A COMPREHENSIVE STUDY
ON DAMAGE TOLERANCE PROPERTIES OF
NOTCHED COMPOSITE LAMINATES

A. S. D. Wang, E. S. Reddy, W. Binienda and Y. Zhong

Department of Mechanical Engineering
Drexel University
Philadelphia, PA. 19104

FINAL TECHNICAL REPORT
Grant No. AFOSR-84-0334
30 September 1984 to 31 December 1987
Approved for Public Release : Distribution Unlimited

for
Director of Aerospace Sciences
Air Force Office of Scientific Research
Bolling Air Force Base, DC 20332

February, 1988
This final report contains the results of an investigation on matrix-related damage mechanisms in notched composite laminates. The theoretical approach taken follows the principles of micromechanics and the mechanics of brittle fracture at the descriptive level considered valid for the so-called weak-elasticity. Namely, the laminate is basically treated as a 3-dimensional elastic solid which is made of distinctly anisotropic layers. Brittle fracture can initiate and propagate within any layer having a weaker axis of material anisotropy, and within any one of the weaker layer interfaces due to the 3-dimensional interlaminar stresses. Owing to the particular microstructure of the laminate, growth of such sublamine cracks constitutes a load- or time-dependent evolutionary process.

A computer simulation methodology is developed to describe the modes and the extent of damage caused initially by the presence of the notch, and subsequently by the damages themselves. Experiment using a graphite-epoxy laminate is then conducted to validate the simulation results.
# TABLE OF CONTENTS

**FOREWORD** .................................................................................................................. 1

**INTRODUCTION** ........................................................................................................... 2

Objective of Research  
Theoretical Approach  
Crack Growth Simulation  
Major Tasks Performed

**SPECIFIC TASKS AND RESULTS** .................................................................................. 5

3-D Finite Element Code  
Assessing the Accuracy of the Finite Element Method  
Establishing A Mixed-Mode Fracture Criterion  
Simulation of Matrix Cracks in Notched Laminates

**CONCLUSIONS** ............................................................................................................. 8

**REFERENCE** ................................................................................................................ 8

**APPENDIX:**

I "Fracture due to A Kinked Crack in Unidirectional Fiber Reinforced Composites"  
Paper presented at the ASME Winter Annual Meeting, Boston, 1987; also in  

II "A Criterion for Mixed-Mode Matrix Cracking in Graphite-Epoxy Composites"  
Paper presented at the ASTM 9th Symposium on Composites, Reno, 1988; to  
appear in ASTM STP.

III "Three-Dimensional Simulation of Crack Growth in Notched Laminates"  
Paper presented at the 2nd Annual Meeting, Society for Composites, Univ. of  
Delaware, 1987; also in *Proceedings of the American Society for Composites*,  

IV "Simulation of Matrix Cracks in Composite Laminates Containing a Small Hole"  
Paper presented at the ASME Winter Annual Meeting, Boston, 1987; Also in  

FOREWORD

This is the final report for a comprehensive study on damage tolerance properties of notched composite laminates under the Air Force Grant AFOSR-84-0334. The grant was awarded to Dr. A. S. D. Wang of Drexel university with the initial grant period covering from 30 September 1984 to 31 December 1986. However, during the period from 1 September 1986 to 31 August 1987, Dr. worked at the AFOSR as visiting scientist under the Intergovernment Personnel Loan Program; Dr. C. W. Lau then served as an interim principal investigator, with the termination date of the grant extended to 31 December 1987.

The research was performed by Dr. A. S. D. Wang and his assistants: Dr. E. S. Reddy, Drexel University post-doctoral fellow, Dr. W. Binienda and Mr. Y. Zhong, Drexel University graduate students.

Major David A. Glasgow and Lt. Col. George K. Haritcs of AFOSR served successively as technical monitors during the course of this research.
INTRODUCTION

Objectives of Research.

The main objective of this research is to investigate matrix-related damage mechanisms in composite laminates that have a through-the-thickness line-notch or a small hole. A computer simulation methodology is then developed to describe the modes and the extent of damage growth caused initially by the presence of the notch (or hole), and subsequently by the damages themselves.

Theoretical Approach.

The theoretical approach taken in this endeavor followed the principles of micromechanics and the mechanics of brittle fracture at the descriptive level considered valid for the so-called ply-elasticity. Namely, the laminate is basically treated as a 3-dimensional elastic solid which is made of distinctly anisotropic layers. While each layer is assumed homogeneous and endowed with a set of effective elastic constants (see, e.g. [1]), brittle fracture can initiate and propagate within any layer having a weaker axis of material anisotropy, and within any one of the weaker layer interfaces due to the 3-dimensional interlaminar stresses.

Since the propagation modes and the growth behaviors between fracture in a layer and fracture in a layer-to-layer interface differ fundamentally owing to the particular microstructure of the laminate, growth of damages in the form of sublaminate cracks constitute a load-time dependent evolutionary process. The general premise of ply-elasticity and theory of brittle fracture on which a simulation model is based has recently been discussed in detail by Wang [2].

Crack Growth Simulation.

With laminates having a through-the-thickness line-notch or a small hole, stress
concentrations and hence sublamine damages near the notch (or hole) are expected when the laminate is loaded by externally applied load. In order to simulate the damage initiation and damage growth as a function of the applied load, a 3-dimensional analysis of the stress field near the notch or hole must be first performed. Such a stress field, however, contains regions of stress concentration caused not only by the notch (or hole) itself in the usual sense, but also by the interaction of the free edges of the notch (or hole) with the layer interfaces known as free-edge effect [3].

In addition, if one or more sublamine cracks have already initiated near the notch (or hole), the stress field disturbed by the presence of these cracks and the new conditions for these cracks to grow must be continuously analyzed.

Clearly, to effectively analyze such a complex system requires, as a prerequisite, an efficient and accurate finite element computational routine on one hand, and a set of physically consistent material conditions that govern the various crack growth behaviors on the other. Of course, the finite element routine must be developed in accordance within the basic confines of ply-elasticity and the theory of fracture mechanics. Similarly, material conditions governing the various crack growth behaviors must be determined independent of the laminate geometry, both in its overall shape and its lamination structure.

Finally, the simulation methodology must be validated by experiment in which actual growth of sublamine damages is recorded as a function of the applied load. The recorded damage must be measured in quantity units consistent with those simulated numerically so that a direct comparison between the two can be made.

Major Tasks Performed.

Within the context of the foregoing discussions, the following major tasks have been performed during the course of the research:

1. Development of a 3-dimensional finite element code based on ply-elasticity and the linear theory of fracture mechanics. The code is capable of simulating the initiation and growth mechanisms of sublamine cracks
expected to occur in certain notched laminates when they are specifically loaded.

2. Development of rigorous solutions based on anisotropic elasticity and fracture mechanics for a crack problem similar to that anticipated to occur in laminates but mathematically tractable without compromising accuracy. The same problem is then analyzed by means of the developed finite element code. Comparison of results from the two independent solution methods adjudicates the general accuracy of the finite element method.

3. Experiment to establish material conditions that govern the initiation and growth behaviors of the kinds of cracking anticipated to occur in notched laminates. This is accomplished by testing a family of specially designed specimens in which the anticipated cracking occurred, and by simulations of the observed cracking using the developed finite element code.

4. Validation of the simulation method by testing actual laminates that have through-the-thickness line-notches or small holes. Comparisons are then made between the test results and the simulation results, which display the adequacy and/or limitations of the simulation methodology.

In the next section, specifics in each of the tasks are discussed in more detail along with highlights of the results obtained therein. The actual results and the manner in which these results are obtained have been reported in open literature. Four full-length papers and one computer code with user's guide are appended to this report for reference.

The last section outlines a set of concluding remarks pertinent to the major themes of this research.
SPECIFIC TASKS AND RESULTS

3-D Finite Element Code.

As mentioned, the finite element code is developed on the basis of ply-elasticity and the theory of linear fracture mechanics. Its main functions are

1. To compute the 3-D stress field in a laminate of given lamination structure, overall laminate shape, manner of loading, the exact geometry and location of the notch. Because of the expected stress concentrations near the notch region, the code is capable of generating the desired mesh in the region around the notch. The computed stress field provides 6 stress components at any point. In general, stress distribution on any specified plane can be displayed graphically in various isometric forms.

2. To compute the strain energy release rates at any crack-tip with specified direction of propagation. If one or more cracks are already present near the notch, the code can compute the associate stress field as well as the strain energy release rate at one of the crack tip. In cases where the crack may propagate in mixed-modes, then the energy release rate corresponding to each mode can also be calculated. The calculated strain energy release rates are expressed in terms of the appropriately unit for the applied load.

Input data required to run the code include the geometry for the overall laminate specimen shape, the applied load and boundary conditions, the laminate stacking sequence and fiber orientations, the effective elastic constants (including thermal expansion coefficients if appropriate) for each of the laminating layers relative to their respective principal material axes, the location, size and orientation of the notch, and the suspected matrix crack or delamination near the notch.

Appendix V contains the user's guide in which a considerable detail about the code
is discussed. To help run the code, illustrative examples are provided with explanations and actual input/output results. A list of the source code, written in Fortran-IV, is also included.

**Assessing the Accuracy of the Finite Element Method.**

As the developed finite element code is to be used to compute both the stress fields and the fracture quantities for small cracks in layered, anisotropic solids, an effort is made to assess the numerical accuracy the code can provide. To this end, a problem of an ideal overall configuration and loading condition is treated rigorously on the basis of the anisotropic theory of elasticity and fracture mechanics.

The specific problem treated is a unidirectional laminate of infinite domain as illustrated in Figure 1. The laminate contains initially a kink crack and is loaded in uniform tension applied off-axis, making an arbitrary angle $\theta$ with the fibers. The base of the kink crack is normal to the applied tension while the kink itself is in the fiber direction. Thus, the problem is one that involves self-similar, mixed-mode fracture at the kink tip. Within the framework of elasticity theory and linear fracture mechanics, the problem can be formulated exactly and solved rigorously by means of singular integrals and the boundary collocation method.

Solutions to this rigorously formulated problem serve as branch mark from which the finite element solutions can be compared. As it turns out, it is possible to tune the finite element shape and mesh selections in order to yield as accurate numerical results as the rigorous solutions.

Detailed development of this effort has been published in the paper entitled "Fracture due to A Kink Crack in Unidirectional Fiber reinforced Composites." This paper is appended here as Appendix I.

**Establishing A Mixed-Mode Fracture Criterion.**

Another essential element in the present effort to simulate mixed-mode sublamine crack is to ascertain the material condition under which the crack propagates. The problem
is complicated by the fact that fracture of different modes often involves different mechanisms at the microscale, which in turn result in different material conditions for propagation. For fracture propagating in arbitrary combination of modes, a general set of conditions is need. This, however, is not always possible without actually specifying the material.

In the present work, the AS4-3501-06 graphite-epoxy composite system is used in all experiments and simulations. To establish the desired mixed-mode fracture criterion for matrix cracks in this material, a test specimen is designed which can yield crack propagation under 28 different mixed-mode conditions. The test specimen is shown in Figure 2.

It is an off-axis unidirectional tensile coupon with a pair of side notches cut normal to the applied tension. At the critical loading, a kink crack is initiated at one the notch tips and is propagated along the fiber direction in mixed-mode. By varying the off-axis angle $\theta$ and the notch depth, the nature of the mode-mix as well as the critical conditions can thus be altered.

Correlation between experiment and finite element analysis concludes that a useful criterion governing mixed-mode fracture in this material appears to be the total strain energy release rate that exists at the crack tip.

The details of this subject have been included in the paper entitled "A Criterion for Mixed-Mode Matrix Cracking in Graphite Epoxy Composites." This paper is appended here in Appendix II.

Simulation of Matrix Cracks in Notched Laminates.

For simulation of matrix crack growth in laminates, the graphite-epoxy (AS4-3501-06) $[0_{2}/90_{2}]_{S}$ laminate coupon is chosen. The dimension of the actual coupon is 1" wide and 9" long; it is notched in two different forms: (a) a pair of side notches and (b) a small center hole. The applied load is uniaxial tension. Under the applied loading, both in-ply matrix cracks and interply delaminations are expected to occur and grow with the increasing load. In particular, these cracks can occur interactively. It should also be emphasized that in all cases the resulting sublamineate cracks propagate in mixed-modes of various degree of mode-mix.

Evolution of the matrix cracks and delamination in the specimen is both recorded in
experiment and simulated independently by the finite element routine.

Results from this part of the study have been reported in two papers entitled "Three-Dimensional Simulation of Crack Growth in Notched Laminates," and "Simulation of Matrix Cracks in Composite Laminates Containing A Small Hole." These papers are appended here as Appendix II and Appendix IV, respectively.

CONCLUSIONS

In this research program, a simulation method is developed to describe the evolution of matrix cracks in the vicinity of notches in composite laminates. The method is based on a generic approach of the problem in which actual cracking mechanisms are closely modeled. Still, these mechanisms are extremely complex and the simulation has to resort to some degree of idealization. This then causes discrepancies between the simulation and experiment, as is evident by the results reported in the papers appended herein. It is conceivable that these difficulties could be considerable removed if more is known about the interactive mechanisms of the various cracks at the microscopic scale and if a more realistic simulation technique becomes available.

REFERENCE


Figure 1. Kink crack in an infinite unidirectional laminate subjected to uniform tension.
Figure 2. Geometry of test specimen used to establish mixed-mode fracture criterion.
A COMPREHENSIVE STUDY
ON DAMAGE TOLERANCE PROPERTIES OF
NOTCHED COMPOSITE LAMINATES

Appendix I

Fracture due to A Kinked Crack
in Unidirectional Fiber Reinforced Composites

Paper presented at the ASME Winter Annual Meeting, Boston, 1987;
also in Damage Mechanics in Composites, AD-12, ASME, 1987. pp. 73-81.
FRACTURE DUE TO A KINKED CRACK IN UNIDIRECTIONAL FIBER REINFORCED COMPOSITES

W. Binienda and A. S. D. Wang
Department of Mechanical Engineering and Mechanics
Drexel University
Philadelphia, Pennsylvania

F. Delale
Department of Mechanical Engineering
The City College of New York
New York, New York

ABSTRACT

This paper presents an analysis for a kinked crack in a unidirectionally fiber reinforced composite plate. The plate is assumed infinite and contains a through-thickness crack of initial length \( L_1 \), which makes an angle \( \theta \) with the direction of fibers. When the plate is subjected to a uniform far-field tensile stress normal to the crack, the crack can only propagate in the preferential direction of fibers due to the weak strength of the fiber-matrix interface. The result is a kinked crack propagating in mixed mode, with the degree of modal mixture depending on the angle \( \theta \) and the ratio between the length of the kink \( L_2 \) and the length of the initial crack \( L_1 \).

To determine the parameters relevant to mixed-mode fracture at the tips of the kinked crack, the problem is formulated in terms of singular integral equations with generalized Cauchy kernels. The resulting system of equations is then solved numerically employing a Gaussian quadrature and the collocation method. Stress intensity factors, \( k_1 \) and \( k_2 \), and the strain energy release rates, \( G_I \) and \( G_{II} \), of the kinked crack are obtained for various values of \( \theta \) and \( L_2/L_1 \) ratios.

1. INTRODUCTION

Failure in fiber reinforced polymeric composites frequently occurs in the form of matrix cracks due to weak fiber/matrix interface strength. Depending on the local fiber geometry, a matrix crack may propagate in the preferential fiber direction under mixed-mode conditions. Invariably, the relevant fracture parameters which govern matrix crack propagation are dominated by the anisotropic properties of the material. This makes it necessary to formulate an anisotropic criterion for fracture propagation.

Within the framework of the original Griffith theory for brittle fracture, a number of mixed-mode crack propagation criteria have been used for various types of materials, including fiber reinforced composites [1-6]. Sih [5,6], for example, proposed a criterion based on the local strain energy density. Others have used criteria in the general form of \( f(k_1,k_{II})=k_{eff} \). In the experiment by Wu [3], who tested notched balsa wood and unidirectional fiber glass reinforced composite plates, the fracture criterion \( (k_1/k_{IC})+(k_{II}/k_{IIc})^2=1 \) was shown to apply.

In a series of recent papers by Wang, Crossman, et. al. [7-10], the critical energy release rate \( G_{IC} \) was used as a criterion for the initiation and propagation of mode-I cracks in multi-layered laminates. When the crack is blunted by a local fiber or layer interface, the crack would kink and a mixed-mode or shear-dominated fracture would result. In this case, the total critical energy release rate \( (G_{I})_C \) has been employed as a criterion [11].

Regardless of the form of the fracture criteria, it is essential to treat the crack conditions correctly and determine the associated fracture parameters accurately. Fracture problems in homogeneous anisotropic materials have been rigorously studied, see e.g. [12-15]. But for fracture in fibrous composites, material inhomogeneity and the associated microstructure often prevent an analytical solution. A numerical technique such as the finite element method is employed, without a rigorous interrogation of the fracture conditions near the crack tip.

This paper treats a kinked crack in a unidirectionally fiber reinforced composite plate. The plate is assumed to contain a through-thickness crack of initial length \( L_1 \), which makes an angle \( \theta \) with the direction of fibers. When the plate is subjected to a uniform far-field tensile stress normal to the crack, the crack can only propagate in the direction of the fiber because of the weak strength of the fiber-matrix interface. Thus, a kinked crack is induced propagating in mixed mode. Clearly, the nature of the propagation depends on the kink angle \( \theta \), the lengths of the kink \( L_2 \) and the main crack \( L_1 \).

To determine the parameters relevant to mixed-
mode fracture at the tips of the kinked crack, first the problem of two separate cracks embedded in an infinite orthotropic plate is considered. Namely, one crack is the main crack of length \( L_1 \) and the second crack of length \( L_2 \) is assumed to lie along the direction of fibers. The line of \( L_2 \) intersects the line of \( L_1 \) at the origin of the \( x-y \) coordinates as shown in Figure 1. Using the crack surface derivatives as unknown, the problem is formulated on the basis of two-dimensional theory of elasticity and the field equations are expressed in terms of singular integrals with Cauchy type kernels. The system of integral equations is then solved numerically by employing a Gaussian quadrature and the collocation method.

Next, the actual kinked crack is considered. This is accomplished by letting the approaching tips of the kink and the main cracks to touch each other at the intersect of the two crack lines. In this configuration, the singular integral equations are still valid but some of the kernels become singular, giving rise to generalized Cauchy kernels. In fact, it is shown that at the point of touch the stresses are singular and the power of singularity is different from 1/2. Thus, for the kinked crack geometry, a set of singular integral equations with singular kernels is solved. Stress intensity factors, \( K_1 \) and \( K_2 \), and strain energy release rate components \( G_1 \) and \( G_2 \), at the tips of the kinked crack are obtained for various values of \( \beta \) and \( L_2/L_1 \) ratios. Note that the problem of the plate containing only the main crack corresponds to \( L_2 \rightarrow 0 \).

2. FORMULATION OF THE PROBLEM

As stated previously, the problem at hand is a kinked crack in an infinite plate, and it is treated first by considering two separate cracks as depicted in Figure 1. Let the plate be orthotropic with principal directions \( x_1 \) and \( y_1 \). The far-field uniform tension is applied in the direction of \( y_2 \) which makes angle \( \theta \) with \( y_1 \). The main crack of length \( L_1 \) lies on the \( x_2 \) axis, while the inclined crack of length \( L_2 \) (the future kink) lies on the \( x_1 \) axis (which is the direction of the fibers). The stress fields for the individual cracks are first solved, and the stress field for the interacting cracks is then obtained by superposition. A brief outline of the solution procedures is given below; details are contained in Reference [16].

Crack Parallel to the Fibers.

For the crack parallel to the fibers, the governing field equation is expressed in terms of the stress function \( F_1(x_1, y_1) \) in the principal coordinates \( (x_1, y_1) \):

\[
\frac{\partial^4 F_1}{\partial x_1^4} + \beta_1 \frac{\partial^4 F_1}{\partial y_1^4} + \beta_2 \frac{\partial^4 F_1}{\partial x_1^2 \partial y_1^2} = 0
\]  

(1)

where

\[ \beta_1 = \frac{a_{11}}{a_{22}} \quad \beta_2 = \frac{2a_{12} + a_{66}}{a_{22}} \]

(2)

and

\[ a_{11} = \frac{1}{E_{LL}} \quad a_{12} = -\frac{v_{TT}}{E_{LL}} \quad a_{22} = \frac{1}{E_{TT}} \quad a_{66} = \frac{1}{G_{TT}} \]

(3)

\( E_{LL}, E_{TT}, G_{LT}, v_{LT} \) being the engineering elastic constants for the orthotropic material.

Fourier transformation of the stress function \( F_1(x_1, y_1) \) can be defined as:

\[
F_1(x_1, y_1) = \frac{1}{2\pi} \int \phi_1(s, y_1) e^{-isx} ds
\]

(4)

Substituting equation (4) in (1), Ordinary Differential Equation (ODE) with constant coefficients is obtained. The solution of such equation can be expressed as:

\[
\phi_1(s, y_1) = e^{s\gamma_1}
\]

(5)

so the following characteristic equation is obtained:

\[
\beta_1 \omega^4 - \beta_2 \omega^2 + 1 = 0
\]

(6)

The roots of equation (6) are: \( \omega_1, -\omega_1, \omega_2, -\omega_2 \), such that \( \text{Re}(\omega_1) > 0 \) and \( \text{Re}(\omega_2) > 0 \).

Taking into consideration the fact that the stress and displacements must vanish at infinity, the stress function may then be written as:

\[
F_1(x_1, y_1) = \frac{1}{2\pi} \left[ \sum_{i=1}^{2} [A_i e^{s\gamma_1} + B_i e^{-s\gamma_1}] e^{-isx} ds \right], \quad y_1 > 0
\]

(7)

\[
F_1(x_1, y_1) = \frac{1}{2\pi} \left[ [C_i e^{s\gamma_2} + D_i e^{-s\gamma_2}] e^{-isx} ds \right], \quad y_1 < 0
\]

(8)

Using the continuity of stress at \( y_1 = 0 \) and introducing the following crack surface displacement derivatives as the new unknowns,

\[
f_1(x_1) = \frac{\partial}{\partial x_1} \{ u(x_1, 0^+) - u(x_1, 0^-) \}, \quad x_c < x_1 < x_d
\]

(9)

\[
f_2(x_1) = \frac{\partial}{\partial x_1} \{ v(x_1, 0^+) - v(x_1, 0^-) \}
\]

(10)

the stresses may be expressed as:

\[
\sigma_{x_1 x_1} = \frac{1}{2\pi (\omega_1^2 - \omega_2^2)} \left\{ \frac{f_1(t_1) \omega_1 y_1 + f_2(t_1)(t_1 - x_1) \omega_1}{\omega_1^2 y_1^2 + (t_1 - x_1)^2} - \frac{f_1(t_1) \omega_2 y_1 + f_2(t_1)(t_1 - x_1) \omega_2}{\omega_2^2 y_1^2 + (t_1 - x_1)^2} \right\} dt_1
\]

(11)
For more details about the formulation one may refer to [16].

Crack Making an Angle $\theta$ with the Fibers.

In this configuration the crack is assumed to lie along the $x_2$ axis. For a formulation using the $x_2$-$y_2$ coordinate system the material has to be taken as fully anisotropic, giving the following governing equation in terms of the Airy Stress Function $f_2(x_2,y_2)$:

$$
\frac{\partial^2 f_2}{\partial x_2^2} + \frac{\partial^2 f_2}{\partial y_2^2} + \frac{\partial^2 f_2}{\partial x_2 \partial y_2} + \frac{\partial^2 f_2}{\partial x_2 \partial y_2} = 0
$$

(13)

$$
\begin{align*}
\gamma_1 &= - \frac{\partial b_{26}}{\partial y_2} + \frac{\partial b_{12}}{\partial x_2} + \frac{\partial b_{66}}{\partial x_2} \\
\gamma_3 &= - \frac{\partial b_{16}}{\partial y_2} + \frac{\partial b_{11}}{\partial x_2} \\
\gamma_4 &= \frac{\partial b_{16}}{\partial x_2} + \frac{\partial b_{11}}{\partial y_2}
\end{align*}
$$

(14)

and

$$
\begin{align*}
b_{11} &= a_{11} \cos \theta + (2a_{12}^* + a_{46}) \sin \theta \cos^2 \theta + a_{22} \sin \theta \\
b_{22} &= a_{11} \sin \theta + (2a_{12}^* + a_{66}) \sin \theta \cos^2 \theta + a_{22} \cos \theta \\
b_{12} &= a_{12} + (a_{11}^* + a_{22}^* - 2a_{12}^2 + a_{66}^* \sin \theta \cos \theta) \\
b_{66} &= a_{66} + (a_{11}^* + a_{22}^* - 2a_{12}^2 + a_{66}^* \sin \theta \cos \theta) \\
b_{16} &= [a_{22} \sin \theta + a_{11} \cos \theta + \frac{1}{2} (2a_{12}^* + a_{66}^*) \cos 2\theta \sin 2\theta] \cos 2\theta \\
b_{26} &= [a_{22} \sin \theta - a_{11} \cos \theta - \frac{1}{2} (2a_{12}^* + a_{66}^*) \cos 2\theta \sin 2\theta] \cos 2\theta
\end{align*}
$$

(15)

Again following the same procedure, the stress can be expressed in terms of the crack displacement derivatives $f_3(t_2)$ and $f_4(t_2)$ as follows:

$$
\sigma_{x_2} = \frac{1}{2n} \int \left\{ \frac{f_3(t_2) \omega_3(t_1 - x_1) - f_2(t_2) \omega_2 y_1}{\omega_2 y_1 + (t_1 - x_1)^2} + \frac{f_4(t_2) \omega_4(t_1 - x_1) - f_2(t_2) \omega_2 y_1}{\omega_2 y_1 + (t_1 - x_1)^2} \right\} \, dt_1
$$

(12)

For more details about the formulation one may refer to [16].

The Integral Equations.

Stress field for two-crack system is generated by superimposing the two solutions briefly described in two previous sections. It is noted that the stresses are given in different coordinate systems. Therefore the following coordinate transformations are used:

$$
\begin{align*}
x_2 &= x_1 \cos \theta - y_1 \sin \theta \\
y_2 &= x_1 \sin \theta + y_1 \cos \theta
\end{align*}
$$

(21)

or

$$
\begin{align*}
x_1 &= x_2 \cos \theta + y_2 \sin \theta \\
y_1 &= -x_2 \sin \theta + y_2 \cos \theta
\end{align*}
$$

(22)

The total solution for the stress field can be expressed in either $(x_1,y_1)$ or $(x_2,y_2)$. Let superscript $(T)$ be used to denote the total stresses in either system. To satisfy the boundary conditions along $y_2=0$ and $y_1=0$ we may write:

$$
\sigma_{y_2}^T = -\sigma_0
$$

(23)

and

$$
\sigma_{x_2}^T = 0
$$
the problem:

\[
\int_{-1}^{1} f_1(t_1) \, dt_1 = 0
\]

\[
\int_{-1}^{1} f_2(t_2) \, dt_2 = 0
\]

\[
\int_{-1}^{1} f_3(t_1) \, dt_1 = 0
\]

\[
\int_{-1}^{1} f_4(t_2) \, dt_2 = 0
\]

The expressions for the kernels \( K_{ij} \) are functions of material constants and crack geometry [16]. The system of integral equations (27-34) can be solved by using one of the Gaussian quadrature technique [17],[18]. It should be noted that this system of integral equations contain Cauchy type kernels, so the stress and strains will have a square-root singularity and one may therefore use the classical definition of stress intensity factors to evaluate them at the crack tips [12-14].

Solution for the Kinked Cracked

The geometry of interest is that of a kinked cracked. We can arrive at that configuration by letting \( x_b = 0 \) and \( x_c = 0 \). In this case the integral equations (27-30) remain valid but some of the kernels become singular while approaching the tips, giving rise to a singularity of unknown power \( P \) at the apex. The singularity \( P \) can be derived by requiring the displacements of common end to match what giving the following transcendental characteristic equation:

\[
- \frac{n^4}{4} \cos^2 n\beta C_{12} C_{21} C_{22} C_{44} + \frac{n^2}{4} \cos^2 n\beta C_{12} C_{24} A_{22} A_{41} + \frac{n^2}{4} \cos^2 n\beta C_{12} C_{33} A_{24} A_{41} + \frac{n^2}{4} \cos^2 n\beta C_{12} C_{44} A_{13} A_{32} + \frac{n^2}{4} \cos^2 n\beta C_{21} C_{42} A_{14} A_{32} + \frac{n^2}{4} \cos^2 n\beta C_{21} C_{44} A_{13} A_{32} = 0
\]

These equations must be solved with the following single-valuedness conditions which complete the formulation of...
Far the details of the derivation and definition of $A_{kl}$ and $C_{mn}$ one may again refer to [16].

For the same reason two conditions from (31-34) are replaced by:

$$\left( x_{d} \int \frac{f_{2}(\tau_{1})}{x_{d}} d\tau_{1} \right)^{-1} = x_{d} \int \left[ \left( \frac{f_{2}(\tau_{2})}{x_{d}} \sin \theta - \frac{f_{4}(\tau_{2})}{x_{d}} \cos \theta \right) \right] d\tau_{2}$$

(36)

$$x_{d} \int f_{2}(\tau_{1}) d\tau_{1} = x_{d} \int \left[ \left( \frac{f_{2}(\tau_{2})}{x_{d}} \cos \theta + \frac{f_{4}(\tau_{2})}{x_{d}} \sin \theta \right) \right] d\tau_{2}$$

(37)

The singular integral equations have generalized Cauchy kernels and may be solved by using a Gauss-Jacobi [17] or Lobatto-Jacobi quadrature technique [22]. The stress intensity factors at the crack tips can again be derived using their classical definitions.

3. RESULTS AND DISCUSSION.

The important results are those pertaining to the kinked crack case. Here for conciseness only this case is studied in details. To determine the stress intensity factors one must first obtain the singularity $\beta$ by solving equation (35), so certain material properties of orthotropic plate have to be used. The singularity $\beta$ for an isotropic wedge is given in [19]. Similar results are reported for an orthotropic wedge in [20] and [21]. The numerical values of $\beta$ obtained from equation (35) for the special cases of isotropic and orthotropic materials compared closely with those computed in [19-21]. Figure 2 shows the variation of the stress singularity power $(-\beta)$ with the angle $\theta$ for an isotropic orthotropic material. For the orthotropic case the material properties are listed in Table 1.

As expected for $\theta=0$ (i.e. for a half plane) there is no singularity ($\beta=0$) and the singularity increases with increasing wedge angle. The value of $\beta$ must eventually reach the value -0.5 (the well-known square-root singularity) for a crack (i.e. when $\theta=180^\circ$). It is interesting to note that for some orthotropic materials the stress may not be singular even if the wedge angle is larger than $180^\circ$.

The stress intensity factors are obtained by solving equations (27-30) in conjunction with equations (36) and (37). Since the integral equations have generalized Cauchy kernels, the collocation methods described in [17] and [22,23] are used. In the results given subsequently, the stress intensity factors are normalized with respect to the uniaxial load $\sigma_{0}$ and the square-root of their respective half crack length. To check the accuracy of the technique the results are first compared with the solutions of special cases that exist in the literature. Table 2. shows the comparison of the mixed-mode stress intensity factors at the tips of a kinked crack embedded in an infinite isotropic plate with those found in [24-25].

As one may infer from the Table, the results compare rather well. The stress intensity factors at the tips of a kinked crack are given in Figures 3-6. Figures 3 and 4 show the variation of the normalized stress intensity factors with respect to crack length ratio $L_{2}/L_{1}$ whereas Figures 5 and 6 display the same results with respect to the angle $\theta$. Results are obtained for orthotropic as well as isotropic materials.

It is seen that (Figures 3 and 4) for a fixed angle $\theta$, normalized $k_{1}(d)$ and $k_{2}(d)$ decrease with increasing $L_{2}/L_{1}$, however the strain energy release rates increase with increasing $L_{2}/L_{1}$ (Figures 7 and 8). So there is a very small chance for crack arrest, as it is illustrated for the $30^\circ$ plate. On the other hand for varying angle $\theta$ (Figures 5 and 6), $k_{1}(d)$ decreases while $k_{2}(d)$ first increases then decreases with increasing $\theta$. For this case the total strain energy is a monotonic function of $\theta$ (Figures 9 and 10). Thus the resistance to fracture may strongly depend on the direction of reinforcing fibers. It may be seen that (Figure 9) for isotropic materials $G$ is a monotonically decreasing function with increasing $\theta$, while for the orthotropic material used in the calculations (Figure 10), $G$ first decreases then...
increases due to strong influence of moduli component of the strain energy release rate.

REFERENCES


Figure 1. Superposition Scheme for the Infinite Composite Plate with Two Embedded Cracks.

Table 1. Material constants for orthotropic plate.

<table>
<thead>
<tr>
<th>$E_L$</th>
<th>$21.08 \times 10^6$ psi.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_T$</td>
<td>$1.5 \times 10^6$ psi.</td>
</tr>
<tr>
<td>$G_{LT}$</td>
<td>$0.98 \times 10^6$ psi.</td>
</tr>
<tr>
<td>$V_{LT}$</td>
<td>$0.3$</td>
</tr>
</tbody>
</table>

Table 2. Comparison of present solution with references for the special case of isotropic material.

<table>
<thead>
<tr>
<th>$\theta$</th>
<th>$k_1(e)$</th>
<th>$k_2(e)$</th>
<th>$k_1(d)$</th>
<th>$k_2(d)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>30°</td>
<td>[24] 1.3559</td>
<td>0.0327</td>
<td>1.0873</td>
<td>0.6833</td>
</tr>
<tr>
<td></td>
<td>[25] 1.3508</td>
<td>0.0325</td>
<td>1.0830</td>
<td>0.6804</td>
</tr>
<tr>
<td>Present</td>
<td>1.3421</td>
<td>0.0328</td>
<td>1.0949</td>
<td>0.6855</td>
</tr>
<tr>
<td>45°</td>
<td>[24] 1.2502</td>
<td>0.0211</td>
<td>0.7463</td>
<td>0.4095</td>
</tr>
<tr>
<td></td>
<td>[25] 1.2887</td>
<td>0.0208</td>
<td>0.7438</td>
<td>0.3777</td>
</tr>
<tr>
<td>Present</td>
<td>1.2732</td>
<td>0.0217</td>
<td>0.7546</td>
<td>0.8450</td>
</tr>
<tr>
<td>60°</td>
<td>[24] 1.2221</td>
<td>-0.0109</td>
<td>0.3900</td>
<td>0.8319</td>
</tr>
<tr>
<td></td>
<td>[25] 1.2194</td>
<td>-0.0116</td>
<td>0.3822</td>
<td>0.0292</td>
</tr>
<tr>
<td>Present</td>
<td>1.2082</td>
<td>-0.0108</td>
<td>0.3941</td>
<td>0.0350</td>
</tr>
</tbody>
</table>

Figure 2. Variation of the Stress Singularity Power ($\beta$) with the Angle $\theta$. 
Figure 3. Variation of the Normalized Stress Intensity Factors with $L_2/L_1$. Isotropic Case.

Figure 4. Variation of the Normalized Stress Intensity Factors with $L_2/L_1$. Orthotropic Case.

Figure 5. Variation of the Normalized Stress Intensity Factors with the angle $\theta$. Isotropic Case.

Figure 6. Variation of the Normalized Stress Intensity Factors with the angle $\theta$. Orthotropic Case.

Figure 7. Variation of the Strain Energy Release Rates with $L_2/L_1$. Isotropic Case.

Figure 8. Variation of the Strain Energy Release Rates with $L_2/L_1$. Orthotropic Case.
Figure 9. Mixed-Mode Strain Energy Release Rates at the Kink Tip as a Function of Kink Angle $\theta$. Isotropic Case. ($e_x = 1, L_2 \rightarrow 0$).

Figure 10. Mixed-Mode Strain Energy Release Rates at the Kink Tip as a Function of Kink Angle $\theta$. Orthotropic Case. ($e_x = 1, L_2 \rightarrow 0$).
A COMPREHENSIVE STUDY
ON DAMAGE TOLERANCE PROPERTIES OF
NOTCHED COMPOSITE LAMINATES

Appendix II

A Criterion for Mixed-Mode Matrix Cracking
in Graphite-Epoxy Composites

Paper presented at the ASTM 9th Symposium on Composites, Reno, 1988; also to appear in ASTM STP.
A CRITERION FOR MIXED-MODE MATRIX CRACKING IN GRAPHITE-EPOXY COMPOSITES

W. Binienda, A. S. D. Wang, Y. Zhong and E. S. Reddy
Department of Mechanical Engineering
Drexel University
Philadelphia, PA 19104

ABSTRACT: In this paper, mixed-mode matrix fracture in graphite-epoxy composites has been studied. Experimental investigation was conducted on a family of doubly side-notched unidirectional off-axis specimens. By varying the notch depth and the off-axis angle, a total of 28 fracture conditions of differing mixed-mode ratios was produced. Fracture analysis of the test data suggested that the total strain energy release rate is a suitable material condition for mixed-mode matrix cracking in graphite-epoxy composites.

KEYWORDS: graphite-epoxy, mixed-mode matrix fracture, strain energy release rates, finite element analysis, mixed-mode fracture criterion.

Structural composites, notably laminates made of unidirectional tape systems, can sustain extensive matrix cracking before the load carrying fibers fail. Matrix cracking usually occurs at low stress level due to weak interfacial bond strength between matrix and fiber, and between laminating plies. Thus, propagation of matrix cracks in laminates either follows the fiber-matrix interface or the ply-to-ply interface, or both.

Fig. 1 is an x-radiograph taken from a graphite-epoxy [02/902]s laminate having a center-notch. When the laminate is loaded in uniaxial tension, extensive damage in the form of matrix cracks near the notch can be observed. At this phenomenological scale, matrix cracking can be classified into two major modes. Namely, the intra-ply cracking (fiber-wise splitting) which occurs inside a ply and propagate along the fibers; and the inter-ply cracking...
(delamination) which occurs in the interface between two adjacent plies.

In Fig. 1, the four vertical cracks were initiated first near the hole and then propagated along the fibers in the 0\(^0\)-ply. The driving force here is the interfacial shear due to load-transfer from the fiber bundle cut by the hole to the fiber bundle which is uncut. Because of the constraint stemming from bonding between the 0\(^0\) and the 90\(^0\) plies, the vertical splits propagated stably with the applied tension.

As the vertical cracks propagated away from the hole, another mode of load-transfer then took place between the cracked 0\(^0\)-ply and the uncracked 90\(^0\)-ply. Secondary inter-ply stresses along the roots of the vertical cracks were then induced, which then initiated delamination in the 0/90 interface.

Fracture analysis of the cracked specimen at each major form of cracking reveals that the corresponding crack-tip stress fields are complex and the associated propagation involves both opening and shearing modes.

Model simulation for intra-ply fiber-wise matrix cracking and inter-ply delamination has recently been performed using the strain energy release rate method [1]. This method, when limited to mode-I propagation conditions, has proven useful for modeling brittle matrix cracks in graphite-epoxy systems. In such cases, it is necessary to determine the strain energy release rate \(G_I\) at the crack front as driving force, and to validate the corresponding critical strain energy release rate \(G_{IC}\) as material resistance [1].

**MIXED-MODE FRACTURE CRITERIA**

As illustrated in Fig. 1, most matrix cracking in laminates involves mixed opening and shearing modes. However, the applicability of the energy release rate criterion to mixed-mode cracking has not been as firmly established.

Several studies aimed at establishing criteria for mixed-mode matrix
cracking in unidirectional laminates have been conducted in the past using graphite-epoxy composites. Wilkins, et. al. [2] and Ramkumar, et.al [3] used the cracked-lap shear specimen loaded in uniaxial tension to induce mixed mode-I and mode-II delamination between the lap-layer and the substrate layer. By varying the thickness of the lap-layer relative to the substrate layer, mixed-mode ratio, $G_{II}/G_I$, ranging from 0.35 to 0.45 could be obtained. They observed that the total strain energy release rate $(G_I+G_{II})_c$ obtained under mixed-mode conditions is slightly greater than $G_{IC}$ obtained under pure mode-I conditions. Bradley and Cohen [4] used a cantiliver split-beam specimen loaded by a pair of upward and downward loads applied at the tip of the cantiliver. Variation of the mixed-mode ratio $G_{II}/G_I$ was achieved by changing the ratio of the upward and downward loads. Mixed-mode conditions with $G_{II}/G_I$ ratios ranging from 0 to about 0.6 were produced. They observed that, in composite systems made of brittle matrix, the measured total strain energy release rate $(G_I+G_{II})_C$ increased with $G_{II}/G_I$; but it decreased slightly with $G_{II}/G_I$ in systems of ductile matrix. Wang, et. al. [5] used a double side-notched, off-axis unidirectional laminate specimen loaded in axial tension. By varying the off-axis angle from 0° to 90° and the depth of the notches, mixed-mode conditions with $G_{II}/G_I$ ratios ranging from 0 to about 2.5 were achieved. They found that the total strain energy release rate $(G_I+G_{II})_C$ increased with $G_{II}/G_I$ up to about $G_{II}/G_I = 1.5$; it then remained constant for $G_{II}/G_I$ between 1.5 and 2.5.

Russell and Street [6] used specimens of four different configurations and obtained critical strain energy release rates for a wide range of mixed-mode cracking conditions, including pure mode-II cracking. They showed
that the critical strain energy release rates depended on the test specimen and test method used; hence, a general criterion for all the mixed-mode matrix cracking cases tested could not be established.

One possible reason for the lack of a general criterion has been attributed to the manner in which fracture analysis of the test specimens was performed. In the case of a beam-like specimen, the approximate beam theory was employed, while in the case of the plate-like specimen, a finite element plate model was constructed. These analysis methods lacked the required precision to treat complicated singular stress fields, to simulate the actual loading conditions or to properly represent the exact configuration of the cracked specimens. Significant numerical errors could result in the computed fracture quantities, especially for mixed-mode cracking.

Another possible reason stems from uncertainties about the fracture mechanisms associated with pure mode-II cracking. Specifically, ideally pure mode-II cracking is difficult to simulate by tests. In actual experiment, pure mode-II propagation is often accompanied by some amount of friction between the cracked surfaces. The fracture analysis models do not include any such friction mechanisms. A separate criterion may be needed for pure mode-II cracking.

**THE PRESENT INVESTIGATION**

In this paper, a mixed-mode criterion is suggested for matrix cracks propagating in graphite-epoxy composites. This criterion is based on analysis of test data using specimens of varying cracked configurations, which provide mixed-mode fracture conditions with $G_{II}/G_1$ ratios ranging uniformly from 0 to about 3. The case of predominantly mode-II ($G_{II}/G_1 > 3$) or pure mode-II ($G_1 = 0$) is excluded. Fracture analysis of the test specimens is performed using a finite element crack growth simulation model, as exact solutions for the test
specimen configurations cannot presently be obtained. The accuracy of the simulation model is, however, adjudicated by comparing results of problems of similar crack configurations whose solutions can also be found rigorously.

Experiment

The specimen used in the experiment is a notched off-axis tension coupon prepared from a unidirectional laminate made of Hercules AS4-3501-66 graphite-epoxy prepreg tape. Fig. 2 depicts the general configuration of the coupon. The overall dimension is 23 cm long and 2.5 cm wide. Excluding the 4 cm end-tabs, the clear section of the coupon is about 15 cm in length. The pair side-notches are introduced at the mid-section by an 8-mil (0.2 mm) thick diamond saw.

The depth of the side-notch \( a \) and the off-axis angle \( \theta \) (between the applied tension and the direction of the fibers) are varied in the test program as follows:

\[
\begin{align*}
\theta &= 0^\circ, 5^\circ, 10^\circ, 15^\circ, 20^\circ, 25^\circ, 90^\circ \\
a &= 2.5 \text{ mm}, 3.2 \text{ mm}, 3.8 \text{ mm}, 4.5 \text{ mm}
\end{align*}
\]

As depicted in Fig. 3, the coupon can initiate a kink crack (denoted as \( a' \)) at the side-notch tip and propagate in the fiber direction when the applied tension \( \sigma_0 \) reaches some critical value. The propagation is generally mixed with modes I and II. The degree of mix is determined solely by the angle \( \theta \), if the notch depth \( a \) is held constant. Conversely, if \( \theta \) is held fixed, the critical applied tension at the onset of the kink is determined by the notch depth, \( a \).

In this experiment, a total of 28 mixed-mode fracture conditions were created by varying \( \theta \) and \( a \) as mentioned. This has provided fractures with \( G_{II}/G_I \) ratios ranging uniformly from 0 to about 3. It should be noted that mixed-mode matrix fracture in such a wide \( G_{II}/G_I \) ratio range has not been previously investigated.
In each of the 28 mixed-mode fracture conditions, three to four test specimens were used, with the exception of one case (notch depth = 3.8 mm) where only one specimen was available for some of the off-axis angles.

The tests were conducted in room temperature on a close-loop Instron tester with a load rate of 1800 Kg/min. The critical load at the onset of the kink crack was recorded on a strip chart. Figs. 4, 5 and 6 show the experimental plot of critical laminate stress versus the off-axis angle $\theta$ at the onset of the kink crack for specimens of side-notches 2.5 mm, 3.2 mm and 4.5 mm deep, respectively. The case for $a = 3.8$ mm is not shown because of insufficient numbers of test specimens.

It is seen from the test results that the critical stress, $\sigma_{cr}$, at the onset of the kink decreases sharply with the off-axis angle $\theta$ when the notch depth is held constant. Similarly, the critical stress also decreases with the increase of the notch depth, $a$ when the angle $\theta$ is held constant.

Post-test SEM examination of the fractured surfaces under 500x to 1000x magnifications revealed extensive fiber breaking in the wake of the kink. Fig. 7 presents two such pictures taken near the kink point. Fiber breaks are visible in all cases. It is believed that the observed fiber breakage is due to the good bond between the matrix and the fiber, resulting in fiber nesting and/or fiber bridging across the kink path.

**Finite Element Analysis**

The experimental mixed-mode kinking problem is next simulated by the finite element routine. As mentioned earlier, the simulation model must be adjudicated for its accuracy. In the interest of conciseness, however, details of this development will not be discussed in this paper. Interested readers are referred to Ref. [7].

Return to the off-axis doubly side-notched coupon section shown in Fig. 2. The unidirectional laminate will be assumed an elastic, homogeneous and
orthotropic plate having constants in the principal material coordinates \((L,T)\) determined as follows:

\[
E_L = 145 \text{ Gpa} \quad E_T = 10.3 \text{ Gpa} \quad G_{LT} = 6.7 \text{ Gpa} \quad \nu_{LT} = 0.3
\]

Now, let the coupon be loaded by the far-field strain, \(e_x\). At some critical value of \(e_x\), the stresses near one of the side-notch tips are assumed to cause a kink emanating from the notch tip and propagate stably in the direction of the fibers. Of interest is when the length of the kink is small compared to the notch depth \(a\). Then, the mixed-mode strain energy release rates \(G_I\) and \(G_{II}\) at the kink tip are assumed to control the behavior of the initial kink. The values of \(G_I\) and \(G_{II}\) are calculated by the finite element routine via a crack-closure technique. These can be conveniently expressed in terms of the applied far-field strain in the form:

\[
G_I = C_I(e_x)^2 \quad G_{II} = C_{II}(e_x)^2
\]

where \(C_I\) and \(C_{II}\) are coefficients from the finite element calculations.

Figs. 8 and 9 show, respectively, the coefficients \(C_I\) and \(C_{II}\) plotted against the off-axis angle \(\theta\), and with the side-notch depth \(a\) as an independent parameter. It is seen that the kink is mixed in fracture modes for off-axis angles up to \(30^\circ\). Beyond \(30^\circ\), the fracture is essentially mode-I. Variation of the mixed-mode ratio, \(C_{II}/C_I\), with the off-axis angle \(\theta\) is shown in Fig. 10. This ratio depends principally on \(\theta\), and is almost independent of the notch depth \(a\).

Since for each test coupon the critical stress \(\sigma_{cr}\) at the onset of the kink was measured experimentally. The corresponding critical strain \((e_x)_{cr}\) can be calculated by dividing \(\sigma_{cr}\) by the coupon's axial modulus, \(E_x\). Then, using the values of \(C_I\) and \(C_{II}\), the critical strain energy release rates \((G_I)_{cr}\) and \((G_{II})_{cr}\) at
the initial kink for each test case can be calculated via Eq. 1.

For test cases where $G_I$ dominated, the deduced $(G_I)_{cr}$ is clearly $G_{IC}$. However, for the cases where both mode-I and mode-II were present, a combination of $(G_I)_{cr}$ and $(G_{II})_{cr}$ in some form would control the behavior of the kink. Fig. 11 is a diagram depicting the interactions between $(G_I)_{cr}$ and $(G_{II})_{cr}$ determined from all the test cases.

Though the test data show some degree of scatter, the overall trend indicates that the total strain energy release rate $(G_T)_{cr}$ remain more or less a constant. This strongly suggests that $(G_T)_{cr}$ or $G_{TC}$ essentially controls the behavior of the kink, including the special case of mode-I fracture.

Of course, this suggestion is based only on mixed-mode fracture data with $G_{II}/G_I$ ratios ranging from 0 to about 3. In this range, pure mode-II or predominantly mode-II fracture is not included.

It is also noted that, for graphite-epoxy composites, critical strain energy release rate data for matrix fracture have mostly been limited to $G_{IC}$. Generally, the measured values for $G_{IC}$ lie in the range between 120 to 260 J/m² depending on the material system used. In this study, $G_{IC}$ has the value in the order of 300 J/m². This seems to be on the high side compared to most other accepted values. However, in the present tests, fiber breakage in the wake of matrix cracking was detected in all cases. This could account for the higher measured value for $G_{IC}$.

