DOT HS-801 847

FINITE ELEMENT ANALYSIS OF AUTOMOTIVE STRUCTURES UNDER CRASH LOADINGS
Volume II-Technical Report

Contract No. DOT-HS-105-3-697
March 1976
Final Report

PREPARED FOR:

U.S. DEPARTMENT OF TRANSPORTATION

NATIONAL HIGHWAY TRAFFIC SAFETY ADMINISTRATION

WASHINGTON, D.C. 20590

Document is available to the public through
the National Technical Information Service,
Springfield, Virginia 22161

The contents of this report reflect the views of IIT Research Institute which is responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views or policy of the Department of Transportation. This report does not constitute a standard, specification or regulation.

TECHNICAL REPORT STANDARD TITLE PAGE

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
DOT HS-801 847		

4. Title and Subtitle	5. Report Date
FINITE ELEMENT ANALYSIS OF AUTOMOTIVE STRUCTURES UNDER CRASH LOADINGS, Volume II. Technical Report	March 1976
	6. Performing Organization Code
7. Author(s) R. E. Welch; R. W. Bruce; T. Belytschko	8. Performing Organization Report No. J6321
9. Performing Organization Name and Address IIT Research Institute 10 West 35 Street Chicago, Illinois 60616	10. Work Unit No.
	11. Contract or Grant No. DOT-HS-105-3-697
12. Sponsoring Agency Name and Address Department of Transportation, NHTSA 400 7 Street S.W. Washington, D.C. 20590	13. Type of Report and Period Covered 6-30-1973 - 6-30-1975 Technical Final Report
	14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract A research project was undertaken to develop a finite element computer program for use in the dynamic analysis of vehicle structures, including sheet metal, in a crash environment. This research program involved the following major tasks:

- A technique was developed for the finite element analysis of the dynamic response of plate beam structures involving very large displacements and rotations and elastic-plastic material behavior. The principal feature of this technique involves the decomposition of the element displacement field into rigid body components and deformation components thus allowing the use of a small deflection formulation in the analysis.
- A computer program was developed incorporating this analysis procedure for beam and plate elements and rigid links together with appropriate time integration procedures and material property descriptions.

17. Key Words	18. Distribution Statement
finite element computer program, dynamic analysis, vehicles, crashworthiness	Document is available to the public through the National Technical Information Service, Springfield, Virginia 22161

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	157	

Form DOT F 1700.7 (8-69)

16. Abstract (Concl)

- A substantial number of test and demonstration problems were analyzed with this computer program ranging from simple classical solutions for beams and plates through large scale simulations of actual crash tests.

PREFACE

This final report, entitled "Finite Element Analysis of Automotive Structures under Crash Loadings" presents the results of a research project undertaken from 1 July 1973 to 30 June 1975 by IIT Research Institute (IITRI) for the Department of Transportation, National Traffic and Safety Administration (NHTSA) under Contract DOT-HS-105-3-697 (IITRI Project J6321). The report is presented in two volumes: Volume I, Summary Final Report, and Volume II, Technical Final Report.

The NHTSA Contract Technical Manager initially was Dr. Michael Chi and subsequently Mr. James Hackney. Dr. R. E. Welch served as the IITRI Project Manager under general supervision of Mr. A. Longinow, Manager, Structural Analysis Section, and Dr. K. E. McKee, Director, Engineering Mechanics. IITRI personnel who made significant technical contributions to the project are R. W. Bruce, M. Smith and R. E. Welch. Prof. T. Belytschko of the Circle Campus, University of Illinois, served as a consultant to the project.

CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

1. INTRODUCTION

The increased concern in recent years over vehicle safety has focused, in part, on the ability of a vehicle to sustain a crash event; to absorb, redirect and otherwise manage the severe forces and energy associated with the event; and thereby, to lessen the severity of the environment to which passengers are exposed. In addition to basic vehicle crashworthiness, serious attention has also been devoted to such factors as the propensity of a vehicle to damage in minor crash events and the threat posed to pedestrians or other vehicles by particular vehicle designs.

Since quantitative knowledge of the deformation characteristics of vehicle structures is an intrinsic requirement in all considerations of this type, it is not surprising that the developing concern for vehicle safety has given rise to intense and continuing efforts to provide mathematical descriptions of structural deformations in the crash environment. This report presents the results of a research project undertaken by IIT Research Institute (IITRI) for the Department of Transportation to develop a finite element computer program for use in the dynamic analysis of vehicle structure, including sheet metal, in a crash environment.

1.1 Previous Analyses

The literature relating to vehicle crash simulations is extensive and has been summarized by several previous investigators (Refs. 1 through 5). Present analytical procedures fall largely into two broad categories which may be termed lumped parameter or equivalent systems and formal approximation techniques (primarily finite element). In the former category, the vehicle is treated as an assemblage of lumped masses and resistance elements which are selected to represent specific subassemblies and components. The properties of the various elements of such models are determined by a static and dynamic crush testing, direct measurement, and the judicious use of simple analysis procedures. The use of such techniques has become widespread in studies of vehicle crashworthiness

as represented by Tani and Emory (Ref. 6), Kamal (Ref. 7), and the models developed by Battelle (Ref. 8) and Dynamic Sciences (Ref. 9). Such techniques are valuable in that they provide an efficient and inexpensive method of simulation, are fairly accurate for well established vehicle configurations and, in effect, serve as a depository for previous test and design results. The limitations of this type of procedure are principally that they are heavily reliant on previous testing; largely limited to well-established vehicle configurations and only weakly tied to basic principles of structural behavior.

The finite element approach represents an attempt to make up for the limitations inherent in the lumped parameter analyses, by employing more formal approximation techniques in the discretization of the structure and a greater reliance on more fundamental structural representations and properties. The limitations of this approach are found in the inherent tendency toward more complicated and expensive computations and the difficulties found in modeling the extensively complex phenomena associated with the crash environment. Such phenomena include large deflections and rotations in the deformed structure, regions of intense curvature (wrinkling), material strain rate effects and interference and contact among structural components during the response. These procedures, indeed, are not totally free of a reliance on testing and analytical judgment and, in fact, can be viewed as a somewhat more formal lumped parameter system and used in connection with the more empirical procedure in hybrid, combination models.

A variety of finite element analyses directed toward the dynamic analysis of vehicle structures have been reported previously. Selna and Salinas (Ref. 10) report a three-dimensional, linear elastic dynamic analysis of a two-door sedan in a 60 mph frontal barrier impact. Their results indicate that significant material yielding and collapse would occur and clearly demonstrate the body and engine pitch motions which ensue in such a crash. Kirioka (Refs. 11 through 13) in a series of papers have reported an elaborate computational system for the analysis of vehicle structures. The analysis includes nonlinear material and geometric effects, and

static, dynamic, and modal procedures and employs graphic systems and mesh generation as analysis aids. Thompson (Ref. 14) has investigated the side-on collision of passenger vehicles using finite element beam elements to model the vehicle structure. The elements include nonlinear material properties and a geometric stiffness matrix for beam columns and shows reasonable correlation with test data despite the lack of plate and membrane elements in the analysis. Shieh (Ref. 15) has developed a framework analysis for the precollapse dynamic response of plane, ideal, elastic-plastic frame structures which he applies to the analysis of framework devices used to attenuate crash effects. Finally, Young (Ref. 16) and Melosh (Ref. 17) have reported on an analysis employing an energy minimization technique for large displacements of elastic-plastic beam and truss elements. Results reported in Ref. 17 suggest that although the equivalent framework used is substantially stiffer than the actual structure the computed responses are similar in character to those obtained in test.

1.2 Analysis Requirements

The response of vehicle structures under crash loadings is a complex process primarily involving:

- transient, dynamic behavior
- complicated framework and shell assemblages
- large deflections and rotations
- extensive plastic deformation.

Previous attempts at a formal analysis of this process have been only partly successful due to a variety of limitations which, in particular instances, have included inadequacies in element formulations, material representations or solution procedures. The work presented in this report represents an attempt to develop a finite element program which is specifically tailored to the class of problems inherent in vehicle crash response, and which employs or extends current avenues in finite element analysis which seem best suited to such problems. The field of nonlinear finite element analysis is currently an extremely active area of research with an extensive, related literature and a variety of methods

and approaches. Consequently, a formal review of the field as background for the present analysis approach is not attempted. Instead, major features of the present work are briefly described, and some rationale is offered for their use in the context of vehicle analysis.

1.2.1 Solution Procedure – The analysis procedure is based on explicit, timewise numerical integration of the equations of motion for the node points (three translations and three rotations per node). The choice of an explicit, rather than implicit, procedure is exploited throughout the analysis by using "lumped" nodal masses and by calculating the internal forces at the nodes from direct integration of the element stress fields without reference to element or assembled stiffness matrices for the structure. The relative merits of implicit and explicit procedures and the importance of the mass formulation have been studied by Krieg and Key (Ref. 18), and the use of a direct calculation of internal forces has been described by Belytschko and Chiapetta (Ref. 19) and by Oden (Ref. 20). The choice of an explicit integration procedure and a direct calculation of nodal forces results in a program with minimum (but still substantial) computer storage requirements. This, in turn, is equivalent to a capability of processing reasonably detailed and extensive structural models with relative ease. An early capability for handling relatively large problems is considered essential in developing realistic simulations of actual crash events because of the complex geometry and construction found in real vehicles. Furthermore, this procedure lends itself to a reasonably well organized modular program which admits subsequent change and development, including the further incorporation of an implicit method.

1.2.2 Element Formulation – The treatment of large displacements and rotations employs a decomposition of the element displacement field into a rigid body rotation and translation associated with a local coordinate system attached to and moving with the element, and a remaining displacement field which describes the deformation of the element relative to the current position of the

element axes. This transformation, in effect, removes the average rigid body rotation of the element and allows the use of small or moderate deflection element formulations in the calculation of element forces. In this manner, extremely large rotations and deflections can be accommodated by the analysis with accuracy depending primarily on the size of the elements relative to the curvature of the structure. Although a formal convergence theorem is lacking, the decomposition does hold for infinitesimal regions and numerical studies show excellent agreement with classical solutions. Previous applications of this technique may be found in work by Argyris, Kelsey and Kamel (Ref. 21); Wempner (Ref. 22); Murray and Wilson (Ref. 23); Belytschko and Hsieh (Ref. 24); and other references which are cited in Section 2. The computer program at present includes low order triangular plate elements and three-dimensional beam elements. A triangular membrane-hinge line element is also available but is not presently compatible with the beam formulation.

1.2.3 Material Properties – The computer program currently uses simple elastic-plastic stress strain laws; a uniaxial relation for beam elements and a biaxial strain hardening Mises model for plates. Element forces and bending moments for given strain fields are calculated by piecewise linear numerical integration of the stresses at selected points in the cross section. The program is designed to accommodate other constitutive relations readily or to accept explicit relations among thrust, moment, extension and curvature. In fact, linear relationships are provided at this level which result in more efficient computations for this class of problems.

1.2.4 Program Validation – An extensive series of test problems were investigated throughout the program development to ensure that the basic formulation as well as the numerical procedures and programming were performing properly. These range from fairly simple cases such as beam bending, wave propagation and arch buckling problems which were designed to check the basic formulations, through more complex situations for which previously published numerical solutions could be found. Finally, a large demonstration was

attempted consisting of the simulation of actual vehicle-barrier tests. In a subsequent section, the test cases are summarized and the large-scale simulation is described in detail.

1.3 Report Organization

Section 2 of this report contains a description of the analysis formulation in fairly complete detail. In Section 3 the test problems are summarized and a detailed description of the large-scale vehicle-barrier test simulation is provided. A summary of the results of the investigation and recommendations for further application and development of the computer program are given in Section 4. Appendix A gives the complete plate element curvature functions and detailed descriptions of the program input, program operations and program listings are provided in Appendixes B, C, and D., respectively.

2. ANALYSIS FORMULATION

This description of the analysis formulation is intended to roughly parallel the sequence of computation in the computer program. A typical program cycle is: integrate equations of motion; transform displacement to element coordinate systems; calculate element forces; and transform forces to global and nodal coordinate systems. Solution procedures employed in the computer program are also described.

2.1 Equations of Motion

The equations of motion at a typical node point* for translations and rotations respectively, are written as follows:

Translations

$$[M][\ddot{u}_G] = [f_G^E] - [f_G^I] \tag{1}$$

where

$[M]$ - the diagonal, lumped mass matrix for the node

$[\ddot{u}_G] = [\ddot{u}_{xG}, \ddot{u}_{yG}, \ddot{u}_{zG}]$ is the acceleration vector for the node referred to a global system $[X_G, Y_G, Z_G]$ common to all nodes

$[f_G^E]$ - the vector of external, applied loads at the node referred to the global system

$[f_G^I]$ - the vector of forces at the node in the global system which results from deformations in all elements associated with the node.

*Unless otherwise specified, all equations and variables refer to a typical node or element, depending on context, rather than to the ensemble for all nodes or elements. Since there is no global stiffness matrix the formation of the ensembles is relatively simple.

Rotation

$$\overline{I_x}\,\overline{\alpha_x} = \bar{m}_x^E - \bar{m}_x^I + \overline{\omega_y}\,\overline{\omega_z}\,(\bar{I}_y - \bar{I}_z) \tag{2a}$$

$$\overline{I_y}\,\overline{\alpha_y} = \bar{m}_y^E - \bar{m}_y^I + \overline{\omega_z}\,\overline{\omega_x}\,(\bar{I}_z - \bar{I}_x) \tag{2b}$$

$$\overline{I_z}\,\overline{\alpha_z} = \bar{m}_z^E - \bar{m}_z^I + \overline{\omega_x}\,\overline{\omega_y}\,(\bar{I}_x - \bar{I}_y) \tag{2c}$$

where all components are referred to a coordinate system $[\bar{x},\bar{y},\bar{z}]$ which coincides with the principal axes of the lumped mass moments of inertia at each node and which rotates with the node (i.e., a set of rigid body axes for each node) and where $[\overline{I_x},\overline{I_y},\overline{I_z}]$ are the principal mass moments, $[\overline{\alpha_x},\overline{\alpha_y},\overline{\alpha_z}]$ are instantaneous angular accelerations, $[\overline{\omega_x},\overline{\omega_y},\overline{\omega_z}]$ are the angular velocities, and where $[\bar{m}^E]$ and $[\bar{m}^I]$ are the external and internal moment vectors at the node point.

The orientation of the nodal axes $[\bar{x},\bar{y},\bar{z}]$ with respect to the global axes at any time during the motion is established by the components of three unit vectors $\vec{b_1},\vec{b_2},\vec{b_3}$ which remain fixed along the nodal axes $[\bar{x},\bar{y},\bar{z}]$, respectively, as the node rotates. If the global components of these three unit vectors are

$$\vec{b_1}\text{: } (b_{x\bar{x}},b_{y\bar{x}},b_{z\bar{x}}) \tag{3a}$$

$$\vec{b_2}\text{: } (b_{x\bar{y}},b_{y\bar{y}},b_{z\bar{y}}) \tag{3b}$$

$$\vec{b_3}\text{: } (b_{x\bar{z}},b_{y\bar{z}},b_{z\bar{z}}) \tag{3c}$$

then the components of any vector, $\vec{v}$, transform from the nodal to global system by the transformation

$$[v_G] = [B][\bar{v}] \tag{4}$$

where the columns of [B] are the components of the three unit vectors

$$\begin{bmatrix} b_{x\bar{x}} & b_{x\bar{y}} & b_{x\bar{z}} \\ b_{y\bar{x}} & b_{y\bar{y}} & b_{y\bar{z}} \\ b_{z\bar{x}} & b_{z\bar{y}} & b_{z\bar{z}} \end{bmatrix} \tag{5}$$

Initially, the elements of the [B] matrix are obtained from the orientation of the principal axes of the mass moments of inertia for the node in the undeformed structure and are subsequently calculated from the integration of the rotational equations of motion; equations (2).

2.2 Time Integration

The numerical technique employed to integrate the equations of motion consists of the Newmark-beta method (Ref. 25). This method relates displacement, velocity, and acceleration at the beginning (u_o, v_o, a_o) and end (u_1, v_1, a_1) of a time interval, h, by the relations

$$u_1 = u_o + v_o h + (\tfrac{1}{2} - \beta)\, a_o h^2 + \beta h^2 a_1 \tag{6a}$$

$$v_1 = v_o + \frac{h}{2}(a_o + a_1) \tag{6b}$$

where β is an assumed parameter related to an assumed variation of acceleration with time over the interval. Solutions to the equations of motion, equations (1) and (2), may be obtained for a proper choice of β by using equations (6), (1), and (2) in an iterative manner, predicting u_1 and v_1 by means of equations (6) and correcting a_1 by means of equations (1) and (2) until the process converges. The full solution procedure, with iteration, is available in the computer program but early trials indicated that the most efficient process was obtained with $\beta = 0$ and no iterations, and all subsequent computations have been carried out with these parameters.

The solution procedure is applied to the equations of motion for translational degrees of freedom, equation (1), exactly in the manner outlined above. The treatment of the rotational degrees of freedom, equations (2), differs slightly in that the rotation term (analogous to u_o or u_1) has no physical significance as calculated and is not subsequently used in the calculations. Instead of an explicit calculation of nodal displacements or Euler angles, an auxiliary numerical integration is employed to calculate the current value of the [B] transformation based on its previous value and the current values of angular velocity and acceleration. Applying the integration algorithm (with $\beta = 0$) to a typical unit vector, $\vec{b}_3$, results in

$$\vec{b}_3\big|_{t+h} = \vec{b}_3\big|_t + \dot{\vec{b}}_3 h + \frac{1}{2} h^2 \ddot{\vec{b}}_3 \tag{7}$$

The derivatives of the unit vector are related to the angular velocity and acceleration by the relations

$$\dot{\vec{b}}_3 = \vec{\omega} \times \vec{b}_3 \tag{8a}$$

$$\ddot{\vec{b}}_3 = \dot{\vec{\omega}} \times \vec{b}_3 + \vec{\omega} \times \dot{\vec{b}}_3$$

$$= \dot{\vec{\omega}} \times \vec{b}_3 + \vec{\omega} \times \vec{\omega} \times \vec{b}_3 \tag{8b}$$

Expanding into scaler components in the previous $[\bar{x},\bar{y},\bar{z}]$ system, these relations become

$$b_{3\bar{x}}\Big|_{t+h} = + \bar{\omega}_y h + \frac{h^2}{2}(\bar{\omega}_x\bar{\omega}_z + \alpha_y) \tag{9a}$$

$$b_{3\bar{y}}\Big|_{t+h} = - \bar{\omega}_x h + \frac{h^2}{2}(\bar{\omega}_y\bar{\omega}_z + \alpha_x) \tag{9b}$$

with the third component, $b_{3\bar{z}}$, obtained from the condition that $\vec{b}_3$ remains a unit vector. A similar process is applied to another of the unit vectors, say $\vec{b}_2$, and the third vector is obtained from the cross product relation, $\vec{b}_1 = \vec{b}_2 \times \vec{b}_3$. The components of the new unit vectors referred to the global coordinate system then constitute the current transformation matrix [B].

2.3 Element Coordinate Systems and Deformations

A coordinate system $[\hat{x},\hat{y},\hat{z}]$ is embedded in each element and serves to define the rigid body rotation of the element and as a reference for element distortions and forces. The components of a vector, $\vec{V}$, transform from the element coordinate system to the global coordinate system by the time varying transformation

$$[v_G] = [E][\hat{v}] \tag{10}$$

where the elements of E are

$$\begin{bmatrix} e_{x\hat{x}} & e_{x\hat{y}} & e_{x\hat{z}} \\ e_{y\hat{x}} & e_{y\hat{y}} & e_{y\hat{z}} \\ e_{z\hat{x}} & e_{z\hat{y}} & e_{z\hat{z}} \end{bmatrix} \tag{11}$$

which correspond columnwise with the global components of unit vectors $\vec{e}_1, \vec{e}_2, \vec{e}_3$ oriented with the element axes $[\hat{x}, \hat{y}, \hat{z}]$ respectively, that is

$$\vec{e}_1: [e_{x\hat{x}}, e_{y\hat{x}}, e_{z\hat{x}}] \tag{12a}$$

$$\vec{e}_2: [e_{x\hat{y}}, e_{y\hat{y}}, e_{z\hat{y}}] \tag{12b}$$

$$\vec{e}_3: [e_{x\hat{z}}, e_{y\hat{z}}, e_{z\hat{z}}] \tag{12c}$$

2.3.1 Plate Elements – The element coordinate system for the three-dimensional plate element (Figure 1) is constructed from the current positions (i.e., undeformed position plus displacement) of the three points (I,J,K) which define the element. In terms of unit vectors pointing from I to J ($\vec{S}_{IJ}$) and from I to K ($\vec{S}_{IK}$) the global components of the three unit vectors, $\vec{e}_1, \vec{e}_2, \vec{e}_3$ are calculated as

$$\vec{e}_1 = (\vec{S}_{IJ} + \vec{S}_{IK})/2 \tag{13a}$$

$$\vec{e}_3 = \vec{S}_{IJ} \times \vec{S}_{IK} \tag{13b}$$

$$\vec{e}_2 = \vec{e}_3 \times \vec{e}_1 \tag{13c}$$

These operations apply equally well to both the deformed and undeformed positions of the element and, in particular, define $\vec{e}_3^{\,o}$, the normal to the undeformed plate surface, which is subsequently employed to establish element rotations in the deformed position.

Element displacements and rotations are defined with respect to the element coordinate system and are determined from the global components of displacements and the transformations [B] associated with the three node points I,J,K. Having determined the [E] transformation as described above the global components of displacement, u_G, at each of the nodes as determined from the integration of equation (1) can be transformed to the element system by the operation

$$[\hat{u}] = [E]^T [u_G] \tag{14}$$

where $[E]^T$ denotes the transpose of [E]. Since the three node points uniquely determine the transformation [E] this calculation

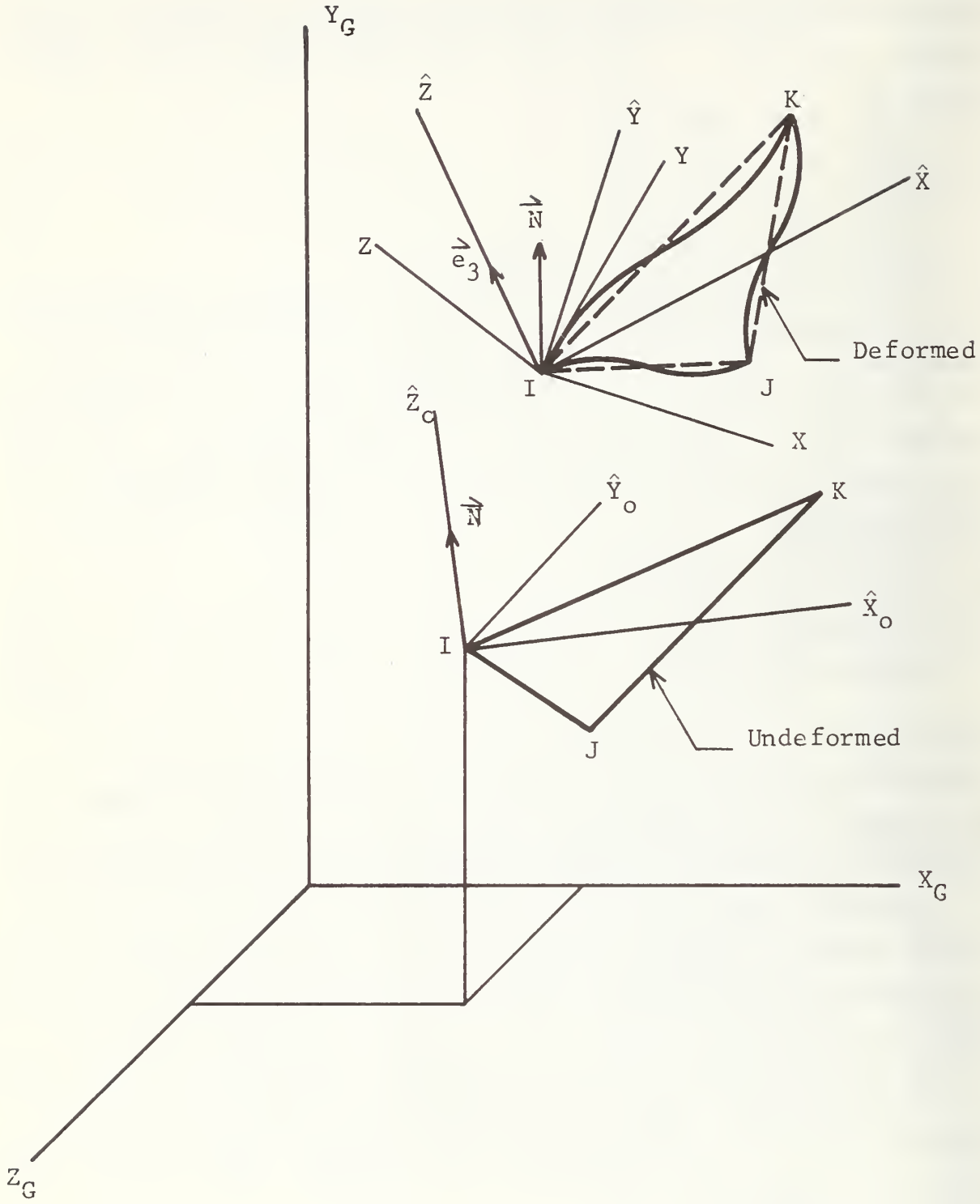


Figure 1 Three-Dimensional Plate Element

would eliminate the component of displacement normal to the plane of the plate, $\hat{u}_z = 0$. For a small deflection plate formulation then the nodal displacements $\hat{u}_x, \hat{u}_y$ lie in the plane IJK and determine the membrane midsurface strains while the plate curvatures are determined by the rotations alone. The computer program actually employs a somewhat different, but equivalent and more efficient, approach to the calculation of membrane strains. The lengths of the element sides in the deformed position are calculated directly from the global displacements and these new lengths determine the extensions parallel to the three element sides. For instance, side IJ

$$E_{IJ} = \frac{L_{IJ} - L^o_{IJ}}{L^o_{IJ}} \tag{15}$$

where L_{IJ} and L^o_{IJ} are the lengths of the side IJ in the deformed and undeformed positions respectively. Since a constant membrane strain element is employed the three extensions uniquely determine the membrane strain in the element.

The calculation of rotations at a typical node for a plate element involves both the [E] transformation for the element and the [B] transformation for the node and is based on the usual thin plate assumption that lines which are initially straight and normal to the surface of the plate remain so during the deformation. In the present terminology, an initial normal to the plate, $\vec{e}^o_3$, located at node I rotates with the nodal axes (i.e., $[\bar{x},\bar{y},\bar{z}]$ system) at node I and its components in the nodal system remain unchanged during the motion. These components are given by

$$[\bar{n}] = [B_o]^T [E_o] [\hat{e}^o_3] \tag{16}$$

where

$$[\hat{e}^o_3] = [0,0,1]^T$$

and $[\bar{n}]$ are the components of the normal to the deformed surface, $\vec{n}$, at node I. It should be noted that since the components of $\vec{n}$ are constant in the nodal coordinate system which rotates during

the deformation, the components of $\vec{n}$ in the global or element systems are time varying. At any time during the deformation the angle, θ, between the normal to the deformed plate at node I, $\vec{n}$, and the current normal to the rigid body plane of the plate, $\vec{e_3}$, may be calculated from the vector cross product

$$\vec{\theta} = \sin\theta \, \vec{e_\theta} = \vec{e_3} \times \vec{n} \tag{17}$$

Assuming that this is a sufficiently small angle, its components in the element system are

$$\hat{\theta}_x = - \hat{n}_y \tag{18a}$$

$$\hat{\theta}_y = \hat{n}_x \tag{18b}$$

where the components of the deformed normal, $\vec{n}$, in the element system are determined from

$$[\hat{n}] = [E]^T[B][\bar{n}] \tag{19}$$

The rotations $\hat{\theta}_x, \hat{\theta}_y$ (two per node per element) are used in subsequent calculations of element curvatures and, in turn, element forces and moments

2.3.2 Beam Elements – The element coordinate system for the three-dimensional beam element (Figure 2) is constructed somewhat differently than for the plate element. In the undeformed position, the unit vector $\vec{e_1^o}$ lies along the axis of the beam (the $\hat{x}_o$ axis) and is defined by the positions of the end point I and J. The unit vector, $\vec{e_2^o}$ (the $\hat{y}_o$ axis) is directed normal to the beam axis and lying in a plane defined by a reference or pointer node specified in the input data and containing one of the principal axes of the beam cross section. The third unit vector is determined by $\vec{e_3^o} = \vec{e_1^o} \times \vec{e_2^o}$ and establishes the $\hat{z}_o$ axis. In this case two unit vectors $\vec{\eta}$ and $\vec{\xi}$ which initially coincide with the undeformed $\hat{x}_o$ and $\hat{y}_o$ axes are embedded in the nodal coordinate system $[\bar{x}, \bar{y}, \bar{z}]$ at each end of the beam. The nodal components of these vectors are

$$[\bar{\eta}] = [B_o]^T[E_o][\hat{e}_1^o] \tag{20}$$

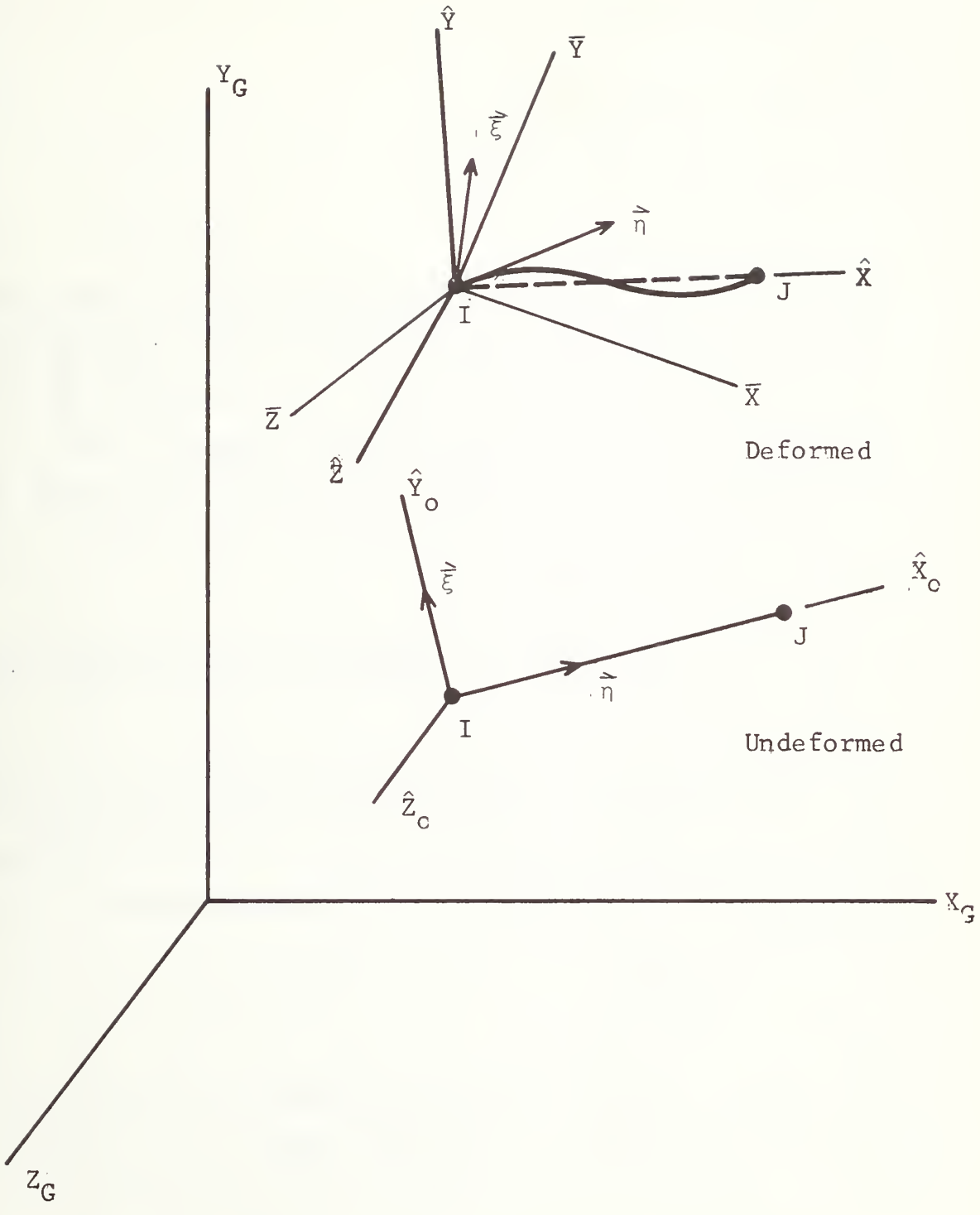


Figure 2 Three-Dimensional Beam Element

where

$$[\hat{e}_1^o] = [1,0,0]^T$$

and

$$[\bar{\xi}] = [B_o]^T[E_o][\hat{e}_2^o] \tag{21}$$

where

$$[\hat{e}_2^o] = [0,1,0]$$

The construction of element axes for the deformed beam differs somewhat from the procedure employed for the plate since the position data for the two nodes I and J are not sufficient to locate the $\hat{y}$ and $\hat{z}$ axes uniquely, although they are sufficient to establish the $\hat{x}$ axes and the $\vec{e_1}$ unit vector directly. The $\hat{z}$ is taken to be normal to the plane defined by the $\hat{x}$ axis and the average of the unit vectors $\vec{\xi}_I$ and $\vec{\xi}_J$ at the ends of the beam

$$\vec{e_3} = \vec{e_1} \times (\vec{\xi}_I + \vec{\xi}_J)/2 \tag{22}$$

and the $\hat{y}$ axis is formed to complete the right-handed system

$$\vec{e_2} = \vec{e_3} \times \vec{e_1} \tag{23}$$

Displacements in the element coordinate system are determined in a manner analogous to that used for the plate element. The extension of the beam is calculated directly from the deformed and undeformed lengths as determined by the displacement data for the end points

$$E_{IJ} = \frac{L_{IJ} - L_{IJ}^o}{L_{IJ}^o} \tag{24}$$

The rotations related to bending at node I are found from the unit vector $\vec{\eta}$ which defines the deformed position of the beam axis at the node and $\vec{e_1}$ which lies along the $\hat{x}$ axis by means of the relation

$$\vec{\theta} = \vec{e_1} \times \vec{\eta} \tag{25}$$

For small angles this results in

$$\hat{\theta}_z = \hat{\eta}_y \tag{26a}$$

$$\hat{\theta}_y = -\hat{\eta}_z \tag{26b}$$

where the components of $\vec{\eta}$ in the element system are determined from

$$[\hat{\eta}] = [E]^T[B][\bar{\eta}] \tag{27}$$

The torsional rotation $\hat{\theta}_x$ is determined directly from the cross product relation $(\vec{\xi}_I \times \vec{\xi}_J)$ in terms of components $[\hat{\eta}_I]$, $[\hat{\eta}_J]$ in the element system.

2.3.3 Rigid Links – The computer program incorporates a rigid link or master-slave node relationship whereby one or more nodes may be identified as being rigidly connected to and driven by another master node. The slave or driven nodes have no mass properties and their motions are determined by the displacements and rotations of the master node and the appropriate kinematic relations for a connecting rigid link which is embedded in the nodal (rigid body) axes for the master. This feature is intended to be used as a means of introducing rigid body masses into a structural model to simulate such items as a vehicle center of gravity, engine blocks, wheel suspensions, etc.

In terms of the nodal axes $[\bar{x},\bar{y},\bar{z}]$ and transformations [B] previously defined, the equations governing these links can be developed in a straightforward manner. If I and J are taken to denote the slave and master nodes, respectively, then the position of the slave node with respect to the master in the undeformed structure may be written as

$$[R_{IG}] = [R_{JG}] + [\Delta_G^o] \tag{28}$$

where $[R_{IG}] = [X_{IG}, Y_{IG}, Z_{IG}]$ is the position vector for node I in global coordinates and $[\Delta_G^o]$ is a vector whose components are the initial differences in position between the slave and master nodes. A rigid body motion of node I with respect to node J implies the components of $[\Delta_G^o]$ in the nodal coordinate system at node J are constant during the motion.

These components are

$$[\bar{\Delta}] = [B_{Jo}]^T[\Delta_G^o] \tag{29}$$

Thus, at any time during the motion the position of node I is determined from the position of node J by

$$[R_{IG}] + [u_{IG}] = [R_{JG}] + [u_{JG}] + [\Delta_G] \tag{30}$$

where $[u_{IG}]$ and $[u_{JG}]$ are the displacements of nodes I and J and $[\Delta_G]$ are the current global components of the rigid link as given by

$$[\Delta_G] = [B_J][\bar{\Delta}] \tag{31}$$

Equation (30) may be rearranged to give the displacement of node I directly as

$$[u_{IG}] = [u_{JG}] + [\Delta_G] - [\Delta_G^o] \tag{32}$$

2.4 Element Forces

Having reduced the displacements to rigid body and deformation components, the next step in the process is to calculate the forces on the nodes, taken element by element, from the displacements referred to the element coordinate systems. In general, the procedure by which the nodal forces are calculated is derived from the principal of virtual work which states that for a kinematically admissible variation in the element displacements, the work done by the nodal forces is equal to the work done by the stresses in the element. In the present notation this relation for a typical element is written as

$$[f]^T[\delta u] = \int_V [\sigma(\varepsilon)]^T[\delta\varepsilon]\, dV \tag{33}$$

where [f] is the vector* of all forces and moments at the nodes (i,j,...) defining the element

$$[f]^T = [f_i, f_j, \ldots]$$

$$[f_i]^T = [f_{ix}, f_{iy}, f_{iz}, m_{ix}, m_{iy}, m_{iz}]$$

[u] is the corresponding vector of element displacements, $[\sigma]$ and $[\varepsilon]$ are the biaxial stresses and strains at a point in the plate

$$[\sigma] = [\sigma_{xx}, \sigma_{yy}, \sigma_{xy}]$$

$$[\varepsilon] = [\varepsilon_{xx}, \varepsilon_{yy}, \varepsilon_{xy}]$$

and δ denotes a variation.

The strains are related to the displacements through a matrix consisting of the appropriate derivatives of the element displacement functions

$$[\varepsilon] = [\Phi(x,y,z)][u] \tag{34}$$

Introducing this relation into equation (33) and noting that the resulting expressions must be satisfied term by term in the virtual displacements, leads to the following definition of the forces acting on the nodes.

$$[f]^T = \int_V [\sigma(\varepsilon)]^T [\Phi] dV \tag{35}$$

The development of the nodal forces for particular elements is described below in terms that are convenient to each element formulation.

2.4.1 Plate Element – The plate element used in the program is a flat triangular element lying in the x-y plane of the element coordinate system defined by the deformed positions of the corner nodes, i,j,k (see Figure 3).

*The superscript notation $[\hat{f}]$, $[\hat{u}]$, etc. is omitted since all quantities involved in this section are referred to a typical element coordinate system $[\hat{x},\hat{y},\hat{z}]$.

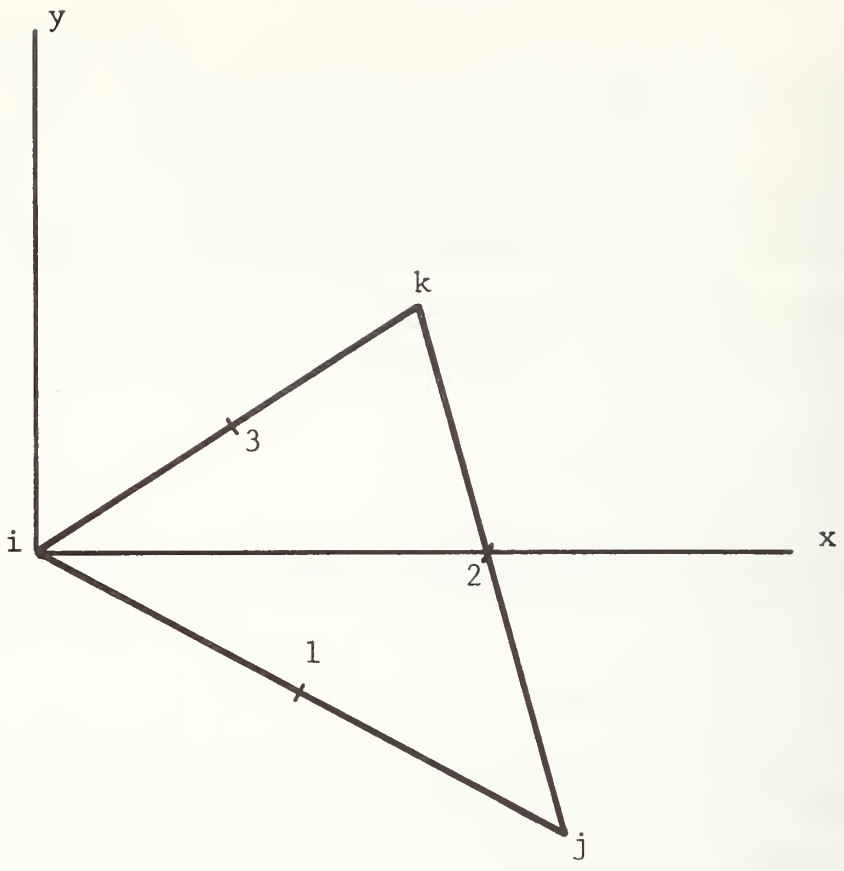


Figure 3 Triangular Plate Element

The element formulation is based on small deflection linear plate theory involving a linear displacement (i.e., constant strain) field for the midplane deformations and a cubic displacement field for the plate bending deformations as given by Zienkiewicz (Ref. 26). Node point degrees of freedom consist of the two in-plane motions u_x and u_y which determine the midplane strains and the two nodal rotations θ_x and θ_y which determine the plate curvatures. The out-of-plane displacements, u_z, of the nodes are zero since the x-y element coordinate plane was established by the deformed position of the nodes. In addition, nodal rotations normal to the plane of the plate, θ_z, are not admitted as element degrees of freedom (as in Ref. 26) for the element formulation which is employed.

Strain Displacement Relations: It is convenient to represent the element strains $[\varepsilon(x,y,z)]$ in terms of the midplane strains $[\varepsilon^o(x,y)]$ and curvatures $[\kappa(x,y)]$ in the form

$$[\varepsilon] = [\varepsilon^o] - z[\kappa] \tag{36}$$

where

$$[\varepsilon^o] = [\varepsilon^o_{xx},\ \varepsilon^o_{yy},\ \varepsilon^o_{xy}]^T$$

$$[\kappa] = [\kappa_{xx},\ \kappa_{yy},\ \kappa_{xy}]^T$$

since these quantities uncouple the midplane strain and out-of-plane bending displacement fields. By regrouping the displacements and rotations at the element nodes in a corresponding manner the midplane strains and curvatures may be written as

$$\begin{aligned} [\varepsilon^o] &= [\phi][u] \\ [\kappa] &= [\psi][\theta] \end{aligned} \tag{37}$$

where $[\phi]$ and $[\psi]$ together are equivalent to $[\Phi]$ in equation (35) and

$$[u] = [u_{ix}, u_{iy}, u_{jx}, u_{jy}, u_{kx}, u_{ky}]^T$$

$$[\theta] = [\theta_{ix}, \theta_{iy}, \theta_{jx}, \theta_{jy}, \theta_{kx}, \theta_{ky}]^T$$

The elements of $[\phi]$ and $[\psi]$ are derivable from the shape functions given in Ref. 26. In particular $[\phi]$ is constant over the element and in terms of local element coordinates (with $x_i = 0$ and $y_i = 0$), has the form

$$[\phi] = \frac{1}{2A}\begin{bmatrix} y_j - y_k, & 0, & y_k, & 0, & -y_j, & 0 \\ 0, & x_k - x_j, & 0, & -x_k, & 0, & x_j \\ x_k - x_j, & y_j - y_k, & -x_k, & y_k, & x_j, & -y_j \end{bmatrix} \tag{38}$$

where A is the area of the element. The elements of $[\psi]$ vary linearly over the plate element and are obtained by differentiation of the cubic shape functions as given in Ref. 26 and by Meek

(Ref. 27) with the curvatures defined as

$$\begin{bmatrix} \kappa_{xx} \\ \kappa_{yy} \\ \kappa_{xy} \end{bmatrix} = \begin{bmatrix} \frac{\partial^2 u_z}{\partial x^2} \\ \frac{\partial^2 u_z}{\partial y^2} \\ 2\,\frac{\partial^2 u_z}{\partial x \partial y} \end{bmatrix} = [\psi][\theta] \tag{39}$$

where u_z is the transverse plate displacement. The derivation of the elements of $[\psi]$ is given in detail in Appendix A.

