

MICROCOPY RESOLUTION TEST CHART

U.S. GOVERNMENT PRINTING OFFICE: 1963 O 454902

AD-A192 952

DTIC FILE 002

AFOSR-TR-88-0280

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

DTIC  
SELECTED  
MAR 28 1988  
S H D

88 3 28 153

UNCLASSIFIED

UNCLASSIFIED

ADA192952

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified		1b. RESTRICTIVE MARKINGS	
2a. SECURITY CLASSIFICATION AUTHORITY		3. DISTRIBUTION/AVAILABILITY OF REPORT A <b>Approved for public release, distribution unlimited</b>	
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE		4. PERFORMING ORGANIZATION REPORT NUMBER(S)	
5. MONITORING ORGANIZATION REPORT NUMBER(S) <b>AFOSR-TR. 88-0280</b>		6a. NAME OF PERFORMING ORGANIZATION Drexel University	
6b. OFFICE SYMBOL (If applicable)		7a. NAME OF MONITORING ORGANIZATION AFOSR/NA	
6c. ADDRESS (City, State, and ZIP Code) 32nd and Chestnut Sts. Philadelphia, Pa. 19104		7b. ADDRESS (City, State, and ZIP Code) Build 410 Bolling AFB DC 20332-6448	
6d. NAME OF FUNDING/SPONSORING ORGANIZATION AFOSR/NA		8b. OFFICE SYMBOL (If applicable) NH	
9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER AFOSR-84-0334		10. SOURCE OF FUNDING NUMBERS	
6c. ADDRESS (City, State, and ZIP Code) Build. 410 Bolling AFB DC 20332-6448		PROGRAM ELEMENT NO. 61102 F	PROJECT NO. 2302
		TASK NO. B2	WORK UNIT ACCESSION NO.
11. TITLE (Include Security Classification) "A COMPREHENSIVE STUDY ON DAMAGE TOLERANCE PROPERTIES OF NOTCHED COMPOSITE LAMINATES" (U)			
12. PERSONAL AUTHOR(S) A.S.D. Wang, E.S. Reddy, W. Binenda and Yu Zhong			
13a. TYPE OF REPORT Final Report	13b. TIME COVERED FROM 30/9/84 TO 31/12/87	14. DATE OF REPORT (Year, Month, Day) 25 February 1988	15. PAGE COUNT 260
16. SUPPLEMENTARY NOTATION			
17. COSATI CODES		18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)	
FIELD	GROUP	Graphite-Epoxy laminates, notches, holes, stress concentration, matrix cracks, mixed-modes, crack growth, finite element fracture mechanics	
19. ABSTRACT (Continue on reverse if necessary and identify by block number) 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 ply-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 sublaminar 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.			
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT <input type="checkbox"/> UNCLASSIFIED/DUNLIMITED <input checked="" type="checkbox"/> SAME AS RPT <input type="checkbox"/> DTIC USERS		21. ABSTRACT SECURITY CLASSIFICATION Unclassified	
23a. NAME OF RESPONSIBLE INDIVIDUAL Dr. Carter K. Williams		22b. TELEPHONE (Include Area Code) (301) 762-0423	22c. OFFICE SYMBOL NH

DD FORM 1473, 84 MAR

83 APR edition may be used until exhausted. All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE

U.S. Government Printing Office: 1985-607-047

UNCLASSIFIED



## TABLE OF CONTENTS

<b>FOREWORD .....</b>	<b>1</b>
<b>INTRODUCTION .....</b>	<b>2</b>
Objective of Research	
Theoretical Approach	
Crack Growth Simulation	
Major Tasks Performed	
<b>SPECIFIC TASKS AND RESULTS .....</b>	<b>5</b>
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	
<b>CONCLUSIONS .....</b>	<b>8</b>
<b>REFERENCE</b>	
<b>APPENDIX:</b>	

- 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.
- 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, 1987. pp. 444-457.
- IV "Simulation of Matrix Cracks in Composite Laminates Containing a Small Hole"  
Paper presented at the ASME Winter Annual Meeting, Boston, 1987; Also in Damage Mechanics in Composites, AD-12, ASME, 1987. pp. 83-91.
- V "3 D Finite Element Crack Simulation Code - User's Guide and Source Code"

## 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. Haritos of AFOSR served successively as technical monitors during the course of this research.



<b>Accession For</b>	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution/ _____	
<b>Availability Codes</b>	
Dist	Avail and/or Special
A-1	

## 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 sublaminar 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 sublaminar 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 sublaminar 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 sublaminar 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 forgoing 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 sublaminar 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 sublaminar 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 needed. 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 of 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 sublaminar 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 considerably 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

- [1] Tsai, S. W. and Hahn, H. T., "Introduction to Composite Materials," Technomic Pub., Lancaster, Pa. 1980.
- [2] Wang, A. S. D., "Fracture Mechanics of Sublaminar Cracks in Composite Materials," Composite Technology Review, Vol. 6, 1984, pp. 45-62.
- [3] Wang, A. S. D. and Crossman, F. W., "Some New Results on Edge Effects in Symmetric Composite Laminates," Journal of Composite Materials, Vol. 11, 1977, pp. 92-108.

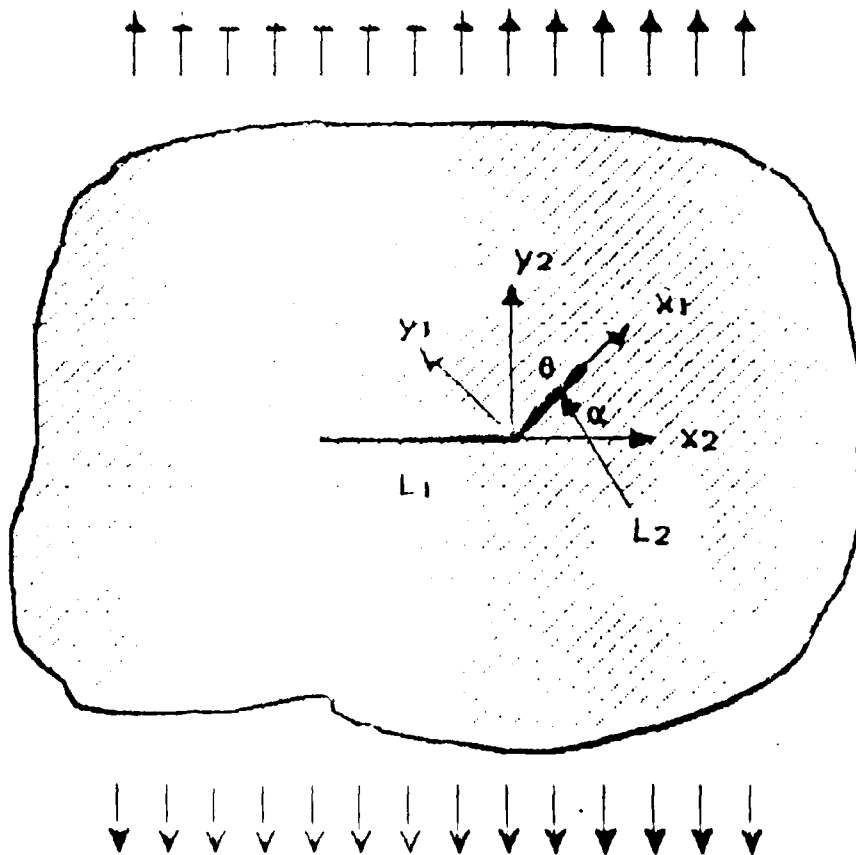


Figure 1. Kink crack in an infinite unidirectional laminate subjected to uniform tension.

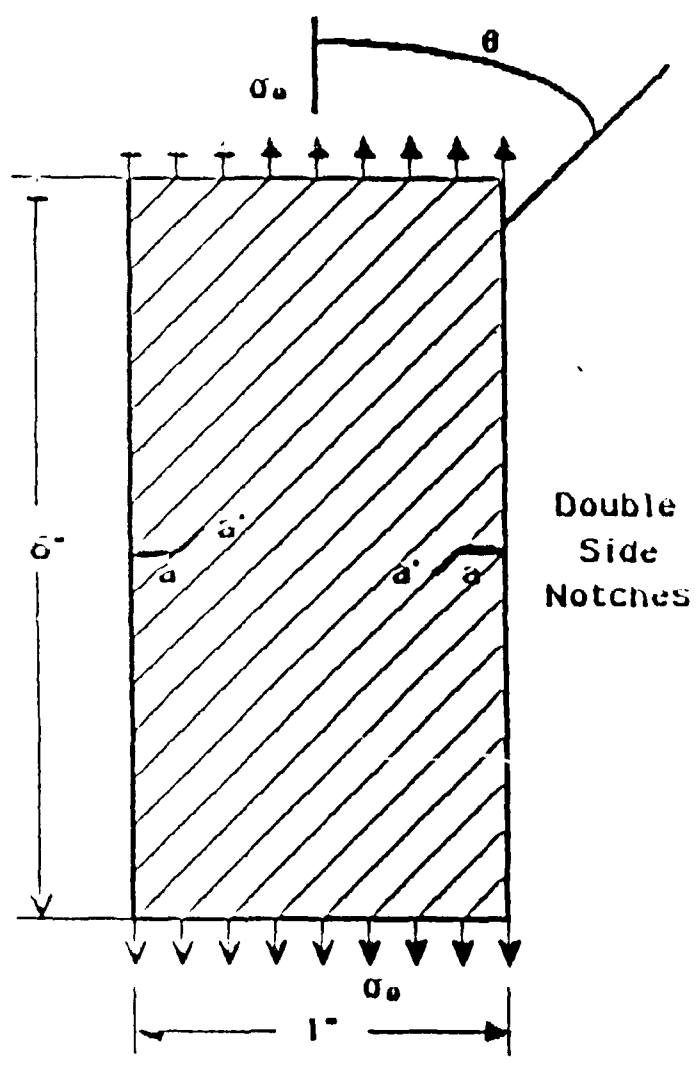


Figure 2. Geometry of test specimen used to establish mixed-mode fracture criterion.