CONCLUDING REMARKS

In this paper, mixed-mode matrix fracture in graphite-epoxy composites has been studied using a doubly side-notched, unidirectional off-axis specimen. This specimen has a configuration which is simple to fabricate and versatile in
geometrical variation. As a result, a total of 28 mixed-mode fracture conditions could be produced, which yielded a set of $G_{11}/G_{1}$ ratios covering uniformly from 0 to about 3.

Based on this data, a more definitive conclusion could be reached regarding the criterion for mixed-mode matrix fracture. Specifically, the total strain energy release rate $G_{Tc}$ appears to be a suitable criterion. This criterion, however, may not be applicable to pure mode-II or predominantly mode-II matrix fracture. The latter may involve additional energy dissipating mechanisms such as friction. If so, a separate criterion may be necessary.

**REFERENCE**


Fig. 1. X-radiograph of matrix crack development in a notched [0₂/90₂]s graphite-epoxy laminate loaded in axial tension
Figure 2    Geometry of the double side-notched specimen.
Figure 3  Geometry of kink cracks in the tested specimen.
Figure 4  Critical stresses at onset of kink crack. \( a = 2.54 \) mm.
Figure 5 Critical stresses at onset of kink crack. \( a = 3.18 \text{ mm} \).
Figure 6  Critical stresses at onset of kink crack. $a = 4.45\, \text{mm}$. 
Figure 7  Photomicrographs of fractured surface near kink point. Above: $\theta = 0^\circ$, $a = 3.81 \text{ mm}$; below: $\theta = 5^\circ$, $a = 2.54 \text{ mm}$. 
Figure 8  Mode-I strain energy release rate coefficients as function of off-axis angle \( \theta \).
Figure 9  Mode-II strain energy release rate coefficients as function of off-axis angle \( \theta \).
Figure 11 Interaction diagram of mixed-mode strain energy release rate data.
A COMPREHENSIVE STUDY
ON DAMAGE TOLERANCE PROPERTIES OF
NOTCHED COMPOSITE LAMINATES

Appendix III

Three-Dimensional Simulation of Crack Growth
in Notched Laminates

Three-Dimensional Simulation of Crack Growth in Notched Laminates

A. S. D. Wang, E. S. Reddy and Yu Zhong

ABSTRACT

This paper discusses the matrix cracking sequence in a \([02/90_2s]\) graphite-epoxy laminate with double-side notches. Experiments were performed on specimens loaded in uniaxial, quasi-static tension. The specimens were inspected at ascending load increments by x-radiography for patterns of matrix cracks caused by stress concentration near the notched region.

A numerical procedure based on a 3-D finite element method was then developed to simulate the observed matrix crack initiation, crack interaction and load-dependent crack growth sequence. The simulation begins with an analysis of the 3-D stress field near the notched region. This is followed by a search of possible modes of matrix cracking and the associated condition for propagation. The concept of brittle fracture is invoked to provide the necessary criterion for identifying the appropriate cracking modes and for determining the associated critical loads for their initiation. A comparison between experiment and prediction is presented.

INTRODUCTION

For a class of structural laminates, initial material damage involves two basic forms of matrix cracking [1]. One form is referred to as intraply cracking where a ply, or a layer of several plies of like fiber orientation, suffers a through-the-thickness crack along the fiber direction. Take the \([02/90_2s]\) laminate coupon under uniaxial tension as an example. A fiber-wise crack in the inner 90°-layer, known as transverse cracking, is a case of intraply cracking.

A. S. D. Wang, E. S. Reddy and Yu Zhong, Professor, Post-doctoral Fellow and graduate student, respectively, Department of Mechanical Engineering and Mechanics, Drexel University, Philadelphia, Pa. 19104
Similarly, a fiber-wise crack in the outer 0°-layer, known as longitudinal splitting, is also a case of intraply cracking. The other basic form is referred to as interply cracking where two adjacent plies suffer a separation in their interface. A delamination in the 0/90 ply interface of the [02/902]s laminate coupon mentioned above is a case of interply cracking. These two basic forms of matrix cracking may occur independently or interactively, depending on the manner of loading and the lamination structure [2]. Generally, one or both of these cracking modes occur before the load-carrying fibers break.

The initiation and growth mechanisms of intraply and interply matrix cracks, when occurring independently, have successfully been described within the framework of anisotropic ply elasticity and the fracture theory of brittle cracks [3,4]. A 3-dimensional treatment based on the same analysis concept was recently applied to laminates where the two basic cracking modes occur interactively [5]. In these previous studies, the laminate configuration was that of a straight flat coupon, where free-edge effects dominated the mechanisms.

In this paper, we use laminate coupons with double-side notches to study the formation of interactive matrix cracks that emanate from the notch rather than from the free edge. Since the notch is orientated normal to the applied tension, a very strong stress concentration is induced near the notch-tip. Thus, the intensity of concentration is sensitive to the depth of the notch and alters the matrix cracking characteristics.

Experiments were performed on specimens made of a graphite-epoxy laminate in the form of [02/902]s tension coupons with side-notches of various depths. For each test specimen, matrix cracking patterns near the notch-tip were inspected by x-radiography at prescribed ascending load increments in order to obtain a load-sequence of the matrix cracking events.

A numerical procedure based on a 3-D finite element method was then developed to simulate the observed load-dependent crack growth. The simulation is based on the strain energy release rate analysis method for non-interactive matrix cracking [3,4] and interactive matrix cracking [5].

A comparison is made between the predicted load-sequence of events and those recorded experimentally for specimens of different notch depths.

EXPERIMENT

The material used in the experiment was the AS4-3501-06 graphite-epoxy unidirectional system. [02/902]s laminate panels were made using an autoclave curing procedure. Test coupons were cut from these laminate panels, with dimensions of 25.4 mm wide and 228.6 mm long; the specimen thickness was about 1.016 mm. Double side-notches were introduced at the mid-section.
of the coupon by an 8-mil (0.008 in.) diamond saw. Specimens of four notch depths were so prepared (2.54 mm, 3.175 mm, 3.81 mm and 4.445 mm).

Tensile loading was applied to the test specimen through an Instron tester with the cross-head speed set at 0.25 mm per minute. At prescribed ascending load increments, the specimen was x-radiographed at the notched section in order to determine the developing matrix cracking patterns.

For all the specimens tested, the x-radiographs revealed three major forms of matrix cracking during loading. In order of their occurrence, these include longitudinal splitting in the $0^\circ$-layer which emanates from the notch-tip, transverse cracks in the $90^\circ$-layer along with the progression of $0^\circ$-layer splitting and, at some higher load, $0/90$ interface delamination growing stably along the length of the $0^\circ$-layer splitting boundary.

Fig. 1 is a sketch of the developing cracking pattern from a specimen with side-notch 3.175 mm deep. It is seen that at the laminate stress of 112 Mpa, a pair of $0^\circ$-layer splits of measurable length emanated from the notch-tip. Initially, the split at one notch-tip grew upward while the split at the other notch-tip grew downward. The growth was extremely stable. At 172 Mpa, splits in four directions emerged from the notch-tips; and a few $90^\circ$-layer transverse cracks appeared between the parallel splits. The $0^\circ$-layer splits grew in length while the $90^\circ$-layer transverse cracks grew in numbers as the laminate stress increased; see sketch corresponding to 259 Mpa. Then, while the splits were still growing, a measurable $0/90$ interface delamination initiated along the split boundary near the notch-tip, see sketch corresponding to 319 Mpa. The delamination grew stably as the laminate stress increased; see sketch corresponding to 345 Mpa. The specimen ruptured through the notch section at laminate stress well beyond 600 Mpa.

Fig. 2 is a plot of the measured length of the $0^\circ$-layer split versus the laminate stress, using data from two test specimens having notch depth of 3.175 mm. The scatter in the data is due to variation of the split lengths in four directions. The mean length is taken as the average of the splits in four directions. The laminate stress levels at which $0^\circ$-layer splitting, $90^\circ$-layer transverse cracking and $0/90$ interface delamination initiated were all recorded.

Fig. 3 is a plot of the measured $0/90$ interface delamination (in area) versus the laminate stress from the same two test specimens. The delamination area at different load increments were measured from prints of x-radiographs using an Lemont Scientific image analyzer. The procedure involves magnification of the delamination area by a high resolution video camera which traverses the contour of the delamination. The scatter of the measured values is due to the variation in areas from the four branches of delamination. From the plot, onset of $0/90$ interface delamination may be extrapolated. In this case, delamination onset had occurred at about 260 Mpa.
Table 1 summarizes the onset stresses of the three major forms of matrix cracking from specimens of four notch depths. It is seen that for each form of matrix cracking, onset stress decreases with increase of notch depth. This is expected because the deeper the notch the larger is the stress concentration at the notch-tip.

SIMULATIONS

The Finite Element Model. To simulate the specimen used in the experiment, let us consider the [02/902]s laminate having double side-notches at regular interval as shown in Fig. 4a. Assume that these double side-notches are spaced so far apart that they do not interact with one another. Then a periodic element of the laminate which contains only one pair of notches is isolated as shown in Fig. 4b. This element thus represents the test specimen. Note that the laminate is symmetric with respect to the laminate mid-plane (the x-y plane), and the y-axis lies in the plane of the notches. Hence, it is sufficient to model one-eighth of the element shown in Fig. 4b. A schematic finite element mesh is shown in Fig. 4c. Due to expected stress concentration near the notch-tip and ply interfaces, a finer mesh is always deployed in these regions.

The finite element routine was developed based on the assumption that the unidirectional ply is an elastic, homogeneous and orthotropic medium. The elastic and other pertinent material constants for the AS-3501-06 system were characterized by routine tests [6], and their values are listed in Table 2. Solutions for stresses and other quantities, such as strain energy release rates, were obtained by employing a 21-node brick element. The actual computation was carried out on VAX-11/750 and Cray X/MP computers. These and other computational details are found in [7].

Notch-Tip Stress Fields. The laminate stress fields were calculated for two types of loading. The first is by prescribing a far-field laminate strain of $\varepsilon_x = 10^{-6}$, and the other is by prescribing a uniform temperature change of $\Delta T = -10^\circ C$. Stresses due to applied laminate tension (by giving a value for $\varepsilon_x$) and laminate post-cure cooling (by giving a value for $\Delta T$) can then be obtained by superposition.

Although there are six stress components at each finite element node, it is of interest to examine only those components that are responsible for the observed matrix cracking initiation.

First, let us examine $\sigma_y$ in the $0^\circ$-layer. This stress is thought to cause $0^\circ$-layer split, which in fact was observed as the first mode of failure under a very low tensile loading. For the case of $\varepsilon_x = 10^{-6}$, $\sigma_y$ is tensile throughout the thickness of the $0^\circ$-layer near the notch region. Its value varies from the top to
the bottom of the layer, with the minimum occurring near the 0/90 interface. Fig. 5a shows the $\sigma_y$ distribution in the $0^\circ$-layer near the 0/90 interface for the specimen having notch depth of 3.175 mm. A sharp rise of $\sigma_y$ in tension is seen to occur at the notch-tip, displaying a singular behavior. A similarly behaved in-plane shear stress $\tau_{xy}$ is also present at the notch-tip; it's planar distribution near the 0/90 interface is shown in Fig. 5b. The concentration intensities of $\sigma_y$ and $\tau_{xy}$ at the notch-tip are about the same.

For the case of $\Delta T = -1^\circ C$, $\sigma_y$ is also tensile throughout the $0^\circ$-layer. Fig. 5c shows the $\sigma_y$ distribution in the $0^\circ$-layer near the 0/90 interface. Here, stress concentration due to the notch is much less. But, by the magnitude of this stress throughout the $0^\circ$-layer is quite large. Thus, the combined tensile and thermal loading will cause the $0^\circ$-layer splitting to be in mixed modes.

Next, let us examine $\sigma_x$ in the $90^\circ$-layer. This stress causes $90^\circ$-layer transverse cracking. Again, for the case of $e_x = 10^{-6}$, this stress is tensile and varies throughout the thickness of the $90^\circ$-layer near the notch region, with the minimum occurring near the 0/90 interface and the maximum at the mid-plane. Fig. 6a shows the $\sigma_x$ distribution in the $90^\circ$-layer near the laminate mid-plane for the specimen having 3.175 mm notch depth. It is seen that a sharp tensile stress is again developed at the notch-tip.

Similarly, for the case of $\Delta T = -1^\circ C$, $\sigma_x$ in the $90^\circ$-layer is also tensile with significant magnitude; but stress concentration caused by the notch is minimal, Fig. 6b. Other stress components also exist in the $90^\circ$-layer near the notch-tip; but their magnitudes appear to be negligible.

Finally, the nature of the interlaminar stresses ($\sigma_z$, $\tau_{xz}$, $\tau_{yz}$) should be examined because these stresses are responsible for interface delamination. For the same specimen considered under $e_x = 10^{-6}$ loading, its $\sigma_z$ distribution on the 0/90 interface is shown in Fig. 7a, while distribution on the 90/90 plane is shown in Fig. 7b. It is seen that $\sigma_z$ can be tensile and of significant magnitude; but it exists only near the notch-tip. As for $\sigma_z$ caused by thermal cooling, the associated magnitude for $\sigma_z$ is relatively small. Similarly, the interlaminar shear stresses, $\tau_{xz}$ and $\tau_{yz}$ also exist with highly localized magnitudes at the notch-tip.

From the above analysis, it appears that $0^\circ$-layer splitting and $90^\circ$-layer transverse cracking are equally likely to occur, while the likelihood for interface delamination is comparatively smaller. However, judgement regarding relative occurrence of these cracking events cannot be made based on the computed stresses, as they all display some degree of stress concentration. In what follows, we attempt to simulate the onset of the observed cracking modes from a fracture point of view.

**Simulation of $0^\circ$-Layer Splitting.** To simulate the initiation and growth of $0^\circ$-layer splitting, we shall assume that $90^\circ$-layer transverse cracking will
not simultaneously occur. Then, at the notch-tip, we issue a small $0^0$-layer split of length $s_0$ as shown by the insert in Fig. 8. This small split represents an effective flaw which exists at the notch-tip and propagates to become a $0^0$-layer split whenever a certain condition is reached. Under a constant far-field strain loading, the split is assumed to propagate stably to reach a length $s > s_0$. Thus, the finite element simulation is to calculate the split-tip stresses and the associated fracture quantity. For the latter, we calculate the split-tip strain energy release rate $G$ as a function of the split length, $s$.

As was mentioned earlier, the tensile normal stress $\sigma_y$ and the in-plane shear stress $\tau_{xy}$ in the $0^0$-layer are the major stress components causing splitting. Fig 8 is a plot of the split-tip stress $\sigma_y$ versus the split length, $s$, for the specimen having 3.175 mm side-notches subjected to $e_x = 10^{-6}$ loading. It is seen that $\sigma_y$ is larger when $s$ is small, but it decreases sharply with increase of $s$. On the other hand, the associated shear stress $\tau_{xy}$ (not shown) became relatively more dominant with increasing $s$.

When subject to thermal cooling of $\Delta T = -10^0C$, $\sigma_y$ in the $0^0$-layer is also tensile, see Fig. 5b. But variation of $\sigma_y$ at split-tip due to growth of split is rather insignificant.

To facilitate a prediction for the load versus split-growth relationship, we then calculate the split-tip strain energy release rate, $G(s)$. This quantity is conveniently expressed in terms of the loads $e_x$ and $\Delta T$ [5]:

$$G(s) = \left[\sqrt{C_e e_x} + \sqrt{C_t \Delta T}\right]^2 d$$

(1)

where $\Delta T$ represents thermal cooling and $d$ is a length scale which is set at unity in this study. The coefficients $C_e$ and $C_t$ are functions of $s$ and represents the strain energy release rates, corresponding to $e_x=1$ and $\Delta T=-10^0C$, respectively.

Figs. 9a and 9b show, respectively, the coefficients $C_e$ and $C_t$ versus the split length $s$ for the specimen with 3.175 mm notch depth. It is seen that $C_e$ and $C_t$ both contain mixed modes. However, $C_e$ is predominantly of mode-II, while $C_t$ predominantly of mode-I. When the two load agencies are combined, as in Eq. (1), a mixed-mode cracking of approximately equal ratio results. Note that the overall strain energy release rate is one which decreases with the split length, $s$; This indicates a stable splitting growth, a behavior consistent with that observed in the experiment.

The load versus split-growth relation is derived from the fracture criterion,

$$G(s) = G_c$$

(2)

where $G_c$ is the critical strain energy release rate for mixed mode cracking. Assume that the value of $\Delta T$ is given. Then, by combining Eqs. (1) and (2) we
obtain the critical laminate strain $(\varepsilon_x)_{cr}$ as a function of split length, $s$.

For the material system used in this study, $\Delta T$ and the mixed mode $G_c$ have been determined elsewhere [6], and their values are listed in Table 2. Thus, the predicted $(\varepsilon_x)_{cr}$ can be converted to $(\sigma_x)_{cr}$. For the specimen just considered, the computed $(\sigma_x)_{cr}$ versus $s$ relations is shown by the solid line in Fig 2. It is seen that the predicted result agrees well with the initial portion of the experimental split-growth data, where the splitting was not yet significantly complicated by the development of $90^\circ$-layer transverse cracks. The predicted curve, however, departs away from the observed results as $90^\circ$-layer transverse cracks developed in higher density. To include the effects of these transverse cracks on split growth will require a major modification of the simulation model.

The predicted onset stresses for $0^\circ$-layer splitting for specimens of four notch depths are listed in Table 1 along with their experimental counterparts. In all cases, the model seems to predict well the initiation of the splitting.

**Simulation of $0/90$ Interface Delamination.** Delamination of the $0/90$ interface takes place at a much higher load. Once initiated, it grows stably along the boundary of the $0^\circ$-layer splits. The delamination pattern is shown schematically in Fig 10. As we have observed in the experiment, $90^\circ$-layer transverse cracks actually formed continuously as the $0/90$ interface delamination grew, see Fig 1. An analytical/computational simulation of this complex interactive cracking phenomenon, though not impossible, is quite tedious and probably not fruitful. Thus, a simplified version is attempted instead. Namely, we shall assume that only $0^\circ$-layer splitting precedes the initiation of the $0/90$ interface delamination and the effect of $90^\circ$-layer transverse cracking is negligible.

The simulation follows a similar procedure as used for $0^\circ$-layer splitting. A set of densely meshed finite elements is deployed near the intended delamination region, double nodes are assigned on the plane of delamination and these are then released in sequence so as to mimic the actual growth pattern observed in the experiment. In the node-releasing process, the fracture energy release rate coefficients, $C_e$ and $C_t$ are computed as functions of the delaminated area [7]. Figs. 11a and 11b show, respectively, the computed $C_e$ and $C_t$ coefficients versus delamination area for the specimen having the notch depth of 3.175 mm.

From the energy release rate curves, it is seen that the delamination is primarily of mode-I under the applied tensile loading $(\varepsilon_x=1)$, while primarily in mode-I and II under thermal cooling $(\Delta T=-10^\circ \text{C})$. Thus, the combined effect is again one of mixed modes. The overall energy release rate, however, decreases sharply with increasing delamination area, indicating a stable growth. This is
also consistent with the behavior observed in the experiment.

Using the computed energy release rate coefficients, stress levels corresponding to the prescribed delamination node-releasing sequence can be predicted by means of Eqs (1) and (2). For example, in the case of the specimen having notch depth of 3.175 mm, the predicted delamination growth curve is shown by the solid line in Fig. 3. Here, again, the agreement between prediction and experiment is quite close for the initial portion of the delamination growth. Apparently, as the delamination grows larger, many transverse cracks are formed in the 90°-layer, the associated cracking mechanisms then becomes more complicated than the model has portrayed.

The critical stresses for 0/90 delamination in specimens having other notch depths were also computed. These are listed in Table 1 for comparison with their experimental counterparts.

CONCLUSIONS

In this paper, we have presented a method of simulation for matrix cracks that develop in laminate specimens having double side-notches. The analysis entails a 3-D stress analysis and computer simulation of fracture growth near the notched region. The purpose of the study is to understand the damage mechanisms at a level below the lamination structure. The actual specimen chosen for analysis could represent a critical element [8] in a large laminated structure whose global strength and/or fatigue properties are to be evaluated.

Acknowledgments: Results reported in this paper were obtained during the course of research supported by a grant from the Air Force Office of Scientific Research.

REFERENCES


### Table 1. Experimental and Predicted (in parenthesis) Onset Stresses

<table>
<thead>
<tr>
<th>Notch Depth</th>
<th>2.54 mm</th>
<th>3.175 mm</th>
<th>3.81 mm</th>
<th>4.445 mm</th>
</tr>
</thead>
<tbody>
<tr>
<td>0°-layer</td>
<td>100 Mpa</td>
<td>75 Mpa</td>
<td>60 Mpa</td>
<td>60 Mpa</td>
</tr>
<tr>
<td>Split</td>
<td>(80 Mpa)</td>
<td>(70 Mpa)</td>
<td>(60 Mpa)</td>
<td>(60 Mpa)</td>
</tr>
<tr>
<td>90°-layer</td>
<td>170 Mpa</td>
<td>160 Mpa</td>
<td>150 Mpa</td>
<td>150 Mpa</td>
</tr>
<tr>
<td>Transverse</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Crack</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0/90 interface</td>
<td>350 Mpa</td>
<td>260 Mpa</td>
<td>225 Mpa</td>
<td>220 Mpa</td>
</tr>
<tr>
<td>Delamination</td>
<td>(300 Mpa)</td>
<td>(230 Mpa)</td>
<td>(200 Mpa)</td>
<td>(180 Mpa)</td>
</tr>
</tbody>
</table>
Table 2. Pertinent Material Constants for AS4-3501-06 UD Ply

\[
\begin{align*}
E_{LL} &= 145 \text{ Gpa} & E_{TT} &= E_{ZZ} = 103 \text{ Gpa} & G_{LT} &= G_{LZ} = 6.8 \text{ Gpa} & G_{TZ} &= 3.5 \text{ Gpa} \\
\nu_L &= \nu_{LZ} = 0.3 & \nu_{TZ} &= 0.54 & \alpha_L &= 0.4 \times 10^{-6} / \degree C & \alpha_T &= \alpha_Z = 28.8 \times 10^{-6} / \degree C \\
\\
\Delta T &= -140 \degree C & (G_C)_{\text{total}} &= 289 \text{ J/m}^2 & \text{Ply Thickness} &= 0.127 \text{ mm}
\end{align*}
\]

Fig. 1. Development of Matrix Cracks in Specimen of Notch Depth 3.175 mm

Fig. 2. Split Length Growth versus Applied Tension (Notch Depth, 3.175 mm)
Fig. 3. Delamination Area Versus Applied Tension (Notch Depth 3.175 mm)

Fig. 4. Finite Element Model for the Double Side-Notched Specimen.
Fig 5. (a) $\sigma_y$ Distribution in the 0°-Layer ($e_x=10^{-6}$)  
(b) $T_{xy}$ Distribution in the 0°-Layer ($e_x=10^{-6}$)  
(c) $\sigma_y$ Distribution in the 0°-Layer ($\Delta T=-10^\circ C$)

Fig 6. $\sigma_x$ Distribution in 90°-Layer due to (a) $e_x=10^{-6}$ and (b) $\Delta T=-10^\circ C$
Fig. 7. $\sigma_z$ Distribution on (a) 0/90 and (b) 90/90 interface.

Fig. 8. $\sigma_y$ at Split-tip versus Split Length, $s$.

Fig. 9. Energy Release Rates at Split-tip (a) $C_e$ and (b) $C_t$.

$C_e$ (J/sq.m) and $C_t$ (J/sq.m/$^\circ$C$^2$) for $\Delta T = -1^\circ$C.
Fig. 10. Finite Element Simulation of 0/90 interface Delamination.

Fig. 11. Energy Release Rate at Delamination Front: (a) for $e_x = 1$, (b) $\Delta T = -1^\circ C$
A COMPREHENSIVE STUDY
ON DAMAGE TOLERANCE PROPERTIES OF
NOTCHED COMPOSITE LAMINATES

Appendix IV

Simulation of Matrix Cracks in Composite Laminates
Containing a Small Hole

Paper presented at the ASME Winter Annual Meeting, Boston, 1987;
SIMULATION OF MATRIX CRACKS IN COMPOSITE LAMINATES CONTAINING A SMALL HOLE

E. S. Ruddy, A. S. D. Wang, and Y. Zhong
Drexel University
Philadelphia, Pennsylvania

ABSTRACT

This paper studies the matrix cracking sequence in \([0/90]_2\) graphite-epoxy laminates that contain a small central hole. Experiment was performed first using specimens loaded in uniaxial, quasi-static tension, followed by inspections of the specimen at several prescribed loading increments by means of x-radiography. The inspection provides a qualitative measurement and a physical analysis of matrix cracking patterns near the hole.

A numerical procedure based on a three dimensional finite element method was then employed to simulate the observed matrix cracking patterns, including their initiation and growth behaviors. Here, the theory of ply elasticity and the concept of brittle fracture are used as basis for the finite element simulation.

A comparison between the simulated and the experimental results is presented.

1. INTRODUCTION

Failure analysis of fiber-reinforced composites has attracted increased interest in recent years. Application of composites in high-performance aircraft and spacecraft structures has led the researchers to carry out intensive experimental and theoretical studies on failure mechanisms in a variety of composite materials. For a special class of composites, namely, polymeric laminates made by laminating unidirectional continuous fiber systems, failure initiation usually involves some forms of matrix cracking.

When viewed at the laminating ply level, these can be classified into two basic forms. One basic form is known as intraply cracking, where a ply or several plies of the same fiber orientation that formed a layer, suffers a through-the-thickness crack along the fiber direction. A simple example of intraply cracking is found in a \([0/90]_2\) type laminate under axial tension, in which the inner 90°-layer suffers multiple transverse cracks. The other basic form is interply cracking, where two adjacent plies in the laminate suffer a separation in their interface. Free edge delamination in a \([+-/90\pm0]_2\) laminate loaded in axial tension, for instance, provides a case of interply cracking.

Studies of the individual growth mechanisms of the two basic forms of matrix cracking have been extensively reported in the literature (see, e.g., [1,2]). Interactions of the two basic forms of cracking were examined partially in [3,4]. The problems studied in [1-3] concerned cracking development in plain laminates, while the problem studied in [4] involved laminates that contain sharp through-the-thickness notches.

The problem of laminates with a through-hole has also attracted considerable attention. Effects of ply stacking sequence [5] and different material combinations [6] on global laminate strength reduction due to presence of a small hole were among the early interests. Subsequent analyses have focussed on the detailed stress distribution around the hole, especially the interlaminar stresses that cause local delamination [7-9]. In these works, delamination (interply cracking) is assumed to take place as the only matrix cracking mode. Experiments using graphite-epoxy laminates have shown, however, that the first matrix cracking form near the hole is usually not delamination.

In the present study, we use a \([0/90]_2\) graphite-epoxy laminate with a small central hole to examine the initiation and growth patterns of matrix cracks near the hole. In this case, a three dimensional stress analysis based on ply elasticity performed, which shows that severe stress concentrations along the hole boundary are present, and matrix cracks of different forms may initiate and propagate at these locations. At the same time, experiments performed on test specimens and inspected by x-radiography at different loading levels reveal the exact sequence of the various cracking events. Thus, the purpose of this study is to relate the experimental events with the analysis by means of a finite element simulation. A comparison is then made between the simulated and the experimental results.
2. EXPERIMENT AND RESULTS

In the experiment, test coupons were made from AS4-3501-06 graphite-epoxy prepreg tapes. The lamination stacking sequence was limited to [0°/90°]. The dimensions of the test coupons were 25.4 mm wide, 228.6 mm long and 1.016 mm thick. The radius of the central hole was 3.175 mm. Loading was applied axially on an Instron tester, with the cross-head speed set at 0.25 mm per minute. The loaded specimens were periodically inspected by DIB enhanced X-radiography.

Experimental results show three major forms of matrix cracking that emanate from the hole during loading. In their order of occurrence, these are (1) horizontal transverse cracks (intraly cracking) in the inner 90°-layer in the immediate region of the hole initially, and away from the hole region subsequently; (2) vertical splitting cracks (also a form of intraly cracking) in the outer 0°-layers emanating from the hole and propagating stably away from the hole; and (3) delamination (interply cracking) in the 0/90 interface along the length of the 0°-layer splits, which displays a very stably growth behavior. Figure 1 is a schematic illustration of the cracking development patterns at five typical laminate stress levels. It can be seen that a few (two or three) 90°-layer transverse cracks and an initial sign of 0°-layer splits are present at about 276 MPa. At 379 MPa, four branches of 0°-splitting have already formed and propagated stably towards the top and bottom of the specimen. Note that propagation of the splits are accompanied by more 90°-layer transverse cracks, see illustration at 465 MPa. At 552 MPa, delamination in the 0/90 interface has already occurred along each of the four 0°-layer splits. While the delamination grows stably with load, more transverse cracks have formed and the longer the 0°-spits have grown, see illustration at 724 MPa. At this load level, matrix-related damage around the hole is substantial; but no significant fiber breakage has yet occurred. In fact, the specimen can sustain a laminate stress of more than 1000 MPa before it breaks completely through the hole section.

To express quantitatively the damage matrix cracking, we choose to display two separate cracking quantities in terms of the applied laminate stress. The first is the linear length of the 0°-layer split and the second is the area of the 0/90 interface delamination. Since for each specimen there are four branches of splits, which grow stably with load, a mean length is obtained from measuring all four splits at each load interval. Figure 2 shows the mean split length plotted against the laminate stress (in MPa), where the data are from a sample of six specimens. It is seen that the mean split grows almost linearly with the applied laminate stress; and by extrapolation of the data, we can deduce that the onset stress for 0°-layer split is at about 120 MPa. The solid line in the figure represents the simulated split growth. The adequacy of the simulated result will be discussed in the next section.

Similarly, Figure 3 shows the mean delamination area measured from the same six specimens (the areas were measured from the X-radiographs using an image analyzer). Here again, we see that the delamination growth rate is quite slow initially but becomes rapid as the laminate stress is increased. Data extrapolation yields the onset stress at about 220 MPa. The solid line in this figure is the simulated results. However, we shall defer the discussion on simulation later in the next section.

3. SIMULATIONS AND RESULTS

The Finite Element Model

The problem of a laminate with a single central hole stems from the large composite structural panel with bolt or rivet holes. In these large structural laminates, the holes are often periodically placed. Assuming that the holes are located so far apart that they do not interact with each other, then a periodic element of the panel containing only one hole can be considered for analysis, see Figure 4a. For the problem considered here, the lay-up of the specimen is [0°/90°] and the hole is placed at the laminate center. Thus, it is sufficient to model one-eighth the specimen due to symmetry as shown in Figure 4b. Since stress concentration is expected around the hole boundary, a finer finite element mesh is deployed in this region in order to capture the true nature of stress concentration.

It is noted that the basis of the finite element analysis is the assumption of ply elasticity, that is that the graphite-epoxy unidirectional ply is assumed as an elastic, homogeneous and orthotropic medium. The elastic and the thermal expansion constants of the AS-3501-06 ply system were characterised and given in [11]. The basic finite element is a 21-node solid brick and the computation is performed on a CRAY X/MP computer. Details of the computational procedures are contained in a separate user's manual [12].

Stresses Near the Hole Boundary

The laminate stress fields are calculated for two types of loading, the first is by prescribing a far-field laminate strain of $\varepsilon = 10^4$, and the other is by prescribing a uniform temperature change of $\Delta T = -1^\circ C$. Stress due to combined tension and temperature change can be obtained by superposition. The stress field near the hole boundary is in a complicated three-dimensional state. It is of interest here to examine all of the stress distributions in detail. Rather, we shall display only some typical ones that are thought to cause matrix cracking.

Figure 5 is a display of the $xy$-plane distribution for the six stress components which exist in the central hole in the laminate $\phi_{x}$ which is in the fiber direction. But when compared to the ply strength in the fiber direction this stress is rather insignificant in causing failure. Other stresses that may cause matrix failure are $\tau_{xy}$ and $\tau_{yx}$. These two combined can cause longitudinal splitting in this layer. The interlaminar shear stresses $\tau_{zx}$ and $\tau_{zy}$ are relatively small but could precipitate 0/90 interface delamination.

Figure 6 is a similar display for the stresses in the 90° layer near the mid-plane of the laminate. Here, we see the dominance of $\sigma_{yy}$ which is normal to the fibers in this layer. Concentration near the hole region will be certain to cause transverse cracks. All other stresses however have secondary influence.

The above stress distributions are computed for tension
loading only. We also need the stress distribution due to thermal cooling of the laminate to evaluate the combined stress state. For the laminate used, $\Delta T$ is set at -140°C. For simplicity however, we shall omit the display of the thermal stresses here.

From the stress analysis, it appears that $0^\circ$-layer splitting and $90^\circ$-layer transverse cracking are equally likely to occur first. However, because of the high stress gradient near the hole boundary it is not possible to make a prediction as to which of these two forms of matrix cracking will first occur. In what follows, we attempt to simulate the onset and propagation of some of these cracking forms from a fracture point of view.

**Simulation of $0^\circ$-Layer Splitting**

In the simulation of $0^\circ$-layer splitting, it is assumed that $90^\circ$-layer transverse cracks are absent while the split grows with loading. This assumption is necessary to reduce the geometric complexity of the cracked laminate. It is felt that omission of the transverse cracks will not adversely affect the accuracy of the simulation, at least not the onset of splitting. To simulate, a small split length $s_0$ is introduced in the $0^\circ$-layer as shown by the inserted sketch in Figure 7. This small split represents an effective material flaw which exists at the hole boundary and propagates to become a split whenever the critical condition is reached. Under the far-field constant strain loading, the split is assumed to propagate stably in the fiber direction. Thus, the finite element routine is to calculate the stresses and the strain energy release rate $G$ at the split-tip as a function of split length $s$.

Figure 7 shows the variation of $\sigma_x$ at the split-tip with the split length $s$. It is believed that this stress is responsible for the initiation and continuation of the split. From the figure, we see that $\sigma_x$ is large when $s$ is small; but it decreases sharply with increase of $s$. On the other hand, the associated split-tip shear stress $\tau_{xy}$, which is not shown here, becomes relatively more dominant with increase in $s$. This indicates that once the split starts, it will propagate stably and in shearing mode.

The corresponding stresses due to thermal cooling are also calculated; their effect on splitting is included in the prediction, which is to be discussed below.

As mentioned, we first introduce a small split length $s_0$ and then calculate the split-tip strain energy release rate $G(s)$ as a function of $s \geq s_0$. $G(s)$ can be expressed in terms of the applied tension $\varepsilon_x$ and thermal loading $\Delta T$ as [5]:

$$G(s) = \left\{ (\sqrt{C_C \varepsilon_x} + \sqrt{C_T \Delta T})^2 \right\}_i, \quad i = I, II, III \quad (1)$$

where the coefficients $C_C$ and $C_T$ are functions of $s$ and represent the strain energy release rates corresponding to $\varepsilon_x = 1$ and $\Delta T = -1$ °C, respectively. The parameter $d$ is a length scale factor which is set at unity in this study. Finally, the subscript $i$ refers to cracking modes of I, II and III (open, sliding and antiplane shear).

Now, Figures 8 and 9 show, respectively, the coefficients $C_C$ and $C_T$ versus the split length $s$. Note that $C_C$ is predominantly of mode II, while $C_T$ is predominantly of mode I. Thus, the combined crack growth is in mixed mode.

The growth behavior of mixed mode matrix cracking is discussed in [11] and a criterion based on the total energy release rate is suggested:

$$\sum G(s) = G_c \quad (2)$$

where $G_c$ is the total critical strain energy release rate for mixed mode cracking. For the material used here, $G_c$ has a value estimated at 289 J/m².

By using the coefficient curves in Figures 8 and 9, we can obtain from (1) and (2) the critical laminate strain $\varepsilon_x$ as a function of split length $s$. The computed critical laminate strain can be readily converted in laminate stress; the stress versus $s$ relation is shown in Figure 2 by the solid line. It is seen that the calculated result agrees initially with the experiment, predicting the onset of splitting. As the split grows longer, the splitting mechanisms are complicated by transverse cracking and also by delamination. Since these complicating mechanisms are not included in the splitting simulation model, a discrepancy between the experiment and the prediction results.

**Simulation of 0/90 Interface Delamination**

In the simulation of the $0/90$ interface delamination, an idealization is also made. Namely, we assume that delamination occurs after the $0^\circ$-layer splitting has grown a sufficient length so that delamination is proceeding as an independent event. The simulation is carried out to mimic the delamination shape as observed in the experiment. The simulated shape is schematically shown in Figure 10. Here, we calculate the mean strain energy release rate coefficients at the delamination front as a function of the total delaminated area, see Figures 11 and 12. Then, by means of the criterion in (2) we obtain the delamination area versus laminate stress relation as shown in Figure 3 by the solid line. Again, the prediction for the onset of delamination is close, but discrepancy results once the delamination has grown larger. This is expected because the actual growth of delamination is concurrent with other forms of matrix cracking, as discussed earlier in the experimental study. This complex mechanics mechanisms was not included in the simulation model.

**4. CONCLUSIONS**

In this paper, we have shown that growth of matrix cracks in the vicinity of a small hole in a laminate can be reasonably simulated by a finite element routine based on a careful fracture mechanisms analysis. Still, the actual mechanisms are complicated and the simulation has to resort to some degree of idealization. This causes discrepancies between the simulated results and the experiment. It is conceivable that these discrepancies can be considerably removed if more is known about the physics of the phenomenon at the microscopic level and if a more powerful simulation technique is available.

**Acknowledgment:** Results reported in this paper were obtained during the course of research supported by a grant from the Air Force Office of Scientific Research.
REFERENCES


---

Figure 1. Progression of Matrix Cracking Under Ascending Laminate Stress
Figure 2. Splitting Length Versus Applied Laminate Stress.

Figure 3. Delamination Area Versus Applied Laminate Stress.

Figure 4. Finite Element Model for the Specimen Containing A Small Hole.
Figure 5. Distribution of Stresses in 0°-Layer Near 0/90 Interface.
Figure 6. Distribution of Stresses in 90°-Layer Near 90/90 Interface.
Figure 7. Split-Tip Normal (tensile) Stress Versus Split Length.

Figure 8. Energy Release Rate Coefficient $C_e$ for $0^\circ$-Layer Splitting.

Figure 9. Energy Release Rate Coefficient $C_e$ for $0^\circ$-Layer Splitting.

Figure 10. Simulation Model for 0/90 Interface Delamination.
Figure 11. Energy Release Rate Coefficient $C_\phi$ for 0/90 Interface Delamination.

Figure 12. Energy Release Rate Coefficient $C_T$ for 0/90 Interface Delamination.
A COMPREHENSIVE STUDY
ON DAMAGE TOLERANCE PROPERTIES OF
NOTCHED COMPOSITE LAMINATES

Appendix V

3-D Finite Element Crack Simulation Code
User's Guide and Source Code
## CONTENTS

1. **GENERAL PROGRAM CHARACTERISTICS** ..................................... 1.1
   1.1 Introduction ........................................................................ 1.1
   1.2 The Preprocessor Program .................................................. 1.2
   1.3 The Main Code: KSAP II .................................................... 1.2
   1.4 The Post Processor Program ................................................. 1.2

2. **THE STRUCTURE OF THE PREPROCESSOR PROGRAM** ..................... 2.1
   2.1 Introduction ........................................................................ 2.1
   2.2 Data Input of the Preprocessor Program ................................. 2.5
      2.2.1 Details of the Data Input ............................................. 2.5

3. **MODIFICATION OF THE PREPROCESSOR OUTPUT DATA** ................. 3.1
   3.1 Introduction ........................................................................ 3.1
   3.2 Double Nodes and Crack Opening Sequence ............................. 3.1
   3.3 Details of Data Modification ............................................... 3.3

4. **THE STRUCTURE OF THE MAIN CODE: KSAP II** .......................... 4.1
   4.1 Introduction ........................................................................ 4.1
   4.2 General Features of the Code .............................................. 4.1
   4.3 The Output Data from KSAP II ............................................ 4.3
   4.4 Limitations of KSAP II Code ............................................... 4.4

5. **POSTPROCESSOR PROGRAM** ................................................... 5.1
   5.1 Introduction ........................................................................ 5.1
   5.2 Details of Data File ............................................................ 5.1
6 ILLUSTRATIVE EXAMPLE......................................................6.1
6.1 Introduction...............................................................6.1
6.2 Preprocessor Input Data..................................................6.3
6.3 Modifications of Preprocessor Output Data..........................6.8
6.4 Output of KSAP II Program..............................................6.12

APPENDIX
A LISTING OF THE PREPROCESSOR.........................................A-1
B LISTING OF THE MAIN CODE 'KSAP II'....................................B-1
C LISTING OF THE POSTPROCESSOR, 'PLOT'.................................C-1
D LISTING OF THE EXAMPLE RESULTS.......................................D-1
1 GENERAL PROGRAM CHARACTERISTICS

1.1 INTRODUCTION

This computer code has been developed for an independent and self-contained operation. The program is written in FORTRAN 77 language, adaptable to any medium or large computer. The main function of the program is to simulate numerically the initiation and growth of a plane crack(s) in a 3-D solid, specifically, delamination or splitting or delamination with a split in composite plates. The plate may be subjected to either mechanical loading, thermal loading or both. In order to determine the layer interface which is likely to suffer delamination under the given loading, a search must be conducted by computing the interlaminar stresses. Once the site of delamination is determined, the program will then simulate the delamination growth under the applied loads.

The present computer code can handle (i) splitting along the fiber direction, (ii) delamination having a plane-contour of arbitrary shape and (iii) delamination in the presence of an opened split. The changes in the boundary conditions as the delamination grows are automatically adjusted in the program. There is no limitation to the number of layers or the stacking sequence. The layers may have different thicknesses and material properties. Each layer is assumed to be a homogeneous, orthotropic elastic medium with one of its principal axes aligned in the thickness directions of the plate (z-axis).

The code is divided into three independent programs: the preprocessor, the main code, and the post processor. The separation of the code in three stages allows modifications to be made in the data at the end of each stage.
of each particular program so that certain parametric studies can be performed in one stage without repeating the calculations performed in the previous stage.

1.2 THE PREPROCESSOR PROGRAM

This is the first stage in the solution of the delamination problem. The input data necessary for this program consists of the specimen geometry, mesh plan, layer material properties, boundary conditions and the double nodes (double nodes are a pair of nodal points which occupy the same spatial position). The output of this program consists of the full details of the finite element mesh together with the numbered nodes, including the double nodes. Although this output data is sufficient to run the second stage, the data to be input into the main code, the data still needs to be supplemented with the crack opening sequence data set which can be formulated only following the output from the preprocessor.

1.3 THE MAIN CODE, KSAP II

As the name implies, this is the main part in the solution procedure. The output data from the first stage, together with the crack opening sequence data serves as the input data for this program. The program solves the three dimensional problem using an 8 or 21 node solid element with three degrees of freedom (x,y,z) for each node.

1.4 THE POSTPROCESSOR PROGRAM

The post processor mainly produces 3-D plots of the stresses with hidden lines removed. The input data for this program is a modified
output file from the KSAP II program. Various stress distribution plots can be output along any specified plane. The three-dimensional plots can be processed at any specified viewing direction.

The details of preprocessor program and the input can be found in Chapter 2. The modifications to the preprocessor output which are needed before it can be used further are found in Chapter 3. Chapter 4 describes the features of the KSAP II code and Chapter 5 describes the details of postprocessor program. Chapter 6 contains an illustrative example to explain the working of the total code. The FORTRAN source listings of the various parts of the code and their outputs are included in the Appendices A through D.
FLOW CHART FOR 'PREPROCESSOR' PROGRAM
START

Modified Data from Preprocessor

Read the Mesh and Material Data along with Double Nodes from the Input File

Compute System Stiffness Matrix

Solve the Linear System for Displacement Fields with the Prescribed Boundary Conditions

Compute Nodal Forces and Stress Fields

Compute Components of Energy Released in X, Y and Z Directions

Crack Growth

Yes

Adjust Boundary Conditions according to Input Data

STOP

FLOW CHART FOR 'KSAP II' PROGRAM
2 THE STRUCTURE OF THE PREPROCESSOR PROGRAM

2.1 INTRODUCTION

The preprocessor program generates the input data required for the main code, KSAP II. The input data required for the preprocessor program pertains to the dimensions of the plate, mesh plan, material properties of the layers, and the boundary conditions. In its present form, this program can generate data only for brick type elements with either 8 nodes representing the 8 corners of the element or 21 nodes as shown in Figure 2.1.

There are two options in generating the mesh. One is for rectangular mesh for laminates without any curved boundaries and the other is for generating mesh in a laminate with a central hole. The mesh pattern in the later one is chosen to accommodate split (or split growth) tangential to the hole boundary along the loading direction. There is no limitation on the number of layers or the stacking sequence. Depending on the symmetry in geometry and/or loading, one-half, one-quarter or one-eighth of the plate may be analyzed. The displacement and force boundary conditions have to be appropriately specified in order to take the advantage of symmetry.

The program automatically assigns numbers to nodal points, and cartesian coordinates to each node according to input data. The nodes are numbered in an orderly fashion in x, y, z-directions and the 8 (or 21) nodes for each element can be generated arbitrarily from the set of coordinates given in x, y and z directions. The dimensions of elements in any direction can be controlled by changing the coordinates in that direction and thereby the density of the mesh in any region can be changed.
FIG. 2.1 THREE DIMENSIONAL 21 NODE ISOPARAMETRIC ELEMENT
The thermal loading simulation requires two data sets: assigning stress free temperature for each element and prescribing the temperature at which the plate is to be analyzed for delamination. The stress free temperature is assigned to each element while generating the elements. The temperature at which the plate is to be analyzed is provided while generating nodal points. The temperature distribution need not be uniform for the whole plate and each node can be assigned a different temperature. The details of this data input is explained in the next section 2.2.

Mechanical loading can be either a prescription of forces or a prescription of non-zero displacements at the nodes. The details of prescribing force boundary conditions are found in section 2.2. A plate subjected to uniform strain can be simulated by assigning non-zero displacements to the appropriate nodes. These non-zero displacements are changed to force boundary conditions by attaching a linear spring with a large stiffness value (k) to each node in the given displacement (d) direction and applying a force (k \times d) at the other end of the spring. These boundary elements do not increase the total degrees of freedom of the stiffness matrix. The nodes having zero displacements, which are used to specify the symmetric planes, do not make use of these boundary elements and they essentially remove those degrees of freedom from the system of equations.

The main program, KSAP II can simulate a crack opening along a symmetric plane or along any plane given by x=constant or y=constant or z=constant. For example: the interlaminar boundary (layer interface) corresponds to z=constant. A crack along a symmetric plane (e.g., the mid-plane of a symmetric laminate) is simulated by suitably changing the boundary conditions at those nodes on that plane, which will be released to
simulate crack opening. The degrees of freedom of these nodes must be 
retained if a crack is to be simulated along the plane of symmetry. Hence 
they should not be removed by giving zero displacement in the direction of 
the plane of symmetry. Hence they should not be removed by giving zero displacement in the direction of crack opening. The crack opening instruction along these nodes is 
explained in the next chapter. If a crack opening along a plane other than 
the symmetric plane is to be simulated then double nodes are to be assigned 
for each nodal point located on that plane. The double nodes need not be 
taken into account while generating the initial set of elements and nodes. 
Given the plane of crack (1 for yz-plane, 2 for zx-plane and 3 for 
xy-plane), the preprocessor program has the capability to renumber the mesh 
and update the node numbers for each element when the instruction 
pertaining to the double nodes is supplied.

A complete listing of the preprocessor program can be found in 
Appendix A. The following flow chart shown on the next page illustrates 
the general structure of the preprocessor program.
2.2 DATA INPUT TO PREPROCESSOR PROGRAM

The input data required for the preprocessor is made very simple and is kept to a minimum. For example, the element generation (assigning node numbers to the elements) can be done in only a few cards as explained in Section 2.2.2.

2.2.1 Details of the Data Input

The input data is arranged in the following nine groups of cards. Each group consists of one or more cards. Data in the groups I, VI, VII and VIII must be given in the specified format. Each entry must be made in the specified columns and a brief explanation of the entry can be found in entry description. The name listed under 'variable' is the name used for that entry in the program listing. The data in the groups II, III, IV and V may be given in free format. In these groups, if the data does not fit on one card, it may be continued on an immediately following card. Each of the groups II, VI and VIII may have several cards and the program recognizes the termination of that group only when it encounters a card with -1 as the first entry.

Group I: Heading Card (Format A72):
MFD(72) - heading information to be printed with the outputs

Group II: Mesh Generation Cards (Free Format)
card 1:
NTYPE - Type of Element (8 node or 21 node)
card 2:
NONX, NONY, NONZ, RAD
- Number of Nodes in X, Y and Z directions. Radius of the hole (if RAD=0 given, rectangular mesh will be generated.
NOTE: If NTYPE=21, NONX, NONY, NONZ have to be odd numbers.
card 3:
XX(1), XX(2), .... XX(NONX)
- X- coordinates of the nodes in the x-direction.
card 4:
YY(1), YY(2), .... YY(NONY)
- Y- coordinates of the nodes in the y-direction.
NOTE: For 21-node element generation, even numbered coordinates should be middle points of immediate neighboring points. i.e., for \( i = 2, 4, \ldots \)

\[
\begin{align*}
xx(i) &= (xx(i-1) + xx(i+1))/2. \\
nn(i) &= (nn(i-1) + nn(i+1))/2. \\
nn(i) &= (nn(i-1) + nn(i+1))/2.
\end{align*}
\]

Any mistake in the coordinates of even numbered coordinates will be corrected by the preprocessor.