Nodal Forces and Moments: Forces and moments acting at the nodes are calculated from equation (35) as specialized to the nomenclature for this element. The in-plane forces are obtained from the integrals

$$\begin{bmatrix} f_{ix} \\ f_{iy} \\ f_{jx} \\ f_{jy} \\ f_{kx} \\ f_{ky} \end{bmatrix}^T = \int_A [P]^T [\phi]\, dA \tag{40}$$

where the cross section plate forces [P] are obtained by integrating through the thickness

$$[P] = \begin{bmatrix} P_{xx} \\ P_{yy} \\ P_{xy} \end{bmatrix} = \int_{-h/2}^{h/2} [\sigma(\varepsilon)]\, dz \tag{41}$$

In a similar manner the nodal moments are calculated from integrals of the form

$$\begin{bmatrix} m_{ix} \\ m_{iy} \\ m_{jx} \\ m_{jy} \\ m_{kx} \\ m_{ky} \end{bmatrix}^T = \int_A [M]^T [\psi] dA \tag{42}$$

where the cross section plate moments [M] are also obtained by integration through the thickness

$$[M] = \begin{bmatrix} M_{xx} \\ M_{yy} \\ M_{xy} \end{bmatrix} = -\int_{-h/2}^{h/2} Z[\sigma(\varepsilon)] dA \tag{43}$$

The remaining nodal forces f_{iz}, f_{jz}, f_{kz} are found from overall equilibrium of the element after the nodal moments have been established. Taking moments about node i, the origin of the element coordinate system, results in the following relation for the forces

$$\begin{bmatrix} f_{jz} \\ f_{kz} \end{bmatrix} = \frac{1}{2A} \begin{bmatrix} y_3 & -x_3 \\ -y_2 & x_2 \end{bmatrix} \begin{bmatrix} m_{iy} + m_{jy} + m_{ky} \\ -m_{ix} + m_{jx} + m_{kx} \end{bmatrix} \tag{44}$$

The remaining force f_{iz} is found from transverse equilibrium

$$f_{iz} = -f_{jz} - f_{kz} \tag{45}$$

Numerical Integrations: The integrals involved in the definitions of cross section forces and moments (equations (41) and (43)) are evaluated numerically at the midpoints of the three sides of the triangle (points 1, 2, and 3 in Figure 3). The integration procedure involves a piecewise linear trapezoidal integration rule employing stresses at 'n' evenly spaced points

through the thickness with the first and last points lying on the outer surfaces of the plate. At least two points must be used since this number provides exact results for elastic problems and four or five points have proven to provide satisfactory accuracy for all cases in which detailed accuracy studies have been made. The general forms given in equations (41) and (43) apply to any consistent, nonlinear biaxial stress strain relations although applications to date have been limited to bilinear elastic-plastic material properties. It should be noted that the formal definition of internal forces and moments has been maintained in the computer program so that explicit relations among extension, curvature, force and moment may be incorporated with relative ease. This has been found useful in purely elastic problems as a means of improving the efficiency of the program.

Integrations over the surface of the plate (equations (40) and (42)) are evaluated by three point Gaussian integration using the three midside points described above. In this case (see Ref. 26) the weighting factors are 1/3 and a typical integration (equation (42) for instance) results in

$$\int_A [M]^T[\psi]dA = \frac{A}{3}\sum_L [M_L]^T[\psi_L] \qquad L = 1,2,3 \tag{46}$$

where $[M_L]$ and $[\psi_L]$ are the values at the midside points.

Stress-Strain Relations: The stress-strain relations employed in the plate element formulation are those of a bilinear, elastic-plastic strain hardening material as indicated in Figure 4. The plastic flow calculations are based on a Mises yield criterion and isotropic hardening for the plane stress conditions which are assumed to apply in the plates. The computational algorithm follows that presented by Hartzman and Hutchinson (Ref. 28) as specialized for small strain and plane stress conditions. Given a prior stress state at a point in the plate, $\sigma^o_{xx}, \sigma^o_{yy}, \sigma^o_{xy}$ and a small increment in strain from the prior to the current state $\Delta\varepsilon_{xx}, \Delta\varepsilon_{yy}, \Delta\varepsilon_{xy}$ the current stress state is established by the following procedure.

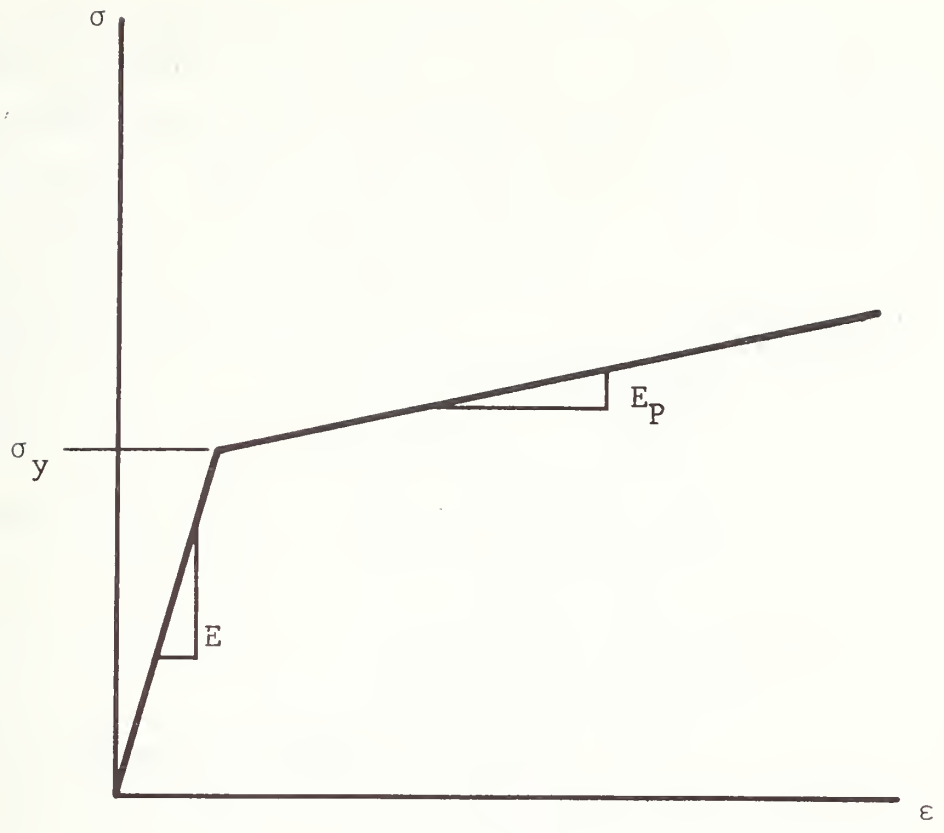


Figure 4 Uniaxial Bilinear Stress-Strain Relation

A tentative, current stress state $\overline{\sigma_{xx}}, \overline{\sigma_{yy}}, \overline{\sigma_{xy}}$ is calculated as though the strain increment were completely elastic:

$$\overline{\sigma_{xx}} = \sigma^{o}_{xx} + \frac{E}{1-\nu^2}(\Delta\varepsilon_{xx} + \nu\ \Delta\varepsilon_{yy}) \tag{47}$$

$$\overline{\sigma_{yy}} = \sigma^{o}_{yy} + \frac{E}{1\ \nu^2}(\Delta\varepsilon_{yy} + \nu\ \Delta\varepsilon_{xx}) \tag{47}$$

$$\overline{\sigma_{xy}} = \sigma^{o}_{xy} + G\ \Delta\varepsilon_{xy}$$

where E, ν and G are the usual elastic constants. The corresponding value of effective stress for the Mises yield criterion is also calculated from the expression

$$\overline{\sigma_e} = [\overline{\sigma}^2_{xx} - \overline{\sigma}_{xx}\overline{\sigma}_{yy} + \overline{\sigma}^2_{yy} + \frac{3}{2}\overline{\sigma}_{xy}]^{1/2} \tag{48}$$

which is used to determine if plastic flow has taken place during the strain increment. If $\bar{\sigma}_e$ is less than the most recent effective stress σ_e^o at which yielding occurred the material is unloading and the tentative stresses are, in fact, the current stress values $\sigma_{xx}, \sigma_{yy}, \sigma_{xy}$. Otherwise the tentative stress state must be modified to correct for plastic behavior. As shown in Ref. 28, the true value of the current effective stress will be

$$\sigma_e = \frac{\left[\sigma_e^o + \left(\frac{H'}{3G}\right)\ \bar{\sigma}_e\right]}{1 + \left(\frac{H'}{3G}\right)} \tag{49}$$

where H' is the slope of the stress versus plastic strain curve given by

$$H' = \frac{E\ E_p}{E + E_p}$$

with E_p being the plastic modulus shown in Figure 4. The current stress state is then given by

$$\begin{aligned} \sigma_{xx} &= \frac{1}{(1+3\lambda)}\ [\bar{\sigma}_{xx} + \lambda(\bar{\sigma}_{xx} + \bar{\sigma}_{yy})] \\ \sigma_{yy} &= \frac{1}{(1+3\lambda)}\ [\bar{\sigma}_{yy} + \lambda(\bar{\sigma}_{xx} + \bar{\sigma}_{yy})] \\ \sigma_{xy} &= \frac{1}{1+3\lambda}\ [\bar{\sigma}_{xy}] \end{aligned} \tag{50}$$

where the factor, λ, is given by

$$\lambda = \frac{1}{3}\left(\frac{\bar{\sigma}_e}{\sigma_e} - 1\right)$$

Further details concerning the development of this algorithm may be found in Ref. 28.

2.4.2 Beam Element – The beam element employed in the program is based on the conventional small deflection formulation involving cubic displacement fields (i.e., linear curvature distribution) for the transverse displacements and linear displacement distributions for the axial and torsional motions (i.e., constant axial strain and twist). The development of the element forces parallels that used for the plate with such changes and simplifications as are appropriate for beam theory.

Strain-Displacement Relations: The longitudinal strain in the beam, $\varepsilon_{xx}(x)$, is written in terms of the extension at the centroid $\varepsilon^{o}(x)$ and curvatures $\kappa_{yy}(x)$ and $\kappa_{zz}(x)$ about the principal axes of the cross section (see Figure 5) in the form

$$\varepsilon_x = \varepsilon^o - y\,\kappa_{zz} - z\,\kappa_{yy} \tag{51}$$

The extensions and curvatures are in turn written in terms of the nodal displacements using the appropriate derivatives of the displacement fields which results in

$$\varepsilon^o = \frac{(u_{xj} - u_{xi})}{L}$$

$$\kappa_{yy} = 2(2-3\xi)\,\frac{\theta_{yi}}{L} + 2(1-3\xi)\,\frac{\theta_{yj}}{L} \tag{52}$$

$$\kappa_{zz} = 2(3\xi-2)\,\frac{\theta_{zi}}{L} + 2(3\xi-1)\,\frac{\theta_{zj}}{L}$$

where L is the length of the beam and $\xi = X/L$.

Nodal Forces and Moments: Forces and moments acting at the nodes are determined by introducing these strain displacement relations, equations (51) and (52), into the appropriate form of equation (35) with the following results:

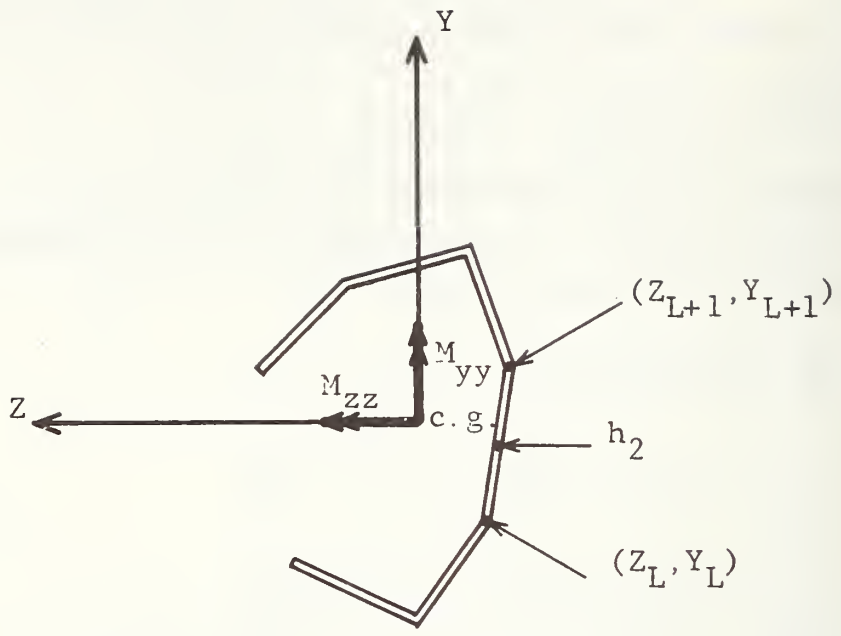
$$f_{xj} = \frac{1}{L}\int_L P_{xx}\,dx; \qquad f_{xi} = -f_{xj} \tag{53}$$

where P_{xx} is the internal cross section force

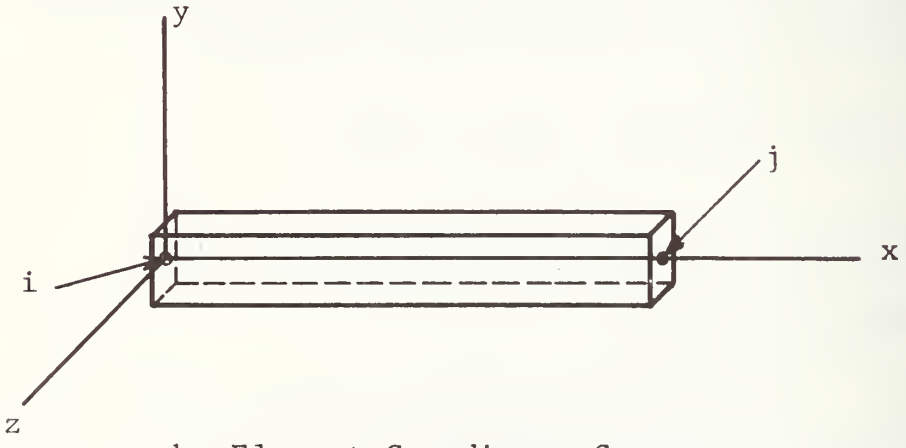
$$P_{xx} = \int_A \sigma_{xx}\,dA$$

$$m_{yi} = -\frac{2}{L}\int_L (2-3\xi)\,M_{yy}\,dx \tag{54}$$

$$m_{yj} = -\frac{2}{L}\int_L (1-3\xi)\,M_{yy}\,dx$$



a. Cross Section Geometry and Moments (Positive Face)



b. Element Coordinate System

Figure 5 Beam Coordinates and Cross Section Geometry

where M_{yy} is the internal cross section moment* about the y axis

$$M_{yy} = \int_A z\, \sigma_{xx}\, dx$$

$$m_{zi} = \frac{2}{L}\int_L (3\xi-2)\, M_{zz}\, dx \qquad (55)$$

$$m_{zy} = \frac{2}{L}\int_L (3\xi-1)\, M_{zz}\, dx$$

where M_{zz} is the internal cross section moment about the z axis

$$M_{zz} = -\int_A y\, \sigma_{xx}\, dA$$

The torsional moments are omitted from the formal development given above and are obtained directly in terms of the torsional stiffness for the beam section as

$$m_{xj} = K_t\, G\, \frac{(\theta_{xj} - \theta_{xi})}{L}\ ; \quad m_{xi} = -m_{xj} \qquad (56)$$

where K_t is the torsional constant for the cross section and G is the shear modulus. Finally, the transverse nodal forces are determined from equilibrium conditions using the nodal moments as

$$f_{zj} = \frac{m_{yi} + m_{yj}}{L}\ ; \qquad f_{zi} = -f_{zj}$$

$$f_{yj} = \frac{m_{zi} + m_{zj}}{L}\ ; \qquad f_{yi} + -f_{yj} \qquad (57)$$

Numerical Integrations: The integrals involved in the definition of internal cross section moments and forces are obtained by numerical integration of the stress distribution over the cross section. The cross sections are treated as thin-walled plate work sections defined as indicated in Figure 5 by tables of the coordinates of points on the centerline of the wall and the thicknesses

*The definitions of the cross section moments, M_{yy} and M_{zz} are chosen to be in the positive sense on the positive face of the beam cross section and to result in the usual sign convention for elastic material, i.e., $M_{zz} = E\, I_{zz}\, \kappa_{zz}$ and $M_{yy} = -EI_{yy}\, \kappa_{yy}$.

of wall sections. The integration algorithm is a generalization to two dimensions (the y-z plane) of the procedure employed in the plate element and assumes a piecewise linear variation in stress around the periphery of the cross section. As in the plate cross section integrations this procedure preserves the exact elastic stiffness of the cross section provided that integration points include all corners in the cross section. The stress strain algorithm relation employed in these integrations is the one-dimensional version of that prescribed in connection with the plate element which can be obtained from equations (47) through (50) simply by deleting all occurrences of σ_{yy} and σ_{xy}. As with the plate the definitions of internal cross section forces and moments have been maintained in the computer program to facilitate subsequent use of explicit definitions of these quantities.

The integrals over the length of the beam are not carried out explicitly but instead are evaluated in terms of P_{xx}, M_{yy}, M_{zz} at the end points of the beam:

$$
\begin{aligned}
f_{xj} &= \frac{1}{2}\left[P_{xx}\Big|_{x=o} + P_{xx}\Big|_{x=L}\right] \\
m_{yi} &= -M_{yy}\Big|_{x=o} \qquad m_{yj} = M_{yy}\Big|_{x=L} \\
m_{zi} &= -M_{zz}\Big|_{x=o} \qquad m_{zj} = M_{zz}\Big|_{x=L}
\end{aligned}
\tag{58}
$$

For the moments this is equivalent to assuming that the internal moments M_{yy} and M_{zz} vary linearly along the beam and is consistent with the usual beam formulation which implies constant shear and linear moment variation along the beam

2.5 Nodal Force Transformations

The preceding section describes the manner in which forces and moments acting on the nodes of a given element are calculated for specified values of nodal displacement. These are calculated on an element by element basis with reference to the individual element coordinate system $(\hat{x},\hat{y},\hat{z})$ and must be transformed to the global (x_G,y_G,z_G) and nodal $(\bar{x},\bar{y},\bar{z})$ systems respectively in order

to be entered into the appropriate equations of motion (equations (1) and (2)). For a particular element the nodal forces, $[\hat{f}]$, are transformed to the global system by means of the element transformation

$$[f_G] = [E][\hat{f}] \tag{59}$$

and entered directly into the force summations in equation (1). The moments $[\hat{m}]$ are first transformed to the global system and, if the nodes in question are not part of a master-slave relation, subsequently to the nodal coordinate system for the particular node

$$[\bar{m}] = [B]^T[E][\hat{m}] \tag{60}$$

and entered into the moment summations (equations (2)). If the node in question is associated with a master node, the moments are first transformed to the global system by the transformation

$$[m_{GI}] = [E][\hat{m}_I] \tag{61}$$

where I denotes quantities at the slave node, and then transferred to the master node (denoted by J) by the cross-product relation

$$\vec{m}_{JG} = \vec{m}_{IG} + \vec{\Delta}_G \times \vec{f}_G \tag{62}$$

The moments for the master node $[m_{JG}]$ are then transformed to the nodal system for that node

$$[\bar{m}_J] = [B]^T[m_{JG}] \tag{63}$$

and entered into the appropriate moment summations in equation (2).

3. ANALYSIS RESULTS

A substantial number of test and demonstration problems have been treated during the development of the analysis and computer program reported here. For the most part the test problems dealt with relatively simple structural configurations having known results and were selected to isolate and test particular aspects of the formulation and program during development. Thirteen such test problems of successively increasing complexity were formally reported in progress reports. In all cases the analyses were pursued until satisfactory comparison with previous results were obtained or at least to the point at which differences between results could clearly be reconciled on the grounds of differences in loadings, boundary conditions, or ambiguities in the results used for comparison purposes. The contents of these test problems are recapitulated briefly in Table 1, and results for two more complicated problems, namely dynamic snap-through for an elastic dome and the response of an elastic-plastic panel under impulsive load are reviewed in the following sections. Results are also presented in subsequent sections which involve the simulation of two actual crash events taken from a previous series of full scale barrier crash tests.

3.1 Test Problems

Results for two test problems are presented in this section. These have been chosen to demonstrate the capabilities of the analysis procedure in terms of fairly difficult problems for which some results are available in the literature. It should be noted however that the results used as references are also based on numerical computations and approximations and are not considered to be "exact" in drawing comparisons.

3.1.1 Dynamic Buckling of a Spherical Shell Segment – This problem concerns the dynamic response of a clamped, axisymmetric, elastic spherical cap under suddenly applied external pressure.

Table 1

SUMMARY OF TEST PROBLEMS

Problem No.	Description	Element Type and Material Properties	Purpose
1	Cantilever beam with suddenly applied lateral tip load	Two-dimensional beam element; elastic properties	Verify small deflection, dynamic solutions
2	Slightly cambered simply supported column with sudden axial end shortening	Two-dimensional beam element; elastic properties	Verify large deflection, dynamic solutions
3	Dynamic loading on circular arch	Two-dimensional beam element; elastic properties	Verify dynamic, snap-through solution
4	Impulsive transverse load on axially restrained beam	Two-dimensional beam elements; elastic-plastic properties	Verify elastic-plastic formulation and interaction with axial constraints
5	Plane stress wave propagation	Membrane triangles; elastic and elastic-plastic properties	Verify membrane formulation for in-plane effects
6	Circular membrane with transverse impulsive load	Membrane triangles; elastic and elastic-plastic properties	Verify membrane formulation for out-of-plane effects
7	Elastic, cantilever plate with suddenly applied lateral tip load	Three-dimensional plate bending elements; elastic properties	Verify plate formulation for small deflections
8	Dynamic loading on circular arch	Three-dimensional plate bending elements; elastic properties	Verify plate formulation for plane snap-through problem
9	Elastic cantilever plate	Membrane-hinge line elements, elastic properties	Verify membrane-hinge line formulation

Table 1 (Concl)

Problem No.	Description	Element Type and Material Properties	Purpose
10	Plane-stress wave propagation	Three-dimensional plate bending elements; elastic properties	Verify membrane damping terms
11	Dynamic loading on circular dome	Three-dimensional plate bending elements; elastic properties	Verify plate formulation for three-dimensional snap-through problem
12	Hood structure subjected to end loadings	Three-dimensional plate; elastic-plastic properties	First "representative" vehicle structure and loading
13	Impulsive load on cylindrical panel	Three-dimensional plate; elastic-plastic properties	Verify plate formulation for three-dimensional elastic-plastic response
14	End-on frame crash simulation	Three-dimensional beam elements; elastic-plastic properties	Simulation of actual stub frame test
15	End-on frame and sheet metal crash simulation	Beams and plates; elastic-plastic properties	Simulation of actual frame and sheet metal test

As indicated in Figure 6, the cap consists of a segment of a sphere of 60 in. radius, with a shallow rise of 0.30 in. and a thickness of 0.08 in. A plan view of the finite element mesh employed in the solution is shown in Figure 7. The mesh consists of 144 plate elements and 82 nodes and covers the complete dome with no constraints or geometric features introduced to enforce an axisymmetric response pattern. This problem has also been treated by Archer and Lange in Ref. 30 by means of a numerical integration of the finite difference equations derived from Marguerre's equations for the axisymmetric response of spherical shells. Displacement time histories for the vertical motion of the center of the cap under an inward pressure of 44 psi are shown in Figure 6 for the present results and for Ref. 30. It is evident that there is fairly close agreement between the two results with the finite element response being slightly delayed (i.e., stiffer) than the results of Ref. 30. It is also apparent that the response lies in the snap-through range since the maximum displacement approaches twice the undeformed rise of the shell. This is also clearly shown in the successive deformed shapes of the shell shown in Figure 8.

3.1.2 Dynamic Response of a Cylindrical Panel – The purpose of this test problem is to compare the results of the present analysis against previously obtained results involving dynamic, elastic-plastic shell buckling response. As indicated in Figure 9, the problem consists of a 120 deg segment of an aluminum cylindrical shell, supported and clamped at all four edges and loaded explosively over a portion of its surface, as shown. Previous experimental and analytical results are available for this problem in the literature, particularly in Refs. 31 and 32.

The mesh geometry employed is illustrated in Figure 10 and consists of 64 elements and 45 nodes representing one-half of the symmetrical shell. The displacement-time history of node 5 is shown in Figure 11 together with corresponding results from Refs. 31 and 32. The explosive load according to Refs. 31 and 32 corresponds to an initial (i.e., impulsive) velocity of 5650 in./sec directed radially inward over the loaded portion of the shell.

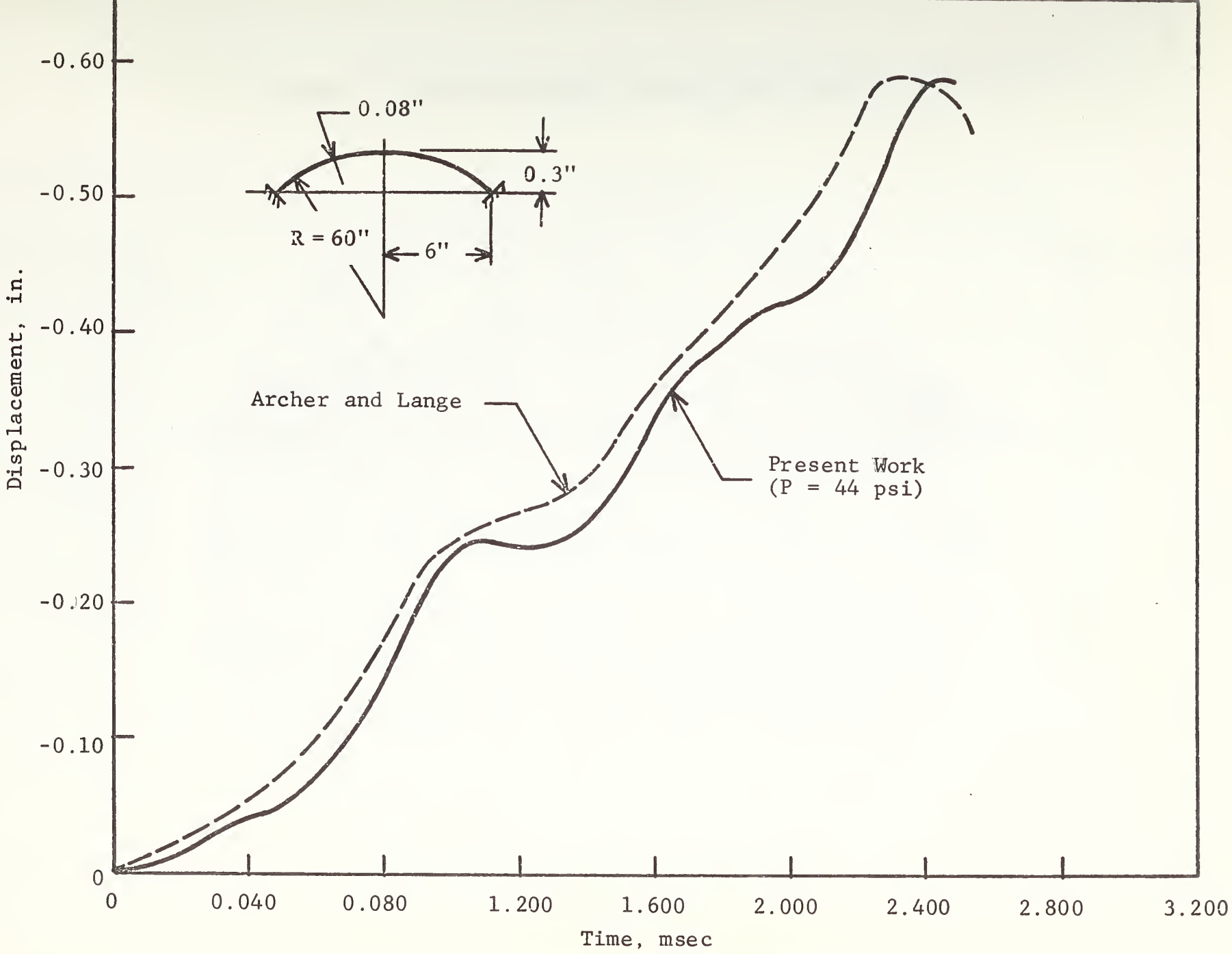


Figure 6 Displacement-Time History of a Clamped Spherical Shell

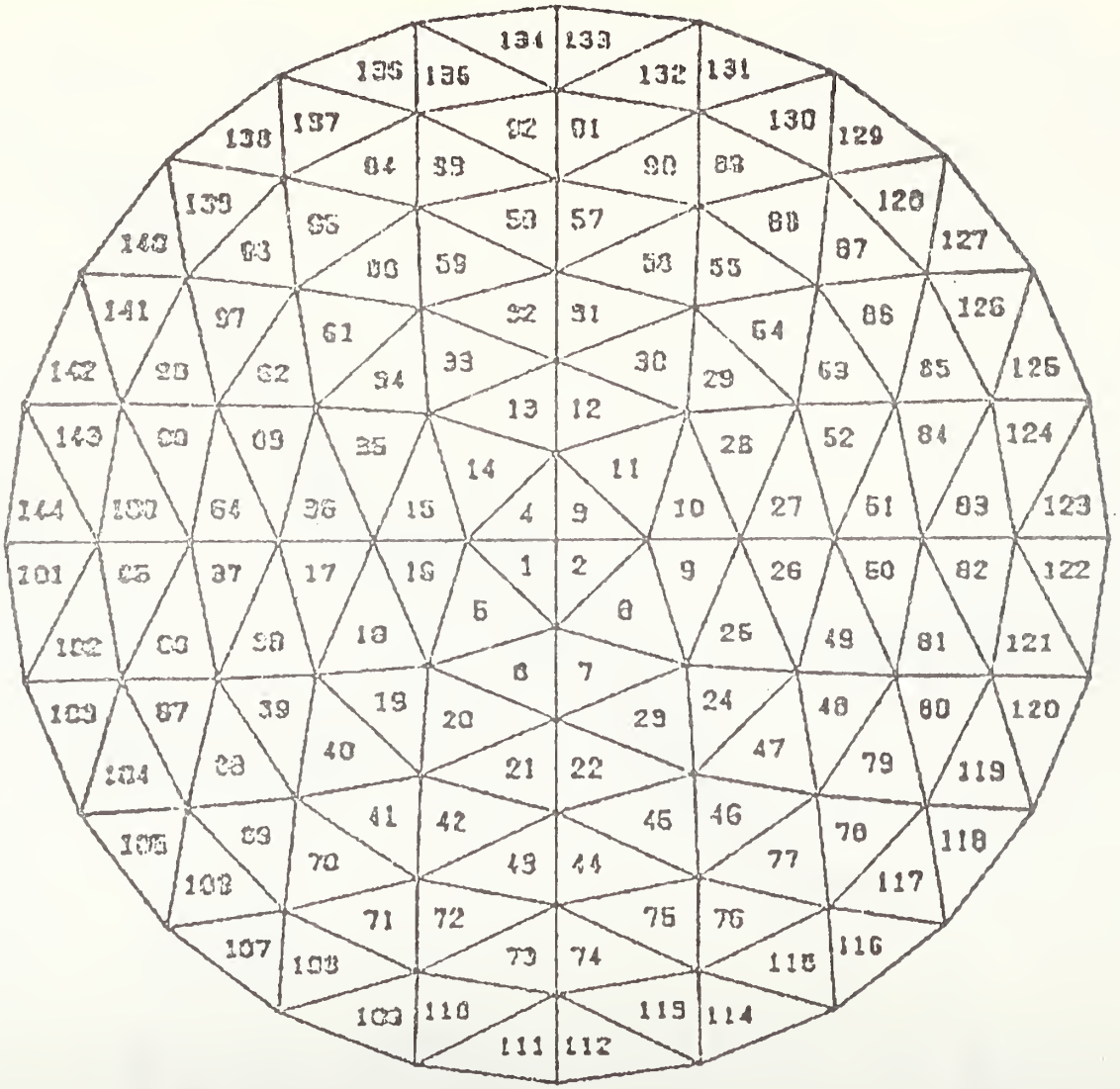


Figure 7 Spherical Shell Finite Element Mesh

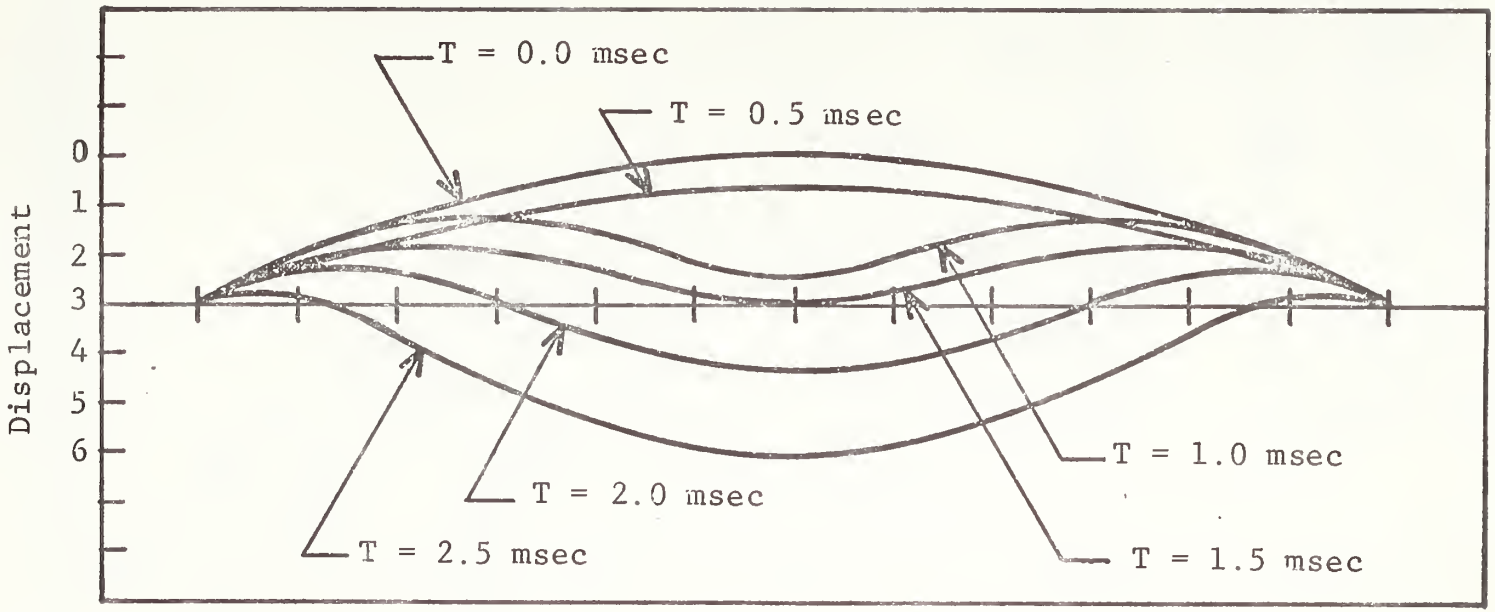
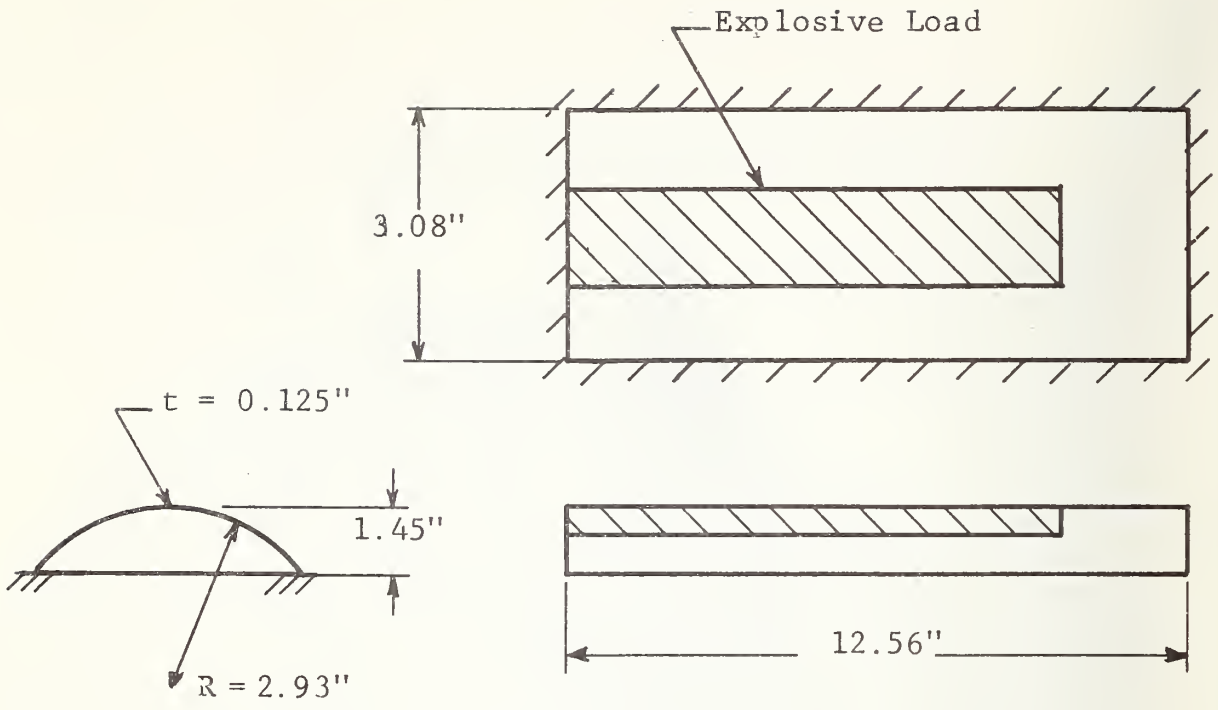
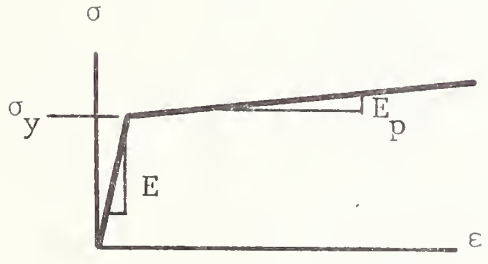


Figure 8 Progressive Deformation Stages of a Clamped Spherical Shell



a. Geometry



b. Properties

AL 6061-T6

$E = 10.5 \times 10^6$ psi; $\nu = 0.333$

$\rho = 2.54 \times 10^{-4}$ lb_f - sec^2/in

$\sigma_y = 44{,}000$ psi; $E_p = 0$

Figure 9 Cylindrical Shell - Geometry and Properties

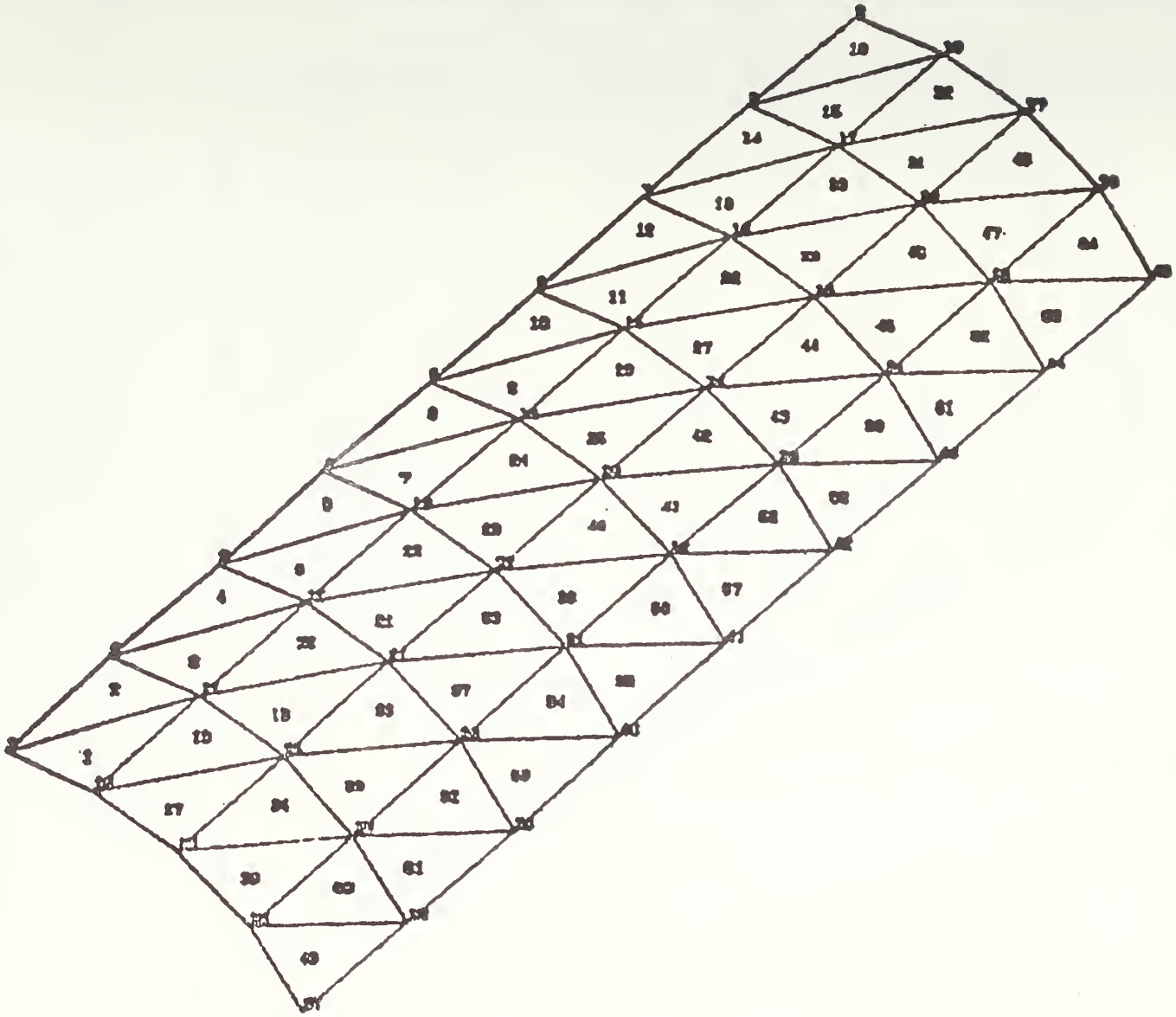


Figure 10 Cylindrical Shell Segment - Undeformed

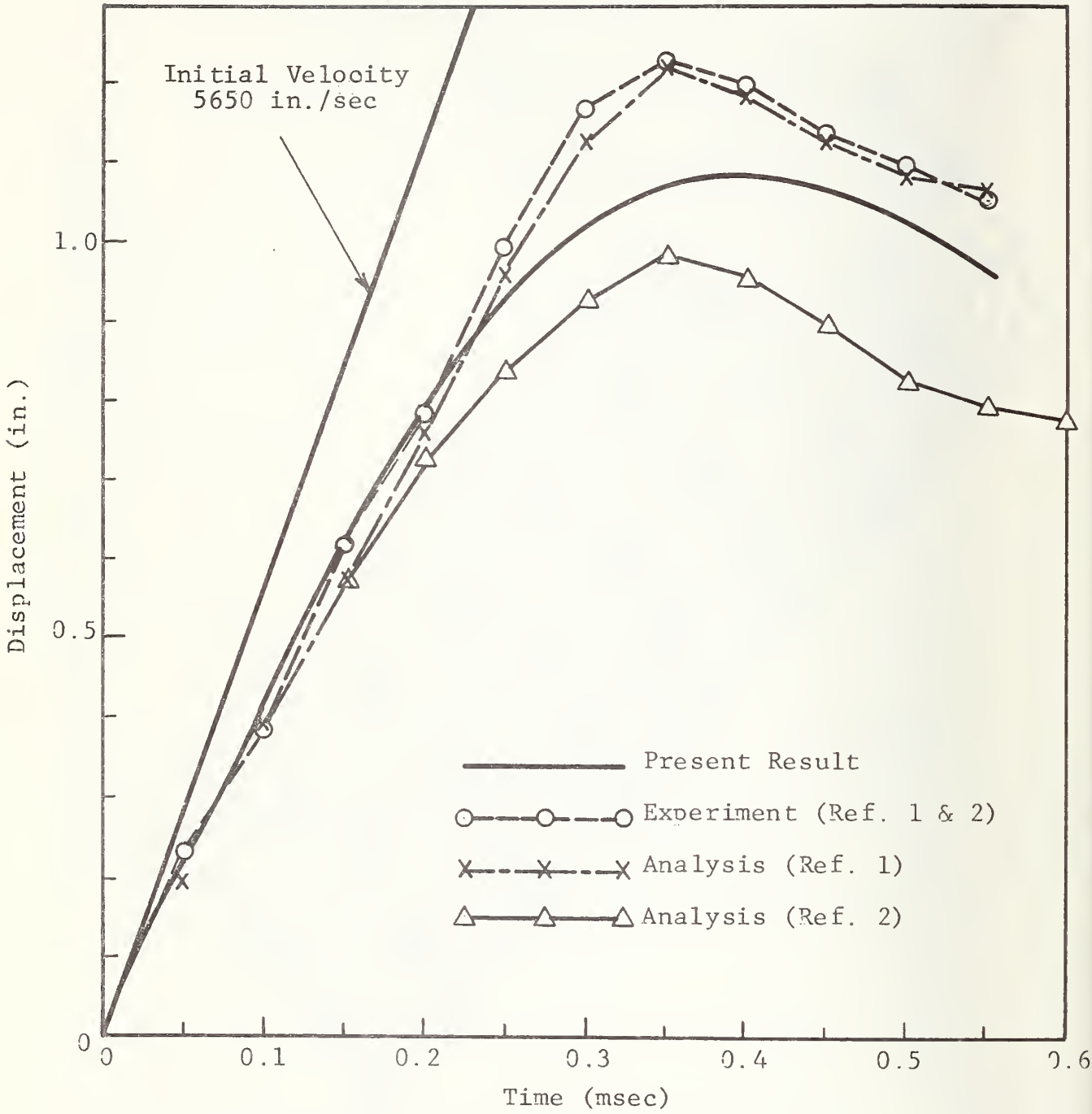


Figure 11 Cylindrical Shell Segment Crown Displacement versus Time (Node 6)

This corresponds to the constant velocity line in Figure 11 which is tangent to the displacement curve at t = 0. The present results are bracketed by those of Refs. 31 and 32 with the improved analysis results of Ref. 32 lying below and the experimental results above. The older analysis results from Ref. 31 are very close to the experimental curve but subsequent improvements in that procedure (Ref. 32) result in lower displacements. Improved correlations were obtained in Ref. 32 by introducing yielding boundaries and this was not attempted in the present work.