**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 far-field uniform 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  $f(k_I, k_{II}) = k_{eff}$ . In the experiment by Wu [3], who tested notched balsa wood and unidirectional fiber glass reinforced composite plates, the fracture criterion  $(k_I/k_{Ic}) + (k_{II}/k_{IIc})^2 = 1$  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  $(G_T)_c$  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  $G_I$  and  $G_{II}$ , at the tips of the kinked crack are obtained for various values of  $\theta$  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  $(x_1, y_1)$ :

$$\frac{\partial^4 F_1}{\partial x_1^4} + \beta_2 \frac{\partial^4 F_1}{\partial x_1^2 \partial y_1^2} + \beta_1 \frac{\partial^4 F_1}{\partial y_1^4} = 0 \quad (1)$$

where

$$\beta_1 = \frac{a_{11}}{a_{22}}; \quad \beta_2 = \frac{2a_{12} + a_{66}}{a_{22}} \quad (2)$$

and

$$a_{11} = \frac{1}{E_{LL}}; \quad a_{12} = -\frac{\nu_{LT}}{E_{LL}}; \quad a_{22} = \frac{1}{E_{TT}}; \quad a_{66} = \frac{1}{G_{LT}} \quad (3)$$

$E_{LL}$ ,  $E_{TT}$ ,  $G_{LT}$ ,  $\nu_{LT}$  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:

$$F_1(x_1, y_1) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \phi_1(s, y_1) e^{-isx_1} ds \quad (4)$$

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

$$\phi_1(s, y_1) \sim e^{\omega s y_1} \quad (5)$$

so the following characteristic equation is obtained:

$$\beta_1 \omega^4 - \beta_2 \omega^2 + 1 = 0 \quad (6)$$

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:

$$F_1(x_1, y_1^+) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} [A e^{-\omega_1 |s| y_1} + B e^{-\omega_2 |s| y_1}] e^{-isx_1} ds, \quad y_1 > 0 \quad (7)$$

$$F_1(x_1, y_1^-) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} [C e^{\omega_1 |s| y_1} + D e^{\omega_2 |s| y_1}] e^{-isx_1} ds, \quad y_1 < 0$$

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

$$f_1(x_1) = \frac{\partial}{\partial x_1} [u(x_1, 0^+) - u(x_1, 0^-)] \quad x_c < x_1 < x_d \quad (8)$$

$$f_2(x_1) = \frac{\partial}{\partial x_1} [v(x_1, 0^+) - v(x_1, 0^-)] \quad (9)$$

the stresses may be expressed as:

$$\sigma_{x_1 x_1} = \frac{1}{2\pi(\omega_1^2 - \omega_2^2)a_{11}} \int_{x_c}^{x_d} \left\{ \frac{f_1(t_1) \omega_1^3 y_1 + f_2(t_1)(t_1 - x_1) \omega_1}{\omega_1^2 y_1^2 + (t_1 - x_1)^2} - \frac{f_1(t_1) \omega_2^3 y_1 + f_2(t_1)(t_1 - x_1) \omega_2}{\omega_2^2 y_1^2 + (t_1 - x_1)^2} \right\} dt_1 \quad (10)$$

$$\sigma_{y_1 y_1} = \frac{-1}{2\pi(\omega_1^2 - \omega_2^2)a_{11}} \int_{x_c}^{x_d} \left\{ \frac{f_1(t_1) \omega_1 y_1 + f_2(t_1) \frac{t_1 - x_1}{\omega_1}}{\omega_1^2 y_1^2 + (t_1 - x_1)^2} - \frac{f_1(t_1) \omega_2 y_1 + f_2(t_1) \frac{t_1 - x_1}{\omega_2}}{\omega_2^2 y_1^2 + (t_1 - x_1)^2} \right\} dt_1 \quad (11)$$

$$\tau_{x_1 y_1} = \frac{1}{2\pi(\omega_1^2 - \omega_2^2) a_{11}} \int_{x_a}^{x_b} \left\{ \frac{f_1(t_1) \omega_1 (t_1 - x_1) - f_2(t_1) \omega_1 y_1}{\omega_1^2 y_1^2 + (t_1 - x_1)^2} - \frac{f_1(t_1) \omega_2 (t_1 - x_1) - f_2(t_1) \omega_2 y_1}{\omega_2^2 y_1^2 + (t_1 - x_1)^2} \right\} dt_1 \quad (12)$$

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  $F_2(x_2, y_2)$ :

$$\frac{\partial^4 F_2}{\partial x_2^4} + \gamma_1 \frac{\partial^4 F_2}{\partial x_2^2 \partial y_2^2} + \gamma_2 \frac{\partial^4 F_2}{\partial x_2^2 \partial y_2^2} + \gamma_3 \frac{\partial^4 F_2}{\partial x_2 \partial y_2^3} + \gamma_4 \frac{\partial^4 F_2}{\partial y_2^4} = 0 \quad (13)$$

where

$$\begin{aligned} \gamma_1 &= -\frac{2b_{26}}{b_{22}}; & \gamma_2 &= \frac{2b_{12} + b_{66}}{b_{22}}; \\ \gamma_3 &= -\frac{2b_{16}}{b_{22}}; & \gamma_4 &= \frac{b_{11}}{b_{22}}; \end{aligned} \quad (14)$$

and

$$\begin{aligned} b_{11} &= a_{11} \cos^4 \theta + (2a_{12} + a_{66}) \sin^2 \theta \cos^2 \theta + a_{22} \sin^4 \theta \\ b_{22} &= a_{11} \sin^4 \theta + (2a_{12} + a_{66}) \sin^2 \theta \cos^2 \theta + a_{22} \cos^4 \theta \\ b_{12} &= a_{12} + (a_{11} + a_{22} - 2a_{12} - a_{66}) \sin^2 \theta \cos^2 \theta \\ b_{66} &= a_{66} + (a_{11} + a_{22} - 2a_{12} - a_{66}) \sin^2 \theta \cos^2 \theta \\ b_{16} &= [a_{22} \sin^2 \theta - a_{11} \cos^2 \theta + \frac{1}{2}(2a_{12} + a_{66}) \cos 2\theta] \sin 2\theta \\ b_{26} &= [a_{22} \cos^2 \theta - a_{11} \sin^2 \theta - \frac{1}{2}(2a_{12} + a_{66}) \cos 2\theta] \sin 2\theta \end{aligned} \quad (15)$$

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:

$$\sigma_{x_2 y_2} = \frac{1}{2\pi} \int_{x_a}^{x_b} \left[ \frac{R_1 f_3(t_2) - R_2 f_4(t_2)}{y_2(a+ib) + i(t_2 - x_2)} + \frac{R_3 f_4(t_2) - R_4 f_3(t_2)}{y_2(c+id) + i(t_2 - x_2)} + \frac{R_5 f_3(t_2) - R_6 f_4(t_2)}{y_2(a-ib) - i(t_2 - x_2)} + \frac{R_7 f_4(t_2) - R_8 f_3(t_2)}{y_2(c-id) - i(t_2 - x_2)} \right] dt_2 \quad (16)$$

$$\sigma_{y_2 y_2} = \frac{1}{2\pi} \int_{x_a}^{x_b} \left[ \frac{R_9 f_3(t_2) - R_{10} f_4(t_2)}{y_2(a+ib) + i(t_2 - x_2)} + \frac{R_{11} f_4(t_2) - R_{12} f_3(t_2)}{y_2(c+id) + i(t_2 - x_2)} + \frac{R_{13} f_3(t_2) - R_{14} f_4(t_2)}{y_2(a-ib) - i(t_2 - x_2)} + \frac{R_{15} f_4(t_2) - R_{16} f_3(t_2)}{y_2(c-id) - i(t_2 - x_2)} \right] dt_2 \quad (17)$$

$$\tau_{x_2 y_2} = \frac{1}{2\pi} \int_{x_a}^{x_b} \left[ \frac{R_{17} f_3(t_2) - R_{18} f_4(t_2)}{y_2(a+ib) + i(t_2 - x_2)} + \frac{R_{19} f_4(t_2) - R_{20} f_3(t_2)}{y_2(c+id) + i(t_2 - x_2)} + \frac{R_{21} f_3(t_2) - R_{22} f_4(t_2)}{y_2(a-ib) - i(t_2 - x_2)} + \frac{R_{23} f_4(t_2) - R_{24} f_3(t_2)}{y_2(c-id) - i(t_2 - x_2)} \right] dt_2 \quad (18)$$

where

$$f_3(x_2) = \frac{\partial}{\partial x_2} [u(x_2, 0^+) - u(x_2, 0^-)] \quad (19)$$

$$x_a < x_2 < x_b$$

$$f_4(x_2) = \frac{\partial}{\partial x_2} [v(x_2, 0^+) - v(x_2, 0^-)] \quad (20)$$

and  $R_i \quad i=1,2,\dots,24$  are given in [16]. Here it must be noted that the formulation leading to expressions (16)-(18) is quite lengthy and tedious. The intermediate steps can be found in [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:

$$x_2 = x_1 \cos \theta - y_1 \sin \theta \quad (21)$$

$$y_2 = x_1 \sin \theta + y_1 \cos \theta$$

or

$$x_1 = x_2 \cos \theta + y_2 \sin \theta$$

$$y_1 = -x_2 \sin \theta + y_2 \cos \theta \quad (22)$$

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:

$$\sigma_{y_2 y_2}^T = -\sigma_0 \quad (23)$$

$$x_a < x_2 < x_b$$

$$\tau_{x_2 y_2}^T = 0$$

and

$$\begin{aligned} \sigma_{y_1 y_1}^T &= -\sigma_0 \cos^2 \theta \\ \tau_{x_1 y_1}^T &= -\sigma_0 \sin \theta \cos \theta \end{aligned} \quad x_c < x_1 < x_d \quad (24)$$

