    , 21,0,0     ,
     *            21,0,0      /
C
C 10   COMMON/JBARTZ/
C
      DIMENSION C10(   11) , NC10(   33)
      EQUIVALENCE (C10(1),BVAR(103)) , (NC10(1),NDIM(1,103))
      DATA C10/ 8HMNPL      ,8HMNBLT    ,8HMNSEG     ,8HMNBAG    ,8HMPL       ,
     *          8HMBLT      ,8HMSEG     ,8HMBAG      ,8HNTPL     ,8HNTBLT     ,
     *          8HNTSEG     /
      DATA NC10/  20,0,0  ,   8,0,0  ,   22,0,0  ,   6,0,0  ,   3,5,20  ,
     *            3,5,8   ,   3,5,22 ,   3,10,6  ,   5,20,0 ,   5,8,0   ,
     *            5,22,0  /
C
C 11   COMMON/FORCES/
C
      DIMENSION C11(   10) , NC11(   30)
      EQUIVALENCE (C11(1),BVAR(114)) , (NC11(1),NDIM(1,114))
      DATA C11/ 8HPSF       ,8HBSF      ,8HSSF       ,8HBAGSF    ,8HNPSF      ,
     *          8HNBSF      ,8HNSSF     ,8HNBGSF     ,8HNPANEL   ,8HPRJNT     /
      DATA NC11/  7,20,0  ,   4,20,0 ,   10,20,0 ,   3,20,0 ,   0,0,0    ,
     *            0,0,0   ,   0,0,0  ,   0,0,0   ,   6,0,0  ,   6,21,0   /
C
C 12   COMMON/INTEST/
C
      DIMENSION C12(    2) , NC12(    6)
      EQUIVALENCE (C12(1),BVAR(124)) , (NC12(1),NDIM(1,124))
      DATA C12/ 8HSGTEST    ,8HXTEST    /
      DATA NC12/  3,4,22  ,   3,88,0  /
C
C 13   COMMON/CSTRNT/
C
      DIMENSION C13(   17) , NC13(   51)
      EQUIVALENCE (C13(1),BVAR(126)) , (NC13(1),NDIM(1,126))
      DATA C13/ 8HA13       ,8HA23      ,8HB31       ,8HB32      ,8HHHT       ,
     *          8HRK1       ,8HRK2      ,8HQQ        ,8HTQQ      ,8HRQQ       ,
     *          8HHQQ       ,8HSQQ      ,8HCFQQ      ,8HNQ       ,8HKQ1       ,
     *          8HKQ2       ,8HKQTYPE   /
      DATA NC13/  3,3,24  ,   3,3,24 ,   3,3,24  ,   3,3,24 ,   3,3,12   ,
     *            3,12,0  ,   3,12,0 ,   3,12,0  ,   3,12,0 ,   3,12,0   ,
     *            3,12,0  ,   12,0,0 ,   12,0,0  ,   0,0,0  ,   12,0,0   ,
     *            12,0,0  ,   12,0,0 /
C
C 14   COMMON/TABLES/
C
      DIMENSION C14(    7) , NC14(   21)
```

```
      EQUIVALENCE (C14(1),8VAR(143)) , (NC14(1),NDIM(1,143))
      DATA C14/ 8HMXNTI    ,8HMXNT8    ,8HMXT81    ,8HMXT82    ,8HNTI      ,
     *          8HNTA8     ,8HTA8      /
      DATA NC14/ 0,0,0   ,  0,0,0    ,  0,0,0    ,  0,0,0    ,  50,0,0   ,
     *          500,0,0  ,  2000,0,0/
C
C 15  COMMON/COMAIN/
C
      DIMENSION C15(  13) , NC15(  39)
      EQUIVALENCE (C15(1),BVAR(150)) , (NC15(1),NDIM(1,150))
      DATA C15/ 8HVAR      ,8HDER      ,8HDT       ,8HHO       ,8HHMAX     ,
     *          8HHMIN     ,8HRSTIME   ,8HISTEP    ,8HNSTEPS   ,8HNDINT    ,
     *          8HNEQ      ,8HIRSIN    ,8HIRSOUT   /
      DATA NC15/ 120,0,0 ,  120,0,0  ,  0,0,0    ,  0,0,0    ,  0,0,0    ,
     *           0,0,0   ,  0,0,0    ,  0,0,0    ,  0,0,0    ,  0,0,0    ,
     *           0,0,0   ,  0,0,0    ,  0,0,0    /
C
C 16  COMMON/CDINT /
C
      DIMENSION C16(  13) , NC16(  39)
      EQUIVALENCE (C16(1),8VAR(163)) , (NC16(1),NDIM(1,163))
      DATA C16/ 8HE        ,8HFF       ,8HGG       ,8HY        ,8HU        ,
     *          8HH        ,8HHPRINT   ,8HTSAVE    ,8HTPRINT   ,8HTSTART   ,
     *          8HICNT     ,8HIDBL     ,8HIFLAG    /
      DATA NC16/ 3,120,0 ,  5,120,0  ,  5,120,0  ,  5,120,C  ,  5,120,0  ,
     *           0,0,0   ,  0,0,0    ,  0,0,0    ,  0,0,0    ,  0,0,0    ,
     *           0,0,0   ,  0,0,0    ,  0,0,0    /
C
C 17  COMMON/DAMPER/
C
      DIMENSION C17(   6) , NC17(  18)
      EQUIVALENCE (C17(1),8VAR(176)) , (NC17(1),NDIM(1,176))
      DATA C17/ 8HAPSDM    ,8HAPSDN    ,8HASD      ,8HNSD      ,8HMSDM     ,
     *          8HMSDN     /
      DATA NC17/ 3,20,0  ,  3,20,0   ,  5,20,0   ,  0,0,0    ,  20,0,0   ,
     *           20,0,0  /
C
C 18  COMMON/CEULER/
C
      DIMENSION C18(   7) , NC18(  21)
      EQUIVALENCE (C18(1),8VAR(182)) , (NC18(1),NDIM(1,182))
      DATA C18/ 8HIEULER   ,8HHIR      ,8HANG      ,8HANGD     ,8HFE       ,
     *          8HTQE      ,8HCONST    /
      DATA NC18/ 22,0,0  ,  3,3,21   ,  3,21,0   ,  3,21,0   ,  3,21,0   ,
     *           3,21,0  ,  3,21,0   /
C
C 19  COMMON/TEMPVI/
C
      DIMENSION C19(   5) , NC19(  15)
      EQUIVALENCE (C19(1),8VAR(189)) , (NC19(1),NDIM(1,189))
```