For a laminate with a hole, \( x \) and \( y \) coordinates can be given with hole center as the origin. The coordinates of some of the nodes will be transformed and results in a mesh as shown in Figure 2.2.

Group III  Element, Nodal Property Definition Cards (Free Format)

Card 1: Nodal temperature at which analysis is to be carried out

- \( N, \ TEMP, NEND, INC \)

- \( N \) - starting node number
- \( TEMP \) - magnitude of temperature of the node
- \( NEND \) - last node up to which nodes have same temperature
- \( INC \) - increment between \( N \) and \( NEND \)

\(-1, 0.0, 0, 0\) - data termination card

NOTE: This temperature will be different from stress free temperature of the element (see next card) for thermal loading only.

Card 2: Element stress free temperature

- \( N, \ TEMP, NEND, INC \)

- \( N \) - starting element number
- \( TEMP \) - stress free temperature of the element (curing temp.)
- \( NEND \) - last element number up to which elements have same temperature
- \( INC \) - increment between \( N \) and \( NEND \)

\(-1, 0.0, 0, 0\) - data termination card

Card 3: Element material definition

- \( N, \ MATEL, NEND, INC \)

- \( N \) - starting element number
- \( MATEL \) - material identification number (ex:1,2,3,..)
- \( NEND \) - last element number up to which nodes have same temperature
- \( INC \) - increment between \( N \) and \( NEND \)

\(-1, 0, 0, 0\) - data termination card

NOTE: This card is to identify the elements to which material they belong to. Material properties for different identification numbers are given in later cards.
FIG. 2.2 FINITE ELEMENT MESH IN XY-PLANE IN A LAMINATE WITH A HOLE
card 4: Element material axis orientation definition
N, MORTT, NEND, INC
- N - starting element number
- MORTT - material axes orientation set number (ex: 1, 2, 3, ..)
- NEND - last element number up to which nodes have same temperature
- INC - increment between N and NEND
  -1, 0, 0, 0 - data termination card

card 5: Element stiffness matrix reuse definition
M1, M2
M3, M4
- -
  - M1, M3 - starting element number
  - M2, M4 - last element number up to which element stiffness is same
  -1, 0 - data termination card

NOTE: These cards to identify the elements with the same stiffness matrix and thereby saves computational time. A number of ranges (M1, M2; M3, M4; ..etc.) can be given one after another.

Group IV Split or plane notch definition data (Free Format)
card 1:
NSD, IDIR
  - NSD - number of nodes lying inside the split or plane notch region
  - IDIR - direction number normal to the plane of the notch
    if the normal is parallel to x-axis, IDIR=1
    if the normal is parallel to y-axis, IDIR=2
    if the normal is parallel to z-axis, IDIR=3

NOTE: If no split is required enter 0,1 and skip card 2.
card 2: node numbers defining split region
N, NEND, INC
  - N - starting node number
  - NEND - last node number
  - INC - increment between N and NEND
  -1, 0, 0, - data termination card

NOTE: With this card, number of splits can be defined in parallel planes. Split defined by this card is simulated by doubling the nodes but these double nodes cannot be used to simulate crack growth. In order to read the displacement output of the nodes inside the split, refer to corresponding double nodes given in the output file 'PMPO10.PAT'.

2.8
Group V  Delamination definition data (Free Format)

card 1:

NTD
-NTD - total number of nodes defining delamination region
If there are no double nodes in the problem enter 0 and skip card 2

card 2: node numbers defining delamination region
NOND(1), NOND(2), ....NOND(NTD)

NOTE: The double nodes and the corresponding original nodes are not written in a separate output file. They are given in KSAPIN,DAT itself. They are arranged in the ascending order of the original nodes to facilitate easy modification of KSAPIN,DAT for crack growth simulation. So, it is advisable to give the nodes here in the order of their release.

For 21 node element, face center nodes are not used in KSAP II and hence they are eliminated from the double node list by the preprocessor.

card 3:

XL, XU, YL, YU, ZL, ZU

- XL, XU - lower and upper bounds of x-coordinate of the laminate boundary in which second set of double nodes are to be placed.
- YL, YU - lower and upper bounds of y-coordinate of the laminate boundary in which second set of double nodes are to be placed.
- ZL, ZU - lower and upper bounds of z-coordinate of the laminate boundary in which second set of double nodes are to be placed.

Group VI  Material Property Data

Orthotropic, temperature dependent material properties may be prescribed. Here L,T,Z are the principal axes of the material. For each different material the following group (a) cards must be supplied.

(a) Material Properties (Format A4,14,15,17,7)

card 1:

<table>
<thead>
<tr>
<th>columns</th>
<th>entry description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-4</td>
<td>bbEL</td>
</tr>
<tr>
<td>5-8</td>
<td>material identification number</td>
</tr>
<tr>
<td>13-29</td>
<td>value of Young's modulus in L-direction</td>
</tr>
</tbody>
</table>

card 2:

<table>
<thead>
<tr>
<th>columns</th>
<th>entry description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-4</td>
<td>bbET</td>
</tr>
<tr>
<td>Card</td>
<td>Columns</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
</tr>
<tr>
<td>3</td>
<td>1-4</td>
</tr>
<tr>
<td>4</td>
<td>1-4</td>
</tr>
<tr>
<td>5</td>
<td>1-4</td>
</tr>
<tr>
<td>6</td>
<td>1-4</td>
</tr>
<tr>
<td>7</td>
<td>1-4</td>
</tr>
<tr>
<td>8</td>
<td>1-4</td>
</tr>
<tr>
<td>9</td>
<td>1-4</td>
</tr>
<tr>
<td>10</td>
<td>1-4</td>
</tr>
<tr>
<td>11</td>
<td>1-4</td>
</tr>
</tbody>
</table>
card 12:

<table>
<thead>
<tr>
<th>columns</th>
<th>entry description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-4</td>
<td>ALFZ</td>
</tr>
<tr>
<td>5-8</td>
<td>material identification number</td>
</tr>
<tr>
<td>13-29</td>
<td>value of the thermal expansion coeff., $\alpha_z$</td>
</tr>
</tbody>
</table>

NOTE: If any of these 12 cards are not supplied then that particular value will be set equal to zero.

The 12 constants ($\Ell, \Ett, \ldots, \z$) are defined with respect to a set of axes ($L, T, Z$) which are the principal material directions.

(b) Data termination Card (Format A4)

<table>
<thead>
<tr>
<th>columns</th>
<th>entry description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-2</td>
<td>-1 indicates the end of material property cards.</td>
</tr>
</tbody>
</table>

(c) Material Axes Orientation

In this set the data regarding the material principal axes ($L, T, Z$) relative to the global axes ($x, y, z$) is furnished. There can be several sets of orientations and one card should be input for each orientation as follows:

<table>
<thead>
<tr>
<th>columns</th>
<th>variable</th>
<th>entry description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - 5</td>
<td>MORT</td>
<td>material axes orientation set number</td>
</tr>
<tr>
<td>6 - 10</td>
<td>NI</td>
<td>node number for point &quot;i&quot;</td>
</tr>
<tr>
<td>11 - 15</td>
<td>NJ</td>
<td>node number for point &quot;j&quot;</td>
</tr>
<tr>
<td>16 - 20</td>
<td>NK</td>
<td>node number for point &quot;k&quot;</td>
</tr>
</tbody>
</table>

NOTE: Orientation set numbers (MORT) must be input in increasing sequence beginning with "1".

Orthotropic material axes orientations are specified by means of the three node numbers, NI, NJ, NK. For the special case where orthotropic material axes coincide with the global axes ($x, y, z$), it is not necessary to input data in this section. Let $f_1, f_2, f_3$ be the three orthogonal vectors which define the axes of material orthotropy then their directions are as shown below:

[Diagram showing $f_1, f_2, f_3$ vectors]
Node numbers NI,NJ,NK are only used to locate points i,j,k respectively and any convenient nodes may be used.

End the material orientation definition cards with -1 card.

Group VII  Force boundary conditions
(Format 16,1X,A4,1X,F10.0,12X,I6,I6)

<table>
<thead>
<tr>
<th>columns</th>
<th>variable</th>
<th>entry description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-6</td>
<td>N</td>
<td>node at which force acts</td>
</tr>
<tr>
<td>8-11</td>
<td>LABEL</td>
<td>direction of force (in nodal coordinate system) FY, FY, or FZ</td>
</tr>
<tr>
<td>13-22</td>
<td>FORCE</td>
<td>value of the force</td>
</tr>
<tr>
<td>35-40</td>
<td>NEND</td>
<td>If NEND is greater than N (for N positive) all nodes from N thru NEND in steps of INC has this specified force (if INC is left blank it is assumed to be 1)</td>
</tr>
<tr>
<td>41-46</td>
<td>INC</td>
<td>If NEND is greater than N (for N positive) all nodes from N thru NEND in steps of INC has this specified force (if INC is left blank it is assumed to be 1)</td>
</tr>
</tbody>
</table>

NOTE: N=-1 signifies the end of this set of cards

Group VIII  Displacement Boundary Conditions
(Format 16,1X,A4,1X,F10.0,12X,2I6)

This set of cards is used to constrain nodal displacements to specified values and to compute support reactions. Boundary elements are used to specify strain for the specimen. The boundary element is essentially a spring which has an axial displacement stiffness and it is defined by a single directed axis through the specified nodal point. If any nodal displacement (UX, UY or UZ) is specified to have 0.0 value then that degree of freedom is eliminated from the stiffness matrix.

<table>
<thead>
<tr>
<th>columns</th>
<th>variable</th>
<th>entry description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-6</td>
<td>N</td>
<td>node number at which this displacement will be used</td>
</tr>
<tr>
<td>8-11</td>
<td>LABEL</td>
<td>type of displacement boundary condition</td>
</tr>
<tr>
<td>13-22</td>
<td>V</td>
<td>value of the displacement</td>
</tr>
<tr>
<td>35-40</td>
<td>NEND</td>
<td>If NEND is greater than N (for N positive) then all nodes N thru NEND in steps of INC will have this specified displacement</td>
</tr>
<tr>
<td>41-46</td>
<td>INC</td>
<td>If NEND is greater than N (for N positive) then all nodes N thru NEND in steps of INC will have this specified displacement</td>
</tr>
</tbody>
</table>

NOTE: N=-1 signifies the end of this set of cards.

LABEL can be UX, UY or UZ (upper case) which means that the
specified displacement is in x, y, & z directions respectively.

Group IX Stress Output Locations (Free Format)

LOC1, LOC2, LOC3, LOC4, LOC5, LOC6, LOC7
- location numbers in ascending order

NOTE: In KSAP II, there is a provision to obtain stresses at a maximum of 7 locations in an element. Any 7 of the 27 locations shown in Figure 2.3 can be chosen.
FIG. 2.3 STRESS OUTPUT LOCATION NUMBERS IN AN ELEMENT
3 MODIFICATION OF THE PREPROCESSOR OUTPUT DATA

3.1 INTRODUCTION

The output data of the preprocessor program is to be modified before it can serve as input data to the main code, KSAP II. The preprocessor program will output two files of data: one file will serve as input data file to the KSAP II code and the other file contains the renumbered double nodes for split or notch simulation. The first file is to be supplemented with information regarding the location of double nodes and the information regarding crack opening node sequence. In addition, it is also possible to give commands to selectively print the stress output.

3.2 DOUBLE NODES AND CRACK OPENING SEQUENCE

A double node is originally one node which has two node numbers. These are provided in the plane along which the crack propagation is to be simulated. The double nodes serve two purposes. If the displacements of both the nodes are specified to be same, then they behave as one single node. If, on the other hand, the displacements of the nodes are specified to be independent then they behave as two separate nodes thus simulating crack propagation through that node. Usually, a node has three degrees of freedom, in x,y,z-directions. In the case of double nodes each node has three degrees of freedom after they are separated. However, if the double node is on a symmetric plane, then each node will not have three degrees of freedom after they are separated. For example, Figure 3.1 shows the one-eighth part of a laminate subjected to a force in y-direction. x=0, y=0 and z=0 are symmetric planes. Let there be a transverse crack in the bottom ply as shown by the shaded area. The double nodes 50 and 51 are not
FIG. 3.1 DOUBLE NODE ARRANGEMENT FOR DELAMINATION SIMULATION
on any symmetric plane and hence, the three displacements \((U_x, U_y, & U_z)\) of node 50 are respectively the same as those of node 51 before the nodes are separated. The nodes 50 and 51 each will have three independent degrees of freedom once they are separated. However, the double nodes 10 and 11 will behave differently. When they are together \(U_y = 0\) since they lie on the symmetric plane \(y=0\); and \(U_x\) and \(U_z\) of node 10 are equal to \(U_x\) and \(U_z\) of node 11. When the two nodes are separated as the crack propagates, the upper node 11 will be still on the symmetric plane and it will have \(U_x\) and \(U_z\) degrees of freedom, whereas the bottom node 10 is no more constrained and will have all the three degrees of freedom, \(U_x, U_y\) and \(U_z\), free.

### 3.3 DETAILS OF DATA MODIFICATION

At the end of the preprocessor output the following cards have to be added with regard to double nodes and the crack opening sequence:

1) **Details of the constrained degrees of freedom** (Free format)

a) \(NB\) - total number of degrees of freedom of those double nodes which are constrained by the symmetric plane or constrained by specified displacements before they are released.

b) Details of the constrained degrees of freedom. There should be \(NB\) following cards. Each card contains the following input:

- **NC** - node number
- **IFIX** - degree of freedom
  1 for \(x\), 2 for \(y\), 3 for \(z\) degree of freedom
- **ISAVE** - value of the specified displacement
  \((=0,\) for nodes on that particular symmetric plane)

2) **Details of the other double nodes' degrees of freedom**

a) \(NP\) - number of pairs of all double nodes including those constrained on the symmetric plane

b) There should be \(NP\) following cards. Each card will give details of one pair of double nodes and the possible degrees of freedom after the nodes are open.
NI'C1 = node number 1 of the pair
NI'C2 = node number 2 of the pair
NPX = 0 or 1

NPX=1 signifies that the two nodes are constrained to have the same displacement Ux, before the nodes are open and the nodes are completely free of each other in x-direction of freedom after they are open. NPX=0 signifies that their degrees of freedom are already specified as explained in Group I.

NPY = 0 or 1 ! In Y and Z directions similar to the X-direction as explained above for NPX.

III) Data for each step of opening of nodes:

a) Opening of the double nodes to simulate crack propagation (Free Format)

N1 - ! the node numbers of the paired double nodes
N2 - ! which are to be opened
IDF - 1, 2 or 3 ! the degree of freedom which is to be freed.
IDF=1 denotes x-degree of freedom is freed from its double node's x-degree of freedom and the nodal force Fx becomes 0. Likewise, IDF=2 or 3 denote y or z-degree of freedom is freed.

For those nodes on the symmetric plane and constrained by specified displacement which are to be freed the above card should be modified as

N1 = node number
N2 = 0
IDF = 1, 2, or 3 depending on x, y, or z degree of freedom which is to be freed

There should be as many cards as there are degrees of freedom to be freed.

N1=N2-IDF= 0 signifies the end of the crack opening instruction and the stress and energy released associated for this opening will be calculated.

b) Selective stress print option (Free Format)

NBEG ! stresses will be output for the element from
NBND ! elements from NBEG thru NBND
This card should not be left blank. If 0,0 is entered stresses will not be printed.

If another step of opening is desired, the above (a) and (b) will be repeated. This may be continued until all the
nodes in Group-I and II are relaxed.

It may be noted that if the displacement and stress solution is desired before any crack is simulated a 0,0,0 card is necessary after Group I and Group II cards.

IV) The crack propagation is terminated by a card containing '9999 9999 0' as input.
4 THE STRUCTURE OF THE MAIN CODE, KSAP II

4.1 INTRODUCTION

KSAP II is the main program in the analysis of a delamination or splitting problem of a composite plate. The program simulates the crack opening using the data regarding the finite element mesh and the predetermined crack opening sequence. At each step, the program computes the energy released together with the stress and displacement fields.

4.2 GENERAL FEATURES OF THE CODE

The code uses an 8-node or 21-node solid brick element to calculate the stiffness matrix. Each node is assumed to have three degrees of freedom in x, y, z-directions. General orthotropic material properties can be assigned to the element. It is assumed that the whole element is at a uniform temperature given by the average of the temperatures at the 8 (or 21) nodes. The thermal loads are calculated using the difference between this average temperature and the stress-free temperature of the element.

KSAP II code has the capability to simulate crack opening along the surface which passes through the points where double nodes are prescribed. Initially, the two nodes in each pair are assigned the same displacements in the three degrees of freedom. The system of linear equations are solved with the appropriate boundary conditions (mechanical loading or thermal loading or both) for nodal displacements and nodal forces. The stresses at the prescribed locations in each element are also calculated. The nodal forces of the double nodes are nothing but the internal forces holding these two nodes together. These forces are stored and will be used in the
next iteration as the crack opening is simulated through those nodes. The crack opening is simulated by changing the boundary conditions of the double nodes. This implies, obviously, that the displacements of the two nodes will not be the same. Then the system of linear equations are solved for nodal displacements and nodal forces under the changed boundary conditions. The difference in the displacements of the two nodes through which crack opening is simulated will be the crack opening displacement. Using the internal force which was necessary to hold them together (as found in the preceding iteration), the strain energy released can be computed as the crack opening is simulated through that node. This procedure can be continued until all the double nodes are opened.

Thus, strain energy released as the crack passes through successive double nodes can be calculated at each step. At each step the crack opening can be simulated through one or more pairs of double nodes and there is no limitation on the crack front shape.

If the crack is simulated along a symmetric plane, there is no need for double nodes in that plane. The crack opening can be done by simply changing the boundary conditions of the nodes on that plane from displacement boundary conditions to free force boundary conditions.

Once the strain energies released are calculated at each step then the energy release rates (energy released per unit area) may be obtained by dividing the energy released by the increment in crack area at that step.

A complete listing of the KSAP II code can be found in Appendix B. The following flow chart illustrates the general structure of the KSAP II program.
4.3 THE OUTPUT DATA FROM KSAP II

The output data from KSAP II consists of the details of the finite element mesh of the given problem as well as the solution of the laminated plate for the given crack simulation. For easy reference, the stresses and energies released are written in separate files, KSAPOUT.DAT and WORK.WOK respectively. The rest of the output (control information and displacements) is written in DISP.OUT.

The output file DISP.OUT contains the following information:

i) control information

ii) the nodal point data: cartesian coordinates of each node, the temperature at each node, and the boundary condition codes (1 means restrained, 0 means free to move in that degree of freedom)

iii) the equation numbers assigned to each nodal degree of freedom

iv) boundary elements data which are attached to nodes where non-zero displacement boundary conditions are prescribed

v) the material property tables for different layers

vi) element data which consists of corner nodes and the material table number to which the element belongs

vii) data regarding equations, i.e., number of equations, bandwidth etc.

viii) the solution data at each step consists of the nodal displacements (U, V, W) and forces (Fx, Fy, Fz)

(The first step is numbered as zero, the second step is numbered 1 and so on)

The output file KSAPOUT.DAT contains the following information:

i) element stresses (SIG-XX, SIG-YY, SIG-ZZ, SIG-XY, SIG-YZ, SIG-ZX) at the prescribed locations in each element.

ii) from the second step onwards, energy released in x, y, and z directions will also be output after the element stresses. This energy released output is the sum of the
energies released at all the double nodes relaxed at that iteration if more than one double node are relaxed.

4.4 LIMITATIONS OF KSAP II CODE

At present the program can handle up to 3000 nodes, and 100 pairs of double nodes. Should a problem involve more than 3000 nodes then the following dimension statements have to be suitably changed:

1) The degrees of freedom (3 times the total number of nodes) have to be changed in ICR(*), R(*) in the statement with serial numbers 3380, 3386, 3844 and 3849.

2) The number of nodes in ID(*,6) have to be changed in the statements with serial number 3379.

3) The double nodes' total degrees of freedom (6 times the pairs of double nodes) have to be changed in TMAH(*,*) , TMAH(*,*) , TCOL(*,1), TCOL(*), TCOLM(*), IST(*) in the statements with serial numbers 3384, 3385, 3386, 3848 and A(*,*) , B(*,1), IPIVOT (*), INDEX(*,*) , DT(*) in the statement with serial number 4366 to the same value.
5 POSTPROCESSOR PROGRAM

5.1 INTRODUCTION

The postprocessor program 'plot' is designed to present the stress output of the main code, KSAP II, in a graphical form. The stress distribution in any plane parallel to xy plane can be displayed on a graphics terminal or graphics output can be obtained on printronix printer or Hewlett Packard plotter. The code uses 3-D graphics routines from Template package. The program is written to run interactively and the interactive input consists of choice of device, stress number (1 for xx, 2 for yy, 3 for zz, 4 for xy, 5 for yz and 6 for zx stress), viewing position coordinates, scale factor to scale stress values. The stress values, coordinates and the related data are read from a prescribed data file.

5.2 DETAILS OF DATA FILE

For an 8 node element the following data precede stress data:

Heading - data file identification heading

NX, NY, NL

NX - number of coordinates in x direction
NY - number of coordinates in y direction
NL - number of layers of finite elements in the laminate

XX(I), I=1,NX - coordinate values in x direction
YY(I), I=1,NY - coordinate values in y direction

For 21 node element the following data should precede stress data:

Heading - data file identification heading

NNODES, NLOC

NNODES - number of nodes of finite element (8 or 21)
NLOC - number of stress output locations requested
NONX, NONY, NONZ

- NONX - number of coordinates in x direction
- NONY - number of coordinates in y direction
- NONZ - number of coordinates in z direction

LOC(I), I=1, NLOC - stress output locations

- X(I), I=1, NONX - coordinate values in x direction
- Y(I), I=1, NONY - coordinate values in y direction
- Z(I), I=1, NONZ - coordinate values in z direction

At the end of the above set of data, one set of stress output (for all the elements) should be copied from KSAPOUT.DAT.

The following plots (Figs. 5.1 and 5.2) are obtained from stress output at step 0 in the example problem. Fig. 5.1 is the normal stress along y direction (stress no. 2 and finite element layer no. 3) in 0° layer. The interlaminar stress (stress no. 3 and finite element layer no. 3) in 0° layer is shown in Fig. 5.2. Both the plots are generated on Hewlett Packard plotter using eye coordinates (30, -30, 30).

5.2
FIG. 3.1 NORMAL STRESS DISTRIBUTION ALONG Y DIRECTION IN 0° LAYER

EYEX = 30; EYEX = -30; EYEX = 30
SCALE: 1 DIV. = 5000 psi
FIG. 5.2 INTERLAMINAR NORMAL STRESS DISTRIBUTION IN 0° LAYER

EYX=30; EYX=-3; EY=3
SCALE: 1 DIV. = 5 psi
6.1 INTRODUCTION

In this section we consider an example problem. The example is of a simple laminate construction and it does not represent a practical problem. The purpose here is to illustrate the procedure to operate the computer code. However, the code is developed for a more general use, subjected to the limitations discussed in the preceding sections.

The following paragraphs present the actual working steps in using the present code to generate the interlaminar stress distribution in a given interface plane. All input and output data for this example problem are found in Appendix D.

Laminate Geometry:
(Because of symmetry only one-eighth of the laminate is considered)

- width of the laminate is 8.0"
- length of the laminate is 6.0"
- number of layers is 2
- thickness of layer 1 is 1.0"
- thickness of layer 2 is 1.0"

As shown in the Figure 6.1 x=0, y=0 and z=0 are symmetric surfaces and a uniform farfield strain is applied in y-direction.

The delamination cracking and mesh size are selected as follows:

4 equal divisions in x-direction
2 equal divisions in y-direction
4 equal divisions in z-direction

Material properties and loading information for each layer are furnished in the following manner:

Layer 1 (90° layer)

- $E_1 = 21.0 \times 10^6$ psi
- $G_{12} = 1.7 \times 10^6$ psi
- $G_{13} = 1.7 \times 10^6$ psi
(a) Transverse Crack/Free Edge Induced Delamination

(b) One-eighth part of the laminate simulated

FIG. 6.1 THE ISOMETRIC VIEWS OF THE EXAMPLE PROBLEM
\[ v_{11} = 0.30 \]
\[ v_{12} = 0.30 \]
\[ v_{12} = 0.54 \]
\[ G_{11} = 0.94 \times 10^6 \text{ psi} \]
\[ G_{12} = 0.94 \times 10^6 \text{ psi} \]
\[ G_{13} = 0.50 \times 10^6 \text{ psi} \]
\[ \alpha_t = 0.20 \times 10^{-6} /\text{F} \]
\[ \gamma_t = 16.0 \times 10^{-4} /\text{F} \]
\[ \gamma_t = 16.0 \times 10^{-4} /\text{F} \]

Layer 2 (0° layer)

The same properties as above.

A uniform displacement of 0.001" is applied in y-direction simulating a constant strain loading and no thermal loading is applied (temp. = 0). The delamination is assumed to take place between layer 1 and layer 2 starting at the outer edge at the intersection of free edge and transverse crack. Boundary conditions are provided to make x-0, y-0 and z-0 symmetric surfaces.

The initial finite element mesh without double nodes is as shown in the Figure 6.2.

6.1 PREPROCESSOR INPUT DATA

Group I

**node fl.[02/902]s; delam- MECH load: 7x3x3 MESH-mon.inp (10/30/87)**

In this first group the heading to be printed is given on one card.
FIG. 6.2 INITIAL FINITE ELEMENT MESH WITHOUT DOUBLE NODES
The first card indicates that 8-node brick element is to be used. The number of coordinates in x, y and z directions are 5, 3 and 5 respectively. These are given in the second card. The fourth entry in this card is 0.0 and it indicates that there is no hole (hole radius = 0.0). The values of x, y and z coordinates are given in the subsequent three cards. No data termination card is required for this set of data.

In this group, the cards 2, 4, 6, 9 and 12 are for data termination. The first card indicates that all the nodes from 1 to 75 in increments of 1 have a temperature of 300.0 °F. Similarly, the third card is for elements which signifies that stress free temperature for elements from 1 to 32 in increments of 1 is 300.0 °F. The material serial number to which each element belongs to is given in 5th card. In the present problem, two layers of laminate are made up of same material. So, in this card, it is given that elements 1 to 32 (in increments of 1) belong to material set 1.

6.5
However, the two layers have different orientations and they are indicated in the 7th and 8th cards. The elements 1 to 16 have the material axis orientation set 1 and 17 to 32 have set 2.

The 10th and 11th cards are to take advantage of the set of identical elements made of the same material. The first card denotes that elements 2 thru 16 are identical to element no. 1 and the same element stiffness matrix is used. Similarly, the 2nd card denotes that elements 18 thru 32 are the same as the preceding element no. 17.

Group IV

This card is for split or notch simulation. The first entry (0) indicates that there are 0 double nodes for split generation. The second entry 2 is for split plane parallel to xz-plane. Since the number of double nodes are zero, the value of second entry can be 1, 2 or 3 and no split will be generated.

Group V

The information of nodes which are to be doubled is given in this set of cards. The first card says that there are 15 nodes to be doubled and the succeeding card gives the original numbers of the nodes which are to be doubled. The last card gives limits of the coordinates of the solid in which the second set of double nodes are to be placed.
Group VI

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>EL</td>
<td>21.0E6</td>
</tr>
<tr>
<td>ET</td>
<td>1.7E6</td>
</tr>
<tr>
<td>EZ</td>
<td>1.7E6</td>
</tr>
<tr>
<td>NULT</td>
<td>0.3</td>
</tr>
<tr>
<td>NULZ</td>
<td>0.3</td>
</tr>
<tr>
<td>NUTZ</td>
<td>0.54</td>
</tr>
<tr>
<td>GLT</td>
<td>0.94E06</td>
</tr>
<tr>
<td>GLZ</td>
<td>0.94E06</td>
</tr>
<tr>
<td>GTZ</td>
<td>0.50E06</td>
</tr>
<tr>
<td>ALFL</td>
<td>0.2E-6</td>
</tr>
<tr>
<td>ALFT</td>
<td>0.16E-4</td>
</tr>
<tr>
<td>ALFZ</td>
<td>0.16E-4</td>
</tr>
</tbody>
</table>

This set of cards will furnish data regarding the material properties. These properties can be given in any order. The last three cards are to define 2 sets of material principal axes orientations.

Group VII

This set of cards is to specify force boundary conditions. In this particular problem a single -1 card signify that there are no force boundary conditions.

Group VIII

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>UY</td>
<td>0.0</td>
</tr>
<tr>
<td>UY</td>
<td>0.0</td>
</tr>
<tr>
<td>UX</td>
<td>0.0</td>
</tr>
<tr>
<td>UX</td>
<td>0.0</td>
</tr>
<tr>
<td>UX</td>
<td>0.0</td>
</tr>
<tr>
<td>UX</td>
<td>0.0</td>
</tr>
<tr>
<td>UX</td>
<td>0.0</td>
</tr>
<tr>
<td>UX</td>
<td>0.0</td>
</tr>
<tr>
<td>UX</td>
<td>0.0</td>
</tr>
<tr>
<td>UZ</td>
<td>0.0</td>
</tr>
<tr>
<td>UZ</td>
<td>0.0</td>
</tr>
<tr>
<td>UZ</td>
<td>0.0</td>
</tr>
<tr>
<td>UY</td>
<td>0.001</td>
</tr>
</tbody>
</table>

-1
The displacement boundary conditions are prescribed in this set of cards. For example, the first card specifies the y-component of displacement as 0.0 for the nodes from 4 thru 24 at 5 node intervals. That is, y-component displacement of nodes 4, 9, 14, 19, 24 have 0.0 value. The boundary conditions of other nodes are prescribed in the succeeding cards of this last set. In this set the original node numbers are to be given and they will be modified using the double nodes information given in Group V.

6.3 MODIFICATIONS OF PREPROCESSOR OUTPUT DATA

The output of the preprocessor program will be two data files if NSD is not equal to zero. The file KSAPIN.DAT will consist most of the data necessary to run the main code, KSAP II. The other file FORO10.DAT will contain the information about the modified numbers of the double nodes and their original node numbers on the split plane. These are only for reference and do not appear in the modifications of KSAPIN.DAT.

The original node numbers related to delamination are listed towards the end of KSAPIN.DAT. Both the double nodes of each pair will have the same displacements before the crack passes through them, and both these will have free force boundary conditions once the crack passes through that pair and they are separated. However, some of these pairs lying on y=0 plane behave differently.

The double nodes (3,4),(9,10),(15,16), (21,22), and (27,28) are located on the symmetric plane which also happens to be the plane of transverse crack. The y-component of the displacement (Vy) of the node in each pair before they are opened have the same value. When the crack passes through
that node, the two nodes will be separated. The y-component of the force (F_y) of the bottom node will be zero whereas the top node will be still on the symmetric plane (y=0) and hence will have a displacement boundary condition (U_y=0). It may also be observed that the other two components of the forces (F_x, F_z) will become zero for both the nodes of the pair once the crack passes through that pair. To facilitate these two types of degrees of freedom of the double nodes the input data of KSAP II has to be supplemented with the data as shown below.

```
10
  3 2 0.0
  4 2 0.0
  9 2 0.0
 10 2 0.0
 15 2 0.0
 16 2 0.0
 21 2 0.0
 22 2 0.0
 27 2 0.0
 28 2 0.0
 15
  3 4 1 0 1
  6 10 1 0 1
 15 16 1 0 1
 21 22 1 0 1
 27 28 1 0 1
 33 34 1 1 1
 39 40 1 1 1
 45 46 1 1 1
 51 52 1 1 1
 57 58 1 1 1
 63 64 1 1 1
 69 70 1 1 1
 75 76 1 1 1
 81 82 1 1 1
 87 88 1 1 1
```

The number (10) in the first data card denotes the total number of y-degrees of freedom of double nodes (5x2-10) on the symmetric plane (y=0). The following 10 data cards input the details of the node number, degree of freedom (1 for U_x, 2 for U_y and 3 for U_z) and the value of the displacement. All these data corresponds to that value before the nodes...
are opened. Following this will be the comprehensive data input for the overall double nodes. The number 15 denotes that there are 15 pairs of double nodes whose details are given in the 15 succeeding cards. The first two numbers in each card (for example, 3 and 4) are two node numbers in each pair. The three following numbers (1's or 0's) will describe the behavior of the double nodes in x, y, z degrees of freedom respectively when the nodes are opened. The number 1 for any degree of freedom signifies that the nodes will have the same displacement before opening and will have zero nodal force in that degree of freedom after opening. Number 0 for any particular degree of freedom signifies that the nodes will not behave in the above manner with regard to that degree of freedom. Thus the degrees of freedom (y) for the double nodes on the symmetric plane, y=0 which are described in the previous set will have zeros in this set of data. For example, the nodes 3 and 4 will have Uy=0 corresponding to y-degree of freedom as they are specified to behave in another manner in the preceding set of data. The nodes 27 and 28 will have all 1's as they are located away from the plane y=0.

After furnishing the above data regarding the degrees of freedom of the double nodes, comes the data regarding the opening of the double nodes thus simulating a crack propagation as shown below:

<table>
<thead>
<tr>
<th>0</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>32</th>
</tr>
</thead>
<tbody>
<tr>
<td>27</td>
<td>0</td>
<td>2</td>
<td>27</td>
<td>28</td>
</tr>
<tr>
<td>27</td>
<td>28</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>32</td>
<td>21</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>21</td>
<td>22</td>
<td>3</td>
<td>57</td>
<td>38</td>
</tr>
<tr>
<td>57</td>
<td>38</td>
<td>2</td>
<td>57</td>
<td>38</td>
</tr>
</tbody>
</table>
At any step (iteration) the crack can be made to pass through one or more number of double nodes. Each data card consists of three numbers. The last number corresponds to the degree of freedom which is relaxed, that is, which will have free force boundary condition. If there is a zero as the second number then the first number should be a node number of the double nodes on the symmetric plane y=0 whose data is specified in the first set. For example, the '27 0 2' specifies that the y-degree of freedom (2) of node no. 27 (which is a node in 1st set of data) is relaxed (Fy=0). That is, this node is free to move in y-direction. If the second number is also non zero then the first two numbers correspond to the two nodes of a pair of double nodes and as explained above the third number specifies the degree of freedom in which these two nodes are free to move. Thus the data specifies that the crack passes through the pair of nodes 27 and 28 and these two nodes are free to move in x, z directions whereas in the y-direction only node no. 27 is free to move. That implies that node 28 will have the earlier specified displacement (Uy=0). All three zeros signify the end of the crack opening data for that step. Thus when it is desired to calculate the stresses and displacements before any crack is simulated it is necessary to place this card (0,0,0) as in line 1. Immediately following this card in each step a selective stress print option can be given. The two numbers signify the range of elements for which the stress print out is desired. If the first two numbers in the crack opening data are prescribed as 9999 and 9999 then that signals the termination of crack propagation sequence.
6.4 OUTPUT OF KSAP II PROGRAM

As can be seen the output of KSAP II is self explanatory. To start with, it consists of all the mesh details regarding the nodes, coordinates, the degrees of freedom, elements etc. It also furnishes information of the material properties used and the material number to which each element belongs. The output also provides some details about total number of equations, bandwidth, number of blocks etc.

Then the results will be output as the crack is simulated. For each step the nodal displacements and element stresses (at the center of the element) are printed as desired in the input data. Usually, these results are output starting from no crack state (STEP 0) and crack can be simulated opening one or more nodes at each step. At each succeeding step (STEP 1, 2, ....) the energy released is calculated and is printed immediately after nodal displacements and forces. If the energy release rate is desired then it can be calculated by dividing these energies released by the incremental crack areas.
APPENDIX - A

LISTING OF THE PREPROCESSOR
THIS CODE GENERATES INPUT DATA FOR 'KSAP II'

version : September 1987

Reads from Input file and writes to KSAPIN.DAT

NOTE: 1) Modified to generate 21-node element directly from the regular mesh WITHOUT ELEMENT GENERATION CODES

Assign 21 / 8 (= NTYPE ) as the 2nd card in the input data file

CAUTION:

2) The nodal input order for the elements (NTYPE=21) SHOULD be as in SAPIV MANUAL

3) Double nodes region is to be given (xl,xu, yl, yu, and zl, zu in that order)

R(), NR() --- temporary storage of data read
NOS() --- number of sets to be generated at i-th level
NOE() --- number of elements in the set to be generated
NINC() --- increment of node number
NO(I,J) --- node numbers of element -i
IDM --- i.d. number of matl.
MAI(I) --- 1 : stiffness matrix of element -i is the same as previous 0 : not the same as last
MAXES()--- I.D.no. of matl. axis orientation set(see SAP4 manual)
TZ() --- Stress free reference temperature for element -I
X(I) --- coordinets of node -I
Y(I) 
Z(I) 
XX() --- coor. increments in level generation
YY() 
ZZ() 
T(I) --- Nodal temperature
E11 .. --- Material properties
IX() --- Degree of freedom code, node -i, freedom -j
NB() --- No. of boundary element FOR DISPLACEMENT
ND() --- 4 nodes used for defining the direction of displ.
KD() --- Means displ. if equals to 1
KR() --- , , rotation if equals to 1
NBF() --- node no. of force boundary
FX() --- concentrated force
FY() 
FZ() 
L(4) --- Node defining vector criss multiplied to give
the direction of displ. or rotation
VD()  --- Value of displ.
VR()  --- Value of rotation
LAX(2,3)  --- 3 nodes (number) to define matl. princ. axes
(max. of 2 sets)
HED(18)  --- reading information as first line of output
MORT()  --- material axes orientation set number
NI()  --- definition of material principle axes
NK() 
cccc...ccc

CHARACTERA30 FILNAM
CHARACTERMESH
DIMENSION R(7,NR(15),NRR(21),NOS(3),NOE(3),NINC(3)),
XX(20),YY(20),ZZ(15),XMESH(4000),YMESH(4000),
X(4000),Y(4000),Z(4000),IX(4000,6),I(4000),mip(4000,2),
IDM(3000),MAI(3000),NN(3000,21),MAXES(3000),TZ(3000),
E11(4),E22(4),E33(4),ANU12(4),ANU13(4),ANU23(4),
G12(4),G13(4),G23(4),NBD(200),ND(200,4),KD(200),
L(4),VD(200),VK(200),KR(200),HED(18),
NOND(200),XD(200),YD(200),ZD(200), nans(400),
ALP1(4),ALP2(4),
ALP3(4),MORT(10),NI(10),NJ(10),NK(10),
EX(200),FY(200),T(200),MBE(200),ID1(200),
-- mip () - the corresponding new nodal numbers
PI=3.1415926535897932
eps=1.0e-04
WRITE (5,A) '...ENTER INPUT FILE NAME....'
READ (5,55) FILNAM
IRD=56
OPEN (UNIT=IRD,FILE=FILNAM,STATUS='OLD')
READ(IRD,25) (HED(I),I=1,18)
IF (NTYPE.EQ.21) THEN
  IRRS=4
  INTT=2
END IF
!!!!!! FOLLOWING STATEMENTS ARE FOR VARYING GRID GENERATION.
C NODE GENERATION
READ (IRD, A), NNONX, NNONY, NNONZ, RHOLE
READ (IRD, A)(XX(I), I=1, NNONX)
READ (IRD, A)(YY(I), I=1, NNONY)
READ (IRD, A)(ZZ(I), I=1, NNONZ)
C-- correction for inside nodal coordinates for 21-node el.
if (ntype.eq.21) then
do 1444 i=1, NNONX
   if (mod(i,2).eq.0) xx(i)=(xx(i-1)+xx(i+1))/2
   type k, (xx(i), i=1, NNONX)
do 1445 i=1, NNONY
   if (mod(i,2).eq.0) yy(i)=(yy(i-1)+yy(i+1))/2
   type k, (yy(i), i=1, NNONY)
do 1446 i=1, NNONZ
   if (mod(i,2).eq.0) zz(i)=(zz(i-1)+zz(i+1))/2
   type k, (zz(i), i=1, NNONZ)
end if
if (RHOE.GT.0.001) call HOLE (RHOE, NNONX, NNONY, NNONZ, XX, YY, ZZ, X, Y, Z)
   IF (RHOE.GT.0.001) GO TO 444
   IF (RHOE.GT.0.001) GO TO 444
   DO J=1, NNONY
      DO I=1, NNONX
         DO K=1, NNONZ
            N=(J-1)*NNONX*NNONZ+(I-1)*NNONZ+K
            X(N)=XX(I)
            Y(N)=YY(J)
            Z(N)=ZZ(K)
         END DO
      END DO
   END DO
C END OF POLAR MESH GENERATION
444 NTON=NNONX*NNONY*NNONZ
C-- MESH PLOTTING OPTION ON HP PLOTTER-------
WRITE (5, A) 'Do you need MESH plot Original nodes?..(Y/N)'
READ (5, 55) MESH
55 FORMAT (A)
IF (MESH.EQ.'Y'.OR.MESH.EQ.'y') CALL .MESHPL (X, Y, Z, NNONX, NNONY, NNONZ, MIFP, 1, ntype)
   DO I=1, NT1
      XMESH(I)=X(I)
   END DO
   IF (NN(N1O,7).GT.0.001) THEN
      DO 666 I=1, N1O
         Y_MESH(I)=Y(I)
      END DO
      DO 666 J=1, NTYPE
         IF (NN(I, J).GT.0.001) NN(I, J)=NN(I, J)-NT0N
      END DO
   END IF
A- 3
END IF
165 148 READ(IKD,A) N,TEMP,NEND,INC
166 IF(N.EQ.-1) GO TO 149
167 DO I=N,NEND,INC
168 T(I)=TEMP
169 END DO
170 GO TO 148
171
172 149 continue
173 300 CONTINUE
174 C  ELEMENT GENERATION
175 C
176 C
177 N1=NONX
178 N2=NONY
179 N3=NONZ
180 N13=N1AN3
181 IF(NTYPE.EQ.8) THEN
182 NOS(3) =N3-1 ' third level gener. code
183 NINC(3)=1
184 NOE(3) = (N2-1)*(N1-1)
185 NOS(2) =N2-1 ' second level gener. code
186 NINC(2)=N1AN3
187 NOE(2) =N1-1
188 NOS(1) =N1-1 ' first level gener. code
189 NINC(1)=N3
190 NOE(1) =1
191
192 NKK(1)=2+ N3+ N13
193 NKR(2)=2+ N13
194 NKR(3)=2
195 NKR(4)=2+ N3
196 NKR(5)=1+ N3+ N13
197 NKR(6)=1 + N13
198 NKR(7) =1
199 NKR(8)=1+ N3
200
201 ELSE IF(NTYPE.EQ.21) THEN
202 NOS(3) = (N3-1)/2 ' third level gener. code
203 NINC(3)=2
204 NOE(3) = (N2-1)*(N1-1)/4
205 NOS(2) = (N2-1)/2 ' second level gener. code
206 NINC(2)=2AN1AN3
207 NOE(2) = (N1-1)/2
208 NOS(1) = (N1-1)/2 ' first level gener. code
209 NINC(1)=2AN3
210 NOE(1) =1
211
212 NKR(1) =3+2AN3+2AN13
213 NKR(2) =3 +2AN13
NRR(3) = 3
NRR(4) = 3 + 2AN3
NRR(5) = 1 + 2AN3 + 2AN13
NRR(6) = 1 + 2AN13
NRR(7) = 1
NRR(8) = 1 + 2AN3
NRR(9) = 1 + N3 + 2AN13
NRR(10) = 1 + N13
NRR(11) = 1 + N3
NRR(12) = 1 + 2AN3 + 2AN13
NRR(13) = 1 + N3 + 2AN13
NRR(14) = 1 + N13
NRR(15) = 1 + N3
NRR(16) = 1 + 2AN3 + 2AN13
NRR(17) = 1 + 2AN3 + 2AN13
NRR(18) = 2 + 2AN13
NRR(19) = 2
NRR(20) = 2 + 2AN3
NRR(21) = 2 + N3 + N13
NRR(22) = 1 + N3 + 2AN13

END IF
DO 132 I=1,NTYPE
132 NN(NO+1,I)=NRR(I)

Perform element generation
DO 138 M=1,3
DO 138 K=2,NOS(M)
DO 138 I=1,NOE(M)
N=NO+I+NOE(M)*(K-1)
DO 1375 J=1,NTYPE
1375 NN(N,J)=NN(NO+I,J)+NINC(M)*(K-1)
continue
NO=NN
NTOE=NO
DO 1415 IE=1,NTOE
1415 TYPE IE, 'AAAA ELEMENT IE', IE
1415 TYPE 105, (NN(IE,J), J=1,NTYPE)
1415 FORMAT(8I5)

Read element stress free temperature
READ(IRO,A) N, TEMP, NEND, INC
IF(N.EQ.-1) GO TO 159
DO 1585 IE=N, NEND, INC
1585 TZ(IE)=TEMP
GO TO 158
159 continue

Read element material identification number
READ(IRO,A) N, MATRL, NEND, INC
IF(N.EQ.-1) GO TO 1459
DO 14585 IE=N,NEND,INC
14585 IDM(IE)=MATRL
DO 1459 continue
1459 continue
C --- MAXES(I) = MATL. AXIS ORIENT.: 
! Read element material axis orientation identification number
1658 READ(IXD,A) N,MORTT,NEND,INC
IF(N.EQ.-1) GO TO 1659
DO 16585 IE=N,NEND,INC
16585 MAXES(IE)=MORTT
GO TO 1658
1659 continue
1659 continue
! Read if the stiffness matrix of an element (or a group of elements)
! is same as the previous element
DO 145 I=I,NTOE
145 MAT(I)=0
READ(IXD,A) M1,M2
WRITE(111,A 'M1,m2',m1,n2
IF(n1.EQ.-1)GO TO 1765
DO 1762 Iznli,2
1762 MAT(I)=1
GO TO 176
1765 continue
C---------------------------------------------------------------------------
C BOUNDARY CONDITION GENERATION
C Following statement are to fix all rotations when
C only translational d.o.f. (eltype #8 is used)
DO 2010 I=I,NTOE
DO 2010 J=1,6
IE(J.LE.3) IX(1,J)=0
IF(J.GE.4) IX(I,J)=1
2010 continue
!
While generating 21-node elements some nodes are at the center of faces
These degrees of freedom will be removed below:
DO IE=1,NTOE
DO IN=1,NTIP E
II=MN(IE,IN)
IX(II,6)=10 !to identify which nodes are used
doi
end do
!
DO IND=1,NTOE
IF(IX(IND,6).EQ.10)THEN
IX(IND,6)=1
ELSE
DO IDG=1,6
IX(IND,IDG)=1
end do
A-6
END DO
END IF
end do

FOR DOUBLING NODES

do ip=1,nton
mip(ip,2)=0
end do

For Splitting

READ(IRD,A) NSD,DIR
! No. of nodes to be doubled, Direction Vector
IF(NSD.NE.0) THEN
END
NSD=0
READ (IRD,A) N,NEND,INC
IF (N.EQ.-1) GO TO 688
DO 687 I=N,NEND,INC
NSD1=0
NSD=NSD+1
NSD1=NSD+1
END
READ(IRD,A) NSD1=I
GO TO 686

READ(IRD,A) (NONS(I),I=1,NSD) !original node # to be doubled
IF (NSD.NE.0) THEN
READ(IRD,A) NSD1=I
END IF
READ(IRD,A) N,NEND,INC
IF (N.EQ.-1) GO TO 688
DO 687 I=N,NEND,INC
NSD1=NSD1+1
NSD1=NSD1+1
END
READ(IRD,A) NSD1=I
GO TO 686

READ(IRD,A) (NONS(I),I=1,NSD) !original node # to be doubled
IF (NSD.NE.0) THEN
READ(IRD,A) NSD1=I
END IF
READ(IRD,A) N,NEND,INC
IF (N.EQ.-1) GO TO 688
DO 687 I=N,NEND,INC
NSD1=NSD1+1
NSD1=NSD1+1
END
READ(IRD,A) NSD1=I
GO TO 686

READ(IRD,A) N,NEND,INC
IF (N.EQ.-1) GO TO 688
DO 687 I=N,NEND,INC
NSD1=NSD1+1
NSD1=NSD1+1
END
READ(IRD,A) NSD1=I
GO TO 686
MIP(I,2)=IP
END IF
END DO
IF (NSD.NE.0)
"call split (NTON,nsd,1dir,nans,mip,x,y,z,I,ntype,nos,NN,ntoe)
! For Delamination............
DO I=I,NTON
TYPE K,I,(MIP(I,J),J=1,2)
END DO
IF(NTD.EQ.0) GO TO 899
C CORRECTIONS FOR DOUBLE NODES IN planes normal to x, y or z -directions
FIND KND (THE # OF THE NODE CURRENTLY TO BE DOUBLED)
DO 525 I=I,NTD
DO J=NOND(I),NTON
DIST=SQRT((X(J)-XD(I))**2+ (Y(J)-YD(I))**2+ (Z(J)-ZD(I))**2)
IF(DIST.LT.0.00001)
KND=J
C BEGIN CHANGING NODE NUMBERS & COORD.
DO 346 JE=I,NTOE
DO 345 K=1,NTYPE
IF (NN(JE,K).GT.ND) NN(JE,K)=NN(JE,K)+1
346 CONTINUE
C Also move coords. downstream by one slot and also assign (Knd+1)th
same as (Knd)th
DO M=NTON,KND,-1
X(M+1)=X(M)
Y(M+1)=Y(M)
Z(M+1)=Z(M)
T(M+1)=T(M)
END DO
C-- Allocate nodes of the pair to the appropriate elements depending
on which side of the double nodes' plane they (element) lie
DO 3455 K=1,NTOE
IF(NTYPE.EQ.8) THEN ! Find the coordinates of the center of the
XC=0.
YC=0.
ZC=0.
DO IA=1,8
III=NN(K,IA)
XC=XC+X(III)
YC=YC+Y(III)
ZC=ZC+Z(III)
END DO
XC=XC/8.
ELSE
III=NN(K,21)
XC=X(III)
YC=Y(III)
ZC=Z(III)

END IF

IF(K.EQ.32) THEN

TYPE A, '... ELEMENT # ... ', K
TYPE A, ' CENTER: ', XC,YC,ZC
TYPE A, ' BEFORE ...
TYPE A, (NN(K,M),M=1,NFACE)

END IF

DO 3453 M=1,nface

IF (NN(K,M).EQ.KND
 .and.(XC.GT.XL.AND.XC.LT.XU)
 .and.(YC.GT.YL.AND.YC.LT.YU)
 .and.(ZC.GT.ZL.AND.ZC.LT.ZU))

NN(K,M)=KND+1
3453 END DO
3455 END DO

NTON=NTON+1

go to 525 ! so that the modifications are done only once
for each double node

END IF ! (DIST.LT.0.00001)
END DO
! J - LOOP
END DO
! I - loop
899 CONTINUE ! SKIP IF NTD=0

write(5,A)'...Do You Need Mesh with Double Nodes (Y/N)..'
READ (5,55) MESH
IF (MESH.EQ.'Y'.OR.MESH.EQ.'y') CALL
MESHPL (XMIN,YMIN,NMIN,NMAX,NMAX,MIP,3,ntype)

IF (MESH.EQ.'Y'.OR.MESH.EQ.'y')
.write(5,A)'...Do You Want to Continue (Y/N)..'
IF (MESH.EQ.'Y'.OR.MESH.EQ.'y')
.READ (5,55) MESH
IF (MESH.EQ.'Y'.AND.MESH.EQ.'y') stop

!!!!!!!!!!! ABOVE ARE FOR DOUBLING NODES !!!!!!!!!!!!!!
487 550 FORMAT(A4,14,4X,G17.7)
488 IF(A.EQ.'EL') E11(I)=v
489 IF(A.EQ.'ET') E22(I)=v
490 IF(A.EQ.'EZ') E33(I)=v
491 IF(A.EQ.'NUTL') ANU12(I)=VAE22(I)/E11(I)
492 IF(A.EQ.'NUTZ') ANU23(I)=VAE33(I)/E22(I)
493 IF(A.EQ.'NULZ') ANU13(I)=VAE33(I)/E11(I)
494 IF(A.EQ.'GLT') G12(I)=v
495 IF(A.EQ.'GLZ') G13(I)=v
496 IF(A.EQ.'GTZ') G23(I)=v
497 IF(A.EQ.'AFL') ALP1(I)=v
498 IF(A.EQ.'ALTER') ALP2(I)=v
499 IF(A.EQ.'ALEF') ALP3(I)=v
500 IF(A.EQ.'L') GO TO 560
501 GO TO 500
502 503 560 DO I=1,10
504 READ(IRD,570) (NR(J),J=1,4)
505 570 FORMAT(4I5)
506 IF(NR(1).EQ.-1) GO TO 579
507 N0RT(1)=NR(1)
508 N01(1)=NR(2)
509 N1J(1)=NR(3)
510 NK(1)=NR(4)
511 END DO
512 579 NORTH0=I-1
513 514
515 516 !THE BELOW PORTION IS MODIFICATION TO FIND DIRECTION VECTOR
517 !above one does not work for hole problems (polar mesh).
518 C TYPE A, 'NTON', NTON
519 DO 991 I=1,NTON
520 DO 992 J=1,NTON
521 IF (J.EQ.I) GO TO 992
522 IF (ABS(Y(J))-ABS(Y(I)).GT.1.E-06) GO TO 992
523 IF (ABS(Y(J))-Y(I)).GT.1.E-06) GO TO 992
524 IF (ABS(Z(J))-ABS(Z(I)).GT.1.E-06) GO TO 992
525 IF (ABS(Z(J))-Z(I)).GT.1.E-06) GO TO 992
526 IF (X(J)-X(I)).LT.0.0) GO TO 992
527 DO 993 K=1,NTON
528 IF (K.EQ.I.OR.K.EQ.J) GO TO 993
529 IF (ABS(X(K))-ABS(X(I)).GT.1.E-06) GO TO 993
530 IF (ABS(X(K))-X(I)).GT.1.E-06) GO TO 993
531 IF (ABS(Z(K))-ABS(Z(I)).GT.1.E-06) GO TO 993
532 IF (ABS(Z(K))-Z(I)).GT.1.E-06) GO TO 993
533 IF (Y(K)-Y(I)).LT.0.0) GO TO 993
534 DO 994 LLL=1,NTON
535 IF (LLL.EQ.I.OR.LLL.EQ.J.OR.LLL.EQ.K) GO TO 994
536 IF (ABS(XLLL)-ABS(X(I)).GT.1.E-06) GO TO 994
537 IF (ABS(XLLL)-X(I)).GT.1.E-06) GO TO 994
538 IF (ABS(YLLL)-ABS(Y(I)).GT.1.E-06) GO TO 994
539 IF (ABS(YLLL)-Y(I)).GT.1.E-06) GO TO 994
540 IF (ABS(ZLLL)-Z(I)).LT.0.0) GO TO 994

A-10
LO=I  ly & z coord. same as J, xcoord. diff.
LX=J  ly & z coord. same as I, xcoord. diff.
LY=K  lyn I or J, x & z coord. same as I and y is diff.
LZ=LLL  lyn I, J or K, x & y coord. same as I and z diff.