By means of a normalization procedure by substituting the following:

$$\begin{aligned} t_2 &= \frac{\tau_2(x_b - x_a)}{2} + \frac{x_b + x_a}{2} \\ x_2 &= \frac{s_2(x_b - x_a)}{2} + \frac{x_b + x_a}{2} \quad -1 < \tau_2, s_2 < 1 \\ dt_2 &= \frac{x_b - x_a}{2} d\tau_2 \end{aligned} \quad (25)$$

and

$$\begin{aligned} t_1 &= \frac{\tau_1(x_d - x_c)}{2} + \frac{x_d + x_c}{2} \\ x_1 &= \frac{s_1(x_d - x_c)}{2} + \frac{x_d + x_c}{2} \quad -1 < \tau_1, s_1 < 1 \\ dt_1 &= \frac{x_d - x_c}{2} d\tau_1 \end{aligned} \quad (26)$$

(23) and (24) lead to the following system of Cauchy type singular integral equations:

$$\begin{aligned} C_{12} \int_{-1}^1 \frac{f_2(\tau_1)}{\tau_1 - s_1} d\tau_1 + \int_{-1}^1 K_{13} f_3(\tau_2) d\tau_2 + \int_{-1}^1 K_{14} f_4(\tau_2) d\tau_2 \\ = -\sigma_0 \cos^2 \theta \end{aligned} \quad (27)$$

$$\begin{aligned} C_{21} \int_{-1}^1 \frac{f_2(\tau_1)}{\tau_1 - s_1} d\tau_1 + \int_{-1}^1 K_{23} f_3(\tau_2) d\tau_2 + \int_{-1}^1 K_{24} f_4(\tau_2) d\tau_2 \\ = -\sigma_0 \sin \theta \cos \theta \end{aligned} \quad (28)$$

$$\begin{aligned} C_{33} \int_{-1}^1 \frac{f_3(\tau_2)}{\tau_2 - s_2} d\tau_2 + C_{34} \int_{-1}^1 \frac{f_4(\tau_2)}{\tau_2 - s_2} d\tau_2 + \int_{-1}^1 K_{31} f_1(\tau_1) d\tau_1 \\ + \int_{-1}^1 K_{32} f_2(\tau_1) d\tau_1 = -\sigma_0 \end{aligned} \quad (29)$$

$$\begin{aligned} C_{43} \int_{-1}^1 \frac{f_3(\tau_2)}{\tau_2 - s_2} d\tau_2 + C_{44} \int_{-1}^1 \frac{f_4(\tau_2)}{\tau_2 - s_2} d\tau_2 + \int_{-1}^1 K_{41} f_1(\tau_1) d\tau_1 \\ + \int_{-1}^1 K_{42} f_2(\tau_1) d\tau_1 = 0 \end{aligned} \quad (30)$$

These equations must be solved with the following single-valuedness conditions which complete the formulation of

the problem:

$$\int_{-1}^1 f_1(\tau_1) d\tau_1 = 0 \quad (31)$$

$$\int_{-1}^1 f_3(\tau_2) d\tau_2 = 0 \quad (32)$$

$$\int_{-1}^1 f_2(\tau_1) d\tau_1 = 0 \quad (33)$$

$$\int_{-1}^1 f_4(\tau_2) d\tau_2 = 0 \quad (34)$$

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  $\beta$  at the apex. The singularity  $\beta$  can be derived by requiring the displacements of common end to match what giving the following transcendental characteristic equation:

$$\begin{aligned} -\pi^4 \cos^4 \pi \beta C_{12} C_{21} C_{33} C_{44} - \frac{\pi^2}{4} \cos^2 \pi \beta C_{12} C_{34} A_{23} A_{41} \\ - \frac{\pi^2}{4} \cos^2 \pi \beta C_{12} C_{43} A_{24} A_{31} + \frac{\pi^2}{4} \cos^2 \pi \beta C_{12} C_{33} A_{24} A_{41} \\ + \frac{\pi^2}{4} \cos^2 \pi \beta C_{12} C_{44} A_{23} A_{31} + \pi^4 \cos^4 \pi \beta C_{12} C_{21} C_{33} C_{44} \\ + \frac{\pi^2}{4} \cos^2 \pi \beta C_{21} C_{44} A_{13} A_{32} + \frac{1}{16} A_{13} A_{24} A_{31} \\ - \frac{1}{16} A_{13} A_{24} A_{32} A_{41} - \frac{\pi^2}{4} \cos^2 \pi \beta C_{21} C_{34} A_{13} A_{42} \\ - \frac{\pi^2}{4} \cos^2 \pi \beta C_{21} C_{43} A_{14} A_{32} - \frac{1}{16} A_{14} A_{23} A_{31} A_{42} \\ + \frac{1}{16} A_{14} A_{23} A_{32} A_{41} + \frac{\pi^2}{4} \cos^2 \pi \beta C_{21} C_{33} A_{14} A_{42} = 0 \end{aligned} \quad (35)$$



For 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:

$$x_d \int_{-1}^1 f_2(\tau_1) d\tau_1 = x_a \int_{-1}^1 [f_3(\tau_2) \sin\theta - f_4(\tau_2) \cos\theta] d\tau_2 \quad (36)$$

$$x_d \int_{-1}^1 f_1(\tau_1) d\tau_1 = -x_a \int_{-1}^1 [f_3(\tau_2) \cos\theta + f_4(\tau_2) \sin\theta] d\tau_2 \quad (37)$$

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.

### The Strain Energy Release Rate

From the fracture point of view, perhaps the most important physical quantity is the strain energy release rate  $G$ . Using the usual definition [26], it can be written that:

$$G = \frac{d}{da} (U - V) \quad (38)$$

at  $x=x_d$ , we may write:

$$dU - dV = \frac{1}{2} \int_{x_d}^{x_d+da} \left[ \sigma_{y_1 y_1}(x_1, 0) [v(x_1 - da, 0^+) - v(x_1 - da, 0^-)] + \tau_{x_1 y_1}(x_1, 0) [u(x_1 - da, 0^+) - u(x_1 - da, 0^-)] \right] dx_1 \quad (39)$$

The expressions of normal and shear stresses can be found using definition for stress intensity factors. Thus,

$$\sigma_{y_1 y_1}(x_1, 0) = \frac{k_1(x_d)}{\sqrt{2(x_1 - x_d)}} + \text{higher order terms} \quad (40)$$

$$\tau_{x_1 y_1}(x_1, 0) = \frac{k_2(x_d)}{\sqrt{2(x_1 - x_d)}} + \text{higher order terms} \quad (41)$$

To obtain the asymptotic expressions for  $[u(x_1, 0^+) - u(x_1, 0^-)]$  and  $[v(x_1, 0^+) - v(x_1, 0^-)]$ , we can use equations (8) and (9). Following the procedure of derivation as in [27,28], it can be shown that :

$$G_I = \frac{1}{4} \frac{k_1^2(x_d)}{C_{12}} \quad (42)$$

$$G_{II} = \frac{1}{4} \frac{k_2^2(x_d)}{C_{21}} \quad (43)$$

$$\text{and} \quad G = G_I + G_{II} \quad (44)$$

For all the details one may refer to [16].

### 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 and 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 ( $\beta \geq 0$ ) 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 stresses 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 not a monotonous 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 mode-II component of the strain energy release rate.

#### REFERENCES

- [1] F. Erdogan and G.C. Sih, "On the Crack Extension in Plates Under Plane Loading and Transverse Shear", *Trans. ASME, J. Basic Engng.*, Vol.85, pp. 519-527, 1963.
- [2] J.G. Williams and P.D. Ewing, "Fracture Under Complex Stress - The Angled Crack Problem", *Int. J. Fracture*, Vol.9, pp. 441-445, 1972.
- [3] E.M. Wu, "Application of Fracture Mechanics to Anisotropic Plates", *Trans. ASME, J. Appl. Mech.*, Vol. 34, pp. 967-974, 1967.
- [4] B. Cotterell, "The Influence of the Stress Distribution at the Tip of a Crack", *Int. J. Fracture*, Vol. 2, pp. 526-533, 1966.
- [5] G.C. Sih, "Some Basic Problems in Fracture Mechanics and New Concepts", *Engng. Fracture Mech.*, Vol. 5, pp. 365-377, 1973.
- [6] G.C. Sih, "Strain Energy Density Factor Applied to Mixed Mode Crack Problems", *Int. J. Fracture*, Vol. 10, pp. 305-321, 1974.
- [7] A.S.D. Wang and F.W. Crossman, "Initiation and Growth of Transverse Cracks and Edge Delamination in Composite Laminates, Part 1, An Energy Method", *J. Composite Materials*, Vol. 4, pp. 71-87, 1980.
- [8] F.W. Crossman and W.J. Warren, A.S.D. Wang and G.E. Law, "Initiation and Growth of Transverse Cracks and Edge Delamination in Composite Laminates, Part 2, Experimental Correlation", *J. Composite Materials*, Vol. 14, pp. 88-106, 1980.
- [9] A.S.D. Wang, "Growth Mechanisms of Transverse Cracks and Ply Delamination in Composite Laminates", in *Advances in Composite Materials*, Proc. ICCM-III, pp. 170-182, 1980.
- [10] A.S.D. Wang, "Fracture Mechanics of Sub-Laminate Cracks in Composite Laminates", Proc. AGARD Struct. Matls. Specialist Meeting, London, 1983.
- [11] A.S.D. Wang, N.N. Kishore and W. Feng, "On Mixed Mode Fracture in Off-Axis Unidirectional Graphite Epoxy Composites", ICCM-4, Tokyo, Japan, 1982.
- [12] F. Delale and F. Erdogan, "The Problem of Internal and Edge Cracks in an Orthotropic Strip", *ASME, Journal of Applied Mechanics*, Vol. 44, pp.237-242, 1977.
- [13] F. Delale, I. Bakirtas and F. Erdogan, "The Problem of an Inclined Crack in an Orthotropic Strip", *ASME, Journal of Applied Mechanics*, Vol. 46, pp. 90-96, 1979.
- [14] F. Delale and F. Erdogan, "Bonded Orthotropic Strips with Cracks", *Int. J. of Fracture*, Vol. 15, pp.343-364, 1979.
- [15] F. Delale, "Fracture Analysis of Notched Composites", *ASME, Journal of Engineering Materials and Technology*, Vol. 109, pp. 27-35, 1987.
- [16] A.S.D. Wang, W. Binienda, Yu Zhong and E.S. Reddy, "Mixed Mode Fracture of Uniaxial Fiber Reinforced Composites", NADC Technical Report, April, 1987.
- [17] F. Erdogan, "Mixed Boundary-Value Problems in Mechanics", *Mechanics Today*, edited by S. Nemat-Nasser, Vol. 4, pp. 1-86, Pergamon Press, Oxford, 1978.
- [18] P.S. Theocaris and N.I. Ioakimidis, "Numerical Integration Methods for the Solution of Singular Integral Equations", *Quarterly of Appl. Mathematics*, pp. 173-183, 1977.
- [19] M.L. Williams, "Stress Singularities Resulting from Various Boundary Conditions in Angular Corners of Plates in Extension", *J. Appl. Mech.*, Vol. 19, pp. 526-528, 1952.
- [20] M.C. Kuo and D.B. Bogy, "Plane Solutions for the Displacement and Traction-Displacement Problems for Anisotropic Elastic Wedges", *J. Appl. Mech.*, Vol. 41, pp. 197-202, 1974.
- [21] M.C. Kuo and D.B. Bogy, "Plane Solution for Traction Problems on Orthotropic Unsymmetrical Wedges and Symmetrically Twinned Wedges", *J. Appl. Mech.*, Vol. 41, pp. 203-208, 1974.
- [22] N.I. Ioakimidis and P.S. Theocaris, "The Numerical Evaluation of a Class of Generalized Stress Intensity Factors by Use of the Lobatto-Jacobi Numerical Integration Rule", *Int. J. of Fracture*, Vol. 14, pp. 469-484, 1978.
- [23] N.I. Ioakimidis and P.S. Theocaris, "On the Selection of Collocation Points for the Numerical Solution of Singular Integral Equations with Generalized Kernels Appearing in Elasticity Problems", *Computers & Structures*, Vol. 11, pp. 289-295, 1980.
- [24] G.D. Gupta, "Strain Energy Release Rate for Mixed Mode Crack Problem", *ASME, 76-WA/PVP-7*, 1977.
- [25] S.N. Chatterjee, "The Stress Field in the Neighborhood of a Branched Crack in an Infinite Elastic Sheet", *Int. J. of Solids & Structures*, Vol. 11, pp. 521-538, 1975.
- [26] G.P. Cherepanov, "Mechanics of Brittle Fracture",