The first part of the velocity history for node 5 is shown in Figure 12. The velocity decreases from its initial value of 5650 in/sec, increases again as the shell rebounds, and then decreases fairly smoothly to zero at approximately 0.4 msec. The corresponding displacement pattern at 0.4 msec is shown in Figure 13. While further efforts could be made to improve the correlations shown in Figure 11, principally in the boundary conditions and in a refined mesh, the present results are considered satisfactory.

3.2 Barrier Test Simulations

The test problems presented throughout the progress reports and in the preceding section are an essential means of assuring that the analysis formulation and computer program are capable of describing the types of phenomena for which they are intended. However, the transition from this class of problems, which are fairly small and idealized, to the analysis of actual vehicle structures in realistic crash events is by no means a trivial step. In the latter case the analysis must deal with much longer duration events and structures of substantially greater size and geometric complexity with realistic, rather than idealized, boundary conditions and connection details. In fact, the proper and efficient application of an analysis procedure to real structures and events is nearly as much a research undertaking, at least initially, as is the original formulation and coding of the analysis.

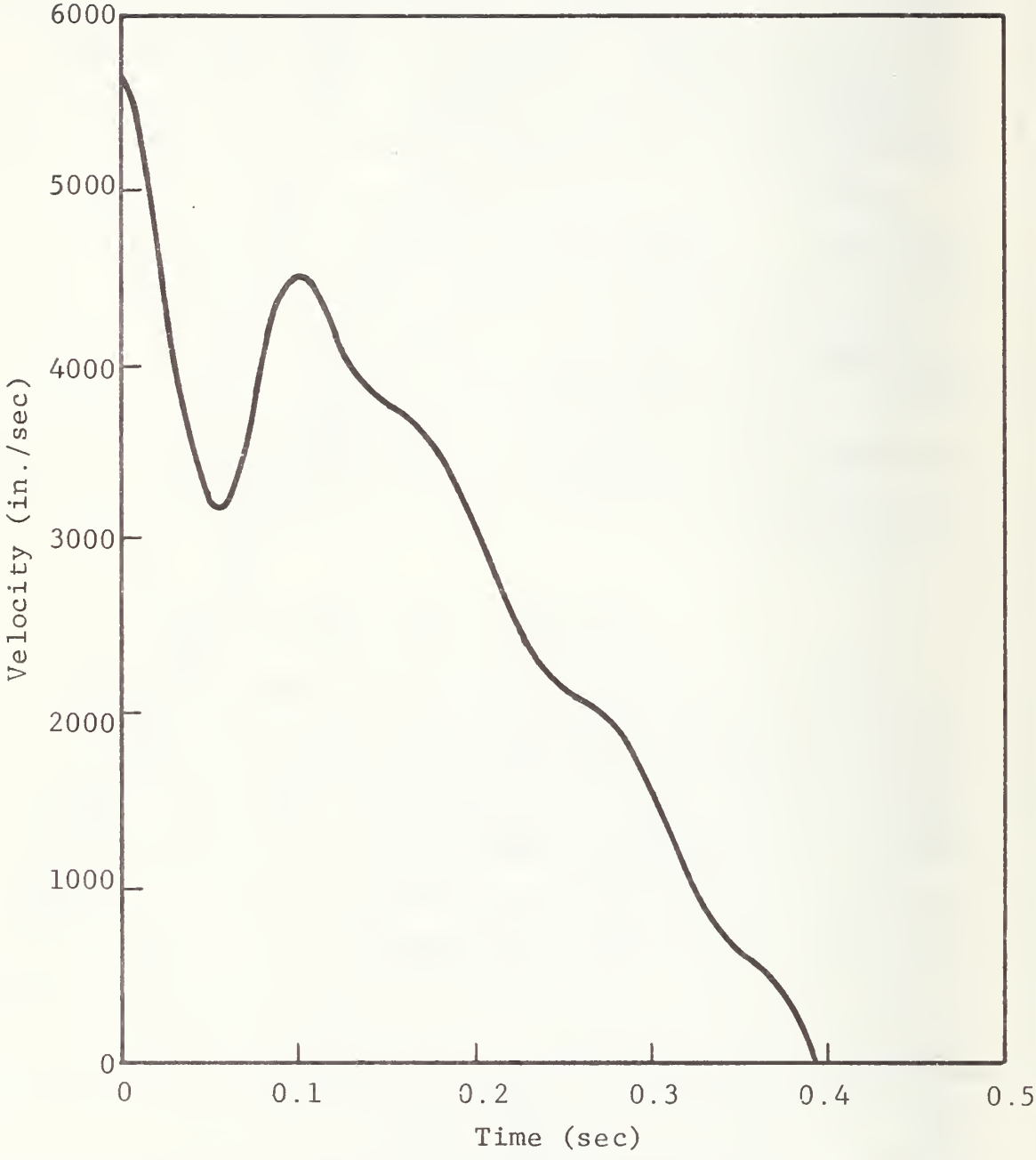


Figure 12 Cylindrical Shell Segment Crown Velocity versus Time

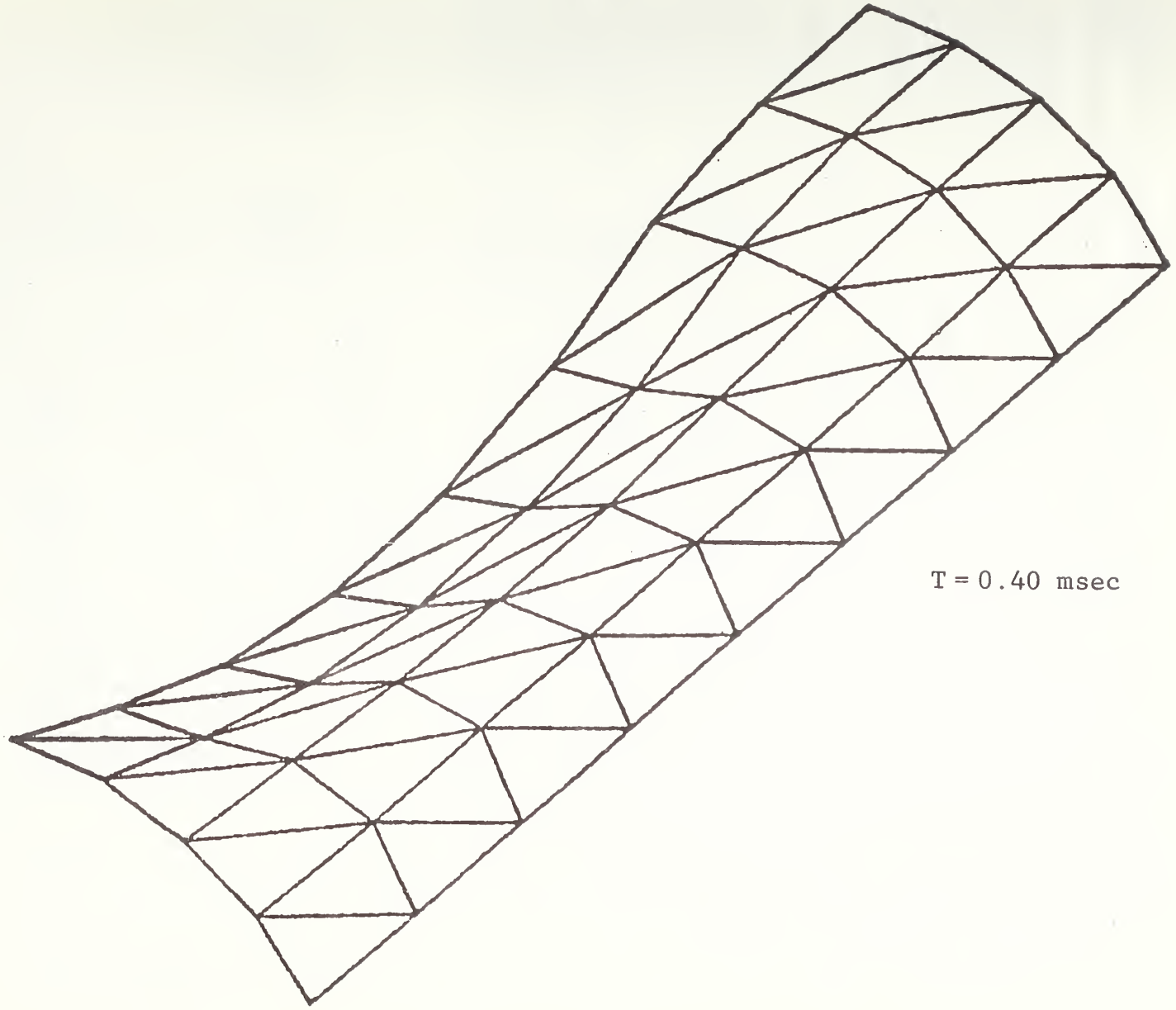


Figure 13 Cylindrical Shell Segment - Deformed

Also, the planning and execution of such analyses has many aspects in common with the planning and execution of full scale crash tests, not the least of which is the acquisition and handling of the large quantities of data which are inherent in either activity.

This section describes the results of our efforts to obtain relatively large scale simulations of actual vehicle crash events. The events which are the objects of these simulations were selected from a series of barrier tests conducted by Dynamic Science (Ref. 33). The series of tests of interest involved frontal impacts at various velocities of a 1968 Plymouth Fury having different portions of the front end structure and machinery removed in different tests as summarized in Table 2. The object of these tests was to isolate and identify the dynamic resistance properties of the various front end components in three groups, namely: frame structure, sheet metal structure and the engine and machinery components. Simulations were developed for two test configurations involving the stub frame alone (test 283-51) and the stub frame and sheet metal together (test 283-53) since these configurations did not involve the added complications associated with the engine components. The test involving sheet metal alone was excluded due to the extreme and unrealistic deformation patterns which appeared in the test results.

Table 2
PLYMOUTH FURY TEST SERIES

Test No.	Configuration	Total Weight (lbs)	Impact Velocity (mph)
283-49	Sheet metal	2665	19.85
283-50	Engine	3914	19.48
283-51	Stub frame	2909	29.64
283-52	Full vehicle	4192	51.45
283-53	Stub frame and sheet metal	2997	43.99

The finite element models developed for these two test configurations represent one-half of the structural components forward of the vehicle fire wall and are based on an assumption of symmetry for both the vehicle and its response modes about the center plane. The frame model consists entirely of beam elements and the frame and sheet metal model contains the frame model, plate elements representing the sheet metal, and additional beam elements representing stiffener sections in the hood. In both cases the appropriate inertial properties of the remainder of the vehicle are provided as added masses at the vehicle center of gravity and are attached to the frontal structure by means of rigid links. Data relating to the tests and the vehicle geometry were obtained from several sources including the following:

- Reports and high-speed motion pictures associated with the test program (obtained through DOT)
- Sheet metal profiles, thicknesses, and material properties obtained from Stanford Research Institute
- Additional measurements and observations of a 1968 Plymouth Fury by IITRI staff.

3.2.1 Stub Frame Impact (Test 283-51) – Plan and side views of the finite element mesh for the stub frame are shown by the dashed (i.e., undeformed) lines in Figures 14 and 15. The model consisted of 22 nodes and 21 beam elements plus a rigid link connecting the rear of the frame at the firewall attachment point to the center of gravity of the vehicle. Additional mass was lumped at the approximate location of the tire connection and also at the vehicle center of gravity in order to introduce the proper inertial characteristics into the analysis. Particular emphasis was placed in accurately modeling the various open and closed wall beam sections that comprise the stub frame. Six different beam cross sections were employed in the model as summarized in Table 3. In all cases the material properties were taken as those of mild steel with an elastic modulus of 30×10^6 psi, a Poisson's ratio of 0.3 and a yield stress of 36,000 psi. Boundary conditions consisted of fixing the fore and aft translatory motion of the nodes in immediate contact with the barrier.

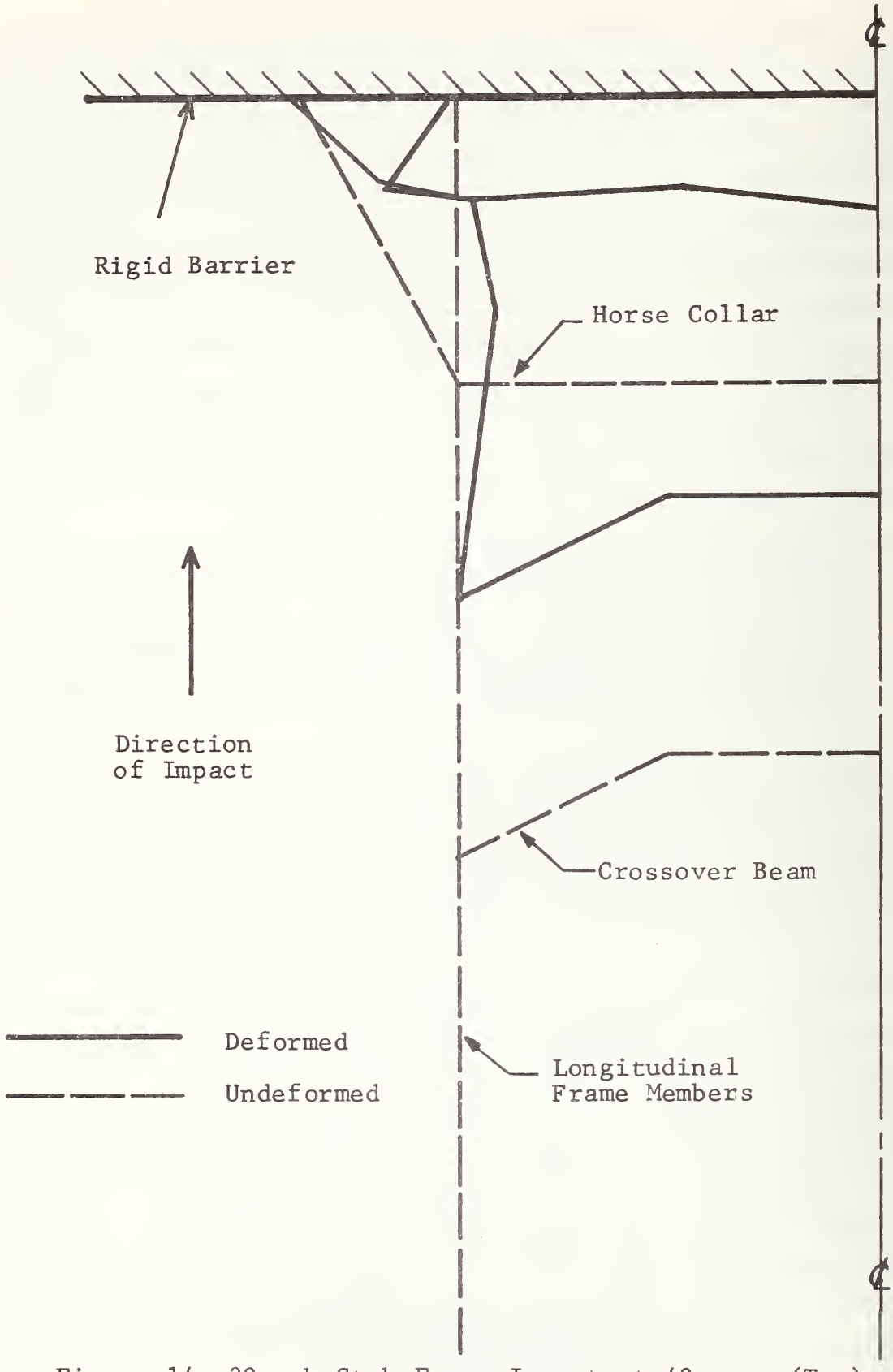


Figure 14 30 mph Stub Frame Impact at 40 msec (Top)

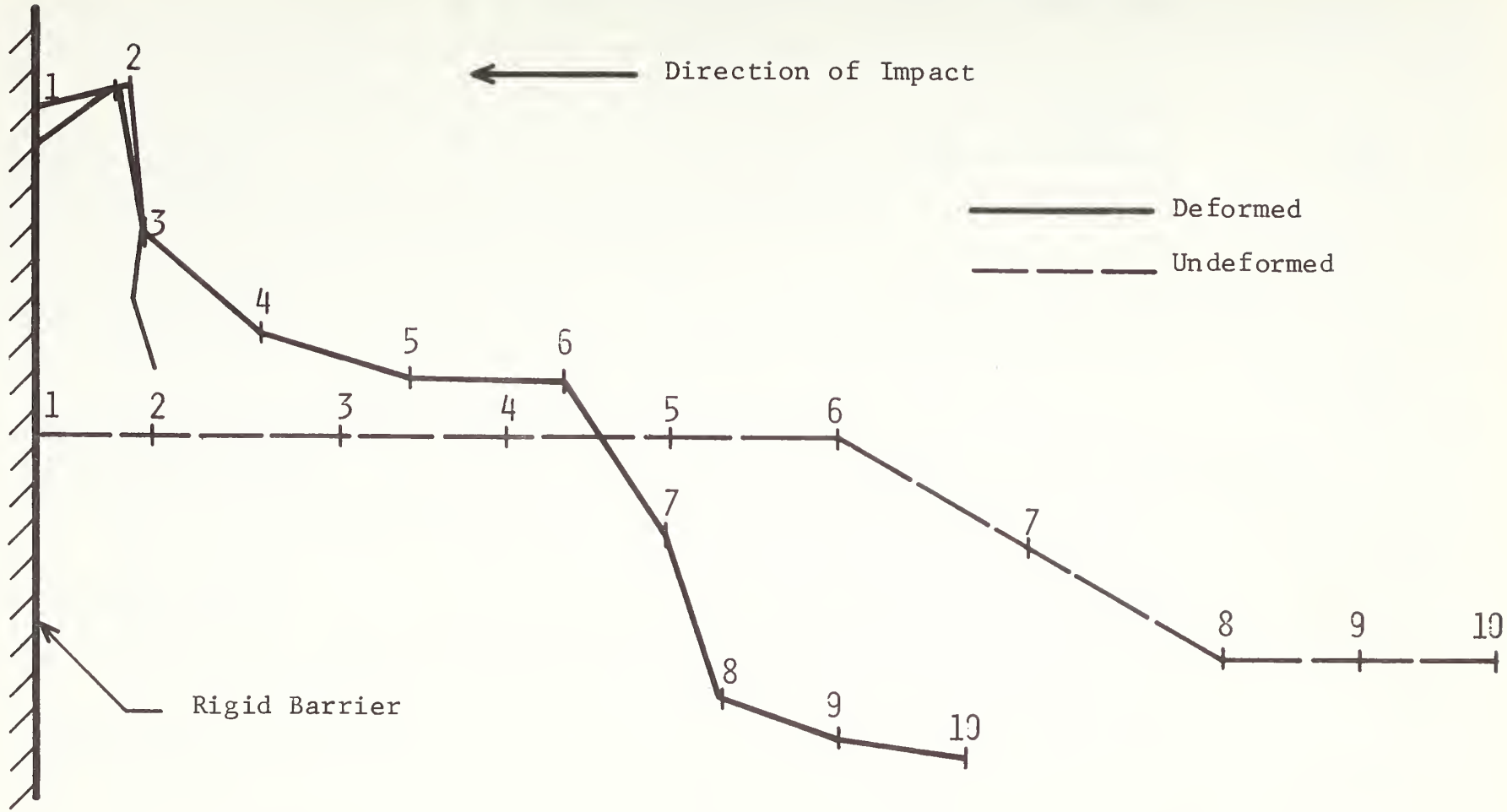
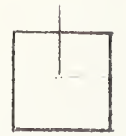
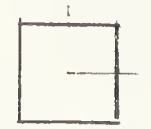


Figure 15 30 mph Stub Frame Impact at 40 msec (Side)

Table 3

STUB FRAME BEAM PROPERTIES

Member Location	Number of Elements	Thickness (in.)	Dimensions (in.)	Shape
Longitudinal Frame (Rear of Crossover Beam)	4	0.085	3 x 3 box	
Crossover Beam	4	0.080	3-1/2 x 3-1/2 box	
Longitudinal Frame (Crossover Beam to Bumper Support)	3	0.085	2 x 6 channel	
Horse Collar Beam	2	0.060	4 x 3 Z-section	
Bumper Support	4	0.060	1-1/2 x 4 channel	
Bumper	4	0.085	2 x 9 bumper	

Rebounding of the frame from the rigid barrier was not taken into account, therefore, once a node came in contact with the barrier, it remained in contact. It should be noted that at the beginning of the solution phase of the analysis it was assumed that all nodes along the bumper were in contact with the barrier. The structure was driven with an initial velocity of 30 mph at all nodes.

Top and side views of the deformed frame at 40 msec after impact are also shown as the solid lines in Figures 14 and 15. It should be noted that the deformed structures are plotted to the same scales as the undeformed mesh and no magnification factors have been introduced. In the plan view, Figure 14. all members show substantial forward motion with bowing in the forward longitudinal frame section and collapse of the bumper supports. Similar behavior is seen in the test films including bowing of the frame members and severe distortions in the bumper supports. In the films the triangle formed by the bumper supports also pivots away from the wall after collapse but this was prevented by the boundary constraints used in the analysis. In the side view, Figure 15, the frame bends throughout the forward portion containing the bumper, bumper supports, and forward frame are seen to have climbed the barrier a distance of approximately 15 in. This same pattern, and particularly the upward motion of the front part of the frame, is clearly evident in the test films although in the films the collapse of the bumper supports is somewhat more severe and the end of the forward frame section comes in contact with the wall. At this point in the motion the frame has begun to rebound from its maximum bent position and the center of mass begins to pivot downward about the point of contact with the wall. At this point the film sequences indicate that this motion is reacted by the rear wheel suspension systems but in the analysis this pivoting continues since the rear wheels were not incorporated in the model, nor were any vertical constraints applied to the center of gravity.

Longitudinal velocity and acceleration at the vehicle center of gravity for 40 msec of motion from the analysis and test results are shown in Figures 16 and 17. As seen in the velocity-time histories, the vehicle has lost only about 20 percent of its initial impact velocity (about 36 percent of its initial kinetic energy). In this test however, this time period contains the maximum vehicle decelerations and barrier loads. A time shift of approximately 10 msec between the analysis and test results is clearly evident in the velocity histories which may be the result of filtering in the test data or differences in zero time between the test and analysis procedures. Whatever the cause, this time shift is found in comparing all corresponding results and will not be commented on further in subsequent discussion. The acceleration time record obtained from the analysis (Figure 17) shows an early deceleration peak of about 10 g in the first 3 msec (note that a signal transit from bumper to center of gravity takes approximately 0.25 msec) and a series of sharp peaks on the order of 30 g between 15 to 27 msec. Peaks in the data are about 22 g at 12 msec and 40 g at 30 msec with a substantial rebound peak at 20 msec. In this case the test data clearly show the effects of filtering. The early peak in the analysis clearly is related to the initial resistance and collapse of the bumper supports while the subsequent peaks are probably attributable to collapse and overall bending of the frame. The analysis also routinely provides contact forces at the barrier but in this case these data were lost beyond the first 5 msec of response.

As indicated in Figure 18 however the early value of the total barrier force was approximately 23,000 lbs (for half of the vehicle) as compared to a value of about 45,000 lbs in the corresponding data (for the full vehicle). This apparent agreement may not be precise however when filtering in the data is taken into account.

3.2.2 Stub Frame and Sheet Metal Impact (Test 283-53) —An isometric view of the finite element mesh for the stub frame and sheet metal model before impact is shown in Figure 19.

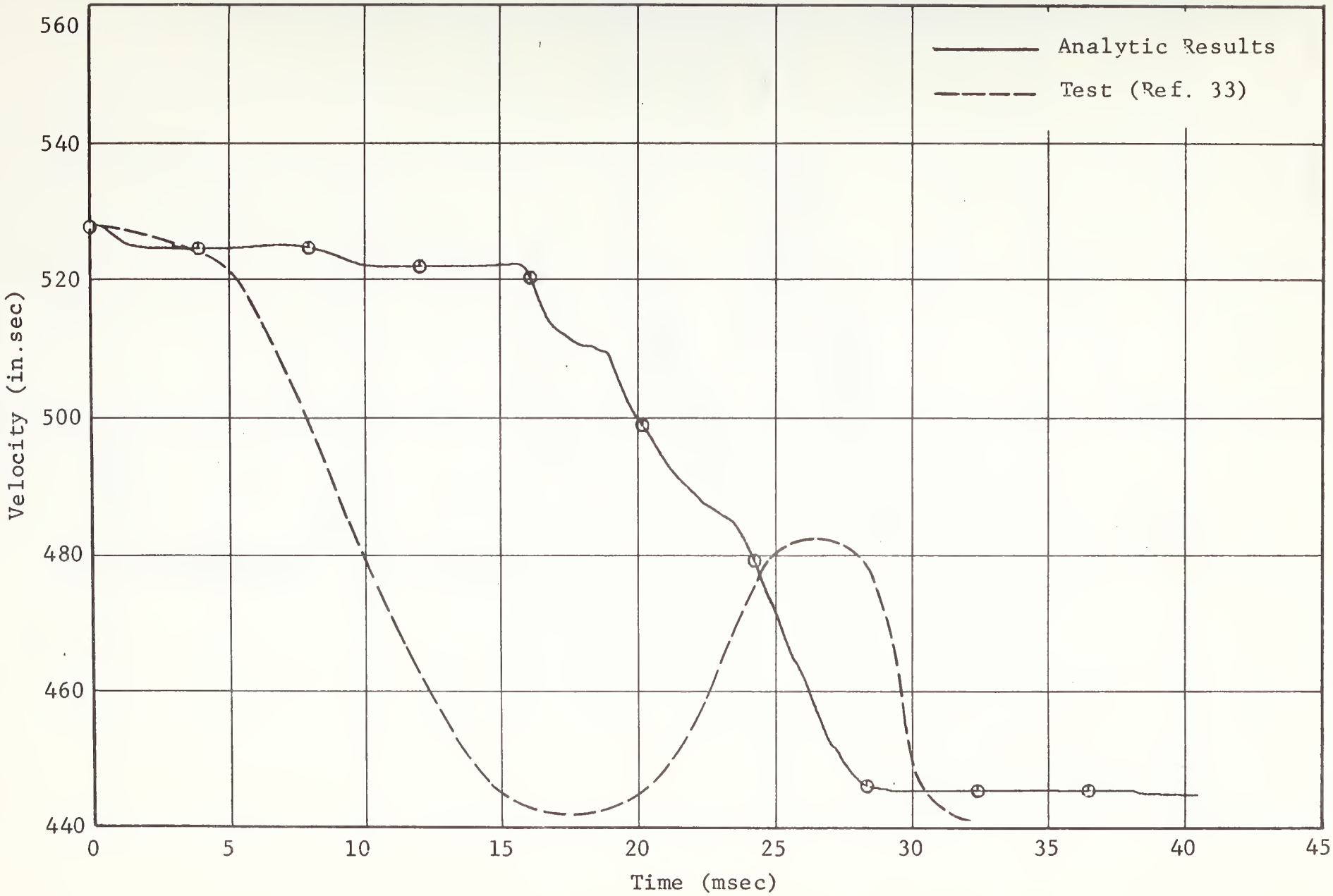


Figure 16 Vehicle Center of Gravity Velocity for 30 mph Stub Frame Test

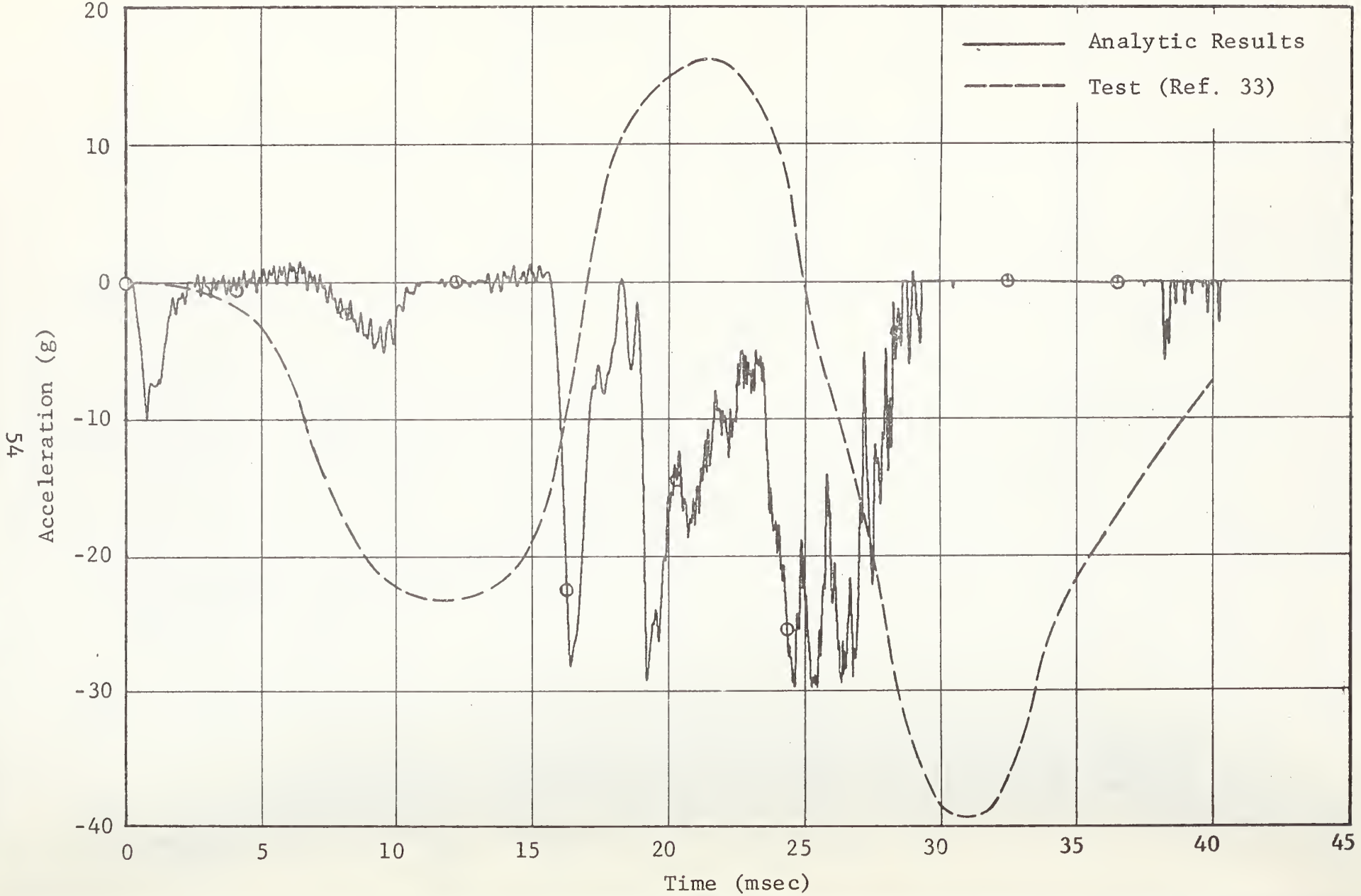


Figure 17 Vehicle Center of Gravity Acceleration for 30 mph Stub Frame Test

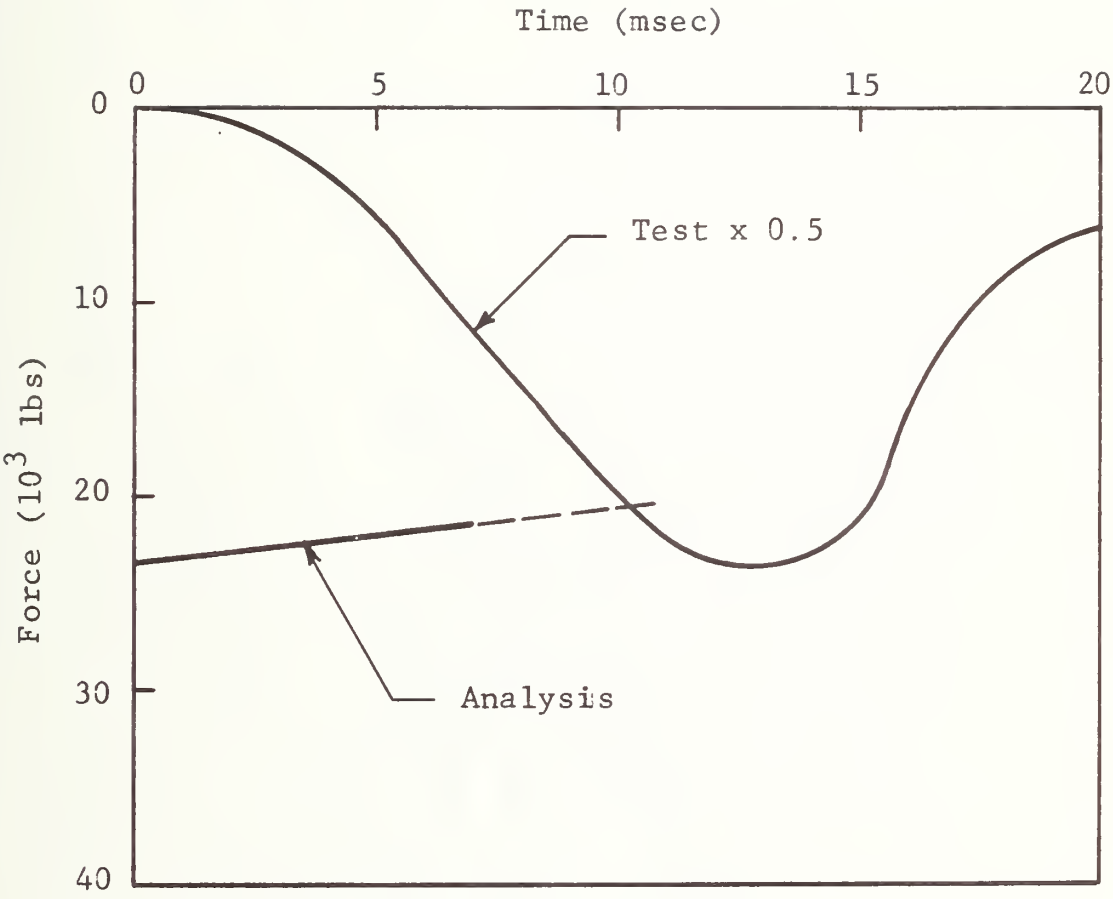


Figure 18 Stub Frame Test - Barrier Force

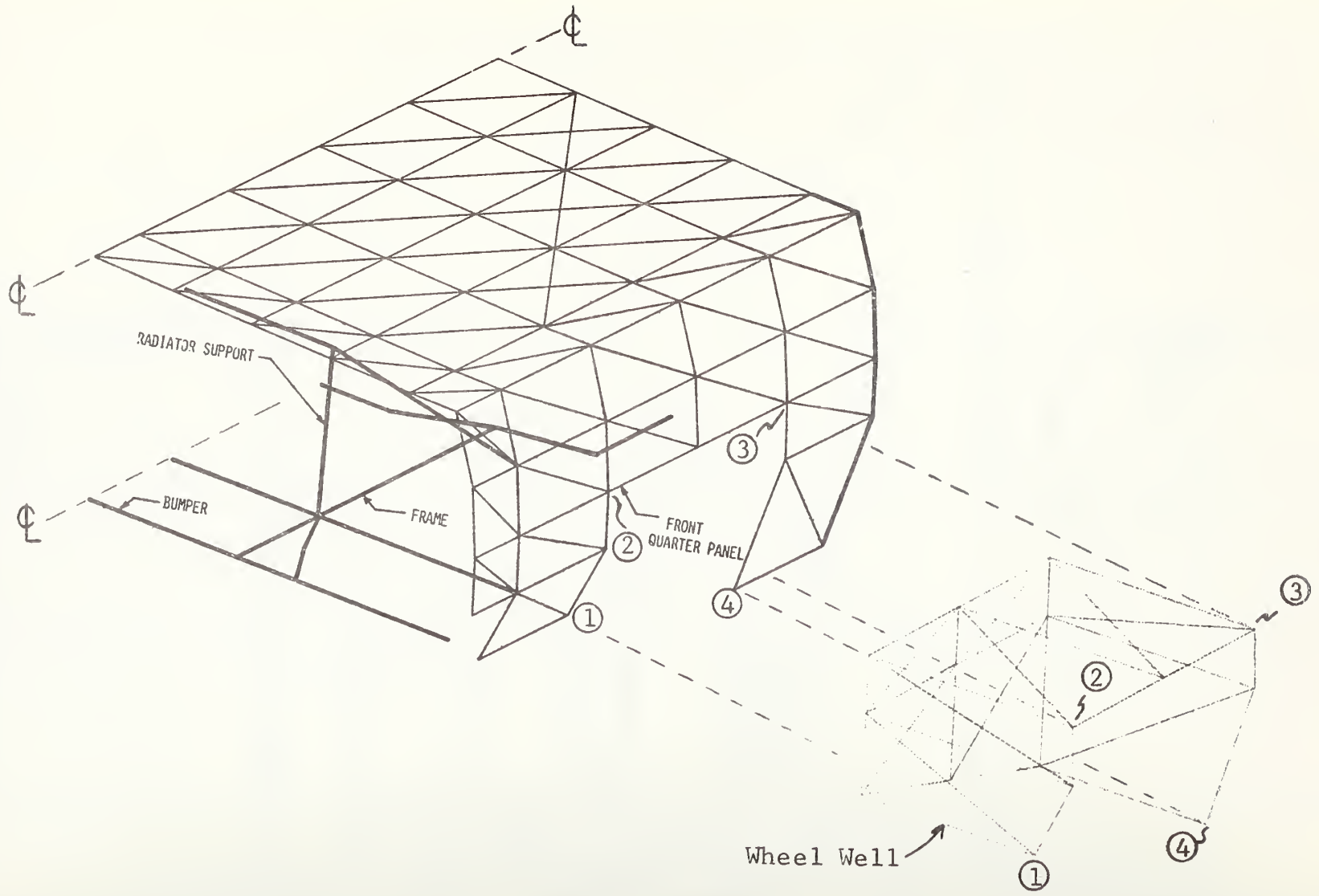


Figure 19 Stub Frame and Sheet Metal Model (Undeformed)

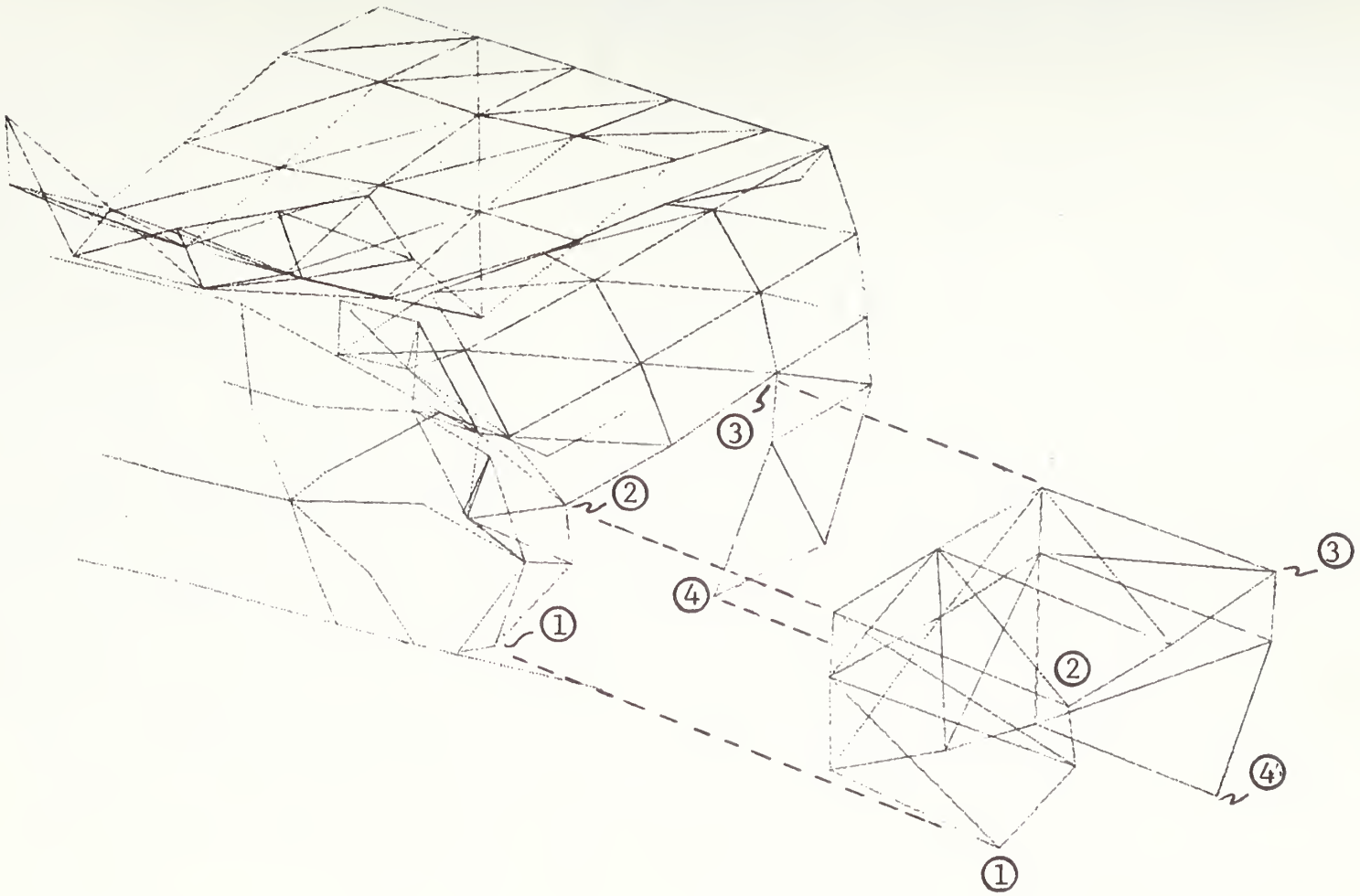


Figure 20 Stub Frame and Sheet Metal Model (Deformed, 20 msec)

This model comprises the left front quarter of the 1968 Plymouth employed in the test. The major components of the model are the hood, left front fender, wheel well and frame and radiator support. The wheel well is shown separately for the sake of clarity in the figure but is actually an integral part of the model and is attached along the lines defined by the match points 1 through 4 indicated in the figure. The thickness of all sheet metal elements was taken as 0.039 in. material. The frame portion of the model is identical to that employed in the stub frame simulation with the addition of the radiator support structure indicated. The hood portion of the model is free to move relative to the fender at their line of intersection, and, in addition to the plate elements shown, contains 2.0 x 1.0 x 0.05 in. hat section stiffeners at reinforcements. As with the stub frame model additional mass was lumped at the approximate location of the wheel suspension and also at the vehicle center of gravity. In this case all points along the rear edge of the fender and hood as well as the end of the frame were attached to the vehicle center of gravity by means of rigid links. The model was driven by an initial longitudinal velocity of 774 in./sec for all nodes with longitudinal constraints applied at nodes in the forward section of the model successively as they came into contact with the barrier.