```
      DATA C19/ 8HTT1      ,8HR11      ,8HR21      ,8HCREST    ,8HJSTOP    / SEAR2010
      DATA NC19/  3,0,0    ,  3,0,0    ,  3,0,0    ,  0,0,0    ,  4,2,21   / SEAR2020
C                                                                            SEAR2030
C 20  COMMON/WJONES/                                                         SEAR2040
C                                                                            SEAR2050
      DIMENSION C20(  64) , NC20( 192)                                       SEAR2060
      EQUIVALENCE (C20(1),8VAR(194)) , (NC20(1),NDIM(1,194))                 SEAR2070
      DATA C20/ 8HFORCE    ,8HTORA     ,8HX8M      ,8HZDEP     ,8HV8AGG    , SEAR2080
     *          8HVSCS     ,8H8PH1     ,8HD8R      ,8HDPVCTR   ,8HDEPLOY   , SEAR2090
     *          8HAB       ,8HSPRK     ,8HCYTD     ,8HCYPA     ,8HCYSP     , SEAR2100
     *          8HCYTO     ,8HCYVC     ,8HCYCD     ,8HCYK      ,8HCYR      , SEAR2110
     *          8HCYAT     ,8HCYPV     ,8HCYCDO    ,8HCYAO     ,8HCYPO     , SEAR2120
     *          8HCYSS     ,8HCYLO     ,8HCYC      ,8HCYRHOO   ,8HCYVMAX   , SEAR2130
     *          8HCYORFC   ,8HCYRHO    ,8HCYT      ,8HCYP      ,8HCYMIN    , SEAR2140
     *          8HCYMOUT   ,8H8AGPV    ,8HPD       ,8HVBAG     ,8HVOL8P    , SEAR2150
     *          8HSWITCH   ,8H1FULL    ,8HTMP      ,8HTMP1     ,8HA        , SEAR2160
     *          8HPF       ,8HTORQ     ,8HTQ8      ,8HFR8      ,8HVOL      , SEAR2170
     *          8HDELF     ,8H8        ,8HZ8       ,8HZR       ,8HBFB      , SEAR2180
     *          8HDRR      ,8HD8       ,8HPCGV     ,8HPMEG     ,8HVOLP     , SEAR2190
     *          8HFRA      ,8HPREVT    ,8HCK       ,8HCMASS    /             SEAR2200
      DATA NC20/  3,5,0    ,  3,5,0    ,  5,0,0    ,  3,5,0    ,  5,0,0    , SEAR2210
     *            5,0,0    ,  3,5,0    ,  3,3,5    ,  3,5,0    ,  3,5,0    , SEAR2220
     *            3,5,0    ,  5,0,0    ,  5,0,0    ,  5,0,0    ,  5,0,0    , SEAR2230
     *            5,0,0    ,  5,0,0    ,  5,0,0    ,  5,0,0    ,  5,0,0    , SEAR2240
     *            5,0,0    ,  5,0,0    ,  5,0,0    ,  5,0,0    ,  5,0,0    , SEAR2250
     *            5,0,0    ,  5,0,0    ,  5,0,0    ,  5,0,0    ,  5,0,0    , SEAR2260
     *            5,0,0    ,  5,0,0    ,  5,0,0    ,  5,0,0    ,  5,0,0    , SEAR2270
     *            5,0,0    ,  5,0,0    ,  5,0,0    ,  5,0,0    ,  5,0,0    , SEAR2280
     *            5,0,0    ,  6,0,0    ,  18,0,0   ,  3,0,0    ,  3,3,0    , SEAR2290
     *            3,0,0    ,  3,0,0    ,  3,10,0   ,  3,10,0   ,  10,0,0   , SEAR2300
     *            3,0,0    ,  9,4,5    ,  3,4,5    ,  3,4,5    ,  9,4,5    , SEAR2310
     *            9,4,5    ,  9,4,5    ,  3,4,5    ,  3,4,5    ,  4,5,0    , SEAR2320
     *            3,4,0    ,  0,0,0    ,  5,0,0    ,  5,0,0    /             SEAR2330
C                                                                            SEAR2340
C 21  COMMON/RSAVE/                                                          SEAR2350
C                                                                            SEAR2360
      DIMENSION C21(   3) , NC21(   9)                                       SEAR2370
      EQUIVALENCE (C21(1),BVAR(258)) , (NC21(1),NDIM(1,258))                 SEAR2380
      DATA C21/ 8HXSG      ,8HNSG      ,8HMSG      /                         SEAR2390
      DATA NC21/  3,20,3   ,  7,0,0    ,  20,7,0   /                         SEAR2400
C                                                                            SEAR2410
C 22  COMMON/FLXBLE/                                                         SEAR2420
C                                                                            SEAR2430
      DIMENSION C22(   5) , NC22(  15)                                       SEAR2440
      EQUIVALENCE (C22(1),8VAR(261)) , (NC22(1),NDIM(1,261))                 SEAR2450
      DATA C22/ 8HHF       ,8HB42      ,8HV4       ,8HNFLEX    ,8HNFLX     / SEAR2460
      DATA NC22/  4,12,8   ,  3,3,24   ,  3,8,0    ,  3,8,0    ,  0,0,0    / SEAR2470
C                                                                            SEAR2480
C 23  COMMON/HRNESS/                                                         SEAR2490
      DIMENSION C23(   8) , NC23(  24)                                       SEAR2500
```

```
      EQUIVALENCE (C23(1),8VAR(266)) , (NC23(1),NDIM(1,266))
      DATA C23/ 8H8AR      ,8HXLONG    ,8HIBAR     ,8HNTHRNS  ,8HNHRNSS  ,
     *          8HNBLTPH   ,8HNFBLT    ,8HNPTSPB   /
      DATA NC23/  6,100,0 ,  20,0,0  ,  2,100,0 ,  20,0,0  ,   0,0,0  ,
     *            5,0,0   ,  5,20,0  ,  20,0,0  /
C
C 24  COMMON/KALEPS/
      DIMENSION C24(   4) , NC24(  12)
      EQUIVALENCE (C24(1),8VAR(274)) , (NC24(1),NDIM(1,274))
      DATA C24/ 8HWTIME    ,8HIWIND    ,8HMWSEG    /
      DATA NC24/  30,0,0  ,  30,0,0  ,  5,22,0  /
      NCOM = 50
      IF (AVAR.EQ.BLANK) GO TO 99
C
C     SEARCH FOR VARIABLE NO. IV.
C
      NCOM = C
      DO 10 IV=1,NVAR
      IF (AVAR.EQ.BVAR(IV)) GO TO 12
   10 CONTINUE
      GO TO 99
C
C     SEARCH FOR COMMON NO. IC.
C
   12 DO 20 IC=1,KOM
      IF (IV.GE.KOUNT(IC).AND.IV.LT.KOUNT(IC+1)) GO TO 22
   20 CONTINUE
      GO TO 99
C
C     COMPUTE ITEM NO. FOR VARIABLE IV IN COMMON IC.
C
   22 K1 = KOUNT(IC)
      K2 = IV-1
      ITEM = 1
      IF (K1.EQ.IV) GO TO 25
      DO 24 K=K1,K2
      NI = 1
      DO 23 I=1,3
      IF (NDIM(I,K).NE.0) NI=NI*NDIM(I,K)
   23 CONTINUE
   24 ITEM = ITEM+NI
   25 DO 26 I=1,3
      IF (INDEX(I).EQ.0 .AND. NDIM(I,IV).NE.0) GO TO 99
      IF (NDIM(I,IV).EQ.0 .AND. INDEX(I).GT.1) GO TO 99
      NJ(I) = MAX0(INDEX(I)-1,C)
      NK(I) = MAX0(NDIM(I,IV),1)
      IF (NJ(I).GE.NK(I)) GO TO 99
   26 CONTINUE
      ITEM = ITEM+NJ(1)+NJ(2)*NK(1)+NJ(3)*NK(2)*NK(1)
      NCOM = IC
```

```
99 RETURN                                                          SEAR3010
   END                                                             SEAR3020
```

```
      SUBROUTINE UPDATE(I)
C                                                     REV 12 12/19/74
C     CALLED BY SUBROUTINE DINT
C       (I=2)  AT THE END OF A SUCCESSFUL INTEGRATION STEP TO COMPLETE
C              CALCULATIONS FOR THAT STEP FOR OUTPUT (SUBROUTINE AIRBG3)
C
C       (I=1)  AT THE START OF A NEW STEP TO SETUP ANY NEW CONDITIONS
C              TO BE VALID FOR ENTIRE INTEGRATION STEP
C                A. UPDATE FORCE DEFLECTION FUNCTIONS(SUBROUTINE UPDFDC)
C                B. TEST FOR LOCKED JOINTS
C
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/CONTRL/NSEG,NJNT,NS3,NJ3,NPL,NBLT,NBAG,NVEH,NGRND,NPRT(40)
      COMMON/JBARTZ/ MNPL(  20),MNBLT(   8),MNSEG(  22),MNBAG(   6),
     *            MPL(3,5,20),MBLT(3,5,8),MSEG(3,5,22),MBAG(3,10,6),
     *             NTPL(5,20),NTBLT(5,8),NTSEG(5,22)
      COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(500),TAB(2000)
      COMMON/FORCES/PSF(7,20),BSF(4,20),SSF(10,20),BAGSF(3,20),
     *             NPSF,NBSF,NSSF,NBGSF,NPANEL(6),PRJNT(6,21)
      COMMON/DESCRP/ PHI(3,22),W(22),SR(3,42),HA(3,42),HB(3,42)
     *            ,HT(3,3,42),RPHI(3,22),RW(22),SPRING(5,63)
     *            ,VISC(7,63),JNT(21),IPIN(21),NS,ISING(22)
     *            ,IGLOB(21)
      COMMON/SGMNTS/D(3,3,22),WMEG(3,22),WMEGO(3,22),U1(3,22),U2(3,22)
     *            ,SEGLP(3,22),SEGLV(3,22),SEGLA(3,22),NSYM(22)
      COMMON/CMATRX/V1(3,21),V2(3,21),V3(3,12),B12(3,3,42),A22(3,3,42)
     *            ,F(3,21),TQ(3,21),WJ(21)
      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)
     *            ,NQ,KQ1(12),KQ2(12),KQTYPE(12)
      COMMON/TEMPVI/ TTI(3),R1I(3),R2I(3),CREST,JSTOP(4,2,21)
      COMMON/CEULER/ IEULER(22),HIR(3,3,21),ANG(3,21),ANGD(3,21),
     *             FE(3,21),TQE(3,31),CONST(3,21)
      COMMON/HRNESS/ BAR(6,100) , XLONG(20), IBAR(2,100), NTHRNS(20),
     *             NHRNSS, NBLTPH(5), NFBLT(5,20), NPTSPB(20)
      DIMENSION TQTEST(3),LOCK(8,3)
      DATA LOCK/-8, 6, 5, 7,-3,-2,-4, 1,
     *          6,-8, 4,-3, 7,-1,-5, 2,
     *          5, 4,-8,-2,-1, 7,-6, 3/
C
      CALL ELTIME(1,7)
      IF (I.NE.2) GO TO 9
C
C     CALL AIRBG3 FOR AIRBAG, IF ANY.
C
      IF (NBAG.NE.0) CALL AIRBG3
    9 IF (I.NE.1) GO TO 99
      IF (NPL.LE.0) GO TO 12
```