GO TO 995

CONTINUE
CONTINUE
CONTINUE
CONTINUE
CONTINUE
CONTINUE

C-- READ & GENERATE CONCENTRATED LOAD DATA:

NTBF=0
NINCC=1
READ(IRD,710) NL,A,V,N2,INCC
IF(N1.EQ.-1) GO TO 699
DO I=N1,N2,NINCC
NTBF=NTBF+1
NBE(NTBF)=I
IF(A.EQ.'FX') FX(NTBF)=V
IF(A.EQ.'FY') FY(NTBF)=V
IF(A.EQ.'FZ') FZ(NTBF)=V
END DO
GO TO 640
CONTINUE

C-- READ & GENERATE DISPL. B.C.E. DATA

NTBD=0
NINCC=1
READ(IRD,710) NL,A,V,N2,INCC
FORMAT(16,1X,A4,IX,F10.0,12-X,16)
IF(N1.EQ.-1) GO TO 799
DO I=N1,N2,NINCC
IF(V.EQ.0) THEN
DO I=N1,N2,NINCC
IF(A.EQ.'UX') IX(I,1)=1
IF(A.EQ.'UY') IX(I,2)=1
IF(A.EQ.'UZ') IX(I,3)=1
END DO
ELSE
MELTYP=2
V1=V
V2=0
JD=1
JR=0
IF(A.EQ.'UX') then
L(1)=LO
L(2)=LY !LO -> LY ALONG Y DIR.
L(3)=LZ !LO -> LZ ALONG Z DIR.
else if (A.EQ. 'UZ') then

L(1) = L0
L(2) = LX
L(3) = L0
L(4) = LY

else if (A.EQ. 'UY') then

L(1) = L0
L(2) = L0
L(3) = LZ
L(4) = LX

END IF

DO I = N1, N2, NIMCC
MTBD = MTBD + 1
MBD(MTBD) = I
DO J = 1, 4
MBD(J, MTBD) = L(J)
END DO
VD(MTBD) = V1
VR(MTBD) = V2
KD(MTBD) = JD
KR(MTBD) = JK
END DO

ND = MAX(ND, 1)
NBD = MIN(NBD, 1)
KD(K) = MIN(KD(K), 1)

IF (MOD(NBD, 2) .EQ. 0) THEN
NBD = NBD + 1
END IF

if (ntbd .GE. 1) then
DO K = 1, NTD
MBDK1 = MIP(NBD(K), 1)
MBDK2 = MIP(NBD(K), 2)
KD(K) = MBDK1
KR(K) = MIP(KR(K), 1)
KD(K) = MIP(KD(K), 1)

IF (MBDK2 .NE. 0) THEN
ntbd = ntbd + 1
NBD(MTBD) = MBDK2
KR(NTBD)=KR(K)
KD(NTBD)=KD(K)
VD(NTBD)=VD(K)
VR(NTBD)=VR(K)
DO M=1,4
ND(NTBD,M)=ND(K,M)
END DO
END
DO
C
RENUMBER NI,NJ,NK (ORIENT. DEFINITION)
DO K=1,NORTH
NI(K)=MIP(NI(K),1)
NJ(K)=MIP(NJ(K),1)
NK(K)=MIP(NK(K),1)
END DO
end if ! ntd.ge.1
C
Renumbering the force boundary conditions and adding
new force boundary conditions of double nodes if necessary
DO K=1,NTBF
if (mip(nbf(k),2).ne.0) THEN
WRITE (5,A) '.... D.N at force b.c; NODE no..', NBF(K)
STOP 'STOPPING due to double node at force boundary cond.'
END IF
NBF(K)=mip(NBF(K),1)
END DO
C
OUTPUT NODE DATA
WRITE(9,1100) (HED(I),I=1,18)
1100 FORMAT(18A4)
IADOF=NTDA6
IADOF===ADDITIONAL D.O.F. DUE TO DOUBLE NODES & DISP.BCS.
WRITE(9,1101) NTON,NELTYP,IADOF
1101 FORMAT(5,5,4X,'',14X,''1XI)
WRITE(9,1105) (IX(I,J),J=1,6),X(1),Y(1),Z(1),I(1)
1105 FORMAT(4X,'1C',14,5I5,3F10.4,5X,F10.8)
KN=0
WRITE(9,1102) 2,(IX(2,J),J=1,6),X(2),Y(2),Z(2),KN,T(2)
1102 FORMAT(4X,'1C',14,5I5,3F10.4,5X,F10.8)
KNM=0
DO I=3,NTON-1
IXM1=ix(i,1)-ix(i-1,1)
IXM2=ix(i,2)-ix(i-1,2)
IXM3=ix(i,3)-ix(i-1,3)
IXM4=ix(i,4)-ix(i-1,4)
IXM5=ix(i,5)-ix(i-1,5)
IXM6=ix(i,6)-ix(i-1,6)
DXM=X(I)-X(I-1)
DYM = Y(I) - Y(I-1)
DZM = Z(I) - Z(I-1)
DTM = T(I) - T(I-1)
IXP1 = ix(i+1,1) - ix(i,1)
IXP2 = ix(i+1,2) - ix(i,2)
IXP3 = ix(i+1,3) - ix(i,3)
IXP4 = ix(i+1,4) - ix(i,4)
IXP5 = ix(i+1,5) - ix(i,5)
IXP6 = ix(i+1,6) - ix(i,6)
DXP = X(I+1) - X(I)
DYP = Y(I+1) - Y(I)
DZP = Z(I+1) - Z(I)
DTP = T(I+1) - T(I)

IF (IXM1.EQ.IXP1.AND.IXM2.EQ.IXP2.AND.IXM3.EQ.IXP3.AND.
   IXM4.EQ.IXP4.AND.IXM5.EQ.IXP5.AND.IXM6.EQ.IXP6.AND.
   DXM.EQ.DXP.AND.DYM.EQ.DYP.AND.DZM.EQ.DZP.AND.
   DTM.EQ.DTP) KN = 1
IF (KN.EQ.0) WRITE(9,1102) I, (1X 1,J),J=1,6),X(I),Y(I),Z(I),KN,T(I)
1102 FORMAT(I5,6I5,3F10.4,15,F10.0)
END DO

WRITE(9,1102) I, (1X(I,J),J=1,6),X(I),Y(I),Z(I),KN,T(I)

IF (IELTYP.GT.1) THEN OUTPUT FOR B.C.E. #7
1201 FORMAT(4X,'7',I5/8X,'1.',)
DO I = 1, NITB
WRITE(9,1202) NBD(I),(ND(I,J),J=1,4),KD(I),KR(I),VD(I),VR(I)
1202 FORMAT(715,5X,2F10.7,' 0.100E+21 ',)
END DO

WRITE(9,1300) IT, E11(I), E12(I), E33(I), ANU12(I), ANU13(I),
   ANU23(I), G12(I), G13(I), G33(I), ALP1(I), ALP2(I), ALP3(I)
1300 FORMAT(4X,'8',I5,15,4X,'0',I5,5X,15)

WRITE(9,1301) I, ITZ(I), E11(I), E22(I), E33(I), ANU12(I), ANU13(I),
   ANU23(I), G12(I), G13(I), G33(I), ALP1(I), ALP2(I), ALP3(I)
1301 FORMAT(I5,4X,'1',20X,'AXIS#1==0-LAYER; AXIS#2==90-LAYER.',/ 
   3F10.0,3F10.4/3F10.0,3F10.7)

WRITE(9,13010) I, E11(I), E22(I), E33(I), ANU12(I), ANU13(I),
   ANU23(I), G12(I), G13(I), G33(I), ALP1(I), ALP2(I), ALP3(I)
13010 FORMAT(F10.0,3F10.0,3F10.4/3F10.0,3F10.7)
END DO
DO I=1,NORTH
WRITE(9,13011) MORT(I),M(I),NJ(I),NK(I)
13011 FORMAT(4I5)
END DO
13005 FORMAT(4I5/4I5/3I5)
READ(IRD,A) LOC1,LOC2,LOC3,LOC4,LOC5,LOC6,LOC7
WRITE(9,13008) LOC1,LOC2,LOC3,LOC4,LOC5,LOC6,LOC7
13008 FORMAT(7I5)
TA=1.0
IF(T(I).EQ.TZ(I)) TA=0.0
WRITE(9,13012) TA
13012 FORMAT(7F10.0)
IOP=1 ! I.D.# OF STRESS OUTPUT LOCATION SET
ISKIFO=0
DO I=1,NTOE
KGM=0
if(i.gt.1.and.
.TH. .AND.TZ(I).EQ.TZ(I-1).AND.MAT(I).EQ.MAT(I-1)) then
KGM1=NN(I,7)-NN(I-1,7)
KGM2=NN(I,8)-NN(I-1,8)
KGM3=NN(I,9)-NN(I-1,9)
KGM4=NN(I,6)-NN(I-1,6)
KGM5=NN(I,3)-NN(I-1,3)
KGM6=NN(I,4)-NN(I-1,4)
KGM7=NN(I,1)-NN(I-1,1)
KGM8=NN(I,2)-NN(I-1,2)
KGMX=MAX0(KGM1,KGM2,KGM3,KGM4,KGM5,KGM6,KGM7,KGM8)
KGMMN=MIN0(KGM1,KGM2,KGM3,KGM4,KGM5,KGM6,KGM7,KGM8)
IE(KGMX,EQ.KGMMN) KGM=KGMX
end if
KGP=0
if(i.lt.NTOE.and.
.IDM(I).EQ.IDM(I+1).AND.MAXES(I).EQ.MAXES(I+1).AND.IOP.EQ.IOP
.TH. .AND.TZ(I).EQ.TZ(I+1).AND.MAT(I).EQ.MAT(I+1)) then
KGP1=NN(I+1,7)-NN(I,7)
KGP2=NN(I+1,8)-NN(I,8)
KGP3=NN(I+1,9)-NN(I,9)
KGP4=NN(I+1,6)-NN(I,6)
KGP5=NN(I+1,3)-NN(I,3)
KGP6=NN(I+1,4)-NN(I,4)
KGP7=NN(I+1,1)-NN(I,1)
KGP8=NN(I+1,2)-NN(I,2)
KGPX=MAX0(KGP1,KGP2,KGP3,KGP4,KGP5,KGP6,KGP7,KGP8)
KGPMN=MIN0(KGP1,KGP2,KGP3,KGP4,KGP5,KGP6,KGP7,KGP8)
IE(KGPX,EQ.KGPMN) KGP=KGPX
end if
ISKIP=0
ISKIP=1
KG2=0
if (iskipo.eq.0 .and. kgp.gt.0) kgz=kgp

if (iskip.eq.0) THEN

WRITE(9,A) I,ISKIPO,ISKIP,KGM,KGP

WRITE(9,13025) I,il(I),NNH(I),JX=1,NTH

END IF

13025 FORMAT(I5,I15,215,F10.0,15,10X,I5/16I5/15)

ISKIPO=ISKIP

1302 FORMAT(I5,I15,215,F10.0,15X,I5/15)

NBMN=0

DO J=1,NBF

NB=99999

DO I=1,NBF

IF (NBF(I).LT.2:NM. .AND. NBF(I).GT.2:NBMN) THEN

NBMN=NBF(I)

IM=I

END IF

1303 FORMAT(I5,3F10.4)

WRITE(9,1303) NBF(I),EX(I),EY(I),FZ(I)

END DO

1304 FORMAT(I5,4X,'1',3F10.4)

WRITE(9,1305)

NBMN=NBMN+1

1306 FORMAT(215,7h 1 1 1,'',I5)

A-16
do i=1,ntono
c
if(mip(i,2).ne.0) then
   c- writing in the order they are given in the input file-----
do i=1,ntnd
   ii=nond(i)
do j=1,3
   write(9,8128) mip(ii,1),mip(ii,2),j,11
   end do
   8128 format(2i5,i3,5x,' ',i5)
end if
end do
end

CLOSE (UNIT=9)
STOP
END

subroutine split (NTON,ntd,idir,nond,mip,x,y,z,T,ntype,nos,NN,ntoe)
dimension nond(l),mip(4000,2),iface(9),jface(9),nos(l),T(l),
x(400),y(400),z(400),xn(3000,21)
NFF=4
IF (NTYPE.EQ.21) NFF=9
STORE COORD. OF THE NODES TO REAL
 CORRECTIONS FOR DOUBLE NODES IN x or y or z -direction--
DO 819 I=1,4
819 JFACE(1)=I+4
819 JFACE(1)=1
819 JFACE(5)=13
819 JFACE(6)=14
819 JFACE(7)=27
819 JFACE(8)=16
819 JFACE(9)=15
819 JFACE(5)=9
819 JFACE(6)=10
819 JFACE(7)=26
819 JFACE(8)=12
819 JFACE(9)=11
NTELR=NOS(1)+NOS(2) ! NTELR--TOTAL NUMBER OF ELEM. IN AN ELE-LAYER
IF (IDIR.LT.1.OR.IDIR.GT.3) IDIR=3
IF (IDIR=2) 901,902,903
901 JFACE(1)=2
901 JFACE(3)=3
901 JFACE(3)=6
901 JFACE(4)=7
901 JFACE(1)=1
901 JFACE(2)=4
901 JFACE(3)=5
901 JFACE(4)=8
901 JFACE(5)=10
901 JFACE(6)=18
901 JFACE(7)=23
901 JFACE(8)=19
IFACE(9) = 14
JFACE(5) = 12
JFACE(6) = 17
JFACE(7) = 22
JFACE(8) = 20
JFACE(9) = 16
NTELR = 1
GO TO 903
JFACE(1) = 3
JFACE(2) = 4
JFACE(3) = 2
JFACE(4) = 1
JFACE(5) = 6
JFACE(6) = 5
JFACE(7) = 11
JFACE(8) = 19
JFACE(9) = 25
JFACE(10) = 20
JFACE(11) = 15
JFACE(12) = 9
JFACE(13) = 18
JFACE(14) = 24
JFACE(15) = 17
JFACE(16) = 13
NTELR = NOE(1)
CONTINUE
DO 701 I = 1, NIU - 1
ICHANGE = 0
DO 702 J = 1, NIU - 1
JJ = J + 1
 IF (NOND(J) .LT. NOND(JJ)) GO TO 702
ICHANGE = 1
AA = NOND(J)
NOND(J) = NOND(JJ)
NOND(JJ) = AA
CONTINUE
IF (ICHANGE .EQ. 0) GO TO 703
CONTINUE
DO 830 I = 1, NIU
XD(I) = X(NOND(I))
YD(I) = Y(NOND(I))
ZD(I) = Z(NOND(I))
END DO
END
C-- CORRECTIONS DOUB. NODES. ENDS.--
IF(Y(J).NE.YD(I)) GO TO 820
IF(Z(J).NE.ZD(I)) GO TO 820
IF (DIF.GT.1.0E-13) GO TO 920
KND=J
GO TO 825 ! GET OUT OF J LOOP
820 CONTINUE
825 CONTINUE
C BEGIN CHANGING NODE NUMBERS & COORD.
DO J=I,NTOE
DO K=1,ntype
IF (NN(J,K).GT.KND) THEN
NN(J,K)=NN(J,K)+1
END IF
END DO
END DO
C-- substitution FOR DOUB. NODES : or y or z dir.--
DO K=1,NTOE
DO M=1,NFF
MM=IFACE(M)
IF (NN(K,MM).EQ.KND) THEN
MJ=IFACE(M)
NN(K+NTELR,MJ)=KND+1
END IF
END DO
C CHANGING COORD.
DO J=NTON,KND,-1
X(J+1)=X(J)
Y(J+1)=Y(J)
Z(J+1)=Z(J)
END DO
NTON=NTON+1
830 CONTINUE
return
end
 subroutine delete(nonz,nony,nonx,nd,nid,nond)
dimension nond(1)
1xx=nonz
1yy=nonx
1zz=nony
IFNO=6
DO 422 I=1,NONZ
DO 422 J=1,NONY
DO 422 K=1,NOX
IF (MOD(I,2).EQ.0) GO TO 411
IF (MOD(J,2).NE.0.OR.MOD(K,2).NE.0) GO TO 422
1-odd, j,k-even........
SUBROUTINE MESHPL (XX,YY,NUMX,NUMY,NUMZ,NIP,IPLOT,NTYPE)

LOGICAL ETX,ESC
CHARACTER A1,JUNK,MESH
DATA TX/3/,ESC/133/
DIMENSION X(4000),L(000),XX(1),YY(1)

NTOT=NUMX*NUMY*NUMZ
DO 1=1,NTOT
X(I)=XX(I)
Y(I)=YY(I)
1 IX=NUMZ
IY=NUMX*NUMZ

SCALING THE COORDINATES--------

XMIN=5000.
XMAX=-5000.
YMIN=5000.
YMAX=-5000.
DO 100 I=1,NTOT
IF (X(I).GT.XMAX) XMAX=X(I)
IF (X(I).LT.XMIN) XMIN=X(I)
IF (Y(I).GT.YMAX) YMAX=Y(I)
IF (Y(I).LT.YMIN) YMIN=Y(I)
WRITE (5,A,20)'TYPE LEVEL no. to plotted.'
20          A =5,x,' "'
READ (5, A) LEVEL
N = LEVEL
XLL = XMIN
YLL = YMIN
XMM = XMAX
YMM = YMAX
XL = XMM - XLL
YL = YMM - YLL
WRITE (5, A) 'DO YOU WANT TO GIVE X,Y LIMITS..? (Y/N)'
READ (5, 33) MESH
IF (MESH.EQ. 'Y' .OR. MESH.EQ. 'y') THEN
WRITE (5, A) 'ENTER X- LIMITS'
READ (5, A) XLL, XMM
WRITE (5, A) 'ENTER Y- LIMITS'
READ (5, A) YLL, YMM
XL = XMM - XLL
YL = YMM - YLL
END IF
XO = 650.0
YO = 1246.0
Xc = 8800.0/XL
Ye = 6000.0/YL
SC = xc
IF (Xcye1.9t.6000.0) SC = yc
IF (XL.LT.YL) THEN
WRITE (5, A) 'R090;'
WRITE (5, A) 'IP;IW;'
END IF
XMM = (XMM-XLL)ASC+XO
YMM = (YMM-YLL)ASC+YO
XLL = XO
YLL = YO
WRITE (5, 9999) ESC, ESC, ESC, ESC
J1=level
J2=(Nony-1)*A1+y+j1
DO 111 I=1,Nony
111 TYPE A,Y(I),YLL
 CALL LIMIT (Y,YLL,j1,j2,iy,j1)
 CALL LIMIT (Y,YMM,j1,j2,iy,j2)
 J2=IF2(Y,YMM,j1,Nony,1)
 TYPE A, '20 LOOP', J1, J2
 DO 20 J=J1,J2,iy

NX1=j
NX2=nx1+iy-1
 CALL LIMIT (X,XLL,NX1,NX2,IX,NX1)
 CALL LIMIT (X,XMM,NX1,NX2,IX,NX2)
 NX1=IF1(X,XLL,NX1,NX2,IX)
 NX2=IF2(X,XMM,NX1,NX2,IX)
 TYPE A,' 10 LOOP',NX1,NX2
 TYPE A,X(NX1),Y(NX1),X(NX2),Y(NX2)
 DO 10 I=NX1,NX2,IX
 IF (i.eq.nx1) WRITE (90,101), X(I),Y(I)
 WRITE (90,104), X(I),Y(I)
 continue
 WRITE (90,A) 'PU;

I1=level
I2=(NONX-1)*A1+1
 CALL LIMIT (X,XLL,I1,I2,IX,IX)
 CALL LIMIT (X,XMM,I1,I2,IX,IX)
 I1=IF1(X,XLL,I,NONX,1)
 I2=IF2(X,XMM,I,NONX,1)
 TYPE A,' 30 LOOP',I1,I2
 DO 30 I=I1,I2,IX
 NY1=i
NY2=Ny1+(Nony-1)*iy
 CALL LIMIT (Y,YLL,NY1,NY2,iy,NY1)
 CALL LIMIT (Y,YMM,NY1,NY2,iy,NY2)
 NY1=IF1(Y,YLL,NY1,NY2,iy)
 NY2=IF2(Y,YMM,NY1,NY2,iy)
 TYPE A, ' 40 LOOP ',NY1,NY2
 DO 40 J=NY1,NY2,iy
 IF (J.EQ.NY1) WRITE (90,101), X(J),Y(J)
 WRITE (90,102), X(J),Y(J)
 continue
 FORMAT (' PU;PA',2E11.3,'
 WRITE (5,A) 'DO YOU NEED NODE Nos. (Y/N)...'
 READ (5,33) JUNK
 FORMAT (A)
 IF (JUNK.EQ.'Y'.OR.JUNK.EQ.'y') GO TO 44
 GO TO 66
 WRITE (5,A) '.. ENTER SSX,SSY...'

A-22
1189 ! READ (5,A) SSX,SSY
1190 44 SSX=0.1
1191 SSY=0.15
1192 ! -- WRITING THE NODE NUMBERS---------
1193 WRITE (90,A)'SIO.1,0.15,'
1194 J1=level
1195 J2=(nony-1)Aiy+j1
1196 C CALL LIMIT (Y,YLL,j1,j2,iy,J1)
1197 C CALL LIMIT (Y,YMM,J1,j2,iy,J2)
1198 DO 100 J=J1,J2,iy
1199 NX1=j
1200 NX2=nx1+iy-ix
1201 C CALL LIMIT (X,XLL,NX1,NX2,IX,NX1)
1202 C CALL LIMIT (X,XMM,NX1,NX2,IX,NX2)
1203
1204 ! TYPE A,(KK,KS=NX1,NX2,IX)
1205 DO 100 IN=NX1,NX2,IX
1206 NX=(IN-NX1)/IX+1 !skipping the face nos.
1207 IF (MOD(J,2).EQ.0.AND.MOD(NX,2).EQ.0.and.
1208 .ntype.eq.2) GO TO 100
1209 I=IN
1210 IF (IPLOT.EQ.2) I=MIP(IN,1)
1211 SPL=-0.8
1212 XXX=X(IN)
1213 YYY=Y(IN)
1214 IF (XXX.LT.XLL.OR.YYY.LT.YLL) GO TO 100
1215 IF (XXX.GT.XMM.OR.YYY.GT.YMM) GO TO 100
1216 CALL SYMB (XXX,YYY,SPL,I)
1217 IF (IPLAN.EQ.1) GO TO 100
1218 I=MIP(IN,2)
1219 SPL=0.2
1220 IF (I.NE.0) CALL SYMB (XXX,YYY,SPL,I)
1221 WRITE (90,A) 'FU;'
1222 WRITE (90,A), 'SPO;'
1223 WRITE (5,A) 'TYPE another LEVEL no. to plotted..'
1224 READ (5,A) LEVEL
1225 IF (LEVEL.NE.0) go to 888
1226 RETURN
1227 END
1228 SUBROUTINE SYMB (X,Y,SPL,I)
1229 LOGICAL Al ETX,ESC
1230 DATA ETX/*3/,ESC/*33/
1232 NC=1
1233 IF (I.GT.9) NC=2
1234 IF (I.GT.99) NC=3
1235 IF (I.GT.999) NC=4
1236 IF (I.GT.9999) NC=5
1237 IF (I.GT.99999) NC=6
1238 SPC=-(0.33+0.5A(NC-1))2.
1239 WRITE (90,101), X,Y
1240 IF (NC.EQ.1) WRITE (90,201) SPC,SPL,I,ETX
1241 IF (NC.EQ.2) WRITE (90,202) SPC,SPL,I,ETX
1242 IF (NC.EQ.3) WRITE (90,203) SPC,SPL,I,ETX

A-23
1243 IF (NC.EQ.4) WRITE (90,204) SPC,SPL,I,ETX
1244 IF (NC.EQ.5) WRITE (90,205) SPC,SPL,I,ETX
1245 IF (NC.EQ.6) WRITE (90,206) SPC,SPL,I,ETX
1246 101 FORMAT ('PU;PA',2F11.3,':')
1247 201 FORMAT ('CP',2F6.2,';LB',I1,A2)
1248 202 FORMAT ('CP',2F6.2,';LB',I2,A2)
1249 203 FORMAT ('CP',2F6.2,';LB',I3,A2)
1250 204 FORMAT ('CP',2F6.2,';LB',I4,A2)
1251 205 FORMAT ('CP',2F6.2,';LB',I5,A2)
1252 206 FORMAT ('CP',2F6.2,';LB',I6,A2)
1253 RETURN
1254 END
1255
1256 SUBROUTINE LIMIT (A,AL,I1,I2,IN,IF1)
1257 c-- to find the lower limit of do loop...
1258 DIMENSION A(1)
1259 DO 10 I=I1,I2,IN
1260 C TYPE A,A(I),AL
1261 IF (A(I).GE.(AL-I.OE-03)) THEN
1262 IF1=I
1263 RETURN
1264 END IF
1265 10 CONTINUE
1266 END
1267
1268 INTEGER FUNCTION IE2(A,AL,I1,I2,IN)
1269 c-- to find the lower limit of do loop...
1270 DIMENSION A(1)
1271 IE2=I2
1272 DO 10 I=I1,I2,IN
1273 C type A,A(I),AL
1274 IF (A(I).GE.(AL-I.OE-03)) THEN
1275 IF1=I
1276 RETURN
1277 END IF
1278 10 CONTINUE
1279 END
1280
1281 c====SUBROUTINE HOLE(rx,NONX,NONY,NONZ,XX,YY,ZZ,X,Y,Z)===
1282 C GENERATING POLAR COORDINATES AND POLAR MESH--
1283 DIMENSION XX(1),YY(1),ZZ(1),X(1),Y(1),R(100),NC(4)
1284 IX=NONZ
1285 IY=NONX\NONZ
1286 DO 31 I=1,NONX
1287 rad=rx
1288 if (xx(i).ne.0.0) rad=sqrt(xx(i)*A2+rx*A2)
DO 31 J=1, NONY
   xx=xx(1)
   IF (yy(j).le.(xx-eps)) then
      xx=rad
      IF (yy(j).ne.0.) xx=sqrt(rad+5-yy(j)**2).
      nbk=(j-1)*iy+(i-2)*iix+1
      TYPE A,NBK
      IF ((xx-n(nbk)).lt.(xx(2)-xx(1))) XX=xx(nbk)+xx(2)-xx(1)
   end if
   nyy=(J-1)*iyy+(i-1)*iix
   DO 31 K=1, NNY
   N=nyy+K
   X(N)=xx
   IF (I.EQ.NNY) X(N)=XY(NNY)
   Y(N)=YY(J)
   Z(N)=ZZ(K)
   CONTINUE
   DO 31 K=1, NNY
   N=(J-1)*iyy+(i-1)*iix+K
   X(N)=(X(N-IX)+X(N+IX))/2.0
   CONTINUE
   do 30 j=1, nony
   do 30 i=1, nonx
   n=(j-1)*iy+(i-1)*iix+1
   type a,n,x(n),y(n)
   CONTINUE
   RETURN
END
APPENDIX - B

LISTING OF THE MAIN CODE 'KSAP II'
SIMPLIFIED VERSION OF SAP4 FOR USING ELEMENT TYPE 8 ONLY

September 1987

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

REAL*4 T,T

COMMON /JUNK/ HED(12), JUK(406)
COMMON /ELPAR/ hPAR(14), HUNP, MBAND, MELTYPE, N1, N2, N3, N4, MS, NIOT, NEQ
COMMON /EM/ QQQ(2846)
COMMON /DYN/ IDUS(11), NDYN
COMMON /TAPES/ NQQ(6)
COMMON /EXTRA/ MUX1, NT1, NOSV, MT10, KEQB, NUMEL, T(10)
COMMON /SOL/ NBL0, NEOF, LL, IDUM, NEIG, NAD, NVV, ANORM, NEQ

PROGRAM CAPACITY CONTROLLED BY THE FOLLOWING TWO STATEMENTS...

COMMON A(650001) !CHANGE NIOT ALSO

OPEN SCRATCH FILES

OPEN (UNIT=1, STATUS='UNKNOWN', FORM='UNFORMATTED')

open (unit=1, file='SCR[CASW.EMANI]', status='scratch',
form='unformatted')
open (unit=2, file='SCR[CASW.EMANI]', status='new', blocksize=4800,
form='unformatted')
open (unit=3, file='U1[CASW.EMANI]', status='new', blocksize=4800,
form='unformatted')
open (unit=3, file='MSA0[ESK1]', status='new', blocksize=4800,
form='unformatted')
open (UNIT=4, FILE='SCR[CASW.EMANI]', STATUS='NEW', BLOCKSIZE=4800,
FORM='UNFORMATTED')
open (UNIT=5, FILE='SCR[CASW.EMANI]', STATUS='UNKNOWN', BLOCKSIZE=4800,
FORM='UNFORMATTED')
open (UNIT=8, FILE='SCR[CASW.EMANI]', STATUS='scratch',
FORM='UNFORMATTED')
open (UNIT=9, FILE='SCR[CASW.EMANI]', STATUS='scratch',
FORM='UNFORMATTED')
open (UNIT=15, STATUS='scratch', FORM='UNFORMATTED')
open (UNIT=16, FILE='SCR[CASW.EMANI]', STATUS='scratch',
FORM='UNFORMATTED')
open (UNIT=18, FILE='SCR[CASW.EMANI]', STATUS='scratch',
FORM='UNFORMATTED')
open (UNIT=19, file='workdone.wok', status='new')
open (unit=33, file='SCR\EMANI\DISP.dat', status='new')
56 open (unit=34, file='ksapout.dat', status='new')
58 USE (UNIT=15, FILE='SCR\EMANI\', STATUS='NEW', DLUXKSIZE=4800,
59 .FORM='UNFORMATTED')

! THE following should be 1 less than A() dimension
MTOT= 650000 ! 300000

C USE THE IBM FORTRAN EXTENDED ERROR HANDLING FACILITY TO
C ELIMINATE PRINTOUT OF UNDERFLOW ERROR MESSAGE (ERROR NUMBER 208)
CALL ERRSET (208,256,-1,1)

CALL STIME

MT8 = 8
REWIND MT8
MT10= 10
REWIND MT10
N1=1

PROGRAM CONTROL DATA

CALL TTIME(T(1))
READ (5,100,END=990) HED,NUMNP,NELTYP,LL,NF,NDYN,MODEX,NAD,
1 KEQB,N1OSV,NDOF
IF(MODEX.GT.0) MODEX = 1
IF (NUMNP.EQ.0) STOP
WRITE (33,200) HED,NUMNP,NELTYP,LL,NF,NDYN,MODEX,NAD,
KEQB,N1OSV,NDOF
WRITE (19,299) HED !WORKDONE.WOK FILE TITLE....
WRITE (34,299) HED !KSAPOUT.DAT (STRESSES) FILE TITLE....
IF(KEQB.LT.2) KEQB = 99999
IF (NDYN.NE.0) LL=1
IF(NDYN.LT.0) GO TO 20
IF(LL.GE.1) GO TO 10
WRITE (33,300)
STOP

DATA PORTHOLE SAVE
IF(MODEX.EQ.1)
WRITE (NT8) HED,NUMNP,NELTYP,LL,NF,NDYN
KDYN = IABS(NDYN) +1
IF(KDYN.LE.5) GO TO 14
WRITE (33,310) KDYN
STOP
RE-START MODE ACTIVATED IF NDYN.EQ.-2 OR NDYN.EQ.-3
IF(NDYN.LT.0) GO TO 20

INPUT JOINT DATA
N2 = N1 + 6 * NUMNP
N3 = N2 + NUMNP
N4 = N3 + NUMNP
N5 = N4 + NUMNP
N6 = N5 + NUMNP
IF (N6 > MTOT) CALL ERROR (N6 - MTOT)
CALL INPUTJ (A(N1), A(N2), A(N3), A(N4), A(N5), NUMNP, NEQ)

CALL TIME(T(2))
MBAND = 0
NUMEL = 0
REWIND 1
REWIND 2
DO 900 M = 1, NELTYPE
READ (5, 1001) NPAR
DATA PORHOLE SAVE
IF (MODEX.EQ.1) WRITE (NT8) NPAR
WRITE (1) NPAR
NUMEL = NUMEL + NPAR(2)
MTYPE = NPAR(1)
CALL ELTYPE (MTYPE)
CONTINUE

DETERMINE BLOCK SIZE
ADDSIF
LL1 = LL + NDOF  \* in the following LL is replaced with LLI
NEOB = (MTOT - 4 * LL) / (MBAND + LLI + 1) / 2  \* modified with NDOF

OVER-RIDE THE SYSTEM MATRIX BLOCKSIZE WITH THE INPUT (NON-ZERO) VALUE, KEQB.
THIS OVER-RIDE ENTRY IS TO ALLOW PROGRAM CHECKING OF MULTI-BLOCK ALGORITHMS WITH WHAT WOULD NORMALLY BE ONE BLOCK DATA.
IF (KEQB .LT. NEQB) NEQB = KEQB
GO TO (690, 700, 700, 700, 730), KDYN

STATIC SOLUTION
690 CONTINUE
NEOB1 = (MTOT - MBAND) / ((2 * (MBAND + LLI + 1)) + 1)
NEOB2 = (MTOT - MBAND - LLI (MBAND - 2)) / ((3 * LLI + MBAND + 1)
IF (NEOB1 .LT. NEQB) NEQB = NEOB1
IF (NEOB2 .LT. NEQB) NEQB = NEOB2
NBLOCK = (NEQ - 1) / NEQB + 1

b- 3
IF(NEQB.GT.NEQ) NEQB=NEQ
GO TO 790
EIGENSOLUTION
1. DETERMINANT SEARCH ALGORITHM
IF (NEQB.LT.NEQ) GO TO 710
NIM=3
NC=NE + NIM
NUM=6
NCA=NEQAMAXO(MBAND,NC)
MTOT=NCA + 4ANEQ + 2ANUMANEQ + 5ANC
NEIG=0
IF(MT0T.LE.NTOT) GO TO 720
2. SUBSPACE ITERATION ALGORITHM
NV=MINO(2ANEQ,NE+6)
IF (NAD.NE.0) NV=NAD
NEQB1=(MTOT - MBAND)/(2AMBAND + 1)
NEQB2=(MTOT - MBAND - 2ANV - NVA(MBAND-2))/(3ANV + MBAND + 1)
NEQB3=(MTOT - 3ANVANV - 3ANV)/(2ANV + 1)
NEQB4=(MTOT - 6ANV)/(1 + MBAND)
IF (NEQB1.LT.NEQB) NEQB=NEQB1
IF (NEQB2.LT.NEQB) NEQB=NEQB2
IF (NEQB3.LT.NEQB) NEQB=NEQB3
IF (NEQB4.LT.NEQB) NEQB=NEQB4
NEIG=1
CONTINUE
NBLOCK = (NEQ-I)/NEQB +1
IF (NE0B.GE.NEQ) NEUB=NEO
HISTORY OR SPECTRUM ANALYSIS
KREM = 1000
MTOT = NBLOCKANEQBANE + KREM
IF(MT0T.LT.NTOT) AWRITE (33,320)
GO TO 790
STEP-BY-STEP DIRECT INTEGRATION
CONTINUE
DISPLACEMENT COMPONENTS FOR DIRECT OUTPUT (ANSDA)
NN2 = NEO
DISPLACEMENT COMPONENTS REQUIRED FOR RECOVERY OF ALL OF THE
REQUESTED ELEMENT STRESS COMPONENTS (ANSSA)
NN3 = NEO
1. DECOMPOSITION
NEQB1 = (MTOT-NN2-NN3-NEO-MBAND)/(2AMBAND+1)
217 C 2. TIME INTEGRATION PHASE
218 C
219 C mcal = MBAND + 2A(NN2 + NN3) + SANEQ + (2AMBAND + 1)
220 221
222 C write (33,555) mcal
223 555 format('5x, minimum dimension, MTOT, required for array A( )
224 . = ', 10/SX,50(H1+)//)
225 IF(MTOT.LE.MCAL) STOP 'Abnormal stop as dim. of A is insufficient
226
227 C NEQB2 = (MTOT-AMBAND-2A(NN2+NN3)-SANEQ)/(AMBAND+1)
228 C
229 C IF(NEQBl .LT. NEQB) NEQB = NEQB1
230 IF(NEQB2 .LT. NEQB) NEQB = NEQB2
231 IF(NEQB .GT. NEQ) NEQB = NEQ
232 NBLOCK = (NEQ-1)/NEQB + 1
233 C
234 C 3. INPUT PHASE
235 C
236 C NUMBER OF TIME FUNCTIONS (ANENA)
237 C NN2 = 10
238 C MAXIMUM NUMBER OF FUNCTION DEFINITION POINTS (ANXLPA)
239 C NN3 = 40
240 C
241 C NN4 = GANUMNP + 2ANN2SANEQ
242 IF(NN4 .GT. MTOT) WRITE (33,320)
243 IF(NN4 .GT. MTOT) WRITE (33,320)
244 IF(NN4 .GT. MTOT) WRITE (33,320)
245 IF(NN4 .GT. MTOT) WRITE (33,320)
246 IF(NN4 .GT. MTOT) WRITE (33,320)
247 IF(NN4 .GT. MTOT) WRITE (33,320)
248 C
249 790 CONTINUE
250 C
251 C INPUT MODAL LOADING
252 C
253 C N3=N2+NEQBALL
254 C N4=N3+GALL
255 C WRITE (33,201) NEQ, MBAND, NEQB, NBLOCK
256 C
257 C CALL TTIME(I(3))
258 C
259 C CALL INL(A(N1), A(N2), A(N3), A(N4), NUMNP, NEQB, LL)
260 C
261 C CALL TTIME(I(4))
262 C
263 C FORM TOTAL STIFFNESS
264 C
265 C NE2B=2ANEOB
266 C N2=N1+NEQBAMBAND
267 C N3=N2+NEQBALL
268 C N4=N3+GALL
269 C NN2=N1+NE2BAMBAND
270 C NN3=NN2+NE2BALL
271  NN4=NN3+4ALL
272  CALL ADDSIF (A(N1), A(NN2), A(NN3), A(NN4), NUMEL, NBLCK, NE2B, LL, MBAND, L, ANORM, NNV)
273  CALL TTIME(T(S))
274  SOLUTION PHASE
275  IF(NODEX.EQ.0) GO TO 32
276  DO 31 I=6,10
277  T(I) = T(5)
278  GO TO 90
279  CALL SOLEG
280  CALL TTIME(T(6))
281  DO 33 I=7,10
282  T(I) = T(6)
283  GO TO 90
284  EIGENVALUE EXTRACTION
285  T(6) = T(5)
286  CALL SOLEG
287  CALL TTIME(T(7))
288  IF(NDYN.LT.0) GO TO 52
289  CALL SOLEG
290  CALL TTIME(T(7))
291  DO 53 I=1,6
292  T(I+1) = T(I)
293  REWIND 2
294  READ (2) NEQ, NBLCK, NEOB, MBAND, N1, NE, (QQQ(I), I=1, NE)
295  REWIND 55
296  IMAX=NEQ+NEB
297  READ (55) (A(I), I=1, NE)
298  DO 56 L=1,NBLCK
299  READ (55) (A(I), I=1, IMAX)
300  CONTINUE
301  CALL HISTRY
302  CALL TTIME(T(8))
303  T(9) = T(8)
435 \text{T}(10) = \text{T}(8)
436 \text{GO TO 90}
437 C
438 C \text{RESPONSE SPECTRUM ANALYSIS}
439 C
440 \text{GO T} = \text{E}(5)
441 \text{IF} \text{(NDYN.LT.0)} \text{GO TO 62}
442 C \text{CALL SOLEIG}
443 C \text{CALL TTIME (T(7))}
444 \text{T}(8) = \text{T}(7)
445 \text{GO TO 64}
446 \text{DO 63 I=1,7}
447 \text{T(I+1) = T(I)}
448 \text{REWIND } 2
449 \text{READ (2) NEQ,NBLOCK,NEOB,MBAND,N1,NE}
450 \text{REWIND } 55
451 \text{IMAX=NEOBANF}
452 \text{READ (55) (A(I),I=1,NE)}
453 \text{DO 66 L=1,NBLOCK}
454 \text{READ (55) (A(I),I=1,IMAX)}
455 \text{CONTINUE}
456 C \text{CALL RESPEC}
457 \text{CALL TTIME (T(9))}
458 \text{T}(10) = \text{T}(9)
459 \text{GO TO 90}
460 C
461 \text{STEP-BY-STEP (DIRECT INTEGRATION) ANALYSIS}
462 C
463 \text{DO 71 I=6,9}
464 \text{T(I) = T(5)}
465 C \text{CALL STEP}
466 \text{CALL TTIME(T(10))}
467 C
468 \text{COMPUTE AND PRINT OVERALL TIME LOG}
469 C
470 \text{IT } = 0.0
471 \text{DO 95 I=1,9}
472 \text{T(I) = T(I+1)-T(I)}
473 \text{IT = IT + T(I)}
474 \text{CONTINUE}
475 C
476 \text{WRITE (33,203) (T(K),K=1,9),IT}
477 C
478 \text{GO TO 5}
479 \text{CONTINUE}
480 C \text{CLOSE ALL SCRATCH FILES}
481 \text{CLOSE (UNIT=1)}
482 \text{CLOSE (UNIT=2)}
483 \text{CLOSE (UNIT=3)}
484 \text{CLOSE (UNIT=4)}
485 \text{CLOSE (UNIT=55)}
486 \text{CLOSE (UNIT=8)}
487 \text{CLOSE (UNIT=9)}
488 \text{CLOSE (UNIT=15)}
IMPLICIT REAL A(I,J, K)

CALLED BY: MAIN

FORMS GLOBAL EQUILIBRIUM EQUATIONS IN BLOCKS

DIMENSION A(NQB, MBAND), B(NQB, LL), STR(I, LL), IMASS(NQB)

COMMON /BET4//NT, NOT, ALEA, DT, BETA, NEN, NGM, NAT, NDYN

COMMON /EM/ LR, MB, LM(G3), IFAD, SS(2331)

COMMON /EXTRA/ MODEX, MB, IFILL(14)

NEQB = NQB/2
K = NEQB + 1
X = NBLOCK
MB = DSQRT(X)
MB = MB/2 + 1
MEB = MB*E2B
MM = 1
NDEG = 0

NVV = 0

ANORM = 0

NSHIFT = 0

REWRITE 3

REWRITE 9

READ ELEMENT LOAD MULTIPLIERS

WRITE (33, 2000)

DO 50 L = 1, LL

READ (5, 1002) (STR(I, L), I = 1, 4)

50 WRITE (33, 2002) (STR(I, L), I = 1, 4)

IF(MODEX.EQ.0) WRITE (8) STR

FOR A STEP-BY-STEP ANALYSIS (NDYN.EQ.4) READ THE SOLUTION

CONTROL CARD. THE TIME STEP (DT) AND THE DAMPING COEFFICIENTS

(ALPHA/BETA) ARE REQUIRED FOR THE ASSEMBLY OF THE EFFECTIVE

SYSTEM STIFFNESS MATRIX IN THIS ROUTINE.