The deformed shape of the model at 20 msec after impact is shown in Figure 20. At this point the forward motion of the center of gravity is approximately 15 in. and the forward portions of the frame assembly and the first two lines of nodes in the hood and fender are in contact with the barrier. Severe distortions are indicated in the forward portions of the model with the bumper and bumper brackets collapsed downward and the hood and fender collapsing upward and outward respectively. Separation at the hood and fender junction line is clearly evident and the rear portion of the hood has begun to buckle upward away from the fender. The wheel well is not greatly involved although the forward wall has bent to conform to the distortions in the forward section of the fender.

A comparison of longitudinal acceleration time histories at the vehicle center of gravity is given in Figure 21 for 40 msec of motion. The analysis result is similar to that obtained from the analysis of the stub frame configuration in that an early acceleration pulse of approximately 10 g associated with collapse of the bumper support system is apparent. Further the maximum acceleration levels during this phase of the response is approximately 30 g as in the results for the stub frame. In this case however the 30 g response is clearly associated with the impact of fender and hood sheet metal rather than with substantial frame bending. The corresponding test results indicate a peak of about 25 g at a somewhat later time and also clearly show the effects of filtering in comparison to the analysis data.

A similar comparison of total barrier force is provided in Figure 22 in which 50 percent of the test total is compared with the current result obtained from a model of half of the vehicle. The analysis results show an early force level of approximately 40,000 lbs associated with the collapse of the bumper support and frame system which gradually decays for the remainder of the response. The impact of the forward edges of the hood and fender appear as a sudden rise in barrier force at about 6 msec to a level of approximately 80,000 lbs which then decays over the following 10 to 15 msec. This peak is associated with only the first line of nodes in the hood and fender since forces from the subsequent impact of succeeding sheet metal nodes were lost due to a data processing error. In fact, the first three lines of nodes in the fender and hood were involved with the barrier and are included in the results given in Figure 21. In Figure 22, the impact of succeeding nodes would appear as subsequent peaks in the force time history which would appear about the time that the peak due to the previous nodes had decayed substantially. In reality, the progressive contact of the sheet metal with the barrier is a more or less continuous process and the successive impacts of the nodes would amount to a stepwise approximation of this process.

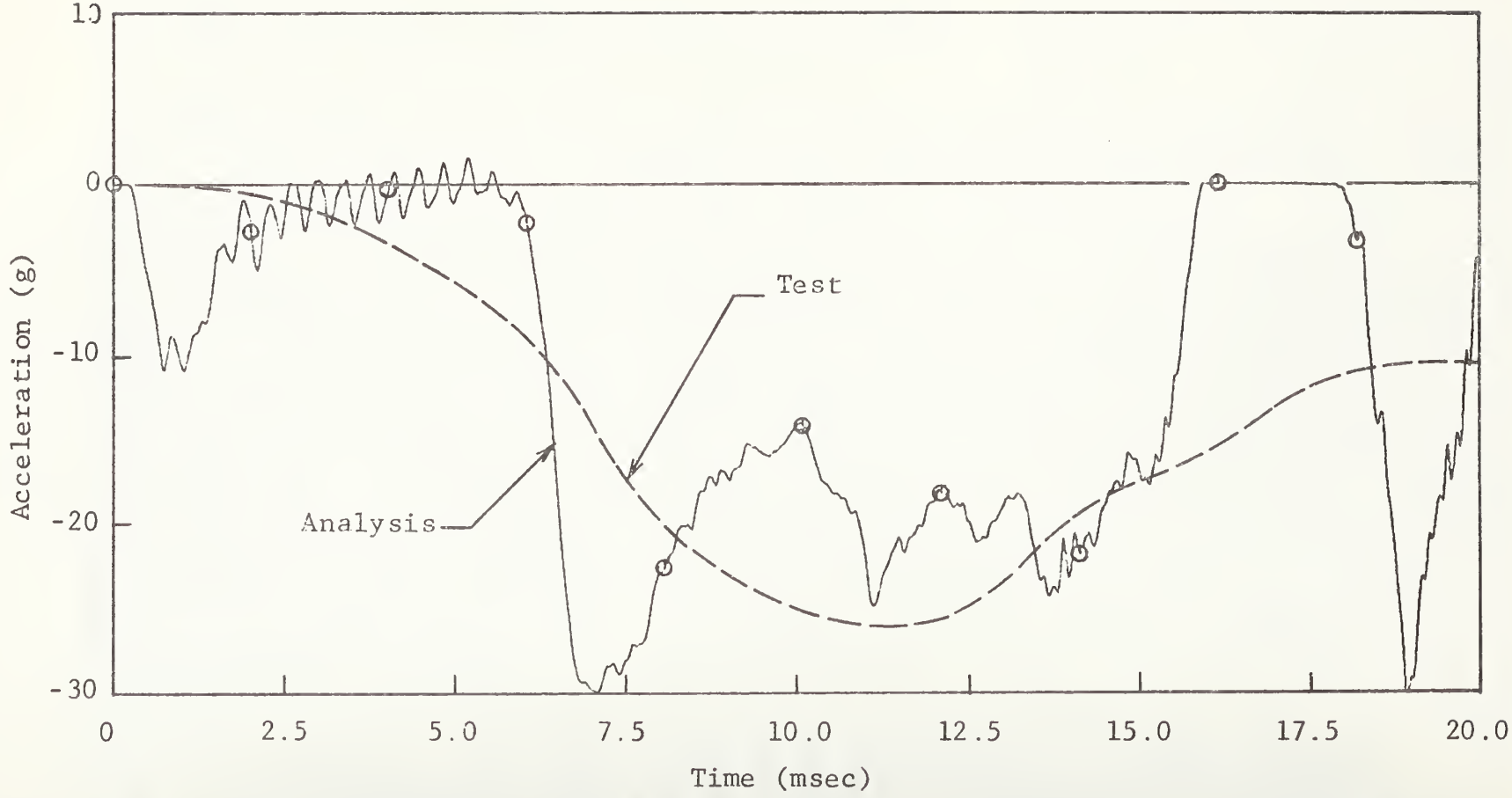


Figure 21 Stub Frame and Sheet Metal Vehicle Center of Gravity Acceleration

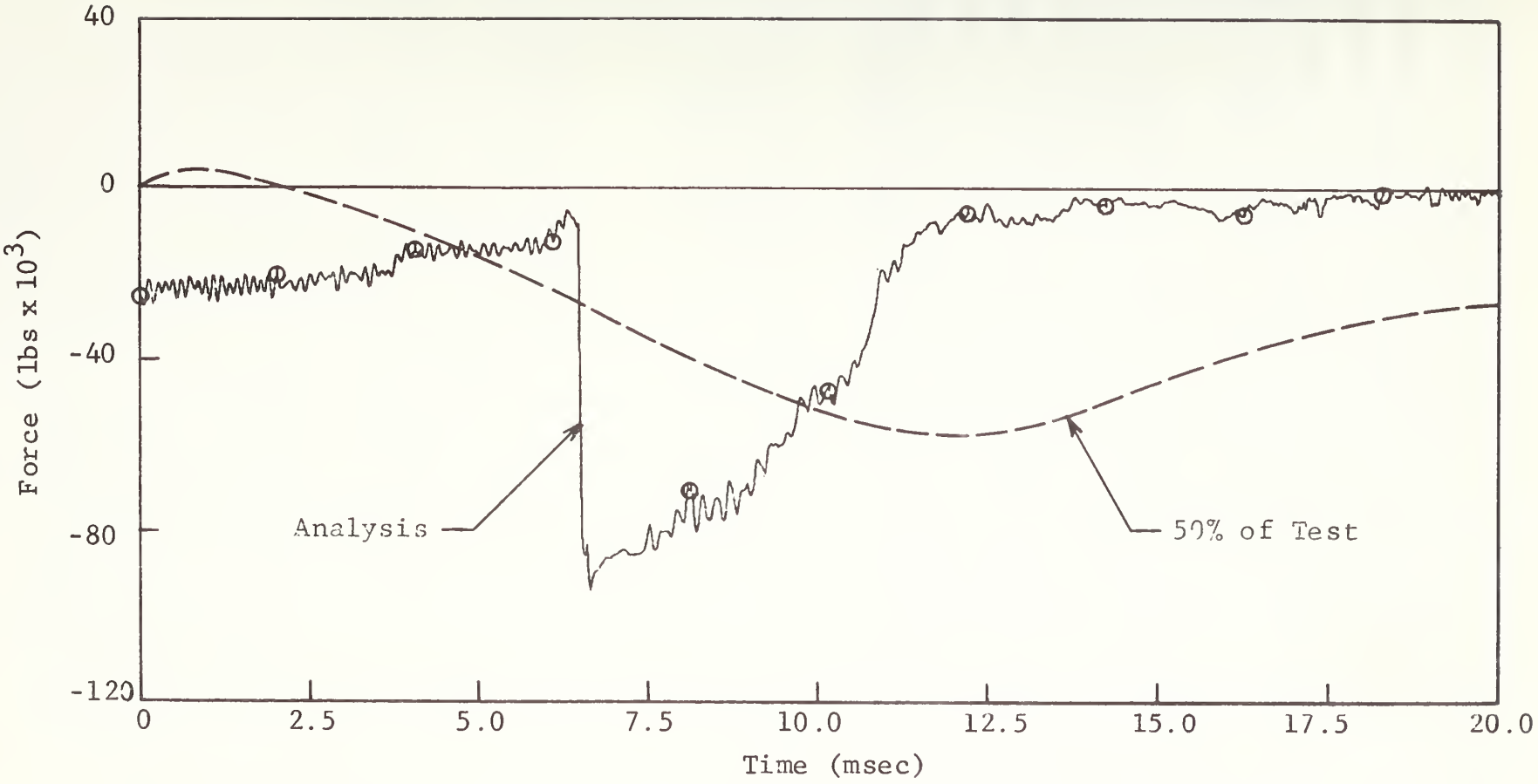


Figure 22 Stub Frame and Sheet Metal Total Barrier Force

A study of the force-time histories for individual nodes at the front of the hood and fender indicates that of the 80,000 lbs total force in the first peak, about 46,000 lbs is derived from collapse of the hood and about 34,000 lbs from the fender.

The test simulations presented here are by no means a complete or exhaustive study even for the limited series of tests which were used for comparison purposes. For a variety of reasons, the results given apply only to the initial phases of the respective crash events and improvements could also be made in the models (the provision of rear wheels or other vertical constraints at the center of gravity, for instance) and in the associated data processing. Despite these limitations, however, the results appear to be a representative portrayal of the phenomena associated with vehicle barrier collisions generally, and a reasonable correlation with the particular test data for which comparisons were attempted.

4. CONCLUSIONS AND RECOMMENDATIONS

This report presents the results of a research project undertaken by IITRI for DOT to develop a finite element computer program for use in the dynamic analysis of vehicle structure, including sheet metal, in a crash environment. This research program involved the following major tasks:

- A technique was developed for the finite element analysis of the dynamic response of plate beam structures involving very large displacements and rotations and elastic-plastic material behavior. The principal feature of this technique involves the decomposition of the element displacement field into rigid body components and deformation components thus allowing the use of a small deflection element formulation in the analysis.
- A computer program was developed incorporating this analysis procedure for beam and plate elements and rigid links together with appropriate time integration procedures and material property descriptions.
- A substantial number of test and demonstration problems were analyzed with this computer program ranging from simple classical solutions for beams and plates through large scale simulations of actual crash tests.

4.1 Conclusions

Based on the work carried out in this research program our principal conclusions are as follows:

(1) The analysis formulation, while somewhat novel, is a formally sound procedure and a proper and useful approximation technique for the dynamic analysis of beam and plate structures involving large deflections and rotations.

(2) The computer program developed is a correct rendering of this analysis technique and provided accurate results with reasonable efficiency in comparison to available known solutions.

(3) Finally, and most importantly, the computer program is readily applicable to realistic vehicle structures and provides credible results for actual crash events as demonstrated by the simulations of the end-on barrier tests.

4.2 Recommendations

In our opinion the analysis and computer program resulting from this research is a substantial improvement over previous attempts at a finite element analysis of vehicle structures and is a potentially valuable addition to the tools available for the study of vehicle crashworthiness. Although many possibilities suggest themselves in terms of further development and refinement in the analysis procedure and computer program (some of which are noted below) the most important and potentially fruitful activity lies in the immediate and continued application of the analysis to actual vehicles and crash events. Such activity will bring the analysis to real vehicle related problems more quickly and will serve as a guide to its further development and application. Thus our primary recommendation is that provision be made for the continued and extensive use of the program in connection with actual vehicle behavior. Some suggestions for such applications as well as for further analysis and program development are as follows:

Applications

- Additional simulation of controlled crash events along the lines of those attempted in this work but extended to include machinery and drive train components and side-on impacts.
- Use of the program in connection with studies of alternate vehicle designs, including potential structural modifications and configuration changes.
- Use of the program to generate structural crush data for subsequent incorporation in the more simple lumped parameter or hybrid class of programs.
- Comparative studies of the deformation and crashworthiness of existing vehicle designs.

Program Development

- Development of suitable approximate moment-curvature relations for sheet metal as a means of improving computational efficiency.

- Incorporation of plate and beam hinges at selected locations in the main version of the program.

- Development of higher order and special hybrid element formulations possibly to include tearing or wrinkling within the element.

- Development of an implicit rather than explicit integration procedure either as an alternate to the present explicit procedure or for selective use for part of the structure or portions of its response history.

APPENDIX A

TRIANGULAR PLATE DISPLACEMENT AND CURVATURE FUNCTIONS

This appendix provides detailed expressions for the transverse displacement and curvature in the triangular plate elements. In terms of triangular coordinates, Zienkiewicz (Ref. 26) gives the displacement of the plate as follows:

$$\hat{u}_z = \beta_1\zeta_1 + \beta_2\zeta_2 + \beta_3\zeta_3 + \beta_4(\zeta_1\zeta_2^2 + \tfrac{1}{2}\zeta_1\zeta_2\zeta_3) + \ldots + \beta_9(\zeta_1^2\zeta_2 + \tfrac{1}{2}\zeta_1\zeta_2\zeta_3) \tag{A1}$$

where $\zeta_1, \zeta_2, \zeta_3$ are the triangular coordinates and $\beta_1, \beta_2, \ldots, \beta_9$ are the constants depending on the boundary conditions. We define the nodal rotations as

$$\hat{\theta}_x = \frac{\partial \hat{u}_z}{\partial \hat{y}} \; ; \qquad \hat{\theta}_y = - \frac{\partial \hat{u}_z}{\partial \hat{x}} \tag{A2}$$

Since the displacements are zero at the nodes, the unknowns of equation (A1) can be determined in terms of nodal rotations, and equation (A1) can be expressed as

$$\{\hat{u}_z\} = [N_i]\{\Theta\} \tag{A3}$$

where

$$\{\Theta\}^T = (\hat{\theta}_{1x} \; \hat{\theta}_{1y} \; \hat{\theta}_{2x} \; \hat{\theta}_{2y} \; \hat{\theta}_{3x} \; \hat{\theta}_{3y})$$

and $N_i, (i=1,\ldots,6)$ are the shape functions given by Meek (Ref. 27). Once the deformation is assumed, the slopes follow

$$\begin{Bmatrix} \dfrac{\partial \hat{u}_z}{\partial \hat{x}} \\ \dfrac{\partial \hat{u}_z}{\partial \hat{y}} \end{Bmatrix} = \begin{bmatrix} \dfrac{\partial N_i}{\partial \hat{x}} \\ \dfrac{\partial N_i}{\partial \hat{y}} \end{bmatrix} \{\Theta\} \tag{A4}$$

and finally the curvatures are given as

$$\begin{Bmatrix} \kappa_{xx} \\ \kappa_{yy} \\ \kappa_{xy} \end{Bmatrix} = \begin{bmatrix} \dfrac{\partial^2 \hat{u}_z}{\partial \hat{x}^2} \\ \dfrac{\partial^2 \hat{u}_z}{\partial \hat{y}^2} \\ 2\dfrac{\partial^2 \hat{u}_z}{\partial \hat{x} \partial \hat{y}} \end{bmatrix} = [\psi]\{\theta\} \tag{A5}$$

where

$$[\psi]^T = \left(\frac{\partial^2 N_i}{\partial \hat{x}^2} \quad \frac{\partial^2 N_i}{\partial \hat{y}^2} \quad 2\frac{\partial^2 N_i}{\partial \hat{x} \partial \hat{y}} \right)$$

With the element displacement defined by equation (A1) and the slopes by equation (A2), Meek (Ref. 27) gets the following shape functions:

$$\begin{aligned}
N_1 &= b_2(\zeta_1^2\zeta_3 + \tfrac{1}{2}\zeta_1\zeta_2\zeta_3) - b_3(\zeta_1^2\zeta_2 + \tfrac{1}{2}\zeta_1\zeta_2\zeta_3) \\
N_2 &= a_2(\zeta_1^2\zeta_3 + \tfrac{1}{2}\zeta_1\zeta_2\zeta_3) - a_3(\zeta_1^2\zeta_2 + \tfrac{1}{2}\zeta_1\zeta_2\zeta_3) \\
N_3 &= b_3(\zeta_2^2\zeta_1 + \tfrac{1}{2}\zeta_1\zeta_2\zeta_3) - b_1(\zeta_2^2\zeta_3 + \tfrac{1}{2}\zeta_1\zeta_2\zeta_3) \\
N_4 &= a_3(\zeta_2^2\zeta_1 + \tfrac{1}{2}\zeta_1\zeta_2\zeta_3) - a_1(\zeta_2^2\zeta_3 + \tfrac{1}{2}\zeta_1\zeta_2\zeta_3) \\
N_5 &= b_1(\zeta_3^2\zeta_2 + \tfrac{1}{2}\zeta_1\zeta_2\zeta_3) - b_2(\zeta_3^2\zeta_1 + \tfrac{1}{2}\zeta_1\zeta_2\zeta_3) \\
N_6 &= a_1(\zeta_3^2\zeta_2 + \tfrac{1}{2}\zeta_1\zeta_2\zeta_3) - a_2(\zeta_3^2\zeta_1 + \tfrac{1}{2}\zeta_1\zeta_2\zeta_3)
\end{aligned} \tag{A6}$$

When second partial derivatives of these shape functions are obtained, corresponding to the expressions in equation (A5), we get

$$\begin{aligned}
\frac{\partial^2 N_1}{\partial \hat{x}^2} = \frac{1}{(2A)^2} [&\zeta_1 b_1 b_3 (b_2 - b_3) + \zeta_2 b_1 b_3 (b_2 - b_3 - 2b_1) \\
&+ \zeta_3 b_1 b_2 (b_2 - b_3 - 2b_1)]
\end{aligned} \tag{A7}$$

$$\frac{\partial^2 N_2}{\partial \hat{x}^2} = \frac{1}{(2A)^2}\Big\{\zeta_1[a_2b_3(b_2+4b_1) - a_3b_2(b_3+4b_1)] + \zeta_2 b_1[a_2b_3 - a_3(b_3+2b_1)] + \zeta_3 b_1[a_2(b_2+2b_1) - a_3b_2]\Big\}$$

$$\frac{\partial^2 N_3}{\partial \hat{x}^2} = \frac{1}{(2A)^2}[\zeta_1 b_2b_3(2b_2+b_3-b_1) + \zeta_2 b_1b_3(b_3-b_1) + \zeta_3 b_1b_2(b_3-2b_2-b_1)]$$

$$\frac{\partial^2 N_4}{\partial \hat{x}^2} = \frac{1}{(2A)^2}\Big\{\zeta_1[a_3b_2(2b_2+b_3) - a_1b_2b_3] + \zeta_2[a_3b_1(4b_2+b_3) - a_1b_3(4b_2+b_1)] + \zeta_3[a_3b_1b_2 - a_1b_2(2b_2+b_1)]\Big\}$$

$$\frac{\partial^2 N_5}{\partial \hat{x}^2} = \frac{1}{(2A)^2}[\zeta_1 b_2b_3(b_1-b_2-2b_3) + \zeta_2 b_1b_3(b_1-b_2+2b_3) + \zeta_3 b_1b_2(b_1-b_2)]$$

(A7) (Contd)

$$\frac{\partial^2 N_6}{\partial \hat{x}^2} = \frac{1}{(2A)^2}\Big\{\zeta_1[a_1b_2b_3 - a_2b_3(b_2+2b_3)] + \zeta_2[a_1b_3(b_1+2b_3) - a_2b_1b_3] + \zeta_3[a_1b_2(b_1+4b_3) - a_2b_1(b_2+4b_3)]\Big\}$$

$$\frac{\partial^2 N_1}{\partial \hat{y}^2} = \frac{1}{(2A)^2}\Big\{\zeta_1[a_3b_2(a_2+4a_1) - a_2b_3(a_3+4a_1)] + \zeta_2[a_1a_3b_2 - a_1b_3(a_3+2a_1)] + \zeta_3[a_1b_2(a_2+2a_1) - a_1a_2b_3]\Big\}$$

$$\frac{\partial^2 N_2}{\partial \hat{y}^2} = \frac{1}{(2A)^2}[\zeta_1 a_2a_3(a_2-a_3) + \zeta_2 a_1a_3(a_2-a_3-2a_1) + \zeta_3 a_1a_2(a_2-a_3+2a_1)]$$

$$\frac{\partial^2 N_3}{\partial \hat{y}^2} = \frac{1}{(2A)^2}\Big\{\zeta_1 a_2[b_3(2a_2+a_3) - a_3b_1] + \zeta_2[a_1b_3(a_3+4a_2) - a_3b_1(a_1+4a_2)] + \zeta_3 a_2[a_1b_3 - b_1(a_1+2a_2)]\Big\}$$

$$\frac{\partial^2 N_4}{\partial \hat{y}^2} = \frac{1}{(2A)^2}\,[\zeta_1 a_2 a_3 (a_3 + 2a_2 - a_1) + \zeta_2 a_1 a_3 (a_3 - a_1) + \zeta_3 a_1 a_2 (a_3 - 2a_2 - a_1)]$$

$$\frac{\partial^2 N_5}{\partial \hat{y}^2} = \frac{1}{(2A)^2}\,\{\zeta_1 a_2 [a_2 b_1 - b_2 (a_2 + 2a_2)] + \zeta_2 a_3 [b_1 (a_2 + 2a_3) - a_1 b_2] + \zeta_3 [a_2 b_1 (a_1 + 4a_3) - a_1 b_2 (a_1 + 4a_3)]\}$$

$$\frac{\partial^2 N_6}{\partial \hat{y}^2} = \frac{1}{(2A)^2}\,[\zeta_1 a_2 a_3 (a_1 - a_2 - 2a_3) + \zeta_2 a_1 a_3 (a_1 - a_2 + 2a_3) + \zeta_3 a_1 a_2 (a_1 - a_2)]$$

$$\frac{\partial^2 N_1}{\partial \hat{x} \partial \hat{y}} = \frac{1}{2(2A)^2}\{\zeta_1 [(b_2 - b_3)(a_2 b_3 + a_3 b_2) + 4b_1 (a_3 b_2 - a_2 b_3)] + \zeta_2 [(b_2 - b_3)(a_3 b_1 + a_1 b_3) - 4a_1 b_1 b_3] + \zeta_3 [(b_2 - b_3)(a_2 b_1 + a_1 b_2) + 4a_1 b_1 b_2]\} \qquad \text{(A7) (Contd)}$$

$$\frac{\partial^2 N_2}{\partial \hat{x} \partial \hat{y}} = \frac{1}{2(2A)^2}\{\zeta_1 [(a_2 - a_3)(a_3 b_2 + a_2 b_3) + 4a_1 (a_2 b_3 - a_3 b_2)] + \zeta_2 [(a_2 - a_3)(a_1 b_3 + a_3 b_1) - 4a_1 a_3 b_1] + \zeta_3 [(a_2 - a_3)(a_1 b_2 + a_2 b_1) + 4a_1 a_2 b_1]\}$$

$$\frac{\partial^2 N_3}{\partial \hat{x} \partial \hat{y}} = \frac{1}{2(2A)^2}\{\zeta_1 [(b_3 - b_1)(a_3 b_2 + a_2 b_3) + 4a_2 b_2 b_3] + \zeta_2 [(b_3 - b_1)(a_3 b_1 + a_1 b_3) + 4b_2 (a_1 b_3 - a_3 b_1)] + \zeta_3 [(b_3 - b_1)(a_2 b_1 + a_1 b_2) - 4a_2 b_1 b_2]\}$$

$$\frac{\partial^2 N_4}{\partial \hat{x} \partial \hat{y}} = \frac{1}{2(2A)^2}\{\zeta_1 [(a_3 - a_1)(a_3 b_2 + a_2 b_3) + 4a_2 a_3 b_2] + \zeta_2 [(a_3 - a_1)(a_3 b_1 + a_1 b_3) + 4a_2 (a_3 b_1 - a_1 b_3)] + \zeta_3 [(a_3 - a_1)(a_3 b_1 + a_1 b_2) - 4a_1 a_2 b_2]\}$$

$$\frac{\partial^2 N_5}{\partial\hat{x}\partial\hat{y}} = \frac{1}{2(2A)^2}\Big\{\zeta_1[(b_1-b_2)(a_3b_2+a_2b_3) - 4a_3b_2b_3]$$
$$+\zeta_2[(b_1-b_2)(a_3b_1+a_1b_3) + 4a_3b_1b_3]$$
$$+\zeta_3[(b_1-b_2)(a_2b_1+a_1b_2) + 4b_3(a_2b_1-a_1b_2)]\Big\}$$

$$\frac{\partial^2 N_6}{\partial\hat{x}\partial\hat{y}} = \frac{1}{2(2A)^2}\Big\{\zeta_1[(a_1-a_2)(a_3b_2+a_2b_3) - 4a_2a_3b_3] \qquad \text{(A7)(Concl)}$$
$$+\zeta_2[(a_1-a_2)(a_3b_1+a_1b_3) + 4a_1a_3b_3]$$
$$+\zeta_3[(a_1-a_2)(a_1b_2+a_2b_1) + 4a_3(a_1b_2-a_2b_1)]\Big\}$$

where A is the area of the plate and the constants are related to the coordinates of the node points by:

$$b_1 = \hat{y}_j - \hat{y}_k \;; \qquad a_1 = \hat{x}_k - \hat{x}_j$$

$$b_2 = \hat{y}_k \;; \qquad a_2 = -\hat{x}_k$$

$$b_3 = -\hat{y}_j \;; \qquad a_3 = \hat{x}_j$$

APPENDIX B

COMPUTER PROGRAM INPUT

This appendix presents a description of the computer program input including definition of the input quantities, descriptions of the user supplied driving subroutine, FREEFD, and external data file options and requirements.

1. INPUT FORMAT FOR DOT VEHICLE STRUCTURES ANALYSIS PROGRAM

Card No.	Format	Variable Name	Description
1	20A4	TITLE(20)	80 character run title
2	6I5,E10.6, 2I5	NNODE, NELE, NUMMAT, NUMDIS, MXSTEP, NDGREE, DELT, NCORD, NSECT	NNODE - number of active nodes NELE - number of elements NUMMAT - number of materials NUMDIS - number of displacement specified node points MXSTEP - number of time steps NDGREE - number of degrees of freedom per node (= 6 this version) DELT - time step (sec) NCORD - number of inactive beam pointer nodes NSECT - number of different beam cross section types
3	10I5	KONTRL(10)	Miscellaneous control flags KONTRL(1) = none (2) = none (3) = expansion factor in mesh generation (4) = none (5) = 0;(β parameter) (6) = 0;(iteration parameter) (7) = number of thickness integration points (= 2, if zero) (8) = check point interval = 0;no operation > 0;made check point < 0;restart and make check points (9) = restart checkpoint number (10) = time history data flag = 0;no operation > 0;make time history file
4.1	I5	MTYPE	MTYPE = material type number

Card No.	Format	Variable Name	Description
4.2	8E10.4	E(1) thru E(8)	Material properties E(1) - density (lb-sec^2/in.^{4}) E(2) - E; modulus of elasticity E(3) - σy; yield stress (psi) 0 ; elastic E(4) - Ep; plastic modulus E(5) - plate thickness (in.) E(6) - σ_{ult}; ultimate stress (beams) E(7) - ν; Poissons ratio E(8) - membrane damping as fraction of critical
4.3	8E10.4	E(9) thru E(16)	E(9) - none E(10) - none E(11) - none E(12) - none E(13) - none E(14) - none E(15) - none E(16) - none
			Repeat Group 4 NUMMAT Times
5.1	2I5,2E10.4	NSEG, KLOS, FIYY, FIZZ	NSEG - number of plate segments in beam cross section KLOS - flag for closed wall sections = 0; closed wall = 1; open wall FIYY,- cross section moments of inertia about Y and Z FIZZ principal axes, respectively
5.2	3E10.4	Y(I),Z(I),T(I)	Y(I),- cross section coordinates of point at begi Z(I) ning of segment I (in.)(See figure 5) T(I) - segment thickness (in.)

Card No.	Format	Variable Name	Description
			Repeat Card 5.2 NSEG+KLOS Times
			Repeat Group 5 NSECT Times
			Omit Group 5 if NSECT=0
6	I5, 5X, 3E10.4	N, XC(N), YC(N), ZC(N), PMASS(I,N),I=1,4	Node point data N - node number XC(N) - x YC(N) - y } coordinates of node point ZC(N) - z PMASS - additional (nonstructural) lumped mass at node N (1,N) = translation mass (2,N) (3,N) (4,N) = { added rotational mass Ix,Iy,Iz in global components
			Repeat Group 6 NNODE + NCORD Times
			Missing nodes are filled in sequentially and spaced uniformly or according to KONTRL(3) parameter.
7	10I5	M, NODE(9)	Element definition: M - element number Plate element: NODE(1) NODE(2) NODE(3) } plate corner nodes NODE(4) NODE(5) NODE(6) NODE(7) } not applicable

Node No.	Format	Variable Name	Description
7			Beam element: NODE(1) } beam end points NODE(2) } NODE(3) } master nodes for NODE(1) NODE(4) } and NODE(2), respectively NODE(5) - not applicable NODE(6) - cross section type NODE(7) - pointer node for initial principal axis $\hat{y}$ NODE(8) - element type = 0 or 1; plate = 2; beam NODE(9) - material type = 1 if zero
			Repeat Group 7 up to NELE Times
			Missing elements generated sequentially as: NODE(I,M+1) = NODE(I,M)+1 NODE(9,M+1) = NODE(9,M)
			Last element in sequence must be read from card
8	I10, 3E10.4	NODDIS(I), ANGLE(3,I)	Node constraint data: NODDIS - packed word - NABC (right justified) where N = node number and positions A,B,C correspond to X,Y,Z (or $\theta_x, \theta_y, \theta_z$) Displacement constraint: A,B, or C = 0, no constraint = 1, constrained = 2, specified in FREEFD

Card No.	Format	Variable Name	Description
8			Rotation constraint: A,B, or C = 3; no constraint = 4; constrained = 5; specified in FREEFD
			ANGLE(3) - constraint angles; inoperative this version
			Repeat Group 8 NUMDIS Times
			Omit Group 8 if NUMDIS = 0
9	6I10	NPFREQ, NPRU, NPRS, NPRV, NPRP, NPIC	Output control parameters NPFREQ - output frequency (number of iterations) NPRU - number of output nodes NPRS - number of stress output elements NPRV - inoperative NPRP - inoperative NPIC - number of picture output times
9.1	I10	OUT(1)	Displacement output - packed work - NABC (right justified) N = node number A = 1 - x = 2 - y = 3 - z = 4 - θ_x = 5 - θ^y = 6 - θ^z B = 0 - A = 1 - Á = 2 - Ă C = 0
			Repeat Group 9.1 NPRU Times; Omit for NPRU = 0

Card No.	Format	Variable Name	Description
9.2	I10	IOUT(1)	Stress output Packed work - N; currently inoperative
		Repeat Group 9.2 NPRS Times; Omit for NPRS = 0	
9.3	2I10	NPOUT(2)	Picture output NPOUT(1) - time count for picture output NPOUT(2) - picture type =1; displacement =2; velocity =3; acceleration =4; stresses
		Repeat Group 9.3 NPIC Times; Omit for NPIC = 0	

2. EXTERNAL FILE REQUIREMENTS

Restart unit - Logical unit 25

Permanent storage for checkpoint data; approximately 27,000 words per checkpoint required. Read and write operations are unformatted FORTRAN type.

Time history unit - Logical unit 22

Permanent storage containing header records and all output quantities (i.e., NPRU, etc.); approximately NPRUxMXSTEP words are required. Read and write operations are unformatted FORTRAN type.

Mesh data file - Logical unit 20

Permanent storage containing initial mesh data (nodes and elements) and displacements for all nodes at selected time steps (NPIC); the approximate number of words required is

5xNUMEL+4x(NPIC+1)xNNODE.

Read and write operations are formatted FORTRAN type.

3. FORCING FUNCTION SPECIFICATION

Forcing functions (i.e., displacements on nodal forces as functions of time) are presently provided through a user supplied subroutine FREEFD. The program may be used both for initialization and subsequent driving forces and motions based on the fixed point variable, KONWAV.

KONWAV=2 - initialization
KONWAV=1 - forcing functions

Displacements, velocities, accelerations and external nodal forces are found in UD, UD1, UD2 and FORCD; counted six per node in the sequence u_x, u_y, u_z, θ_x, θ_y, θ_z. The program is called at every time step and the current valued time and the time count are contained in variables TIME and NTSTEP. These may be used as arguments in expressions for forces or motions as functions of time.

APPENDIX C

PROGRAM DESCRIPTION

1. GENERAL

This appendix presents a description of the computer program developed in this investigation for the dynamic, large deflection elastic-plastic response of vehicle structures. The program contains beam and triangular plate finite elements and treats the large deflection phenomena by decomposing the element deformations into rigid body and deformation components before computing element forces. The numerical integration technique employed is an explicit Newmark β-method with $\beta = 0$ and a lumped mass approximation for the inertial properties.

As described in this appendix the program consists of approximately 2000 cards and normally executes in 65K words of core storage on the UNIVAC 1108 computer under the EXEC-8, version 27 operating system. A version of the program has been prepared and executed in 240K bytes on an IBM 370/158 system under the H compiler with no optimization. The capacity of the program as described here is approximately 150 nodes, 150 elements, 100 displacement boundary conditions, 5 different sets of material properties and 10 distinct beam cross sections. The solution procedure is completely in core and makes no use of peripheral devices for temporary (i.e., scratch) storage during execution. However, a restart capability is provided via logical unit 25 and output data storage is provided via units 20 and 22. The use of these files is optional and they should be defined as dummy files if not required. The following sections of this appendix provide a brief description of the arrangement and function of the various subroutines and the allocation of the main data storage.

2. SUBROUTINE DESCRIPTIONS

The overall structure and the names and logical relation among subroutines is shown in Figure C-1. The subroutines collect into three main operating groups relating to data input (READIN), preliminary calculations (ASSBLE), and the main solution procedure (SOLVE) plus four small subroutines (utilities) which are used throughout the program. Brief descriptions of the functions of the individual programs follow.

MAIN. The main program is fairly short and contains dimensioning for the main data storage and calls to READIN, ASSBLE, and SOLVE. No computational or control operations are carried out in this program except the calls to these subroutines.

READIN. This program reads all of the input data from cards (except for beam cross section data) according to the input formats described in Appendix B and initializes the data files if required.

BGEOM. Reads data describing the cross section properties of beams and generates certain additional data such as segment lengths and the torsional constants.

ASSBLE. This program initilizes the element data storage, assembles the mass matrix and the initial transformations, $[B_o]$ for the nodal coordinate systems and generates the initial values of the pointing vectors $[\overline{n}]$, $[\overline{\eta}]$ and $[\overline{\xi}]$ and the nodal components of the rigid links $[\overline{\Delta}]$.

TASME. Calculates contributions to the lumped mass matrix for plate elements and forms the initial element coordinate system [E] for the plate.

QUAD. An algorithm for lumping the plate rotational inertia at the three defining nodes based on the properties of three quadrilaterals formed in the triangle by extending perpendiculars from the centroid to three sides.

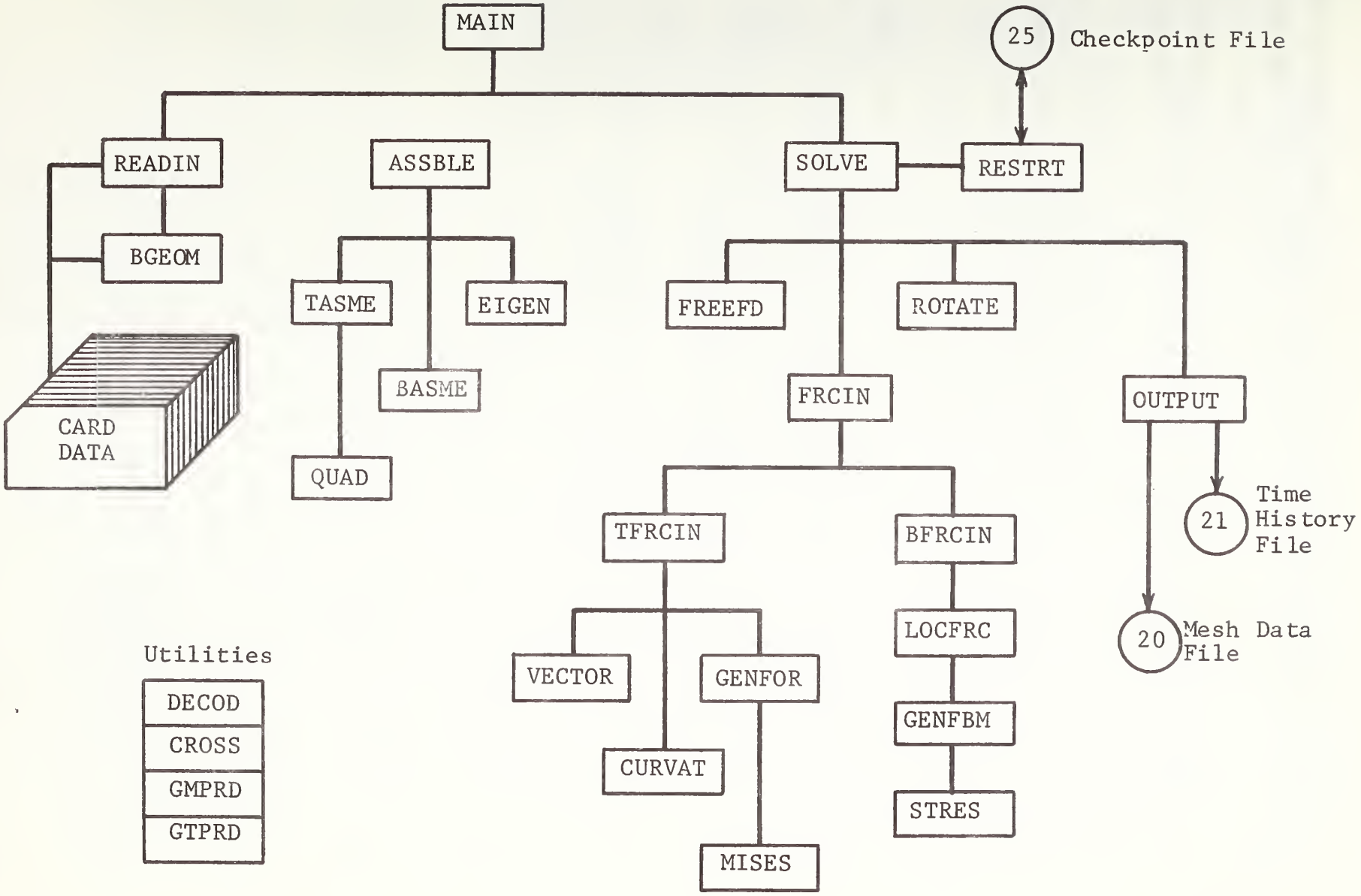


Figure C-1 Computer Program Structure

BASME. Calculates contributions to the lumped mass matrix for beam elements and forms the initial element coordinate system [E].

EIGEN. Generates the principal values and directions for the assembled rotational mass matrix at a node. An IBM utility program found in Ref. 29.

SOLVE. This program contains the numerical integration procedure and the equations of motion (equation (2)). The principal subroutines called by this program are FREEFD for external driving forces or motions, FRCIN for internal nodal forces and OUTPUT which provides printed output and data files.

FREEFD. This program provides nonzero initial conditions or external driving forces or motions at specified nodes and is modified for each application as described in Appendix B.

RESTRT. This program generates or reads checkpoint data files which contain sufficient information to restart the solution procedure at specified values of time step.

ROTATE. The program is used to constrain nodal motions in directions other than parallel to the global axes.

FRCIN. Provides internal nodal forces for use in the equations of motion within SOLVE. The program updates the nodal coordinate transformations [B], and calls subroutines TFRCIN for plate forces and BFRCIN for beam forces.

TFRCIN. Calculates plate element deformations in the element coordinate system, calculates nodal forces in the element system and transforms forces to the nodal coordinate system.

VECTOR. Calculates the deformed length of plate element sides and the components of a vector normal to the plate reference surface defined by the displaced positions of the three node points.

CURVAT. Provides algebraic expressions for curvature components at the midpoints of the three element sides as functions of the nodal rotations.

GENFOR. Controls the numerical integration of stresses through the thickness of the plate to obtain internal cross section moments and forces.

MISES. Provides the biaxial, elastic-plastic stress strain algorithm described in Section 2.4.1.

BFRCIN. Calculates beam element deformations in the element coordinate system, calculates nodal forces in the element system, performs the operations associated with the master-slave relations and transforms forces to the nodal coordinate systems.

LOCFRC. Calculates midplane strains and curvatures for the beam and nodal forces and moments in the beam element coordinate system.

GENFBM. Controls the numerical integration of stresses over the cross section of the beam to obtain internal cross section moments and forces.

STRES. Provides the uniaxial, elastic-plastic stress strain algorithm.

OUTPUT. Provides printed output of those quantities selected for output, and updates the time-history and deformed geometry data files.

3. DATA ALLOCATION

The allocation of the main items of data* throughout the program is as follows.

COMMON (blank)

XC(300)	-- Undeformed nodal coordinates
YC(300)	
ZC(300)	
UD(900)	-- Nodal displacement, velocity and acceleration (six per node)
UD1(900)	
UD2(900)	
FORCD(900)	-- External nodal forces
E(16,5)	-- Properties for up to five different materials
IX(9,300)	-- Array of element node correspondences and other element parameters
NODDIS(100)	-- Array of packed word constraint data
NUMNP	-- Number of node points
NUMEL	-- Number of elements
NDGREE	-- Degrees of freedom per node (6)
MUD	-- Not used
NUMDIS	-- Number of displacement points

COMMON/CONTRL/

KONTRL(10)	-- Control flags

COMMON/DYNAM/

DELT	-- Time increment
TIMEND	-- Maximum time value
MXSTEP	-- Maximum number of time steps
NTSTEP	-- Current step count
TIME	-- Current value of time

*The names of arrays, and number of subscripts differ in some subroutines with correspondence arranged through subroutine calling arguments or explicit EQUIVALENCE statements.

COMMON/XSECT/

NSEG(10) -- Number of segments in beam cross sections

Y(20,10) -- Beam cross section coordinates

Z(20,10)

XLEN(20,10) -- Beam cross section segment lengths and thicknesses

T(20,10)

TORCON(10) -- Torsional constant

KLOS(10) -- Closed wall section flag

ABM(10) -- Cross section area and moments of inertia about $\hat{x}$, $\hat{y}$ and $\hat{z}$ axes

FIXX(10)

FIYY(10)

FIZZ(10)

COMMON/STR/

I1 -- Number of plate cross section integration points

INDEX(300) -- Array of pointers to first item in element stress array STRS

STRAIN(45) -- Working storage for stress and strain calculations

STRESS(45)

SIDE(900) -- Initial lengths of plate element sides (three per plate element)

IP(6) -- Plate element side/node counter

STRS(10000) -- Element stress and strain storage. Format for plates is 3xI1x7 with each block of 7 laid out as

$[\varepsilon_{xx}, \varepsilon_{yy}, \varepsilon_{xy}, \sigma_{xx}, \sigma_{yy}, \sigma_{xy}, \sigma_{y}]$

for I1 points on 3 sides for each element. Format for beams is 2xNPTSx4 with each block of 4 laid out as

$[\varepsilon_{xx}, \sigma_{xx}, \sigma_{y}$, blank]

for NPTS points at 2 ends of the beam. Note that NPTS=NSEG+KLOS.