```
C
C     CALL UPDFDC FOR EACH ALLOWED PLANE-SEGMENT CONTACT.
C
      NPSF = 0
      DO 11 J = 1,NPL
      NK = MNPL(J)
      IF (NK.LE.0) GO TO 11
      DO 10 K = 1, NK
      NPSF = NPSF+1
      NT = NTPL(K,J)
      NF = NTAB(NT+5)
      CALL UPDFDC(NT)
      IF (NT.GT.0.OR.TAB(NF+3).EQ.0.0) GO TO 10
      CALL IMPULS(1,K,J)
      I = -1
   10 CONTINUE
   11 CONTINUE
   12 IF (NBLT.LE.0) GO TO 15
C
C     CALL UPDFDC FOR EACH ALLOWED BELT-SEGMENT CONTACT.
C
      DO 14 J = 1,NBLT
      NK = MNBLT(J)
      IF (NK.LE.0) GO TO 14
      DO 13 K = 1,NK
      NT = NTBLT(K,J)
      NF = NTAB(NT+5)
      NT6 = NT+6
      CALL UPDFDC(NT)
C
C        AND FOR 2ND FUNCTION, IF FULL BELT FRICTION.
C
   13 IF (NF.NE.0) CALL UPDFDC(NT6)
   14 CONTINUE
C
C     CALL UPDFDC FOR EACH ALLOWED SEGMENT-SEGMENT CONTACT.
C
   15 NSSF = 0
      DO 17 J=1,NSEG
      NK = MNSEG(J)
      IF (NK.LE.0) GO TO 17
      DO 16 K = 1,NK
      NSSF = NSSF+1
      NT = NTSEG(K,J)
      NF = NTAB(NT+5)
      CALL UPDFDC(NT)
      IF (NT.GT.0.OR.TAB(NF+3).EQ.0.0) GO TO 16
      CALL IMPULS(3,K,J)
      I = -1
   16 CONTINUE
```

```
   17 CONTINUE
      IF (NHRNSS.LE.0) GO TO 71
C
C     CALL UPDFDC FOR EACH BELT OF HARNESS-BELT SYSTEMS.
C
      J1 = 1
      K1 = 1
      DO 70 II=1,NHRNSS
      IF (NBLTPH(II).LE.0) GO TO 70
      J2 = J1 + NBLTPH(II) - 1
      DO 69 J=J1,J2
      IF (NPTSPB(J).LE.0) GO TO 69
      K2 = K1 + NPTSPB(J) - 1
      NT = NTHRNS(J)
      NF = NTAB(NT+5)
      CALL UPDFDC(NT)
      K1 = K2+1
   69 CONTINUE
      J1 = J2+1
   70 CONTINUE
   71 IF (NJNT.LE.0) GO TO 39
C
C            CHECK FOR IMPULSE ON JOINT STOPS
C     TO BE CALLED IF IN JOINT STOP (JSTOP(1)=1) THIS TIME STEP
C     BUT NOT IN IN JOINT STOP (JSTOP(2)=0) AT PREVIOUS TIME.
C
       DO 20 K=1,NJNT
      IF (JNT(K).EQ.0) GO TO 20
      IF (IABS(IPIN(K)).NE.4 .AND. VISC(7,3*K-2).EQ.0.0) GO TO 19
      DO 18 J=1,3
      K3J = 3*K-3+J
      IF (IABS(IPIN(K)).NE.4) K3J=3*K-2
      IF (IABS(IPIN(K)).EQ.4 .AND. VISC(7,K3J).EQ.0.0) GO TO 18
      IF (JSTOP(J,1,K).NE.1.OR.JSTOP(J,2,K).NE.0) GO TO 18
      CALL IMPULS(4,J,K)
      I = -1
   18 JSTOP(J,2,K) = JSTOP(J,1,K)
   19 IF (IGLOB(K).EQ.0) GO TO 20
      NT = IGLOB(K)
      MT = NTAB(NT+5)
      CALL UPDFDC(NT)
      IF (TAB(MT+3).EQ.0.0) GO TO 20
      IF (JSTOP(4,1,K).NE.1.OR.JSTOP(4,2,K).NE.0) GO TO 20
      CALL IMPULS(4,4,K)
      I = -1
   20 JSTOP(4,2,K) = JSTOP(4,1,K)
C
C     TEST TO LOCK OR UNLOCK JOINTS
C
C
```

```
C     CONDITIONS TO CHANGE SIGN OF IPIN(J)
C
C                    PINNED              UNPINNED
C         LOCKED (-1) |H.TQ| > T1     (-2) |TQ| > T1
C
C       UNLOCKED  (+1) |H.TQ| < T2     (+2) |TQ| < T2
C                            OR                  OR
C                        WJ  < T3            WJ  < T3
C
      DO 30 J=1,NJNT
      IF (IABS(IPIN(J)).EQ.4) GO TO 30
      IF (IPIN(J)) 21,30,22
   21 T1 = VISC(4,3*J-2)
      IF (T1.EQ.0.0) GO TO 30
      IF (IPIN(J).LE.-2) TQM = DSQRT(TQ(1,J)**2+TQ(2,J)**2+TQ(3,J)**2)
      IF (IPIN(J).EQ.-1) TQM = DABS(XDY(HB(1,2*J),D(1,1,J+1),TQ(1,J)))
      IF (TQM-T1) 30,30,29
   22 T2 = VISC(5,3*J-2)
      IF (T2.EQ.0.0) GO TO 23
      IF (IPIN(J).GE.2)  TQM = DSQRT(TQ(1,J)**2+TQ(2,J)**2+TQ(3,J)**2)
      IF (IPIN(J).EQ. 1) TQM = DABS(XDY(HB(1,2*J),D(1,1,J+1),TQ(1,J)))
      IF (TQM-T2) 28,30,30
   23 T3 = VISC(6,3*J-2)
      IF (T3.EQ.0.0) GO TO 30
      IF (WJ(J)-T3) 28,30,30
   28 CALL IMPLS2(0,J,HB(1,2*J))
      1 = -1
   29 IPIN(J) = -IPIN(J)
   30 CONTINUE
C
C     TEST TO LOCK OR UNLOCK EULER JOINTS AXES.
C     USE SAME TEST AS ABOVE BUT ON EACH AXIS SERARATELY.
C
C      IF LOCK(IEULER,K) IS NEGATIVE, AXIS K IS LOCKED;
C         TO UNLOCK AXIS SET IEULER TO -LOCK(IEULER,K).
C
C     IF LOCK(IEULER,K) IS POSITIVE, AXIS K IS UNLOCKED;
C       TO LOCK AXIS SET IEULER TO LOCK(IEULER,K).
C
      DO 60 J=1,NJNT
      IF (IABS(IPIN(J)).NE.4) GO TO 60
      JEULER = IEULER(J)
      CALL DOT(HIR(1,1,J),TQ(1,J),TQTEST,3,1,3)
      DO 55 K=1,3
      K3J = 3*J-3+K
      NLOCK = LOCK(JEULER,K)
      IF (NLOCK.GT.0) GO TO 52
      IF (VISC(4,K3J).EQ.0.0) GO TO 55
      IF (DABS(TQTEST(K)).GT.VISC(4,K3J)) JEULER = -NLOCK
      GO TO 55
```