McGraw-Hill, N.Y., 1979.

[27] F. Delale, "Mode III Fracture of Bonded Nonhomogeneous Materials", Engineering Fracture Mechanics, Vol. 22, pp. 213-226, 1985.

[28] S. Krenk, "Influence of Transverse Shear on an Axial Crack in a Cylindrical Shell", Int. J. Fracture, Vol. 14, pp. 123-143, 1978.

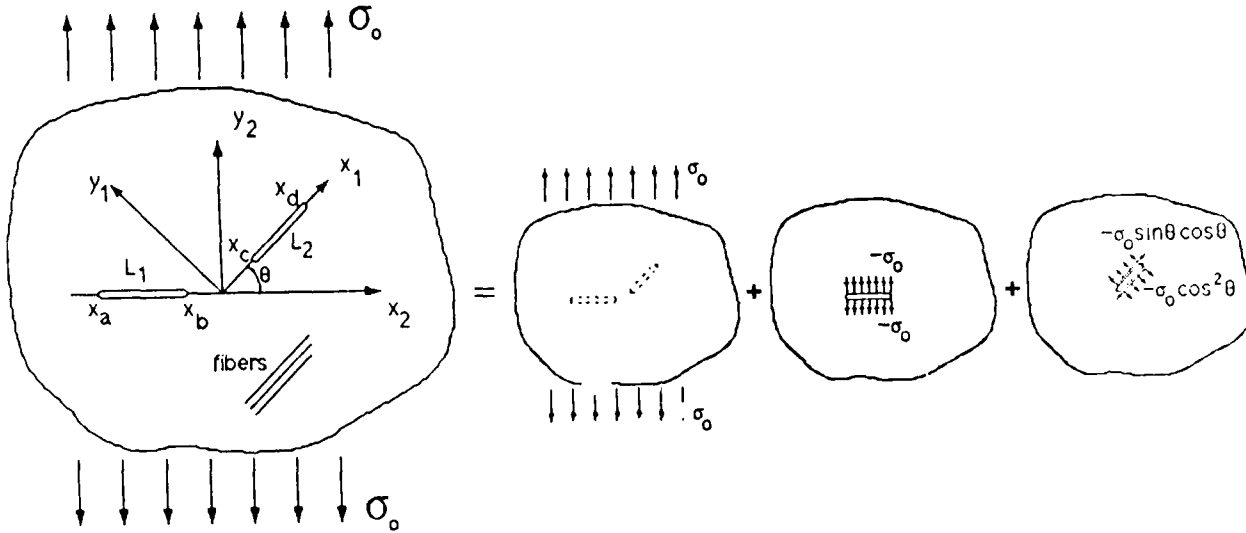


Figure 1. Superposition Scheme for the Infinite Composite Plate with Two Embedded Cracks.

Table 1. Material constants for orthotropic plate.

$E_L$	21.08 e+06 psi.
$E_T$	1.5 e+06 psi.
$G_{LT}$	0.98 e+06 psi.
$\nu_{LT}$	0.3

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

$\theta$		$k1(a)$	$k2(a)$	$k1(d)$	$k2(d)$
30°	[24]	1.3559	0.0327	1.0873	0.6833
	[25]	1.3508	0.0325	1.0830	0.6804
	Present	1.3421	0.0328	1.0949	0.6855
45°	[24]	1.2902	0.0211	0.7463	0.8405
	[25]	1.2887	0.0208	0.7438	0.8377
	Present	1.2732	0.0217	0.7546	0.8450
60°	[24]	1.2221	-0.0109	0.3900	0.8319
	[25]	1.2194	-0.0116	0.3822	0.8292
	Present	1.2082	-0.0108	0.3941	0.8350

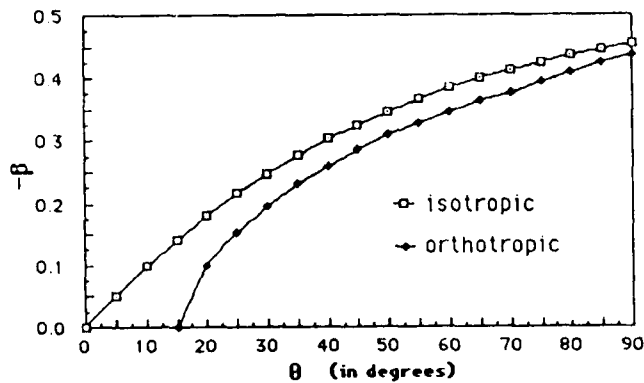


Figure 2. Variation of the Stress Singularity Power ( $-\beta$ ) with the Angle  $\theta$ .

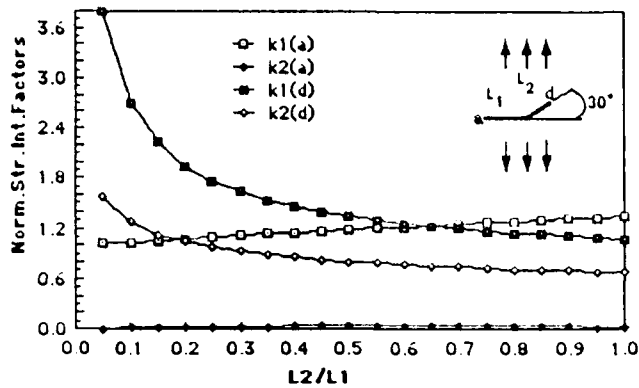


Figure 3. Variation of the Normalized Stress Intensity Factors with  $L_2/L_1$ . Isotropic Case.

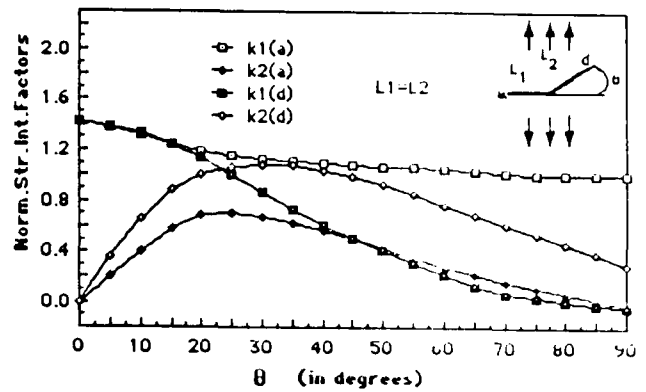


Figure 6. Variation of the Normalized Stress Intensity Factors with the angle  $\theta$ . Orthotropic Case.

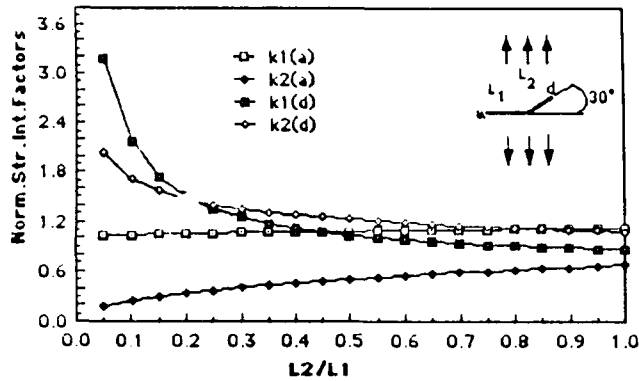


Figure 4. Variation of the Normalized Stress Intensity Factors with  $L_2/L_1$ . Orthotropic Case.