IF (NDYN.NE.4) GO TO 65

READ (5, 1004) NE, NGM, NAT, NT, NOT, DT, ALEA, BETA

WRITE (33, 2004) NE, NGM, NAT, NT, NOT, DT, ALEA, BETA

IF (NAT.EQ.0) NAT = 1

IF (NOT.EQ.0) NOT = 1

IF (DT.GT.1.0E-12) GO TO 55

WRITE (33, 3000)

STOP

COMPUTE INTEGRATION COEFFICIENTS FOR ASSEMBLY OF EFFECTIVE

SYSTEM STIFFNESS (STEP-BY-STEP ANALYSIS ONLY)
55 TE1A = 1.4
56 DT1 = TETAADT
57 DT2 = DT1**2
58 A0 = (6.*3.AALEAADT1)/(DT3+3.ABETAADT1)
59 C
60 IF(MODEX.EQ.1) RETURN
61 C
62 FORM EQUATIONS IN BLOCKS (2 BLOCKS AT A TIME)
63 C
64 DO 1000 M=1,NBLOCK,2
65 DO 100 I=1,NE2B
66 DO 100 J=1,MBAND
67 100 A(I,J)=0.
68 READ (3) ((B(I,L),I=1,NEQB),L=1,LL),(IMASS(I),I=1,NEQB)
69 IF (M.EQ.NBLOCK) GO TO 200
70 READ (3) ((B(I,L),I=K,NE2B),L=1,LL),(IMASS(I),I=K,NE2B)
71 200 CONTINUE
72 C
73 REWIND 55
74 REWIND 2
75 NA=55
76 IF (MM.NE.1) GO TO 75
77 NA=2
78 NUME=NUMEL
79 NUM7 =0
80 75 DO 700 N=1,NUME
81 READ (NA) LRD,ND,(LM(I),I=1,ND),(SS(I),I=1,LRD)
82 MSHET = ND A (ND+1)/2 +4 AND
83 DO 600 I=1,ND
84 LM=1-LM(I)
85 II=LM(I)-NSHIFT
86 IF (II.LE.0.OR.II.GT.NE2B) GO TO 600
87 IMS=I+MSHET
88 TMASS(II)=TMASS(II)+SS(IMS)
89 DO 300 L=1,LL
90 DO 300 J=1,ND
91 JJ=LM(J)*LmN
92 IF(JJ) 500,500,390
93 390 IF(JJ) 500,500,390
94 KK = ND A(ND+1)/2 + NDA(J-1)
95 300 B(J,J)=B(J,J)+SS(1+KK)ASTR(J,L)
96 DO 500 J=1,ND
97 JJ=LM(J)*LmN
98 IF(JJ) 500,500,390
99 390 IF(JJ) 500,500,390
100 390 IF(JJ) 500,500,390
101 KK = NDA1-(I-1)A1/2 +J-ND
102 GO TO 400
103 KK =NDAJ -(J-1)A1/2+I-ND
104 400 A(II,JJ)=A(II,JJ)+SS( KK)
105 500 CONTINUE
106 600 CONTINUE
107 C
108 DETERMINE IF STIFFNESS IS TO BE PLACED ON TAPE 55
109 C
110 IF (MM.GT.1) GO TO 700
111
DO 650 I=1,ND
542 II=LM(I) -NSHIFT
543 IF(II.GT.NE2B.AND.II.LE.NE2B) GO TO 660
544 650 CONTINUE
545 GO TO 700
546 660 WRITE (55) LRD,ND,(LM(I),I=1,ND),(SS(I),I=1,LRD)
547 NUM7=NUM7+1
548 C
549 700 CONTINUE
550 DO 710 L=1,NEQB
551 ANORM=ANORM + A(L,1)
552 IF (A(L,1).NE.0.) NDEG=NDEG + 1
553 IF (A(L,1).EQ.0.) A(L,1)=1.E+20
554 IF (TMASS(L).NE.0.) NVV=NVV + 1
555 710 CONTINUE
556 C
557 FOR STEP-BY-STEP ANALYSIS ADD THE MASS CONTRIBUTIONS TO
558 C THE EQUATION DIAGONAL COEFFICIENTS
559 C
560 IF(NDYN.NE.4) GO TO 716
561 DO 714 I=1,NEQB
562 714 A(I,1) = A(I,1) + AOK TMASS(I)
563 WRITE (4) ((A(I,J),I=1,NEQB),J=1,MBAND)
564 GO TO 718
565 716 WRITE (4) ((A(I,J),I=1,NEQB),J=1,MBAND),((B(I,L),I=1,NEQB),L=1,LL)
566 718 WRITE (9) (TMASS(I),I=1,NEQB)
567 C
568 IF(M.EQ.NBLOCK) GO TO 1000
569 DO 720 L=K,NE2B
570 ANORM=ANORM + A(L,1)
571 IF (A(L,1).NE.0.) NDEG=NDEG + 1
572 IF (A(L,1).EQ.0.) A(L,1)=1.E+20
573 IF (TMASS(L).NE.0.) NVV=NVV + 1
574 720 CONTINUE
575 C
576 IF(NDYN.NE.4) GO TO 726
577 DO 724 I=K,NE2B
578 724 A(I,1) = A(I,1) + AOK TMASS(I)
579 WRITE (4) ((A(I,J),I=K,NE2B),J=1,MBAND)
580 GO TO 728
581 726 WRITE (4) ((A(I,J),I=K,NE2B),J=1,MBAND),((B(I,L),I=K,NE2B),L=1,LL)
582 728 WRITE (9) (TMASS(I),I=K,NE2B)
583 C
584 IF (MM.EQ.MB) MM=0
585 MM=MM+1
586 1000 NSHIFT=NSHIFT+NE2B
587 IF (NDEG.GT.0) GO TO 730
588 WRITE (33,1010)
589 STOP
590 730 ANORM=(ANORM/NDEG)/1.E-8
591 C
592 RETURN
593 1002 FORMAT (4F10.0)
594 1004 FORMAT (5S,3E10.0)
SUBROUTINE BOUND
IMPLICIT REALA8(A-H,O-Z)
COMMON A(1)
COMMON /ELPAR/ NPAR(14), NUMNP, MBAND, NELTYP, N1, N2, N3, N4, N5, NIO, NEQ
COMMON /JUNK/ L1, L2, L3, IPAD, SIG(20), IFILL(386)
COMMON /EXTRA/ MODEX, N1OSV, NT1O, IFILL2(12)
IF (NPAR(1).EQ.0) GO TO 500
CALL CLAhP (NPAR(2), A(N1), A(N3), A(N4), NUMNP, MBAND)
RETURN
500 continue
WRITE (33,2002)
NUME=NPAR(2)
DO 800 M=1, NUME
CALL STRSC (A(N1), A(N3), NEQ,0)
WRITE (33,2001)
DO 800 L=L1, L2
CALL STRSC (A(N1), A(N3), NEQ,1)
WRITE (33,3002) MM, L, (SIG(1), I=1,2) !printing suppressed
IF (N1OSV.EQ.0) GO TO 500
CALL CLAMP (NPAR(2), A(N1), A(N2), A(N3), A(N4), NUMNP, MBAND)
RETURN
2001 FORMAT (/)
2002 FORMAT (48H B O U N D A R Y E L E M E N T F O R C E S /)
1H M O M E N T S , // 8H E L E M E N T , 3X, 4HLOAD, 14X, 5HFORCE,
9X, 6MMOMENT, // 8H NUMBER, 3X, 4HCASE, // 1X)
3002 FORMAT (1B,17,4X,2E15.5)
END
SUBROUTINE CALBAN (MBAND,NDIF,LN,XM,S,P,ND,NDM,NS)
IMPLICIT REALA(A-H,O-Z)
CALLED BY:  RUSS,IEAN,PLMAX,BRICK8,TPLATE,CLAMP,ELST3D,PILER
C-----CALCULATES BAND WIDTH AND WRITES STIFFNESS MATRIX ON TAPE 2
DIMENSION LN(1),XM(1),S(NDM,NDM),P(NDM,4)
COMMON /EXTRA/ MODEX,NT8,IFILL(14)
MIN=100000
MAX=0
DO 800 L=1,ND
   IF (LN(L).EQ.0) GO TO 800
   IF (LN(L).GT.MAX) MAX=LN(L)
   IF (LN(L).LT.MIN) MIN=LN(L)
800 CONTINUE
NDIF=MAX-MIN+1
IF (NDIF.GT.MBAND) MBAND=NDIF
IF (MODEX.EQ.1) GO TO 810
C
LRD=NDL(ND+1)/2+5*ND
WRITE(2) LRD,ND,(LN(I),I=1,ND),((S(I,J),J=1,ND),I=1,ND)
DO 30 NI=1,ND
   XM(NI)=0.0
20 S(NI,NJ)=0.0
30 CONTINUE
DO 540 NK=1,NS
   DO 40 NL=1,ND
   40 S(NI,NJ)=0.0
540 CONTINUE
SUBROUTINE CLAMP (NUMEL,ID,X,Y,Z,NUMNP,MBAND)
IMPLICIT REALA(A-H,O-Z)
COMMON /EN/LN(24),ND,NS,S(24,24),P(24,4),XM(24),SA(12,24),IT(12,4),
      IFILL1(3048)
DIMENSION X(1),Y(1),Z(1),ID(NUMNP,1)
COMMON /JUNK/ R(6),RM(4),IFILL2(410)
COMMON /EXTRA/ MODEX,NT8,IFILL3(14)
WRITE (33,2000) NUMEL
NS=2
ND=6
READ(5,1005) km
WRITE (33,2005) km
IF(MODEX.EQ.1) WRITE (NT8) km
DO 30 NI=1,ND
   XM(NI)=0.0
DO 20 NJ=1,ND
   S(NI,NJ)=0.0
20 CONTINUE
595 1010 FORMAT (SING STRUCTURE WITH NO DEGREES OF FREEDOM, CHECK DATA )
596 2000 FORMAT (/// 10H STRUCTURE, 13X, 7HELEMENT, 4X, 4HLLOAD, 4X,
597 111MULTIPLIERS, / 10H LOAD CASE, 13X, 1MA, 9X, 1MB, 9X, 1MC, 9X, 1MD, / 1X)
598 2002 FORMAT (16, 7X, 4F10.3)
599 2004 FORMAT (45818 TEBYSTE PST0L10N,
600 2 5X, 35H NUMBER OF TIME VARYING FUNCTIONS =, 15 //
601 2 <V TIME VARYING MOTION INDICATOR =, 15 /

703 40 SA(NK, NL) = 0.0
704 GO 50 NI = 1, 4
705 TT(NK, NI) = 0.0
706 50 CONTINUE
707 NE = 0
708 WRITE (33, 2007)
709 210 KG = 0
710 MARK = 0
711 200 READ (5, 1000) HF, NI, NJ, NK, NL, KD, KR, KN, SD, SR, TRACE
712 IF (TRACE.EQ.0.) TRACE = 1.0E+10
713 LK = (KG.GT.0.) GO TO 250
714 KG = KN
715 IF (MODEX.EQ.1.) GO TO 230
716 IF (NJ.EQ.0.) GO TO 250
717 X1 = X(NI) - X(NJ)
718 Y1 = Y(NJ) - Y(NI)
719 Z1 = Z(NJ) - Z(NI)
720 X2 = X(NL) - X(NK)
721 Y2 = Y(NL) - Y(NK)
722 Z2 = Z(NL) - Z(NK)
723 T1 = Y1*Z2 - Y2*Z1
724 T2 = Z1*X2 - Z2*X1
725 T3 = X1*Y2 - X2*Y1
726 GO TO 260
727 250 X1 = X(NI) - X(NP)
728 Y2 = Y(NI) - Y(NP)
729 Z3 = Z(NI) - Z(NP)
730 260 XL = T1*T1 + T2*T2 + T3*T3
731 XL = DSORT(AL)
732 IF (XL.GT.1.0E-6) GO TO 270
733 WRITE (33, 3000)
734 3000 FORMAT (32H NOAAA ERROR ZERO ELEMENT LENGTH, / 1X)
735 STOP
736 270 CONTINUE
737 XT1 = T1/XL
738 T2 = T2/XL
739 T3 = T3/XL
740 IF (KD.EQ.0.) GO TO 280
741 SA(1, 1) = T1*TRACE
742 SA(1, 2) = T2*TRACE
743 SA(1, 3) = T3*TRACE
744 S(1, 1) = T1*TRACE
745 S(1, 2) = T2*TRACE
746 S(1, 3) = T3*TRACE
747 S(2, 2) = T2*TRACE
748 S(2, 3) = T3*TRACE
749 S(3, 3) = T3*TRACE
750 PP = TRACEAPS
751 R(1) = T1APP
752 R(2) = T2APP
753 R(3) = T3APP
754 GO TO 250
755 280 GO 310 L=1, 3
756 R(1) = 0.0
758  SA(I,J)=0.0
759  DO 310 J=1,3
760 310 IF (KR.EQ.0) GO TO 400
761  SA(2,5)=T2ATRACE
762  SA(2,4)=T1ATRACE
763  SA(2,6)=T3ATRACE
764  S(4,4)=T1AT4ATRACE
765  S(4,5)=T1AT2ATRACE
766  S(4,6)=T1AT3ATRACE
767  S(5,5)=T2AT2ATRACE
768  S(5,6)=T2AT3ATRACE
769  S(6,6)=T3AT3ATRACE
770  PP=TRACEASK
771  R(4)=T1APP
772  R(5)=T2APP
773  R(6)=T3APP
774  GO TO 450
775 400  DO 410 I=4,6
776 410  R(I)=0.0
777  SA(2,1)=0.0
778  DO 410 J=1,6
779 410  S(I,J)=0.0
780 450  DO 500 I=1,ND
781 500  DO 500 J=1,ND
782 500  S(I,J)=S(I,J)
783 520  DO 520 I=1,ND
784 520  DO 520 J=1,4
785 520  F(I,J)=R(I)*R(J)
786 530  NN=NIP
787  NNI=NI
788  NNJ=NJ
789  NNK=NK
790  NNL=NL
791  NND=ND
792  NKR=KR
793  SSD=SD
794  SSK=SR
795  TTR=TRACE
796  GO TO 560
797  550  MARK=1
798  555  NN=NN+KG
799  NNI=NNI+KG
800  560  KEL=NE+1
801  WRITE (33,2010) KEL,NN,NNI,NNJ,NNK,NNL,KND,NNK,KND,SSD,SSR,TTK
802  NE=NE+1
803  IF(MODEX.EQ.0)  GOTO 650
804  Write(MDT) NE,NN,NNI,NNJ,NNK,NNL,KND,NNK,SSD,SSR,TTK
805  DO 600 I=1,ND
806  600  LM(I)=ID(NN,1)
807  600  NDM=34
808  CALL CALBAAN(M,BAND,MDF,M,LH,XM,S,P,ND,NDM,NS)
809  IF(MODEX.EQ.1) GOTO 650
810  WRITE (1) ND,NS,LX(L),L=1,ND),((SA(L,K),L=1,NS),K=1,ND),
811 1 ((II(L,K),L=1,NS),K=1,4)
812 650 CONTINUE
813 IF (NE.EQ.NUMEL) RETURN
814 IF (NN.LT.NP) GO TO 555
815 IF (MARK.EQ.1) GO TO 210
816 GO TO 200
817 1000 FORMAT (815,3F10.0)
818 1005 FORMAT (4F10.0)
819 2000 FORMAT (34HBOUNDARY ELEMENTS, //)
820 1 27H ELEMENT TYPE = 7, /
821 2 21H NUMBER OF ELEMENTS =,16  //1X)
822 2005 FORMAT (3OH
823 1 7HCASE(B),8X,7HCASE(C),8X,7HCASE(D),/ 4E15.4  // 1X)
824 2007 FORMAT (53H ELEMENT NODE NODES DEFINING CONSTRAINT DIRECTION, 825 1 3X,38HCODE CODE GENERATION SPECIFIED,6X,
826 2 22HSPECIFIED SPRING, /
827 3 53H NUMBER (N) (NI) (NJ) (NK) (NL),
828 4 3X,38H KB KR CODE (KN) DISPLACEMENT,6X,
829 5 22H ROTATION RATE, / 1X)
831 END
832 SUBROUTINE CROSS2 (A,B,C,IERR)
833 C CALLED BY : INP21
834 IMPLICIT REALA8(A-H,O-Z)
835 X = A(2) A B(3) - A(3) A B(2)
836 Y = A(3) A B(1) - A(1) A B(3)
837 Z = A(1) A B(2) - A(2) A B(1)
838 XLN =DSORT(X*XtY+ZAZ)
839 IERR = 1
840 IF(XLN.LE.1.0E-08) RETURN
841 XLN = 1.0 /XLN
842 C(3) = Z A XLN
843 C(2) = Y A XLN
844 C(1) = X A XLN
845 IERR = 0
846 RETURN
847 END
848 SUBROUTINE DER3DS (NEL,XX,B,DET,R,S,T,NOD9,H,P,IELD,IEL,
849 C CALLED BY : THDEF
850 IMPLICIT REALA8(A-H,O-Z)
851 C
PROGRAM

EVALUATES STRAIN-DISPLACEMENT MATRIX B AT POINT (R,S,T)

CURVILINEAR HEXAHEDRON 8 TO 21 NODES

DIMENSION A(6,1),NOD9(1),H(1),P(3,1)

DIMENSION A(3,3),XJ(3,3)

FIND INTERPOLATION FUNCTION AND THEIR DERIVATIVES

EVALUATE JACOBIAN MATRIX AT POINT (R,S,T)

COMPUTE DETERMINANT OF JACOBIAN MATRIX AT POINT (R,S,T)

CALL ENCT (R,S,T,H,P,NOD9,XJ,DET,XX,IELD,IELX,NEL)

COMPUTE INVERSE OF JACOBIAN MATRIX

DO 130 K=1,IELD

K2=K3

DO 115 L=1,3

B(L,K2-2) = 0.0

B(L,K2-1) = 0.0

115 B(L,K2 ) = 0.0

DO 120 I=1,3

DIRECT STRAINS (1=EIX, 2=EYY, 3=EZZ)

DO 130 K=1,IELD
\[ B(I,K2-2) = B(I,K2-2) + XJI(1,I)A P(I,K) \]
\[ B(2,K2-1) = B(2,K2-1) + XJI(2,I)A P(I,K) \]
\[ B(3,K2) = B(3,K2) + XJI(3,I)A P(I,K) \]

\[ B(4,K2-2) = B(2,K2-1) \]
\[ B(4,K2-1) = B(1,K2-2) \]
\[ B(5,K2-1) = B(3,K2) \]
\[ B(5,K2) = B(2,K2-1) \]
\[ B(6,K2-2) = B(3,K2) \]
\[ B(6,K2) = B(1,K2-2) \]

\[ B(4,K2-2) = B(2,K2-1) \]
\[ B(2,K2-1) = B(1,K2-2) \]
\[ B(3,K2) = B(3,K2) + XJI(3,I)A P(I,K) \]

"SHEAR STRAINS \(4=EXY, 5=EYZ, 6=EZX\)"

\[ B(I,K2-2) = B(I,K2-2) + XJI(1,I)A P(I,K) \]
\[ B(2,K2-1) = B(2,K2-1) + XJI(2,I)A P(I,K) \]
\[ B(3,K2) = B(3,K2) + XJI(3,I)A P(I,K) \]

\[ B(4,K2-2) = B(2,K2-1) \]
\[ B(4,K2-1) = B(1,K2-2) \]
\[ B(5,K2-1) = B(3,K2) \]
\[ B(5,K2) = B(2,K2-1) \]
\[ B(6,K2-2) = B(3,K2) \]
\[ B(6,K2) = B(1,K2-2) \]

\[ B(I,K2-2) = B(I,K2-2) + XJI(1,I)A P(I,K) \]
\[ B(2,K2-1) = B(2,K2-1) + XJI(2,I)A P(I,K) \]
\[ B(3,K2) = B(3,K2) + XJI(3,I)A P(I,K) \]

\[ B(4,K2-2) = B(2,K2-1) \]
\[ B(4,K2-1) = B(1,K2-2) \]
\[ B(5,K2-1) = B(3,K2) \]
\[ B(5,K2) = B(2,K2-1) \]
\[ B(6,K2-2) = B(3,K2) \]
\[ B(6,K2) = B(1,K2-2) \]

B-18
973 GO TO 900
974 C PLATE BENDING ELEMENTS
975 C
976 C 6 CONTINUE
977 C 6 CALL SHELL
978 GO TO 900
979 C
980 C 7 CALL BOUND
981 C GO TO 900
982 C
983 C THICK SHELL ELEMENTS
984 C
985 C 8 CALL SOL21
986 C GO TO 900
987 C
988 C 9 WRITE (33,100) MTYPE
989 C GO TO 900
990 C
991 C 10 WRITE (33,100) MTYPE
992 C
993 C 11 WRITE (33,100) MTYPE
994 C
995 C 12 CALL PIPE
996 C
997 C RETURN
998 C
999 C STRAIGHT OR CURVED PIPE ELEMENTS
1000 C
1001 12 CONTINUE
1002 C 12 CALL PIPE
1003 C
1004 900 RETURN
1005 C
1006 100 FORMAT ('OELEMENT',I4,' IS NOT IMPLEMENTED YET')
1007 C END
1008 C
1009 SUBROUTINE ERROR(N)
1010 WRITE (33,2000) N
1011 2000 FORMAT //20H STORAGE EXCEEDED BY 16)
1012 STOP
1013 C SUBROUTINE FACEPR (NEL,KDIS,KXYZ,XX,NOD9,H,P,PL,NEACE,LT,PWA,KLS)
1014 C
1015 C CALLED BY : THDFE
1016 C CALLS : ENCT
1017 C IMPLICIT REAL*8(A-H,O-Z)
1018 C THIS ROUTINE COMPUTES NODE FORCES DUE TO APPLIED ELEMENT FACE
1019 C PRESSURE DISTRIBUTIONS
1020 C
DIMENSION XX(3,1), NOD9(1), H(1), P(3,1), PL(1), PWA(1)
DIMENSION XJ(3,3), ETA(3), KEACE(6,8), KCRD(6), EVAL(6), IPRM(3),
1 PR(8), NODES(8), IPR4(4)
COMMON /GAUSS/ XK(4,4), WGT(4,4)
C
DATA KEACE / 1, 2, 1, 4, 1, 5,
1 4, 3, 5, 8, 2, 6,
2 8, 7, 6, 7, 3, 7,
3 5, 6, 2, 3, 4, 8,
4 12, 10, 17, 20, 9, 13,
5 20, 19, 13, 15, 10, 14,
6 16, 14, 18, 19, 11, 15,
7 17, 18, 9, 11, 12, 16/
C
DATA KCRD / 1, 1, 2, 2, 3, 3/
DATA EVAL / 1.,-1., 1.,-1., 1.,-1./
DATA IPRM / 2, 3, 1/
DATA IPR4 / 2, 3, 4, 1/
C
DETERMINE THE ELEMENT NODES CONTRIBUTING TO FORCE CALCULATIONS
ON THIS FACE
DO 2 I=1,4
NODES(I) = KFACE(NFACE, I)
NODES(I+4) = 0
2 CONTINUE
IF(KDIS.LT.9) GO TO 9
NN9 = KDIS-8
DO 8 K=5,8
DO 4 I=1,NN9
J = I
IF(KFACE(NFACE,K).EQ.NOD9(I)) GO TO 6
4 CONTINUE
GO TO 8
6 NODES(K) = J
8 CONTINUE
GO TO (10,30), GT
9 CONTINUE
SET UP THE PRESSURE VECTOR FOR THE FOUR FACE CORNER NODES
1. ADJUST THE SIGN OF THE PRESSURES SO THAT POSITIVE
PRESSURE ALWAYS COMpresses THE ELEMENT
FACT = -EVAL(NFACE)
GO TO (10,30), LT
B-20
2. DISTRIBUTED PRESSURE GIVEN AT THE CORNER NODES

1081 C 10 DO 25 K=1,8
1082 C IF(NODES(K).EQ.0) GO TO 25
1083 C IF(K.GT.4) GO TO 15
1084 C PK(K) = PWA(K) * FACT
1085 C GO TO 25
1086 C 25 CONTINUE
1087 C

3. ELEMENT FACE EXPOSED TO HYDROSTATIC PRESSURE

1091 C 15 J = K-4
1092 C L = IPK4(J)
1093 C PK(K) = (PWA(J) + PWA(L)) * 0.5 * FACT
1094 C
1095 C

1096 C 25 CONTINUE
1097 C GO TO 75
1098 C
1099 C

1101 C 30 GAMMA = PWA(1) * FACT
1102 C
1103 C XLN = 0.0
1104 C DO 35 K=1,3
1105 C ETA(K) = (PWA(K+4) - PWA(K+1))
1106 C 35 XLN = XLN + ETA(K)*AA2
1107 C XLN = DSQRT(XLN)
1108 C
1109 C IF(XLN.GT.1.0E-6) GO TO 40
1110 C
1111 C WRITE (33,3000) KLS,NEL
1112 C 3000 FORMAT (31HERROR PRESSURE LOAD SET (,13,15H) FOR ELEMENT (,
1113 C 15,45H) HAS UNDEFINED HYDROSTATIC SURFACE NORMAL, /1X)
1114 C STOP
1115 C
1116 C 40 DO 45 K=1,3
1117 C 45 ETA(K) = ETA(K)/ XLN
1118 C
1119 C DO 70 N=1,8
1120 C
1121 C IF(NODES(N).EQ.0) GO TO 70
1122 C
1123 C XLN = 0.0
1124 C NOD = NODES(N)
1125 C IF(N.LT.4) NOD = NOD + 8
1126 C
1127 C DO 50 I=1,3
1128 C 50 XLN = XLN + (XX(I,NOD) - PWA(I+1))* ETA(I)
1129 C
1130 C PK(N) = XLN * GAMMA
1131 C IF(XLN.LT.0.0) PK(N) = 0.0
1132 C
1133 C 70 CONTINUE

V-21
1135   75 CONTINUE
1136   C SET UP VARIABLES FOR THE SURFACE INTEGRATION
1137   C
1138   ML = KCRD(NFACE)
1139   MM = IPRH(ML)
1140   MN = IPRH(MN)
1141   C SURFACE INTEGRATION LOOP
1142   C ETA(ML) = EVAL(NFACE)
1143   C DO 300 LX = 1, N
1144   C ETA(MM) = AX(LX, 3)
1145   C DO 300 LY = 1, N
1146   C ETA(MN) = BX(LY, 3)
1147   WT = WGT(LX, 3) * WGT(LY, 3)
1148   C EVALUATE THE INTERPOLATION FUNCTIONS AND JACOBIAN MATRIX
1149   C CALL FNCT (ETA(1), ETA(2), ETA(3), M, P, NODE, XJ, DET, XX, KBIS, XX2, NEL)
1150   C COMPUTE THE DIRECTION COSINES OF THE UNIT SURFACE NORMAL VECTOR
1151   C AT THIS SAMPLE POINT
1152   C
1153   A1 = XJ(MM, 2) * XJ(MM, 3) - XJ(MM, 3) * XJ(MM, 2)
1154   A2 = XJ(MM, 3) * XJ(MM, 1) - XJ(MM, 1) * XJ(MM, 3)
1155   A3 = XJ(MM, 1) * XJ(MM, 2) - XJ(MM, 2) * XJ(MM, 1)
1156   AA = DSQRT(A1 * A1 + A2 * A2 + A3 * A3)
1157   IF (AA.GT.1.0E-8) GO TO 100
1158   C WRITE (33, 3010) NFACE, NEL
1159   C 3010 FORMAT (3HERROR AA UNDEFINED NORMAL IN FACE (, I1, 5H) FOR,
1160   C / I1 ELEMENT (, I5, 2H) , / I1)
1161   C COMPUTE THE FIRST FUNDAMENTAL FORM (AREA DIFFERENTIAL)
1162   C
1163   AA = 0.0
1164   BB = 0.0
1165   CC = 0.0
1166   DO 120 I = 1, N
1167       AA = AA + XJ(MM, I) * A1
1168       CC = CC + XJ(MM, I) * A3
1169 120   CONTINUE
1170   FACT = 1.0 / AA
1171   A1 = A1 * FACT
1172   A2 = A2 * FACT
1173   A3 = A3 * FACT
1174   C 100 FORMAT (I1)
1175 100   STOP
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
B-32
1189 120 BB = BB + XJ(MH,1)*XJ(MN,1)
1190 C = DSORT(AAACC - BBAA2)
1191 C
1192 C INTERPOLATE FOR THE PRESSURE AT THIS SAMPLE POINT
1193 C
1194 PRESS = 0.0
1195 C DO 130 K=1,8
1196 C
1197 C IF(NODES(K).EQ.0) GO TO 130
1198 C NOD = NODES(K)
1199 C IF(K.GT.4) NOD = NOD + 6
1200 C PRESS = PRESS + H(NOD)*P(K)
1201 C 130 CONTINUE
1202 C FACT = WTA CA PRESS
1203 C ASSEMBLE THE NODE FORCE CONTRIBUTION
1204 C DO 160 L=1,8
1205 C
1206 C IF(NODES(L).EQ.0) GO TO 160
1207 C IF(L.GT.4) GO TO 140
1208 C 1. CORNER NODES
1209 C
1210 N = NODES(L)
1211 K = 3AM
1212 GO TO 150
1213 C 2. SIDE NODES
1214 C
1215 140 J = NODES(L)
1216 N = J+3
1217 K = 3A NOD9(J)
1218 C 150 GO = FACTA H(N)
1219 C
1220 PL(K-2) = PL(K-2) + UDA A1
1221 PL(K-1) = PL(K-1) + UDA A2
1222 PL(K ) = PL(K ) + UDA A3
1223 160 CONTINUE
1224 C
1225 C 360 CONTINUE
1226 C
1227 RETURN
1228 END
1229 C=================================================================================================
1230 C SUBROUTINE FNLFR (K,UL,T,NP,NOD9,X1,DET,XX,IELD,IELX,IEL)
1231 C CALLED BY : FALCFR
1232 C
IMPLICIT REAL*8(A-H,O-Z)

TO FIND INTERPOLATION FUNCTIONS (H)
AND POINTS OF A CURVILINEAR ISOPARAMETRIC HEXAHEDRON
OR SUBPARAMETRIC HEXAHEDRON (8 TO 21 NODES)
TO FIND JACOBIAN (XI) AND ITS DETERMINANT (DEI)

DIMENSION H(1),P(3,1),NOD9(1),IPERM(8),XI(3,3),XX(3,1)
DATA IPERM / 2,3,4,1,6,7,8,5 /
IELD = IELD
NND9 = IELD-8

RP=1.0 + R
SP=1.0 + S
TP=1.0 + T
RM=1.0 - R
SM=1.0 - S
IM=1.0 - T
RR=1.0 - RAR
SS=1.0 - SAS
TT=1.0 - TAT

INTERPOLATION FUNCTIONS AND THEIR DERIVATIVES
8-NODE BRICK

H(1)=0.125ARPAAP
H(2)=0.125ARMAAPP
H(3)=0.125ARMAAPP
H(4)=0.125APASAATP
H(5)=0.125APASAATP
H(6)=0.125APASAATP
H(7)=0.125APASAATP
H(8)=0.125APASAATP

P(1,1)=0.125APASAATP
P(1,2)=-P(1,1)
P(1,3)=-0.125APASAATP
P(1,4)=-P(1,3)
P(1,5) = 0.125*SPATM
P(1,6) = -P(1,5)
P(1,7) = 0.125*SNMATM
P(1,8) = -P(1,7)
P(2,1) = 0.125*ARFATP
P(2,2) = 0.125*ARMAATP
P(2,3) = -P(2,2)
P(2,4) = -P(2,1)
P(2,5) = 0.125*ARFATM
P(2,6) = 0.125*ARMAATM
P(2,7) = -P(2,6)
P(2,8) = -P(2,5)
P(3,1) = 0.125*ARFATP
P(3,2) = 0.125*ARMAATP
P(3,3) = 0.125*ARMAATM
P(3,4) = 0.125*ARPAATM
P(3,5) = -P(3,1)
P(3,6) = -P(3,2)
P(3,7) = -P(3,3)
P(3,8) = -P(3,4)
       IF (IEL.EQ.8) GO TO 50
ADD DEGREES OF FREEDOM IN EXCESS OF 8
       I=0
       2 I = I + 1
       IE (I.GT.NMW) GO TO 40
       NN = MOD (I, 8) - 8
       GO TO (9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21) , NN
       9 H(9) = 0.25*ARASKPATP
       P(1,9) = -0.50*ARASKPATP
       P(2,9) = 0.25*ARAKATP
       P(3,9) = 0.25*ARASKATP
       GO TO 2
       10 H(10) = 0.25*ARMAASATP
       P(1,10) = -0.25*ARMAASATP
       P(2,10) = -0.50*ARMAASATM
       P(3,10) = 0.25*ARMAAS
       GO TO 2
       11 H(11) = 0.25*ARASKMATP
       P(1,11) = -0.50*ARASKMATP
       P(2,11) = 0.25*ARAKATP
       P(3,11) = 0.25*ARASKATP
       GO TO 2
       12 H(12) = 0.25*ARPAASATP
       P(1,12) = 0.25*ARPAASATP
       P(2,12) = -0.50*ARPAASATM
       P(3,12) = 0.25*ARPAAS
       GO TO 2
1351 13  \( H(13) = 0.25 \times ARPASATM \)  
1352 13  \( P(1,13) = -0.50 \times ARPASATM \)  
1353 13  \( P(2,13) = 0.25 \times ARPASATM \)  
1354 13  \( P(3,13) = -0.25 \times ARPAS \)  
1355 13  GO TO 2  
1356 14  \( H(14) = 0.25 \times ARMASSATM \)  
1357 14  \( P(1,14) = -0.25 \times ARMASSATM \)  
1358 14  \( P(2,14) = -0.50 \times ARMASSATM \)  
1359 14  \( P(3,14) = -0.25 \times ARMASS \)  
1360 14  GO TO 2  
1361 15  \( H(15) = 0.25 \times ARPASMAIM \)  
1362 15  \( P(1,15) = -0.50 \times ARPASMAIM \)  
1363 15  \( P(2,15) = -0.25 \times ARPASATM \)  
1364 15  \( P(3,15) = -0.25 \times ARPAS \)  
1365 15  GO TO 2  
1366 16  \( H(16) = 0.25 \times ARPASATM \)  
1367 16  \( P(1,16) = 0.25 \times ARPASATM \)  
1368 16  \( P(2,16) = -0.50 \times ARPASATM \)  
1369 16  \( P(3,16) = -0.25 \times ARPAS \)  
1370 16  GO TO 2  
1371 17  \( H(17) = 0.25 \times ARPASATM \)  
1372 17  \( P(1,17) = 0.25 \times ARPASATM \)  
1373 17  \( P(2,17) = 0.25 \times ARPASATM \)  
1374 17  \( P(3,17) = -0.50 \times ARPAS \)  
1375 17  GO TO 2  
1376 18  \( H(18) = 0.25 \times ARPASATM \)  
1377 18  \( P(1,18) = -0.25 \times ARPAS \)  
1378 18  \( P(2,18) = 0.25 \times ARPASATM \)  
1379 18  \( P(3,18) = -0.50 \times ARMASS \)  
1380 18  GO TO 2  
1381 19  \( H(19) = 0.25 \times ARMASSATM \)  
1382 19  \( P(1,19) = -0.25 \times ARMASSATM \)  
1383 19  \( P(2,19) = -0.25 \times ARMASS \)  
1384 19  \( P(3,19) = -0.50 \times ARMASSATM \)  
1385 19  GO TO 2  
1386 20  \( H(20) = 0.25 \times ARMASSATM \)  
1387 20  \( P(1,20) = 0.25 \times ARMASSATM \)  
1388 20  \( P(2,20) = -0.50 \times ARMASSATM \)  
1389 20  \( P(3,20) = -0.25 \times ARMASS \)  
1390 20  GO TO 2  
1391 21  \( H(21) = ARPASAT \)  
1392 21  \( P(1,21) = -2.0 \times ARPASAT \)  
1393 21  \( P(2,21) = -2.0 \times ARPASAT \)  
1394 21  \( P(3,21) = -2.0 \times ARPASAT \)  
1395 21  GO TO 2  
1396 C  
1397 C  MODIFI FIRST 8 FUNCTIONS IF 9 OR MORE NODES IN ELEMENT  
1398 C  
1399 40  \( IH = 0 \)  
1400 41  \( IH = IH + 1 \)  
1401 42  IF \( IH.GT.NND9 \) GO TO 50  
1402 43  \( II = IH + 7 \)  
1403 44  IF \( II.EQ.IELX \) GO TO 51  
1404 45  \( IN = NND9(IH) \)
IF (IN.GT.16) GO TO 46
1407 I2=IPERM(I1)
1408 H(I1)=H(I1) - 0.5*AH(IN)
1409 H(I2)=H(I2) - 0.5*AH(IN)
1410 H(IN+8)=H(IN)
1411 DO 45 J=1,3
1412 P(J,I1)=P(J,I1) - 0.5*AP(J,IN)
1413 P(J,I2)=P(J,I2) - 0.5*AP(J,IN)
1414 45 P(J,IN+8)=P(J,IN)
1415 GO TO 41
1416 46 IF (IN.EQ.21) GO TO 50
1417 I1=IN -16
1418 I2=I1 + 4
1419 H(I1)=H(I1) - 0.5*AH(IN)
1420 H(I2)=H(I2) - 0.5*AH(IN)
1421 H(IN+8)=H(IN)
1422 DO 47 J=1,3
1423 P(J,I1)=P(J,I1) - 0.5*AP(J,IN)
1424 P(J,I2)=P(J,I2) - 0.5*AP(J,IN)
1425 47 P(J,IN+8)=P(J,IN)
1426 GO TO 41
1427 C
1428 C MODIFY FIRST 20 FUNCTIONS IF NODE 21 IS PRESENT
1429 C
1430 30 IH=0
1431 31 IH=IH + 1
1432 IN=NO9(IH)
1433 IF (IN.EQ.21) GO TO 35
1434 IF (IN.GT.16) GO TO 33
1435 I1=IN -8
1436 I2=IPERM(I1)
1437 H(I1)=H(I1) + 0.125*AH(21)
1438 H(I2)=H(I2) + 0.125*AH(21)
1439 DO 32 J=1,3
1440 P(J,I1)=P(J,I1) + 0.125*AP(J,21)
1441 32 P(J,I2)=P(J,I2) + 0.125*AP(J,21)
1442 GO TO 31
1443 33 I1=IN -16
1444 I2=I1 + 4
1445 H(I1)=H(I1) + 0.125*AH(21)
1446 H(I2)=H(I2) + 0.125*AH(21)
1447 DO 34 J=1,3
1448 P(J,I1)=P(J,I1) + 0.125*AP(J,21)
1449 34 P(J,I2)=P(J,I2) + 0.125*AP(J,21)
1450 GO TO 31
1451 35 DO 36 I=1,3
1452 H(IN)=H(IN) - 0.125*AH(21)
1453 DO 36 J=1,3
1454 36 P(J,I)=P(J,I) - 0.125*AP(J,21)
1455 HH=HH+1
1456 IF (HH.EQ.8) GO TO 50
1457 DO 38 I=9,HH
1458 H(IN)=H(IN) - 0.125*AH(21)
DO 38 J=1,3
38 P(J,I)=P(J,I) - 0.25AP(J,21)
H(NND9+8)=H(21)
DO 39 J =1,3
39 P(J,NND9+B)=P(J,21)
C EVALUATE JACOBIAN MATRIX AT POINT (R,S,T)
C
50 IF (IELX.LT.IELD) RETURN
51 DO 100 I=1,3
52 DUM=0.0
53 DO 90 K=1,IELX
90 DUM=DUM + P(I,K)*XX(J,K)
100 XJ(I,J)=DUM
C COMPUTE DETERMINANT OF JACOBIAN MATRIX AT POINT (R,S,T)
C
DET = XJ(I,1)XX(J,2)XX(J,3)
1 + XJ(I,2)XX(J,2)XX(J,3)
2 + XJ(I,3)XX(J,2)XX(J,3)
3 - XJ(I,3)XX(J,2)XX(J,3)
4 - XJ(I,1)XX(J,2)XX(J,3)
5 - XJ(I,1)XX(J,2)XX(J,3)
IF(DET.GT.1.0E-8) GO TO 110
WRITE (33,2000) NEL,R,S,T
STOP
110 IF (IELX.LT.IELD) GO TO 42
RETURN
C
C SUBROUTINE INL(ID,B,Tk,TMASS,NUMNP,NEQB,LL)
C
2000 FORMAT (49HOERRKKKAKK...NEGATIVE OR ZERO JACOBIAN DETERMINANT,
1 ISH COMPUTED FOR ELEMENT (,IS,1H), /
1 2KX, 3HR =, F10.5 /
3 12X, 3HS =, F10.5 /
4 12X, 3HI =, F10.5 / 1X)
END
C

DIMENSION ID(NUMNP,6),B(NEQB,LL),TR(6,LL),IMASS(NEQB)

COMMON /JUNK/ R(6),TXM(6),IFILL(406)

COMMON /EXTRA/ MODEX,NIB,IFILL2(14)

C

NI=3

REWIND NI

KSHE=0

WRITE (33,2002)

IF(MODEX.EQ.1) GO TO 300

DO 750 I=1,NEQB

IMASS(I)=0.

DO 750 K=1,LL

750 B(I,K)=0.0

C

50 DO 900 NH=1,NUMNP

C

DO 100 I=1,6

TXM(I)=0.

DO 100 J=1,LL

100 TR(I,J)=0.0

C

IF(HN.EQ.1) GO TO 300

150 IF(HN.EQ.HN) GO TO 400

DO 200 I=1,6

IF (L) 180,180,190

180 TXM(I)=R(I)

GO TO 200

190 TR(I,L)=R(I)

200 CONTINUE

300 READ (5,10V1) N,L,K

IF (N.EQ.0) GO TO 150

WRITE (33,2001) N,L,K

GO TO 150

C

400 IF(MODEX.EQ.1) GO TO 900

450 DO 600 J=1,6

II=ID(HN,J)-KSHE

500 IF (II) 800,800,500

500 DO 600 K=1,LL

600 R(K,J,K)=TR(J,K)

IMASS(II)=1*IMASS(J)

610 IF(HN.EQ.NEQB) GO TO 800

650 WRITE (33,2001) N,TMARR

550 KSHE=KSHE+NEQB

550 DO 700 I=1,NEQB

550 TMASS(I)=v.

600 DO 700 K=1,LL

700 B(I,K)=0.0

800 CONTINUE

900 CONTINUE

C

IF(MODEX.EQ.1) RETURN

C
1567  WRITE (NT) B,IMASS
1568  C
1569  RETURN
1570  1001 FORMAT (215,7E10.4)
1571  2001 FORMAT (2(3X,I4),6E15.S)
1572  2002 FORMAT (47HASSIGNED DYNAMIC LOADING (STATICAL OR),
1573  A 29HAMASSES (DYNAMIC),///
1574  B 3X,4HNODE,3X,4HLOAD,
1575  1 2(9X,6HAXIS,9X,6HAXIS,9X,6HAXIS), /7H NUMBER,3X,THENCE,
1576  2 3(10X,5HEQCE), 3(9X,6HMOMENT), /1X)
1577  END
1578 C&===================================================================
1579  SUBROUTINE INPUTJ(ID,X,'i,Z,T,NUMNP,NEO)
1580  C
1581  IMPLICIT REAL(A-H,O-Z)
1582  C
1583  C CALLED BY: MAIN
1584  C
1585  DIMENSION X(1),Z(1),ID(NUMNP,6),T(1)
1586  C
1587  COMMON /EXTRA/ MODEX,NT8,IFILL(14)
1588  C
1589  C---- SPECIAL NODE CARD FLAGS
1590  C
1591  C IT = COORDINATE SYSTEM TYPE (CC 1, ANY NODE CARD)
1592  C EQ.C, CYLINDRICAL
1593  C IPR = PRINT SUPPRESSION FLAG (CC 6, CARD FOR NODE 1 ONLY)
1594  C EQ.D, NORMAL PRINTING
1595  C EQ.A, SUPPRESS SECOND PRINTING OF NODAL ARRAY DATA
1596  C EQ.B, SUPPRESS PRINTING OF ID-ARRAY
1597  C EQ.C, BOTH AAA AND ABA
1598  C
1599  DIMENSION IPRC(4)
1600  C
1601  DATA IPRC/"IH ,IHA,IHB,IHC/
1602  C
1603  C IPR = IPRC(I)
1604  RAD = DATAN(1.E0)/45.000
1605  C
1606  C
1607  C---- READ OR GENERATE NODAL POINT DATA---------------------------
1608  WRITE (33,2000)
1609  WRITE (33,2001)
1610  MOLD=0
1611  10 READ (5,1000) IT,N,IRH,(ID(N,I),I=1,6),X(N),Y(N),Z(N),HR,I(N)
1612  C
1613  C
1614  C** ** NEXT LINE IS DELETED FOR NOT PRINTING NODAL INPUT DATA
1615  WRITE (33,2002) IT,N,IRH,(ID(N,I),I=1,6),X(N),Y(N),Z(N),IN,T(N)
1616  C** **
1617  C
1618  IF(N.EQ.1) IPR = JPR
1619  IF(IT.NE.IPRC(4)) GO TO 15
1620  DUM = Z(N)*RA

6-30
Z(N) = X(N)cos(θ) - Y(N)sin(θ)

X(N) = X(N)cos(θ) + Y(N)sin(θ)

1623 CONTINUE
1624 IF(NOLD.EQ.0) GO TO 50
1625 C----- CHECK IF GENERATION IS REQUIRED---------------------
1626 DO 20 I=1,6
1627 IF(ID(K,I).EQ.0.AND.ID(NOLD,I).LT.0) ID(N,I)=ID(NOLD,I)
1628 CONTINUE
1629 IF(KM.EQ.0) GO TO 50
1630 NUM=(N-NOLD)/KM
1631 XNUM=NUM
1632 IF(XNUM.LT.1) GO TO 50
1633 DX=(X(N)-X(NOLD))/XNUM
1634 DY=(Y(N)-Y(NOLD))/XNUM
1635 DZ=(Z(N)-Z(NOLD))/XNUM
1636 DT=(T(N)-T(NOLD))/XNUM
1637 K=NOLD
1638 DO 30 J=1,XNUM
1639 K=K+KN
1640 X(K)=X(KK)+DX
1641 Y(K)=Y(KK)+DY
1642 Z(K)=Z(KK)+DZ
1643 T(K)=T(KK)+DT
1644 DO 30 I=1,6
1645 ID(K,I)=ID(KK,I)
1646 IF (ID(K,I).LT.1) ID(K,I)=ID(KK,I)+KN
1647 CONTINUE
1650 C
1651 50 NOLD=N
1652 IF(N.NE.NUMNP) GO TO 10
1653 C----- PRINT ALL NODE POINT DATA---------------------
1654 C
1655 C
1656 IF(IFK.EQ.IPRC(2).OR.IFK.EQ.IPRC(4)) GO TO 52
1657 WRITE (33,2003)
1658 WRITE (33,2001)
1659 WRITE (33,2005) (N, ID(N,I), I=1,6),X(N),Y(N),Z(N),T(N),N=1,NUMNP
1660 CONTINUE
1661 C
1662 C----- NUMBER UNKNOWNS AND SET MASTER NODES NEGATIVE-------------
1663 C
1664 NEQ=0
1665 DO 60 N=1,NUMNP
1666 ID(N,I)=ABS(ID(N,I))
1667 NEQ=NEQ+1
1668 IF(ID(N,I)=1) 57,58,59
1669 CONTINUE
1670 ID(N,I)=NEQ
1671 GO TO 60
1672 58 ID(N,I)=0
1673 GO TO 60
1674 59 ID(N,I)=-ID(N,I)
GO CONTINUE
CONTINUE
CONTINUE
IF(MODEX.EQ.0) GO TO 70
WRITE (NT8) ((ID(N,I),I=I,6),N=1,NUNNP)
WRITE (NT8) (X(N),N=1,NUMNP)
WRITE (NT8) (Y(N),N=1,NUMNP)
WRITE (NT8) (Z(N),N=1,NUMNP)
WRITE (NT8) (T(N),N=1,NUMNP)
ENDFILE NT
REWRITE (2) ID
RETURN
CONTINUE
REWRITE (8) ID
RETURN
1000 FORMAT (2(AI4),5I15,3F10.0,IS,F10.0)
2000 FORMAT (//23H NODAL POINT INPUT DATA )
2001 FORMAT (5HONODE 3X 24HBOUNDARY CONDITION CODES
2002 FORMAT (1X,AI,14,AI,I3,5I15,3F13.3,I5,F13.3)
2003 FORMAT (//21H GENERATED NODAL DATA)
2004 FORMAT (15,6I15,4F13.3)
END
SUBROUTINE INP21 (NUMMATMAXTP,NORTHO,NDLS,NOPSET,NT8SV,NUMNP,X,
Y,Z,DEN,RHO,NTP,EE,DCA,NFACE,LT,PWA,LOC,MAXPTS)
CALLED BY : THDFE
CALLS : VECTR2,CROSS
IMPLICIT REAL*8(A-H,O-Z)
1-32
THIS ROUTINE READS AND PRINTS ALL 21-NODE SOLID ELEMENT DATA BETWEEN THE CONTROL CARD AND THE ELEMENT DATA CARDS

COMMON /JUNK/ XLF(4),YLE(4),ZLF(4),YLE(4),PLE(4),FILL1(22),VEC
COMMON /EXTRA/ MODEx,NIG

DIMENSION X(1),E(1),Z(1),DEN(1),RHO(1),NTP(1),EE(MAXTP,13,1),
    1 DCA(3,3,1),N2ACE(1),LT(1),PWA(7,1),LOC(7,1),
    2 MAXPTS(1),
DIMENSION HED(6)

READ AND PRINT OF MATERIAL PROPERTIES

READ (33,3000)

DO 10 I=1,NUMMAT

READ (5,1001) n,NTP(I),DEN(I),RHO(I),HED(N),N=I,6)

SET DEFAULT VALUES IF REQUIRED AND CHECK FOR INPUT ERRORS

IF(RHO(I).EQ.0.0) RHO(I) = DEN(I) / 386.4
IF(NTP(I).EQ.0.0) NTP(I) = 1

WRITE (33,3002) n,NTP(I),DEN(I),RHO(I),HED(N),N=I,6)

IF(I.EQ.M) GO TO 2
WRITE (33,4001)
STOP
2 IF(NTP(M).LE.MAXTP) GO TO 4
WRITE (33,4002) MAXTP
STOP
4 NT = NTP(M)