```
 52 IF (VISC(5,K3J).EQ.0.0) GO TO 53
    IF (DABS(TQTEST(K)).LT.VISC(5,K3J)) JEULER =  NLOCK
    GO TO 55
 53 IF (VISC(6,K3J).EQ.0.0) GO TO 55
    IF (DABS(ANGD(K,J)).LT.VISC(6,K3J)) JEULER =  NLOCK
 55 CONTINUE
    IF (JEULER.EQ.IEULER(J)) GO TO 60
    IF (JEULER.EQ.8)         GO TO 59
    MODE = -1
    K = JEULER
    IF (K.LE.3) GO TO 57
    MODE =  1
    K = K-3
    IF (K.GT.3) MODE=0
 57 IEULER(J) = 8
    IPIN(J)   = 4
    CALL IMPLS2(MODE,J,HIR(1,K,J))
    I = -1
 59 IEULER(J) = JEULER
    IPIN(J) = 4
    IF (IEULER(J).NE.8) IPIN(J) = -4
 60 CONTINUE
C
 39 IF (NQ.LE.0) GO TO 99
    DO 40 K=1,NQ
    IF (KQTYPE(K).LT.3) GO TO 40
    IF (KQTYPE(K).GT.4) GO TO 40
    IF (CFQQ(K).LT.0.0) KQTYPE(K) = -KQTYPE(K)
    IF (CFQQ(K).LT.0.0) GO TO 42
C
C   TEST IF ROLLING CONSTRAINT SHOULD BE SLIDING AND VICE VERSA.
C
    QN = -XDY(TQQ(1,K),HHT(1,1,K),QQ(1,K))
    IF (NPRT(24).NE.0) WRITE (6,41) KQTYPE(K),KQ1(K),KQ2(K),
   *                    (RK1(II,K),II=1,3),(RK2(II,K),II=1,3),
   *                    ((HHT(II,J,K),J=1,3),II=1,3),
   *        (QQ(II,K),II=1,3),(TQQ(II,K),II=1,3),(RQQ(II,K),II=1,3),
   *                (HQQ(II,K),II=1,3),SQQ(K),CFQQ(K),QN
 41 FORMAT('0 UPDATE ROLL-SLIDE TEST'/(2X,9G14.6))
    IF (QN.LT.0.0) KQTYPE(K) = -4
    IF (QN.LT.0.0) GO TO 42
    QDOTQ = QQ(1,K)**2 + QQ(2,K)**2 + QQ(3,K)**2
    QT = DSQRT(QDOTQ-QN**2)
    IF (KQTYPE(K).EQ.3 .AND. QT.LE.CFQQ(K)*QN) GO TO 40
    IF (KQTYPE(K).EQ.4 .AND. QT.GE.0.9*CFQQ(K)*QN) GO TO 40
    KQTYPE(K) = 7-KQTYPE(K)
 42 CALL OUTPUT(0)
    CALL SETUP2
    CALL DAUX(K)
    CALL OUTPUT(1)
```

```
      CALL PRINT(6HUPDATE)                                               UPDA2510
      I = -1                                                             UPDA2520
   40 CONTINUE                                                           UPDA2530
   99 CALL ELTIME(2,7)                                                   UPDA2540
      RETURN                                                             UPDA2550
      END                                                                UPDA2560
```

```
      SUBROUTINE VEHPOS
C                                                        REV 12 12/16/74
C
C     COMPUTE COMPONENTS OF VEHICLE POSITION ANO MOTION AS A FUNCTION
C     OF TIME USING DATA ANO TABLES PRODUCED BY SUBROUTINE VINPUT.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/CONTRL/NSEG,NJNT,NS3,NJ3,NPL,NBLT,NBAG,NVEH,NGRND,NPRT(40)
      COMMON/SGMNTS/O(3,3,22),WMEG(3,22),WMEGD(3,22),U1(3,22),U2(3,22)
     *              ,SEGLP(3,22),SEGLV(3,22),SEGLA(3,22),NSYM(22)
      COMMON/VPOSTN/ TIME,XO(3),XDOTO(3),XCOMP(3),XVCOMP(3),AX(3),
     *              ANGLE(3),VMPH,VTIME,ATAB(15,100),ATO,ADT,OMEGA,
     *              NATAB,NACLR,DVEH(3,3),VMEG(3),VMEGO(3),XACOMP(3),
     *              THET(3),ZPLT(3)
      COMMON/CNSNTS/ PI, RAOIAN,G,THIRO,EPS1,EPS4,EPS6,EPS8,
     *              EPS12,EPS15,EPS20,EPS24,UNITL,UNITM,UNITT,GRAVTY(3)
      DIMENSION AC(3)
      DATA TLAST/-100000.0/
      T = TIME
      IF(NATAB.NE.0) GO TO 20
C
C     HALF-SINE WAVE DECELERATION
C
      IF(T.GT.VTIME) T=VTIME
      WT = OMEGA*T
      CWT1 = OCOS(WT)-1.0
      SWT = OSIN(WT)
      DO 10 I=1,3
      AW = AX(I)*OMEGA
      XACOMP(I) = -AW*OMEGA*SWT
      XCOMP(I) = AX(I)*SWT + T*(XDOTO(I)-AW)+XO(I)
   10 XVCOMP(I) = AW*CWT1 + XDOTO(I)
      GO TO 99
   20 IF (NATAB.LT.0) GO TO 30
C
C      UNIDIRECTIONAL DECELERATION
C
      IF (T.LT.VTIME) GO TO 21
C
C         TIME POINT EXCEEDS TABLE, EXTRAPOLATE.
C
      DLT = T-VTIME
      ACO   = ATAB(1,NATAB)
      AC(1) = ATAB(2,NATAB) + G*ACO*DLT
      AC(2) = ATAB(3,NATAB) + AC(1)*DLT + 0.5*G*ACO*DLT**2
      GO TO 25
C
C         USE QUAORATIC INTERPOLATION FROM TABLES FOR CURRENT VALUE OF
C         TIME TO BE CONSISTENT WITH SIMPSON INTEGRATION OF TABLES.
C
   21 J= 0.5*(T-ATO)/ADT +1.0
```

```
      XK = T/ADT -DFLOAT(2*J-1)
      X1 = XK+1.0
      X2 = XK**2-XK+I.0
      X3 = XK-1.0
      UNITS = -G
      DO 23 I=1,2
      T1 = (ATAB(I,2*J-1)-2.0*ATAB(I,2*J)+ATAB(I,2*J+1))/6.0
      T2 = (ATAB(I,2*J+1)-ATAB(I,2*J-1))/4.0
      T3 = ATAB(I,2*J)
      AC(I) = ATAB( I+I,2*J-1)+ADT*X1*(X2*T1+X3*T2+T3)*UNITS
   23 UNITS = 1.0
      ACO = 0.5*XK*X3*ATAB(1,2*J-1)
     *    -      X3*X1*ATAB(1,2*J   )
     *    + 0.5*XK*X1*ATAB(1,2*J+1)
C
C         COMPONENTS OF VEHICLE ACCELERATION, VELOCITY AND POSITION.
C
   25 DO 29 I=I,3
      XACOMP(I) = -G*AX(I)*ACO
      XVCOMP(I) = AX(I)*AC(1)
   29 XCOMP(I) = XO(I)+AX(I)*AC(2)
      GO TO 99
C
C      OMNIDIRECTIONAL DECELERATION
C
   30 IF (TIME.EQ.TLAST) GO TO 99
      DLTA      = TIME-TLAST
      IF (TLAST.EQ.-100000.0) DLTA = 0.0
      TLAST     = TIME
      J         = (TIME-ATO)/ADT+1.0
      IF (J.GE.-NATAB) GO TO 32
C
C         INTERPOLATION FROM VINPUT TABLES OF COMPONENTS OF VEHICLE
C         LINEAR AND ANGULAR ACCELERATION, VELOCITY AND DISPLACEMENT.
C
      TJ        = ATO + DFLOAT(J-1)*ADT
      DLT       = TIME-TJ
      DO 31 I=I,3
      AL2       = (ATAB(I,J+1)-ATAB(I,J))*DLT/ADT*G
      AL1       = G*ATAB(I,J)
      XACOMP(I)= -AL1-AL2
      AL2       = 0.5*AL2
      XVCOMP(I)= ATAB(I+3,J)-DLT*(AL1+AL2)
      XCOMP(I) = ATAB(I+6,J)+DLT*(ATAB(I+3,J)-DLT*(0.5*AL1+AL2/3.0))
      AA2       = (ATAB(I+9,J+1)-ATAB(I+9,J))*RADIAN/ADT
      THET(I)  = DLTA*(VMEG(I)+DLTA*(0.5*VMEGD(I)+DLTA*AA2/6.0))
      AA2       = AA2 * DLT
      AA1       = ATAB(I+9,J)*RADIAN
      VMEGD(I) = AA1 + AA2
   31 VMEG(I)  = ATAB(I+12,J)*RADIAN + DLT*(AA1+0.5*AA2)
```