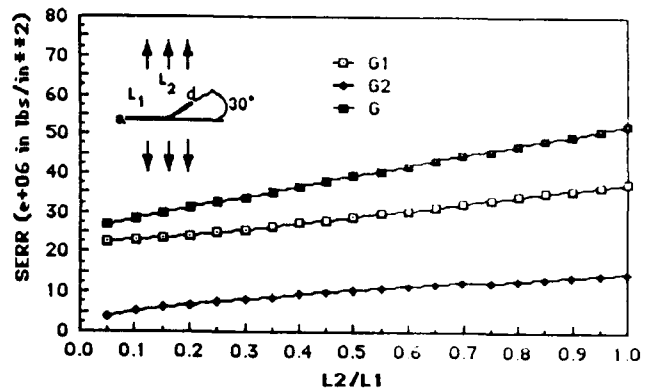


Figure 7. Variation of the Strain Energy Release Rates with  $L_2/L_1$ . Isotropic Case.

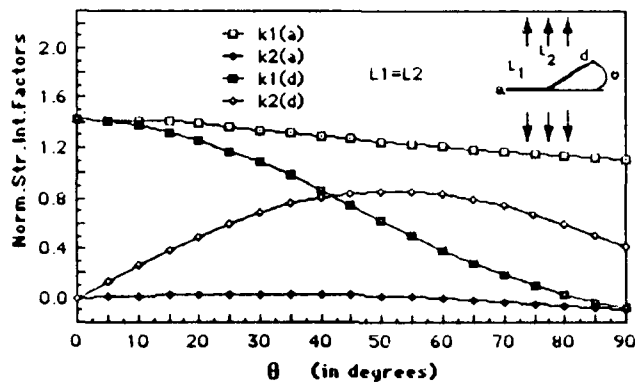


Figure 5. Variation of the Normalized Stress Intensity Factors with the angle  $\theta$ . Isotropic Case.

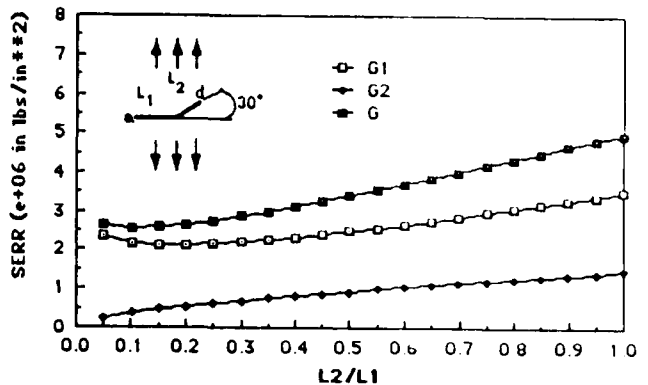


Figure 8. Variation of the Strain Energy Release Rates with  $L_2/L_1$ . Orthotropic Case.

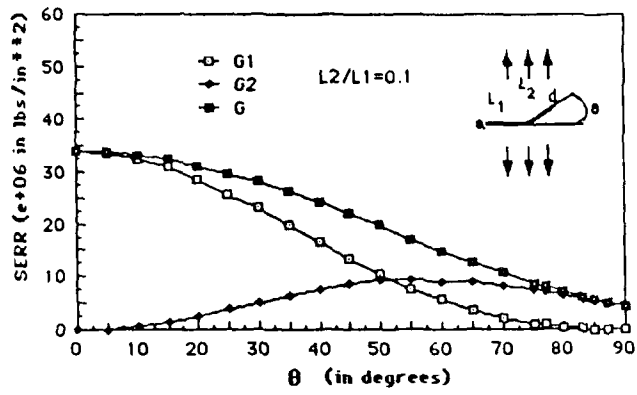


Figure 9. Mixed-Mode Strain Energy Release Rates at the Kink Tip as a Function of Kink Angle  $\theta$ . Isotropic Case. ( $e_x=1, L_2 \rightarrow 0$ ).

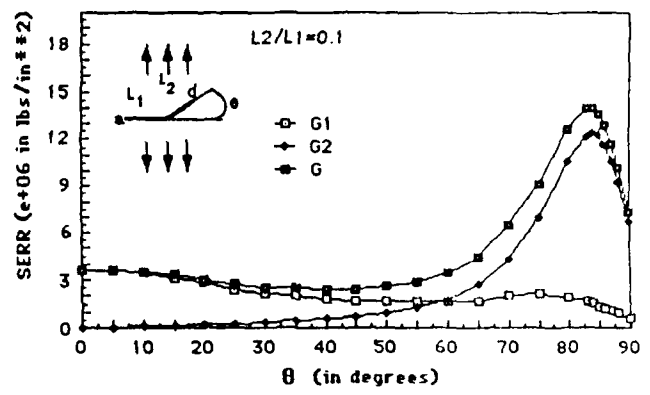


Figure 10. Mixed-Mode Strain Energy Release Rates at the Kink Tip as a Function of Kink Angle  $\theta$ . Orthotropic Case. ( $e_x=1, L_2 \rightarrow 0$ ).

**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 [0<sub>2</sub>/90<sub>2</sub>]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  $0^\circ$  to  $90^\circ$  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  $G_{II}/G_I$  ratios ranging uniformly from 0 to about 3. The case of predominantly mode-II ( $G_{II}/G_I > 3$ ) or pure mode-II ( $G_I = 0$ ) 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 AS4-3501-06 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:

$$\theta = 0^{\circ}, 5^{\circ}, 10^{\circ}, 15^{\circ}, 20^{\circ}, 25^{\circ}, 90^{\circ}$$

$$a = 2.5 \text{ mm}, 3.2 \text{ mm}, 3.8 \text{ mm}, 4.5 \text{ mm}$$

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:

$$G_I = C_I (e_x)^2 \quad G_{II} = C_{II} (e_x)^2 \quad (1)$$

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/m<sup>2</sup> depending on the material system used. In this study,  $G_{IC}$  has the value in the order of 300 J/m<sup>2</sup>. 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  $G_{II}/G_I$  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

- [1] A. S. D. Wang, "Fracture Analysis of Sublaminar Cracks in Composite Materials," Composite Technology Review, Vol. 6, 1984, pp. 45-62.
- [2] D. J. Wilkins, J. R. Eisenmann, R. A. Camin, W. S. Margolis and R. A. Benson, "Characterizing Delamination Growth in Graphite Epoxy," in Damage in Composite Materials, ASTM STP-775, 1982. pp. 168-183.
- [3] R. L. Ramkumar and J. D. Whitcomb, "Characterization of Mode-I and Mixed-Mode Delamination Growth in T300/5208 Graphite-epoxy," in Delamination and Debonding of Materials, ASTM STP-876, 1985. pp. 315-335.
- [4] W. L. Bradley and R. N. Cohen, "Matrix Deformation and Fracture in Graphite Reinforced Epoxies," in Delamination and Debonding of Materials, ASTM STP-876, 1985. pp. 389-410.
- [5] A. S. D. Wang, N. N. Kishore and W. W. Feng, "On Mixed-Mode Fracture in Off-Axis Unidirectional Graphite-Epoxy Composites," in Progress in Science and Engineering of Composites, Japan Society for Composite Materials, Vol. 1, 1982. pp. 599-606.
- [6] A. J. Russell and K. N. Street, "Moisture and Temperature Effects on the Mixed-Mode Delamination Fracture of Unidirectional Graphite-Epoxy," in Delamination and Debonding of Materials, ASTM STP-876, 1985. pp. 349-370.

[7] W. Binienda, "Mixed-Mode Fracture in Fiber-Reinforced Composite Materials," Ph.D. Thesis, Drexel University, Jan. 1988



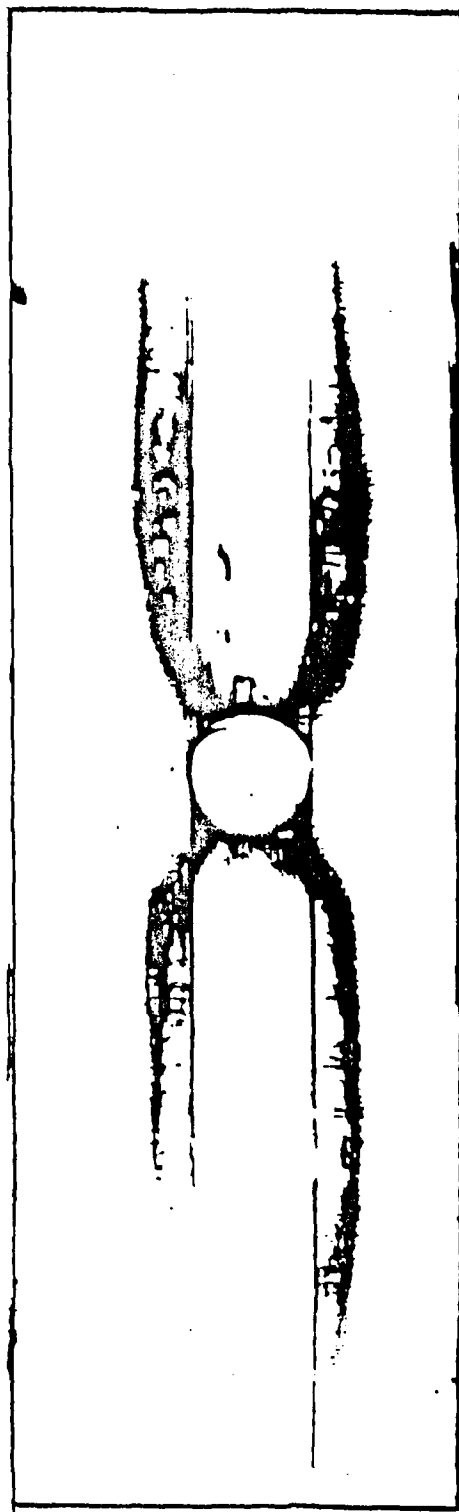


Fig. 1. X-radiograph of matrix crack development in a notched  $[0_2/90_2]_s$  graphite-epoxy laminate loaded in axial tension