COMMON/JUNK/

TL(9,300) -- Components of the [B] transformation laid out by columns of three, three columns per node

POINT(3,3,300)-- Components of initial plate normal in $[\bar{x},\bar{y},\bar{z}]$ system

COMMON/BEAMS/

RMASS(3,3,100) -- Components of assembled rotational inertia tensor before diagonalization

DICOS(3,3,150) -- Components of initial element transformation [E]

BAR(1200) -- Beam and rigid link storage laid out as

$[\bar{\eta}_o; \bar{\xi}_o; \Delta^o_G; \bar{\Delta}]$ = 12 words

at two ends of beam = 24 words per beam

PMASS(4,100) -- Added nonstructural lumped mass at nodes -- one translational and three rotational components per node

NSECT -- Number of beam cross section types

COMMON/OUTPA/

NPRU -- Number of displacement output points

NPRS -- Number of stress output points

NPFREQ -- Print frequency

NPIC -- Number of picture output intervals

OUT(9,40) -- Packed word output list

NPOUT(2,10) -- Picture output list

TITLE(20) -- Problem title

APPENDIX D

PROGRAM LISTINGS

```
C                                                                       00000100
C        * * *   W R E C K E R * * *                                    00000200
C                                                                       00000300
C        A FINITE ELEMENT COMPUTER PROGRAM FOR THE ANALYSIS OF          00000400
C        AUTOMOTIVE STRUCTURES UNDER CRASH LOADINGS                     00000500
C                                                                       00000600
C      BY    IIT RESEARCH INSTITUTE                                     00000700
C      FOR    NHTSA - DOT                                             00000800
C                                                                       00000900
C      JUNE 1975                                                        00001000
C                                                                       00001100
C                                                                       00001200
      COMMON/WORK/ANGLE(600)                                            00001300
      COMMON XC(300),YC(300),ZC(300),UD(900),UD1(900),UD2(900),         00001400
     *FORCD(900),E(16,5),                                               00001500
     *IX(9,300),NODDIS(100),NUMNP,NUMEL,NDGREE,MUD,NUMDIS               00001600
      COMMON/CONTRL/KONTRL(10)                                          00001700
       COMMON/DYNAM/DELT,TIMEND,MXSTEP,NTSTEP,TIME,EYMED                00001800
      COMMON/STR/I1,INDEX(300),STRAIN(45),STRESS(45),SIDE(900),IP(6),   00001900
     *STRS(10000)                                                       00002000
       COMMON /BEAMS/RMASS(3,3,100),DICOS(3,3,150),BAR(1200),           00002100
     *PMASS(4,100),NSECT                                                00002200
      DIMENSION XO(900),X1(900),V(900),AO(900),A1(900),SMASS(900)       00002300
        DIMENSION FINT(900)                                             00002400
      EQUIVALENCE (X1(1),UD(1)),(V(1),UD1(1)),(AO(1),UD2(1))            00002500
    1  CALL READIN(NUMEL,NUMNP,NDGREE,MUD,NUMDIS,NODDIS,                00002600
     *E,XC,YC,ZC,IX,ANGLE,UD)                                           00002700
      CALL ASSBLE(NUMEL,NUMNP,NDGREE,IX,XC,YC,ZC,E,SMASS,STRS,STRESS,   00002800
     1 STRAIN,INDEX,I1,SIDE,IP)                                         00002900
      CALL SOLVE ( NUMEL,NUMNP,NDGREE,NUMDIS,NODDIS,SMASS,XC,YC,ZC,IX,  00003000
     *E,XO,X1,V,AO,A1,FORCD,FINT,ANGLE)                                 00003100
       GO  TO  1                                                        00003200
      END                                                               00003300
```

```
      SUBROUTINE ROTATE(ANGLE,JNOD,U,ND,KONTRL)                          00000100
     DIMENSION U(1),ANGLE(1)                                             00000200
     DIMENSION S(3),C(3),UC(3),K(3)                                      00000300
      NDGREE=3                                                           00000400
     DO 100 J=1,NDGREE                                                   00000500
     S(J)=SIN(ANGLE(J))                                                  00000600
     C(J)=COS(ANGLE(J))                                                  00000700
 100 K(NDGREE+1-J)=NDGREE*JNOD+1-J                                       00000800
     ENTRY ROTAT(U,KONTRL)                                               00000900
     K1=K(1)                                                             00001000
     K2=K(2)                                                             00001100
     K3=K(3)                                                             00001200
     DO 105 J=1,NDGREE                                                   00001300
     IC=K(J)                                                             00001400
 105 UC(J)=U(IC)                                                         00001500
     GO TO (110,125), KONTRL                                             00001600
 110 GO TO (115,115,120), NDGREE                                         00001700
 115 U(K1)=C(1)*UC(1)+S(1)*UC(2)                                         00001800
     U(K2)=-S(1)*UC(1)+C(1)*UC(2)                                        00001900
     RETURN                                                              00002000
 120 U(K1)=(C(1)*C(3)-S(1)*S(2)*S(3))*UC(1)+(S(1)*C(3)                   00002100
    1+C(1)*S(2)*S(3))*UC(2)-C(2)*S(3)*UC(3)                              00002200
     U(K2)=-S(1)*C(2)*UC(1)+C(1)*C(2)*UC(2)+S(2)*UC(3)                   00002300
     U(K3)=(C(1)*S(3)+S(1)*S(2)*C(3))*UC(1)+(S(1)*S(3)                   00002400
    1-C(1)*S(2)*C(3))*UC(2)+C(2)*C(3)*UC(3)                              00002500
     RETURN                                                              00002600
 125 GO TO (130,130,135), NDGREE                                         00002700
 130 U(K1)=C(1)*UC(1)-S(1)*UC(2)                                         00002800
     U(K2)=S(1)*UC(1)+C(1)*UC(2)                                         00002900
     RETURN                                                              00003000
 135 U(K1)=(C(1)*C(3)-S(1)*S(2)*S(3))*UC(1)-S(1)*C(2)*UC(2)              00003100
    1+(C(1)*S(3)+S(1)*S(2)*C(3))*UC(3)                                   00003200
     U(K2)=(S(1)*C(3)+C(1)*S(2)*S(3))*UC(1)+C(1)*C(2)*UC(2)              00003300
    1+(S(1)*S(3)-C(1)*S(2)*C(3))*UC(3)                                   00003400
     U(K3)=-C(2)*S(3)*UC(1)+S(2)*UC(2)+C(2)*C(3)*UC(3)                   00003500
     RETURN                                                              00003600
     END                                                                 00003700
```

```
      SUBROUTINE CURVAT(CURV,AREA,XROT,YROT,D,CX,CY)                    00000100
      DIMENSION FZ(1),SMX(1),SMY(1),XROT(1),YROT(1),D(1)                00000200
      DIMENSION C(3,6,3),CURV(3,3),EM(3,3)                              00000300
      A2=-CX*D(3)                                                       00000400
      A3=CX*D(1)                                                        00000500
      A1=-A2-A3                                                         00000600
      B2=-CY*D(3)                                                       00000700
      B3=-CY*D(1)                                                       00000800
      B1=-B2-B3                                                         00000900
      CONST=(0.5/AREA)**2                                               00001000
      DO 130 J=1,3                                                      00001100
      GO TO (110,115,120), J                                            00001200
  110 CONTINUE                                                          00001300
      C(1,1,1)=0.5*B1*(B3**2-B2**2)                                     00001400
      C(1,2,1)=0.5*B1**2*(A2-A3)                                        00001500
      C(1,3,1)=B1*(0.5*B2*(3.*B3-B2)+B3**2)                             00001600
      C(1,4,1)=0.5*B1**2*(A1-A3)-B2*(A1*B2+4.*AREA)                     00001700
      C(1,5,1)=B1*(0.5*B3*(B3-3.*B2)-B2**2)                             00001800
      C(1,6,1)=0.5*B1**2*(A2-A1)+B3*(A1*B3-4.*AREA)                     00001900
      C(2,1,1)=0.5*A1**2*(B2-B3)                                        00002000
      C(2,2,1)=0.5*A1*(A3**2-A2**2)                                     00002100
      C(2,3,1)=0.5*A1**2*(B1-B3)-A2*(A2*B1-4.*AREA)                     00002200
      C(2,4,1)=A1*(0.5*A2*(3.*A3-A2)+A3**2)                             00002300
      C(2,5,1)=0.5*A1**2*(B2-B1)+A3*(A3*B1+4.*AREA)                     00002400
      C(2,6,1)=A1*(0.5*A3*(A3-3.*A2)-A2**2)                             00002500
      C(3,1,1)=0.5*A1*B1*(B2-B3)                                        00002600
      C(3,2,1)=0.5*A1*B1*(A2-A3)                                        00002700
      C(3,3,1)=A1*(B1**2-B2**2+0.5*B1*B2)                               00002800
      C(3,4,1)=B1*(A1**2-A2**2+0.5*A1*A2)                               00002900
      C(3,5,1)=A1*(B3**2-B1**2-0.5*B1*B3)                               00003000
      C(3,6,1)=B1*(A3**2-A1**2-0.5*A1*A3)                               00003100
      GO TO 125                                                         00003200
  115 CONTINUE                                                          00003300
      C(1,1,2)=B2*(0.5*B1*(B1-3.*B3)-B3**2)                             00003400
      C(1,2,2)=0.5*B2**2*(A3-A2)+B1*(A2*B1-4.*AREA)                     00003500
      C(1,3,2)=0.5*B2*(B1**2-B3**2)                                     00003600
      C(1,4,2)=0.5*B2**2*(A3-A1)                                        00003700
      C(1,5,2)=B2*(0.5*B3*(3.*B1-B3)+B1**2)                             00003800
      C(1,6,2)=0.5*B2**2*(A2-A1)-B3*(A2*B3+4.*AREA)                     00003900
```

```
      C(2,1,2)=0.5*A2**2*(B3-B2)+A1*(A1*B2+4.*AREA)                      00004000
      C(2,2,2)=A2*(0.5*A1*(A1-3.*A3)-A3**2)                             00004100
      C(2,3,2)=0.5*A2**2*(B3-B1)                                        00004200
      C(2,4,2)=0.5*A2*(A1**2-A3**2)                                     00004300
      C(2,5,2)=0.5*A2**2*(B2-B1)-A3*(A3*B2-4.*AREA)                     00004400
      C(2,6,2)=A2*(0.5*A3*(3.*A1-A3)+A1**2)                             00004500
      C(3,1,2)=A2*(B1**2-B2**2-0.5*B1*B2)                               00004600
      C(3,2,2)=B2*(A1**2-A2**2-0.5*A1*A2)                               00004700
      C(3,3,2)=0.5*A2*B2*(B3-B1)                                        00004800
      C(3,4,2)=0.5*A2*B2*(A3-A1)                                        00004900
      C(3,5,2)=A2*(B2**2-B3**2+0.5*B2*B3)                               00005000
      C(3,6,2)=B2*(A2**2-A3**2+0.5*A2*A3)                               00005100
      GO TO 125                                                         00005200
  120 CONTINUE                                                          00005300
      C(1,1,3)=B3*(0.5*B1*(3.*B2-B1)+B2**2)                             00005400
      C(1,2,3)=0.5*B3**2*(A3-A2)-B1*(A3*B1+4.*AREA)                     00005500
      C(1,3,3)=B3*(0.5*B2*(B2-3.*B1)-B1**2)                             00005600
      C(1,4,3)=0.5*B3**2*(A1-A3)+B2*(A3*B2-4.*AREA)                     00005700
      C(1,5,3)=0.5*B3*(B2**2-B1**2)                                     00005800
      C(1,6,3)=0.5*B3**2*(A1-A2)                                        00005900
      C(2,1,3)=0.5*A3**2*(B3-B2)-A1*(A1*B3-4.*AREA)                     00006000
      C(2,2,3)=A3*(0.5*A1*(3.*A2-A1)+A2**2)                             00006100
      C(2,3,3)=0.5*A3**2*(B1-B3)+A2*(A2*B3+4.*AREA)                     00006200
      C(2,4,3)=A3*(0.5*A2*(A2-3.*A1)-A1**2)                             00006300
      C(2,5,3)=0.5*A3**2*(B1-B2)                                        00006400
      C(2,6,3)=0.5*A3*(A2**2-A1**2)                                     00006500
      C(3,1,3)=A3*(B3**2-B1**2+0.5*B1*B3)                               00006600
      C(3,2,3)=B3*(A3**2-A1**2+0.5*A1*A3)                               00006700
      C(3,3,3)=A3*(B2**2-B3**2-0.5*B2*B3)                               00006800
      C(3,4,3)=B3*(A2**2-A3**2-0.5*A2*A3)                               00006900
      C(3,5,3)=0.5*A3*B3*(B1-B2)                                        00007000
      C(3,6,3)=0.5*A3*B3*(A1-A2)                                        00007100
  125 CONTINUE                                                          00007200
      DO 130 I=1,3                                                      00007300
  130 CURV(I,J)=CONST*(C(I,1,J)*XROT(1)+C(I,2,J)*YROT(1)+C(I,3,J)       00007400
     1*XROT(2)+C(I,4,J)*YROT(2)+C(I,5,J)*XROT(3)+C(I,6,J)*YROT(3))      00007500
      RETURN                                                            00007600
      ENTRY NODALF(EM,SMX,SMY,FZ,AREA)                                  00007700
      CONST=CONST*AREA/3.                                               00007800
```

```
79.             DO 140 J=1,3                                                    00007900
80.             SMX(J)=0.                                                       00008000
81.             SMY(J)=0.                                                       00008100
82.             DO 140 K=1,3                                                    00008200
83.             DO 140 I=1,3                                                    00008300
84.             SMX(J)=SMX(J)+CONST*C(I,2*J-1,K)*EM(I,K)                        00008400
85.         140 SMY(J)=SMY(J)+CONST*C(I,2*J,K)*EM(I,K)                          00008500
86.             SMXT=-SMX(1)-SMX(2)-SMX(3)                                      00008600
87.             SMYT=SMY(1)+SMY(2)+SMY(3)                                       00008700
88.             CONST=0.5/AREA                                                  00008800
89.             FZ(2)=CONST*(A2*SMXT+B2*SMYT)                                   00008900
90.             FZ(3)=CONST*(A3*SMXT+B3*SMYT)                                   00009000
91.             FZ(1)=-FZ(2)-FZ(3)                                              00009100
92.             RETURN                                                          00009200
93.             END                                                             00009300
```

```
 1.            SUBROUTINE DECOD(ML,JJ)                                  00000100
 2.            DIMENSION JJ(1)                                          00000200
 3.            MK=ML                                                    00000300
 4.            JJ(4)=MK/1000                                            00000400
 5.            MK=MK-JJ(4)*1000                                         00000500
 6.            DO 10 I=1,2                                              00000600
 7.            J=10**(3-I)                                              00000700
 8.            JJ(I)=MK/J                                               00000800
 9.         10 MK=MK-JJ(I)*J                                            00000900
10.            JJ(3)=MK                                                 00001000
11.            RETURN                                                   00001100
12.            END                                                      00001200
```

```
1.          SUBROUTINE PRESS(IP,P)                                          00000100
2.           COMMON/DYNAM/DELT,TIMEND,MXSTEP,NTSTEP,TIME,EYMED              00000200
3.          P=100.                                                          00000300
4.          RETURN                                                          00000400
5.          END                                                             00000500
```

```
      SUBROUTINE QUAD(X1,Y1,X2,Y2,X3,Y3,A,XI,YI)                        00000100
C                                                                       00000200
C CALCULATES THE MOMENT OF INERTIA OF THE SUBTRIANGLES OF AN ELEMENT          00
C                                                                       00000400
      ICOUNT=0                                                          00000500
      E1=0.5                                                            00000600
      E2=0.5                                                            00000700
      E3=0.0                                                            00000800
   40 XI=A*((Y1*E1+Y2*E2+Y3*E3)**2)/3.+XI                               00000900
      YI=A*((X1*E1+X2*E2+X3*E3)**2)/3.+YI                               00001000
      ICOUNT=ICOUNT+1                                                   00001100
      GO TO (10,20,30),ICOUNT                                           00001200
   10 E1=0.0                                                            00001300
      E3=0.5                                                            00001400
      GO TO 40                                                          00001500
   20 E1=0.5                                                            00001600
      E2=0.0                                                            00001700
      GO TO 40                                                          00001800
   30 RETURN                                                            00001900
      END                                                               00002000
```

```
      SUBROUTINE GMPRD(A,B,R,N,M,L)                                     00000100
      DIMENSION A(1),B(1),R(1)                                          00000200
      IR=0                                                              00000300
      IK=-M                                                             00000400
      DO 10 K=1,L                                                       00000500
      IK=IK+M                                                           00000600
      DO 10 J=1,N                                                       00000700
      IR=IR+1                                                           00000800
      JI=J-N                                                            00000900
      IB=IK                                                             00001000
      R(IR)=0.                                                          00001100
      DO 10 I=1,M                                                       00001200
      JI=JI+N                                                           00001300
      IB=IB+1                                                           00001400
   10 R(IR)=R(IR)+A(JI)*B(IB)                                           00001500
      RETURN                                                            00001600
      END                                                               00001700
```

```
      SUBROUTINE GENFOR (EXT,CURV,AM,FX,E,H,I1,STRS                     )      00
C                                                                       00000200
C     PROGRAM FOR THRUST AND MOMENT FROM EXTENSION AND CURVATURE BY             00
C     NUMERICAL INTEGRATION AT I1 POINTS                                00000400
      DIMENSION E(1),EPS(3,5 ),S(3,5 ),STRS(1),Y(5 ),AM(3),FX(3),       00000500
     $EXT(3),CURV(3)                                                    00000600
C     I1.GE.2.AND.I1.LE.5  NOS. OF POINTS IN SECTION                    00000700
      FI1=I1                                                            00000800
      DH = H/(FI1-1.)                                                   00000900
C                                                                       00001000
C     CALCULATE CROSS-SECTION STRAIN AT 3 DIFFERENT DIRECTIONS ID OF            00
C     EACH ELEMENT                                                      00001200
C                                                                       00001300
      DO 1 I = 1,I1                                                     00001400
      FI = I-1                                                          00001500
      Y(I) = .5*H-FI*DH                                                 00001600
C                                                                       00001700
C     ID = DIRECTION                                                    00001800
C                                                                       00001900
      DO 2 ID = 1,3                                                     00002000
      EPS(ID,I) = EXT(ID)-Y(I)*CURV(ID)                                 00002100
    2 CONTINUE                                                          00002200
      IND = 7*(I-1)+1                                                   00002300
      CALL MISES (E,S(1,I),STRS(IND),EPS(1,I))                          00002400
    1 CONTINUE                                                          00002500
C                                                                       00002600
C     FORM FORCES AND MOMENTS FOR EACH DIRECTION ID                     00002700
C                                                                       00002800
      DO 3 ID = 1,3                                                     00002900
      FX(ID) = 0.                                                       00003000
      AM(ID) = 0.                                                       00003100
      DO 3 I = 2,I1                                                     00003200
      FX(ID) = FX(ID)+S(ID,I)+S(ID, I-1 )                               00003300
      AM (ID) = AM(ID)+S(ID,I)*(Y(I-1)+2.*Y(I))+S(ID, I-1 )*(2.*Y(I-1)          00
     $+Y(I))                                                            00003500
    3 CONTINUE                                                          00003600
      DO 4 ID = 1,3                                                     00003700
      FX (ID) = .5*DH*FX(ID)                                            00003800
      AM (ID) =  AM(ID)*DH/6.                                           00003900
```

```
40.          4 CONTINUE                                                 00004000
41.            RETURN                                                   00004100
42.            END                                                      00004200
```

```
 1.            SUBROUTINE MISES(E,S,STRS,EPS)                                  00000100
 2.             DIMENSION E(1),S(1),EPS(1),STRS(1)                             00000200
 3.           DIMENSION D(3),SO(3)                                             00000300
 4.           DO 100 J=1,3                                                     00000400
 5.           D(J)=EPS(J)-STRS(  J)                                            00000500
 6.       100 SO(J)=STRS(  J+3)                                                00000600
 7.       C   ELASTIC STRESSES                                                 00000700
 8.           STX=SO(1)+E(2  )/(1.-E(7  )**2)*(D(1)+E(7  )*D(2))               00000800
 9.           STY=SO(2)+E(2  )/(1.-E(7  )**2)*(D(2)+E(7  )*D(1))               00000900
10.           STXY=SO(3)+E(2  )/(1.+E(7  ))*D(3)/2.                            00001000
11.           IF(E(3  ).EQ.0.) GO TO 125                                       00001100
12.       C   EQUIVALENT STRESSES                                              00001200
13.           SIGNEW=SQRT(STX**2-STX*STY+STY**2+3.*STXY**2)                    00001300
14.           IF(SIGNEW.LE.STRS(  7)) GO TO 125                                00001400
15.           CONST=2.*E(4  )*(1.+E(7  ))/(3.*(E(2  )-E(4  )))                 00001500
16.           EQSTS=(STRS(7  )+CONST*SIGNEW)/(1.+CONST)                        00001600
17.           DLB=(SIGNEW/EQSTS-1.)/3.                                         00001700
18.           CON=1./(1.+3.*DLB)                                               00001800
19.           STRS(7  )=EQSTS                                                  00001900
20.           S(1)=(STX+DLB*(STX+STY))*CON                                     00002000
21.           S(2)=(STY+DLB*(STX+STY))*CON                                     00002100
22.           S(3)=STXY*CON                                                    00002200
23.           GO TO 20                                                         00002300
24.       125 S(1)=STX                                                         00002400
25.           S(2)=STY                                                         00002500
26.           S(3)=STXY                                                        00002600
27.        20 DO 10 I=1,3                                                      00002700
28.           STRS(I)=EPS(I)                                                   00002800
29.        10 STRS(I+3)=S(I)                                                   00002900
30.           RETURN                                                           00003000
31.           END                                                              00003100
```

```
      SUBROUTINE GTPRD(A,B,R,N,M,L)                                     00000100
      DIMENSION A(1),B(1),R(1)                                          00000200
      IR=0                                                              00000300
      IK=-N                                                             00000400
      DO 10 K=1,L                                                       00000500
      IJ=0                                                              00000600
      IK=IK+N                                                           00000700
      DO 10 J=1,M                                                       00000800
      IB=IK                                                             00000900
      IR=IR+1                                                           00001000
      R(IR)=0                                                           00001100
      DO 10 I=1,N                                                       00001200
      IJ=IJ+1                                                           00001300
      IB=IB+1                                                           00001400
   10 R(IR)=R(IR)+A(IJ)*B(IB)                                           00001500
      RETURN                                                            00001600
      END                                                               00001700
```

```
      SUBROUTINE STRES(STRAIN,SIGNEW,E,STRS)                            00000100
C                                                                       00000200
C       PROGRAM FOR LINEAR HARDENING BEAM MATERIAL                      00000300
C                                                                       00000400
C     STRS(1)=OLD STRAIN                                                00000500
C          2  = OLD STRESS                                              00000600
C          3  =  OLD YIELD STRESS                                       00000700
C                                                                       00000800
C                                                                       00000900
        DIMENSION E(1),STRS(1)                                          00001000
        EPSOLD=STRS(1)                                                  00001100
       SIGOLD=STRS(2)                                                   00001200
      SIGYLD=STRS(3)                                                    00001300
       DEPS=STRAIN-EPSOLD                                               00001400
       SIGNEW=SIGOLD+E(2)*DEPS                                          00001500
        IF(E(3).EQ.0.) GO TO 1                                          00001600
       ASIGN=ABS(SIGNEW)                                                00001700
        ASIGY=ABS(SIGYLD)                                               00001800
        IF(ASIGN.LE.ASIGY) GO TO 1                                      00001900
        DSIGE=ASIGY-ABS(SIGOLD)                                         00002000
       IF(SIGNEW*SIGOLD.LT.0.) DSIGE=ASIGY+ABS(SIGOLD)                  00002100
      DEPSE=SIGN(DSIGE/E(2),DEPS)                                       00002200
        SIGNEW=SIGN(ASIGY,SIGNEW)+E(4)*(DEPS-DEPSE)                     00002300
      IF(E(6).NE.0..AND.ABS(SIGNEW).GT.E(6))                            00002400
     * SIGNEW=SIGN(E(6),SIGNEW)                                         00002500
      STRS(3)=SIGNEW                                                    00002600
    1 STRS(1)=STRAIN                                                    00002700
       STRS(2)=SIGNEW                                                   00002800
      RETURN                                                            00002900
      END                                                               00003000
```

```
      SUBROUTINE EIGEN(A,R,N,MV)                                        00000100
      DIMENSION A(1),R(1)                                               00000200
    5 RANGE=1.0E-6                                                      00000300
      IF(MV-1)10,25,10                                                  00000400
   10 IQ=-N                                                             00000500
      DO 20 J=1,N                                                       00000600
      IQ=IQ+N                                                           00000700
      DO 20 I=1,N                                                       00000800
      IJ=IQ+I                                                           00000900
      R(IJ)=0.0                                                         00001000
      IF(I-J)20,15,20                                                   00001100
   15 R(IJ)=1.0                                                         00001200
   20 CONTINUE                                                          00001300
   25 ANORM=0.0                                                         00001400
      DO 35 I=1,N                                                       00001500
      DO 35 J=1,N                                                       00001600
      IF(I-J)30,35,30                                                   00001700
   30 IA=I+(J*J-J)/2                                                    00001800
      ANORM=ANORM+A(IA)*A(IA)                                           00001900
   35 CONTINUE                                                          00002000
      IF(ANORM)165,165,40                                               00002100
   40 ANORM=1.414*SQRT(ANORM)                                           00002200
      ANRMX=ANORM*RANGE/FLOAT(N)                                        00002300
      IND=0                                                             00002400
      THR=ANORM                                                         00002500
   45 THR=THR/FLOAT(N)                                                  00002600
   50 L=1                                                               00002700
   55 M=L+1                                                             00002800
   60 MQ=(M*M-M)/2                                                      00002900
      LQ=(L*L-L)/2                                                      00003000
      LM=L+MQ                                                           00003100
   62 IF(ABS(A(LM))-THR)130,65,65                                       00003200
   65 IND=1                                                             00003300
      LL=L+LQ                                                           00003400
      MM=M+MQ                                                           00003500
      X=0.5*(A(LL)-A(MM))                                               00003600
   68 Y=-A(LM)/SQRT(A(LM)*A(LM)+X*X)                                    00003700
      IF(X)70,75,75                                                     00003800
   70 Y=-Y                                                              00003900
```

```
40.        75 SINX=Y/SQRT(2.0*(1.0+(SQRT(1.0-Y*Y))))                                00004000
41.           SINX2=SINX*SINX                                                       00004100
42.        78 COSX=SQRT(1.0-SINX2)                                                  00004200
43.           COSX2=COSX*COSX                                                       00004300
44.           SINCS=SINX*COSX                                                       00004400
45.           ILQ=N*(L-1)                                                           00004500
46.           IMQ=N*(M-1)                                                           00004600
47.           DO 125 I=1,N                                                          00004700
48.           IQ=(I*I-I)/2                                                          00004800
49.           IF(I-L)80,115,80                                                      00004900
50.        80 IF(I-M)85,115,90                                                      00005000
51.        85 IM=I+MQ                                                            00005100
52.           GO TO 95                                                           00005200
53.        90 IM=M+IQ                                                            00005300
54.        95 IF(I-L)100,105,105                                                 00005400
55.       100 IL=I+LQ                                                            00005500
56.           GO TO 110                                                          00005600
57.       105 IL=L+IQ                                                            00005700
58.       110 X=A(IL)*COSX-A(IM)*SINX                                            00005800
59.           A(IM)=A(IL)*SINX+A(IM)*COSX                                        00005900
60.           A(IL)=X                                                            00006000
61.       115 IF(MV-1)120,125,120                                                00006100
62.       120 ILR=ILQ+I                                                          00006200
63.           IMR=IMQ+I                                                          00006300
64.           X=R(ILR)*COSX-R(IMR)*SINX                                          00006400
65.           R(IMR)=R(ILR)*SINX+R(IMR)*COSX                                     00006500
66.           R(ILR)=X                                                           00006600
67.       125 CONTINUE                                                           00006700
68.           X=2.0*A(LM)*SINCS                                                  00006800
69.           Y=A(LL)*COSX2+A(MM)*SINX2-X                                        00006900
70.           X=A(LL)*SINX2+A(MM)*COSX2+X                                        00007000
71.           A(LM)=(A(LL)-A(MM))*SINCS+A(LM)*(COSX2-SINX2)                      00007100
72.           A(LL)=Y                                                            00007200
73.           A(MM)=X                                                            00007300
74.       130 IF(M-N)135,140,135                                                 00007400
75.       135 M=M+1                                                              00007500
76.           GO TO 60                                                           00007600
77.       140 IF(L-(N-1))145,150,145                                             00007700
78.       145 L=L+1                                                              00007800
```

```
      GO TO 55                                                          00007900
  150 IF(IND-1)160,155,160                                              00008000
  155 IND=0                                                             00008100
      GO TO 50                                                          00008200
  160 IF(THR-ANRMX)165,165,45                                           00008300
  165 IQ=-N                                                             00008400
      DO 185 I=1,N                                                      00008500
      IQ=IQ+N                                                           00008600
      LL=I+(I*I-I)/2                                                    00008700
      JQ=N*(I-2)                                                        00008800
      DO 185 J=I,N                                                      00008900
      JQ=JQ+N                                                           00009000
      MM=J+(J*J-J)/2                                                    00009100
      IF(A(LL)-A(MM))170,185,185                                        00009200
  170 X=A(LL)                                                           00009300
      A(LL)=A(MM)                                                       00009400
      A(MM)=X                                                           00009500
      IF(MV-1)175,185,175                                               00009600
  175 DO 180 K=1,N                                                      00009700
      ILR=IQ+K                                                          00009800
      IMR=JQ+K                                                          00009900
      X=R(ILR)                                                          00010000
      R(ILR)=R(IMR)                                                     00010100
  180 R(IMR)=X                                                          00010200
  185 CONTINUE                                                          00010300
      RETURN                                                            00010400
      END                                                               00010500
```

```
      SUBROUTINE BGEOM(NSECT)                                          00000100
C                                                                      00000200
C                                                                      00000300
C     PROGRAM PROVIDES BEAM CROSS-SECTION GEOMETRY                     00000400
C      AND PROPERTIES                                                  00000500
C                                                                      00000600
C                                                                      00000700
      COMMON/XSECT/NSEG(10),Y(20,10),Z(20,10),XLEN(20,10),T(20,10),      00
     *TORCON(10),KLOS(10),ABM(10),FIXX(10),FIYY(10),FIZZ(10)          00000900
      WRITE(6,200)                                                     00001000
  200 FORMAT(////30X,'BEAM CROSS-SECTION DATA'/)                       00001100
          DO 10 N=1,NSECT                                              00001200
         READ(5,100) NSEG(N),KLOS(N),FIYY(N),FIZZ(N)                   00001300
       FIXX(N)=FIYY(N)+FIZZ(N)                                         00001400
      TORCON(N)=0.0                                                    00001500
        K=NSEG(N)                                                      00001600
      IF(KLOS(N).EQ.1)K=K+1                                            00001700
        READ(5,110)(Y(I,N),Z(I,N),T(I,N),I=1,K)                        00001800
        ABM(N)=0.                                                      00001900
       DO 20 I=1,K                                                     00002000
       IF(I.EQ.K) GO TO 30                                             00002100
        DY=Y(I+1,N)-Y(I,N)                                             00002200
       DZ=Z(I+1,N)-Z(I,N)                                              00002300
        GO TO 40                                                       00002400
   30  CONTINUE                                                        00002500
       DY=Y(1,N)-Y(I,N)                                                00002600
       DZ=Z(1,N)-Z(I,N)                                                00002700
   40  CONTINUE                                                        00002800
      XLEN(I,N)=SQRT(DZ*DZ+DY*DY)                                      00002900
        ABM(N)=ABM(N)+XLEN(I,N)*T(I,N)                                 00003000
   20  CONTINUE                                                        00003100
      IF(KLOS(N).EQ.1)GO TO 50                                         00003200
      TORCON(N)=2.*T(1,N)*T(2,N)*((XLEN(1,N)-T(1,N))**2)               00003300
     C*((XLEN(2,N)-T(2,N))**2)/(XLEN(1,N)*T(1,N)+XLEN(2,N)*T(2,N)      00003400
     C-T(1,N)*T(1,N)-T(2,N)*T(2,N))                                    00003500
       GO TO 10                                                        00003600
   50 DO 60 I=1,K                                                      00003700
   60 TORCON(N)=TORCON(N)+XLEN(I,N)*T(I,N)**3/3.                       00003800
   10 CONTINUE                                                         00003900
```

```
      DO 130 N=1,NSECT                                                  00004000
      WRITE(6,210) N,NSEG(N),KLOS(N),ABM(N),FIXX(N),FIYY(N),FIZZ(N),      00
     1TORCON(N)                                                         00004200
  210 FORMAT(/  10X,3H  N,3X,4HNSEG,3X,4HKLOS,8X,4H ABM,8X,4HFIXX,      00004300
     18X,4HFIYY,8X,4HFIZZ,6X,6HTORCON/6X,3I7,5E12.5/ )                  00004400
      WRITE(6,220)                                                      00004500
  220 FORMAT(/39X,1HI,9X,1HY,9X,1HZ,6X,4HXLEN,9X,1HT)                   00004600
       K=NSEG(N)+KLOS(N)                                                00004700
      DO 120 I=1,K                                                      00004800
      WRITE(6,230) I,Y(I,N),Z(I,N),XLEN(I,N),T(I,N)                     00004900
  230 FORMAT( 38X,I2,4E10.2)                                            00005000
  120 CONTINUE                                                          00005100
  130 CONTINUE                                                          00005200
  100 FORMAT(2I5,2E10.4)                                                00005300
  110  FORMAT(3E10.4)                                                   00005400
        RETURN                                                          00005500
        END                                                             00005600
```

```
      SUBROUTINE BASME (I,NDGREE,IX,XC,YC,ZC,E,SMASS,DICOS,RMASS,PMASS,      00
     *I1,STRS,NEND)                                                    00000200
C       CALCULATES THE MASS MATRIX OF A BEAM ELEMENT THREE DIMENSIONAL  00000300
       COMMON/ANG/ ANGO(150),HT(150),AL(150)                            00000400
       COMMON/XSECT/NSEG(10),Y(20,10),Z(20,10),XLEN(20,10),T(20,10),    00
     *TORCON(10),KLOS(10),ABM(10),FIXX(10),FIYY(10),FIZZ(10)           00000600
       DIMENSION XC(1),YC(1),ZC(1),E(16,1),IX(09,1),SMASS(1)            00000700
       DIMENSION DICOS(3,3,1),PMASS(4,1),RMASS(3,3,1)                   00000800
       DIMENSION XPER(6),SK(6),SM(6)                                    00000900
       DIMENSION TEMP(3,3)                                              00001000
       DIMENSION STRS(1)                                                00001100
       N1 = IX(1,I)                                                     00001200
       N2 = IX(2,I)                                                     00001300
       MTYP=IX(9,I)                                                     00001400
C                                                                       00001500
C                                                                       00001600
C      INITIALIZE  STRS ARRAY                                           00001700
C                                                                       00001800
C                                                                       00001900
       KBM=IX(6,I)                                                      00002000
        NPT=NSEG(KBM)+KLOS(KBM)                                         00002100
       NEND =8*NPT                                                      00002200
        DO 10 J=1,NEND                                                  00002300
   10 STRS(J)=0.                                                        00002400
       NT=2*NPT                                                         00002500
       DO 11 J=1,NT                                                     00002600
        K=(J-1)*4+3                                                     00002700
   11  STRS(K)=E(3,MTYP)                                                00002800
       XX=XC(N2)-XC(N1)                                                 00002900
       YY = YC(N2) - YC(N1)                                             00003000
       ZZ = ZC(N2) - ZC(N1)                                             00003100
       AL(I) = SQRT(XX*XX+YY*YY+ZZ*ZZ)                                  00003200
       DICOS(1,1,I) = XX/AL(I)                                          00003300
       DICOS(2,1,I) = YY/AL(I)                                          00003400
       DICOS(3,1,I) = ZZ/AL(I)                                          00003500
       LOCNOD=IX(7,I)                                                   00003600
       VAX=XC(LOCNOD) - XC(N1)                                          00003700
       VAY=YC(LOCNOD) - YC(N1)                                          00003800
       VAZ=ZC(LOCNOD) - ZC(N1)                                          00003900
```

```
40.            A=SQRT(VAX*VAX+VAY*VAY+VAZ*VAZ)                                  00004000
41.            AX=VAX/A                                                         00004100
42.            AY=VAY/A                                                         00004200
43.            AZ=VAZ/A                                                         00004300
44.            E3X=AZ*DICOS(2,1,I)-AY*DICOS(3,1,I)                              00004400
45.            E3Y=AX*DICOS(3,1,I)-AZ*DICOS(1,1,I)                              00004500
46.            E3Z=AY*DICOS(1,1,I)-AX*DICOS(2,1,I)                              00004600
47.            ANORM=SQRT(E3X*E3X+E3Y*E3Y+E3Z*E3Z)                              00004700
48.            DICOS(1,3,I)=E3X/ANORM                                           00004800
49.            DICOS(2,3,I)=E3Y/ANORM                                           00004900
50.            DICOS(3,3,I)=E3Z/ANORM                                           00005000
51.      C      ESTABLISH LOCAL-GLOBAL DIRECTION COSINE MATRIX                   00005100
52.            DICOS(1,2,I)=DICOS(2,3,I)*DICOS(3,1,I)-DICOS(3,3,I)*DICOS(2,1,I) 00005200
53.            DICOS(2,2,I)=DICOS(3,3,I)*DICOS(1,1,I)-DICOS(1,3,I)*DICOS(3,1,I) 00005300
54.            DICOS(3,2,I)=DICOS(1,3,I)*DICOS(2,1,I)-DICOS(2,3,I)*DICOS(1,1,I) 00005400
55.      C      ASSEMBLE TRANSLATIONAL MASS                                      00005500
56.             EMASS=E(1,MTYP)*ABM(KBM)*AL(I)                                  00005600
57.              RT=.5*EMASS                                                    00005700
58.             RX=.5*E(1,MTYP)*AL(I)*FIXX(KBM)                                 00005800
59.             RY=EMASS*AL(I)*AL(I)/24.+.5*E(1,MTYP)*AL(I)*FIYY(KBM)           00005900
60.            RZ=EMASS*AL(I)*AL(I)/24.+.5*E(1,MTYP)*AL(I)*FIZZ(KBM)            00006000
61.      C********** COMPUTE VIBRATION PERIOD FOR BEAMS **********               00006100
62.            NCRO=IX(6,I)                                                     00006200
63.            SK(1)=ABM(NCRO)*E(2,MTYP)/AL(I)                                  00006300
64.             SK(2)=TORCON(NCRO)*E(2,MTYP)/(2.*AL(I)*(1.+E(7,MTYP)))          00006400
65.            SK(3)=4.0*E(2,MTYP)*FIZZ(NCRO)/AL(I)                             00006500
66.            SK(4)=3.0*SK(3)/AL(I)**2                                         00006600
67.             SK(5)=SK(3)*FIYY(NCRO)/FIZZ(NCRO)                               00006700
68.             SK(6)=SK(4)*FIYY(NCRO)/FIZZ(NCRO)                               00006800
69.             SM(1)=RT                                                        00006900
70.             SM(2)=RX                                                        00007000
71.             SM(3)=RZ                                                        00007100
72.             SM(4)=RT                                                        00007200
73.              SM(5)=RY                                                       00007300
74.              SM(6)=RT                                                       00007400
75.            PI=3.1416                                                        00007500
76.            DO 20 K=1,6                                                      00007600
77.            XPER(K)=SQRT(SM(K)/SK(K))*2.0*PI                                 00007700
78.         20 CONTINUE                                                         00007800
```

```
      WRITE(6,30) I,(XPER(J),J=1,6)                                          00007900
   30 FORMAT(6X,I5,3X,6E10.3)                                                00008000
      DO 4  N = 3,4                                                          00008100
      KK=IX(N,I)                                                             00008200
      NN=(KK-1)*NDGREE                                                       00008300
      DO 5  M = 1,3                                                          00008400
    5 SMASS(NN+M) = RT + SMASS(NN+M)                                         00008500
    4 CONTINUE                                                               00008600
C       ASSEMBLE ROTATIONAL MASS IN GLOBAL COORDINATES                       00008700
        DO 50 II=1,3                                                         00008800
        DO 50 JJ=1,3                                                         00008900
        TEMP(II,JJ)=DICOS(II,1,I)*DICOS(JJ,1,I)*RX+                          00009000
     * DICOS(II,2,I)*DICOS(JJ,2,I)*RY+DICOS(II,3,I)*DICOS(JJ,3,I)*RZ           00
   50 CONTINUE                                                               00009200
        DO 60 M=3,4                                                          00009300
        N=IX(M,I)                                                            00009400
         DO 70 II=1,3                                                        00009500
        DO 70 JJ=1,3                                                         00009600
   70 RMASS(II,JJ,N)=RMASS(II,JJ,N)+TEMP(II,JJ)                              00009700
   60  CONTINUE                                                              00009800
      RETURN                                                                 00009900
      END                                                                    00010000
```

```
      SUBROUTINE TASME(JE,NDGREE,IX,XC,YC,ZC,E,SMASS,SIDE,IP,STRS,          00
     *NTOT,I1,DICOS,RMASS)                                                00000200
      DIMENSION XC(1),YC(1),ZC(1),E(16,1),IX(9,1),SMASS(1),SIDE(1),IP(1)00000300
      DIMENSION X(3),Y(3),Z(3),XXI(3),YYI(3),ZZI(3)                        00000400
       DIMENSION VA(3),VB(3)                                               00000500
       DIMENSION DICOS(3,3),RMASS(3,3,1)                                   00000600
       DIMENSION STRS(1)                                                   00000700
        DATA LOC/3/,NDAT/7/                                                00000800
      DO 105 I=1,3                                                         00000900
      J=IP(I+1)                                                            00001000
      K=IX(I,JE)                                                           00001100
      L=IX(J,JE)                                                           00001200
      X(I)=XC(L)-XC(K)                                                     00001300
      Y(I)=YC(L)-YC(K)                                                     00001400
      Z(I)=ZC(L)-ZC(K)                                                     00001500
      XXI(I)=0.0                                                           00001600
      YYI(I)=0.0                                                           00001700
      ZZI(I)=0.0                                                           00001800
  105 SIDE(I)=SQRT(X(I)**2+Y(I)**2+Z(I)**2)                                00001900
      AREA=0.5*(X(1)*(Y(2)-Z(2))+Y(1)*(Z(2)-X(2))+Z(1)*(X(2)-Y(2)))        00002000
      A=AREA/6.                                                            00002100
      IF(AREA.LE.0.) GO TO 115                                             00002200
       MTYP=IX(9,JE)                                                       00002300
C                                                                          00002400
C                                                                          00002500
C      INITIALIZE STRS STORAGE                                             00002600
C                                                                          00002700
C                                                                          00002800
       NTOT=LOC*NDAT*I1                                                    00002900
      DO 200 I=1,NTOT                                                      00003000
  200 STRS(I)=0.                                                           00003100
       N=LOC*I1                                                            00003200
        DO 210 I=1,N                                                       00003300
      J=NDAT*I                                                             00003400
  210 STRS(J)=E(3,MTYP)                                                    00003500
C                                                                          00003600
C                                                                          00003700
C        CALCULATE ELEMENT MASS                                            00003800
C                                                                          00003900
```

```
      C                                                                 00004000
            EMASS=AREA*E(1,MTYP)*E(5,MTYP)/3.                           00004100
            RM=E(1,MTYP)*E(5,MTYP)/12.                                  00004200
            DO 110 N=1,3                                                00004300
            J=(IX(N,JE)-1)*NDGREE                                       00004400
             DO 110 I=1,3                                               00004500
        110 SMASS(I+J)=SMASS(I+J)+EMASS                                 00004600
            N1=IX(1,JE)                                                 00004700
            N2=IX(2,JE)                                                 00004800
            N3=IX(3,JE)                                                 00004900
            XCEN=(XC(N1)+XC(N2)+XC(N3))/3.                              00005000
            YCEN=(YC(N1)+YC(N2)+YC(N3))/3.                              00005100
            DO 120 I=1,3                                                00005200
            J=IP(I+1)                                                   00005300
            M=IP(I+2)                                                   00005400
            K=IX(I,JE)                                                  00005500
            L=IX(J,JE)                                                  00005600
            N=IX(M,JE)                                                  00005700
            XBAR=XCEN-XC(K)                                             00005800
            YBAR=YCEN-YC(K)                                             00005900
            XCORD=0.5*(XC(L)-XC(K))                                     00006000
            YCORD=0.5*(YC(L)-YC(K))                                     00006100
            XKORD=0.5*(XC(N)-XC(K))                                     00006200
            YKORD=0.5*(YC(N)-YC(K))                                     00006300
            CALL QUAD(0.,0.,XCORD,YCORD,XBAR,YBAR,A,XXI(I),YYI(I))      00006400
            CALL QUAD(XBAR,YBAR,XKORD,YKORD,0.,0.,A,XXI(I),YYI(I))      00006500
        120 CONTINUE                                                    00006600
            DO 125 J=1,3                                                00006700
            XXI(J)=E(1,MTYP)*E(5,MTYP)*XXI(J)                           00006800
            YYI(J)=E(1,MTYP)*E(5,MTYP)*YYI(J)                           00006900
        125 ZZI(J)=XXI(J)+YYI(J)                                        00007000
      C                                                                 00007100
      C                                                                 00007200
      C     CONSTRUCT INITIAL ELEMENT NORMAL                            00007300
      C                                                                 00007400
      C                                                                 00007500
             VA(1)=X(1)                                                 00007600
             VA(2)=Y(1)                                                 00007700
             VA(3)=Z(1)                                                 00007800
```

```
      VB(1)=-X(3)                                                          00007900
      VB(2)=-Y(3)                                                          00008000
      VB(3)=-Z(3)                                                          00008100
      CALL CROSS(VA,VB,DICOS(1,3),ANG,PMAG,0)                              00008200
      PMAG=0.                                                              00008300
      DO 140 I=1,3                                                         00008400
  140 PMAG=PMAG+((VA(I)-VB(I))**2)                                         00008500
       PMAG=SQRT(PMAG)                                                     00008600
      DO 145 I=1,3                                                         00008700
  145 DICOS(I,1)=(VA(I)-VB(I))/PMAG                                        00008800
       CALL CROSS(DICOS(1,3),DICOS(1,1),DICOS(1,2),ANG,PMAG,0)             00008900
      DO 160 II=1,3                                                        00009000
       N=IX(II,JE)                                                         00009100
      DO 160 I=1,3                                                         00009200
      DO 160 J=1,3                                                         00009300
      RMASS(I,J,N)=RMASS(I,J,N)+DICOS(I,1)*DICOS(J,1)*XXI(II)              00009400
     **DICOS(I,2)*DICOS(J,2)*YYI(II)+DICOS(I,3)*DICOS(J,3)*ZZI(II)         00009500
  160 CONTINUE                                                             00009600
      RETURN                                                               00009700
  115 WRITE(6,1000) AREA,JE                                                00009800
 1000 FORMAT(1H0,5X,'AREA=',E10.4,'OF ELEMENT',I5)                         00009900
      STOP                                                                 00010000
      END                                                                  00010100
```

```
      SUBROUTINE FRCIN(NUMEL,NUMNP,NDGREE,IX,E,XO,FINT,OLDS)          00000100
     COMMON/DYNAM/DELT,TIMEND,MXSTEP,NTSTEP                          00000200
      COMMON/JUNK/TL(9,300),POINT(3,3,300)                           00000300
     COMMON/STR/I1,INDEX(300),STRAIN(45),STRESS(45),SIDE(900),IP(6),  00000400
    *STRS(10000)                                                     00000500
      COMMON /BEAMS/RMASS(3,3,100),DICOS(3,3,150),BAR(1200),         00000600
    *PMASS(4,100),NSECT                                              00000700
     COMMON XC(300),YC(300),ZC(300),UD(900),UD1(900),UD2(900)        00000800
     DIMENSION IX(9,1),E(16,1),XO(1),B1(3),B3(3),TEMP1(3),TEMP2(3)   00000900
     DIMENSION TB1(3),TB3(3)                                         00001000
         DIMENSION FINT(1)                                           00001100
        DIMENSION OLDS(1)                                            00001200
      DOUBLE PRECISION B1,B3,B31S,B32S,XL                            00001300
     MEQ=NDGREE*NUMNP                                                00001400
     DO 15 I=1,MEQ                                                   00001500
  15  FINT(I)=0.                                                     00001600
     C1=DELT*DELT/2.                                                 00001700
C                                                                    00001800
C CALCULATE B3 VECTOR IN RIGID BODY COORDINATES                      00001900
C                                                                    00002000
      NRN=NUMNP                                                      00002100
     DO 10 I=1,NRN                                                   00002200
        M=NDGREE*I                                                   00002300
     B3(1)=DELT*UD1(M-1)+C1*(UD1(M-2)*UD1(M)+UD2(M-1))               00002400
     B3(2)=-DELT*UD1(M-2)+C1*(UD1(M-1)*UD1(M)-UD2(M-2))              00002500
     B1(2)=DELT*UD1(M)+C1*(UD1(M-2)*UD1(M-1)+UD2(M))                 00002600
     B31S=B3(1)**2                                                   00002700
     B32S=B3(2)**2                                                   00002800
     B3(3)=-0.5*(B31S+B32S)                                          00002900
     XL=DSQRT(B31S+B32S+(1.+B3(3))**2)                               00003000
     B3(1)=B3(1)/XL                                                  00003100
     B3(2)=B3(2)/XL                                                  00003200
     B3(3)=(1.+B3(3))/XL                                             00003300
     B1(3)=(-B3(1)-B1(2)*B3(2))/B3(3)                                00003400
     B1(1)=DSQRT(1.-B1(2)**2-B1(3)**2)                               00003500
C                                                                    00003600
C TRANSFORM VECTOR FROM RIGID BODY TO GLOBAL COORDINATES             00003700
C                                                                    00003800
      DO 200 KK=1,3                                                  00003900
```

```
      TB1(KK)=B1(KK)                                                    00004000
  200 TB3(KK)=B3(KK)                                                    00004100
      CALL GMPRD(TL(1,I),TB3,TEMP1,3,3,1)                               00004200
      CALL GMPRD(TL(1,I),TB1,TEMP2,3,3,1)                               00004300
C                                                                       00004400
C UPDATE LAMBDA TRANSFORMATION MATRIX                                   00004500
C                                                                       00004600
      TL(1,I)=TEMP2(1)                                                  00004700
      TL(2,I)=TEMP2(2)                                                  00004800
      TL(3,I)=TEMP2(3)                                                  00004900
      CALL CROSS(TEMP1,TEMP2,TL(4,I),BETA,CMAG,1)                       00005000
      TL(7,I)=TEMP1(1)                                                  00005100
      TL(8,I)=TEMP1(2)                                                  00005200
      TL(9,I)=TEMP1(3)                                                  00005300
   10 CONTINUE                                                          00005400
      MJ=0                                                              00005500
      DO 20 JE=1,NUMEL                                                  00005600
      MTYP = IX(9,JE)                                                   00005700
      KEL=IX(8,JE)                                                      00005800
      IND = INDEX(JE)                                                   00005900
      IF(KEL.GT.1) GO TO 100                                            00006000
      CALL TFRCIN(JE,NDGREE ,XC,YC,ZC,IX,E(1,MTYP),STRS(IND),STRESS,    00006100
     *STRAIN,XO,FINT,IP,I1,SIDE,OLDS)                                   00006200
      GO TO 110                                                         00006300
  100 CONTINUE                                                          00006400
      MJ=MJ+1                                                           00006500
      I=IX(3,JE)                                                        00006600
      J=IX(4,JE)                                                        00006700
      M=(MJ-1)*24+1                                                     00006800
      M1=M+12                                                           00006900
      CALL BFRCIN(JE,NDGREE,XC,YC,ZC,IX,E,XO,FINT,STRS,STRAIN,STRESS,       00
     *I1,INDEX,SMASS,TL(1,I),TL(1,J),BAR(M),BAR(M1))                    00007100
  110 CONTINUE                                                          00007200
   20 CONTINUE                                                          00007300
      RETURN                                                            00007400
      END                                                               00007500
```

```
      SUBROUTINE TFRCIN(JE,ND,XC,YC,ZC,NODE,E,STRS,STRESS,STRAIN,UD,
     *FINT,IP,IT,SIDE,OLDS)
       COMMON/JUNK/TL(9,300),POINT(3,3,300)
      COMMON/DYNAM/DELT
      DIMENSION XC(1),YC(1),ZC(1),NODE(9,1),E(1),STRS(1),STRESS(1),
     *STRAIN(1),UD(1),IP(1),SIDE(1)
      DIMENSION FORCE(18),X(3),Y(3),Z(3),UX(3),UY(3),UZ(3),D(3),EPS(3),
     1SF(3),T(3)
       DIMENSION STRNM(3),XROT(3),YROT(3),FZ(3),SMX(3),SMY(3)
      DIMENSION CURV(3,3),EF(3,3),EM(3,3)
         DIMENSION FINT(1)
          DIMENSION OLDS(1)
       DIMENSION TEMP(3)
C     ROTATIONS AT EACH NODE
      CALL VECTOR(XC,YC,ZC,UD,NODE,JE,D,S,IP,A,B,C,ABC)
      C1=1.0/(D(1)*D(3))
      CX=SQRT(S*(S-D(2))*C1)
      CY=-SQRT((S-D(1))*(S-D(3))*C1)
      AREA=-D(1)*D(3)*CX*CY
      THETA=ACOS(CX)
      CALL VECT(THETA,CXX,CYX,CZX)
      THETA=1.570796+THETA
      CALL VECT(THETA,CXY,CYY,CZY)
      DO 100 I=1,3
       NOD=NODE(I,JE)
        K=NOD
       CALL GMPRD(TL(1,K),POINT(1,I,K),T,3,3,1)
      XROT(I)=+CXY*T(1)+CYY*T(2)+CZY*T(3)
  100 YROT(I)=-CXX*T(1)-CYX*T(2)-CZX*T(3)
C
C     STRAIN OF ELEMENT SIDES
      DO 105 I=1,3
      J=IP(I+1)
      I1=NODE(I,JE)
      I2=NODE(J,JE)
      X(I)=XC(I2)-XC(I1)
      Y(I)=YC(I2)-YC(I1)
      Z(I)=ZC(I2)-ZC(I1)
       L1=ND*(I1-1)
```

```
      L2=ND*(I2-1)                                                          00004000
        UX(I)=UD(L2+1)-UD(L1+1)                                             00004100
        UY(I)=UD(L2+2)-UD(L1+2)                                             00004200
  105   UZ(I)=UD(L2+3)-UD(L1+3)                                             00004300
      DO 110 I=1,3                                                          00004400
      DS=SIDE(3*JE+I-3)                                                     00004500
  110 EPS(I)=(2.*(X(I)*UX(I)+Y(I)*UY(I)+Z(I)*UZ(I))+UX(I)**2+UY(I)**2       00004600
     1+UZ(I)**2)/((D(I)+DS)*DS)                                             00004700
C     MEMBRANE STRAIN                                                       00004800
      D1=C1/(4.*CX*CX)                                                      00004900
      D2=C1/(4.*CY*CY)                                                      00005000
      D3=0.5/(CX*CY)                                                        00005100
      STRNM(1)=(D(1)*(D(1)+D(3))*EPS(1)-D(2)**2*EPS(2)                      00005200
     1+D(3)*(D(1)+D(3))*EPS(3))*D1                                          00005300
      STRNM(2)=(D(1)*(D(3)-D(1))*EPS(1)+D(2)**2*EPS(2)                      00005400
     1+D(3)*(D(1)-D(3))*EPS(3))*D2                                          00005500
      STRNM(3)=(EPS(1)-EPS(3))*D3                                           00005600
C     CURVATURES                                                            00005700
      CALL CURVAT(CURV,AREA,XROT,YROT,D,CX,CY)                              00005800
C     STRAINS AND STRESSES ACROSS THICKNESS                                 00005900
      DO 115 M = 1,3                                                        00006000
      IND=7*IT*(M-1)+1                                                      00006100
      H = E(5)                                                              00006200
      CALL GENFOR (STRNM,CURV(1,M),EM(1,M),EF(1,M),E,H,IT,STRS(IND))        00006300
  115 CONTINUE                                                              00006400
      SF1=0.                                                                00006500
      SF2=0.                                                                00006600
      SF3=0.                                                                00006700
      DO 135 I=1,3                                                          00006800
      SF1=SF1+((D(1)+D(3))*D1*EF(1,I)-(D(1)-D(3))*D2*EF(2,I)                00006900
     1+D3/D(1)*EF(3,I))/D(1)                                                00007000
      SF2=SF2+(-D1*EF(1,I)+D2*EF(2,I))                                      00007100
  135 SF3=SF3+((D(1)+D(3))*D1*EF(1,I)+(D(1)-D(3))*D2*EF(2,I)                00007200
     1-D3/D(3)*EF(3,I))/D(3)                                                00007300
      CONST=AREA/3.                                                         00007400
      SF(1)=CONST*SF1                                                       00007500
      SF(2)=CONST*SF2                                                       00007600
      SF(3)=CONST*SF3                                                       00007700
C                                                                           00007800
```

```
      C ADDITION OF MEMBRANE DAMPING (VISCOUS) FORCES                          00007900
      C                                                                        00008000
            ID=3*(JE-1)                                                        00008100
            DO 150 I=1,3                                                       00008200
            EPSS=OLDS(ID+I)                                                    00008300
            DEPS=EPS(I)-EPSS                                                   00008400
            OLDS(ID+I)=EPS(I)                                                  00008500
            SF(I)=SF(I)+2.*E(8)*AREA*(DEPS/DELT)*SQRT(E(1)*E(2))*E(5)          00008600
        150 CONTINUE                                                           00008700
      C     NODAL FORCES AND MOMENTS                                           00008800
            CALL NODALF(EM,SMX,SMY,FZ,AREA)                                    00008900
            DO 140 I=1,3                                                       00009000
            J=IP(I+2)                                                          00009100
              K=ND*(I-1)                                                       00009200
            N=NODE(I,JE)                                                       00009300
            FORCE(K+1)=-(X(I)+UX(I))*SF(I)+(X(J)+UX(J))*SF(J)+A/ABC*FZ(I)      00009400
            FORCE(K+2)=-(Y(I)+UY(I))*SF(I)+(Y(J)+UY(J))*SF(J)+B/ABC*FZ(I)      00009500
            FORCE(K+3)=-(Z(I)+UZ(I))*SF(I)+(Z(J)+UZ(J))*SF(J)+C/ABC*FZ(I)      00009600
             FORCE(K+4)=CXX*SMX(I)+CXY*SMY(I)                                  00009700
             FORCE(K+5)=CYX*SMX(I)+CYY*SMY(I)                                  00009800
             FORCE(K+6)=CZX*SMX(I)+CZY*SMY(I)                                  00009900
             CALL GTPRD(TL(1,N),FORCE(K+4),TEMP,3,3,1)                         00010000
             DO 145 KK=1,3                                                     00010100
             LOC=ND*(N-1)+KK                                                   00010200
              FINT(LOC)=FINT(LOC)+FORCE(K+KK)                                  00010300
              FINT(LOC+3)=FINT(LOC+3 )+TEMP(KK)                                00010400
        145  CONTINUE                                                          00010500
        140 CONTINUE                                                           00010600
            RETURN                                                             00010700
            END                                                                00010800
```

```
      SUBROUTINE BFRCIN (L,ND,XC,YC,ZC,IX,E,UD,FINT,STRS,STRAIN,STRESS,
     1   I1,INDEX,SMASS,EUI,EUJ,TRAI,TRAJ)
C     ROUTINE CALCULATES INTERNAL FORCES FOR THREE-D BEAM
C    TRAI AND TRAJ CONTAIN I BAR 0 ,J BAR 0 AND J HAT 0 FOR NODES I + J
      DIMENSION FINT(1)
      COMMON/ANG/ ANGO(150),HT(150),AL(150)
      DIMENSION XC(1),YC(1),IX(09,1),UD(1),E(16,1)
     1   ,STRAIN(1),SMASS(1),STRESS(1),STRS(1),INDEX(1)
     2  ,ZC(1),EUI(1),EUJ(1),EH(9),TRAI(1),TRAJ(1)
     3 ,EJBI(3),EJBJ(3)
     4  ,TEMP1(3),TEMP2(3),TEMP3(3),TEMP4(3),TEMP5(3),OMEGI(3,3),
     5  OMEGJ(3,3),DISP(2),ROT(6),AFORCE(6),BMOMT(6)
      I2=I1*2
      N1 = IX(1,L)
      N2 = IX(2,L)
      N3=IX(3,L)
      N4=IX(4,L)
      XX=XC(N2)-XC(N1)
      YY= YC(N2)-YC(N1)
      ZZ = ZC(N2)-ZC(N1)
      N1N=(N1-1)*ND
      N2N=(N2-1)*ND
      N3N=(N3-1)*ND
      N4N=(N4-1)*ND
      II = N1
      JJ = N2
      OMEGI(1,1)=0.
      OMEGI(1,2)=TRAI(12)
      OMEGI(1,3)=-TRAI(11)
      OMEGI(2,1)=-TRAI(12)
      OMEGI(2,2)=0.
      OMEGI(2,3)=TRAI(10)
      OMEGI(3,1)=TRAI(11)
      OMEGI(3,2)=-TRAI(10)
      OMEGI(3,3)=0.
      OMEGJ(1,1)=0.
      OMEGJ(1,2)=TRAJ(12)
      OMEGJ(1,3)=-TRAJ(11)
      OMEGJ(2,1)=-TRAJ(12)
```

```
      OMEGJ(2,2)=0.                                                          00004000
      OMEGJ(2,3)=TRAJ(10)                                                    00004100
      OMEGJ(3,1)=TRAJ(11)                                                    00004200
      OMEGJ(3,2)=-TRAJ(10)                                                   00004300
      OMEGJ(3,3)=0.                                                          00004400
      TEMP1(1)=TRAI(10)                                                      00004500
      TEMP1(2)=TRAI(11)                                                      00004600
      TEMP1(3)=TRAI(12)                                                      00004700
      TEMP2(1)=TRAJ(10)                                                      00004800
      TEMP2(2)=TRAJ(11)                                                      00004900
      TEMP2(3)=TRAJ(12)                                                      00005000
      CALL GMPRD(EUI,TEMP1,TEMP3,3,3,1)                                      00005100
      CALL GMPRD(EUJ,TEMP2,TEMP4,3,3,1)                                      00005200
      DIX=UD(N3N+1)+TEMP3(1)-TRAI(7)                                         00005300
      DJX=UD(N4N+1)+TEMP4(1)-TRAJ(7)                                         00005400
      DIY=UD(N3N+2)+TEMP3(2)-TRAI(8)                                         00005500
      DJY=UD(N4N+2)+TEMP4(2)-TRAJ(8)                                         00005600
      DIZ=UD(N3N+3)+TEMP3(3)-TRAI(9)                                         00005700
      DJZ=UD(N4N+3)+TEMP4(3)-TRAJ(9)                                         00005800
        UD(N1N+1)=DIX                                                        00005900
       UD(N1N+2)=DIY                                                         00006000
       UD(N1N+3)=DIZ                                                         00006100
       UD(N2N+1)=DJX                                                         00006200
       UD(N2N+2)=DJY                                                         00006300
       UD(N2N+3)=DJZ                                                         00006400
C                                                                            00006500
C      FIND RIDID BODY ROTATION AFTER DEFORMATION W/T ORIGINAL COORDINAT      00006600
C      FIND STRAINS AT NODES, HT IS THICKNESS OF BEAM ELEMENT                00006700
       DIJX=DJX-DIX                                                          00006800
       DIJY=DJY-DIY                                                          00006900
      DIJZ = DJZ - DIZ                                                       00007000
       DX=XX+DIJX                                                            00007100
       DY=YY+DIJY                                                            00007200
      DZ = ZZ + DIJZ                                                         00007300
      AL2 = AL(L)*AL(L)                                                      00007400
C     FAC = CENTROIDAL AXIS STRAIN                                           00007500
      FAC = 2.*(XX*DIJX+YY*DIJY+ZZ*DIJZ)+DIJX*DIJX+DIJY*DIJY+DIJZ*DIJZ       00007600
       ALN2=AL2+FAC                                                          00007700
       ALN=SQRT(ALN2)                                                        00007800
```

```
      EH(1) =   DX/ALN                                                  00007900
      EH(2) =   DY/ALN                                                  00008000
      EH(3) =   DZ/ALN                                                  00008100
      DO 50 I=1,3                                                       00008200
      I4=3 + I                                                          00008300
      I3=6 + I                                                          00008400
      EJBI(I)=EUI(I)*TRAI(4)+EUI(I4)*TRAI(5)+EUI(I3)*TRAI(6)            00008500
   50 EJBJ(I)=EUJ(I)*TRAJ(4)+EUJ(I4)*TRAJ(5)+EUJ(I3)*TRAJ(6)            00008600
      DO 51 I=1,3                                                       00008700
      I4=3 + I                                                          00008800
   51 EH(I4)=(EJBI(I) + EJBJ(I))/2.                                     00008900
       CALL CROSS(EH(1),EH(4),EH(7),DUMA,DUMB,0)                        00009000
       CALL CROSS(EH(7),EH(1),EH(4),DUMA,DUMB,0)                        00009100
C  DETERMINE DEFORMATION ROTATIONS OF BEAM                              00009200
C   EH CONTAINS MU,    EUI + EUJ CONTAIN LAMDA I + LAMDA J             00009300
C  CALCULATE DEFORMATION ROTATIONS LOCAL  COORDINATES(PHIX,PHIZ ECT.)   00009400