```
      GO TO 34
C
C          TIME POINT EXCEEDS TABLE, EXTRAPOLATE.
C
   32 J = - NATAB
      TJ = ATO + DFLOAT(J-1)*ADT
      DLT = TIME-TJ
      DO 33 I=1,3
      XACOMP(I) = ATAB(I,J)*G
      XVCOMP(I) = ATAB(I+3,J)  +G*ATAB(I,J)*DLT
      XCOMP  (I) = ATAB(I+6,J)  +ATAB(I+3,J)*DLT + 0.5*G*ATAB(I,J)*DLT**2
      VMEGD(I) = 0.0
      VMEG  (I) = ATAB(I+12,J)*RADIAN
   33 THET  (I) = DLTA*VMEG(I)
C
C     UPDATE DIRECTION COSINE MATRIX OF VEHICLE.
C
   34 CALL DSETD(DVEH,THET,THT)
C
C     STORE VEHICLE DATA INTO NVEH SEGMENT DATA.
C
   99 DO 42 I=1,3
      DO 41 J=1,3
   41 D(I,J,NVEH) = DVEH(I,J)
      SEGLP(I,NVEH) =  XCOMP(I)
      SEGLV(I,NVEH) = XVCOMP(I)
      SEGLA(I,NVEH) = XACOMP(I)
      WMEG  (I,NVEH) =  VMEG  (I)
   42 WMEGD(I,NVEH) =  VMEGD(I)
      RETURN
      END
```

```
      SUBROUTINE VINPUT
C                                                     REV 12 12/16/74
C     PERFORMS CARD INPUT AND COMPUTES DATA AND TABLES REQUIRED BY
C     SUBROUTINE VEHPOS TO INTEGRATE THE CRASH VEHICLE MOTION FOR ONE OF
C     THREE PERMISSABLE OPTIONS:
C       (1) HALF SINE-WAVE LINEAR DECELERATION IMPULSE
C       (2) UNIDIRECTIONAL LINEAR DECELERATION TABULAR INPUT
C       (3) OMNIDIRECTIONAL LINEAR AND ANGULAR ACCELERATION TABULAR
C           INPUT (6 DEGREES OF FREEDOM VEHICLE MOTION)
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/CONTRL/NSEG,NJNT,NS3,NJ3,NPL,NBLT,NBAG,NVEH,NGRND,NPRT(40)
      COMMON/SGMNTS/D(3,3,22),WMEG(3,22),WMEGD(3,22),U1(3,22),U2(3,22)
     *              ,SEGLP(3,22),SEGLV(3,22),SEGLA(3,22),NSYM(22)
      COMMON/DESCRP/ PHI(3,22),W(22),SR(3,42),HA(3,42),HB(3,42)
     *              ,HT(3,3,42),RPHI(3,22),RW(22),SPRING(5,63)
     *              ,VISC(7,63),JNT(21),IPIN(21),NS,ISING(22)
      COMMON/VPOSTN/ TIME,X0(3),XDOT0(3),XCOMP(3),XVCOMP(3),AX(3),
     *              ANGLE(3),VMPH,VTIME,ATAB(15,100),AT0,ADT,OMEGA,
     *              NATAB,NACLR,DVEH(3,3),VMEG(3),VMEGD(3),XACOMP(3),
     *              THET(3),ZPLT(3)
      COMMON/CNSNTS/ PI, RADIAN,G,THIRD,EPS1,EPS4,EPS6,EPS8,
     *               EPS12,EPS15,EPS20,EPS24,UNITL,UNITM,UNITT,GRAVTY(3)
      COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),BLTTTL(5,8)
     *              ,PLTTL(5,20),BAGTTL(5,6),SEG(22),JOINT(21)
     *              ,CGS(22),JS(21)
      REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT
      LOGICAL*1 CGS,JS
      REAL VEH,GRND
      DATA VEH/' VEH'/,GRND/'GRND'/
C
C     READ AND PRINT CONTENTS OF CARDS C.1 AND C.2
C
      READ (5,10) VPSTTL
   10 FORMAT (20A4)
      READ(5,11) ANGLE,VMPH,VTIME,X0,NATAB,AT0,ADT
   11 FORMAT(8F6.0,I6,2F6.0)
      WRITE (6,14) VPSTTL,ANGLE,VMPH,VTIME,X0,NATAB,AT0,ADT
   14 FORMAT('1 VEHICLE DECELERATION INPUTS',91X,'CARDS C'//
     *                               3X,20A4//7X,'YAW',9X,'PIT
     *CH',7X,'ROLL',8X,'VMPH',8X,'VTIME',7X,'X0(X)',7X,'X0(Y)',7X,'X0(Z)
     *',7X,'NATAB',4X,'AT0',9X,'ADT'/8F12.3,I10,2X,2F12.6)
      VIPS = VMPH
      DA1 = ANGLE(1)*RADIAN
      DA2 = ANGLE(2)*RADIAN
      AX(3) = DCOS(DA2)
      AX(1) = DCOS(DA1)*AX(3)
      AX(2) = DSIN(DA1)*AX(3)
      AX(3) = DSIN(DA2)
      IF(NATAB.NE.0) GO TO 20
```

```
C
C     HALF-SINE WAVE DECELERATION
C
      OMEGA = PI/VTIME
      AT = 0.5*VIPS/OMEGA
      IF (VIPS.LT.0.0) VIPS = 0.0
      DO 12 I=1,3
      XDDTO(I) = VIPS*AX(I)
   12 AX(I) = AT*AX(I)
      WRITE (6,13) VIPS,UNITL,UNITT,ANGLE,VTIME,UNITT
   13 FORMAT('0 PASSENGER COMPARTMENT DISPLACEMENT HISTORY'/
     *       '  ANALYTICAL HALF-SINE WAVE DECELERATION'/
     *       '  VO=',F8.3,1X,A4,'/',A4,', OBLIQUE ANGLES =',3F7.2,
     *       ' DEGREES, TIME DURATION =',F7.3,1X,A4//)
      GO TO 41
   20 IF (NATAB.LT.0) GO TO 50
C
C     FOR UNIDIRECTIONAL VEHICLE MOTION
C     READ LINEAR DECELERATION TABLES FROM CARDS C.3
C
      READ (5,21) (ATAB(1,I),I=1,NATAB)
   21 FORMAT (12F6.0)
C
C     EXTEND TABLE IF NECESSARY SUCH THAT NATAB IS ODD AND
C     LAST ENTRY NEED NOT BE ZERO. IF TABLE SIZE IS EXCEEDED ON TIME,
C     VALUE OF LAST ENTRY WILL BE USED.
C
      IF (MOD(NATAB,2).EQ.1) GO TO 23
      ATAB(1,NATAB+1) = ATAB(1,NATAB)
      NATAB = NATAB+1
   23 VTIME = ADT * DFLOAT(NATAB-1)
C
C     USING SIMPSON'S INTEGRATION, COMPUTE VELOCITY AND DISPLACEMENT
C     TABLE FOR NATAB EQUALLY SPACED (ADT) TIME POINTS.
C     FOR I=1,NATAB
C       ATAB(1,I) = LINEAR DECELERATION (G'S)
C       ATAB(2,I) = LINEAR VELOCITY (L UNITS/T UNITS)
C       ATAB(3,I) = LINEAR DISPLACEMENT (L UNITS)
C
      ATAB(2,1) = VIPS
      ATAB(3,1) = 0.0
      DA1 = ADT/3.0
      DA2 = ADT/12.0
      UNITS = -G
      DO 30 J=2,3
      DO 25 I=2,NATAB,2
      F1 = ATAB(J-1,I-1) * UNITS
      F2 = ATAB(J-1,I  ) * UNITS
      F3 = ATAB(J-1,I+1) * UNITS
      ATAB(J,I  ) = ATAB(J,I-1) + DA2*(5.0*F1+8.0*F2-F3)
```