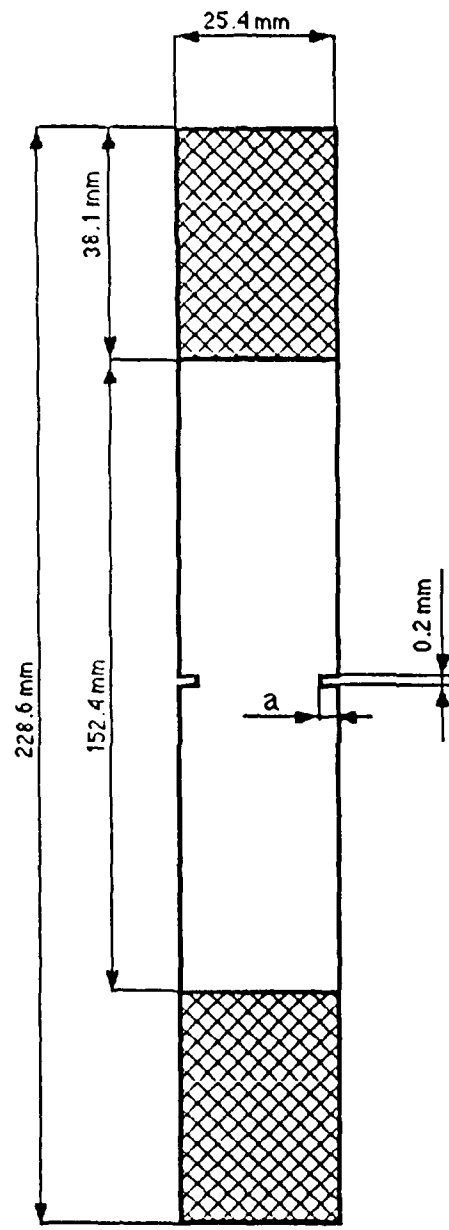


Figure 2 Geometry of the double side-notched specimen.

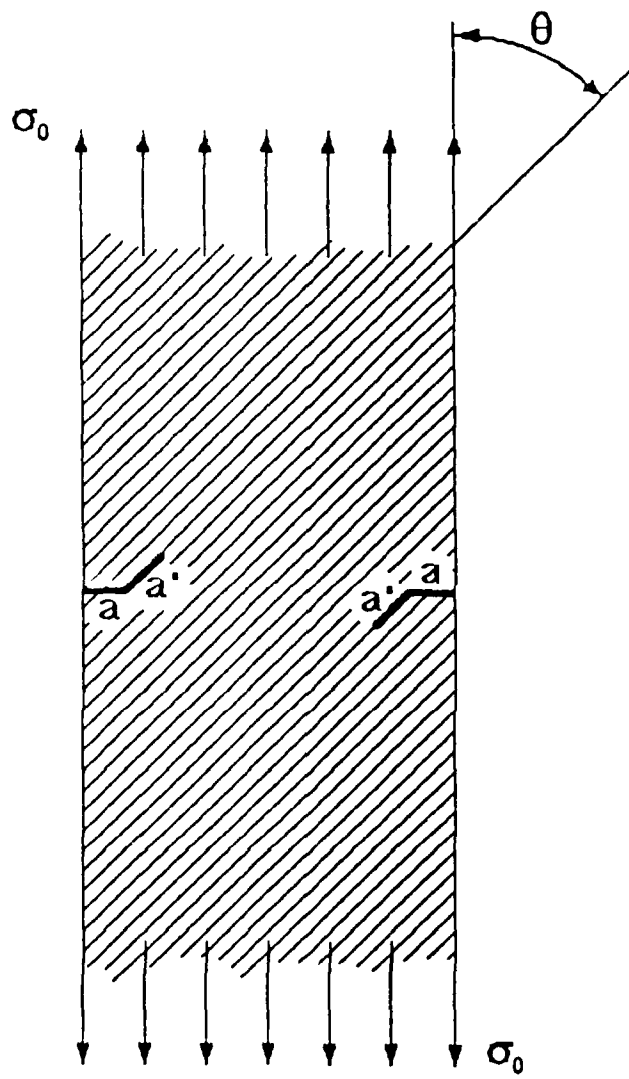


Figure 3 Geometry of kink cracks in the tested specimen.

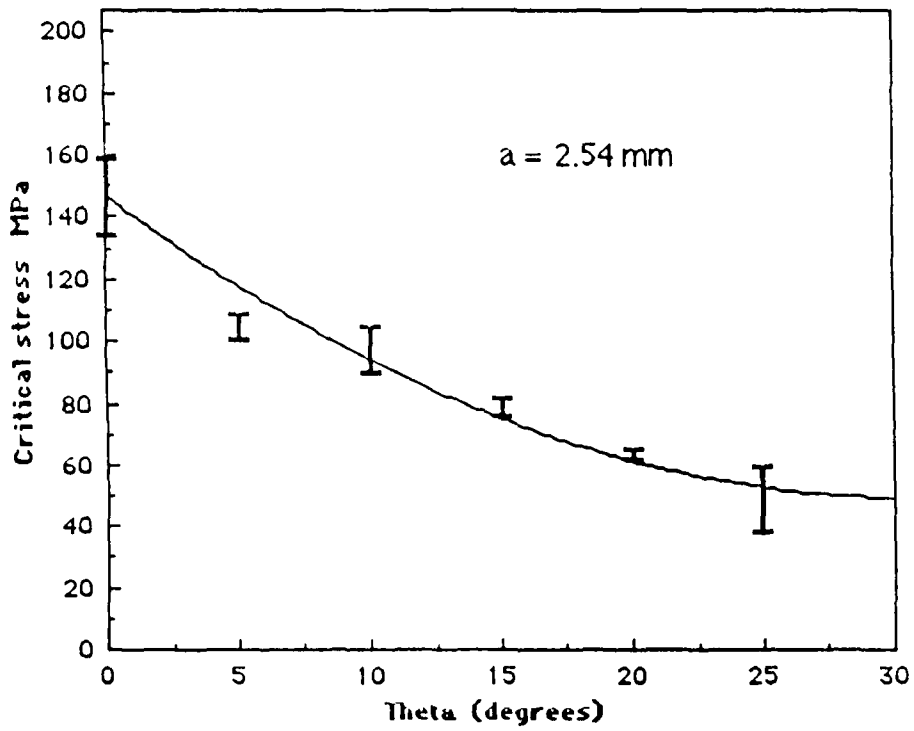


Figure 4 Critical stresses at onset of kink crack.  $a = 2.54$  mm.

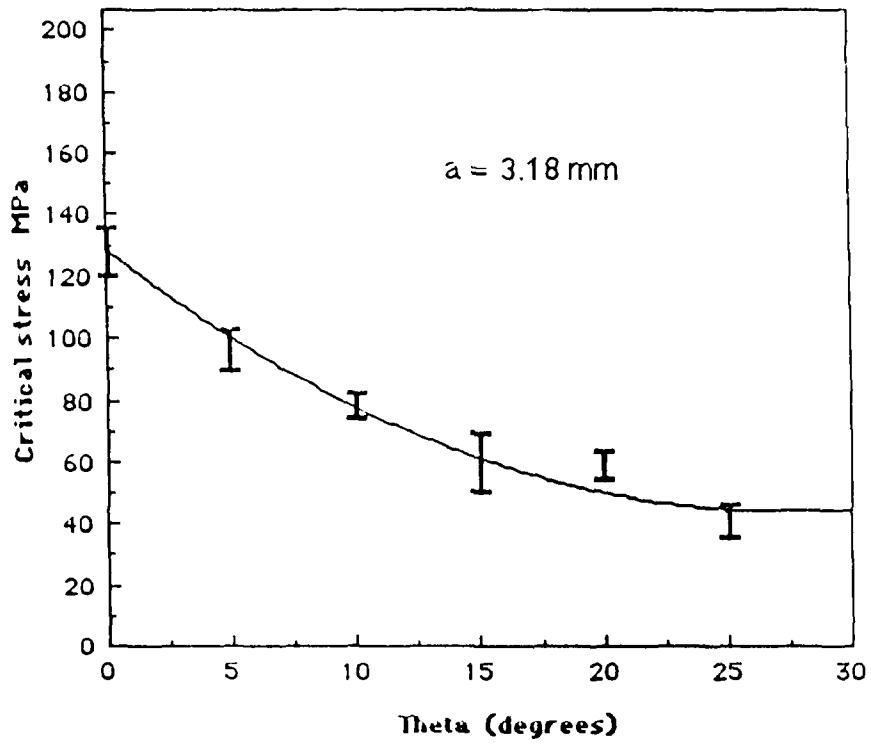


Figure 5 Critical stresses at onset of kink crack.  $a = 3.18$  mm.

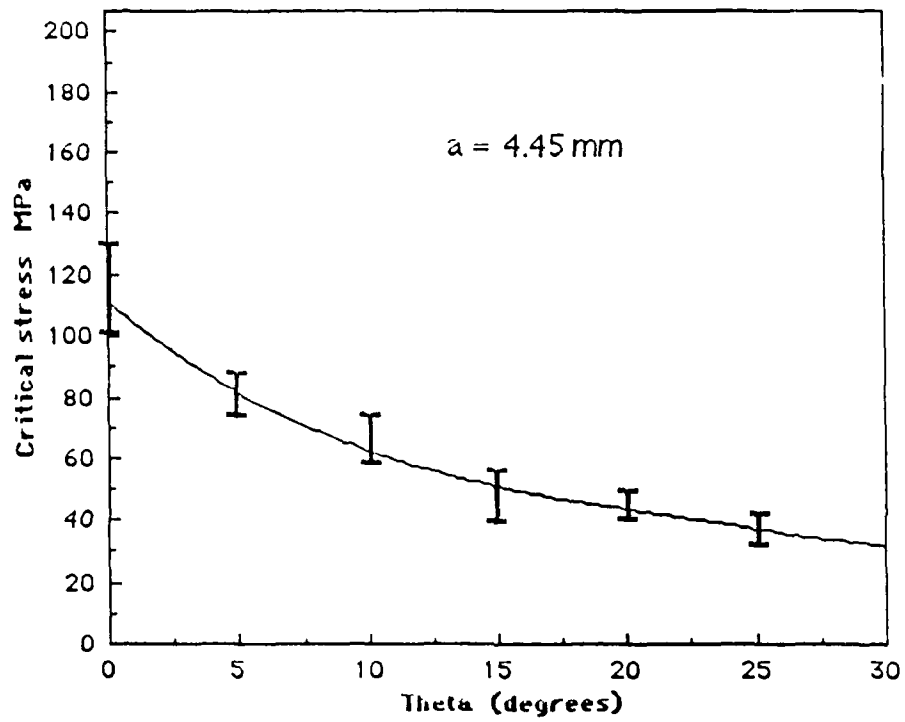


Figure 6 Critical stresses at onset of kink crack.  $a = 4.45$  mm.