       CALL CROSS(EJBI,EJBJ,TEMP3,ROT(1),DUMA,1)                        00009500
       CALL GTPRD(EH,TEMP3,TEMP4,3,3,0)                                 00009600
       ROT(1)=TEMP4(1)                                                  00009700
       CALL GMPRD(EUI,TRAI,TEMP1,3,3,1)                                 00009800
       CALL GTPRD(EH,TEMP1,TEMP2,3,3,1)                                 00009900
       ROT(3)=-TEMP2(3)                                                 00010000
       ROT(5)=TEMP2(2)                                                  00010100
       CALL GMPRD(EUJ,TRAJ,TEMP1,3,3,1)                                 00010200
       CALL GTPRD(EH,TEMP1,TEMP2,3,3,1)                                 00010300
       ROT(4)=-TEMP2(3)                                                 00010400
       ROT(6)=TEMP2(2)                                                  00010500
C      DETERMINE LOCAL INTERNAL FORCES                                  00010600
C                                                                       00010700
      STREH  = FAC/(AL(L)+ALN)/AL(L)                                    00010800
      MTYP = IX(9,L)                                                    00010900
      DISP(1)=ALN                                                       00011000
      DISP(2)=STREH                                                     00011100
       NOPT=3                                                           00011200
      CALL LOCFRC(L,DISP,ROT,AFORCE,BMOMT,E,INDEX,I1,STRAIN,            00011300
     *STRESS,STRS,SMASS,IX,AL(L),MTYP,NOPT,NUMEL)                       00011400
C      ACCUMULATE INTERNAL ELEMENTAL FORCES TO  FINT                    00011500
C      COMPUTE GLOBAL COMPONENTS OF FINT                                00011600
      TEMP1(1)=AFORCE(1)                                                00011700
```

```
      TEMP1(2)=AFORCE(3)                                                00011800
      TEMP1(3)=AFORCE(5)                                                00011900
      CALL GMPRD(EH,TEMP1,TEMP2,3,3,1)                                  00012000
      TEMP3(1)=AFORCE(2)                                                00012100
      TEMP3(2)=AFORCE(4)                                                00012200
      TEMP3(3)=AFORCE(6)                                                00012300
      CALL GMPRD(EH,TEMP3,TEMP1,3,3,1)                                  00012400
      DO 23 I=1,3                                                       00012500
      FINT(N3N+I)=FINT(N3N+I) + TEMP2(I)                                00012600
   23 FINT(N4N+I)=FINT(N4N+I) + TEMP1(I)                                00012700
      CALL GTPRD(EUI,TEMP2,TEMP3,3,3,1)                                 00012800
      CALL GTPRD(OMEGI,TEMP3,TEMP2,3,3,1)                               00012900
      CALL GTPRD(EUJ,TEMP1,TEMP3,3,3,1)                                 00013000
      CALL GTPRD(OMEGJ,TEMP3,TEMP1,3,3,1)                               00013100
      TEMP4(1)=BMOMT(1)                                                 00013200
      TEMP4(2)=BMOMT(3)                                                 00013300
      TEMP4(3)=BMOMT(5)                                                 00013400
      TEMP5(1)=BMOMT(2)                                                 00013500
      TEMP5(2)=BMOMT(4)                                                 00013600
      TEMP5(3)=BMOMT(6)                                                 00013700
      CALL GMPRD(EH,TEMP4,TEMP3,3,3,1)                                  00013800
      CALL GTPRD(EUI,TEMP3,TEMP4,3,3,1)                                 00013900
      CALL GMPRD(EH,TEMP5,TEMP3,3,3,1)                                  00014000
      CALL GTPRD(EUJ,TEMP3,TEMP5,3,3,1)                                 00014100
      I=0                                                               00014200
      DO 24 J=4,6                                                       00014300
      I=I+1                                                             00014400
      FINT(N3N+J)=FINT(N3N+J) + TEMP4(I) + TEMP2(I)                     00014500
   24 FINT(N4N+J)=FINT(N4N+J) + TEMP5(I) + TEMP1(I)                     00014600
       RETURN                                                           00014700
       END                                                              00014800
```

```
      SUBROUTINE LOCFRC(JE,DISP,ROT,AFORCE,BMOMT,E,INDEX,I1,STRAIN,      00000100
     *STRESS,STRS,SMASS,IX,EL,MTYP,NOPT,NUMEL)                          00000200
      COMMON/DYNAM/DELT,TIMEND,MXSTEP,NTSTEP,TIME,PCDAMP                00000300
      COMMON/XSECT/NSEG(10),Y(20,10),Z(20,10),XLEN(20,10),T(20,10),        00
     *TORCON(10),KLOS(10),ABM(10),FIXX(10),FIYY(10),FIZZ(10)            00000500
C                                                                       00000600
C     THIS ROUTINE CALCULATES THE LOCAL INTERNAL FORCES FOR +           00000700
C          NOPT = 1     3-D AXIAL SPRING ELEMENT                      00000800
C          NOPT = 2     3-D LINEAR ELASTIC BEAM ELEMENT               00000900
C          NOPT = 3   3-D ELASTO-PLASTIC BEAM ELEMENT                 00001000
C                                                                       00001100
      DIMENSION DISP(1),ROT(1),AFORCE(1),BMOMT(1),E(16,1),INDEX(1),        00
     *   STRAIN(1),EPSOLD(150),SMASS(1),IX(09,1),STRESS(1),STRS(1)      00001300
C                                                                       00001400
      IF(TIME.GT.0.0) GO TO 1                                           00001500
      EPSOLD(JE)=0.                                                     00001600
    1 IF(NOPT.EQ.3)GO TO 3                                              00001700
C                                                                       00001800
C     CALCULATE DAMPING FORCE FOR SPRING AND ELASTIC BEAM               00001900
C                                                                       00002000
      FDAMP=0.                                                          00002100
      PCDAMP=E(12,MTYP)                                                 00002200
      IF(PCDAMP)12,12,11                                                00002300
   11 N3=IX(3,JE)                                                       00002400
      N4=IX(4,JE)                                                       00002500
      N3N=(N3-1)*6+1                                                    00002600
      N4N=(N4-1)*6+1                                                    00002700
      TMASS=(SMASS(N3N)+SMASS(N4N))/2.                                  00002800
      DDOT=(DISP(2)-EPSOLD(JE))*DISP(1)/DELT                            00002900
      STIFF=E(2,MTYP)*E(7,MTYP)/EL                                      00003000
      FDAMP=2.*PCDAMP*DDOT*SQRT(TMASS*STIFF)                            00003100
      EPSOLD(JE)=DISP(2)                                                00003200
   12 CONTINUE                                                          00003300
C                                                                       00003400
      IF(NOPT.EQ.1)GO TO 13                                             00003500
      DELSTR=DISP(2)                                                    00003600
      GO TO 17                                                          00003700
C                                                                       00003800
C     SPRING ELEMENT                                                    00003900
```

```
C                                                                       00004000
C     TORC IS TENSION OR COMPRESSION CONTROL FOR SPRING ELEMENTS        00004100
C                                                                       00004200
   13 TORC=E(1,MTYP)                                                    00004300
      SLACK=E(4,MTYP)                                                   00004400
      DELSTR=DISP(2)-SLACK                                              00004500
      IF(TORC)15,17,16                                                  00004600
   15 IF(DELSTR.GE.0.)DELSTR=0.                                         00004700
      GO TO 17                                                          00004800
   16 IF(DELSTR.LE.0.)DELSTR=0.                                         00004900
   17 AFORCE(2)=E(2,MTYP)*DELSTR*E(7,MTYP)+FDAMP                        00005000
      AFORCE(1)=-AFORCE(2)                                              00005100
      GO TO 4                                                           00005200
C                                                                       00005300
C     ELASTO-PLASTIC BEAMS                                              00005400
C                                                                       00005500
    3 CONTINUE                                                          00005600
C                                                                       00005700
C       BEAM-  CROSS-SECTION FORCES                                     00005800
C                                                                       00005900
C      LEFT END                                                         00006000
C                                                                       00006100
       KBM=IX(6,JE)                                                     00006200
       NPT=NSEG(KBM)+KLOS(KBM)                                          00006300
        IND=INDEX(JE)                                                   00006400
       EXT=DISP(2)                                                      00006500
        TWIST=ROT(1)/EL                                                 00006600
       CURVZ=-2.*(2.*ROT(5)+ROT(6))/EL                                  00006700
         CURVY=2.*(2.*ROT(3)+ROT(4))/EL                                 00006800
       CALL GENFBM(JE,E(1,MTYP),STRESS,STRAIN,STRS(IND),IX,EXT,CURVZ,           00
     *CURVY,TWIST,FX,AMZ,AMY,AMX)                                       00007000
       BMOMT(1)=-AMX                                                    00007100
        BMOMT(3)=-AMY                                                   00007200
       BMOMT(5)=-AMZ                                                    00007300
        AFORCE(1)=-FX                                                   00007400
C                                                                       00007500
C      RIGHT END                                                        00007600
C                                                                       00007700
       IND=IND+4*NPT                                                    00007800
```

```
      CURVZ=2.*(ROT(5)+2.*ROT(6))/EL                                          00007900
      CURVY=-2.*(ROT(3)+2.*ROT(4))/EL                                         00008000
     CALL GENFBM(JE,E(1,MTYP),STRESS,STRAIN,STRS(IND),IX,EXT,CURVZ,            00
    *CURVY,TWIST,FX,AMZ,AMY,AMX)                                             00008200
      BMOMT(2)=AMX                                                           00008300
      BMOMT(4)=AMY                                                           00008400
     BMOMT(6)=AMZ                                                            00008500
     AFORCE(2)=FX                                                            00008600
      AFORCE(4)=-(BMOMT(5)+BMOMT(6))/EL                                      00008700
      AFORCE(3)=-AFORCE(4)                                                   00008800
      AFORCE(6)=(BMOMT(3)+BMOMT(4))/EL                                       00008900
      AFORCE(5)=-AFORCE(6)                                                   00009000
   4 CONTINUE                                                               00009100
  10 RETURN                                                                 00009200
     END                                                                    00009300
```

```
      SUBROUTINE GENFBM(JE,E,STRESS,STRAIN,STRS,IX,EXT,CURVZ ,CURVY,          00
     *ROT,FX,AMZ,AMY,AMX)                                                   00000200
      DIMENSION E(1),STRESS(1),STRAIN(1),STRS(1),IX(9,1)                    00000300
      COMMON/XSECT/NSEG(10),Y(20,10),Z(20,10),XLEN(20,10),T(20,10),          00
     *TORCON(10),KLOS(10),ABM(10),FIXX(10),FIYY(10),FIZZ(10)                00000500
C                                                                           00000600
C                                                                           00000700
C     MOMENT CURVATURE RELATION BY PIECEWISE LINEAR INTEGRATION             00000800
C       OF STRESSES OVER PLATEWORK TYPE BEAM CROSS-SECTIONS                 00000900
C                                                                           00001000
C       SIGN CONVENTION                                                     00001100
C        MZ=E*IZ*CURVZ                                                      00001200
C        MY=-E*IY*CURVY                                                     00001300
C                                                                           00001400
C                                                                           00001500
      KBM=IX(6,JE)                                                          00001600
      IF(KBM.EQ.0) KBM=1                                                    00001700
       KPTS=NSEG(KBM)                                                       00001800
      IF(KLOS(KBM).EQ.1)KPTS=KPTS+1                                         00001900
      DO 10 I=1,KPTS                                                        00002000
        YS=Y(I,KBM)                                                         00002100
        ZS=Z(I,KBM)                                                         00002200
10    STRAIN(I)=EXT-YS*CURVZ-ZS*CURVY                                       00002300
      DO 20 I=1,KPTS                                                        00002400
      IND=4*(I-1)+1                                                         00002500
20    CALL STRES(STRAIN(I),STRESS(I),E,STRS(IND))                           00002600
       FX=0.                                                                00002700
      AMZ=0.                                                                00002800
       AMY=0.                                                               00002900
      DO 30 I=1,KPTS                                                        00003000
      CON=T(I,KBM)*XLEN(I,KBM)                                              00003100
      YS=Y(I,KBM)                                                           00003200
      ZS=Z(I,KBM)                                                           00003300
      IF(I.EQ.KPTS)GO TO 40                                                 00003400
      YSN=Y(I+1,KBM)                                                        00003500
      ZSN=Z(I+1,KBM)                                                        00003600
       FX=FX+.5*(STRESS(I)+STRESS(I+1))*CON                                 00003700
       AMY=AMY+(STRESS(I)*(ZSN+2.*ZS)+STRESS(I+1)*(ZS+2.*ZSN))*CON/6.        00
       AMZ=AMZ+(STRESS(I)*(YSN+2.*YS)+STRESS(I+1)*(YS+2.*YSN))*CON/6.        00
```

```
      GO TO 30                                                                  00004000
   40 YSN=Y(1,KBM)                                                              00004100
      ZSN=Z(1,KBM)                                                              00004200
      FX=FX+.5*(STRESS(I)+STRESS(1))*CON                                        00004300
      AMY=AMY+(STRESS(I)*(ZSN+2.*ZS)+STRESS(1)*(ZS+2.*ZSN))*CON/6.              00004400
      AMZ=AMZ+(STRESS(I)*(YSN+2.*YS)+STRESS(1)*(YS+2.*YSN))*CON/6.              00004500
   30  CONTINUE                                                                 00004600
       AMZ=-AMZ                                                                 00004700
      AMX=TORCON(KBM)*E(2)*ROT/(2.*(1.+E(7)))                                   00004800
       RETURN                                                                   00004900
       END                                                                      00005000
```

```
      SUBROUTINE VECTOR(XC,YC,ZC,UD,NODE,JE,DIST,S,IP,A,B,C,ABC)          00000100
      DIMENSION XC(1),YC(1),ZC(1),UD(1),NODE(9,1),IP(1),DIST(1)          00000200
      DIMENSION X(3),Y(3),Z(3)                                           00000300
      DO 105 M=1,3                                                       00000400
      N=NODE(M,JE)                                                       00000500
       J=6*(N-1)                                                         00000600
      X(M)=XC(N)+UD(J+1)                                                 00000700
      Y(M)=YC(N)+UD(J+2)                                                 00000800
  105 Z(M)=ZC(N)+UD(J+3)                                                 00000900
      S=0.                                                               00001000
      DO 110 M=1,3                                                       00001100
      N=IP(M+1)                                                          00001200
      DIST(M)=SQRT((X(M)-X(N))**2+(Y(M)-Y(N))**2+(Z(M)-Z(N))**2)         00001300
  110 S=S+DIST(M)                                                        00001400
      S=0.5*S                                                            00001500
      A=Y(1)*(Z(2)-Z(3))+Y(2)*(Z(3)-Z(1))+Y(3)*(Z(1)-Z(2))               00001600
      B=X(1)*(Z(3)-Z(2))+X(2)*(Z(1)-Z(3))+X(3)*(Z(2)-Z(1))               00001700
      C=X(1)*(Y(2)-Y(3))+X(2)*(Y(3)-Y(1))+X(3)*(Y(1)-Y(2))               00001800
      ABC=SQRT(A**2+B**2+C**2)                                           00001900
      RETURN                                                             00002000
      ENTRY VECT(THETA,XCOMP,YCOMP,ZCOMP)                                00002100
      XCOMP=((X(2)-X(1))*COS(THETA)+(B*(Z(2)-Z(1))-C*(Y(2)-Y(1)))        00002200
     **SIN(THETA)/ABC)/DIST(1)                                           00002300
      YCOMP=((Y(2)-Y(1))*COS(THETA)+(C*(X(2)-X(1))-A*(Z(2)-Z(1)))        00002400
     **SIN(THETA)/ABC)/DIST(1)                                           00002500
      ZCOMP=((Z(2)-Z(1))*COS(THETA)+(A*(Y(2)-Y(1))-B*(X(2)-X(1)))        00002600
     **SIN(THETA)/ABC)/DIST(1)                                           00002700
      RETURN                                                             00002800
      END                                                                00002900
```

```
      SUBROUTINE READIN(NELE,NNODE,NDGREE,MUD,NUMDIS,NODDIS,
     *E,XC,YC,ZC,NODE,ANGLE,UD)
      COMMON/STR/I1,INDEX(300),STRAIN(45),STRESS(45),SIDE(900),IP(6),
     *STRS(10000)
      COMMON/DYNAM/DELT,TIMEND,MXSTEP,NTSTEP,TIME,EYMED
      COMMON/OUTPA/NPRU,NPRS,NPFREQ,NPIC,OUT(9,40),NPOUT(2,10),
     1          TITLE(20),NPRV,NPRP
      COMMON/CONTRL/KONTRL(10)
      COMMON /BEAMS/RMASS(3,3,100),DICOS(3,3,150),BAR(1200),
     *PMASS(4,100),NSECT
      DIMENSION XC(1),YC(1),ZC(1),E(16,1),NODE(9,1),NODDIS(1),UD(1),
     *JJ(4),ANGLE(3,1)
      DIMENSION IHD(10),HD(10)
      EQUIVALENCE (INDEX(1),IHD(1)),(INDEX(11),HD(1))
      INTEGER OUT
      READ(5,1000,END=19191) TITLE
 1000 FORMAT(20A4)
      WRITE(6,2000) TITLE
 2000 FORMAT(1H1/10X,20A4)
      READ (5,1011) NNODE,NELE,NUMMAT,NUMDIS,MXSTEP,NDGREE,DELT,NCORD,
     *NSECT
 1011  FORMAT (6I5,E10.6,2I5)
      WRITE (6,2001) NNODE,NELE,NUMMAT,NUMDIS,MXSTEP,NDGREE,DELT,NCORD,
     *NSECT
 2001 FORMAT(1H0,
     1 4X,'NUMBER OF NODES      ',I10/
     2 5X,'NUMBER OF ELEMENTS   ',I10/
     3 5X,'NUMBER OF MATERIALS  ',I10/
     4 5X,'NO. OF DISP. NODES   ',I10/
     5 5X,'MAX.NO.OF TIME STEPS',I10/
     6 5X,'NO. OF DIMENSIONS    ',I10/
     7 5X,'TIME INCREMENT       ',1PE16.5/
     8 5X,'NO. OF COORD NODES   ',I10/
     9 5X,'NO. OF BEAM XSECT    ',I10)
      READ(5,3010)(KONTRL(I),I=1,10)
 3010 FORMAT(10I5)
      IF(KONTRL(7).EQ.0)KONTRL(7)=2
      I1=KONTRL(7)
      WRITE(6,3011)(KONTRL(I),I=1,10)
```

```
 3011 FORMAT(1H0'KONTRL(I)='/10I5)
C
C
C         INTERPRETATION OF KONTRL VECTOR
C
C    KONTRL(3)        EXPANSION FACTOR IN MESH GENERATOR
C    KONTRL(5)        BETAX1000 INTEGRATION PARAMETER =0
C    KONTRL(6)        ITERATION COUNTER =0
C    KONTRL(7)        NUMBER OF CROSS-SECTION INTEGRATION POINTS
C                     =2 IF ZERO (MINIMUM VALUE)
C    KONTRL(8)        CHECKPOINT INTERVAL  =0 NO OPERATION
C                                          \0 MAKE CHECKPOINT
C                                          )0 RESTART AND CHECKPOINT
C    KONTRL(9)        RESTART CHECKPOINT NUMBER
C    KONTRL(10)       PLOT FLAG - MAKE OUTPUT FILE IF NE 0
C
C
C
C            IO UNIT DESIGNATIONS
C
C
C    LOGICAL UNIT 20      MESHPLOT DATA
C    LOGICAL UNIT 25      CHECKPOINT/RESTART DATA
C    LOGICAL UNIT 22      TIME HISTORY DATA FOR POSTPROCESSING
C
C
C
C
      DO 50 I=1,NUMMAT
      READ(5,1050) MTYPE
      READ(5,1051) (E(J,MTYPE),J=1,16)
      WRITE(6,2050) MTYPE,(E(J,MTYPE),J=1,16)
   50 CONTINUE
 1050 FORMAT(I5,E10.4)
 1051 FORMAT(8E10.4)
 2050 FORMAT(1H0,4X,'MAT',I3,10X/3X,'E-MATRIX',1P8E15.5/
     1 11X,1P8E15.5)
C
C
C       GET BEAM CROSS SECTION DATA
```

```
      C                                                                 00007900
      C                                                                 00008000
            IF(NSECT.GT.0) CALL BGEOM(NSECT)                            00008100
      C        READ AND PRINT NODAL POINT DATA                          00008200
         59 WRITE(6,2003)                                               00008300
       2003 FORMAT(1H0//8X,'NODE NO.',12X,'X-ORDINATE',3X,'Y-ORDINATE',3X,  00008400
           1'Z-ORDINATE')                                               00008500
            L=0                                                         00008600
             NTOT=NNODE+NCORD                                           00008700
         60 READ(5,1003)N,XC(N),YC(N),ZC(N),(PMASS(I,N),I=1,4)          00008800
       1003 FORMAT(I5,5X,7E10.4)                                        00008900
            NL=L+1                                                      00009000
            IF(L.EQ.0) GO TO 70                                         00009100
            NML=N-L                                                     00009200
            DS=NML                                                      00009300
            IF(KONTRL(3).EQ.0) GO TO 69                                 00009400
            FAC=KONTRL(3)                                               00009500
            FAC=FAC/100.0                                               00009600
            DS=0.0                                                      00009700
            DO 641 I=1,NML                                              00009800
        641 DS=DS+FAC**(I-1)                                            00009900
         69 DXC=(XC(N)-XC(L))/DS                                        00010000
            DYC=(YC(N)-YC(L))/DS                                        00010100
            DZC=(ZC(N)-ZC(L))/DS                                        00010200
         70 L=L+1                                                       00010300
            IF(N-L)100,90,80                                            00010400
         80 XC(L)=XC(L-1)+DXC                                           00010500
            YC(L)=YC(L-1)+DYC                                           00010600
            ZC(L)=ZC(L-1)+DZC                                           00010700
            IF(KONTRL(3).EQ.0) GO TO 70                                 00010800
            DXC=DXC*FAC                                                 00010900
            DYC=DYC*FAC                                                 00011000
            DZC=DZC*FAC                                                 00011100
            GO TO 70                                                    00011200
         90 WRITE(6,2004) (K,XC(K),YC(K),ZC(K),K=NL,N)                  00011300
             IF(NTOT-N)100,110,60                                       00011400
        100 WRITE(6,2005)N                                              00011500
            CALL EXIT                                                   00011600
        110 CONTINUE                                                    00011700
```

```
 2004 FORMAT(2X,I10,12X,3F13.5)                                          00011800
 2005 FORMAT(1H0,'NODAL POINT ERROR N=',I5)                              00011900
 4002 FORMAT(1H0//28X,'INITIAL  POINTING  VECTOR  COMPONENTS',4X,'INITIA00012000
     *L  EULER  ANGLE  COMPONENTS'/8X,'NODE NO.',17X,'X',12X,'Y',12X,    00012100
     *'Z',11X,'PHI',9X,'THETA',9X,'PSI')                                 00012200
C         READ AND PRINT ELEMENT NODES                                  00012300
      WRITE(6,2006)                                                      00012400
 2006 FORMAT(1H0/2X,'ELEMENT NO.     N1    N2    N3    N4    N5    N6    N7   00012500
     1N8  MTYP')                                                         00012600
      N=0                                                               00012700
  130 READ(5,1001) M,(NODE(I,M),I=1,9)                                  00012800
 1001 FORMAT(10I5)                                                      00012900
      IF(NODE(9,M).EQ.0)NODE(9,M)=1                                     00013000
       IF(NODE(6,M).EQ.0) NODE(6,M)=1                                   00013100
  140 N=N+1                                                             00013200
      IF(M-N) 170,170,150                                               00013300
  150 DO 151 I=1,8                                                      00013400
      NODE(I,N)=NODE(I,N-1)+1                                           00013500
      IF(NODE(I,N).EQ.1) NODE(I,N)=0                                    00013600
  151 CONTINUE                                                          00013700
      NODE(9,N)=NODE(9,N-1)                                             00013800
  170 WRITE(6,2007) N,(NODE(I,N),I=1,9)                                 00013900
      IF(M-N) 180,180,140                                               00014000
  180 IF(NELE-N) 190,190,130                                            00014100
  190 CONTINUE                                                          00014200
 2007 FORMAT(1X,I8,5X, 9I5)                                             00014300
C***      READ DISPLACEMENT NODE CODES                                  00014400
      IF(NUMDIS.EQ.0) GO TO 200                                         00014500
      READ(5,1201) (NODDIS(I),(ANGLE(J,I),J=1,3),I=1,NUMDIS)            00014600
 1201 FORMAT(I10,3E10.4)                                                00014700
       WRITE(6,2201)                                                    00014800
 2201 FORMAT(1H0,5X,'DISPLACEMENT NODES'//                              00014900
     116X,'NODE',2X,'CONDITION',4X,'ANGLE(1)',2X,'ANGLE(2)',2X,         00015000
     2'ANGLE(3)')                                                       00015100
       DO  195  I=1, NUMDIS                                             00015200
       CALL DECOD ( NODDIS(I), JJ )                                     00015300
       IF ( JJ(4) .GT. NNODE )  GO  TO  405                             00015400
      WRITE(6,2202) I,JJ(4),(JJ(L),L=1,3),(ANGLE(J,I),J=1,3)            00015500
      DO 195 J=1,3                                                      00015600
```

```
157.       195 ANGLE(J,I)=0.0174533*ANGLE(J,I)                                    00015700
158.      2202  FORMAT(5X,I5,5X,I4,4X,3I2,3X,3F10.3)                             00015800
159.       200 CONTINUE                                                           00015900
160.      C       READ OUTPUT CODE CARDS                                         00016000
161.              WRITE (6,1404 )                                                00016100
162.      1404  FORMAT (1H0//5X,'OUTPUT CODE')                                   00016200
163.              READ (5,403 ) NPFREQ,NPRU,NPRS,NPRV,NPRP,NPIC                  00016300
164.       403  FORMAT (6I10)                                                    00016400
165.              WRITE(6,1405) NPFREQ,NPRU,NPRS,NPRV,NPRP,NPIC                  00016500
166.      1405  FORMAT (1H0,9X,'NPFREQ=',I10/                                    00016600
167.         1             10X,'NPRU  =',I10/                                   00016700
168.         2             10X,'NPRS  =',I10/                                   00016800
169.         3             10X,'NPRV  =',I10/                                   00016900
170.         4             10X,'NPRP  =',I10/                                   00017000
171.         5             10X,'NPIC  =',I10)                                   00017100
172.               IF (NPRU .EQ. 0 )  GO TO 411                                 00017200
173.              READ (5,400) ((OUT(J,I),J=1,9),I=1,NPRU)                      00017300
174.       400 FORMAT(I10,5A4,3E10.0)                                           00017400
175.              WRITE (6,401) (I,(OUT(J,I),J=1,9),I=1,NPRU)                   00017500
176.       401  FORMAT (1H0,5X,'DISPACEMENT AND VELOCITY OUTPUT'/               00017600
177.          1(4X,I3,I10,5X,5A4,3E15.4))                                       00017700
178.              DO 410 I=1,NPRU                                               00017800
179.               CALL DECOD (OUT(1,I),JJ)                                     00017900
180.               IF ( JJ(4) .GT.  NNODE )   GO TO 405                         00018000
181.       410 CONTINUE                                                         00018100
182.       411  IF (NPRS .EQ. 0 )  GO TO 215                                    00018200
183.              I3=NPRU+1                                                     00018300
184.              I4=NPRU+NPRS                                                  00018400
185.              READ (5,400) ((OUT(J,I),J=1,9),I=I3,I4)                       00018500
186.              WRITE (6,402) (I,(OUT(J,I),J=1,9),I=I3,I4)                    00018600
187.       402 FORMAT(1H0,5X,'STRESS OUTPUT'/(4X,I3,I10,5X,5A4,3E15.4))         00018700
188.              DO 521 II=I3,I4                                               00018800
189.              CALL DECOD (OUT(1,II),JJ)                                     00018900
190.       521 CONTINUE                                                         00019000
191.       215  CONTINUE                                                       00019100
192.       298  IF ( NPIC .LE. 0 )  GO  TO  512                               00019200
193.              WRITE(6,589)                                                  00019300
194.       589  FORMAT(1H0,5X,'NPOUT PICTURE OUTPUT'/12X,'AT STEP',10X,'CODE') 00019400
195.              DO 591  K=1,NPIC                                              00019500
```

```
196.            READ(5,581) (NPOUT(I,K),I=1,2)                                  00019600
197.      581   FORMAT(2I10)                                                    00019700
198.      591   WRITE(6,582) (NPOUT(I,K),I=1,2)                                 00019800
199.      582   FORMAT(10X,I7,10X,I4)                                           00019900
200.      512   J=0                                                             00020000
201.      C                                                                     00020100
202.      C                                                                     00020200
203.      C                                                                     00020300
204.      C       OUTPUT FOR MESH AND DEFORMED STRUCTURE PLOTS                  00020400
205.      C     CHECK RESTART FLAG, SKIP AND POSITION FILE IF NECESSARY         00020500
206.      C                                                                     00020600
207.      C                                                                     00020700
208.      C                                                                     00020800
209.            REWIND 20                                                       00020900
210.            IF(KONTRL(9).NE.0) GO TO 850                                    00021000
211.            WRITE(20,1060)(K,XC(K),YC(K),ZC(K),K=1,NNODE)                   00021100
212.             DO 856 K=1,NELE                                                00021200
213.            DO 857 I=1,4                                                    00021300
214.      857   IHD(I)=NODE(I,K)                                                00021400
215.            IF(NODE(8,K).LE.1) GO TO 858                                    00021500
216.             IHD(3)=0                                                       00021600
217.             IHD(4)=0                                                       00021700
218.      858   CONTINUE                                                        00021800
219.            WRITE(20,1070) K,(IHD(I),I=1,4)                                 00021900
220.      856   CONTINUE                                                        00022000
221.     1060 FORMAT(I5,3E12.5)                                                 00022100
222.     1070 FORMAT(5I5)                                                       00022200
223.            GO TO 851                                                       00022300
224.      850   IF(NPIC.LE.0) GO TO 851                                         00022400
225.            KREC=KONTRL(9)*IABS(KONTRL(8))                                  00022500
226.             J=0                                                            00022600
227.             DO 853 I=1,NPIC                                                00022700
228.             IF(KREC.GE.NPOUT(1,I))J=I                                      00022800
229.      853   CONTINUE                                                        00022900
230.             DO 852 K=1,NNODE                                               00023000
231.      852   READ(20,1060)KREC,(HD(I),I=1,3)                                 00023100
232.             DO 854 K=1,NELE                                                00023200
233.      854 READ(20,1070)(IHD(I),I=1,5)                                       00023300
234.             IF(J.EQ.0) GO TO 851                                           00023400
```

```
      J=J*NNODE                                                          00023500
      DO 855 K=1,J                                                       00023600
  855 READ(20,1060)KREC,(HD(I),I=1,3)                                    00023700
  851 CONTINUE                                                           00023800
C                                                                        00023900
C                                                                        00024000
C     INITIALIZE TIME HISTORY PLOT FILE IF REQUIRED                      00024100
C                                                                        00024200
      IF(KONTRL(10).EQ.0) GO TO 800                                      00024300
      REWIND 22                                                          00024400
C                                                                        00024500
C     CHECK RESTART FLAG AND POSITION FILE IF NECESSARY                  00024600
C                                                                        00024700
      IF(KONTRL(9).EQ.0) GO TO 801                                       00024800
C                                                                        00024900
C     POSITION FILE - THREE HEADER RECORDS                               00025000
C                                                                        00025100
      DO 802 I=1,3                                                       00025200
  802 READ(22) RECORD                                                    00025300
      J=KONTRL(9)*IABS(KONTRL(8))                                        00025400
      DO 803 I=1,J                                                       00025500
  803 READ(22) RECORD                                                    00025600
      GO TO 800                                                          00025700
  801 CONTINUE                                                           00025800
C                                                                        00025900
C     WRITE HEADER RECORDS                                               00026000
C                                                                        00026100
      NTOT=NPRU+NPRS                                                     00026200
      IHD(1)=NTOT                                                        00026300
      IHD(2)=NPRU                                                        00026400
      IHD(3)=NPRS                                                        00026500
      IHD(4)=MXSTEP                                                      00026600
      IHD(5)=MXSTEP                                                      00026700
      IHD(6)=0                                                           00026800
      IHD(7)=0                                                           00026900
      IHD(8)=NELE                                                        00027000
      HD(1)=DELT                                                         00027100
      WRITE(22) IHD,HD,TITLE                                             00027200
      WRITE(22)(NODE(5,I),I=1,NELE)                                      00027300
```

```
274.                WRITE(22)(OUT(1,I),I=1,NTOT)                          00027400
275.          800 CONTINUE                                                00027500
276.        C                                                             00027600
277.              IF ( NPFREQ .LE. 0 )  NPFREQ=1                          00027700
278.              RETURN                                                  00027800
279.          405  WRITE (6,2410 )                                        00027900
280.         2410  FORMAT (1H0,5X,'ERROR IN OUTPUT DATA')                 00028000
281.        19191  CALL EXIT                                             00028100
282.              RETURN                                                  00028200
283.              END                                                     00028300
```

```
      SUBROUTINE FREEFD(KONWAV,NUMDIS,NODDIS,NNODE,NDGREE,XC,YC,ZC,UD,   00000100
     *UD1,UD2,FORCD,E,IX,IP)                                             00000200
      DIMENSION ZC(1),E(16,1),IX(9,1),IP(1),UD2(1)                       00000300
      DIMENSION UD(1),UD1(1),FORCD(1),NODDIS(1),XC(1),YC(1)              00000400
C                                                                        00000500
C     LOADING FOR 4 NODE CANTILEVER BEAM PROBLEM--6 DOF                  00000600
C         10 LB ON NODE FOUR IN Z-DIRECTION                              00000700
C                                                                        00000800
C                                                                        00000900
      FORCD(21)=10.                                                      00001000
      RETURN                                                             00001100
      END                                                                00001200
```

```
      SUBROUTINE OUTPUT                                                 00000100
     COMMON/WORK/ANGLE(600)                                             00000200
     COMMON XC(300),YC(300),ZC(300),UD(900),UD1(900),UD2(900),          00000300
    *FORCD(900),E(16,5),                                                00000400
    *IX(9,300),NODDIS(100),NUMNP,NUMEL,NDGREE,MUD,NUMDIS                00000500
      COMMON/DYNAM/DELT,TIMEND,MXSTEP,NTSTEP,TIME,EYMED                 00000600
     COMMON/OUTPA/NPRU,NPRS,NPFREQ,NPIC,OUT(9,40),NPOUT(2,10),          00000700
    1             TITLE(20),NPRV,NPRP                                   00000800
     COMMON/STR/I1,INDEX(300),STRAIN(45),STRESS(45),SIDE(900),IP(6),    00000900
    *STRS(10000)                                                        00001000
     COMMON/CONTRL/KONTRL(10)                                           00001100
      DIMENSION UU(20),SS(20),PSU(1,1),AUX(900),JJ(4),D(3)              00001200
      EQUIVALENCE (NUMNP,NNODE),(NUMNP,NUMHAL),(NUMEL,NELE)             00001300
      INTEGER OUT                                                       00001400
      IF ( NPRU .LE. 0 .AND. NPRS .LE. 0 .AND. NPIC .LE. 0 ) GOTO 405   00001500
      NP=MOD(NTSTEP,NPFREQ)                                             00001600
      KKK=0                                                             00001700
      NKN=NTSTEP/NPFREQ                                                 00001800
      IF(NP.EQ.0)WRITE(6,103)NTSTEP,TIME                                00001900
  103 FORMAT(1X,'CYCLE=',I5,4X,'TIME=',1PE12.6)                         00002000
      IF ( NPRU .LE. 0 )  GO  TO  104                                   00002100
      DO  151 I=1, NPRU                                                 00002200
      CALL DECOD ( OUT(1,I),JJ)                                         00002300
      II=(JJ(4)-1)*NDGREE +JJ(1)                                        00002400
      IF ( JJ(2) .EQ. 0 )  GO  TO  1                                    00002500
      IF ( JJ(2) .EQ. 1 )  GO  TO  2                                    00002600
      IF ( JJ(2) .EQ. 2 )  GO  TO  3                                    00002700
    1 UU(I)=UD(II)                                                      00002800
      GO TO 15                                                          00002900
    2 UU(I)=UD1(II)                                                     00003000
      GO TO 15                                                          00003100
    3 UU(I)=UD2(II)                                                     00003200
   15 CONTINUE                                                          00003300
      IF ( JJ(3) .EQ. 0 )  GO  TO  151                                  00003400
      KKK=KKK+1                                                         00003500
      PSU(NKN,KKK)=UU(I)                                                00003600
  151 CONTINUE                                                          00003700
      IF(NP.EQ.0) WRITE(6,102)(UU(J),J=1,NPRU)                          00003800
      DO 550 J=1,NPRU                                                   00003900
```

```
      550  AUX(J)=UU(J)                                                    00004000
      102  FORMAT (  25X,1P6E16.6 )                                       00004100
      104  CONTINUE                                                        00004200
           IF ( NPRS .LE. 0 )  GO  TO  220                                 00004300
           I3=NPRU+1                                                       00004400
           I4=NPRU+NPRS                                                    00004500
          I=1                                                              00004600
          NPRS1=NPRS                                                       00004700
           DO 221 II=I3,I4                                                 00004800
           CALL DECOD ( OUT(1,II),JJ)                                      00004900
           J4=JJ(4)/10                                                     00005000
          IF(NDGREE.EQ.2) GO TO 222                                        00005100
          J5=JJ(4)-10*J4                                                   00005200
          K2=JJ(4)/10000                                                   00005300
          IF(K2.EQ.0) GO TO 105                                            00005400
          J3=JJ(4)-10000*K2                                                00005500
          J4=J3/10                                                         00005600
          J5=J3-10*J4                                                      00005700
          K5=J4/100                                                        00005800
          K3=K2/10                                                         00005900
          K1=K2-10*K3                                                      00006000
          K4=K3+100*K5                                                     00006100
          IND=INDEX(K4)-1                                                  00006200
          M1=IND+27*(J5-1)+3*(K1-1)+1                                      00006300
          M2=M1+9                                                          00006400
          GX=SQRT((STRS(M1)-STRS(M1+1))**2+STRS(M1+2)**2)                  00006500
          TX=0.5*SQRT((STRS(M2)-STRS(M2+1))**2+4.*STRS(M2+2)**2)           00006600
          IF(JJ(2).LT.8) GO TO 105                                         00006700
          CALL VECTOR(XC,YC,ZC,UD,IX,K4,D,S,IP,A,B,C,ABC)                  00006800
          THETA=-ACOS(SQRT(S*(S-D(2))/(D(1)*D(3))))                        00006900
          ST=SIN(THETA)                                                    00007000
          CT=COS(THETA)                                                    00007100
      105 IND=INDEX(J4)-1                                                  00007200
          N1=IND+27*(J5-1)+3*(JJ(1)-1)+1                                   00007300
          N2=N1+9                                                          00007400
          GAMMAX=SQRT((STRS(N1)-STRS(N1+1))**2+STRS(N1+2)**2)              00007500
          TAUMAX=0.5*SQRT((STRS(N2)-STRS(N2+1))**2+4.*STRS(N2+2)**2)       00007600
          JJJ=JJ(2)+1                                                      00007700
          GO TO(110,111,112,113,114,115,116,117,118,119), JJJ              00007800
```

```
 79.       110 NPRS1=NPRS1+2                                                    00007900
 80.           CALL VECTOR(XC,YC,ZC,UD,IX,J4,D,S,IP,A,B,C,ABC)                  00008000
 81.           BOTTOM=STRS(N1)-STRS(N1+1)                                       00008100
 82.           THETA=.78539816                                                  00008200
 83.           IF(BOTTOM.EQ.0.) GO TO 108                                       00008300
 84.           THETA=STRS(N1+2)/BOTTOM                                          00008400
 85.       108 THETA=ACOS(SQRT(S*(S-D(2))/(D(1)*D(3))))+0.5*ATAN(THETA)         00008500
 86.           CALL VECT(THETA,XCOMP,YCOMP,ZCOMP)                               00008600
 87.           SS(I)=XCOMP                                                      00008700
 88.           SS(I+1)=YCOMP                                                    00008800
 89.           SS(I+2)=ZCOMP                                                    00008900
 90.           I=I+2                                                            00009000
 91.           GO TO 223                                                        00009100
 92.       111 SS(I)=0.5*(STRS(N1)+STRS(N1+1))+0.5*GAMMAX                       00009200
 93.           IF(K2.NE.0) SS(I)=0.5*(SS(I)+0.5*(STRS(M1)+STRS(M1+1)+GX))       00009300
 94.           GO TO 223                                                        00009400
 95.       112 SS(I)=0.5*(STRS(N1)+STRS(N1+1))-0.5*GAMMAX                       00009500
 96.           IF(K2.NE.0) SS(I)=0.5*(SS(I)+0.5*(STRS(M1)+STRS(M1+1)-GX))       00009600
 97.           GO TO 223                                                        00009700
 98.       113 SS(I)=GAMMAX                                                     00009800
 99.           IF(K2.NE.0) SS(I)=0.5*(SS(I)+GX)                                 00009900
100.           GO TO 223                                                        00010000
101.       114 SS(I)=0.5*(STRS(N2)+STRS(N2+1))+TAUMAX                           00010100
102.           IF(K2.NE.0) SS(I)=0.5*(SS(I)+0.5*(STRS(M2)+STRS(M2+1))+TX)       00010200
103.           GO TO 223                                                        00010300
104.       115 SS(I)=0.5*(STRS(N2)+STRS(N2+1))-TAUMAX                           00010400
105.           IF(K2.NE.0) SS(I)=0.5*(SS(I)+0.5*(STRS(M2)+STRS(M2+1))-TX)       00010500
106.           GO TO 223                                                        00010600
107.       116 SS(I)=TAUMAX                                                     00010700
108.           IF(K2.NE.0) SS(I)=0.5*(SS(I)+TX)                                 00010800
109.           GO TO 223                                                        00010900
110.       117 SS(I)=STRS(N1+18)                                                00011000
111.           GO TO 223                                                        00011100
112.       118 CALL VECTOR(XC,YC,ZC,UD,IX,J4,D,S,IP,A,B,C,ABC)                  00011200
113.           THETA=-ACOS(SQRT(S*(S-D(2))/(D(1)*D(3))))                        00011300
114.          1+STRS(N1+2)*SIN(THETA)*COS(THETA)                                00011400
115.           IF(K2.NE.0) SS(I)=0.5*(SS(I)+STRS(M1)*CT**2+STRS(M1+1)*ST**2     00011500
116.          1+STRS(M1+2)*ST*CT)                                               00011600
117.           GO TO 223                                                        00011700
```

```
      119 CALL VECTOR(XC,YC,ZC,UD,IX,J4,D,S,IP,A,B,C,ABC)                    00011800
          THETA=1.5707963-ACOS(SQRT(S*(S-D(2))/(D(1)*D(3))))                00011900
         1+STRS(N1+2)*SIN(THETA)*COS(THETA)                                 00012000
          IF(K2.NE.0) SS(I)=0.5*(SS(I)+STRS(M1)*CT**2+STRS(M1+1)*ST**2      00012100
         1+STRS(M1+2)*ST*CT)                                                00012200
          GO TO 223                                                         00012300
      222 J12=10*JJ(1)+JJ(2)                                                00012400
           IND=INDEX(J4)-1                                                  00012500
           SS(I)=STRS(IND+J12)                                              00012600
      223 IF(JJ(3).EQ.0) GO TO 221                                          00012700
           KKK=KKK+1                                                        00012800
           PSU(NKN,KKK)=SS(I)                                               00012900
      221 I=I+1                                                             00013000
           IF(NP.EQ.0) WRITE(6,102)(SS(J),J=1,NPRS1)                        00013100
             DO 551 J=1,NPRS                                                00013200
            NXT=NPRU+J                                                      00013300
      551   AUX(NXT)=SS(J)                                                  00013400
      220   CONTINUE                                                        00013500
            IF(KONTRL(10).EQ.0) GO TO 552                                   00013600
C                                                                           00013700
C                                                                           00013800
C     WRITE POST PROCESSOR FILE                                             00013900
C                                                                           00014000
              NTOT=NPRU+NPRS                                                00014100
               WRITE(22)(AUX(J),J=1,NTOT)                                   00014200
      552   CONTINUE                                                        00014300
            IF ( NPIC .LE. 0 )  GO  TO  109                                 00014400
            DO  129  I=1,NPIC                                               00014500
            IF ( NTSTEP .NE. NPOUT(1,I) )  GO  TO  129                      00014600
            WRITE(6,302)  TIME                                              00014700
      302   FORMAT('1', 9X,'TIME=',E20.6/                                   00014800
         1            10X,'DISPLACEMENTS AT NODE',10X,'IN X-DIRECTION',10X, 00014900
         2                 'IN Y-DIRECTION',10X,'IN Z-DIRECTION')           00015000
            N=NDGREE                                                        00015100
             DO 130  LD=1,NUMHAL                                            00015200
              L=NDGREE*(LD-1)                                               00015300
      130   WRITE(6,301) LD,(UD(L+J),J=1,NDGREE)                            00015400
      301   FORMAT(10X,I5,5X,6E12.5)                                        00015500
C                                                                           00015600
```

```
157.      C        OUTPUT FOR DEFKRMED MESH PLOTS                          00015700
158.      C                                                                00015800
159.                DO 1010 J=1,NUMHAL                                     00015900
160.                 L=(J-1)*NDGREE                                        00016000
161.       1010      WRITE(20,1020)J,UD(L+1),UD(L+2),UD(L+3)               00016100
162.       1020   FORMAT(I5,3E12.5)                                        00016200
163.        129   CONTINUE                                                 00016300
164.        109   CONTINUE                                                 00016400
165.              IF ( NTSTEP.LT. MXSTEP )   RETURN                        00016500
166.              IF  ( KKK  .EQ. 0 )  RETURN                              00016600
167.        405 WRITE(6,2410)                                              00016700
168.       2410 FORMAT(1H0,5X,'ERROR IN OUTPUT DATA')                      00016800
169.              CALL EXIT                                                00016900
170.              RETURN                                                   00017000
171.              END                                                      00017100
```

```
      SUBROUTINE RESTRT(IFLAG,NREC,ICHK)                                00000100
      COMMON/JUNK/TL(9,300)                                             00000200
      COMMON/DYNAM/DELT,TIMEND,MXSTEP,NTSTEP,TIME                       00000300
      COMMON/STR/I1,INDEX(300),STRAIN(45),STRESS(45),SIDE(900),IP(6),   00000400
     *STRS(10000)                                                       00000500
      COMMON XC(300),YC(300),ZC(300),UD(900),UD1(900),UD2(900)          00000600
      IF(IFLAG)10,20,30                                                 00000700
   10 REWIND 25                                                         00000800
      IF(NREC.EQ.0)RETURN                                               00000900
      DO 40 I=1,NREC                                                    00001000
   40 READ(25)NTSTEP,TIME,UD,UD1,UD2,STRS,TL,ICHK                       00001100
      WRITE(6,60)NTSTEP                                                 00001200
   60 FORMAT(//5X,'*** RESTART AT TIME STEP ',I4,' ***'//)              00001300
      RETURN                                                            00001400
   30 ICHK=ICHK+1                                                       00001500
      WRITE(6,50)ICHK                                                   00001600
   50 FORMAT(/5X,'*** CHECKPOINT NO. ',I4,' ***'/)                      00001700
      WRITE(25)NTSTEP,TIME,UD,UD1,UD2,STRS,TL,ICHK                      00001800
   20 RETURN                                                            00001900
      END                                                               00002000
```

```
      SUBROUTINE ASSBLE(NUMEL,NUMNP,NDGREE,IX,XC,YC,ZC,E,SMASS,STRS,     00000100
     *STRESS,STRAIN,INDEX,I1,SIDE,IP)                                    00000200
       COMMON/JUNK/TL(9,300),POINT(3,3,300)                              00000300
       COMMON /BEAMS/RMASS(3,3,100),DICOS(3,3,150),BAR(1200),            00000400
     *PMASS(4,100),NSECT                                                 00000500
        DIMENSION EULCO(3,3,300)                                         00000600
       DIMENSION TEMP(9)                                                 00000700
       EQUIVALENCE (TL(1,1),EULCO(1,1,1))                                00000800
      DIMENSION STRS(1),STRESS(1),STRAIN(1),SIDE(1),IP(1)                00000900
      DIMENSION IX(9,1),XC(1),YC(1),ZC(1),E(16,1),SMASS(1),INDEX(1)      00001000
      MEQ=NUMNP*NDGREE                                                   00001100
      DO 100 JE=1,MEQ                                                    00001200
  100 SMASS(JE)=0.0                                                      00001300
       DO 101 I=1,NUMNP                                                  00001400
       DO 101 J=1,3                                                      00001500
       DO 101 K=1,3                                                      00001600
  101  RMASS(J,K,I)=0.                                                   00001700
      INDEX(1)=1                                                         00001800
      I3=I1*3                                                            00001900
      DO 105 I=1,3                                                       00002000
      DO 104 J=1,I3                                                      00002100
      K=15*(I-1)+J                                                       00002200
      STRAIN(K)=0.0                                                      00002300
  104 STRESS(K)=0.0                                                      00002400
      IP(I)=I                                                            00002500
  105 IP(I+3)=I                                                          00002600
      WRITE(6,1)                                                         00002700
    1 FORMAT(///5X,'PERIOD OF VIBRATION FOR BEAMS'///4X,'ELEMENT',3X,        00
     1'ELONGATION',3X,'TORSION',3X,'ZZ-MOM ',3X,'ZZ-SHR ',                 00002900
     23X,'YY-MOM ',3X,'YY-SHR '//)                                         00003000
       DO 115 JE=1,NUMEL                                                 00003100
        MTYP=IX(9,JE)                                                    00003200
        KEL=IX(08,JE)                                                    00003300
         IND=INDEX(JE)                                                   00003400
        IF(KEL.GT.1) GO TO 200                                           00003500
        CALL TASME(JE,NDGREE,IX,XC,YC,ZC,E,SMASS,SIDE(3*JE-2),IP,        00003600
     *STRS(IND),NTOT,I1,DICOS(1,1,JE),RMASS)                             00003700
        GO TO 210                                                        00003800
  200 CONTINUE                                                           00003900
```

```
      CALL BASME(JE,NDGREE,IX,XC,YC,ZC,E,SMASS,DICOS,RMASS,PMASS,      00004000
     *I1,STRS(IND),NTOT)                                               00004100
  210 CONTINUE                                                         00004200
      INDEX(JE+1)=INDEX(JE)+NTOT                                       00004300
  115 CONTINUE                                                         00004400
C                                                                      00004500
C INITIALIZE THE LAMBDA TRANSFORMATION MATRIX                          00004600
C                                                                      00004700
        DO 150 I=1,NUMNP                                               00004800
      DO 150 J=1,9                                                     00004900
  150 TL(J,I)=0.0                                                      00005000
        DO 125 I=1,NUMNP                                               00005100
      TL(1,I)=1.0                                                      00005200
      TL(5,I)=1.0                                                      00005300
  125 TL(9,I)=1.0                                                      00005400
      DO 250 I=1,NUMNP                                                 00005500
      DO 260 J=1,3                                                     00005600
  260 RMASS(J,J,I)=RMASS(J,J,I)+PMASS(J+1,I)                           00005700
      TEMP(1)=RMASS(1,1,I)                                             00005800
      TEMP(2)=RMASS(1,2,I)                                             00005900
      TEMP(3)=RMASS(2,2,I)                                             00006000
      TEMP(4)=RMASS(3,1,I)                                             00006100
      TEMP(5)=RMASS(3,2,I)                                             00006200
      TEMP(6)=RMASS(3,3,I)                                             00006300
      CALL EIGEN(TEMP,TL(1,I),3,0)                                     00006400
      CALL CROSS(TL(1,I),TL(4,I),TL(7,I),DUMA,DUMB,0)                  00006500
      DO 253 J=1,3                                                     00006600
      DO 253 K=1,3                                                     00006700
  253 RMASS(J,K,I)=0.                                                  00006800
      RMASS(1,1,I)=TEMP(1)                                             00006900
      RMASS(2,2,I)=TEMP(3)                                             00007000
      RMASS(3,3,I)=TEMP(6)                                             00007100
  250 CONTINUE                                                         00007200
      DO 220 I=1,NUMNP                                                 00007300
        NN=6*(I-1)                                                     00007400
        DO 230 J=1,3                                                   00007500
  230 SMASS(NN+J)=SMASS(NN+J)+PMASS(1,I)                               00007600
        NN=NN+3                                                        00007700
      DO 220 J=1,3                                                     00007800
```

```
  220  SMASS(NN+J)=RMASS(J,J,I)                                                  00007900
C                                                                               00008000
C                                                                               00008100
C    CONSTRUCT INITIAL TL(TRNSP)*MU(HAT)                                        00008200
C                                                                               00008300
C                                                                               00008400
      MJ = 0                                                                    00008500
      DO 50 I = 1,NUMEL                                                         00008600
       KEL=IX(8,I)                                                              00008700
       IF(KEL.GT.1) GO TO 300                                                   00008800
       DO 310 JJ=1,3                                                            00008900
       NN=IX(JJ,I)                                                              00009000
       CALL GTPRD(TL(1,NN),DICOS(1,3,I),POINT(1,JJ,NN),3,3,1)                   00009100
  310  CONTINUE                                                                 00009200
       GO TO 50                                                                 00009300
  300  CONTINUE                                                                 00009400
      DO 51 J1 = 1,2                                                            00009500
      K = IX(J1,I)                                                              00009600
      KK=IX(J1+2,I)                                                             00009700
      DO 52 L = 1,2                                                             00009800
      DO 53 M = 1,3                                                             00009900
      SUM = 0.0                                                                 00010000
      MJ = MJ+ 1                                                                00010100
      DO 54 N = 1,3                                                             00010200
   54 SUM = SUM + EULCO(N,M,KK)*DICOS(N,L,I)                                    00010300
   53 BAR(MJ) = SUM                                                             00010400
   52 CONTINUE                                                                  00010500
      BAR(MJ+1)=XC(K) - XC(KK)                                                  00010600
      BAR(MJ+2)=YC(K) - YC(KK)                                                  00010700
      BAR(MJ+3)=ZC(K) - ZC(KK)                                                  00010800
      MK=MJ                                                                     00010900
      MJ=MJ+3                                                                   00011000
      DO 55 NN=1,3                                                              00011100
      MJ=MJ+1                                                                   00011200
   55 BAR(MJ)=EULCO(1,NN,KK)*BAR(MK+1) + EULCO(2,NN,KK)*BAR(MK+2)               00011300
     1          + EULCO(3,NN,KK)*BAR(MK+3)                                     00011400
   51 CONTINUE                                                                  00011500
   50 CONTINUE                                                                  00011600
       LSTRS=INDEX(NUMEL+1)-1                                                   00011700
```

```
      WRITE(6,400) LSTRS,MJ                                                    00011800
400   FORMAT(///10X,'  STRESS AND BAR STORAGE   LSTRS=',I5,'   LBAR=',           00
     *I5///)                                                                   00012000
      RETURN                                                                   00012100
      END                                                                      00012200
```

```
      SUBROUTINE SOLVE(NUMEL,NUMNP,NDGREE,NUMDIS,NODDIS,SMASS,XC,YC,ZC,
     *IX,E,XO,X1,V,AO,A1,FORCD,FINT,ANGLE)
C***  DYNAMIC TRANSIENT SOLUTION WITH NEWMARK BETA METHOD
      COMMON/JUNK/TL(9,300)
      COMMON/CONTRL/KONTRL(10)
      COMMON/STR/I1,INDEX(300),STRAIN(45),STRESS(45),SIDE(900),IP(6),
     *STRS(10000)
      COMMON/DYNAM/DELT,TIMEND,MXSTEP,NTSTEP,TIME,EYMED
         DIMENSION FINT(1)
           DIMENSION OLDS(600)
      DIMENSION ANGLE(3,1)
      DIMENSION XO(1),X1(1),V(1),AO(1),A1(1),FORCD(1),NODDIS(1),SMASS(1)
     *,XC(1),YC(1),IX(9,1),E(16,1),JJ(4),ZC(1)
      DATA EPSI/1.E-3/
      DO 956 I=1,3
      IP(I)=I
  956 IP(I+3)=I
       I2=I1*2
       I4=I1*4
       I6=I1*6
      ICHK=0
C        CONTROL VARIABLES FOR INTEGRATION
      BETA=KONTRL(5)
      BETA=BETA/1000.
      ITERMX=KONTRL(6)
      MEQ=NUMNP*NDGREE
      IF (BETA.GT.0.5) JSTOP=1
      IF (ITERMX.GT.5) JSTOP=1
      IF (JSTOP.EQ.1) GO TO 980
C        ZERO DEPENDENT VARIABLES
      DO 212 I=1,MEQ
      FORCD(I)=0.0
      AO(I)=0.0
      A1(I)=0.0
      XO(I)=0.
      X1(I)=0.
  212 V(I)=0.
       DO  732  I=1,  I2
       STRAIN(I)=0.0
```

```
      732  STRESS(I)=0.0                                                    00004000
           DO 733 I=1,600                                                   00004100
      733  OLDS(I)=0.                                                       00004200
C          COMPUTE INTEGRATION PARAMETERS                                   00004300
           TIME=0.0                                                         00004400
           NTSTEP=0                                                         00004500
           C1=(0.5-BETA)*DELT*DELT                                          00004600
           C2=BETA*DELT*DELT                                                00004700
           C3=DELT/2.0                                                      00004800
C               INITIAL CONDITIONS                                          00004900
           CALL FREEFD(2,NUMDIS,NODDIS,NUMNP,NDGREE,                        00005000
          *XC,YC,ZC,XO,V,AO,FORCD,E,IX,IP)                                  00005100
             CALL FRCIN(NUMEL,NUMNP,NDGREE,IX,E,XO,FINT,OLDS)               00005200
C              INTEGRATION LOOP                                            00005300
           IF(KONTRL(8).GE.0)GO TO 200                                      00005400
           CALL RESTRT(-1,KONTRL(9),ICHK)                                   00005500
           KONTRL(8)=-KONTRL(8)                                             00005600
      200  CONTINUE                                                         00005700
      400  ITER=0                                                           00005800
           NTSTEP=NTSTEP+1                                                  00005900
           TIME=TIME+DELT                                                   00006000
      401  DO 410 I=1,MEQ                                                  00006100
           FORCD(I)=0.0                                                     00006200
           XO(I)=X1(I)                                                      00006300
      410  A1(I)=AO(I)                                                      00006400
           CALL FREEFD(1,NUMDIS,NODDIS,NUMNP,NDGREE,                        00006500
          *XC,YC,ZC,XO,V,AO,FORCD,E,IX,IP)                                  00006600
      415  ITER=ITER+1                                                      00006700
C          ALTER PRESCRIBED DISPLACEMENTS                                   00006800
C             ZERO ALL ACC. AND VEL. CORRESPONDING TO PRESCRIBED DISPLACEMENT00006900
            DO  430  I=1,  NUMDIS                                           00007000
           CALL DECOD(NODDIS(I),JJ)                                         00007100
           J4=JJ(4)                                                         00007200
            MLREF=NDGREE*(J4-1)                                             00007300
            IF(JJ(1).LE.2) GO TO 510                                        00007400
            MLREF=MLREF+3                                                   00007500
            DO 500 J=1,3                                                    00007600
      500   JJ(J)=JJ(J)-3                                                   00007700
      510   CONTINUE                                                        00007800
```

```
      DO 430 K=1,3                                                      00007900
      IF(ANGLE(K,I).EQ.0.)GO TO 4202                                    00008000
      CALL ROTATE(ANGLE(K,I),J4,XO,NDGREE,1)                            00008100
       CALL ROTAT (  X1,1 )                                             00008200
       CALL ROTAT (   V,1 )                                             00008300
       CALL ROTAT (  A1,1 )                                             00008400
       CALL ROTAT (  AO,1 )                                             00008500
       CALL ROTAT (FORCD,1)                                             00008600
 4202  ML=MLREF                                                         00008700
       DO 429 J=1,3                                                     00008800
      ML=ML+1                                                           00008900
      IF(JJ(J)-1) 429,428,427                                           00009000
  427 XO(ML)=X1(ML)                                                     00009100
      GO TO 426                                                         00009200
  428 X1(ML)=0.0                                                        00009300
      XO(ML)=0.0                                                        00009400
  426 V(ML)=0.0                                                         00009500
      A1(ML)=0.0                                                        00009600
      AO(ML)=0.0                                                        00009700
      FORCD(ML)=0.0                                                     00009800
  429 CONTINUE                                                          00009900
      IF(ANGLE(K,I).EQ.0.)GO TO 430                                     00010000
      CALL ROTATE(ANGLE(K,I),J4,XO,NDGREE,2)                            00010100
       CALL  ROTAT(   X1,2 )                                            00010200
       CALL  ROTAT(    V,2 )                                            00010300
       CALL  ROTAT(   A1,2 )                                            00010400
       CALL  ROTAT(   AO,2 )                                            00010500
       CALL  ROTAT(FORCD,2 )                                            00010600
  430 CONTINUE                                                          00010700
C        COMPUTE NEW DISPLACEMENTS                                      00010800
      DO 420 I=1,MEQ                                                    00010900
       IF(SMASS(I).EQ.0.) GO TO 420                                     00011000
      X1(I)=XO(I)+V(I)*DELT+C1*AO(I)+C2*A1(I)                           00011100
  420  CONTINUE                                                         00011200
       CALL FRCIN (NUMEL,NUMNP,NDGREE,IX,E,X1,FINT,OLDS)                00011300
      SUM=0.0                                                           00011400
      DO  425  I=1, MEQ                                                 00011500
         IF(SMASS(I).EQ.0.) GO TO 425                                   00011600
       A1I=(FORCD(I)-FINT(I))/SMASS(I)                                  00011700
```

```
118.            DA=ABS(A1I+A1(I))                                                00011800
119.            IF ( DA .LE. 0.0)  GO TO 425                                     00011900
120.            SUM=SUM+ABS(A1I-A1(I))/DA                                        00012000
121.              A1(I)=A1I                                                      00012100
122.          425  CONTINUE                                                     00012200
123.            IF ( ITERMX .LE. 1 )  GO TO  450                                 00012300
124.      C     CHECK WHETHER TO CONTINUE ITERATION                              00012400
125.          440 IF(ITER.GE.ITERMX) GO TO 450                                   00012500
126.            DA=MEQ                                                         00012600
127.            IF(SUM/DA.GT.EPSI) GO TO 415                                   00012700
128.          450 DO 455 I=1,MEQ                                               00012800
129.            XO(I)=X1(I)                                                    00012900
130.            V(I)=V(I)+C3*(A1(I)+AO(I))                                     00013000
131.          455 AO(I)=A1(I)                                                  00013100
132.            CALL OUTPUT                                                    00013200
133.             IF(KONTRL(8).LE.0) GO TO 490                                  00013300
134.            IF(KONTRL(8).GT.0.AND.MOD(NTSTEP,KONTRL(8)).EQ.0)              00013400
135.           *CALL RESTRT(1,KONTRL(9),ICHK)                                  00013500
136.          490  CONTINUE                                                    00013600
137.            IF(NTSTEP.LT.MXSTEP) GO TO 400                                 00013700
138.      C        END OF LOOP                                                 00013800
139.            RETURN                                                         00013900
140.          980 WRITE(6,981) ITERMX,BETA                                     00014000
141.          981 FORMAT(1H0,5X,'DATA ERROR,ITERMX     BETA =', I10,E15.6)     00014100
142.             STOP                                                          00014200
143.            END                                                            00014300
```

```
      SUBROUTINE CROSS(A,B,C,BETA,CMAG,IS)                              00000100
      DIMENSION A(3),B(3),C(3)                                          00000200
C                                                                       00000300
C PROGRAM FORMS CROSS PRODUCT                                           00000400
C AND ANGLE BETA.  C NORMED TO CMAG                                     00000500
C        C = A X B                                                      00000600
C                                                                       00000700
      C(1)=A(2)*B(3)-A(3)*B(2)                                          00000800
      C(2)=A(3)*B(1)-A(1)*B(3)                                          00000900
      C(3)=A(1)*B(2)-A(2)*B(1)                                          00001000
      AMAG=0.0                                                          00001100
      BMAG=0.0                                                          00001200
      CMAG=0.0                                                          00001300
      DO 10 I=1,3                                                       00001400
      AMAG=AMAG+A(I)*A(I)                                               00001500
      BMAG=BMAG+B(I)*B(I)                                               00001600
   10 CMAG=CMAG+C(I)*C(I)                                               00001700
      AMAG=SQRT(AMAG)                                                   00001800
      BMAG=SQRT(BMAG)                                                   00001900
      CMAG=SQRT(CMAG)                                                   00002000
      BETA=CMAG/(AMAG*BMAG)                                             00002100
       IF(ABS(BETA).GT.1.) BETA=SIGN(1.,BETA)                           00002200
       BETA=ASIN(BETA)                                                  00002300
      IF(IS.EQ.1)RETURN                                                 00002400
      DO 20 I=1,3                                                       00002500
   20 C(I)=C(I)/CMAG                                                    00002600
      RETURN                                                            00002700
      END                                                               00002800
```