```
   25 ATAB(J,I+1) = ATAB(J,I-1) + DA1*(     F1+4.0*F2+F3)
   30 UNITS = 1.0
C
C     PRINT TABLES
C
      WRITE (6,36) (UNITL,UNITT,UNITL,I=1,2)
   36 FORMAT('0 UNIDIRECTIONAL VEHICLE POSITION TABLES'//
     *      2('        TIME        ACC       VELOCITY      POSITION    ')/
     *      2('    (MILLESEC)     (G)     (',A4,'/',A4,')',5X,'(',A4,')')/)
      DO 40 J=1,50
      IF (J.GT.NATAB) GO TO 40
      T1 = (ATO + DFLOAT(J-1)*ADT)*1000.0
      IF (J+50.LE.NATAB) GO TO 38
      WRITE (6,37) T1,(ATAB(I,J),I=1,3)
   37 FORMAT(2(F11.5,F10.2,F13.4,F13.5,3X))
      GO TO 40
   38 T2 = (ATO + DFLOAT(J+49)*ADT)*1000.0
      WRITE (6,37) T1,(ATAB(I,J),I=1,3),T2,(ATAB(I,J+50),I=1,3)
   40 CONTINUE
C
C     INITIALIZATION
C
      DO 35 I=1,3
   35 XDOTO(I)= VIPS*AX(I)
   41 DO 43 I=I,3
      DO 42 J=1,3
   42 DVEH(1,J) = 0.0
      DVEH(I,I) = 1.0
      VMEGD(I) = 0.0
   43 VMEG(I)    = 0.0
      GO TO 99
C
C     FOR OMNIDIRECTIONAL (6 DEGREES OF FREEDOM) VEHICLE MOTION
C     READ LINEAR DECELERATION AND AUGULAR ACCELERATION TABLES
C     FROM CARDS C.4.
C
   50 MATAB = -NATAB
      READ(5,51)  ((ATAB(I,J),I=1,3),(ATAB(1,J),I=10,12),J=1,MATAB)
   51 FORMAT(10X,6F10.0)
      DO 60 J=I,MATAB
      IF (MOD(J,50).NE.1) GO TO 53
C
C     PRINT PAGE HEADING AT START OF EACH 50 TIME POINTS.
C
      IF (J.NE.1) WRITE (6,44)
   44 FORMAT('1')
      IPAGE = (J-1)/50 + 1
      WRITE (6,52) IPAGE,UNITL,UNITT,UNITL,UNITT
   52 FORMAT('0 ROTATING VEHICLE LINEAR TIME HISTORY',
     *       67X,'PAGE NO.',I3//
```

```
     *       4X,'TIME',12X,'LINEAR DECELERATIONS (G''S)',
     *                 10X,'LINEAR VELOCITIES (',A4,'/',A4,')',
     *                 11X,'LINEAR DISPLACEMENTS (',A4,')'    /
     $        3X,'(',A4,')',3(11X,'X',11X,'Y',11X,'Z',3X)     /  )
   53 IF (J.GT.1) GO TO 57
C
C     INTEGRATION INITIALIZATION FOR TIME = 0.
C
      DO 54 I=1,3
      ATAB(I+6,J) = X0(I)
      ATAB(I+12,J) = 0.0
   54 THET(I) = ANGLE(I)*RADIAN
      CALL DRCYPR(DVEH,ANGLE,3,2,1)
      DO 55 I=1,3
      XDOTO(I) = VIPS*DVEH(1,I)
   55 ATAB(I+3,J) = XDOTO(I)
      GO TO 59
   57 DO 58 I=1,3
C
C     INTEGRATE LINEAR VELOCITY AND DISPLACEMENT.
C
      ATAB(I+3,J) = ATAB(I+3,J-1)-G*ADT/2.0*(ATAB(I,J-1)+ATAB(I,J))
   58 ATAB(I+6,J) = ATAB(I+6,J-1)
     *     +ADT*(ATAB(I+3,J-1)-G*ADT/6.0*(2.0*ATAB(I,J-1)+ATAB(I,J)))
   59 T1 = (ATO + DFLOAT(J-1)*ADT)
   60 WRITE(6,61) T1,(ATAB(I,J),I=1,9)
   61 FORMAT(F9.5,3(3X,3F12.3))
      DO 70 J=1,MATAB
      IF(MOD(J,50).NE.1) GO TO 63
C
C     PRINT PAGE HEADING AT START OF EACH 50 TIME POINTS.
C
      IPAGE = (J-1)/50+1
      WRITE (6,62) IPAGE,UNITT,UNITT,UNITT
   62 FORMAT('1 ROTATING VEHICLE ANGULAR TIME HISTORY',
     *       66X,'PAGE NO.',I3//
     *        4X,'TIME', 6X,'ANGULAR ACCELERATIONS (DEG/',A4,'**2)',
     *                 10X,'ANGULAR VELOCITIES (DEG/',A4,')',
     *                11X,'ANGULAR DISPLACEMENTS (DEG)' /
     *       3X,'(',A4,')',2(11X,'X',11X,'Y',11X,'Z',3X),
     *       10X,'YAW',8X,'PITCH',8X,'ROLL' /)
   63 IF(J.EQ.1) GO TO 65
C
C     INTEGRATE ANGULAR VELOCITY AND DISPLACEMENT.
C
      DO 64 I=1,3
      ATAB(I+12,J) = ATAB(I+12,J-1)+(ATAB(I+9,J-1)+ATAB(I+9,J))*ADT/2.0
   64 THET(I) = ADT*(ATAB(I+12,J-1)+(2.0*ATAB(I+9,J-1)+ATAB(I+9,J))*ADT
     */6.0)*RADIAN
      CALL DSETD(DVEH,THET,THT)
```

```
   65 CALL YPRDEG(DVEH,THET)
      T1 = (ATO + DFLOAT(J-1)*ADT)
   70 WRITE (6,71)  T1,(ATAB(I,J),I=10,15),THET
   71 FORMAT(F9.5,3(3X,3F12.3))
C
C     PROGRAM INITIALIZATION FOR TIME = 0.
C
      CALL DRCYPR (DVEH,ANGLE,3,2,1)
      DO 72 I=1,3
      VMEG(I)  = ATAB(I+12,1)*RADIAN
   72 VMEGD(I) = ATAB(I+9 ,1)*RADIAN
C
C     SET UP SEGMENT DATA FOR GROUND.
C
   99 NVEH = NSEG+1
      NGRND = NVEH+1
      SEG(NVEH) = VEH
      SEG(NGRND) = GRND
      IF (NVEH -1.GT.NJNT) JNT (NVEH -1) = 0
      IF (NVEH -1.GT.NJNT) IPIN(NVEH -1) = 0
      IF (NGRND-1.GT.NJNT) JNT (NGRND-1) = 0
      IF (NGRND-1.GT.NJNT) IPIN(NGRND-1) = 0
      DO 82 I=1,3
      DO 81 J=1,3
   81 D(I,J,NGRND) = 0.0
      D(I,I,NGRND) = 1.0
      SEGLP(I,NGRND) = 0.0
      SEGLA(I,NGRND) = 0.0
      SEGLV(I,NGRND) = 0.0
      WMEG (I,NGRND) = 0.0
   82 WMEGD(I,NGRND) = 0.0
      DO 83 J=NVEH,NGRND
      W(J) = 0.0
      RW(J) = 0.0
      DO 83 I=1,3
      PHI(I,J)  = 0.0
   83 RPHI(I,J) = 0.0
      RETURN
      END
```