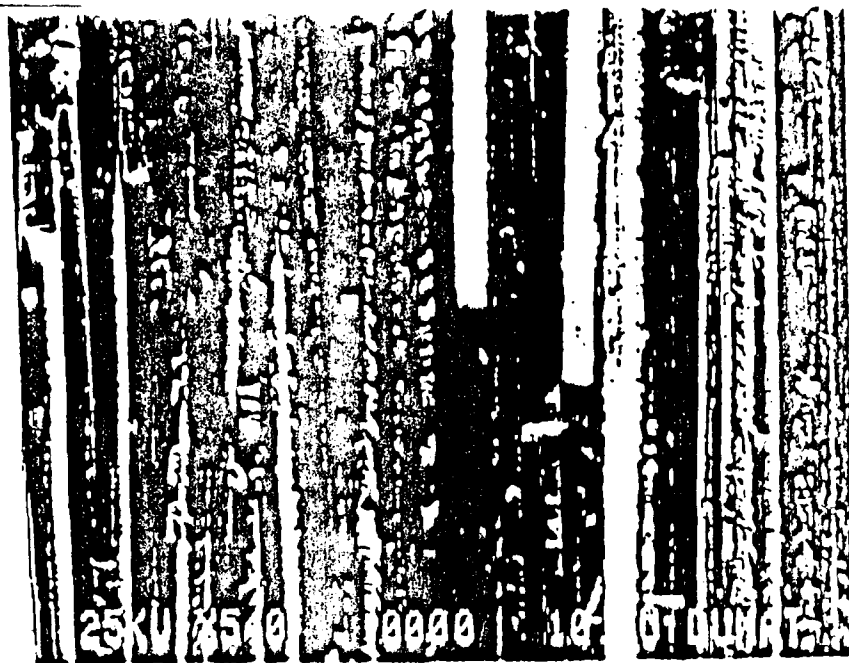


Figure 7 Photomicrographs of fractured surface near kink point. Above:  $\theta = 0^\circ$ ,  $a = 3.81$  mm ; below:  $\theta = 5^\circ$ ,  $a = 2.54$  mm.

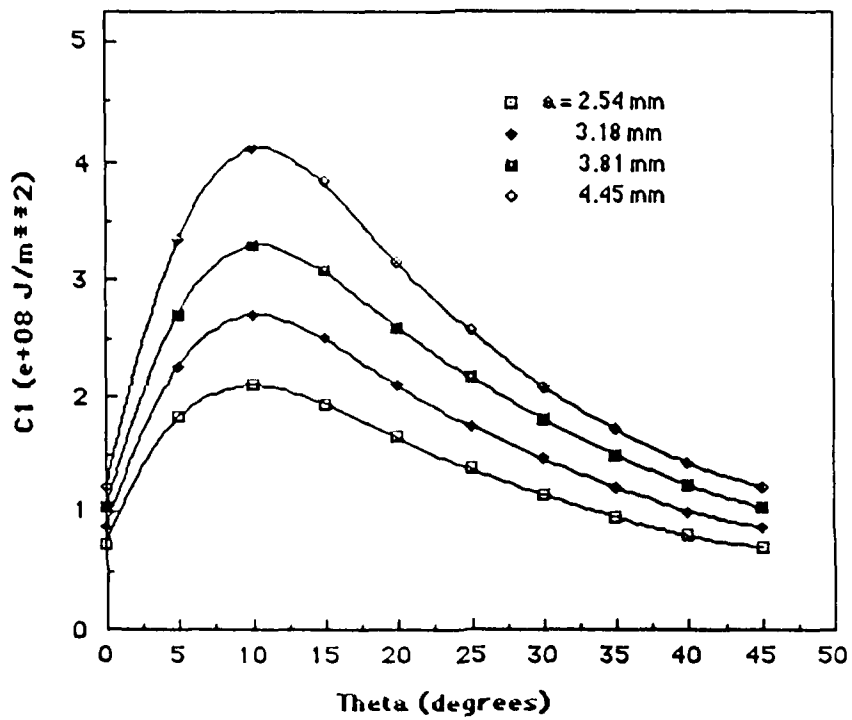


Figure 8 Mode-I strain energy release rate coefficients as function of off-axis angle  $\theta$ .



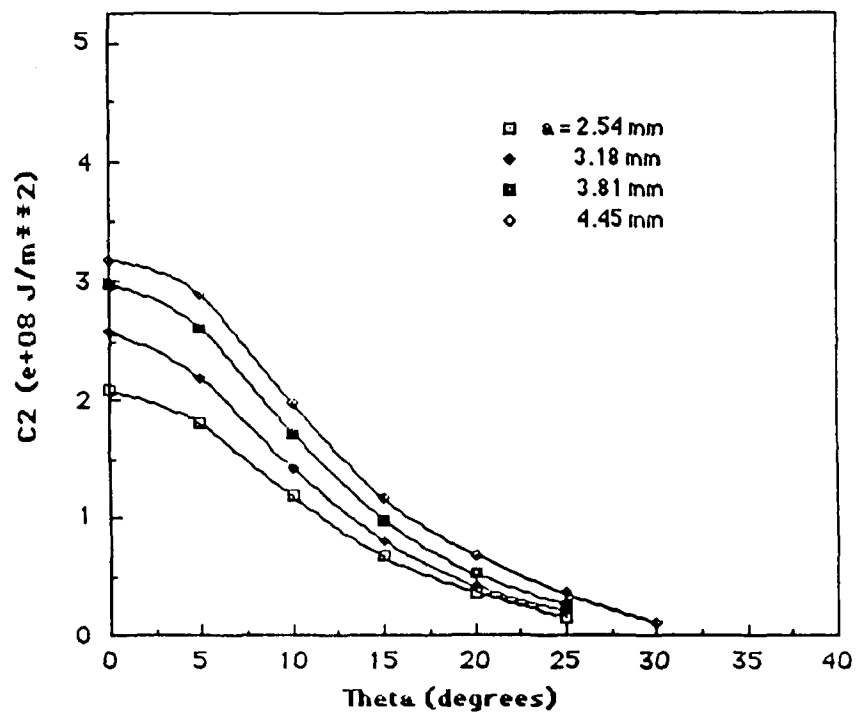


Figure 9 Mode-II strain energy release rate coefficients as function of off-axis angle  $\theta$ .

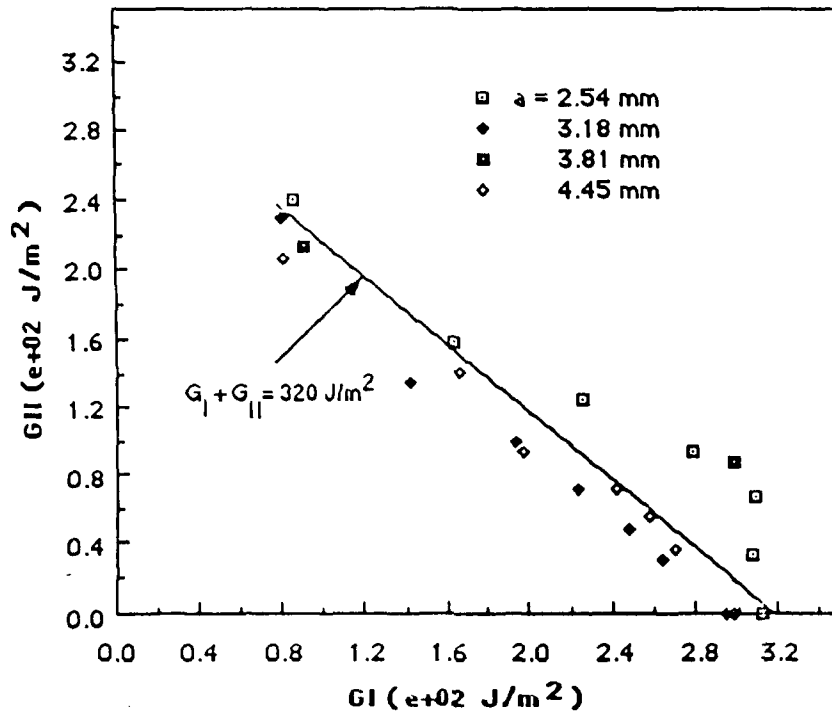


Figure 11 Interaction diagram of mixed-mode strain energy release rate data.

**A COMPREHENSIVE STUDY  
ON DAMAGE TOLERANCE PROPERTIES OF  
NOTCHED COMPOSITE LAMINATES**

**Appendix III**

**Three-Dimensional Simulation of Crack Growth  
in Notched Laminates**

Paper presented at the 2nd Annual Meeting, Society for composites, Univ. Delaware;  
also in Proceedings of the American Society for Composites, 1987. pp. 444-457.

# 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  $[0_2/90_2]_s$  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  $[0_2/90_2]_s$  laminate coupon under uniaxial tension as an example. A fiber-wise crack in the inner  $90^\circ$ -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  $0^{\circ}$ -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  $[0_2/90_2]_s$  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 frame work 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  $[0_2/90_2]_s$  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 comparisons 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.  $[0_2/90_2]_s$  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  $[0_2/90_2]_s$  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 schematical 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  $e_x = 10^{-6}$ , and the other is by prescribing a uniform temperature change of  $\Delta T = -1^\circ\text{C}$ . Stresses due to applied laminate tension (by giving a value for  $e_x$ ) 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  $0^0$ -layer. This stress is thought to cause  $0^0$ -layer split, which in fact was observed as the first mode of failure under a very low tensile loading. For the case of  $e_x = 10^{-6}$ ,  $\sigma_y$  is tensile throughout the thickness of the  $0^0$ -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<sup>0</sup>-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; its 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  $\Delta T = -1^{\circ}\text{C}$ ,  $\sigma_y$  is also tensile throughout the 0<sup>0</sup>-layer. Fig. 5c shows the  $\sigma_y$  distribution in the 0<sup>0</sup>-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<sup>0</sup>-layer is quite large. Thus, the combined tensile and thermal loading will cause the 0<sup>0</sup>-layer splitting to be in mixed modes.