READ PROPERTIES FOR EACH TEMPERATURE

DO 6 K=1,NT
READ (5,1002) (EE(K,L,M),L=1,13)
WRITE (33,3003) (EE(K,L,M),L=1,13)
6 CONTINUE

TEMPERATURE CARDS MUST BE ASCENDING ORDER

IF(NT.EQ.1) GO TO 10
10 IF(I.EQ.1) .GT. EE(1,1) GO TO 8
WRITE (33,4003)
STOP
8 CONTINUE
10 CONTINUE

DATA PORTHOLE SAVE

IF(NTSV.EQ.0) GO TO 12
DO 11 M=1,NUMMAT
WRITE (NT8) M,NTP(M),DEN(M),RH0(M)
NT = NTP(M)
WRITE (NT8) ((EE(K,L,M),L=1,13),K=1,NT)
CONTINUE
CAAA
C
MATERIAL AXIS ORIENTATION SETS
CAAA
12 IF(NORTHO.EQ.0) GO TO 21
CAAA
WRITE (33,3004)
CAAA
11 CONTINUE
CAAA
13 IF(NORTSO.EQ.1) WRITE (NT8) N,NI,NJ,NK
CAAA
CHECK FOR ADMISSIBILITY OF DATA
CAAA
IE(N.EQ.M) GO TO 13
CAAA
WRITE (33,4004)
STOP
CAAA
13 IE(NI.GT.0 .AND. NI.LE.NUHNP) GO TO 5015
CAAA
L = NI
CAAA
5014 WRITE (33,4005) L
STOP
CAAA
5015 IF(NJ.GT.0 .AND. NJ.LE.NUHNP) GO TO 5016
CAAA
L = NJ
CAAA
GO TO 5014
CAAA
5016 IF(NK.GT.0 .AND. NK.LE.NUHNP) GO TO 14
CAAA
L = NK
CAAA
GO TO 5014
CAAA
14 CONTINUE
CAAA
GENERATE DIRECTION COSINE ARRAY FOR THIS DATA SET
CAAA
CALL VECTR2 (DCA(1,1,M),X(NI),Y(NI),Z(NI),X(NJ),Y(NJ),Z(NJ),IERR)
CAAA
IF(IERR.EQ.0) GO TO 16
CAAA
WRITE (33,4006)
STOP
CAAA
16 CALL VECTR2 (V2,NI),Y(NI),Z(NI),X(NK),Y(NK),Z(NK),IERR)
CAAA
IF(IERR.EQ.0) GO TO 17
CAAA
WRITE (33,4007)
STOP
CAAA
17 CALL CROSS2 (DCA(1,1,M),V2,DCA(1,3,M),IERR)
CAAA
IF(IERR.EQ.0) GO TO 18
CAAA
WRITE (33,4008)
STOP
CAAA
18 CALL CROSS2 (DCA(1,3,M),DCA(1,1,M),DCA(1,2,M),IERR)
1837 IF(IERK.EQ.0) GO TO 20
1838 WRITE (23,4009)
1839 STOP
1840 20 CONTINUE
1841 C READ AND PRINT DISTRIBUTED SURFACE LOAD DATA
1842 C 21 IF(NDLS.EQ.0) GO TO 31
1843 C 22 IF(NFACE(m).EQ.0) GO TO 23
1844 C 23 IF(NFACE(m).NE.1 .AND. NFACE(m).LE.6) GO TO 23
1845 C 24 IF(LT(h).EQ.0) GO TO 26
1846 C 25 IF(PWA(i,m).EQ.0) GO TO 26
1847 C 26 CONTINUE
1848 C READ AND PRINT BE STRESS OUTPUT REQUEST LOCATION SETS
1849 C 31 IF(NOPSET.EQ.0) GO TO 49
1850 C 32 STOP
READ (5,1006) (LOC(I,m),I=1,7)
WRITE (33,3011) m,(LOC(I,m),I=1,7)
WRITE (34,3011) m,(LOC(I,m),I=1,7)
C
L = 0
DO 35 J=1,7
IF(LOC(J,m).EQ.0) GO TO 36
L = L + 1
IF(LOC(J,m).GE.1.AND. LOC(J,m).LE.27) GO TO 35
WRITE (33,2012)
L = 1
LOC(I,m) = 21
MAXPT$(m) = L
CONTINUE
C
40 CONTINUE
C DATA PORTHOLE SAVE
IF(NTHP.EQ.1)
WRITE (NT9,((LOC(I,J),I=1,7),J=1,NTHP))
C
C ELEMENT LOAD CASE MULTIPLIERS
C
READ (5,1007) XLE,YLF,ZLF,TLF,PLF
WRITE (33,3013) XLF,YLF,ZLF,TLF,PLF
DATA PORTHOLE SAVE
IF(NTHP.EQ.1)
WRITE (NT9,((LOC(I,J),I=1,7),J=1,NTHP))
C
RETURN
C FORMATS
1001 FORMAT (I2,1S,1J,4.E12)
1002 FORMAT (2F10.4,2E10.4)
1003 FORMAT (4(F10.4))
1004 FORMAT (2F10.4)
1005 FORMAT (2F10.4)
1006 FORMAT (1F10.4)
C FORMATS
3000 FORMAT (2,F30.12) Material Property Tables
3001 FORMAT (F10.6,1H6,1H6,1H6)
3002 FORMAT (F10.6,1H6,1H6,1H6)
C 1.14 1.15 1.16 1.17 1.18 1.19
C
23H IDENTIFICATION = (,GA6,1H),../

1 4X,11HTEMPERATURE,9X,3HE11,9X,3HE22,9X,3HE33,4X,3HV12,4X,3HV13,

2 4X,3HV23,6X,3HV12,6X,3HV13,6X,3HV33,3X,7HALPHA-1,3X,7HALPHA-2,

3 8X,7HALPHA-3,1X)

3003 FORMAT (F12.2,3F12.1,3F7.3,3F11.1,3E10.3)

3004 FORMAT (\"SOM MATERIAL AXIS ORIENTATION \",

1 3X,3HTABLE ,../

2 28H SET NODE MODE NODE /

3 23H NUMBER NI NJ NK,/,1X)

3005 FORMAT (417)

3006 FORMAT (\"SOM DISTRIBUTED SURFACE LOAD \",

1 11HI TABLE ,/,1X)

3007 FORMAT (F7.2,27HLOAD SET NUMBER =,1E ,

1 7X,27LOAD SURFACE ELEMENT FACE =,16 ,

1 7X,27LOAD TYPE CODE =,16/1X)

3008 FORMAT (12H DISTRIBUTED, 11X,4HP(1),11X,4HP(2),11X,4HP(3),11X,

1 4HP(4), /4X,6MPRESSURE,4F15.3)

3009 FORMAT (12H HYDROSTATIC,10X,5HGLAMMA,11X,4HX(S),11X,4HY(C),11X,

1 4HZ(S),11X,4HX(N),11X,4HY(N),11X,4HZ(N), /

1 4X,6MPRESSURE,7F15.3)

3010 FORMAT (\"SOM STRESS OUTPUT REQUEST TABLE \",

1 //

1 A6H SET,7(2X,5HPOLNT), /8H NUMBER ,7(4X,1IH(11,1H)1),1X)

3011 FORMAT (10,71")

3012 FORMAT (\"SOM ELEMENT LOAD CASE \",

1 21HMULTIPLIERS ,//

1 A3I6HCASE A,4X,6HCASE B,4X,6HCASE C,

3013 FORMAT :

1 1 27X-DIRECTION GRAYVTY = ,4F10.2/

1 2 27Y-DIRECTION GRAYVTY = ,4F10.2/

1 3 27Z-DIRECTION GRAYVTY = ,4F10.2/

1 4 27THERMAL LOADING = ,4F10.2/

1 5 5 27PRESSURE LOADING = ,4F10.2 //1X)

3014 C

4001 FORMAT (\"SOM MATERIAL CARDS OUT OF ORDER.,/1X)

4002 FORMAT (\"SOM NUMBER OF TEMPERATURE CARDS EXCEEDS USER,

1 1 10H MAXIMUM (,14,2H) ,/,1X)

4003 FORMAT (\"SOM TEMPERATURES MUST BE INPUT IN ASCENDING ,

1 1 7H ORDER ,/,1X)

4004 FORMAT (\"SOM AXIS ORIENTATION CARD OUT OF ORDER.,/1X)

4005 FORMAT (\"SOM UNDEFINED NODE NUMBER =,15 ,/1X)

4006 FORMAT (\"SOM VECTOR II HAS ZERO LENGTH.,/1X)

4007 FORMAT (\"SOM VECTOR IK HAS ZERO LENGTH.,/1X)

4008 FORMAT (\"SOM II AND IK VECTORS ARE PARALLEL.,/1X)

4009 FORMAT (\"SOM E3 AND EI VECTORS ARE PARALLEL.,/1X)

4010 FORMAT (\"SOM SET NUMBERS MUST BE IN ASCENDING ORDER.,/1X)

4011 FORMAT (\"SOM INVALID SURFACE FACE NUMBER.,/1X)

4012 FORMAT (\"SOM INVALID LOAD TYPE.,/1X)

4013 FORMAT (\"SOM INVALID OUTPUT POINT NUMBER =,15,1X)

1995 C

1996 C

1997 END

1998 C=
SUBROUTINE PRINTD (ID,D,B,NEQB,NUMNP,LL,NBLOCK,NEQ,NT,MO)
IMPLICIT REALA8(A-H,O-Z)

CALLED BY: SOLEQ,SOLEIG,RESPEC

DIMENSION ID(NUmrN,6),B(NEQB,LL),D(G,LL)

DATA 011,021,012,022,013,023/' LOAD', 'CASE', 'EIGEN-', 'VECTOR', 'MODE', 'NUMBER'/

REWIND &
READ (8) ID
M=NEQ
NN=NEQB*NBLUCL

IF (MO.EQ.1) GO TO 50
IF (MO.EQ.3) GO TO 55
REWIND NT
Q1=Q11
Q2=Q21
GO TO 60

50 Q1=Q12
Q2=Q22
GO TO 60

55 Q1=Q13
Q2=Q23
REWIND NT
READ (NT)
GO continue

WRITE (33,2003) Q1,Q2 'removed as there is a print in SOLEQ

N=NUMNP
rewind nt !AAAAAAAAAAAAAA

DO 500 KK=1,NUMNP

I=6
DO 250 I=1,6
DO 100 L=1,LL
100 D(I,L)=0.
IF(M.GT.NN) GO TO 150
IF (M.EQ.6) GO TO 150
READ (NT) B
NN=NN-NEQB
150 IF (ID(N,I).LT.1) GO TO 250
K=M-NN
M=M-1

DO 200 L=1,LL
200 D(I,L)=B(K,L)

I=I-1

WRITE (33,2004) N,L,(D(I,L),I=1,6),L=1,LL
N=N-1
C RETURN

C 2003 FORMAT(1H1,38H NODE DISPLACEMENTS / ,
2004 1 17H ROTATIONS / / 3X,4HNOBE,2X,AG,2(12X,2X-,i-),
2005 2 20H NUMBER,2X,AG,3(3X,1HTRANSLATION),
2006 3 3(6X,SHTRAATION), / 1X)
2007 2004 FORMAT(16,18,CE14.5,/: (7X,18,6E14.5))
2008 C-- 2004 FORMAT(140,16,16,6E14.5 / (7X,18,6E14.5))
2009 C END

C --------------------------------------------------------------
SUBROUTINE SOLI
C CALLED BY : ELTYPE
C CALLS : STSSC
C IMPLICIT REAL(A-H,O-Z)
C 3 / D 8 TO 21 NODE SOLID ELEMENTS
C COMMON /ELTY/: HPAR(14),NUMNP,MBAND,NELTYP,N1,N2,N3,N4,NS,NT11,NEO
C COMMON /EM/: NS,NB,LM(63)
C COMMON/JUNK/: LT,LH,LN,B,SI,4(2),N7,N8,N9,N10,N11,N12,N13,N14,
C 1 N5,N6,N7
C COMMON /EXTRA/: MODEX,NS,NIOSV,NI10
C COMMON A(1)
C IF(NPAR(1).EQ.0) GO TO 500
C ERROR CHECKS AND SET DEFAULT VALUES IF REQUIRED
C WRITE (33,1000)
C IF(NPAR(2).GT.0) GO TO 10
C WRITE (33,1001) (NPAR(I),K=1,10)
C WRITE (33,1002)
C STOP
C 10 IF(NPAR(3).GT.0) GO TO 20
C WRITE (33,1001) (NPAR(I),K=1,10)
C WRITE (33,1003)
C STOP
C 20 IF(NPAR(4).EQ.0) NPAR(4) = 1
C IF(NPAR(7).EQ.0) NPAR(7) = 7
C IF(NPAR(7).LE.8) AND. NPAR(7).LE.21) GO TO 30
C WRITE (33,1001) (NPAR(I),K=1,10)
C WRITE (33,1004)
C STOP
C 30 IF(NPAR(9).EQ.0) NPAR(9) = 2
C IF(NPAR(9).LE.2) AND. NPAR(9).LE.4) GO TO 40
C WRITE (33,1001) (NPAR(I),K=1,10)
C WRITE (33,1005)
C STOP
STORAGE ALLOCATION

A(N6) = STARTING LOCATION OF WEIGHT DENSITY
A(N7) = STARTING LOCATION OF MASS DENSITY
A(N8) = STARTING LOCATION OF VECTOR CONTAINING THE ACTUAL
NUMBER OF TEMPERATURE POINTS FOR EACH MATERIAL TABLE
A(N9) = STARTING LOCATION OF MATERIAL PROPERTY TABLE
A(N10) = STARTING LOCATION OF DIRECTION COSINE ARRAYS FOR
MATERIAL ORIENTATION AXIS
A(N11) = STARTING LOCATION OF SURFACE LOAD FACE NUMBERS
A(N12) = STARTING LOCATION OF SURFACE LOAD CODE TYPES
A(N13) = STARTING LOCATION OF PRESSURE WORKING ARRAY
A(N14) = STARTING LOCATION OF OUTPUT REQUEST LOCATION SETS
A(N15) = STARTING LOCATION OF VECTOR CONTAINING THE ACTUAL
NUMBER OF REQUESTED OUTPUT LOCATION IN EACH OUTPUT SET
A(N16) = STARTING LOCATION OF ELEMENT STIFFNESS MATRIX

50 NG = N5 + NUMNP
N7 = N6 + NPAR(3)
N8 = N7 + NPAR(3)
N9 = N8 + NPAR(3)
M10 = N9 + NPAR(3) * NPAR(4) * 13
N11 = N10 + NPAR(5) * 9
N12 = N11 + NPAR(6)
N13 = N12 + NPAR(6)
N14 = N13 + NPAR(6) * 7
N15 = N14 + NPAR(6) * 7
N16 = N15 + NPAR(8)
N17 = N16 + NPAR(7) * 189
N18 = N17 + NPAR(7)

IF(N17.GT.NTOT) CALL ERROR(N17-NTOT)

PROCESS ELEMENT DATA, AND GENERATE ELEMENT MATRICES

CALL THDFE (A(N1), A(N2), A(N3), A(N4), A(N5), A(N6), A(N7), A(N8), A(N9),
A(N10), A(N11), A(N12), A(N13), A(N14), A(N15), A(N16),
NPAR(3), NPAR(3), NPAR(4), NPAR(5), NPAR(6), NPAR(7),
NPAR(8), NPAR(9), NPAR(10), NUMNP)

RETURN

RECOVER ELEMENT STRESSES (STATIC CASES ONLY)

500 WRITE (34,3001)
NUME = NPAR(2)
read (5,4) n11, num
501 format(315)
if(n11.le.0) n11=1
if(nnu.le.0) nnu=nnu
DO 800 nM=1,NUM
C
C
C
C
C
C
C
CALL STSRC (A(N1),A(N3),REO,0)
IF(N10SV.EQ.1)
WRITE (NTIO) NS
DO 800 nM=1,NUM
C
C
C
CALL STSRC (A(N1),A(N3),REO,1)
LOC = NS/6
K1 = -5
DO 600 nM=1,LOC
K1 = K1 + 6
K2 = K1 + 5
IF(um.ge.nl1.and.um.le.nv) then
IF(N.EQ.1) WRITE (34,3001) mm,L,N,(SIG(I),I=K1,K2)
IF(N.GT.1) WRITE (34,4001) mm,(SIG(I),I=K1,K2)
end if
C
C
C
CALL STSRC (A(N1),A(N3),REO,1)
WRITE (NTIO) NS
DO 800 nM=1,LOC
K1 = K1 + 6
C
DO 600 nM=1,LOC
K1 = K1 + 6
K2 = K1 + 5
IF(um.ge.nl1.and.um.le.nv) then
IF(N.EQ.1) WRITE (34,3001) mm,L,N,(SIG(I),I=K1,K2)
IF(N.GT.1) WRITE (34,4001) mm,(SIG(I),I=K1,K2)
end if
C
C
C
CALL STSRC (A(N1),A(N3),REO,1)
WRITE (NTIO) NS
C
WRITE (34,5000)
C
CONTINUE
C
WRITE (34,5000.)
C
RETURN

FORMATS

1000 FORMAT (53H121 - NODE SOLID ELEMENT INPUT )
1001 FORMAT (43HERROR DETECTED WHILE PROCESSING MASTER ELEMENT )
1002 FORMAT (33H NO MATERIALS REQUESTED /1X)
1003 FORMAT (33H NO 1.D SOLID ELEMENTS SPECIFIED ,/1X)
1004 FORMAT (49H MAXIMUM NUMBER OF NODES MUST BE GE.8 .AND. LE.21 ,/1X)
1005 FORMAT (53H121 - NODE SOLID ELEMENT STRES ")
SUBROUTINE SSLAW (A, \(\varepsilon\), TEMP, DCA, KAXES, KMAT, NEL, DUM, ALPHA)

CALLED BY: IHREE

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

THIS ROUTINE FORMS THE STRESS-STRAIN LAW IN MATERIAL COORDINATES
\((X_1, X_2, X_3)\) AND TRANSFORMS THE MATERIAL SYSTEM LAW TO
GLOBAL COORDINATES \((X, Y, Z)\).

DIMENSION D(6,6), E(12), TEMP(6,6), DCA(3,3), IPRM(3), DUM(6,6),
1 ALPHA(G)

DATA IPRM / 2,3,1 /

FORM THE DIRECT STRAIN PARTITION OF THE STRAIN-STRESS LAW IN
MATERIAL COORDINATES \((X_1, X_2, X_3)\)

DO 20 I=1,3
  ALPHA(I) = E(I+9)
  ALPHA(I+3) = 0.0
  IF(E(I).GT.1.0E-08) GO TO 15
  WRITE (33,3000) I,E(I)
  STOP

DO 60 N=1,3
  X = 1.0/TEMP(N,N)
  DO 30 J=1,3
  TEMP(N,J) = TEMP(J,N)*X
  CONTINUE

TEMP(1,2) = -E(4)*TEMP(2,2)
TEMP(2,1) = TEMP(1,2)
TEMP(1,3) = -E(5)*TEMP(3,3)
TEMP(3,1) = TEMP(1,3)
TEMP(2,3) = -E(6)*TEMP(3,3)
TEMP(3,2) = TEMP(2,3)

INVERT THE DIRECT STRAIN PARTITION

DO 60 N=1,3
  DO 30 J=1,3
  TEMP(N,J) = TEMP(N,J)/X
  CONTINUE

DO 15 I=1,3
  IF(N.EQ.1) GO TO 50

STOP
DU 40 J=1,3
IF(N.EQ.J) GO TO 40
TEMP(I,J) = TEMP(I,J) + TEMP(I,N) * TEMP(N,J)
CONTINUE
TEMP(I,N) = TEMP(I,N) * X
TEMP(N,N) = X
CONTINUE

FORM THE COMPLETE STRESS-STRAIN LAW IN MATERIAL COORDINATES
DO 70 I=1,6
DO 70 J=1,6
70 D(I,J) = 0.0
DO 80 I=1,3
DO 80 J=1,3
80 D(I,J) = TEMP(I,J)
DO 82 64,4 = E(1)
(6,5) = E(2)
D(6,6) = E(3)
TRANSFORM THE MATERIAL LAW TO GLOBAL COORDINATES (X,Y,Z)
IF(AXIS.LT.1) RETURN
TRANSFORMATION BETWEEN MATERIAL STRAINS AND GLOBAL STRAINS
DO 100 I1=1,3
I2 = IPRM(I1)
DO 90 J1 = 1,3
J2 = IPRM(J1)
TEMP(I1,J1) = DCA(J1, I1) x DCA(J1, J1)
TEMP(I1+1,J1) = DCA(J1, I1) x DCA(J1, J1) + 2.0
TEMP(I1,J1+3) = DCA(J1, I1) x DCA(J1, J1)
TEMP(I1+3,J1+3) = DCA(J1, I1) x DCA(J1, J1) +
1
DCA(J1, I1) x DCA(J1, J1)
90 CONTINUE
100 CONTINUE

ROTATE THE MATERIAL LAW TO THE GLOBAL SYSTEM
DO 110 I=1,6
DO 110 J=1,6
X = 0.0
DO 116 K=1,6
110 X = X + D(I,K) x TEMP(I,J)
DUM(I,J) = X
CONTINUE
DO 130 I=1,6
DO 130 J=1,6
X = 0.0
DO 140 K=1,6
140 X = X + TEMP(K,I)ADUM(K,J)
D(I,J) = X
D(J,I) = X
150 CONTINUE
160 CONTINUE

TRANSFORM THE EXPANSION COEFFICIENTS FROM MATERIAL COORDINATES
TO GLOBAL COORDINATES

DO 200 I=1,6
X = 0.0
DO 190 K=1,3
190 X = X + TEMP(K,I)AE(K+I)
IF(I.GT.3) X = X*2.0
200 ALPHA(I) = X
RETURN
END

SUBROUTINE STRESS(STR,B,D,NEQ,B,LB,LL,NEQ,NBLOCK)
IMPLICIT REAL*8(A-H,O-Z)
CALLS: ELTYPE
CALLED BY: SOLEG
DIMENSION B(NEQ,B),B(NEQ,B,LL),STR(4,LL)
COMMON /EPTOK/ NPAR(14),NOMNP,MBAND,NELTYP,N1,N2,N3,N4,N5,N10T,NEQ
COMMON /JUNK/ LT,LH,IFILL(428)
COMMON /EXTRA/ MODEX,NTS,NIOSV,NT10,IFILL2(12)
READ (8) SIR
NT=(LL-1)+LB
LH=0
C STRESS PORTHOLE
IF(NIOSV.EQ.1) AWRITE (NT10) NELTYP,NT
DO 1000 II=1,NT
LT = LH + I
LLT=LT-LI
LH=LT+LB-1
IF(LH.GT.LL) LH=LL
C MOVE DISPLACEMENTS INTO CORE FOR LB LOAD CONDITIONS
REWIND 2
C STRESS PORTHOLE
IF(NIOSV.EQ.1) AWRITE (NT10) LT,LH
NO=NEOBANDBLOCK
DO 200 NN=1,NBLOCK
READ (2) N
N=NEQB
IF (NN.EQ.1) N=NEQ-N+NEQB
NO=NO-NEQB
DO 200 J=1,N
I=NO+J
DO 200 L=LT,LH
K=L+LL
200 D(I,J)=B(I,J)
LK=LH-LT+1
CALCULATE STRESSES FOR ALL ELEMENTS FOR LB LOAD CONDITIONS

REAL' (2. b
N=NEQB
IF (NN.EQ.1) N=NEQ-N+NEQB
NO=NO-NEQB
DO 200 J=1,N
I=NO+J
DO 200 L=LT,LH
K=L+LL
200 D(I,J)=B(I,J)
LK=LH-LT+1
CALCULATE STRESSES FOR ALL ELEMENTS FOR LB LOAD CONDITIONS

REWRITE (MTYPE) NPAB
WRITE (MTYPE) NPAB
MTYPE=NPAB+1;
NPAB(1)=0
CALL ELTYPE(MTYPE)
1000 CONTINUE
1100 RETURN
END
SUBROUTINE STBCL (E, B, S, XX, NOD9, H, P, SIGD, DELT, FT, DL, XM, REL, AU,
       IELD, IELX, KL, KGL, KMS, NINT, NINTZ, WTDEN, MSDEN)

CALLED BY: THD3E
CALLS : DER3DS

IMPLICIT REAL(6,1)

HEXAHEDRAL CURVILINEAR THREE-DIMENSIONAL ELEMENTS

ISOPARAMETRIC OR SUBPARAMETRIC

DIMENSION E(0,1),B(0,1),S(0,3,1),XX(0,3,1),NOD9(1),H(1),P(3,1)
       1 SIGD(1),DELT(1),FT(1),DL(1),XM(1),D(9),SUB(6),BO(3))
       2 W(2,3),IPERM(3,3),KDX(3),LXD(3)

COMMON /GAUSS/ XG(4,4),WGT(4,4)

REAL MSDEN
REALA8 MSDEN

DATA IPERM / 1,4,6, 4,2,5, 6,5,3 /

VOL = 0.0

Determine if the material is orthotropic (ISO.EQ.1, ISOTROPIC)

DUM = 0.0
DO 20 I=4,6
   1 = I-1
DO 20 K=1,1
20 DUM = DUM +DABS(E(K,1))
ISO = 1
IF(DUM.GT.1.0E-6) ISO = 0
IF(ISO.EQ.0) GO TO 24
DO 22 I=2,3
22 DUM = DUM +DABS(E(I ,1 ) -E(I-1,1-1))
24 CONTINUE

VOLUME INTEGRATION LOOP
DO 10 LX=1,NINT
DO 10 L=1,NINT
E1=XG(LX,NINT)
E2=XG(LY,NINT)
DO 10 LZ=1,NINTZ
E3=XG(LZ,NINTZ)

WT=WGT(LX,NINT)*WGT(LY,NINT)*WGT(LZ,NINTZ)

CALL DER3D(NEL,XX,YY,ZZ,DET,E1,E2,E3,MOD9,H,P,IELD,IELX)

FACT = WT*DET
FACT2 =DSQRT(FACT)

DO 25 I=1,IELD
K3 = 3*I
K2 = K3-1
K1 = K2-1
BV(K1) = BV1,K1)*FACT2
BV(K2) = BV2,K2)*FACT2
BV(K3) = BV3,K3)*FACT2
25 CONTINUE

DO 30 I=1,NB
DO 30 J=1,NB
30 S(I,J) = S(I,J) + BV(I)*BV(J)

VOL = VOL + FACT

COMPUTE GRAVITY LOADS

IF(KVL.EQ.0) GO TO 150
DO 150 K=1,IELD
150 BL(K) = BL(K) + H0*K*FACTA*WIDEN

COMPUTE THERMAL LOADING NODE FORCE VECTOR

IF(KTL.EQ.0) GO TO 190

1. ELEMENT TEMPERATURE DIFFERENCE AT THIS INTEGRATION POINT

(B1,C1,T)

BT = 0.0
DO 160 I=1,IELD
160 BT = BT + H1*K*DELT
200 BT = BT*FACT
2539 C 2. INITIAL STRESSES AT (R,S,T)
2540 C
2541 C DO 170 K=1,6
2542 170 SDT(K) = SIGDT(K)*DT
2543 C
2544 C 3. NODE FORCES
2545 C
2546 C DO 180 K=1,ND
2547 DO 175 I=1,6
2548 175 FT(K) = FT(K) + B(I,K)*SDT(I)
2549 180 CONTINUE
2550 C
2551 C WRITE(28,A) ' DT,DELT',DT,(DELT(K),K=1,IELD)
2552 C WRITE(28,A) ' SIGDT ',(SIGDT(K),K=1,6)
2553 C WRITE(28,A) ' FT '--------------------------------------'
2554 C WRITE(28,A) (FT(K),K=1,6)
2555 C 190 CONTINUE
2556 10 CONTINUE
2557 C
2558 C DO 35 I=1,2
2559 DO 35 J=1,IC
2560 IC = ND-I
2561 DO 35 J=1,IC
2562 M=J+I
2563 35 S(M,J) = S(J,M)
2564 C
2565 C COMPLETE THE K-MATRIX WITH APPROPRIATE MATERIAL CONSTANT MULTI-
2566 C PLICATIONS OF THE INTEGRATED B(I)AB(J) ARRAY.
2567 C
2568 C 1. TEST FOR MATERIAL TYPE
2569 C
2570 C IF(ISO.EQ.0) GO TO 75
2571 C
2572 C A. ISOTROPIC MATERIAL
2573 C
2574 C DO 60 I=1,1ELD
2575 DO 60 J=1,IC
2576 DO 60 J=1,IC
2577 C
2578 C DO 60 1=1,1ELD
2579 K3 = 341
2580 K2 = K3-1
2581 K1 = K2-1
2582 K0 = K1-1
2583 DO 60 J=1,1ELD
2584 L3 = 3A1
2585 L2 = L3-1
2586 L1 = L2-1
2587 L0 = L1-1
2588 C
2589 IC = 0
2590 DO 40 I=1,3
2591 M = I+10
2592 DO 40 J=1,3
2593
N = JJ + L0
IC = IC + 1
D(IC) = S(M, N)
40 CONTINUE
C
S(K1, L1) = D(1) * D1 + (D(5) + D(9)) * D3
S(K2, L2) = D(5) * D1 + (D(1) + D(9)) * D3
S(K3, L3) = D(9) * D1 + (D(5) + D(1)) * D3
S(K1, L2) = D(2) * D2 + D(4) * D3
S(K2, L1) = D(4) * D2 + D(2) * D3
S(K3, L2) = D(2) * D2 + D(4) * D3
S(K1, L3) = D(2) * D2 + D(7) * D3
S(K3, L1) = D(4) * D2 + D(2) * D3
S(K2, L3) = D(7) * D2 + D(2) * D3
C
GO TO 110
C
B. ANISOTROPIC MATERIAL
C
DO 100 I=1, IELD
K0 = 3*AI-3
DO 100 J=1, IELD
LO = 3*AJ-3
C
DO 80 II=1, 3
DO 95 JJ=1, 3
II = K0 + JJ
DO 80 II=1, 3
DO 95 JJ=1, 3
K = IF(IV=1, 1, 3)
DO 95 K=1, 3
II = K0 + K
DO 83 L=1, 3
LO = 0 + L
DO 83 L=1, 3
M = IF(IV=1, 2, 3)
DO 90 L=1, 3
K = LDx(JJ)
DO 90 L=1, 3
S = SUM + W[I1, JJ] * ELM[K, L]
C
SUM = 0.0
DO 90 L=1, 3
K1 = LDx(I1)
DO 85 JJ=1, 3
K2 = LDx(JJ)
DO 85 JJ=1, 3
DO 95 K=1, 3
DO 95 JJ=1, 3
DO 90 L=1, 3
DO 90 L=1, 3
GO TO 110
C
S(I1, I2) = SUM
CONTINUE

CONTINUE

REFLECT FOR Symmetry

DO 200 I=1,ND

DO 200 J= I,ND

S(J,I) = S(I,J)

CONSTRUCT THE LUMPED MASS MATRIX

IF(KMS.EQ.0) RETURN

FACT = VOLA/MSDEN/ICLD

DO 200 K=1,ND

XM(K) = FACT

RETURN

SUBROUTINE THDEF (ID,X,Y,Z,T,DEN,RHO,NTP,EE,

DCA,NFACET,LT,PW,A,LOC,MAXPTS,SS,

RETURN

FACT = I/OA/MDEN/IELD

DO 0 K=ND

XM(K) = FACT

RETURN

END

SUBROUTINE THDEF (ID,X,Y,Z,T,DEN,RHO,NTP,EE,

DO 200 K=1,ND

XM(K) = FACT

RETURN

SUBROUTINE THDEF (ID,X,Y,Z,T,DEN,RHO,NTP,EE,

RETURN

SUBROUTINE THDEF (ID,X,Y,Z,T,DEN,RHO,NTP,EE,

RETURN

SUBROUTINE THDEF (ID,X,Y,Z,T,DEN,RHO,NTP,EE,

RETURN

SUBROUTINE THDEF (ID,X,Y,Z,T,DEN,RHO,NTP,EE,

RETURN

SUBROUTINE THDEF (ID,X,Y,Z,T,DEN,RHO,NTP,EE,

RETURN

SUBROUTINE THDEF (ID,X,Y,Z,T,DEN,RHO,NTP,EE,
DATA T1, T2
STPTS(1,1) = 1.
STPTS(2,1) = -1.
STPTS(3,1) = -1.
STPTS(4,1) = 1.
STPTS(5,1) = 1.
STPTS(6,1) = -1.
STPTS(7,1) = -1.
STPTS(8,1) = 1.
STPTS(9,1) = 0.
STPTS(10,1) = -1.
STPTS(11,1) = 0.
STPTS(12,1) = 1.
STPTS(13,1) = 0.
STPTS(14,1) = -1.
STPTS(15,1) = 0.
STPTS(16,1) = 1.
STPTS(17,1) = 1.
STPTS(18,1) = -1.
STPTS(19,1) = -1.
STPTS(20,1) = 1.
STPTS(21,1) = 0.
STPTS(22,1) = 1.
STPTS(23,1) = -1.
STPTS(24,1) = 0.
STPTS(25,1) = 0.
STPTS(26,1) = 0.
STPTS(27,1) = 0.
STPTS(28,1) = 0.
STPTS(29,1) = 0.
STPTS(30,1) = 1.
STPTS(31,1) = 1.
STPTS(32,1) = -1.
STPTS(33,1) = -1.
STPTS(34,1) = 1.
STPTS(35,1) = 1.
STPTS(36,1) = 0.
STPTS(37,1) = -1.
STPTS(38,1) = 1.
STPTS(39,1) = 0.
STPTS(40,1) = 1.
STPTS(41,1) = 0.
STPTS(42,1) = 1.
STPTS(43,1) = 0.
STPTS(44,1) = 1.
STPTS(45,1) = 0.
STPTS(46,1) = 1.
STPTS(47,1) = 0.
STPTS(48,1) = 0.
STPTS(49,1) = -1.
STPTS(50,1) = -1.
STPTS(51,1) = 0.
STPTS(52,1) = 0.
STPTS(53,1) = 1.
STPTS(54,1) = 1.
STPTS(55,1) = -1.
STPTS(56,1) = -1.
STPTS(57,1) = 0.
STPTS(58,1) = 0.
STPTS(59,1) = 1.
STPTS(60,1) = 1.
STPTS(61,1) = -1.
STPTS(62,1) = -1.
STPTS(63,1) = 0.
STPTS(64,1) = 0.
STPTS(65,1) = 1.
STPTS(66,1) = 1.
STPTS(67,1) = -1.
STPTS(68,1) = -1.
STPTS(69,1) = 0.
STPTS(70,1) = 0.
STPTS(71,1) = 1.
STPTS(72,1) = 1.
STPTS(73,1) = -1.
STPTS(74,1) = -1.
STPTS(75,1) = 0.
STPTS(76,1) = 0.
STPTS(77,1) = 1.
STPTS(78,1) = 1.
STPTS(79,1) = -1.
STPTS(80,1) = -1.
STPTS(81,1) = 0.
STPTS(82,1) = 0.
STPTS(83,1) = 1.
STPTS(84,1) = 1.
STPTS(85,1) = -1.
STPTS(86,1) = -1.
STPTS(87,1) = 0.
STPTS(88,1) = 0.
STPTS(89,1) = 1.
STPTS(90,1) = 1.
STPTS(91,1) = -1.
STPTS(92,1) = -1.
STPTS(93,1) = 0.
STPTS(94,1) = 0.
STPTS(95,1) = 1.
STPTS(96,1) = 1.
2755   STPTS(26,2)=0.
2756   STPTS(27,2)=0.
2757   STPTS( 1,3)=1.
2758   STPTS( 2,3)=1.
2759   STPTS( 3,3)=1.
2760   STPTS( 4,3)=1.
2761   STPTS( 5,3)=-1.
2762   STPTS( 6,3)=-1.
2763   STPTS( 7,3)=-1.
2764   STPTS( 8,3)=-1.
2765   STPTS( 9,3)= 1.
2766   STPTS(10,3)= 1.
2767   STPTS(11,3)= 1.
2768   STPTS(12,3)= 1.
2769   STPTS(13,3)=-1.
2770   STPTS(14,3)=-1.
2771   STPTS(15,3)=-1.
2772   STPTS(16,3)= 1.
2773   STPTS(17,3)= 0.
2774   STPTS(18,3)= 0.
2775   STPTS(19,3)= 0.
2776   STPTS(20,3)= 0.
2777   STPTS(21,3)= 0.
2778   STPTS(22,3)= 0.
2779   STPTS(23,3)= 0.
2780   STPTS(24,3)= 0.
2781   STPTS(25,3)= 0.
2782   STPTS(26,3)=-1.
2783   STPTS(27,3)=-1.
2784   XG(1,1) = 0.
2785   XG(2,1) = 0.
2786   XG(3,1) = 0.
2787   XG(4,1) = 0.
2788   XG(1,2) = -.57735029918960
2789   XG(2,2) = .57735029918960
2790   XG(3,2) = 0.
2791   XG(4,2) = 0.
2792   XG(1,3) = -.7745966924150
2793   XG(2,3) = 0.
2794   XG(3,3) = .7745966924150
2795   XG(4,3) = 0.
2796   XG(1,4) = -.6613631159410
2797   XG(2,4) = -.32998104358490
2798   XG(3,4) = .32998104358490
2799   XG(4,4) = .36113631159410
2800   WGT(1,1) = 2.0
2801   WGT(2,1) = 0.0
2802   WGT(3,1) = 0.0
2803   WGT(4,1) = 0.0
2804   WGT(1,2) = 1.0
2805   WGT(2,2) = 1.0
2806   WGT(3,2) = 0.0
2807   WGT(4,2) = 0.0
2808   WGT(1,3) = .55555555555556 D0
WGT(2,3) = .38888888888888888 DO
WGT(3,3) = .55555555555555555 DO
WGT(4,3) = 0.0
WGT(1,4) = .347854851375 DO
WGT(2,4) = .5821451548625 DO
WGT(3,4) = .5821451548625 DO
WGT(4,4) = .347854851375 DO

2816 C
2817 NTSBV = NODEX
2818 DO 10 I=1,4
2819 DO 10 J=1,4
2820 10 R(I,J) = 0.0
2821 DO 14 I=1,4
2822 DO 14 J=1,4
2823 14 SF(1,3)=0.0

2824 C
2825 PRINT ELEMENT CONTROL VARIABLES
2826 C
2827 WRITE (33,3001) NODE,NUMMAT,MAXIP,NORTHO,NDLS,MAXNOE,NOPT,LIT,ELEM,
2828 INT
2829 C
2830 C READ AND CHECK INPUT UP TO THE ELEMENT DATA CARDS
2831 C
2832 CALL INP21 (NUMMAT,MAXIP,NORTHO,NDLS,NOPT,NTBSV,NUMNP.,X,
2833 1
2834 C
2835 C READ ELEMENT DATA CARDS
2836 C
2837 NREAD = 9
2838 IF(MAXNOE.GT.8) NREAD = 21
2839 C
2840 WRITE (33,3014) (I,1=1,8)
2841 IF(MAXNOE.GT.8) WRITE (33,3014) (I,1=9,21)
2842 C
2843 C
2844 NEL = 0
2845 C
2846 C CARD FOR ELEMENT NUMBER ONE ONLY
2847 C
2848 READ (5,1009) INEL,NDIS,NXYZ,NUMAT,MAXES,IOPI,TZ,KG,RESINT,INT
2849 1,IREUSE,(LS(I),I=1,4)
2850 READ (5,1009) (NOD(I),I=1,NREAD)
2851 IREUSE = 0
2852 IF(INEL.EQ.1) GO TO 51
2853 WRITE (33,4014) INEL
2854 WRITE (33,4014)
2855 STOP
2856 C
2857 C CARDS FOR ALL OTHER ELEMENTS
2858 C
2859 50 READ (5,1009) INEL,NDIS,NXYZ,NUMAT,MAXES,IOPI,TZ,KG,RESINT,INT
2860 1,IREUSE,(LS(I),I=1,4)
2861 READ (5,1009) (NOD(I),I=1,NREAD)
2862 C
DATA ADMISSIBILITY CHECK

51 IF(NDIS.EQ.0) NDIS = MAXNOD
52 IF(NDIS.LE.MAXNOD) GO TO 5051
53 IF(MAXNOD.GE.8) GO TO 52
54 IF(MAXE5.LE.MANT.EMAXMAT) GO TO 54
55 IF(MAXES.LE.NORTHO) GO TO 55
56 DO 57 I=1,4
57 CONTINUE
58 IF(REUSE) (LS(I), I=1,4)
59 IF(KG.EQ.0) KG = 1
60 IF(NRSINT.EQ.0) NRSINT = INTS
61 IF(NTINT.EQ.0) NTINT = INTI

B-54
DO 58 I=1,8
IF(NOD(I).GE.1,.AND. NOD(I).LE.NUMNP) 60 TO 58
WRITE (33,3015) INEL,NDIS,NXYZ,NMAT,MAXES,IOP,TZ,KG,NRSINT,NTINT
1,IREUSE,(LS(I),J=1,4)
WRITE (33,4021) I,NOD(I)
STOP
53 CONTINUE
IF(MAINMOD.LT.9) GO TO 60
II = 0
DO 59 I=9,21
IF(NOD(I).EQ.0) GO TO 59
II = II + 1
NOD9(I) = I
WRITE (33,4021) I,NOD(I)
STOP
59 CONTINUE
I = II + 3
IF(I.EQ.NDIS) GO TO 60
WRITE (33,4025) I,NDIS
STOP
NEL = NEL + 1
ML = INEL - NEL
IF(ML) 65,70,60
WRITE (33,4022) INEL
STOP
SAVE THE DATA FOR ELEMENT NUMBER AINELA FOR POSSIBLE USE IN DATA GENERATION
70 KDIS = NDIS
KXYZ = NXYZ
KMAT = NMAT
KAXES = MAXES
KIOP = IOP
TTZ = TZ
KKG = KG
KRSINT = NRSINT
KTINT = NTINT
KREUSE = IREUSE
DO 72 I=1,4
72 KLS(I) = LS(I)
DO 74 I=1,4
74 KOD(I) = NOD(I)
TAG = TAG
GO TO 90
C INCREMENT THE NON-ZERO NODE NUMBERS FROM THE PRECEEDING ELEMENT
DATA HUMIDITY CHECK

51 IF(ND15.EQ.0) NDIS = MKIND

80 DO 85 I=1,MI
81 IF(KOB(1,1).LT.1) GO TO 85
82 KOB(1) = KOB(1) + MKG
83 CONTINUE
84 TAG = TAG
85 DO 100 N = 1, NDIS
86 CONTINUE
87 DO 90 N = 1, NDIS
88 CONTINUE
89 DO 104 I=1,12
90 IE(TAV.GT.EE(I,1,KMAT)) = 0
91 IE(TAV.EO.EE(I,1,KMAT)) = 104
92 WRITE (33,4030) TAV,NEL,<MAT
93 STOP
94 IE(M1.EQ.-1) = -1
95 IE(M1,1) = 0
96 IE(M2,1) = 0
97 IF(N1.1.EQ.1) = 1
98 IE = (TAV - EE(K1,1,KMAT)) / DT
99 CONTINUE
100 DO 106 I=1,12
101 E(I) = EE(I,1,KMAT) + RATIO * EE(K2,1,KMAT) - EE(K1,1,KMAT)
102 CONTINUE
103 CALL E5SLAW (D,E,TEMP,URCA(1,1),K1,KMAT,NMAT,MAT,DUM,ALPHA)
104 CALL E5SLAW (D,E,TEMP,URCA(1,1),K1,KMAT,NMAT,MAT,DUM,ALPHA)
105 CALL E5SLAW (D,E,TEMP,URCA(1,1),K1,KMAT,NMAT,MAT,DUM,ALPHA)
106 CALL E5SLAW (D,E,TEMP,URCA(1,1),K1,KMAT,NMAT,MAT,DUM,ALPHA)
107 CALL E5SLAW (D,E,TEMP,URCA(1,1),K1,KMAT,NMAT,MAT,DUM,ALPHA)
108 CALL E5SLAW (D,E,TEMP,URCA(1,1),K1,KMAT,NMAT,MAT,DUM,ALPHA)
109 CALL E5SLAW (D,E,TEMP,URCA(1,1),K1,KMAT,NMAT,MAT,DUM,ALPHA)
110 CALL E5SLAW (D,E,TEMP,URCA(1,1),K1,KMAT,NMAT,MAT,DUM,ALPHA)
111 CALL E5SLAW (D,E,TEMP,URCA(1,1),K1,KMAT,NMAT,MAT,DUM,ALPHA)
112 CALL E5SLAW (D,E,TEMP,URCA(1,1),K1,KMAT,NMAT,MAT,DUM,ALPHA)
113 CALL E5SLAW (D,E,TEMP,URCA(1,1),K1,KMAT,NMAT,MAT,DUM,ALPHA)
114 CALL E5SLAW (D,E,TEMP,URCA(1,1),K1,KMAT,NMAT,MAT,DUM,ALPHA)
115 CALL E5SLAW (D,E,TEMP,URCA(1,1),K1,KMAT,NMAT,MAT,DUM,ALPHA)
116 CALL E5SLAW (D,E,TEMP,URCA(1,1),K1,KMAT,NMAT,MAT,DUM,ALPHA)
117 CALL E5SLAW (D,E,TEMP,URCA(1,1),K1,KMAT,NMAT,MAT,DUM,ALPHA)
118 CALL E5SLAW (D,E,TEMP,URCA(1,1),K1,KMAT,NMAT,MAT,DUM,ALPHA)
119 CALL E5SLAW (D,E,TEMP,URCA(1,1),K1,KMAT,NMAT,MAT,DUM,ALPHA)
120 CALL E5SLAW (D,E,TEMP,URCA(1,1),K1,KMAT,NMAT,MAT,DUM,ALPHA)
121 CALL E5SLAW (D,E,TEMP,URCA(1,1),K1,KMAT,NMAT,MAT,DUM,ALPHA)
122 CALL E5SLAW (D,E,TEMP,URCA(1,1),K1,KMAT,NMAT,MAT,DUM,ALPHA)
123 CALL E5SLAW (D,E,TEMP,URCA(1,1),K1,KMAT,NMAT,MAT,DUM,ALPHA)
124 CALL E5SLAW (D,E,TEMP,URCA(1,1),K1,KMAT,NMAT,MAT,DUM,ALPHA)
125 CALL E5SLAW (D,E,TEMP,URCA(1,1),K1,KMAT,NMAT,MAT,DUM,ALPHA)
126 CALL E5SLAW (D,E,TEMP,URCA(1,1),K1,KMAT,NMAT,MAT,DUM,ALPHA)
127 CALL E5SLAW (D,E,TEMP,URCA(1,1),K1,KMAT,NMAT,MAT,DUM,ALPHA)
128 CALL E5SLAW (D,E,TEMP,URCA(1,1),K1,KMAT,NMAT,MAT,DUM,ALPHA)
129 CALL E5SLAW (D,E,TEMP,URCA(1,1),K1,KMAT,NMAT,MAT,DUM,ALPHA)
DO 130 I=1,KDIS
  II = MOD(I) 
  IF(I.LT.9) GO TO 125
  JJ = MODM(I-8)
  II = MOD(JJ)
  125 XX(1,I) = X(JJ)
  XX(2,I) = Y(JJ)
  XX(3,I) = Z(JJ)
  130 CONTINUE

COMPUTE THE ELEMENT STIFFNESS, MASS, THERMAL AND GRAVITY LOAD
MAATI

DO 170 I=1,4
  DO 170 J=1,4
  RF(I,J)=0.0
  IF(KREUSE.EQ.1) GO TO 300
  DO 180 I=1,DIS
  DL(I)=0.0
  DO 190 I=1,ND
    THERMAL LOADS
    FT(I)=0.0
    KTL = 0
    DUX = 0.0
    DO 200 I=1,4
      DUX = DUX + DABS(TLF(I))
      IF(DUX.GT.1.0E-06) KTL = 1
      IF(KTL.EQ.1) THEN
        WRITE(99,A) '$$$$$$$$$ ktl==1' !!!!!
      END IF
  180 CONTINUE
  IF(KTL.EQ.0) THEN
    IF(NDYN.GT.0) GO TO 235
  ELSE
    IF(KTL.EQ.0 .DR. NDPIN.GT.0) GO TO 235
  END IF
  A.INITIAL STRESS CONSTANTS
  DO 210 I=1,6
    SIGDT(I) = 0.0
    SIGDT(I) = SIGDT(I) + D(I,K)*ALPHA(K) ! I changed to 1 - first
  210 CONTINUE
  B. VECTOR OF NODE TEMPERATURE DIFFERENCES
  DO 230 I=1,KDIS
    II = MOD(I)
    IF(I.LT.9) GO TO 220
    J = MODM(I-8)
  220 CONTINUE
  230 CONTINUE
II = KOD(J)
220 DELT(I) = T(I) - TIZ
230 CONTINUE

C. CLEAR THE THERMAL LOAD NODE FORCE VECTOR

2. GRAVITY LOADS

3087 DUX=0.0
3088 DO 250 I=1,4
3089 DUX = DUX +DABS(XLF(I)) +DABS(YLF(I)) +DABS(ZLF(I))
3090 KGL = 0
3091 IF(DUX.GT.1.0E-6) KGL = 1
3092 IF (NDYN.GT.0) KGL = 0

3. MASS MATRIX

KMS = 0
3097 IF(NDYN.GT.0) KMS = 1

3099 DO 270 K=1,ND
3100 
3101 4. STIFFNESS MATRIX

3103 270 XM(K) = 0.0
3104 DO 280 I=1,ND
3105 DO 280 K=1,4
3106 280 SS(I,K) = 1.0
            
CALL ST8R21 (D,B,SS,XX,NOD9M,H,P,SIGDT,DELT,FI,DL,XM,NEL,ND,KDIS,
1      KXYZ,KTL,KGL,KMS,KRSINT,KINT,DENC,MAT) ,RHO(MAT) 