```
      SUBROUTINE VISPR(IJ,NJ)
C                                                   REV 12 12/19/74
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
         IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/CONTRL/NSEG,NJNT,NS3,NJ3,NPL,NBLT,NBAG,NVEH,NGRND,NPRT(40)
      COMMON/DESCRP/ PHI(3,22),W(22),SR(3,42),HA(3,42),HB(3,42)
     *              ,HT(3,3,42),RPHI(3,22),RW(22),SPRING(5,63)
     *              ,VISC(7,63),JNT(21),IPIN(21),NS,ISING(22)
     *              ,IGLOB(21),JOINTF(21)
      COMMON/SGMNTS/D(3,3,22),WMEG(3,22),WMEGD(3,22),U1(3,22),U2(3,22)
     *              ,SEGLP(3,22),SEGLV(3,22),SEGLA(3,22),NSYM(22)
      COMMON/FORCES/PSF(7,20),BSF(4,20),SSF(10,20),BAGSF(3,20),
     *              NPSF,NBSF,NSSF,NBGSF,NPANEL(6),PRJNT(6,21)
      COMMON/CMATRX/V1(3,21),V2(3,21),V3(3,12),B12(3,3,42),A22(3,3,42)
     *              ,F(3,21),TQ(3,21),WJ(21)
      COMMON/CEULER/ IEULER(22),HIR(3,3,21),ANG(3,21),ANGD(3,21),
     *               FE(3,21),TQE(3,31),CONST(3,21)
      COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(500),TAB(2000)
       COMMON/TEMPVS/T1(3),T2(3),T3(3),T4(3),T5(3),T6(3),T7(3),T8(3)
     *              ,T9(3),HAD,HBD,WIJM,CV,CSA,CSB,WIJ(3),ANGL(3),TQC
     *              ,THETO,THETOP,DH1(3,3),DH2(3,3),HD3(3,3),CC(3)
      COMMON/TEMPVI/ TTI(3),R1I(3),R2I(3),CREST,JSTOP(4,2,21)
      COMMON/CNSNTS/ PI, RADIAN,G,THIRD,EPS1,EPS4,EPS6,EPS8,
     *               EPS12,EPS15,EPS20,EPS24,UNITL,UNITM,UNITT,GRAVTY(3)
      COMMON /VPOSTN/ TIME
      IF (NJNT.LE.0) GO TO 99
      CALL ELTIME(1,13)
      IF (NPRT(12).NE.0) WRITE (6,11)  TIME
   11 FORMAT('1 VISPR COMPUTATIONS FOR TIME =',F12.6)
      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
   12 TQ(L,J) = 0.0
      WJ(J) = 0.0
C
C     DO NOT COMPUTE TORQUES FOR NULL, LOCKED OR EULER JOINTS.
```

```
C
      I = IABS(JNT(J))
      IF (I.LE.0) GO TO 90
      IF (IPIN(J).LT.0 .OR. IPIN(J).GT.3) GO TO 90
C
C     ZERO T1-T9 ARRAYS, VARIABLES HAD,HBD,WIJM,CV,CSA, AND CS8,
C          WIJ AND ANGL ARRAYS AND VARIABLES TQC,THETO AND THETOP.
C
      DO 10 L=1,42
   10 TI(L) = 0.0
      CALL DOT(D(1,1,I  ),HT(I,I,2*J-1),DH1,3,3,3)
      CALL DOT(D(1,1,J+1),HT(1,I,2*J  ),DH2,3,3,3)
      CALL DOT(DHI,DH2,HD3,3,3,3)
C
C     NOTE: THIS VERSION CORRESPONDS TO OLDER VERSIONS AS FOLLOWS:
C
C       (HT) = ( (HC) (H8) (HA) )
C
C       (DH1) = ( (A) (T2) (TI) )
C
C       (DH2) = ( (B) (T5) (T4) )
C
C            WHERE  A = T2 X T1
C                   B = T5 X T4
C
C                 (  A.B   A.T5   A.T4 )
C       (HD3) =  ( T2.8  T2.T5  T2.T4 )
C                 ( T1.B  TI.T5  T1.T4 )
C
      HAD = HD3(3,3)
      IF (HAD.GT. I.0) HAD =  I.0
      IF (HAD.LT.-I.0) HAD = -1.0
      ANGL(1) = DARCOS(HAD)
      ANGL(2) = 0.0
      IF (HD3(2,3).NE.0.0 .OR. HD3(I,3).NE.0.0)
     *      ANGL(2) = DATAN2(HD3(2,3),HD3(1,3))
      CSAP = 0.0
      IF (NJ.NE.0.AND.IJ.EQ.4) GO TO 27
C
C     CONVERT TO INERTIAL REFERENCE SYSTEM
C        T3= D(I)'*WMEG(I)     T6=D(J+1)'*WMEG(J+1)
C
C     HAD = COS TA = HD3(3,3)
C     WIJ = T3-T6
C     WJ  = |WIJ|
C
       DO 20 L=1,3
       DO I5 M=1,3
       T3(L) = T3(L)+ D(M,L,I)* WMEG(M,I)
   I5  T6(L) = T6(L)+ D(M,L,J+1)* WMEG(M,J+I)
```

```
       WIJ(L)= T3(L)-T6(L)
   20  WIJM = WIJM + WIJ(L)**2
       WIJM = DSQRT(WIJM)
      WJ(J) = WIJM
C
C     T7 = T1 X T4
C     HAC = |T7|
C
      CALL CROSS (DH1(1,3),DH2(1,3),T7)
      HAC = DSQRT((1.0-HAD)*(1.0+HAD))
C
C     COMPUTE CV, THE MAGNITUDE OF VISCOUS AND COULOMB TORQUE/WIJM
C             RA = -SGN TA DOT = WIJ.T7
C        AND CSA, THE MAGNITUDE OF FLEXURE TORQUE/HAC
C
      CV = VISCOS(WIJM,VISC(1,3*J-2))
      CREST = VISC(7,3*J-2)
      RA = WIJ(1)*T7(1)+WIJ(2)*T7(2)+WIJ(3)*T7(3)
      JSTP = 0
      IF (JOINTF(J).EQ.0) CSA = EFUNCT(ANGL(1),RA,SPRING(1,3*J-2),JSTP)
      IF (JOINTF(J).NE.0) CSA = FNTERP(ANGL(1),ANGL(2),JOINTF(J))
      CSAP = CSA
      IF (HAC.NE.0.0)   CSA = CSA/HAC
      IF (NJ.EQ.0) JSTOP(1,1,J) = JSTP
      IF (IPIN(J).EQ.1) GO TO 34
C
C     FOR UNPINNED FREE JOINTS
C     CONVERT TO INERTIAL REFERENCE SYSTEM
C        T2 = D(I)'*HB(NJ)      T5 = D(J+1)'*HB(MJ)
C
C     T8 = T2 X T5
C     HBD = COS TB = T2.T5
C     HBC = |T8|
C
      ANGL(3) = DATAN2(HD3(2,1)-HD3(1,2),HD3(2,2)+HD3(1,1))
C
C     RB = -SGN TB DOT = WIJ.T8
C     COMPUTE CSB, THE MAGNITUDE OF TORSIONAL TORQUE.
C
      RB  = WIJ(1)*DH2(1,3) + WIJ(2)*DH2(2,3) + WIJ(3)*DH2(3,3)
      CSB = EFUNCT(ANGL(3),RB,SPRING(1,3*J-1),JSTP)
      IF (NJ.EQ.0) JSTOP(2,1,J) = JSTP
      IF (NJ.GT.0) GO TO 34
C
C     COMPUTE EFFECT OF GLOBALGRAPHIC JOINT STOP
C
   27 IF (IGLOB(J).EQ.0) GO TO 34
      IF (DABS(HAD).GT.1.0-EPS6) GO TO 34
      NT = IGLOB(J)
      CALL HERRON(HD3(1,3),NTAB(NT+1),THETO,THETOP)
```