REFERENCES

1. Vail, C. F., "Dynamic Modeling of Automobile Structures from Test Data", Systems Identification of Vibrating Structures, New York, N.Y., 1972

2. Rapin, M. P., "Vehicle Structural Crashworthiness", SAE Paper 700413, May 1970

3. McHenry, R. R. and Miller, P. M., "Automobile Structural Crashworthiness", SAE Paper 700412, May 1970

4. Saczalski, K. J. and Park, K. C., "An Interactive Hybrid Technique for Crashworthy Design of Complex Vehicular Structural Systems", Inter. Conf. on Vehicle Structural Mechanics: A Finite Element Application to Vehicle Design, SAE, Detroit, Mich., March 1974

5. Saczalski, J. J., "Structural Problems Associated with the Prediction of Vehicle Crashworthiness", ASME Applied Mechanics Symposium on Mechanical Problems in Transportation, Detroit, Mich., March 1973

6. Tani, M. and Emori, R. I., "A Study on Automobile Crashworthiness", SAE Paper 700175

7. Kamal, M. M., "Analysis and Simulation of Vehicle to Barrier Impact", SAE Paper 700414

8. Herridge, J. T. and Mitchell, R. K., "Development of a Computer Simulation Program for Collinear Car/Car and Car/Barrier Collisions", Battelle Columbus Laboratories, Report DOT-HS-800-645, January 1972

9. Dynamic Science, "Side Impact Crashworthiness of Full-Size Hardtop Automobiles", Report DOT-HS-800-654, January 1972

10. Selna, L. and Salinas, D., "Dynamic Analysis of Automotive Structural Systems", SAE Paper 700844

11. Kirioka, K. and Hirata, T., "Computing System for Structural Analysis of Car Bodies", in Advances in Computational Methods in Structural Mechanics and Design, UAH Press, Huntsville, Ala., 1972

12. Kirioka, K., "An Analysis of Body Structures", SAE Trans. 7C, Paper 650026, 1966

13. Kirioka, K.; Bhkubo, Y.; and Hoffa, Y. "An Analysis of Body Structures - Part II" SAE Paper 710157

14. Thompson, J. E., "Control of Structural Collapse in Automotive Side Impact Collisions", PhD Thesis, Univ. of Detroit, Mich., 1972

15. Rong-Chung Shieh, "Basic Research in Crashworthiness II -- Precollapse Dynamic Analysis of Plane, Ideal Elasto-Plastic Frame Structures Including the Case of Collision into a Narrow Rigid Pole Obstacle", Cornell Aeron. Lab., CAL Rept. BV-2987-V-3

16. Young, J. W., "Crash: A Computer Simulator of Nonlinear Transient Response of Structures", Philco-Ford Corp., Palo Alto, Calif., Report DOT-HS-091-1-125-B, March 1972

17. Melosh, R. J., "Car-Barrier Impact Response of a Computer Simulated Mustang", Philco-Ford Corp., Palo Alto, Calif., Report DOT-HS-091-1-125-A, March 1972

18. Krey, R. D. and Key, S. W., "Transient Shell Response by Numerical Time Integration", Int. J Num. Methods in Engr., Vol. 17, 1973

19. Belytschko, T.; Chiapetta, R. L.; and Bartel, H. D.; "Efficient Large Scale Nonlinear Transient Analysis by Finite Elements", to appear Intl. J. for Numerical Methods in Engr.