3111 
3112 C. NODE FORCES DUE TO THERMAL DISTORTION

3115 300 IF (KTL.EQ.0) GO TO 325
3116 DO 320 I=1,ND
3117 DO 310 K=1,4
3118 310 RF(I,K) = FI(I) TLF(K)
3119 320 CONTINUE
3120 
3121 C. NODE FORCES DUE TO STATIC ACCELERATIONS

3124 325 IF (KGL.EQ.0) GO TO 350
3125 DO 340 I=1,KDIS
3126 K3 = 3[I
3127 K2 = K3-1
3128 K1 = K2-1
3129 DO 330 L=1,4
3130 RF(K1,L) = RF(K1,L) + XLF(L) ADL(I)
3131 RF(K2,L) = RF(K2,L) + YLF(L) DL(I)
3132 330 RF(K3,L) = RF(K3,L) + ZLF(L) DL(I)
3133 340 CONTINUE
3134 C
3135 C COMPUTE NODAL FORCES DUE TO ELEMENT SURFACE LOADINGS
3136 C
3137 350 IF(NSEL.LT.1.OR.NSEL.GT.0) GO TO 405
3138 C
3139 DO 400 L=1,4
3140 IF(NSEL.LT.1.) GO TO 400
3141 M = K(1,L)
3142 IF(M.LT.1) GO TO 400
3143 DO 360 K=1,M
3144 C
3145 360 PL(K) = 0.0
3146 CALL FACES surf,NDL,Lxx,mxx,ND9,m,m,p,PL,GROUND K,LIM
3147 1 PW(1,n,m)
3148 C
3149 DO 370 I=1,M
3150 C
3151 370 RE(I,L) = RE(1,L) + PL(I,L)
3152 400 CONTINUE
3153 405 CONTINUE
3154 C
3155 C ASSIGN EQUATION NUMBERS TO THE ELEMENT DEGREES OF FREEDOM
3156 C
3157 410 K = -3
3158 DO 420 I=1,KDIS
3159 II = KOD(I)
3160 IF(I.LT.9) GO TO 410
3161 JJ = MOD9(I-2)
3162 II = KOD(JJ)
3163 415 K = K+3
3164 DO 430 L=1,3
3165 N = K+L
3166 420 LM(N) = ID(I,L)
3167 C
3168 IF(KIOP.GT.0) NS = GMAXKTS(KIOP)
3169 IF(KIOP.GT.0) NS = 0
3170 IF(NSEL.GT.0) NS=42
3171 C
3172 C SAVE STIFFNESS AND LOAD MATRICES
3173 C
3174 CALL CLEAN(MBAND,NDIF,LM,XM,SS,RE,NB,63,NS)
3175 C
3176 C COMPUTE STRESS RECOVERY MATRICES
3177 C
3178 IF(NSEL.LT.1) GO TO 425
3179 NDF=7
3180 DO 422 I=1,7
3181 422 LOCOP(1)=I + 20
3182 GO TO 450
3183 425 IF(KIOP.GT.0) GO TO 440
3184 NDF = MAX(FTS,KIOP)
3185 DO 430 I=1,NDF
3186 430 LOCOP(I) = LOC(I,KIOP)
3187 C

D-59
3187 GO TO 450
3188 440 MOP = 1
3189 LOCOP(1) = 21
3190 C
3191 450 IF(MODEA.EQ.1) GO TO 510
3192 C
3193 C CONSIDER EACH OUTPUT LOCATION
3194 C
3195 DO 500 L=1,MOP
3196 C
3197 M = LOCOP(L)
3198 E1 = STPTS(M,1)
3199 E2 = STPTS(M,2)
3200 E3 = STPTS(M,3)
3201 C
3202 C COMPUTE THE STRAIN-DISPLACEMENT MATRIX AT THIS LOCATION
3203 C
3204 CALL DECODE (NEL, K, L, DET, E1, E2, E3, MOD9M, H, P, KDIS, KXYZ)
3205 C
3206 DO 470 I=1,6
3207 N = 6*(L-1)+1
3208 DO 465 J=1,N
3209 Q = 0.0
3210 DO 460 K=1,6
3211 460 Q = Q + D(I,K)*B(K,J)
3212 465 SBT(N,J) = Q
3213 470 CONTINUE
3214 C
3215 C FORM THE INITIAL STRESS CORRECTIONS DUE TO THERMAL LOADS
3216 C
3217 IF(KTL.EQ.0 .OR. NODEL.GT.0) GO TO 500
3218 C
3219 C
3220 C 1. TEMPERATURE DIFFERENCE AT THIS LOCATION
3221 C
3222 Q = 0.0
3223 DO 480 K=1,6
3224 C
3225 C 2. VECTOR OF INITIAL STRESSES
3226 C
3227 480 Q = Q + H(K)*DELT(K)
3228 DO 485 K=1,6
3229 485 VIS(K) = -Q * SIGD(K)
3230 C
3231 DO 490 I=1,6
3232 N = 6*(L-1)+1
3233 C
3234 DO 495 K=1,4
3235 490 SF(N,K) = VIS(I)*TLE(K)
3236 C
3237 500 CONTINUE
3238 C
3239 C SAVE THE STRESS RECOVERY ARRAYS
3240 C
3241 C
3242 510 CONTINUE
3243 C
3244      IF(MODEX.EQ.0)
3245 1 WRITE (1) ND,NS,(LM(I),I=1,ND), (SDT(I,1),I=1,NS),J=1,ND.
3246 2 (SF(I,1),I=1,NS),J=1,4)
3247 C
3248 C PRINT DATA FOR THE CURRENT ELEMENT
3249 C
3250 WRITE (33,3015) NEL,KDIS,KXYZ,KMAT,KAXES,KIDP,ITZ,MT,MTL,MTI,
3251 1 KREUSE,KLS
3252 WRITE (33,3017) (KOD(I),I=1,NKHEAD)
3253 C
3254 CAAA DATA FOR HOLE SAVE
3255 IF(NTBV.EQ.1)
3256 1 WRITE (58) NEL,KDIS,KXYZ,KMAT,KAXES,KIDP,ITZ,MT,MTL,MTI,
3257 2 KREUSE,KLS,NKHEAD, 
3258 3 (KOD(I),I=1,NKHEAD)
3259 CAAA
3260 C
3261 C CHECK FOR THE LAST ELEMENT
3262 C
3263 IF(NUMB-NEI) 65,600,530
3264 530 IF(NL) 50, 50, 60
3265 C
3266 600 RETURN
3267 C
3268 FORMATS
3269 C
3270 1008 FORMAT (65,F10.0,15H12)
3271 1009 FORMAT (1615)
3272 C
3273 3001 FORMAT (7X,3AHNUMBER OF 21-NODE ELEMENTS = 16//
3274 7X,3AHNUMBER OF MATERIAL SETS = 16//
3275 2 X,26XMAXIMUM NUMBER OF MATERIAL, /
3276 3 X,3AHTEMPERATURE INPUT POINTS = 16//
3277 4 X,4AHNUMBER OF MATERIAL, /
3278 5 X,4AHMAXIS ORIENTATION SETS = 16//
3279 6 X,4AHNUMBER OF DISTRIBUTED LOAD SETS = 16//
3280 7 X,4AHMAXIMUM NUMBER OF ELEMENT NODES = 16//
3281 8 X,4AHNUMBER OF STRESS OUTPUT SETS = 16//
3282 9 X,4AH NUMBER OF DISTRIBUTED LOAD SETS = 16//
3283 C
3284 3014 FORMAT (52H13 / 6 5 TO 21 NODE SOLID ELE,
3285 1 16H ELEMENT DATA // 8H ELEMENT 2(2X,5HNODES),3,5,
3286 2 5HMA T), 2X,2XHSTRESS,4X,6HSTRESS,2X,4HNODE,2(2X,5HGAUSS,5X,
3287 3 2XH-5X,3XHLSA,3X,3HLSB,3X,3HLSF,3X,3HLSG, /
3288 4 8H -3XH,7X-NDIS-,7H-NXYZ-,3X,5HTABLE,3X,4HAXES,2X,6HMOUTPUT,
3289 5 6X,4HNEE,2X,4HINC.,2(3X,4HFTS.),2X,6HMATRIX,2X,4(2X,4H-UK), /
3290 6 26X,3HNO.,4X,3HSET,5X,3HSET,5X,5HTEMP.,2X,4H-KG-,2X,5H-,-3,4X,
3291 7 3H-),2X,6HRE-USE,2X,8(2X,3HNN-,12)
3292 3015 FORMAT (18,417,18,F10.1,LT,217,18,2X,416)
3293 3016 FORMAT (84,8(2X,2HNN-,12),: / 84X,5(2X,2HNN-,12))
3294 3017 FORMAT (84X,816, : / 84X,816, :) / 84X,516)
SUBROUTINE VECTR2 (V,XI,YI,ZI,XJ,YJ,ZJ,IERR)

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

IERR = 1
X = XJ - XI
Y = YJ - YI
Z = ZJ - ZI
XLN = DSQRT(X**2 + Y**2 + Z**2)
IF(XLN.LE.1.0E-08) RETURN
XLN = 1.0 / XLN
IERR = 0
V(3) = Z * XLN
V(2) = Y * XLN
V(1) = X * XLN
RETURN
END
SUBROUTINE TIME
TS=0.0
RETURN
END

SUBROUTINE TIME
T - CUMULATIVE TASK TIME, RETURNED IN UNITS OF SECONDS
INTEGER get_time, time
DATA get_time /2/
CALL LIBSTAT(TIME, get_time, time)
T = time / 100.0
RETURN
END

SUBROUTINE SORLD
IMPLICIT REAL(A-N,0-Z)

CALLS: ESOL, PRINTU, STRESS
CALLED by: MAIN

STATIC SOLUTION PHASE

COMMON Ali:
COMMON VEL (A-N,0-Z)
COMMON SOL: NELCk, NELA, EUL, NE, IE, TELP(7)
DIMENSION (I, 3000, 6), BL(40000), NX(3, 200)
COMMON/CRA, NCHL, ICH(9000)
DATA NCRA,NCRA:

DIMENSION TMAT(600, 600), TCOL(600), TCOL2(600), TCOLm(600).
IST(600), R(9000)

dimension neq(200), disp(200), npe(2, 200), nu(200),
FORCHE(3, 200), FORCHI(2, 200), FORC2(3, 200), ENER(3)

REAL*4 T, A, L:
INTER=100.

INTERMEDIATE PRINTING

rewind 6
read (6) (i=1, 6, n=1, numnp), i=1, 6) 
write(23, 25) 
write(23, 1023) npe(i, 1), i=1, 6
end do

1023 format (23, 200)
**Statement 3403**

TO read data regarding nodes where forces to be found whose disp. are specified (NB).

**Statement 3405**

& double nodes along the crack propagation (NPARK).

**Statement 3406**

read (5, A) * no of boundary nodes where forces are to be found.

**Statement 3407**

IF(INTPR.LE.2) WRITE(33, A) " nb = ", nb.

**Statement 3408**

-- input node, degree of freedom, displacement

do i=1, nb.

**Statement 3410**

read (5, A) nbc(i), ixi(i), dsav(i)

**Statement 3411**

if (INTPR.LE.2) WRITE (33, A) nbc(i), ixi(i), dsav(i).

**Statement 3413**

end do.

**Statement 3414**

ncrd=0.

**Statement 3415**

if (INTPR.LE.2) WRITE (33, A) node, nx, ndof, ncrkd, icr (ndof), ist (ncrd).

**Statement 3416**

if (INTPR.LE.2) WRITE (33, A) (ij, ixi (ij), 1x=1,6), i=1, 16.

**Statement 3417**

do i=1, nb.

**Statement 3418**

node = nbc(i).

**Statement 3420**

ix = ixi (i).

**Statement 3421**

ndof = id(node, ix).

**Statement 3422**

ncrd = ncrkd + 1.

**Statement 3423**

if (INTPR.LE.2).

**Statement 3424**

WRITE (33, 1029), node, nx, ndof, ncrkd.

**Statement 3425**

icr (ndof) = ncrkd.

**Statement 3426**

ist (ncrd) = ncrkd.

**Statement 3427**

if (INTPR.LE.2).

**Statement 3428**

WRITE (33, 1029), node, nx, ndof, ncrkd, icr (ndof), ist (ncrd).

**Statement 3429**

end do.

**Statement 3430**

format (2x, 518, 518).

**Statement 3431**

if (INTPR.LE.1) WRITE (33, 1028) (icr (ij), ij=1, NEQ).

**Statement 3432**

if (INTPR.LE.2) WRITE (33, 1028) IST.

**Statement 3433**

format (2x, 30f6.2).

**Statement 3434**

read (5, A) npair * no of double nodes & the double nodes

**Statement 3437**

if (INTPR.LE.2).

**Statement 3438**

WRITE (33, A) * npair = ", npair.

**Statement 3439**

do ipair = 1, npair.

**Statement 3440**

read (5, A) (npc (i, ipair), i=1, 2), (nx (ij, ipair), ij=1, 3).

**Statement 3441**

if (INTPR.LE.2).

**Statement 3442**

WRITE (33, A) (npc (i, ipair), i=1, 2), (nx (ij, ipair), ij=1, 3).

**Statement 3443**

do i=1, 2.

**Statement 3444**

node = npc (i, ipair).

**Statement 3445**

do iy = 1, 3.

**Statement 3446**

ndof = id (node, iy).

**Statement 3447**

if (ndof .ne. 0 .and. icr (ndof) .eq. 0 .and. nx (iy, ipair) .eq. 1) then.

**Statement 3448**

ncrd = ncrkd.

**Statement 3449**

icr (ndof) = ncrkd.

**Statement 3450**

ist (ncrd) = ncrkd.

**Statement 3451**

end if.

**Statement 3452**

end do.

**Statement 3453**

end do.

**Statement 3454**

end do.

**Statement 3455**

end do.

**Statement 3456**
SOLVE FOR THE DISPLACEMENT VECTORS

CALL TIME(TT(1))

N1 = 1

NCABD = TOTAL NO. OF ADDITIONAL COLUMNS

LL = 1 + NCABD

NSBO = (MBAND + 1) LL ALGBB

NSB = (MBAND + LL) ALGBB

NS = NSB + 1

N3 = NS + LL ALGBB

NSBB = NSBB * LLX(C, MBAND - 2, NEGB)

IF(NSBB.LT.NSBB) NSBB = NSBB

N4 = NS + MBND

MI = MBAND + NEGB - 1

IF(INTK.LT.2)

WRITE(S,10301) LL, MBAND, NSB, N3, NSBB, N4, MI, N1, N2, NL

WRITE(16,10302) NL

10301 FORMAT(22,128)

rewind 15

rewind 4

do 1j = 1, nbloc

read (4) (A(IK), IK = 1, NSBO)

WRITE(15) (A(IK), IK = 1, NSBO)

end do

CALL SELSL (AM, NL, N3, A(N4), LL, MBAND, NEGB, NSB, MI, N1, N2, NL)

CALL TIME(TT(2))

NL = 2

NL = 18

NWV = LL ALGBB

REWIND NL

REWIND NL1

DO N1 = 1, NL

REWIND NL1

REWIND NWV

WRITE(NL1) AM, N1, NL

END DO

WRITE(16, Imat, col)

do ipair = 1, npair

model = npc1(ipair)

node2 = npc2(ipair)

do idf = 1, 6

ndof1 = 1d(model, idf)

ndof2 = 1d(node2, idf)

if(ndof1.ne.0 .and. ndof2 .ne. 0) then

icrl = icrl + 1

icrl = icrl + 1

if(icrl.ge.6 .and. icrl.ge.2) then

end if

end do

end do
write (33,A)   ---  NODES   ---  D.O.F. RELEASED ---

do itr=1,100
call timer(ttime(tsub1))
read (5,A) ipl,ip2,i;
write (33,A) ipl,ip2,i;
if(ip1.eq.9999.0.and.ip2.eq.9999) go to 1995
do while (ipl.ne.0)
if (ip2.eq.0) then
  node=ipl
  ndof=1d(node,1x)
  if(ndof.ne.0) then
    icr1=icr(ndof)
    ist(icr1)=icr1
  end if
else
  node=ipl
  node2=ip2
  idf=1x
  ndof1=1d(node1,1df)
  ndof2=1d(node2,1df)
  if(ndof1.ne.0.and.ndof2.ne.0) then
    icr1=icr(ndof1)
    icr2=icr(ndof2)
    ist(icr2)=icr2
  end if
else
  read (5,A) ipl,ip2,i;
write (33,A) ipl,ip2,i;
end do
if(INTPR.LE.2) then
WRITE(33,A) '----- IST ----'
WRITE(33,1028) (1st(i,j),i=1,ncrkd)
end if
REWIND 16
READ (16) IMAT,tcol

do i=1,neq
  icr=icr(i)
  ist=1st(icr)
  if(icr.le.9b.and.ist.gt.0) tcol(icr)=tcol(icr)+tcol2(icr1)
end do

do i=1,ncrkd
  ist=1st(i)
  do j=1,ncrkd
    if(jle.0.and.ist.gt.0) tcol(i)=tcol(i)-IMAT(i,j)A(i,j)
disp
    if(ist.gt.0.or.ist.le.0) IMAT(i,j)=0.
    if(ist.le.0.or.ist.jle.0.AND.ist.eq.istj) IMAT(i,j)=1.
end do
A.

. . .

end do

3566 IF(INTRK.LE.1) WRITE(33,A)

3567 ' ' ---- That,TCOL after IST manipulation-----'

3568 DO 1,NCRKD

3569 IF(INTRK.LE.1) WRITE(33,1091) (TIMAT(I,J),J=1,NCRKD),TCOL=1.

3570 END DO

3571 do i=1,ncrkd

3572 tcolm1=0.

3573 do j=1,ncrkd

3574 tmatmj,j=0.

3575 end do

3576 end do

3577 do i=1,ncrkd

3578 isti=1

3579 tcolm1=tcolm1*1+tcol(i)

3580 do j=1,ncrkd

3581 jstj=1

3582 tmatmj,istj=tmatmj,istj+tmatmj,istj

3583 end do

3584 end do

3585 do i=1,ncrkd

3586 if(tmatm(i,.eq.,0) matm(i,1)=1.0

3587 end do

3588 IF(INTRK.LE.1) WRITE(33,A) ' ---- TIMAT,TCOLm before matin-----'

3589 DO 1,NCRKD

3590 IF(INTRK.LE.1) WRITE(33,1091) (TIMAT(I,J),J=1,NCRKD),TCOL=1.

3591 end do

3592 call timewttsw2()

3593 call MATIN(timatm,ncrkd,tcolm,1,DETERM)

3594 call timewttsw2()

3595 IF(INTRK.LE.1) WRITE(33,A) ' ---- TIMAT,TCOLm after matin-----'

3596 DO 1,NCRKD

3597 IF(INTRK.LE.1) WRITE(33,1091) (TIMAT(I,J),J=1,NCRKD),TCOL=

3598 end do

3599 do i=1,ncrkd

3600 isti=1

3601 tcolm1=tcolm1*isti

3602 do j=1,ncrkd

3603 tcolm1=tcolm1+1

3604 end do

3605 if(isti).le.0; tcolm1=dsave(i) ! disp

3606 end do

3607 N24=LLAENGB

3608 Rewind all

3609 do nj=1,nblock

3610 read (all) (nstp,ij-1,now)

3611 end do

3612 do nj=1,nblock

3613 nconst=(nj-1)*neq

3614 read (all) (nstp,ij-1,now)

3615 backspace all

3616 backspace all

3617 do i=1,neq

3618 b(i+nconst)-st
do k=1,ncrk
nk=neqb*(k-1)+neqb+1
b(1+nconst)=b(1+nconst)-(nk)*Acol(k)
end do
end do
IF(INTPK.LE.1) WRITE(33,a) ' --- intermediate solution ---
IF(INTPK.LE.1) WRITE(33,1091) (b(ij),ij=1,neq)
end do
do i=1,neq
ici=iti(1)
if(ici.ne.0) b(1)=tcol(1)
end do
WRITE(33,a) ' --- final displacement solution ---
WRITE(33,1091) (b(ij),ij=1,neq)
REWIND NL
DO NJ=NSBLOCK,1,-1
NCONST=(NJ-1)*NEQB+1
NU=NCONST+NEQB-1
WRITE(NL) (B(IJ),IJ=NCONST,NU)
END DO
!
call tttime(ttsub(4))
do i=1,neq
r(i)=0.
end do
rewind 15
do nj=1,block
DO IJK=NSBO,NSB
A(IJK)=0.0
END DO
read (15) (a(IJK),IJK=1,NSBO)
nconst=(nj-1)*neqb
i=0
j=1+nconst
do i=1,neqb
i=1+j+1
in=1+nconst
r(in)=r(in)+a(1j)Ab(in)
end do
do j=2,nband
i=1,neqb
in=1+nconst
jn=j+nconst+1-1
i=1+j+1
r(in)=r(in)+a(1j)Ab(jn)
r(jn)=r(jn)+a(1j)Ab(in)
end do
end do
end do
WRITE(33,a) ' ------- r vector ------'
WRITE(33,1091) (r(ij),ij=1,neq)
Correction for thermal case --

To find the mechanical loads subtract the thermal loads

It is assumed no external loads are applied at double nodes:

Rewind 15
nsbl=neqbdwbd

do nj=1,nblock
read (15) (aij, ij=1,n.to)
non=(nj-1)neqbd

end do
end do

N=NUMNP
ITRI=ITR-1

WRITE(33,582) ITRI

582 FORMAT (33,582) ITRI

WRITE(33,x) *** STEP *,I4, *** STEP*'/1x,40*(1H),***

WRITE(33,x) **** MECHANICAL LOaDS IN SOL2---
WRITE(33,x) **** MECHANICAL LOaDS IN SOL2---
WRITE(33,x) **** MECHANICAL LOaDS IN SOL2---
WRITE(33,20034)

20034 FORMAT(*,2X,75(1H-))

NAUX=1

DO 500 I=1,NUMNP

IFLAG=0

DO 250 i=1,3

B(i)=0.

D(i+3)=0.0

150 IF(ID(N,I).LT.1) GO TO 250

IDN(I)=IDN(I)

IF(B(i).LT.0.0) IFLAG=1

DO 250 i=1,3

B(i)=B(i)+NAUX

D(i+3)=D(i+3)+NAUX

NAUX=NAUX+1

250 CONTINUE

IF(ITR.NE.1) GO TO 1000

WRITE (33,2004) N,(B*I),I=1,6

500 CONTINUE

1001 FORMAT (33,651.5)

2004 FORMAT(2X,15,6D12.5)

WRITE(33,2004)

20035 FORMAT(2X,'NODU',6X,'U',11X,'W',10X,'F',10X,'E',10X,'T',10X,'F',10X,'E',10X,'T',10X,'F',10X,'E',10X,'T',10X,'F')

20035 FORMAT(2X,'NODU',6X,'U',11X,'W',10X,'F',10X,'E',10X,'T',10X,'F',10X,'E',10X,'T',10X,'F')

IF(ITR.NE.1) THEN

1-69
ENERG(1)=0.0
ENERG(2)=0.0
ENERG(3)=0.0
DO IPAIR=1,NPAIR
NPC=NP(1,IPAIR)
NP2=NP(2,IPAIR)
DO IDG=1,3
ND1=ID(NP1,IDG)
ND2=ID(NP2,IDG)
WRITE(33,A) 'NP1,NPC,ND1,ND2',NP1,NP2,ND1,ND2
WRITE(33,A) 'FORC1,ND1',FORC1(IDG,IPAIR),B(ND1),B(ND2)
nda=nda+IDG,IPAIR)
IF(ND1.GT.0.AND.ND2.GT.0.AND.nda.GE.0)
ENERG(IDG)=ENERG(IDG)-FORCNB(IDG,IDG,AB(ND1),AB(ND2))
END DO
END DO
DO IB=1,NB
NPO=NBC(IB)
DO IDG=1,3
if(IDG.IS.(IB))
ND0=ID(NP0,IDG)
IF(ND0.GT.0)ENERG(IDG)=ENERG(IDG)-FORCNB(IDG,IDG,AB(ND0),0.50
END DO
END DO
WRITE(34,A)
'------- ENERGY RELEASED in (x, y, z) directions -------'
WRITE(34,1048) ENERG
IF (ITR.EQ.2) WRITE(19,1049)
WRITE(19,1048) ENERG
1048 format(Sx,3(g15.8,3x))
1049 format(Sx,'X','Y','Z')
WRITE(34,A) '=================================='
END IF
DO IPAIR=1,NPAIR
NPC=NP(1,IPAIR)
NP2=NP(2,IPAIR)
DO IDG=1,3
ND1=ID(NP1,IDG)
ND2=ID(NP2,IDG)
FORC1(IDG,IPAIR)=R(ND1)
FORC2(IDG,IPAIR)=R(ND2)
WRITE(33,A) 'NP1,NP2,ND1,ND2',NP1,NP2,ND1,ND2
WRITE(33,A) 'IDG,IPAIR,FORC1,FORC2,IDG,IPAIR,FORC1(IDG,IPAIR),'
WRITE(33,A) 'FORC1(IDG,IPAIR)
END DO
END DO
DO IB=1,NB
NPO=NBC(IB)
DO IDG=1,3

NDO=ID(ND, IDG)
END DO
END DO
CALL timer();
C PRINT DISPLACEMENTS
N3=N1+NUMNLX6
N3=N3+2*LL
LL1=1  "REASSIGNED"  "REASSIGNED"  "REASSIGNED"
CALL PRINT(N1, A(N3), A(N3), NEQ, NUMNP, LL1, NBLOCK, NEQ, LL1)
CALL timer();
C COMPUTE AND PRINT ELEMENT STRESSES
M2=N1+4*LL1
M3=N3+NEQ+4*LL1
LB=(MTGT-N3)/(NEQ+12)
CALL STRESS(N1, A(N3), A(N3), NEQ, LB, LL1, NEQ, NBLOCK)
C COMPUTE TIME LOG FOR THE DOUBLE NODES SOLUTION PHASE
DO K=1,4
  ttsub(K) = ttsub(K+1)-ttsub(K)
END DO
IF(INPK.LE.2) WRITE (13,1985) (ttsub(L),L=1,4)
1985  FORMAT('  time for element formation ','F8.2/')
1986  FORMAT('  time for matb ', 'F8.2/')
1987  FORMAT('  time to find global disp. ','F8.2/')
1988  FORMAT('  time to find global nodal forces ','F8.2/')
END DO  KTR
C COMPUTE TIME LOG FOR THE STATIC SOLUTION PHASE
DO 50 K=1,3
  TT(K) = IT(K+1)-IT(K)
50 WRITE (13,2000) IT(L),L=1,3
2000 FORMAT('/ 3 HISTORICAL TIME LOG,  
  0X,21EQUATION SOLUTION =, F8.2,  
  0X,21HOLDISPLACEMENT OUTPUT =, F8.2,  
  0X,21STRESS RECOVERY =, F8.2/')
RETURN
END
SUBROUTINE SESUL
(A,B,MAXA,NAV,NBLOK,NEQB,NAV,MI,NSTIF,NRED,NL,NR)
IMPLICIT REAL*8(A-H,O-Z)
REAL tt(10)
REAL A(10)

COMMON /ELFAKI, NF14, NUMNP, MA, MELYTP, NZ1, NZ2, NZ3, MA4, M5, MTO, NEQ
COMMON/CRK/ICM0, ICNN, ICNN, ICNN, ICNN, ICNN, ICNN, ICNN, ICNN

DATA ICX/0, ICX, ICX, ICX
DATA IDS/0, IDS
DATA DIS/0, DIS
COMMON/IUT/IUT, IUT, IUT, IUT, IUT, IUT, IUT, IUT, IUT
DIMENSION A(NAV, MLA), B(NAV), MAXA(MI)
call timing(1)
INTPR=100

if(INTPR.LE.2) WRITE(33,1029) ( IC(IJ), I=1,NEO)
if(INTPR.LE.2) WRITE(33,1029) ( IST(IJK), IJK=1,NECD)
WRITE(33,1030)

MM=1
MA2=MA - 2
IF(MA2.EQ.0) MA2=1
INC=NEQB - 1
NWA=NEOBANA
NTB=(MA-2)/NEQB + 1
NEB=NTB*NEQB
NEBT=NEB + NEQB
NWV=NEGBANV
NWV=NEBTANV
N1=NL
N2=NR

if(INTPR.LE.2) WRITE(33,1029),( MA, MA2, INC, NEQB, NWA, NTB, NEB, NEBT, NWV, NWV)
if(INTPR.LE.2) WRITE(33,1029), ( NA = 'NAV', NAV)
DO IJ=1, NAV
A(IJ)=0.
B(IJ)=0.
END DO
**Taking the major coefts. out and placing in a vertical list...**

**NCORD - TOTAL NO. OF ADDITIONAL COLUMNS**

3643  IBICJ=IB+(ICJ-1)*NK
3644  B(IBICJ)=0.
3645  END IF
3646  IF(INPR.LE.1) WRITE(63,1016) IB,JB,J,IB,ICJ,ICJ,1J,1J,1J.
3647  endif
3648  1016 FORMAT(2X,6I5,S9.5)
3649  IF(ICI.NE.0.OR.ICJ.NE.0) then
3650  A(1J)=0.
3651  IF(I.EQ.J) A(1J)=1.
3652  END IF
3653  END IF
3654  END DO
3655  END DO
3656  DO 3969 I=1,NK
3657  NJ=NEQBIA
3658  IF(I.EQ.1) then
3659  NJ=NJ+1
3660  ICI=ICI+1
3661  endif
3662  col(I)=0
3663  END IF
3664  end if
3665  end do
3666  DO 3973 J=1,NEQB
3667  NI=NIJ
3668  IF(I.EQ.1) then
3669  NI=NIJ+1
3670  END IF
3671  NJ=I+J+1
3672  A(NIJ)=0
3673  END DO
3674  END DO
3675  write(33,1,'(13,5(i,1x))')
3676  do 3981 I=1,NK
3677  if (INPR.LE.1) then
3678  DO 3977 J=1,NEQB
3679  IA=JI+1
3680  WRITE(33,1019) A(1J),1J-1,IVK,NEQB
3681  1019 FORMAT(2X,1:9.2)
3682  END DO
3683  END if
3684  END DO
3685  WRITE(NI,A)
3686  END DO
3687  END
3688  END
3689  END
3690  END
3691  END
3692  IF(I.EQ.1) then
3693  DO 3993 J=1,NBLOC
3694  READ(NI,A)
3695  WRITE(NSTK1,A)
3696  END DO
3697  END
4000 IF (INPFILE.LE.1) THEN
4001 WRITE(33,4) ---- MAIN ----
4009 DO I=1,NCARD
4002 WRITE(33,1019) (IMAT(I,J),J=1,NCARD),TCOL(I)
4011 END DO
4012 end if
4015 C---- STOP
4016 call timer.
4019 xxx Main loop over all blocks.
4019 99 REWIND WRITE
4019 90 DO 600 I=1,NBLOCK
4019 91 DO 600 J=1,NCARD
4019 92 WRITE(33,4) ---- MAIN ----
4019 93 I=1,NCARD
4019 94 IF (NJ.ME.1) GO TO 100
4019 95 READ (TEST1) (A(IJ),I=1,NAV)
4019 96 IF(INPFILE.LE.1) WRITE(33,4) 'NJ=',NJ,' A-MAP:'
4019 97 IF(INPFILE.LE.1) WRITE(33,1020) (A(IJ),I=1,NAV)
4019 98 FORMAT(410)
4019 99 IF(NOT.GT.1) GO TO 100
4020 KMAX=1
4021 IF(NEG.DT.) RETURN
4022 WRITE(16, 1, MAXA) KMAX
4023 IF(A(1,:).GT.1) RETURN
4024 10 KK=1
4025 C---- IF(INPFILE.LE.1) WRITE(33,1010) KK,A(1)
4026 10 S=L=1,NAV
4027 10 A(I+L)=A(I+L),A(I)
4028 KK=1+NAV
4029 WRITE(16, 2, KK)
4030 RETURN
4031 10 IF(INPFILE.LE.1) GO TO 100
4032 10 REWIND NJ
4033 10 REWIND NJ
4034 10 READ (1, (A(IJ),I=1,NAV))
4035 10 IF(INPFILE.LE.1) WRITE(33,4) 'NJ=',NJ,' A-MAP:
4036 10 IF(INPFILE.LE.1) WRITE(33,1020) (A(IJ),I=1,NAV)
4037 10 RETURN
4038 100 continue
4039 100 IF(INPFILE.LE.1) WRITE(33,4) ' BEFORE FINDING COLUMN HEIGHTS
4040 100 IMX=NEGDB(N+NAV)
4041 100 IF(INPFILE.LE.1) WRITE(33,4) 'NEGDB,NA,NV,IMX',NEGDB,NA,NV,IMX
4042 100 DO I=1,NAV
4043 100 IF(INPFILE.LE.1) WRITE(33,1020) (A(IJ),I=1,NAV)
4044 100 IMX=IMX+1
4045 100 END DO
4047 AAAAA AAA Find column heights.
4048 100 KU=1
4049 100 IM-MIN,NA,NEGDB
4050 100 MAXK=1
DO 110 N=2,MI
1052 IF (N.LE.MM) THEN
1053 KU=KU + NEEQ
1054 KK=KK
1055 MM=MINO(N,KM)
1056 ELSE
1057 KU=KU + 1
1058 NM=KK
1059 IF (N.LE.NEEQ) GO TO 140
1060 NM=NM - 1
1061 END IF
1062 140 DO 160 K=1,MM
1063 IF (A(KK)) 110,160,110
1064 160 KK=KK - INC
1065 110 MAAX(N)=KK
1066
1067 IF (A(1)) 172,174,176
1068 172 KK=NJ-1)ANEEQB + 1
1069 IF (KK.GT.NEEQ) GO TO 590
1070 IF (INTPR.LE.1) WRITE (33,1000) KK
1071 STOP
1072 174 KK=(NJ-1)ANEEQB + 1
1073 C=- IF (INTPR.LE.1) WRITE (33,1010) KK,A(1)
1074
1075 Factorize leading block
1076 DO 200 N=2,NEEQ
1077 NH=MAAX(N)
1078 DO 200 N=2,NEEQ
1079 KL=N-H)200,200,210
1081 DO 210 K=1,INC
1082 D=0
1083 DO 220 KK=KL,HH,INC
1084 D=K - 1
1085 AA=CK)
1086 C=C*K)
1087 D=D + CAAKK
1088 200 A(KK)=C
1089 200 A(N)=A(N) - D
1090
1091 IF (A(N)) 222,224,230
1092 222 KK=(NJ-1)ANEEQB + N
1093 IF (KK.GT.NEEQ) GO TO 590
1094 IF (INTPR.LE.1) WRITE (33,1000) KK
1095 STOP
1096 224 KK=(NJ-1)ANEEQB + N
1097 C=- IF (INTPR.LE.1) WRITE (33,1010) KK,A(N)
1098
1099 DO 250 J=1,MD2
1100 250 IC=NEEB
1101 DO 240 J=1,MD2
1102 MJ=MMA(N+1) - IC
1103 IF (MJ.LE.N) GO TO 240
1104 KU=MIN0(MJ,NM)
4105  \( KN=N+IC \)
4106  \( C=0. \)
4107  DO 300 \( KK=KL,KU,INIC \)
4108  300 \( C=C + A(KK) \times A(KK) + IC \)
4109  \( A(KN)=A(KN) - C \)
4110  IC=IC + NEGB
4111
4111  \( K=N + NIWA \)
4112  DO 450 \( L=1,NJ \)
4114  \( KJ=K \)
4115  \( C=0. \)
4116  DO 440 \( KK=KL,KH,INIC \)
4117  \( KJ=KJ - 1 \)
4118  440 \( C=C + A(KK) \times A(KJ) \)
4119  \( A(K)=A(K) - C \)
4120  450 \( K=R + NEGB \)
4121
4121  200 CONTINUE
4122  IF (INTERLEI) WRITE(33,*) \( ' AFTER FACTORIZING LEADING BLOCK \)
4124  IMX=NEGB+MAX+1
4125  DO 410 \( I=1,NEGB \)
4126  410 C= IF (INTERLEI) WRITE(33,1920) \( (\text{A}(I)),I=1,IMX,NEGB \)
4127  IMX=IMX+1
4128  END DO
4129
4129  1905 FORMAT(33,1920,2)
4130
4130  AAAAA Carry over into trailing blocks
4133  DO 400 \( NK=1,NKTB \)
4135  400 IF (INTERLEI) WRITE(33,*) \( ' NJ,NK ',NJ,NK,' MRNAT' \)
4136  IF \( ((NK+NJ)/GT,NBLOCK) \) GO TO 400
4137  NI=NJ
4138  IF \( ((NJ,EQ.1).OR.(NK.EQ.NKB)) \) NI=NSTIF
4139  READ \( (NI),(S11),I=1,NKTB \)
4141
4141  C=- IF (INTERLEI) WRITE(33,1920) \( (B\(11)),I=1,NKTB \)
4143  ML=NEGB + 1
4144  NK=MINSU(ML-1,NEGB,NK)
4145  IF \( (\text{NK},\text{SU},1) \) ML=\text{NK}
4146  N=ML - ML
4147  KL=NEGB + (NK-1)\text{NEGB}\times\text{NEGB}
4148  N=1
4149
4150  DO 500 \( N=N-ML,NK \)
4151  NH=MAXM(NK)
4152  KL=KL + NEGB
4153  IF \( (NH,LT.KL) \) GO TO 500 AAAAA
4154  K=NEGB
4155  D=0.
4156  DO 520 \( KL=KL,NK,INC \)
4157  C=A(KK) \text{ An} \text{K}
4158  D=D + C \times IC

A(KK) = C
K = K - 1
B(N) = B(N) - D
IF (MD LE 0) GO TO 560
IC = NEQB
DO 540 I = 1, N
MJ = MAXA(M+J) - IC
IF (MJ .LT. KL) GO TO 540
KU = MINO(MJ, NH)
KN = N + IC
C = 0.
DO 575 KK = KL, KU, INC
C = C + A(KK)AMK*IC
B(KN) = B(KN) - C
DO 540 IC = IC + NEQB

578
KN = N + NWA
K = NEQB + NWA
DO 610 L = 1, NW
KJ = K
C = 0.
DO 620 KK = KL, KU, INC
C = C + A(KK)AMK*IC
KJ = KJ - 1
B(KN) = B(KN) - C
KN = KN + NEQB
K = K + NEQB
M = MD - 1
N = N + 1
IF (NTB.NE.1) GO TO 560
WRITE (NRED) A, MH, AM
DO 570 I = 1, NH
A(I) = B(I)
GO TO 600
WRITE (N2)
CONTINUE
N2 = M
WRITE (NRED) A, MH, AM
CONTINUE
CALL TIMETTB(TTT)
AAAA Vector back-substitution
DO 700 K = 1, NWVV
B(K) = 0.
REWIND NL

6-78
4214 DO 800 NJ=1,NBLK
4215 BACKSPACE NRED
4216
4217 READ (NRED) (A(IJ), IJ=1,NAV), (MAXA(IJ), IJ=1,MI)
4218 C-- WRITE(33,A) : Vector back sub. NJ='NJ,' A=' MAT'
4219
4220 C-- IF(INTFL.L.E.1) WRITE(63,1020) (A(IJ), IJ=1,NAV)
4221 BACKSPACE NRED
4222 K=NEBT
4223 GO TO 810 L=1,NAV
4224 DO 820 I=1,NEB
4225 B(K)=B(K)-NEBT
4226 820 K=K -1
4227 310 K=K + NEBT + NEB
4228 KN=0
4229 KN=NEBT
4230 NDIF=NEGB
4231 IF (HJ.LE.1) NDIF=NEGB - (NBLK*NEGB - NEB)
4232 DO 835 L=1,NAV
4233 DO 850 K=1,NDIF
4234 850 B(KN+K)=A(KN+K),A(K)
4235 KK=KK + NEGB
4236 855 KN=KN + NEBT
4237 IF(MA.EQ.1) GO TO 915
4238 ML=NEGB + 1
4239 KL=NEGB
4240 DO 360 M=ML,MI
4241 KL=KL + NEGB
4242 KU=MAXA(M)
4243 IF (KU-KL) GO TO 860,870,870
4244 370 K=NEGB
4245 KN=M
4246 DO 380 L=1,NAV
4247 KJ=K
4248 DO 890 KK=KL,KU,INC
4249 B(KJ)=B(KJ) - A(KJ)*AB(KM)
4250 890 KJ=KJ -1
4251 KN=KN + NEBT
4252 380 K=K + NEBT
4253 360 CONTINUE
4254 N=NEGB
4255 DO 910 L=1,HEGB
4256 KL=M + INC
4257 NU=MAXA(M)
4258 IF (KU-KL) 910,920,920
4259 920 K=N
4260 DO 930 L=1,NAV
4261 KJ=K
4262 DO 940 KK=KL,KU,INC
4263 KJ=KJ -1
4264 940 B(KJ)=B(KJ) - A(KJ)*AB(KM)
4265 930 K=K + NEBT
4266 910 N=K - 1
4267   915  KK=0
4268   916  KN=0
4269   917  DO 950 L=1,NW
4270   918  DO 960 K=1,NEUB
4271   919  KK=KK + 1
4272   920  A(KK)=B(KN+K)
4273   921  KN=KN + NE&T
4274   922  WRITE (NL) (A(K),K=1,NW)
4275   923  IF(INTRP.LE.1) WRITE(33,k) " Solution --'
4276   924  IF(INTRP.LE.1) WRITE (33,1020) (A(K),K=1,NW)
4277   925  CONTINUE
4278   926  call ttime(tt:4))
4279   927  To find y-vector
4280   928  do i=1,ncrkd
4281   929     tcol12(i)=tcol(i)
4282   930     if(i(ist(i)).le.0) tcol(i)=0.
4283   931     end do
4284   932     rewind nstif
4285   933     backspace nl
4286   934     nc2=neqb*ma
4287   935     do nj=1,nblock
4288   936         read (nstif) a
4289   937         nj=nj
4290   938         read (nl, (b(i)),i=1,nv)
4291   939         backspace nl
4292   940         backspace nl
4293   941         nj=neqb(nv-ncrkd)
4294   942         N1J10=NO
4295   943         N1J20=NEQB*(NV-NCRKD)
4296   944         do J=1,ncrkd
4297         do I=1,j
4298         tux=0.0
4299         N1J=110+(I-1)*AEBB
4300         N1J=110+(I-1)*AEBB
4301         do k=1,neQB
4302         N1J=N1J+1
4303         do k=1,neqb
4304         N1J=N1J+1
4305         tux=tux-A(N1J)*AB(N1J)
4306         end do
4307         tmat(i,j)=tmat(i,j)+tux
4308         end do
4309         end do
4310         do J=1,ncrkd
4311         end do
4312         end do
4313         do J=1,ncrkd
4314         end do
4315         end do
4316         end do
4317         end do
4318         end do
4319         end do
4320         do J=1,ncrkd
do i = 1, ncrkd
    tsum = 0.0
    nijs = nijs + (i - 1) * ncp
doi = k = 1, ncp
    nijs = nijs + 1
da do k = 1, ncp
        tsum = tsum + nijs * x(k)
da end do
    tcol(i) = tcol(i) + tsum
end do
if (nintr LE 1.0) then
    WRITE (33, 1091) 'MAT, ICOL towards end -----'
do i = 1, ncrkd
    WRITE (33, 1091) (nim(i, j), j = 1, ncp), ICOL (i)
end do
if (nintr LE 2.0) WRITE (33, 995) (tt (k), k = 1, 4)
daend if
if (nintr LE 2.0) WRITE (33, 995) (tt (k), k = 1, 4)
daend do
end if
call time (tt (5),
if (intr LE 2.0) WRITE (33, 1091) 'TIME LOG IN SESOL -------'
do k = 1, 4
da
    tt (k) = tt (k + 1) - tt (k)
da end do
if (nintr LE 2.0) WRITE (33, 995) (tt (1), i = 1, 4)
da format (995) for mat (1), Time to form mat (1) for double nodes etc. = 18.2;
da format (995) for mat (1), Time to decompose A = matrix = 18.2;
da format (995) for mat (1), Time for vector back substitution = 18.2;
da format (995) for mat (1), Time to form MAT = 18.2)
da STOP (946) STOP XXX ZERO DIAGONAL ENCOUNTERED DURING
nda format (995) for mat (1), 16th EQUATION SOLUTION /
da format (995) for mat (1), 16th EQUATION NUMBER = 16 )
da format (995) for mat (1), 16th EQUATION SOLUTION /
da format (995) for mat (1), 16th EQUATION NUMBER = 16, 5x, THVALUE =, ETC.:
da RETURN
nda END
nda SUBROUTINE MAIN (n, n, n, d, mat)
da IMPLICIT REAL (A-H, O-Z)
da DIMENSION n (999), ncp (600, 1), ipivot (600), index (600, 2), l, row
nda EQUIVALENCE (index, icol), (icol, icol), (amax, t, swap)
da imat (1) = 1;
da do 29 j = 1, n
nda ipivot (j) = 0.0
nda do 556 i = 1, n
nda amax = 0.0
nda do 105 j = 1, n
nda 8-31
4375 IF(IPIVOT(J)-1) 60,105,60
4376 60 do 100 K=1,N
4377 IF(IPIVOT(K)-1) 30,100,740
4378 30 IF(A(MAX, -DABS(A(J,K)))) 35,100,100
4379 35 IROW=J
4380 ICOLUMN=K
4381 AMAX=DABS(A(J,K))
4382 100 CONTINUE
4383 105 CONTINUE
4384
4385 IF(IPIVOT(ICOLUMN)=IPIVOT(ICOLUMN)+1
4386 IE(IPROW-ICOLUMN) 140,260,140
4387 140 DETERM=DETERm
4388 do 200 L=1,N
4389 SWAP=A(IPROW,L)
4390 A(IPROW,L)=A(IPCOLUMN,L)
4391 200 A(IPCOLUMN,L)=SWAP
4392 IF(M) 260,260,210
4393 210 do 250 L=1,M
4394 SWAP=B(IPROW,L)
4395 B(IPROW,L)=B(IPCOLUMN,L)
4396 250 B(IPCOLUMN,L)=SWAP
4397 260 INDEX(I,1)=IROW
4398 INDEX(I,2)=ICOLUMN
4399 PIVOT=A(IPCOLUMN,ICOLUMN)
4400 DT(I)=PIVOT
4401 A(IPCOLUMN,ICOLUMN)=L.0
4402 do 350 L=1,N
4403 350 A(IPCOLUMN,L)=A(IPCOLUMN,L)/PIVOT
4404 IF(M) 380,380,360
4405 360 do 370 L=1,N
4406 370 B(IPCOLUMN,L)=B(IPCOLUMN,L)/PIVOT
4407 380 do 550 L=1, N
4408 IF(L1-ICOLUMN) 400,550,400
4409 400 T=A(L1,ICOLUMN)
4410 A(L1,ICOLUMN)=0.0
4411 do 450 L=1,N
4412 450 A(L1,L)=A(L1,L)-A(IPCOLUMN,L)*T
4413 IF(M) 550,550,460
4414 460 do 500 L=1,N
4415 500 B(L1,L)=B(L1,L)-B(IPCOLUMN,L)*T
4416 550 CONTINUE
4417 do 710 I=1,N
4418 L=N+1-I
4419 C-- DETERM=DETERmADT(L)
4420 IF(INDEX(L,1)-INDEX(L,2)) 630,710,630
4421 630 JROW=INDEX(L,1)
4422 JCOLUMN=INDEX(L,2)
4423 do 705 K=1,N
4424 SWAP=A(K,JROW)
4425 A(K,JROW)=A(K,JCOLUMN)
4426 A(K,JCOLUMN)=SWAP
4427 705 CONTINUE
4428 710 CONTINUE
4429  DO 11 K=1,N
4430  IF(IPIVOT(K).NE.1) GO TO 12
4431  11 CONTINUE
4432  RETURN
4433  12 WRITE(33,991)
4434  991 FORMAT(IUX, MATRIX IS SINGULAR)
4435  740 RETURN
4436  END
APPENDIX - C

LISTING OF THE POSTPROCESSOR, 'PLOT'
THIS IS PROGRAM FOR PLOTTING 3-D GRAPHS USING TEMPLATES.
Routines from I00-II, THIS PROGRAM CAN
SORT OUT STRESSES AND CORRESPONDING COORDINATE LOCATIONS.
THE STRESSES MAY BE SCALING CONVENIENTLY AND EYE
COORDINATES CAN BE CHosen TO OBTAIN DIFFERENT SIZES OF
THE SAME 3-D PLOT.