```
      JSTOP(4,1,J) = 0
      IF (ANGL(1).LE.THETO) GO TO 34
      JSTOP(4,1,J) = 1
      MT = NTAB(NT+5)
      CREST = TAB(MT+3)
      STH2 = 1.0-HAD**2
      STH   = DSQRT(STH2)
      CTH   = HAD/STH
      CST   = DSQRT(STH2+THETOP**2)
      DR    = (ANGL(1)-THETO)*STH/CST
      LT = NTAB(NT)
      TAB(LT) = DR
      TQF = FRCDFL(DR,NT,1)
      TQC   = TQF/CST
      CC(1) = -HD3(2,3) + HD3(1,3)*CTH*THETOP
      CC(2) =  HD3(1,3) + HD3(2,3)*CTH*THETOP
      CC(3) =                      -STH*THETOP
      DO 28 L=1,3
   28 T9(L) = CC(1)*DH1(L,1) + CC(2)*DH1(L,2) + CC(3)*DH1(L,3)
C
C     COMPUTE TOTAL TORQUE IN INERTIAL REFERENCE BY
C     TQ = -CV*WIJ + CSA*T7 + CSB*T8 + TQC*T9
C
   34 IF (NJ.EQ.0) GO TO 36
      CV = 0.0
      IF (IJ.NE.1) CSA = 0.0
      IF (IJ.NE.2) CSB = 0.0
      IF (IJ.NE.4) TQC = 0.0
   36 DO 37 L=1,3
      TQ(L,J) = -CV*WIJ(L) + CSA*T7(L) + CSB*DH2(L,3) + TQC*T9(L)
   37 TTI(L) = TQ(L,J)
      IF (NPRT(12).NE.0) WRITE (6,39)
     *         J,CV,CSA,CSB,TQC,HAD,HAC,HBC,RA,RB,
     *         HD3,WIJ,T7,T9,
     *         (TQ(L,J),L=1,3)
   39 FORMAT(I4,1P9D14.6/(4X,9D14.6))
C
C     ADD TORQUE CONVERTED TO LOCAL REFERENCE BY
C      U2I = U2I + DI*TQ
C      U2J = U2J - DJ*TQ
C
       DO 40 L=1,3
       DO 40 M=1,3
      U2(L,I  ) = U2(L,I  ) + D(L,M,I  )*TQ(M,J)
   40 U2(L,J+1) = U2(L,J+1) - D(L,M,J+1)*TQ(M,J)
C
C     STORE DATA FOR OUTPUT ROUTINE INTO PRJNT ARRAY.
C
      PRJNT(1,J) = ANGL(1)
      PRJNT(2,J) = ANGL(3)
```

```
      PRJNT(3,J) = CSAP                                                 VISP2010
      PRJNT(4,J) = CSB                                                  VISP2020
      PRJNT(5,J) = CV*WIJM                                              VISP2030
      PRJNT(6,J) = DSQRT(TQ(1,J)**2+TQ(2,J)**2+TQ(3,J)**2)              VISP2040
   90  CONTINUE                                                         VISP2050
      CALL ELTIME(2,13)                                                 VISP2060
   99 RETURN                                                            VISP2070
      END                                                               VISP2080
```

```
      SUBROUTINE WINDY(M,MM,N,NN,NT)
C                                                   REV 12 12/20/74
C
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
       IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/CONTRL/NSEG,NJNT,NS3,NJ3,NPL,NBLT,NBAG,NVEH,NGRND,NPRT(40)
      COMMON/TABLES/MXNTI,MXNTB,MXTBI,MXTB2,NTI(50),NTAB(500),TAB(2000)
      COMMON/SGMNTS/D(3,3,22),WMEG(3,22),WMEGD(3,22),U1(3,22),U2(3,22)
     *              ,SEGLP(3,22),SEGLV(3,22),SEGLA(3,22),NSYM(22)
      COMMON/CNTSRF/ PL(17,20),GAB(8,3),BELT(20,8),TPTS(6,8),BD(24,25)
      COMMON/CNSNTS/ PI, RADIAN,G,THIRD,EPSI,EPS4,EPS6,EPS8,
     *               EPS12,EPS15,EPS20,EPS24,UNITL,UNITM,UNITT,GRAVTY(3)
      COMMON/VPOSTN/ TIME,X0(3),XDOTO(3),XCOMP(3),XVCOMP(3),AX(3),
     *              ANGLE(3),VMPH,VTIME,ATAB(15,100),ATO,ADT,OMEGA,
     *              NATAB,NACLR,DVEH(3,3),VMEG(3),VMEGD(3),XACOMP(3),
     *              THET(3),ZPLT(3)
      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)
      COMMON/KALEPS/WTIME(30),IWIND(30)
      CALL ELTIME(1,35)
C
C     COMPUTE PENETRATION DISTANCE; IF NEGATIVE, RETURN.
C
      CALL DOTT(D(1,1,M),D(1,1,N),DMNT,3,3,3)
      DO 10 I=1,3
   10 XMN(I) = SEGLP(I,M) - SEGLP(I,N)
      CALL MAT(D(1,1,M),XMN,XMM,3,3,1,3,3,3)
      CALL MAT(DMNT,PL(1,NN),TM,3,3,1,3,3,3)
      BET = PL(4,NN)
      DO 11 I=1,3
   11 BET = BET - TM(I)*(BD(I+3,MM)+XMM(I))
      CALL MAT(BD(16,MM),TM,RM,3,3,1,3,3,3)
      BTS = TM(1)*RM(1) + TM(2)*RM(2) + TM(3)*RM(3)
      BTE = -DSQRT(BTS)
      P   = BET - BTE
      IF (P.LT.0.0) GO TO 99
C
C     FETCH OR STORE INITIAL PENETRATION TIME.
C
      IWIND(M) = M
      IF (TIME.LE.WTIME(M)) WTIME(M) = TIME
      FTIME = TIME - WTIME(M)
C
C     GET FORCE VECTOR FT FROM TABLE NT FOR TIME = FTIME.
C
   22 KT = NTI(NT)
```

```
      NENTRY = TAB(KT+5)
      KI = K1+IO
      K2 = 4*NENTRY + KT+2
      IF (NENTRY.EQ.1) GO TO 31
      DO 30 K=K1,K2,4
      IF (FTIME.GT.TAB(K)) GO TO 30
      KK = K
      R1 = (TAB(K)-FTIME)/(TAB(K)-TAB(K-4))
      GO TO 32
   30 CONTINUE
   31 KK = K2
      R1 = 0.0
   32 R2 = 1.0 - R1
      DO 33 I=1,3
      K= KK+I
   33 FT(I) = R2*TAB(K) + RI*TAB(K-4)
C
C     COMPUTE PRESENTED AREA TO WIND FORCE.
C
      CALL MAT(D(I,I,M),FT,FF,3,3,1,3,3,3)
      CALL MAT(BD(7,MM),FF,AF,3,3,1,3,3,3)
      FAF = FF(I)*AF(1) + FF(2)*AF(2) + FF(3)*AF(3)
      IF (FAF.LE.0.0) GO TO 99
      TF  = TM(1)*FF(I) + TM(2)*FF(2) + TM(3)*FF(3)
      BREF = DSQRT(BTS-TF*TF/FAF)
      SCALE = (-BET+BREF)/(-BTE+BREF)
      IF (SCALE.GE.1.0) GO TO 99
      IF (SCALE.LT.0.0) SCALE = 0.0
      TRACER = (BD( 7,MM)-AF(1)**2/FAF)*(BD(I1,MM)-AF(2)**2/FAF)
     *        + (BD( 7,MM)-AF(1)**2/FAF)*(BD(I5,MM)-AF(3)**2/FAF)
     *        + (BD(11,MM)-AF(2)**2/FAF)*(BD(15,MM)-AF(3)**2/FAF)
     *        - (BD( 8,MM)-AF(1)*AF(2)/FAF)**2
     *        - (BD( 9,MM)-AF(1)*AF(3)/FAF)**2
     *        - (BD(I2,MM)-AF(2)*AF(3)/FAF)**2
      AREA = (I.0-SCALE**2) * PI / DSQRT(TRACER)
C
C     ADD FORCE AND TORQUES TO UI AND U2 ARRAYS FOR SEGMENT M.
C
      SCALE = SCALE/BTE
      DO 36 I=1,3
      RLM(I) = RM(I)*SCALE + BD(I+3,MM)
      FT (I) = FT(I)*AREA
   36 FF (I) = FF(I)*AREA
      CALL CROSS (RLM,FF,TQM)
      DO 39 I=I,3
      UI(I,M) = U1(I,M) + FT(I)
   39 U2(I,M) = U2(I,M) + TQM(I)
      IF (NPRT(14).NE.0) WRITE (6,41) TIME,M,P,AREA,FT,TQM
   41 FORMAT(' WIND FORCE',F14.6,I6,2F10.3,3X,3F12.5,3X,3F12.5)
   99 CALL ELTIME (2,35)
```


☆ U. S. GOVERNMENT PRINTING OFFICE: 1975 — 657-630/45