20. Oden, J. T., Finite Elements of Nonlinear Continua, McGraw-Hill, 1972

21. Argyris, J. H.; Kelsey, S. and Kamel, H.; Matrix Methods of Structural Analysis, B. F. deVeubeke (ed.), AGARDograph 72, Pergamon Press, 1964

22. Wempner, G. A., "Finite Elements, Finite Rotations and Small Strains of Flexible Shells", Intl. J. Solids Struct., Vol. 5, 1969, p 117-153

23. Murray, D. W. and Wilson, E. L., "Finite Element Large Deflection Analysis of Plates", J. Engr. Mech. Div., ASCE, Vol. 95, 1969, p 145-163

24. Belytschko, T. and Hsieh, B. J., "Nonlinear Transient Finite Element Analysis with Convected Coordinates", to be published Intl. J. of Numerical Methods in Engr., 1972

25. Newmark, N. M., "A Method of Computation for Structural Dynamics", J. Engr. Mech. Div., ASCE, Vol. 85, EM3, July 1959

26. Zienkiewicz, O. C. The Finite Element Method in Engineering Science, McGraw-Hill, 1971

27. Meek, J. L., Matrix Structural Analysis, McGraw-Hill, 1971

28. Hartzman, M. and Hutchinson, J. R., "Nonlinear Dynamics of Solids by the Finite Element Method", Computers and Structures - Vol. 2, Pergamon Press, 1972, p 47-77

29. System/360 Scientific Subroutine Package Version III Programmers Manual, IBM Technical Publications Dept., White Plains, N.Y., 1968

30. Archer, R. R. and Lange, C. G., "Nonlinear Dynamic Behavior of Shallow Spherical Shells", AIAA Journal, Vol. 3, No. 12, December 1965

31. Leech, J. W.; Witmer, E. A.; and Pian, T.H.H.; "Numerical Calculation Technique for Large Elastic-Plastic Transient Deformations of Thin Shells", AIAA Journal, Vol. 6, No. 12, December 1968

32. Morino, L.; Leech, J. W.; and Witmer, E. A.; "An Improved Numerical Calculation Technique for Large Elastic-Plastic Transient Deformations of Thin Shells", Part 1 - Background and Theoretical Formulation; Part 2 - Evaluation and Applications, J Appl. Mech., June 1971

33. Shaw, L. M., "Vehicle Front-End Structure Crash Test Program", Monthly Progress Report 16, Contract DOT-HS-046-2-486, Dynamic Science, October 1973

DOT HS-801 846

FINITE ELEMENT ANALYSIS OF AUTOMOTIVE STRUCTURES UNDER CRASH LOADINGS
Volume I - Summary

Contract No. DOT-HS-105-3-697

March 1976

Final Report

PREPARED FOR:

U.S. DEPARTMENT OF TRANSPORTATION

NATIONAL HIGHWAY TRAFFIC SAFETY ADMINISTRATION

WASHINGTON, D.C. 20590

Document is available to the public through
the National Technical Information Service,
Springfield, Virginia 22151

The contents of this report reflect the views of IIT Research Institute which is responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views or policy of the Department of Transportation. This report does not constitute a standard, specification or regulation.

TECHNICAL REPORT STANDARD TITLE PAGE

1. Report No. DOT HS-801 846	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle FINITE ELEMENT ANALYSIS OF AUTOMOTIVE STRUCTURES UNDER CRASH LOADINGS Volume I Summary		5. Report Date March 1976 6. Performing Organization Code
7. Author(s) R. E. Welch; R. W. Bruce; T. Belytschko		8. Performing Organization Report No. J6321
9. Performing Organization Name and Address IIT Research Institute 10 West 35 Street Chicago, Illinois 60616		10. Work Unit No. 11. Contract or Grant No. DOT-HS-105-3-697
12. Sponsoring Agency Name and Address Department of Transportation, NHTSA 400 7 Street S.W. Washington, D.C. 20590		13. Type of Report and Period Covered Volume I - SUMMARY FINAL REPORT 6-30-1973 6-30-1975 14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract A research project was undertaken to develop a finite element computer program for use in the dynamic analysis of vehicle structures, including sheet metal, in a crash environment. This research program involved the following major tasks:

- A technique was developed for the finite element analysis of the dynamic response of plate beam structures involving very large displacements and rotations and elastic-plastic material behavior. The principal feature of this technique involves the decomposition of the element displacement field into rigid body components and deformation components thus allowing the use of a small deflection formulation in the analysis.
- A computer program was developed incorporating this analysis procedure for beam and plate elements and rigid links together with appropriate time integration procedures and material property descriptions.

17. Key Words	18. Distribution Statement
finite element computer program, dynamic analysis, vehicles, crashworthiness	Document is available to the public through the National Technical Information Service, Springfield, Virginia 22161

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

Form DOT F 1700.7 (8-69)

16. Abstract (Concl)

- A substantial number of test and demonstration problems were analyzed with this computer program ranging from simple classical solutions for beams and plates through large scale simulations of actual crash tests.

PREFACE

This final report, entitled "Finite Element Analysis of Automotive Structures under Crash Loadings" presents the results of a research project undertaken from 1 July 1973 to 30 June 1975 by IIT Research Institute (IITRI) for the Department of Transportation, National Traffic and Safety Administration (NHTSA) under Contract DOT-HS-105-3-697 (IITRI Project J6321). The report is presented in two volumes: Volume I, Summary Final Report, and Volume II, Technical Final Report.

The NHTSA Contract Technical Manager initially was Dr. Michael Chi and subsequently Mr. James Hackney. Dr. R. E. Welch served as the IITRI Project Manager under general supervision of Mr. A. Longinow, Manager, Structural Analysis Section, and Dr. K. E. McKee, Director, Engineering Mechanics. IITRI personnel who made significant technical contributions to the project are R. W. Bruce, M. Smith and R. E. Welch. Prof. T. Belytschko of the Circle Campus, University of Illinois, served as a consultant to the project.

CONTENTS

LIST OF ILLUSTRATIONS

1. INTRODUCTION

The increased concern in recent years over vehicle safety has focused, in part, on the ability of a vehicle to sustain a crash event; to absorb, redirect and otherwise manage the severe forces and energy associated with the event; and thereby, to lessen the severity of the environment to which passengers are exposed. In addition to basic vehicle crashworthiness, serious attention has also been devoted to such factors as the propensity of a vehicle to damage in minor crash events and the threat posed to pedestrians or other vehicles by particular vehicle designs.

Since quantitative knowledge of the deformation characteristics of vehicle structures is an intrinsic requirement in all considerations of this type, it is not surprising that the developing concern for vehicle safety has given rise to intense and continuing efforts to provide mathematical descriptions of structural deformations in the crash environment. This report presents the results of a research project undertaken by IIT Research Institute (IITRI) for the Department of Transportation to develop a finite element computer program for use in the dynamic analysis of vehicle structure, including sheet metal, in a crash environment.

Present analytical procedures fall largely into two broad categories which may be termed lumped parameter or equivalent systems and formal approximation techniques (primarily finite element). In the former category, the vehicle is treated as an assemblage of lumped masses and resistance elements which are selected to represent specific subassemblies and components. The properties of the various elements of such models are determined by a static and dynamic crush testing, direct measurement, and the judicious use of simple analysis procedures. Such techniques are valuable in that they provide an efficient and inexpensive method of simulation, are fairly accurate for well established vehicle configurations and, in effect, serve as a depository for previous test and design results. The limitations of this type of procedure are principally that they are heavily reliant on previous testing;

largely limited to well-established vehicle configurations and only weakly tied to basic principles of structural behavior.

The finite element approach represents an attempt to make up for the limitations inherent in the lumped parameter analyses, by employing more formal approximation techniques in the discretization of the structure and a greater reliance on more fundamental structural representations and properties. The limitations of this approach are found in the inherent tendency toward more complicated and expensive computations and the difficulties found in modeling the extensively complex phenomena associated with the crash environment. Such phenomena include large deflections and rotations in the deformed structure, regions of intense curvature (wrinkling), material strain rate effects and interference and contact among structural components during the response. These procedures, indeed, are not totally free of a reliance on testing and analytical judgment and, in fact, can be viewed as a somewhat more formal lumped parameter system and used in connection with the more empirical procedure in hybrid, combination models.

2. ANALYSIS

The response of vehicle structures under crash loadings is a complex process primarily involving:

- transient, dynamic behavior
- complicated framework and shell assemblages
- large deflections and rotations
- extensive plastic deformation

Previous attempts at a formal analysis of this process have been only partly successful due to a variety of limitations which, in particular instances, have included inadequacies in element formulations, material representations or solution procedures. The work presented in this report represents an attempt to develop a finite element program which is specifically tailored to the class of problems inherent in vehicle crash response, and which employs or extends current avenues in finite element analysis which seem best suited to such problems.

In this section major features of the present work are briefly described, and some rationale is offered for their use in the context of vehicle analysis.

2.1 Solution Procedure

The analysis procedure is based on explicit, timewise numerical integration of the equations of motion for the node points (three translations and three rotations per node). The choice of an explicit, rather than implicit, procedure is exploited throughout the analysis by using "lumped" nodal masses and by calculating the internal forces at the nodes from direct integration of the element stress fields without reference to element or assembled stiffness matrices for the structure.

The equations of motion at a typical node point for translations and rotations, respectively, are written as follows:

Translations

$$[M][\ddot{u}_G] = [f_G^E] - [f_G^I] \qquad (1)$$

where

[M] - the diagonal, lumped mass matrix for the node

$[u_G]$ - $[\ddot{u}_{xG}, \ddot{u}_{yG}, \ddot{u}_{zG}]$ is the acceleration vector for the node referred to a global system $[X_G, Y_G, Z_G]$ common to all nodes

$[f_G^E]$ - the vector of external, applied loads at the node referred to the global system

$[f_G^I]$ - the vector of forces at the node in the global system which results from deformations in all elements associated with the node.

Rotation

$$\overline{I_x}\ \overline{\alpha_x} = \overline{m_x^E} - \overline{m_x^I} + \overline{\omega_y}\ \overline{\omega_z}\ (\overline{I_y} - \overline{I_z}) \tag{2a}$$

$$\overline{I_y}\ \overline{\alpha_y} = \overline{m_y^E} - \overline{m_y^I} + \overline{\omega_z}\ \overline{\omega_x}\ (\overline{I_z} - \overline{I_x}) \tag{2b}$$

$$\overline{I_z}\ \overline{\alpha_z} = \overline{m_z^E} - \overline{m_z^I} + \overline{\omega_x}\ \overline{\omega_y}\ (\overline{I_x} - \overline{I_y}) \tag{2c}$$

where all components are referred to a coordinate system $[\bar{x}, \bar{y}, \bar{z}]$ which coincides with the principal axes of the lumped mass moments of inertia at each node and which rotates with the node (i.e., a set of rigid body axes for each node) and where $[\overline{I_x}, \overline{I_y}, \overline{I_z}]$ are the principal mass moments, $[\overline{\alpha_x}, \overline{\alpha_y}, \overline{\alpha_z}]$ are instantaneous angular accelerations, $[\overline{\omega_x}, \overline{\omega_y}, \overline{\omega_z}]$ are the angular velocities, and where $[\overline{m^E}]$ and $[\overline{m^I}]$ are the external and internal moment vectors at the node point.

The orientation of the nodal axes $[\bar{x}, \bar{y}, \bar{z}]$ with respect to the global axes at any time during the motion is established by the components of three unit vectors $\overline{b_1}, \overline{b_2}, \overline{b_3}$ which remain fixed along the nodal axes $[\bar{x}, \bar{y}, \bar{z}]$, respectively, as the node rotates. If the global components of these three unit vectors are

$$\overrightarrow{b_1}: (b_{x\bar{x}}, b_{y\bar{x}}, b_{z\bar{x}}) \tag{3a}$$

$$\overrightarrow{b_2}: (b_{x\bar{y}}, b_{y\bar{y}}, b_{z\bar{y}}) \tag{3b}$$

$$\overrightarrow{b_3}: (b_{x\bar{z}}, b_{y\bar{z}}, b_{z\bar{z}}) \tag{3c}$$

then the components of any vector, $\vec{v}$, transform from the nodal to global system by the transformation

$$[v_G] = [B][\bar{v}] \tag{4}$$

where the columns of [B] are the components of the three unit vectors.

The choice of an explicit integration procedure and a direct calculation of nodal forces results in a program with minimum (but still substantial) computer storage requirements. This, in turn, is equivalent to a capability of processing reasonably detailed and extensive structural models with relative ease. An early capability for handling relatively large problems is considered essential in developing realistic simulations of actual crash events because of the complex geometry and construction found in real vehicles. Furthermore, this procedure lends itself to a reasonably well organized modular program which admits subsequent change and development, including the further incorporation of an implicit method.

2.2 Element Formulation

The treatment of large displacements and rotations employs a decomposition of the element displacement field into a rigid body rotation and translation associated with a local coordinate system attached to and moving with the element, and a remaining displacement field which describes the deformation of the element relative to the current position of the element axes. This transformation, in effect, removes the average rigid body rotation of the element and allows the use of small or moderate deflection element formulations in the calculation of element forces. In this manner, extremely large rotations and deflections can be accommodated by the analysis with accuracy depending primarily on the size of the elements relative to the curvature of the structure. Although a formal convergence theorem is lacking, the decomposition does hold for infinitesimal regions and numerical studies show excellent agreement with classical solutions. The computer program at present includes low order triangular plate elements and three-dimensional beam elements.

A coordinate system $[\hat{x},\hat{y},\hat{z}]$ is embedded in each element and serves to define the rigid body rotation of the element and as a reference for element distortions and forces. The components of a vector, $\vec{V}$, transform from the element coordinate system to the global coordinate system by the time varying transformation

$$[v_B] = [E][\hat{v}]$$

where the elements of E are

$$\begin{bmatrix} e_{x\hat{x}} & e_{x\hat{y}} & e_{x\hat{z}} \\ e_{y\hat{x}} & e_{y\hat{y}} & e_{y\hat{z}} \\ e_{z\hat{x}} & e_{z\hat{y}} & e_{z\hat{z}} \end{bmatrix}$$

which correspond columnwise with the global components of unit vectors $\vec{e_1},\vec{e_2},\vec{e_3}$ oriented with the element axes $[\hat{x},\hat{y},\hat{z}]$ respectively. An element coordinate system for a triangular plate in the undeformed and deformed positions is shown in Figure 1. The transformations [E] for each element and [B] for each node together with the components of the normal vector, $\vec{n}$, at each node or an element, provide sufficient information to calculate deformations in the element. The deformations, in turn, are used to calculate stresses within the element and, subsequently, forces and moments at the nodes. Finally, the nodal forces and moments are entered in the appropriate summations in the equations of motion (equations (1) and (2)).

2.3 Material Properties

The computer program currently uses simple elastic-plastic stress strain laws; a uniaxial relation for beam elements and a biaxial strain hardening Mises model for plates. Element forces and bending moments for given strain fields are calculated by piecewise linear numerical integration of the stresses at selected points in the cross section. The program is designed to accommodate other constitutive relations readily or to accept explicit relations among thrust, moment, extension and curvature. In fact linear relationships are provided at this level which result in more efficient computations for this class of problem.

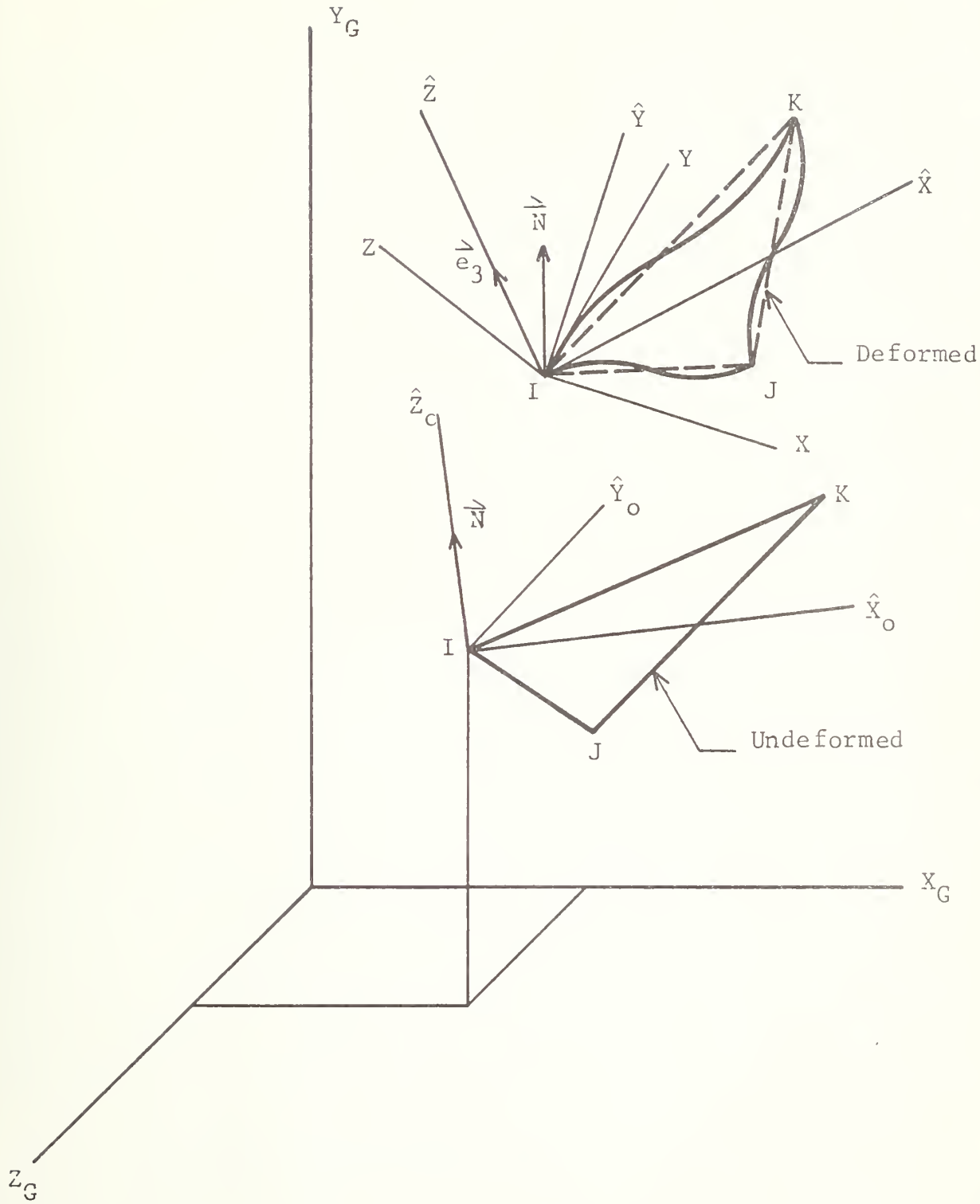


Figure 1 Three-Dimensional Plate Element

3. ANALYSIS RESULTS

A substantial number of test and demonstration problems were treated during the development of the analysis and computer program reported here. For the most part the test problems dealt with relatively simple structural configurations having known results and were selected to isolate and test particular aspects of the formulation and program during development. Thirteen such test problems of successively increasing complexity were formally reported in progress reports. In all cases the analyses were pursued until satisfactory comparison with previous results were obtained or at least to the point at which differences between results could clearly be reconciled on the grounds of differences in loadings, boundary conditions, or ambiguities in the results used for comparison purposes. The contents of these test problems are recapitulated briefly in Table 1.

In addition to the more idealized test problems an effort was made using relatively large scale simulations of actual vehicle crash events. The events which are the objects of these simulations were selected from a series of barrier tests conducted by Dynamic Science which involved frontal impacts at various velocities of a 1968 Plymouth Fury having various portions of the front end structure and machinery removed in different tests. The object of these tests was to isolate and identify the dynamic resistance properties of the various front end components in three groups, namely, frame structure, sheet metal structure and the engine and machinery components. Simulations were developed for two test configurations involving the stub frame alone (test 283-51) and the stub frame and sheet metal together (test 283-53).

The finite element models developed for these two test configurations represent one-half of the structural components forward of the vehicle fire wall and are based on an assumption of symmetry for both the vehicle and its response modes about the center plane.

Table 1

SUMMARY OF TEST PROBLEMS

Problem No.	Description	Element Type and Material Properties	Purpose
1	Cantilever beam with suddenly applied lateral tip load	Two-dimensional beam element; elastic properties	Verify small deflection, dynamic solutions
2	Slightly cambered, simply supported column with sudden axial end shortening	Two-dimensional beam element; elastic properties	Verify large deflection, dynamic solutions
3	Dynamic loading on circular arch	Two-dimensional beam element; elastic properties	Verify dynamic, snap-through solution
4	Impulsive transverse load on axially restrained beam	Two-dimensional beam elements; elastic-plastic properties	Verify elastic-plastic formulation and interaction with axial constraints
5	Plane stress wave propagation	Membrane triangles; elastic and elastic-plastic properties	Verify membrane formulation for in-plane effects
6	Circular membrane with transverse impulsive load	Membrane triangles; elastic and elastic-plastic properties	Verify membrane formulation for out-of-plane effects
7	Elastic, cantilever plate with suddenly applied lateral tip load	Three-dimensional plate bending elements; elastic properties	Verify plate formulation for small deflections
8	Dynamic loading on circular arch	Three-dimensional plate bending elements; elastic properties	Verify plate formulation for plane snap-through problem
9	Elastic cantilever plate	Membrane-hinge line elements, elastic properties	Verify membrane-hinge line formulation

Table 1 (Concl)

Problem No.	Description	Element Type and Material Properties	Purpose
10	Plane-stress wave propagation	Three-dimensional plate bending elements; elastic properties	Verify membrane damping terms
11	Dynamic loading on circular dome	Three-dimensional plate bending elements; elastic properties	Verify plate formulation for three-dimensional snap-through problem
12	Hood structure subjected to end loadings	Three-dimensional plate; elastic-plastic properties	First "representative" vehicle structure and loading
13	Impulsive load on cylindrical panel	Three-dimensional plate; elastic-plastic properties	Verify plate formulation for three-dimensional elastic-plastic response
14	End-on frame crash simulation	Three-dimensional beam elements; elastic-plastic properties	Simulation of actual stub frame test
15	End-on frame and sheet metal crash simulation	Beams and plates; elastic-plastic properties	Simulation of actual frame and sheet metal test

The frame model consists entirely of beam elements and the frame and sheet metal model contains the frame model, plate elements representing the sheet metal, and additional beam elements representing stiffener sections in the hood. In both cases the appropriate inertial properties of the remainder of the vehicle are provided as added masses at the vehicle center of gravity and are attached to the frontal structure by means of rigid links.

As an illustration, we briefly review the results obtained from a simulation of the stub frame and sheet metal barrier test (test 283-53; 44 mph). An isometric view of the finite element mesh for the stub frame and sheet metal model before impact is shown in Figure 2. This model comprises the left front quarter of the 1968 Plymouth employed in the test. The major components of the model are the hood, left front fender, wheel well and frame and radiator support. The wheel well is shown separately for the sake of clarity in the figure but is actually an integral part of the model and is attached along the lines defined by the match points 1 through 4 indicated in the figure. The thickness of all sheet metal elements was taken as 0.039 in. material. The hood portion of the model is free to move relative to the fender at their line of intersection, and, in addition to the plate elements shown, contains 2.0 x 1.0 x 0.05 in. hat section stiffeners at reinforcements. As with the stub frame model additional mass was lumped at the approximate location of the wheel suspension and also at the vehicle center of gravity. In this case all points along the rear edge of the fender and hood as well as the end of the frame were attached to the vehicle center of gravity by means of rigid links. The model was driven by an initial longitudinal velocity of 774 in./sec for all nodes with longitudinal constraints applied at nodes in the forward section of the model successively as they came into contact with the barrier.

The deformed shape of the model at 20 msec after impact is shown in Figure 3. At this point the forward motion of the center of gravity is approximately 15 in. and the forward portions of the

frame assembly and the first two lines of nodes in the hood are indicated in the forward portions of the model with the bumper and bumper brackets collapsed downward and the hood and fender collapsing upward and outward respectively. Separation at the hood and fender junction line is clearly evident and the rear portion of the hood has begun to buckle upward away from the fender. The wheel well is not greatly involved although the forward wall has bent to conform to the distortions in the forward section of the fender.

Longitudinal acceleration at the center of gravity and total barrier force as obtained from test and analysis are shown in Figures 4 and 5. In both cases the test displays the effects of filtering which somewhat obscures the comparisons. The calculated accelerations show an initial peak of about 10 g associated with collapse of the bumper and its supports followed by a larger peak of 30 g as the fender and hood are involved. Maximum acceleration in the test data is about 25 g. The calculated barrier force shows an initial plateau at about 40,000 lbs followed by a sudden rise to about 80,000 lbs as the hood and fender make contact with the barrier. The corresponding maximum barrier load from test (50 percent of full value) is about 60,000 lbs during this phase of the motion.

The test simulations presented here are by no means a complete or exhaustive study even for the limited series of tests which were used for comparison purposes. For a variety of reasons, the results given apply only to the initial phases of the respective crash events and improvements could also be made in the models (the provision of rear wheels or other vertical constraints at the center of gravity, for instance) and in the associated data processing. Despite these limitations, however, the results appear to be a representative portrayal of the phenomena associated with vehicle barrier collisions generally, and a reasonable correlation with the particular test data for which comparisons were attempted.

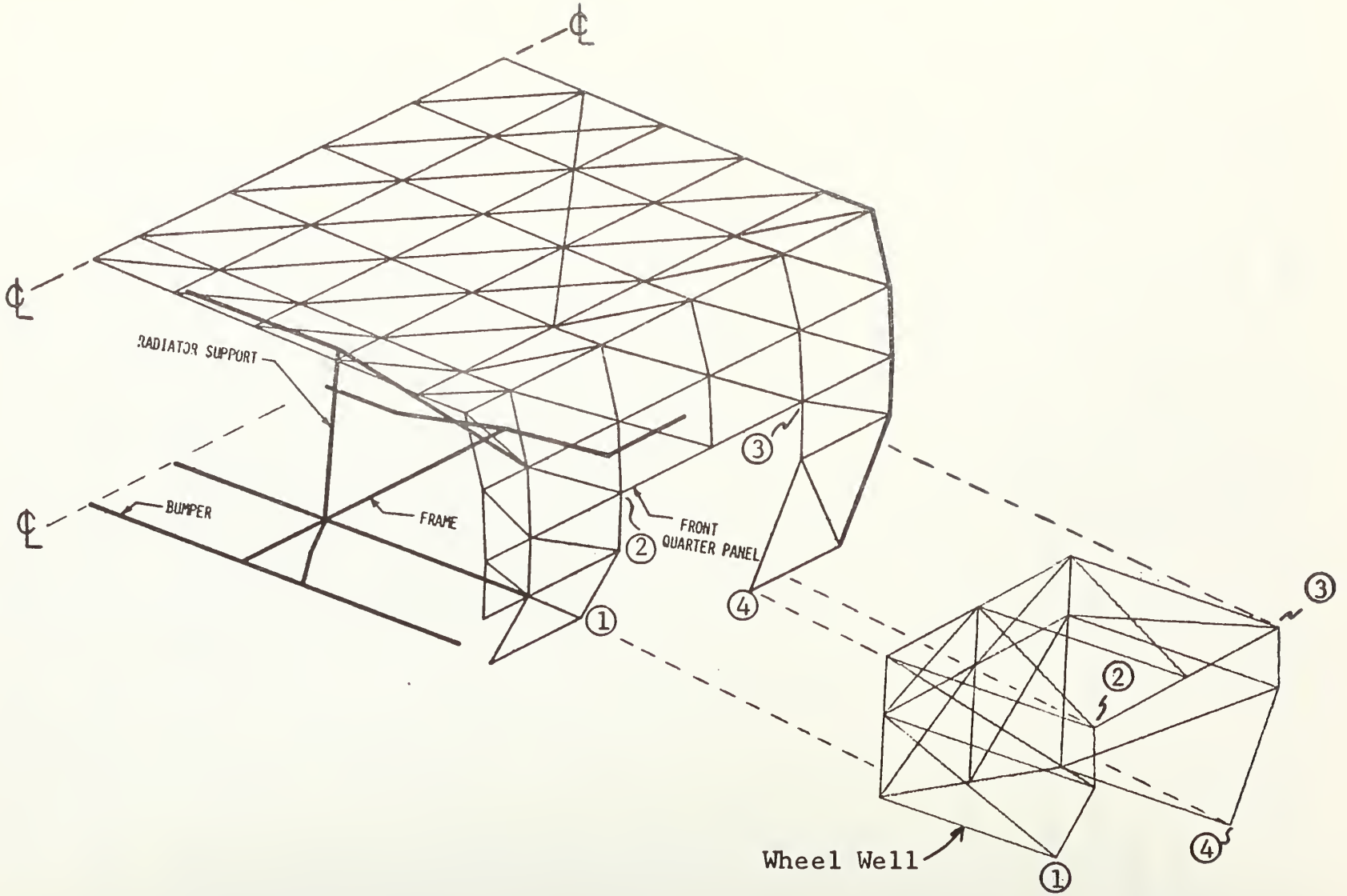


Figure 2 Stub Frame and Sheet Metal Model (Undeformed)

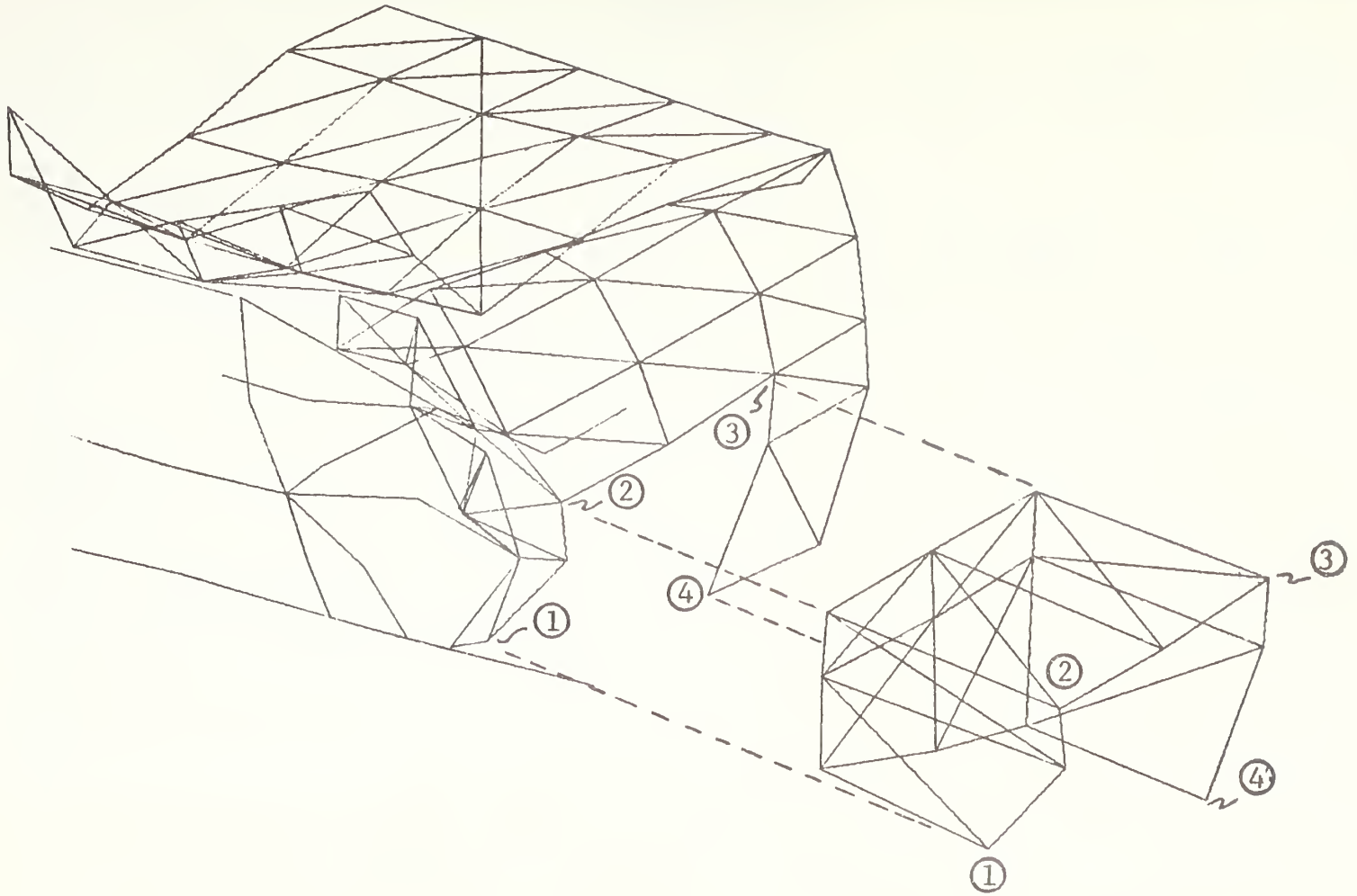


Figure 3 Stub Frame and Sheet Metal Model (Deformed, 20 msec)

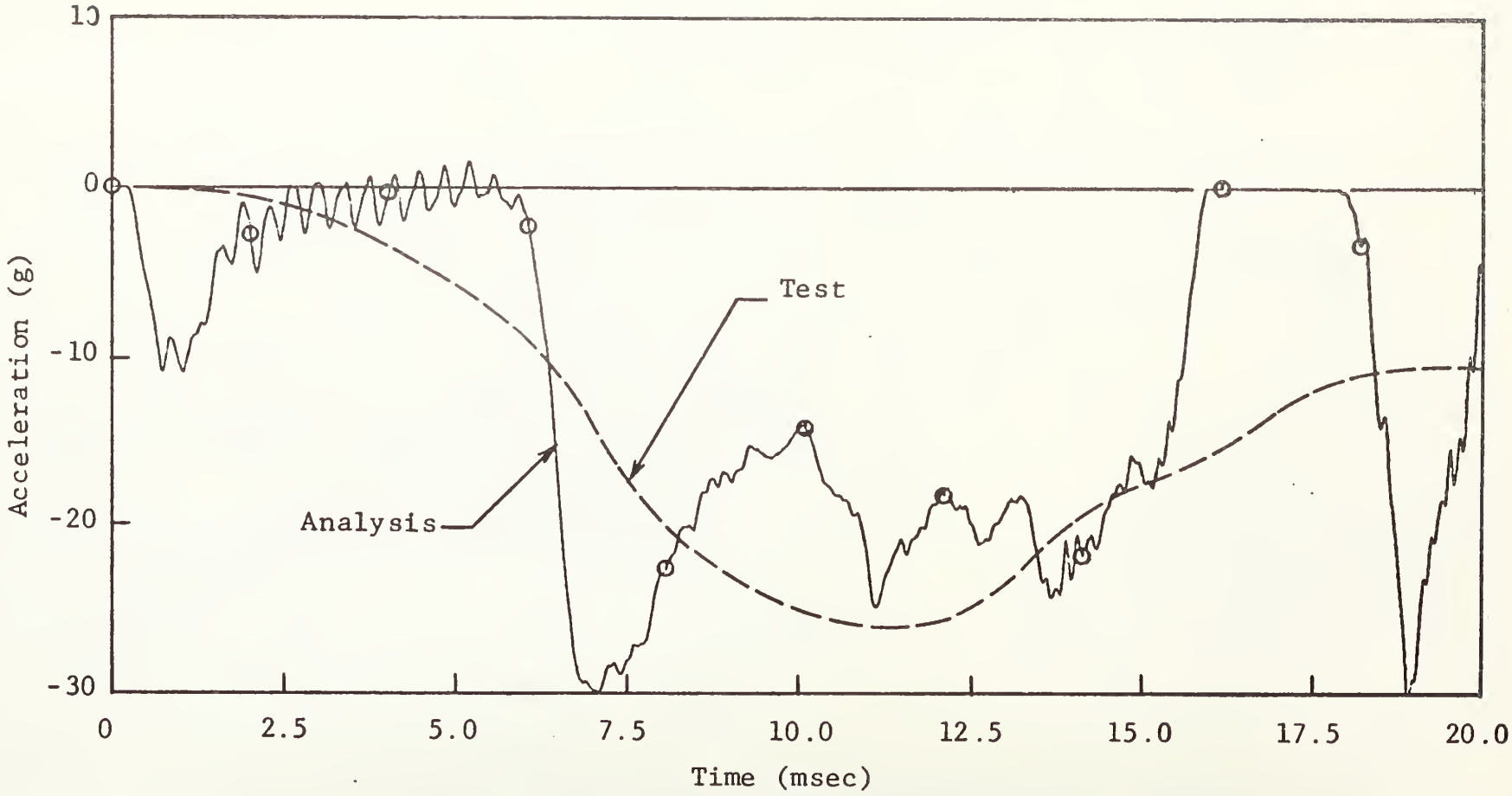


Figure 4 Stub Frame and Sheet Metal Vehicle Center of Gravity Acceleration

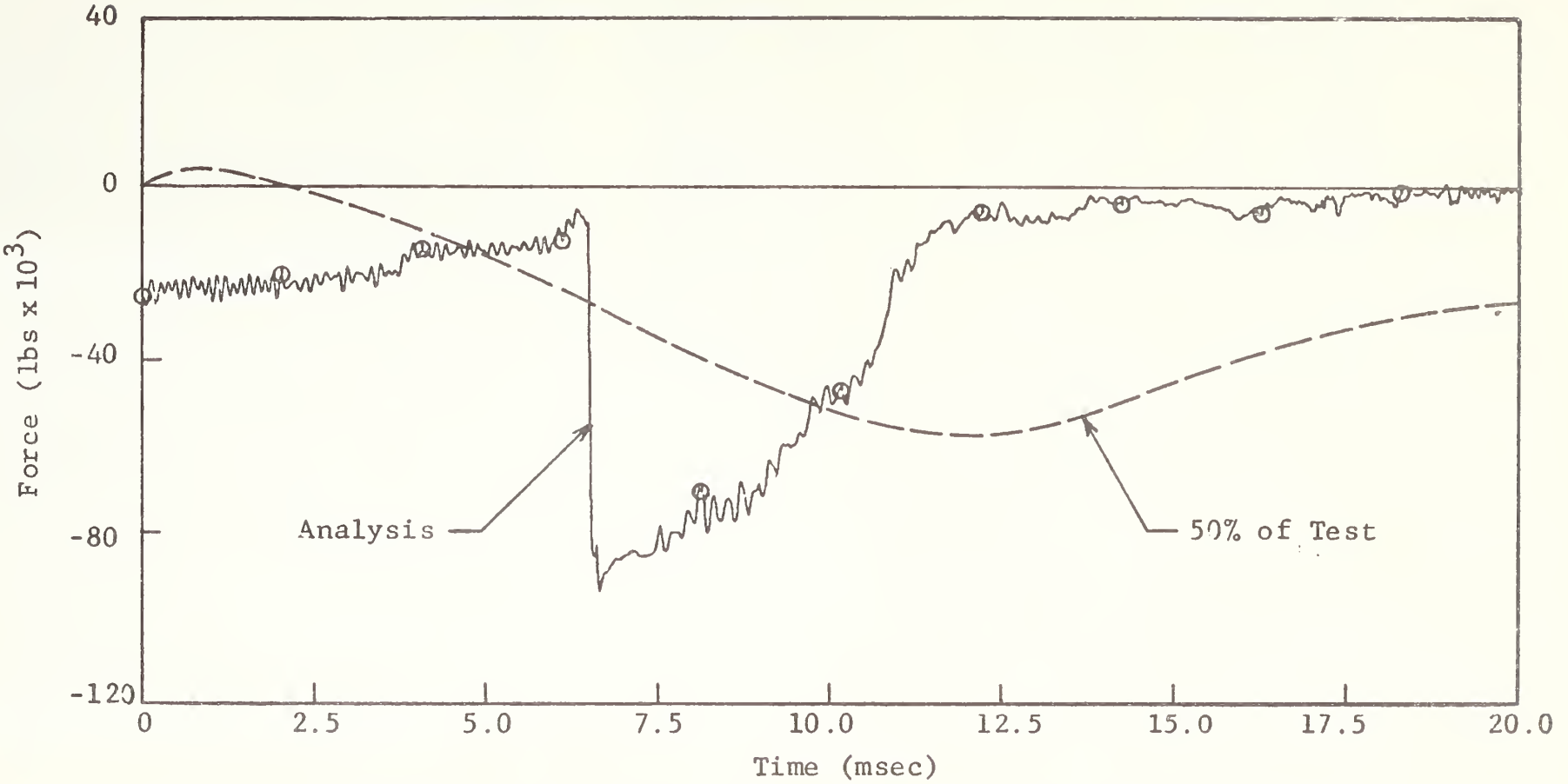


Figure 5 Stub Frame and Sheet Metal Total Barrier Force

4. CONCLUSIONS AND RECOMMENDATIONS

This report presents the results of a research project undertaken by IITRI for DOT to develop a finite element computer program for use in the dynamic analysis of vehicle structures, including sheet metal, in a crash environment. This research program involved the following major tasks:

- A technique was developed for the finite element analysis of the dynamic response of plate beam structures involving very large displacements and rotations and elastic-plastic material behavior. The principal feature of this technique involves the decomposition of the element displacement field into rigid body components and deformation components thus allowing the use of a small deflection element formulation in the analysis.
- A computer program was developed incorporating this analysis procedure for beam and plate elements and rigid links together with appropriate time integration procedures and material property descriptions.
- A substantial number of test and demonstration problems were analyzed with this computer program ranging from simple classical solutions for beams and plates through large scale simulations of actual crash tests.

4.1 Conclusions

Based on the work carried out in this research program our principal conclusions are as follows:

(1) The analysis formulation, while somewhat novel, is a formally sound procedure and a proper and useful approximation technique for the dynamic analysis of beam and plate structures involving large deflections and rotations.

(2) The computer program developed is a correct rendering of this analysis technique and provided accurate results with reasonable efficiency in comparison to available known solutions.

(3) Finally, and most importantly, the computer program is readily applicable to realistic vehicle structures and provides credible results for actual crash events as demonstrated by the simulations of the end-on barrier tests.

4.2 Recommendations

In our opinion the analysis and computer program resulting from this research is a substantial improvement over previous attempts at a finite element analysis of vehicle structures and is a potentially valuable addition to the tools available for the study of vehicle crashworthiness. Although many possibilities suggest themselves in terms of further development and refinement in the analysis procedure and computer program (some of which are noted below) the most important and potentially fruitful activity lies in the immediate and continued application of the analysis to actual vehicles and crash events. Such activity will bring the analysis to real vehicle related problems more quickly and will serve as a guide to its further development and application. Thus our primary recommendation is that provision be made for the continued and extensive use of the program in connection with actual vehicle behavior. Some suggestions for such applications as well as for further analysis and program development are:

Applications

- Additional simulation of controlled crash events along the lines of those attempted in this work but extended to include machinery and drive train components and side-on impacts.
- Use of the program in connection with studies of alternate vehicle designs, including potential structural modifications and configuration changes.
- Use of the program to generate structural crush data for subsequent incorporation in the more simple lumped parameter or hybrid class of programs.
- Comparative studies of the deformation and crashworthiness of existing vehicle designs.

Program Development

- Development of suitable approximate moment-curvature relations for sheet metal as a means of improving computational efficiency.
- Incorporation of plate and beam hinges at selected locations in the main version of the program.

- Development of higher order and special hybrid element formulations possibly to include tearing or wrinkling within the element.

- Development of an implicit rather than explicit integration procedure either as an alternate to the present explicit procedure or for selective use for part of the structure or portions of its response history.