DIMENSION AX(6),BY(6),STRESS(15,15,6),WORK(200),STRESSES(8),REP(4)
DIMENSION READ(7)

CHARACTER*10 FILENAME
INTEGER I,S

DATA IO/57/,I00/CORE,JO/OUTFIL/8.0/.FONFILE/11.0/
DATA PENS/1.0,1.1,1/
DATA IGNOR,FB,FL,1.0,0.0,0.0/

WRITE (5,555) 'ENTER FILE NAME'
READ (5,555) FILENAME

FORMAT (A)
OPEN (UNIT=100,FILE=FILENAME,STATUS='OLD')
WRITE (10,A) 'SCREEN:1:PRINTER,2:PLOTTER'
READ (10,A) DEV
READ (100,12),(READ(1),I=1,6)

FORMAT(1X,0)

READ (100,A) NNODES,NLNC
IF (NLNC.LE.1) CALL SORT21(RAD,IOD,XX,YY,NX,NY,STRES,NNODES,NLNC)
IF (NNODES.LE.21) GO TO 899
READ (100,A) NX,NY,NL
OPEN (UNIT=100,FILE=FILENAME,STATUS='OLD')
WRITE (5,555) NX,NY,NL
READ (100,A) XX(I),I=1,NX
WRITE (5,555) XX(I),I=1,NX
READ (100,A) YY(I),I=1,NY
WRITE (5,555) YY(I),I=1,NY

N=1

10 READ (100,A) XX(I),I=1,NX
READ (100,A) YY(I),I=1,NY
READ (100,A) Z(I),I=1,NL

20 READ (100,A) XX(I),I=1,NX
READ (100,A) YY(I),I=1,NY
READ (100,A) Z(I),I=1,NL

GO TO 10

GO TO 20

WRITE (10,A) '

IF (DIFFERENT) STRESS IS CALLED CONVENIENTLY AND EYE
COORDINATES CAN BE CHosen TO OBTAIN DIFFERENT SIZES OF
THE SAME 3-D PLOT.

DIMENSION AX(6),BY(6),STRESS(15,15,6),WORK(200),STRESSES(8),REP(4)
DIMENSION READ(7)

CHARACTER*10 FILENAME
INTEGER I,S

DATA IO/57/,I00/CORE,JO/OUTFIL/8.0/.FONFILE/11.0/
DATA PENS/1.0,1.1,1/
DATA IGNOR,FB,FL,1.0,0.0,0.0/

WRITE (5,555) 'ENTER FILE NAME'
READ (5,555) FILENAME

FORMAT (A)
OPEN (UNIT=100,FILE=FILENAME,STATUS='OLD')
WRITE (10,A) 'SCREEN:1:PRINTER,2:PLOTTER'
READ (10,A) DEV
READ (100,12),(READ(1),I=1,6)

FORMAT(1X,0)

READ (100,A) NNODES,NLNC
IF (NLNC.LE.1) CALL SORT21(RAD,IOD,XX,YY,NX,NY,STRES,NNODES,NLNC)
IF (NNODES.LE.21) GO TO 899
READ (100,A) NX,NY,NL
OPEN (UNIT=100,FILE=FILENAME,STATUS='OLD')
WRITE (5,555) NX,NY,NL
READ (100,A) XX(I),I=1,NX
WRITE (5,555) XX(I),I=1,NX
READ (100,A) YY(I),I=1,NY
WRITE (5,555) YY(I),I=1,NY

N=1

10 READ (100,A) XX(I),I=1,NX
READ (100,A) YY(I),I=1,NY
READ (100,A) Z(I),I=1,NL

20 READ (100,A) XX(I),I=1,NX
READ (100,A) YY(I),I=1,NY
READ (100,A) Z(I),I=1,NL

GO TO 10

GO TO 20

WRITE (10,A) 'SCREEN:1:PRINTER,2:PLOTTER'
READ (10,A) DEV
READ (100,12),(READ(1),I=1,6)

FORMAT(1X,0)

READ (100,A) NNODES,NLNC
IF (NLNC.LE.1) CALL SORT21(RAD,IOD,XX,YY,NX,NY,STRES,NNODES,NLNC)
IF (NNODES.LE.21) GO TO 899
READ (100,A) NX,NY,NL
OPEN (UNIT=100,FILE=FILENAME,STATUS='OLD')
WRITE (5,555) NX,NY,NL
READ (100,A) XX(I),I=1,NX
WRITE (5,555) XX(I),I=1,NX
READ (100,A) YY(I),I=1,NY
WRITE (5,555) YY(I),I=1,NY

N=1

10 READ (100,A) XX(I),I=1,NX
READ (100,A) YY(I),I=1,NY
READ (100,A) Z(I),I=1,NL

20 READ (100,A) XX(I),I=1,NX
READ (100,A) YY(I),I=1,NY
READ (100,A) Z(I),I=1,NL

GO TO 10

GO TO 20

WRITE (10,A) 'SCREEN:1:PRINTER,2:PLOTTER'
READ (10,A) DEV
READ (100,12),(READ(1),I=1,6)

FORMAT(1X,0)
find stress location coordinates...

from nodal coordinates for 8 node element.

do i=1,nx
  if (i.gt.1) xx(i)=xx(i)+xx(i-1))/2.
end do
xx(nx)=xx(nx+1)
do i=1,ny
  if (i.gt.1) yy(i)=(yy(i)+yy(i-1))/2.
end do
yy(ny)=yy(ny+1)
continue

nxny=nx*ny
smax=-1.0e+30
smin=1.0e+30
do i=1,nxny
  if (stre(i).gt.smax) smax=stre(i)
  if (stre(i).le.smin) smin=stre(i)
end do
write(5,'(s16.7,'s16.7,'s16.7,'s16.7)') stre

write(10,'(a14)') ex,ey,ez
read(10,a) ex,ey,ez
write(5,'(a14)') fact

write(10,a) 'stre(i)=stre(i)/fact'
end do

write(10,a) '-------------------------------------'
write(10,a) 'scale coordinate-------'

if (cmax.lt.yy(ny)) cmax=yy(ny)
write(5,'(s16.7)') cmax
write(10,a) 'cmax','cmax'
cmax=8.0/cmax

do 444 i=1,nx
  xx(i)=xx(i)*cmax
  do 445 j=1,ny
    yy(j)=yy(j)*cmax
  end do
  fx=ny
444 continue
445 continue

109 FY = NY
110 IF (IDEV.EQ.2) CALL USTART(3.0)
111 CALL USTART
112 IF (IDEV.EQ.1) CALL USTART("OUTP",OUTFIL)
113 IF (IDEV.EQ.0) CALL USTART(OUTFIL)
114 CALL USTART (ENTFILE,ENTFILE)
115 CALL USTART (RESNAME)
116 CALL USTART ('START')
117 CALL USTART ('STOP')
118 CALL USET ('START')
119 CALL USET ('STOP')
120 CALL USET (CONT)
121 CALL USTOP (CONT)
122 CALL UWPART (v,v,...,v,...,v,0)
163 IF (MOD(I,2),NE.0) X(I)=(X(I+1)
164 IF (MOD(I,2),EQ.0) X(I)=(X(I+1)+X(J))/2.
165 111 CONTINUE
166 DO 222 I=NONY,1,-1
167 J=I/2
168 IF (MOD(I,2),NE.0) Y(I)=(I+1)
169 222 IF (MOD(I,2),EQ.0) Y(I)=(Y(I+1)+Y(J))/2.
170 DO 333 I=NONZ,1,-1
171 J=I/2
172 IF (MOD(I,2),NE.0) Z(I)=(J+1)
173 333 IF (MOD(I,2),EQ.0) Z(I)=(Z(I+1)+Z(J))/2.
174 END IF
175 WRITE (5,*) ' SELECT LEVEL (z-coord. No.) of av-plane'
176 DO 41 I=1,MONZ
177 WRITE (5,1) I,2(I)
178 1 FORMAT (2X,12_,2X.F.
179 READ (5,*) nlev
180 JS=0
181 IF (MOD(NLEV,2),NE.0) GO TO 710
182 C-- MIDDLE........
183 JS=2
184 DO 20 J=1,3
185 DO 20 I=1,3
186 IF (LOC(I,2,1,EQ.1)GO TO 21
187 CONTINUE
188 GO TO 36
189 710 IF (NLEV,EQ.1) GO TO 23
190 JS=3
191 DO 10 J=1,3
192 DO 10 I=1,3
193 IF (LOC(I,2,1,EQ.1)GO TO 21
194 CONTINUE
195 23 IF (NLEV,EQ.NONZ) GO TO 26
196 JS=1
197 DO 11 J=1,3
198 DO 11 I=1,3
199 IF (LOC(I,2,1,EQ.1)GO TO 21
200 CONTINUE
201 WRITE (5,*) 'LEVEL NO. DOESN'T MATCH WITH LOC.
202 STOP
203 CONTINUE
204 DO 40 J=1,3
205 ID(J)=0
206 DO 50 I=1,3
207 DO 50 K=1,NLOC
208 IF (LOC(K,2,1,EQ.1)THEN
209 ID(J)=1
210 DO 40 10=40
211 ELSE IF
212 CONTINUE
213 40 CONTINUE
214 DO 45 I=1,3
215 WRITE (5,1) I,2(I)
DO 55     1=1,3
DO 55 J=1,10,10
IF (LOC(K),EQ.16M(J,1,1)) THEN
JD(J)=1
GO TO 45
ENDIF
CONTINUE
45 CONTINUE
NEX=(NEXA-1,1)
NX=0
DO 70   J=1,10,10
IF (JD(J),EQ.2,6M(J,1,2),EQ.0) go to 70
IF (JD(J),EQ.2,6M(J,1,3),EQ.0) go to 70
NX=NX+1
XP(NX)=X(J)
CONTINUE
70 CONTINUE
XP(NX)=X(NEXA)
XP(J)=X(J)
CONTINUE
NEX=(NEXA-1,1)
NX=0
DO 80   J=1,10,10
IF (JD(J),EQ.2,6M(J,1,2),EQ.0) go to 80
IF (JD(J),EQ.2,6M(J,1,3),EQ.0) go to 80
NX=NX+1
YP(NY)=NY
CONTINUE
80 CONTINUE
YP(NY)=NY
YP(Y)=Y(J)
CONTINUE
NEX=(NEXA-1,1)
NX=0
DO 90   J=1,10,10
IF (JD(J),EQ.2,6M(J,1,2),EQ.0) go to 90
IF (JD(J),EQ.2,6M(J,1,3),EQ.0) go to 90
NX=NX+1
NY=NEXA
NEE=NEE+1
B-- Reading stresses 1-rel. no.: 3-rel. no.
DO 81 I=1,NZ
DO 81 J=1,NE
READ (I0,12)   SP(I,J,1:3,1:1,1:1)
81 CONTINUE
12 FORMAT (21H5,2A2)
82 continue
90 continue
MM=M+1
IF (IMM,EQ.1,NK) THEN
K=I
DO 10 I=1,NK
N1=N1+1
NK=N1+1
DO 33 JJ=1,3
272 IF (JD(JJ).GE.0) GO TO 93
273 DO 90 I=NL,N2
274 DO 91 II=1,3
275 IF (1D11).GE.0) GO TO 91
276 IL=18MT(JJ,JI,II)
277 LL=0
278 DO 92 L=1,NLOC
279 92 IF (IL.EQ.LOC(L)) LL=L
280 K=K+1
281 IF (II.EQ.1) LL=LOC
282 IF (II.EQ.2) LL=LOC
283 IF (II.EQ.3) LL=LOC
284 93 CONTINUE
285 90 CONTINUE
286 91 CONTINUE
287 IJ=K-NX+1
288 IF (IJ.NE.1) GO TO 98
289 IF (JD(JJ).GE.0) GO TO 88
290 K=K-NX
291 DO 93 KK=1,NX
292 K=K+1
293 94 ZP(K)=(ZP(K)+ZP(K+NX))/2.
294 93 CONTINUE
295 92 CONTINUE
296 DO 11 J=1,NV
297 DO 10 J=1,NV
298 10 Y=J1+NO-1
299 11 FORMAT (20,16CHAR,4)
300 10 CONTINUE
301 RETURN
302 END
APPENDIX - D

LISTING OF THE EXAMPLE RESULTS
**INPUT DATA for "PHASEGECC" : PHAN.INP**

<table>
<thead>
<tr>
<th>Node</th>
<th>El.</th>
<th>ET</th>
<th>CZE</th>
<th>IMOD</th>
<th>KNE</th>
<th>KNE</th>
<th>KNE</th>
<th>KNE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>E1</td>
<td>1.0</td>
<td>2.0</td>
<td>3.0</td>
<td>4.0</td>
<td>5.0</td>
<td>6.0</td>
<td>7.0</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3.0</td>
<td>4.0</td>
<td>5.0</td>
<td>6.0</td>
<td>7.0</td>
<td>8.0</td>
<td>9.0</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>5.0</td>
<td>6.0</td>
<td>7.0</td>
<td>8.0</td>
<td>9.0</td>
<td>10.0</td>
<td>11.0</td>
</tr>
</tbody>
</table>

**Boundary Conditions**

- Node 1: Fixed in x, y, z directions.
- Nodes 2-11: Displacement boundary conditions.

**Material Data**

- EL: Elastic constant
- ET: Thermal expansion coefficient
- CZE: Coefficient of thermal expansion
- IMOD: Modulus of rigidity
- KNE: Knot numbers

**Data Termination Indicators**

- 1: End of data set
- 2: End of element set
- 3: End of node set
- 4: End of data file

**Notes:**

- Elements 1-11: Linear elements
- Nodes 1-11: Nodes for simulation of split, dir. normal to plane
- Elements 1-11: For the solid that has data

---

**Data for Stress Int.**

- EL: 21.0, 1.0, 0.0, 0.0, 0.0

---

**Data for Displacement Boundary Conditions**

- Nodes 1-11: Displacement boundary conditions.
<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>10</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>8</td>
<td>6</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>9</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>10</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>11</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>12</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>13</td>
<td>11</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>14</td>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>15</td>
<td>13</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>16</td>
<td>14</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>17</td>
<td>15</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>18</td>
<td>16</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>19</td>
<td>17</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>20</td>
<td>18</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1.0000 1.0000 1.0000</td>
</tr>
<tr>
<td>21</td>
<td>19</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>22</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>23</td>
<td>21</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>24</td>
<td>22</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>25</td>
<td>23</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>26</td>
<td>24</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>27</td>
<td>25</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>28</td>
<td>26</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>29</td>
<td>27</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>30</td>
<td>28</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>31</td>
<td>29</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>32</td>
<td>30</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>33</td>
<td>31</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>34</td>
<td>32</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>35</td>
<td>33</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>36</td>
<td>34</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>37</td>
<td>35</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>38</td>
<td>36</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>39</td>
<td>37</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>40</td>
<td>38</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>41</td>
<td>39</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>42</td>
<td>40</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>43</td>
<td>41</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>44</td>
<td>42</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>45</td>
<td>43</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>46</td>
<td>44</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>47</td>
<td>45</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>48</td>
<td>46</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>49</td>
<td>47</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>50</td>
<td>48</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>51</td>
<td>49</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>52</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>53</td>
<td>51</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>54</td>
<td>52</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>55</td>
<td>53</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>56</td>
<td>54</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>57</td>
<td>55</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td>58</td>
<td>56</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0000 0.0000 0.0000</td>
</tr>
<tr>
<td></td>
<td>57</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>---</td>
<td>----</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>58</td>
<td>59</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>59</td>
<td>60</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>60</td>
<td>61</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>61</td>
<td>62</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>62</td>
<td>63</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>63</td>
<td>64</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>64</td>
<td>65</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>65</td>
<td>66</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>66</td>
<td>67</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>67</td>
<td>68</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>68</td>
<td>69</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>69</td>
<td>70</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>70</td>
<td>71</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>71</td>
<td>72</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>72</td>
<td>73</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>73</td>
<td>74</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>74</td>
<td>75</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>75</td>
<td>76</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>76</td>
<td>77</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>77</td>
<td>78</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>78</td>
<td>79</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>79</td>
<td>80</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>80</td>
<td>81</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>81</td>
<td>82</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>82</td>
<td>83</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>83</td>
<td>84</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>84</td>
<td>85</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>85</td>
<td>86</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>86</td>
<td>87</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>87</td>
<td>88</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

-1
<table>
<thead>
<tr>
<th></th>
<th>70</th>
<th>1</th>
<th>2</th>
<th>1</th>
<th>7</th>
<th></th>
<th>0.000000</th>
<th>0.000000</th>
<th>0.100E+21</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>64</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>7</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>70</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>7</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>76</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>7</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>82</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>7</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>88</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>7</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>9</td>
<td>32</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>96</td>
<td>21000000</td>
<td>1700000</td>
<td>150000</td>
<td>90000</td>
<td>1.043</td>
<td>0.5400</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>96</td>
<td>940000</td>
<td>940000</td>
<td>560000</td>
<td>1100000</td>
<td>1.000E+00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>1</td>
<td>6</td>
<td>30</td>
<td>60</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>2</td>
<td>50</td>
<td>90</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>21</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

|   |   |   |   |   |   |   |           |           |           |
| 24 |   |   |   |   |           |           |           |           |
| 25 |   |   |   |   |           |           |           |           |
| 26 |   |   |   |   |           |           |           |           |
| 27 |   |   |   |   |           |           |           |           |
| 28 |   |   |   |   |           |           |           |           |
| 29 |   |   |   |   |           |           |           |           |
| 30 |   |   |   |   |           |           |           |           |
| 31 |   |   |   |   |           |           |           |           |
| 32 |   |   |   |   |           |           |           |           |
| 33 |   |   |   |   |           |           |           |           |
| 34 |   |   |   |   |           |           |           |           |
| 35 |   |   |   |   |           |           |           |           |
| 36 |   |   |   |   |           |           |           |           |
| 37 |   |   |   |   |           |           |           |           |
| 38 |   |   |   |   |           |           |           |           |
| 39 |   |   |   |   |           |           |           |           |
| 40 |   |   |   |   |           |           |           |           |
| 41 |   |   |   |   |           |           |           |           |
| 42 |   |   |   |   |           |           |           |           |
| 43 |   |   |   |   |           |           |           |           |
| 44 |   |   |   |   |           |           |           |           |
| 45 |   |   |   |   |           |           |           |           |
| 46 |   |   |   |   |           |           |           |           |
| 47 |   |   |   |   |           |           |           |           |
| 48 |   |   |   |   |           |           |           |           |
| 49 |   |   |   |   |           |           |           |           |
| 50 |   |   |   |   |           |           |           |           |
| 51 |   |   |   |   |           |           |           |           |
| 52 |   |   |   |   |           |           |           |           |
| 53 |   |   |   |   |           |           |           |           |
| 54 |   |   |   |   |           |           |           |           |
| 55 | 47 | 41 | 11 | 17 | 36 | 40 |           |           |           |
| 56 |   |   |   |   |           |           |           |           |
| 57 |   |   |   |   |           |           |           |           |
| 58 |   |   |   |   |           |           |           |           |
| 59 |   |   |   |   |           |           |           |           |
| 60 |   |   |   |   |           |           |           |           |
| 61 |   |   |   |   |           |           |           |           |
| 62 |   |   |   |   |           |           |           |           |
| 63 |   |   |   |   |           |           |           |           |
| 64 |   |   |   |   |           |           |           |           |
| 65 |   |   |   |   |           |           |           |           |
| 66 |   |   |   |   |           |           |           |           |
| 67 |   |   |   |   |           |           |           |           |
| 68 |   |   |   |   |           |           |           |           |
| 69 |   |   |   |   |           |           |           |           |
| 70 |   |   |   |   |           |           |           |           |
| 71 |   |   |   |   |           |           |           |           |
| 72 |   |   |   |   |           |           |           |           |
| 73 |   |   |   |   |           |           |           |           |
| 74 |   |   |   |   |           |           |           |           |
| 75 |   |   |   |   |           |           |           |           |
| 76 |   |   |   |   |           |           |           |           |
| 77 |   |   |   |   |           |           |           |           |
| 78 |   |   |   |   |           |           |           |           |
| 79 |   |   |   |   |           |           |           |           |
| 80 |   |   |   |   |           |           |           |           |
| 81 |   |   |   |   |           |           |           |           |
| 82 |   |   |   |   |           |           |           |           |
| 83 |   |   |   |   |           |           |           |           |
| 84 |   |   |   |   |           |           |           |           |
| 85 |   |   |   |   |           |           |           |           |
| 86 |   |   |   |   |           |           |           |           |
| 87 |   |   |   |   |           |           |           |           |
| 88 |   |   |   |   |           |           |           |           |
| 89 |   |   |   |   |           |           |           |           |
| 90 |   |   |   |   |           |           |           |           |
| 91 |   |   |   |   |           |           |           |           |
| 92 |   |   |   |   |           |           |           |           |
| 93 |   |   |   |   |           |           |           |           |
| 94 |   |   |   |   |           |           |           |           |
| 95 |   |   |   |   |           |           |           |           |
| 96 |   |   |   |   |           |           |           |           |
| 97 |   |   |   |   |           |           |           |           |
| 98 |   |   |   |   |           |           |           |           |
| 99 |   |   |   |   |           |           |           |           |
| 100|   |   |   |   |           |           |           |           |
# Modifications for Crack Spacing Sequence

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>2</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>2</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>2</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>2</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>15</td>
<td>2</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>16</td>
<td>2</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>21</td>
<td>2</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>22</td>
<td>2</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>27</td>
<td>2</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>29</td>
<td>2</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>5</td>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>14</td>
<td>9</td>
<td>10</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>15</td>
<td>15</td>
<td>16</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>16</td>
<td>21</td>
<td>22</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>17</td>
<td>27</td>
<td>28</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>18</td>
<td>33</td>
<td>34</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>19</td>
<td>39</td>
<td>40</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>20</td>
<td>45</td>
<td>46</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>21</td>
<td>51</td>
<td>52</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>22</td>
<td>57</td>
<td>58</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>23</td>
<td>63</td>
<td>64</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>24</td>
<td>69</td>
<td>70</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>25</td>
<td>75</td>
<td>76</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>26</td>
<td>81</td>
<td>82</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>27</td>
<td>87</td>
<td>88</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>28</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>1</td>
<td>32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>27</td>
<td>28</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>27</td>
<td>29</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>27</td>
<td>28</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>1</td>
<td>32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>21</td>
<td>22</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>21</td>
<td>22</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>21</td>
<td>22</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>57</td>
<td>58</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>57</td>
<td>58</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>57</td>
<td>58</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>1</td>
<td>32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>9999</td>
<td>9999</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>44</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**CONTROL INFORMATION**

- Number of modal points: 
- Number of element types: 
- Number of load cases: 
- Number of frequencies: 
- Analysis code (rigid): 
- Analysis code (static): 
- Analysis code (forced response): 
- Analysis code (spectrum): 
- Analysis code (direct integration): 
- Solution mode (modal): 
- Solution mode (execute): 
- Data check: 
- Number of subspace: 
- Iteration vectors (check): 
- Equations per block: 
- Matrix save flag (modal): 
- Matrix save flag (execute): 
- Modal point input data

<table>
<thead>
<tr>
<th>NODE</th>
<th>BOUNDARY CONDITION CODE</th>
<th>NODE</th>
<th>BOUNDARY CONDITION CODE</th>
<th>NODE</th>
<th>BOUNDARY CONDITION CODE</th>
<th>NODE</th>
<th>BOUNDARY CONDITION CODE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Element Type** = 7.

**Number of Elements** = 20

**Element load case multipliers**

<table>
<thead>
<tr>
<th>CASE(A)</th>
<th>CASE(B)</th>
<th>CASE(C)</th>
<th>CASE(D)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

**Element Node**

<table>
<thead>
<tr>
<th>ELEMENT</th>
<th>NODE</th>
<th>NODES BBYING CONSTRAINT DIRECTION</th>
<th>CODE</th>
<th>CODE</th>
<th>GENERATION</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(N)</td>
<td>(N1) (N2) (N3) (N4) (N5) (N6)</td>
<td>(K1)</td>
<td>(K2)</td>
<td>(K3) (N4)</td>
</tr>
<tr>
<td>1</td>
<td>61</td>
<td>1 1 1 1 1 1 7 1 1 0 0 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>62</td>
<td>1 1 1 1 1 1 7 1 1 0 0 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>63</td>
<td>1 1 1 1 1 1 7 1 1 0 0 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>64</td>
<td>1 1 1 1 1 1 7 1 1 0 0 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>65</td>
<td>1 1 1 1 1 1 7 1 1 0 0 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>66</td>
<td>1 1 1 1 1 1 7 1 1 0 0 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>67</td>
<td>1 1 1 1 1 1 7 1 1 0 0 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>68</td>
<td>1 1 1 1 1 1 7 1 1 0 0 0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### NODE SOLID ELEMENT INPUT DATA

<table>
<thead>
<tr>
<th>ID</th>
<th>D1</th>
<th>D2</th>
<th>D3</th>
<th>D4</th>
<th>D5</th>
<th>D6</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>69</td>
<td>71</td>
<td>72</td>
<td>73</td>
<td>74</td>
<td>75</td>
</tr>
<tr>
<td>01</td>
<td>77</td>
<td>78</td>
<td>79</td>
<td>80</td>
<td>81</td>
<td>82</td>
</tr>
<tr>
<td>02</td>
<td>83</td>
<td>84</td>
<td>85</td>
<td>86</td>
<td>87</td>
<td>88</td>
</tr>
<tr>
<td>03</td>
<td>89</td>
<td>90</td>
<td>91</td>
<td>92</td>
<td>93</td>
<td>94</td>
</tr>
</tbody>
</table>

#### CONTROL INFORMATION

<table>
<thead>
<tr>
<th>ID</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>03</td>
<td>NUMBER OF 21-NODE ELEMENTS</td>
<td>12</td>
</tr>
<tr>
<td>05</td>
<td>NUMBER OF MATERIAL SETS</td>
<td>1</td>
</tr>
<tr>
<td>07</td>
<td>MAXIMUM NUMBER OF MATERIAL SETS</td>
<td>1</td>
</tr>
<tr>
<td>08</td>
<td>TEMPERATURE INPUT POINTS</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>NUMBER OF MATERIAL AXIS ORIENTATION SETS</td>
<td>2</td>
</tr>
<tr>
<td>11</td>
<td>NUMBER OF DISTRIBUTED LOAD SETS</td>
<td>3</td>
</tr>
<tr>
<td>15</td>
<td>MAXIMUM NUMBER OF ELEMENT NODES</td>
<td>3</td>
</tr>
<tr>
<td>17</td>
<td>NUMBER OF STRESS OUTPUT SETS</td>
<td>1</td>
</tr>
<tr>
<td>19</td>
<td>X,Y,Z COORDINATE INTEGRATION ORDER</td>
<td>2</td>
</tr>
<tr>
<td>1A</td>
<td>T,COORDINATE INTEGRATION ORDER</td>
<td>2</td>
</tr>
</tbody>
</table>

#### MATERIAL PROPERTY TABLES
**MATERIAL**

<table>
<thead>
<tr>
<th>Number</th>
<th>Material</th>
<th>Density</th>
<th>Modulus (E)</th>
<th>Poisson's Ratio (ν)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Aluminum</td>
<td>2700.00</td>
<td>70000.00</td>
<td>0.324</td>
</tr>
<tr>
<td>2</td>
<td>Steel</td>
<td>7800.00</td>
<td>200000.00</td>
<td>0.280</td>
</tr>
<tr>
<td>3</td>
<td>Brass</td>
<td>8700.00</td>
<td>100000.00</td>
<td>0.300</td>
</tr>
</tbody>
</table>

**STRESS OUTPUT RESULTS**

<table>
<thead>
<tr>
<th>Set</th>
<th>Point</th>
<th>Point</th>
<th>Point</th>
<th>Point</th>
<th>Point</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

**ELEMENT LOAD CASE MULTIPLIERS**

<table>
<thead>
<tr>
<th>Case</th>
<th>Case</th>
<th>Case</th>
<th>Case</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
</tr>
<tr>
<td>1</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>2</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>3</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

**ELEMENT NODES NODES NODE NODE NODE STRESS NODE GAUSS GAUSS**

<table>
<thead>
<tr>
<th>Number</th>
<th>NODES</th>
<th>NODES</th>
<th>NODES</th>
<th>NODES</th>
<th>NODES</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
</tr>
<tr>
<td>3</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
</tr>
<tr>
<td>4</td>
<td>19</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
</tr>
</tbody>
</table>

**ELEMENT 12 TO 21 NODE NODE ELEMENT DATA**

<table>
<thead>
<tr>
<th>Number</th>
<th>NODES</th>
<th>NODES</th>
<th>NODES</th>
<th>NODES</th>
<th>NODES</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
</tr>
<tr>
<td>3</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
</tr>
<tr>
<td>4</td>
<td>19</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
</tr>
<tr>
<td>NODE</td>
<td>U</td>
<td>V</td>
<td>W</td>
<td>Fx</td>
<td>Fy</td>
</tr>
<tr>
<td>------</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td>33</td>
<td>1.00000E+00</td>
<td>0.1147E-02</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.1314E-12</td>
</tr>
<tr>
<td>220</td>
<td>1.00000E+00</td>
<td>0.0864E-03</td>
<td>0.1237E-03</td>
<td>0.0000E+00</td>
<td>0.0710E-13</td>
</tr>
<tr>
<td>229</td>
<td>0.00000E+00</td>
<td>0.0000E+00</td>
<td>0.1373E-05</td>
<td>0.0000E+00</td>
<td>-1.24E-07</td>
</tr>
<tr>
<td>344</td>
<td>0.00000E+00</td>
<td>0.0000E+00</td>
<td>0.4379E-03</td>
<td>0.0000E+00</td>
<td>-5.36E-01</td>
</tr>
<tr>
<td>360</td>
<td>0.00000E+00</td>
<td>0.0000E+00</td>
<td>0.1407E-02</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
</tr>
<tr>
<td>375</td>
<td>0.00000E+00</td>
<td>0.0000E+00</td>
<td>0.3978E-02</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
</tr>
<tr>
<td>387</td>
<td>0.00000E+00</td>
<td>0.0000E+00</td>
<td>0.1144E-03</td>
<td>0.0000E+00</td>
<td>0.2913E-12</td>
</tr>
<tr>
<td>404</td>
<td>0.00000E+00</td>
<td>0.0000E+00</td>
<td>0.2633E-03</td>
<td>0.0000E+00</td>
<td>0.1421E-13</td>
</tr>
<tr>
<td>405</td>
<td>0.00000E+00</td>
<td>0.0000E+00</td>
<td>0.4449E-02</td>
<td>0.0000E+00</td>
<td>-1.7E-07</td>
</tr>
<tr>
<td>406</td>
<td>0.00000E+00</td>
<td>0.0000E+00</td>
<td>0.4881E-02</td>
<td>0.0000E+00</td>
<td>-1.07E-07</td>
</tr>
<tr>
<td>407</td>
<td>0.00000E+00</td>
<td>0.0000E+00</td>
<td>0.4924E-02</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
</tr>
<tr>
<td>408</td>
<td>0.00000E+00</td>
<td>0.0000E+00</td>
<td>0.7132E-02</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
</tr>
<tr>
<td>409</td>
<td>0.00000E+00</td>
<td>0.0000E+00</td>
<td>0.3978E-02</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
</tr>
<tr>
<td>410</td>
<td>0.00000E+00</td>
<td>0.0000E+00</td>
<td>0.1144E-03</td>
<td>0.0000E+00</td>
<td>0.2913E-12</td>
</tr>
<tr>
<td>411</td>
<td>0.00000E+00</td>
<td>0.0000E+00</td>
<td>0.2633E-03</td>
<td>0.0000E+00</td>
<td>0.1421E-13</td>
</tr>
<tr>
<td>412</td>
<td>0.00000E+00</td>
<td>0.0000E+00</td>
<td>0.4449E-02</td>
<td>0.0000E+00</td>
<td>-1.7E-07</td>
</tr>
<tr>
<td>413</td>
<td>0.00000E+00</td>
<td>0.0000E+00</td>
<td>0.4881E-02</td>
<td>0.0000E+00</td>
<td>-1.07E-07</td>
</tr>
<tr>
<td>414</td>
<td>0.00000E+00</td>
<td>0.0000E+00</td>
<td>0.4924E-02</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>47-0.1827E-03</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>49-0.1976E-03</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>52-0.2245E-03</td>
<td>0.2245E+00</td>
<td>0.2245E+00</td>
<td>0.2245E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>50-0.2649E-03</td>
<td>0.3195E+00</td>
<td>0.3195E+00</td>
<td>0.3195E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>54-0.3238E-03</td>
<td>0.3759E+00</td>
<td>0.3759E+00</td>
<td>0.3759E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>56-0.3855E-03</td>
<td>0.4623E+00</td>
<td>0.4623E+00</td>
<td>0.4623E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>57-0.4354E-03</td>
<td>0.5000E+00</td>
<td>0.5000E+00</td>
<td>0.5000E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>58-0.4644E-03</td>
<td>0.5644E+00</td>
<td>0.5644E+00</td>
<td>0.5644E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>54-0.3739E-03</td>
<td>0.3739E+00</td>
<td>0.3739E+00</td>
<td>0.3739E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>55-0.3532E-03</td>
<td>0.5147E+00</td>
<td>0.5147E+00</td>
<td>0.5147E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>56-0.3277E-03</td>
<td>0.5576E+00</td>
<td>0.5576E+00</td>
<td>0.5576E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>57-0.3443E-03</td>
<td>0.5644E+00</td>
<td>0.5644E+00</td>
<td>0.5644E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>58-0.3771E-03</td>
<td>0.3771E+00</td>
<td>0.3771E+00</td>
<td>0.3771E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>59-0.4034E-03</td>
<td>0.5644E+00</td>
<td>0.5644E+00</td>
<td>0.5644E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>60-0.4234E-03</td>
<td>0.5644E+00</td>
<td>0.5644E+00</td>
<td>0.5644E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>61-0.4600E-03</td>
<td>0.6000E+00</td>
<td>0.6000E+00</td>
<td>0.6000E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>62-0.4600E-03</td>
<td>0.6000E+00</td>
<td>0.6000E+00</td>
<td>0.6000E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>64-0.4900E-03</td>
<td>0.6000E+00</td>
<td>0.6000E+00</td>
<td>0.6000E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>65-0.5000E-03</td>
<td>0.6000E+00</td>
<td>0.6000E+00</td>
<td>0.6000E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>66-0.1003E-03</td>
<td>0.1003E+00</td>
<td>0.1003E+00</td>
<td>0.1003E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>67-0.1003E-03</td>
<td>0.1003E+00</td>
<td>0.1003E+00</td>
<td>0.1003E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>68-0.1004E-03</td>
<td>0.1004E+00</td>
<td>0.1004E+00</td>
<td>0.1004E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>69-0.1004E-03</td>
<td>0.1004E+00</td>
<td>0.1004E+00</td>
<td>0.1004E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>70-0.1004E-03</td>
<td>0.1004E+00</td>
<td>0.1004E+00</td>
<td>0.1004E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>71-0.1004E-03</td>
<td>0.1004E+00</td>
<td>0.1004E+00</td>
<td>0.1004E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>72-0.1004E-03</td>
<td>0.1004E+00</td>
<td>0.1004E+00</td>
<td>0.1004E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>73-0.1004E-03</td>
<td>0.1004E+00</td>
<td>0.1004E+00</td>
<td>0.1004E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>74-0.1004E-03</td>
<td>0.1004E+00</td>
<td>0.1004E+00</td>
<td>0.1004E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>75-0.1004E-03</td>
<td>0.1004E+00</td>
<td>0.1004E+00</td>
<td>0.1004E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>76-0.1004E-03</td>
<td>0.1004E+00</td>
<td>0.1004E+00</td>
<td>0.1004E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>77-0.1004E-03</td>
<td>0.1004E+00</td>
<td>0.1004E+00</td>
<td>0.1004E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>78-0.1004E-03</td>
<td>0.1004E+00</td>
<td>0.1004E+00</td>
<td>0.1004E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>79-0.1004E-03</td>
<td>0.1004E+00</td>
<td>0.1004E+00</td>
<td>0.1004E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>80-0.1004E-03</td>
<td>0.1004E+00</td>
<td>0.1004E+00</td>
<td>0.1004E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>81-0.1004E-03</td>
<td>0.1004E+00</td>
<td>0.1004E+00</td>
<td>0.1004E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>82-0.1004E-03</td>
<td>0.1004E+00</td>
<td>0.1004E+00</td>
<td>0.1004E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>83-0.1004E-03</td>
<td>0.1004E+00</td>
<td>0.1004E+00</td>
<td>0.1004E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>84-0.1004E-03</td>
<td>0.1004E+00</td>
<td>0.1004E+00</td>
<td>0.1004E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>85-0.1004E-03</td>
<td>0.1004E+00</td>
<td>0.1004E+00</td>
<td>0.1004E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>86-0.1004E-03</td>
<td>0.1004E+00</td>
<td>0.1004E+00</td>
<td>0.1004E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>87-0.1004E-03</td>
<td>0.1004E+00</td>
<td>0.1004E+00</td>
<td>0.1004E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>88-0.1004E-03</td>
<td>0.1004E+00</td>
<td>0.1004E+00</td>
<td>0.1004E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>89-0.1004E-03</td>
<td>0.1004E+00</td>
<td>0.1004E+00</td>
<td>0.1004E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>90-0.1004E-03</td>
<td>0.1004E+00</td>
<td>0.1004E+00</td>
<td>0.1004E+00</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: The values in the table represent numerical data, possibly related to a scientific or engineering context.
437ti~ss

S1
4 i2

STEP 1

2

ss~

AAAAAAAAAAA AAAAAAA AAAJA. AAA AAAA k AA

AAA

FORCES lo SULEG-----NODAL DISPLACEMENTS ivli
A)
AAAAAAAAAAAAA*AAAAAkAAAA AAAAA. 0

A

1 OOOOOE000.OOOOE-O.Ov'33S-i0.3000E+00 0.0 00E+O3-0.0000UE-13
I504

iOOOOE0 0.00000Et3Q-J.i5094L-)30.03)E0

X',C'E0-.16E

?5.4605D 0.ll156IE-02 0.COOOOEtOO 0.14211E-13 0.11102E-12 0.O0000E+00
30530040.89754E-03-0.21i20E-U3 0.35'527E-14 0.22560E-12-0.16342E-12
9-0.26748E-04 0.00000E,00-0.43272E-03 i5.531
-2546.6
77.612
10-0.2674BE-04 .OOOtO042E-9~5.531
-10733.
_77.61.1
11 0.24131E-04 0.0OOOOE+o-0.64+u42E-o3-o.49r38E-13 0.00000E400 0.85265E-13
12 0.17142E-04 0OOOtO004-jU16E130.00000E+00-0.56843E-13
13-0.66606E-04 0.11865E-02 G.0MOQEt0v-0.91432E-13 0.58620E-13 0.00000E+00
14-3.66621E-04 0.9168GE-03-0.34559E-03 0.0J00OEfOO 0.17764E-14-Q.19185E-12
1'5-0.14033E-03 0.00000E+00-0.79662-03 -606.80
-25I64.5
-666.7'j
666.75
-9013.7
1-0.14033E-03 0.OOOOOEtO-0.72966E-03 606.60
17 0.17186E-03 0.O000EtOO-0.10019E-62-0.17408E-12 0.00OOOE+00-0.48317E-12
19 0.67653E-03 0.OOOOOE'o-0.12%OE-O:-0).13500OE-12 O.OOOOOE+00 0.2131GE-12
1)-3 .84655E-04 0.1373JE-02 O.OOQOOE+00 0.39080E-13 0.36415E-l12 0.OOOOOE+00
0.24158E-12e
~
-. 6202E-04 .17E--022E-3017E--.21521-.70549E-04 0.00E)-.11E0-230 -13 778.4
-0.1065SE-1:
52-0.13701E-02 0.32522E-02-0.5?535E-G2 0.14211E-13 0.21974E-12-0.10232E-11
.OOOOOE00-0.26432E-11I
J-1
1.2-0.2283SE-03 0.OOOO-.603-2054E1
-'IA
0.j 957E-03 0.00(,
t00-0).57u34E-02-0.710'4E-14 O.OOOQQEtOO O.'6380E-11
11..35-~.1107E030.14984E-02
0.J91-E-13 0.47%:E-13 0.0000()E+00
A-Q.12UH5-'201
E0-.'32-3u80E
0.3730"E-13 0.11269E-12

5JO6
507
CM8
Jut)
510
511
512I
C.13
SI I
j1
J16
111
ft

f

..

a

-

0.1Sb7E03
00000E'0-0.0230-030.101:UE-12 -1498.4

-. 15E1

0 027-20
E-0-0
0.67502E-13 0.29843E-12 0.6,2212E-12
Z'3-.i012E02
.OOOO~OO-.8'213-Oj0.14211E-13 0.OOOOOE*00-0.41496E-11
30).Z9968E-03 0.3OOOEtU0O.85j224E-02 0.3)790E-12 O.OOOOOE+00 C.30127E-11
31 0.00000E40 0.305E-02 O.OOOOQEtOU 0.00000E+0 O .68 E-12 0.OOCOOEt00
0.OOOOOE*00 0.3042,3E-02-0.11j31E-03 00
E0-.74E1-.92E1
-31.994
33 0.00000Et00 0.30750E-02-0.41il-L-03 0.000O0E+00 -433.39
34 O.00000E+00
0.00SE0-.11E0 ~0000E+00 433.39
31.994
31 0.00000E+00 031 E-2O6o4.006' 0.211"14E-11 0.42633E-13
36 0.000GOEiN00 .30065E-02-0.8G547E-03 0.000006*00 0.21M7E-12-0.32685E-12
37090 -04 0.30743L-02 0.0000o~f00 0.98E1-.4361
.00O000
39-0.97256E-04 0..050E-02>0.1037E-03 0.10431E-12 0.44?G46-12-0.67502E642
39-0.964256E-04 0366-2.479OJ79.G05
-982
-4.6
4-0.96425E-04 0.3061GE-OZ-0.4137-E-03 -7),601
95..0
46.563
:-.l69E

-9
536
1131
S32
J33
524
535i
54u

57
513
53')

~

41-0.8312E6-04 0.300j9E-02-0.6339)E-01-O.SJ26coE-13 0.14992E-11-0.13474E-I2
4~-0777I6-4
03000E-2-08002E-3-O 1597E13 .131709E-11-0.4689GE-i2

*


<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>Modal Point Input</td>
<td>0.00</td>
</tr>
<tr>
<td>004</td>
<td>Element Stiffness Formation</td>
<td>0.00</td>
</tr>
<tr>
<td>007</td>
<td>Modal Load Input</td>
<td>0.00</td>
</tr>
<tr>
<td>003</td>
<td>Total Stiffness Formation</td>
<td>0.00</td>
</tr>
<tr>
<td>009</td>
<td>Static Analysis</td>
<td>0.00</td>
</tr>
<tr>
<td>010</td>
<td>Eigenvalue Extraction</td>
<td>0.00</td>
</tr>
<tr>
<td>001</td>
<td>Forced Response Analysis</td>
<td>0.00</td>
</tr>
<tr>
<td>002</td>
<td>Response Spectrum Analysis</td>
<td>0.00</td>
</tr>
<tr>
<td>003</td>
<td>Step-by-Step Integration</td>
<td>0.00</td>
</tr>
<tr>
<td>025</td>
<td>Total Solution Time</td>
<td>0.00</td>
</tr>
</tbody>
</table>
### OUTPUT FROM KASAR II: Stress Analysis

#### Stress Output Locations

<table>
<thead>
<tr>
<th>121 - Node Solid Element Stress</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Element Load Location</strong></td>
</tr>
<tr>
<td>11</td>
</tr>
<tr>
<td>12</td>
</tr>
<tr>
<td>13</td>
</tr>
<tr>
<td>14</td>
</tr>
<tr>
<td>15</td>
</tr>
<tr>
<td>16</td>
</tr>
<tr>
<td>17</td>
</tr>
<tr>
<td>18</td>
</tr>
<tr>
<td>19</td>
</tr>
<tr>
<td>20</td>
</tr>
<tr>
<td>21</td>
</tr>
<tr>
<td>22</td>
</tr>
<tr>
<td>23</td>
</tr>
<tr>
<td>24</td>
</tr>
<tr>
<td>25</td>
</tr>
<tr>
<td>26</td>
</tr>
<tr>
<td>27</td>
</tr>
<tr>
<td>28</td>
</tr>
<tr>
<td>29</td>
</tr>
<tr>
<td>30</td>
</tr>
<tr>
<td>31</td>
</tr>
<tr>
<td>32</td>
</tr>
</tbody>
</table>

---

#### Energy Release: Crack Propagation Directions

| 0.24920199 | 0.25120199 |

### ENERGY RELEASE: Crack Propagation Directions

---

<table>
<thead>
<tr>
<th>121 - Node Solid Element Stress</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Element Load Location</strong></td>
</tr>
<tr>
<td>33</td>
</tr>
<tr>
<td>34</td>
</tr>
<tr>
<td>35</td>
</tr>
<tr>
<td>36</td>
</tr>
<tr>
<td>37</td>
</tr>
<tr>
<td>38</td>
</tr>
<tr>
<td>39</td>
</tr>
<tr>
<td>40</td>
</tr>
<tr>
<td>41</td>
</tr>
<tr>
<td>42</td>
</tr>
<tr>
<td>43</td>
</tr>
<tr>
<td>44</td>
</tr>
<tr>
<td>45</td>
</tr>
<tr>
<td>46</td>
</tr>
<tr>
<td>47</td>
</tr>
<tr>
<td>48</td>
</tr>
<tr>
<td>49</td>
</tr>
<tr>
<td>50</td>
</tr>
<tr>
<td>51</td>
</tr>
<tr>
<td>52</td>
</tr>
<tr>
<td>53</td>
</tr>
<tr>
<td>54</td>
</tr>
<tr>
<td>Element</td>
</tr>
<tr>
<td>---------</td>
</tr>
<tr>
<td>01</td>
</tr>
<tr>
<td>02</td>
</tr>
<tr>
<td>03</td>
</tr>
<tr>
<td>04</td>
</tr>
<tr>
<td>05</td>
</tr>
<tr>
<td>06</td>
</tr>
<tr>
<td>07</td>
</tr>
<tr>
<td>08</td>
</tr>
<tr>
<td>09</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>11</td>
</tr>
<tr>
<td>12</td>
</tr>
<tr>
<td>13</td>
</tr>
<tr>
<td>14</td>
</tr>
<tr>
<td>15</td>
</tr>
<tr>
<td>16</td>
</tr>
<tr>
<td>17</td>
</tr>
<tr>
<td>18</td>
</tr>
<tr>
<td>19</td>
</tr>
<tr>
<td>20</td>
</tr>
<tr>
<td>21</td>
</tr>
<tr>
<td>22</td>
</tr>
<tr>
<td>23</td>
</tr>
<tr>
<td>24</td>
</tr>
<tr>
<td>25</td>
</tr>
<tr>
<td>26</td>
</tr>
</tbody>
</table>

--- ENERGY RELEASED in (x, y, z) directions ---

0.645560E+02  1.410608  1.279486
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th>0.4650E+02</th>
<th>0.1774E+05</th>
<th>-0.115167E+03</th>
<th>0.52232E+03</th>
</tr>
</thead>
<tbody>
<tr>
<td>19</td>
<td>1</td>
<td>1</td>
<td>-0.6527E+02</td>
<td>0.146759E+05</td>
<td>0.447546E+02</td>
<td>0.123125E</td>
</tr>
<tr>
<td>20</td>
<td>1</td>
<td>1</td>
<td>0.4064E+02</td>
<td>0.206456E+05</td>
<td>0.784515E+01</td>
<td>-0.84667E+02</td>
</tr>
<tr>
<td>21</td>
<td>1</td>
<td>1</td>
<td>0.4020E+02</td>
<td>0.205723E+05</td>
<td>0.107676E+02</td>
<td>0.11396E</td>
</tr>
<tr>
<td>22</td>
<td>1</td>
<td>1</td>
<td>0.2531E+02</td>
<td>0.116384E+05</td>
<td>0.434294E+02</td>
<td>0.102231E</td>
</tr>
<tr>
<td>23</td>
<td>1</td>
<td>1</td>
<td>0.4251E+02</td>
<td>0.414075E+05</td>
<td>0.722564E+01</td>
<td>0.723111E</td>
</tr>
<tr>
<td>24</td>
<td>1</td>
<td>1</td>
<td>0.4538E+02</td>
<td>0.213909E+05</td>
<td>-0.11839E+02</td>
<td>-0.18302E+02</td>
</tr>
<tr>
<td>25</td>
<td>1</td>
<td>1</td>
<td>0.6509E+02</td>
<td>0.115680E+05</td>
<td>0.260462E+02</td>
<td>-0.10752E+02</td>
</tr>
<tr>
<td>26</td>
<td>1</td>
<td>1</td>
<td>0.4117E+02</td>
<td>0.217931E+05</td>
<td>0.185557E+02</td>
<td>-0.15943E+02</td>
</tr>
<tr>
<td>27</td>
<td>1</td>
<td>1</td>
<td>0.4020E+02</td>
<td>0.205723E+05</td>
<td>0.333374E+02</td>
<td>-0.90096E+01</td>
</tr>
<tr>
<td>28</td>
<td>1</td>
<td>1</td>
<td>0.4213E+02</td>
<td>0.210839E+05</td>
<td>0.288468E+01</td>
<td>-0.113331E</td>
</tr>
<tr>
<td>29</td>
<td>1</td>
<td>1</td>
<td>0.4009E+02</td>
<td>0.210839E+05</td>
<td>0.260223E+01</td>
<td>-0.15857E+02</td>
</tr>
<tr>
<td>30</td>
<td>1</td>
<td>1</td>
<td>0.6509E+02</td>
<td>0.115680E+05</td>
<td>-0.206349E+01</td>
<td>-0.540602E+01</td>
</tr>
<tr>
<td>31</td>
<td>1</td>
<td>1</td>
<td>0.4344E+02</td>
<td>0.205109E+05</td>
<td>0.288468E+01</td>
<td>-0.113331E</td>
</tr>
<tr>
<td>32</td>
<td>1</td>
<td>1</td>
<td>0.1977E+02</td>
<td>0.204226E+05</td>
<td>-0.650713E+02</td>
<td>-0.976097E+02</td>
</tr>
</tbody>
</table>

**STATIC SOLUTION FINE LOG**

- **EQUATION SOLUTION** = 0.00
- **DISPLACEMENT OUTPUT** = 0.00
- **STRESS RECOVERY** = 0.00
END
DATE
FILMED
DTIC
July 88