Next, let us examine  $\sigma_x$  in the 90<sup>0</sup>-layer. This stress causes 90-layer transverse cracking. Again, for the case of  $e_x = 10^{-6}$ , this stress is tensile and varies throughout the thickness of the 90<sup>0</sup>-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<sup>0</sup>-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  $\Delta T = -1^{\circ}\text{C}$ ,  $\sigma_x$  in the 90<sup>0</sup>-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<sup>0</sup>-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<sup>0</sup>-layer splitting and 90<sup>0</sup>-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<sup>0</sup>-Layer Splitting. To simulate the initiation and growth of 0<sup>0</sup>-layer splitting, we shall assume that 90<sup>0</sup>-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  $\Delta T = -1^0\text{C}$ ,  $\sigma_y$  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]:

$$G(s) = [\sqrt{C_e} e_x + \sqrt{C_t} \Delta T]^2 d \quad (1)$$

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  $\Delta T=-1^0\text{C}$ , 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,

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

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  $(\epsilon_x)_{cr}$  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  $(\epsilon_x)_{cr}$  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 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-III under the applied tensile loading ( $\epsilon_x=1$ ), while primarily in mode I and II under thermal cooling ( $\Delta T=-1^\circ\text{C}$ ). 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

1. A. S. D. Wang, "Fracture Mechanics of Sublaminar Cracks in Composite Laminates," Composite Technology Review, Vol. 6, 1984, pp. 45-62.
2. F. W. Crossman and A. S. D. Wang, "The Dependence of Transverse Cracking and Delamination on Ply Thickness in Graphite-Epoxy Laminates," in Fatigue of Fibrous Composite Materials, ASTM STP-775, 1982, pp. 118-139.
3. A. S. D. Wang and F. W. Crossman, "Initiation and Growth of Transverse Cracks and Edge Delamination in Composite Laminates, Part 1: An Energy Method," Jour. Composite Materials, Vol. 14, 1980, pp. 71-87.

4. F. W. Crossman, W. J. Warren, A. S. D. Wang and G. E. Law, "Initiation and Growth of Transverse Cracks and Edge Delamination in Composite Laminates, Part 2: Experimental Correlation," Jour Composite Materials, Vol. 14, 1980. pp. 88-108.
5. A. S. D. Wang, N. N. Kishore and C. A. Li, "On Crack Development in Graphite-Epoxy [0<sub>2</sub>/90<sub>2n</sub>]<sub>s</sub> Laminates Under Uniaxial Tension," Composite Science and Technology, Vol. 24, 1985. pp. 1-31.
6. A. S. D. Wang, W. Binienda and Y. Zhong, "Mixed Mode Fracture of Uniaxial Fiber Reinforced Composites," NADC Technical Report, 1987.
7. A. S. D. Wang, E. S. Reddy and N. N. Kishore, "A three Dimensional Finite Element Code and Users Manual for Analysis of Matrix Cracking in Laminates," Drexel University Technical Report, 1987.
8. K. L. Reifsnider, "Critical Element Concepts for Residual Strength and Life Prediction," Proc. American Society for Composites, Technomic Publishing, 1986. pp. 387-403.

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

Notch Depth	2.54 mm	3.175 mm	3.81 mm	4.445 mm
0 <sup>0</sup> -layer Split	100 Mpa (80 Mpa)	75 Mpa (70 Mpa)	60 Mpa (60 Mpa)	60 Mpa (60 Mpa)
90 <sup>0</sup> -layer Transverse Crack	170 Mpa	160 Mpa	150 Mpa	150 Mpa
0/90 interface Delamination	350 Mpa (300 Mpa)	260 Mpa (230 Mpa)	225 Mpa (200 Mpa)	220 Mpa (180 Mpa)

Table 2. Pertinent Material Constants for AS4-3501-06 UD Ply

$$E_{LL} = 145 \text{ Gpa} \quad E_{TT} = E_{ZZ} = 10.3 \text{ Gpa} \quad G_{LT} = G_{LZ} = 6.8 \text{ Gpa} \quad G_{TZ} = 3.5 \text{ Gpa}$$

$$\nu_L = \nu_{LZ} = 0.3 \quad \nu_{TZ} = 0.54 \quad \alpha_L = 0.4 \times 10^{-6}/^{\circ}\text{C} \quad \alpha_T = \alpha_Z = 28.8 \times 10^{-6}/^{\circ}\text{C}$$

$$\Delta T = -140 \text{ }^{\circ}\text{C} \quad (G_c)_{\text{total}} = 289 \text{ J/m}^2 \quad \text{Ply Thickness} = 0.127 \text{ mm}$$

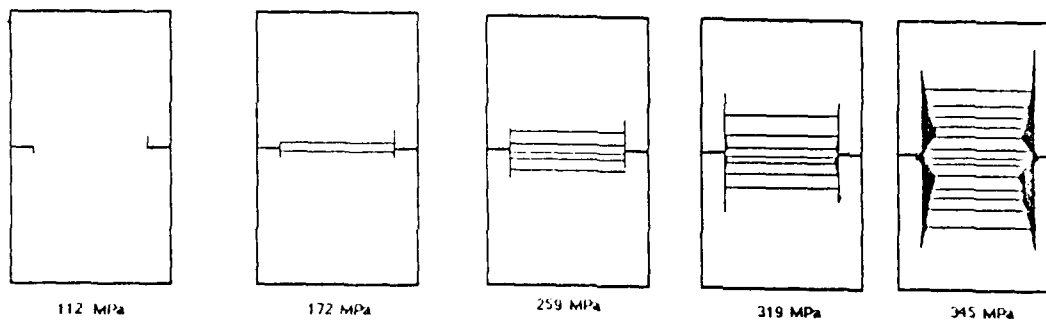


Fig 1. Development of Matrix Cracks in Specimen of Notch Depth 3.175 mm

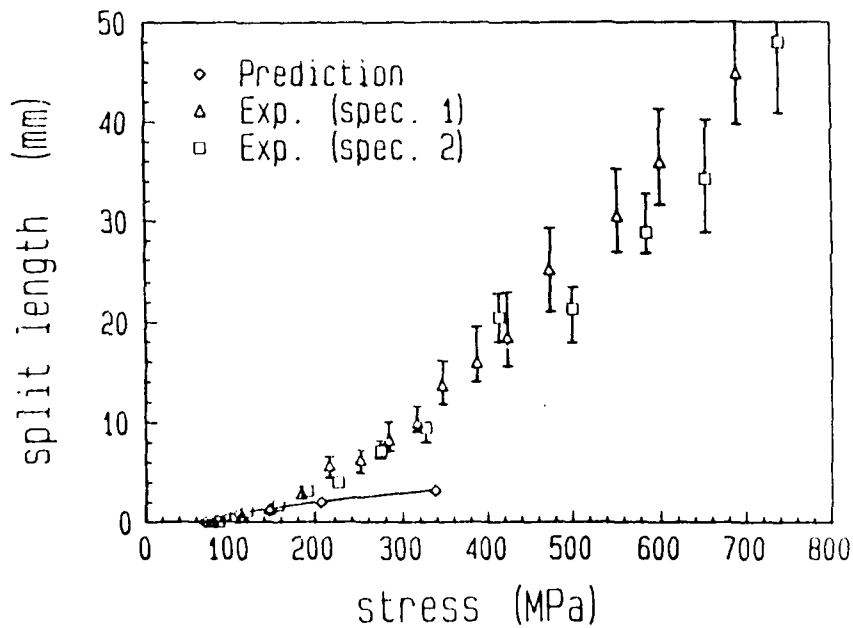


Fig 2. Split Length Growth versus Applied Tension ( Notch Depth, 3.175 mm)

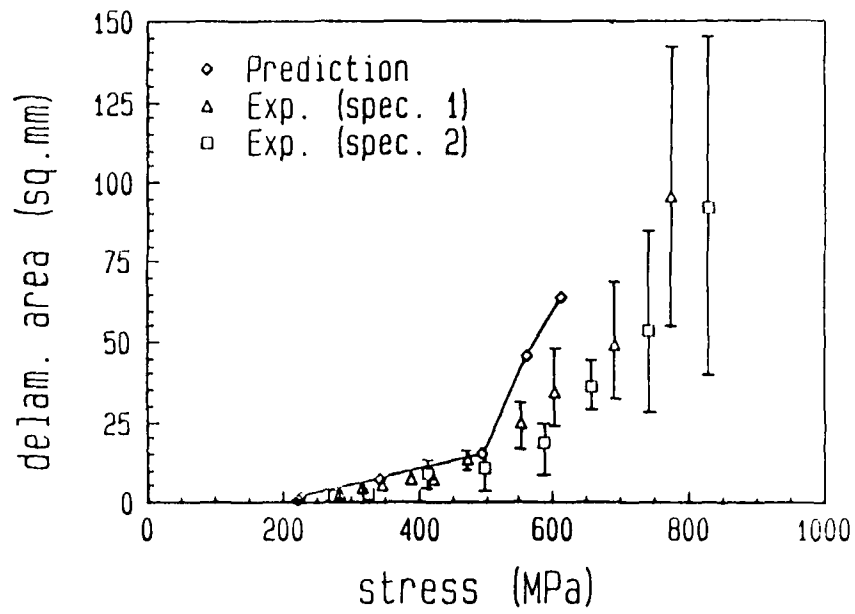


Fig. 3. Delamination Area Versus Applied Tension (Notch Depth 3.175 mm)

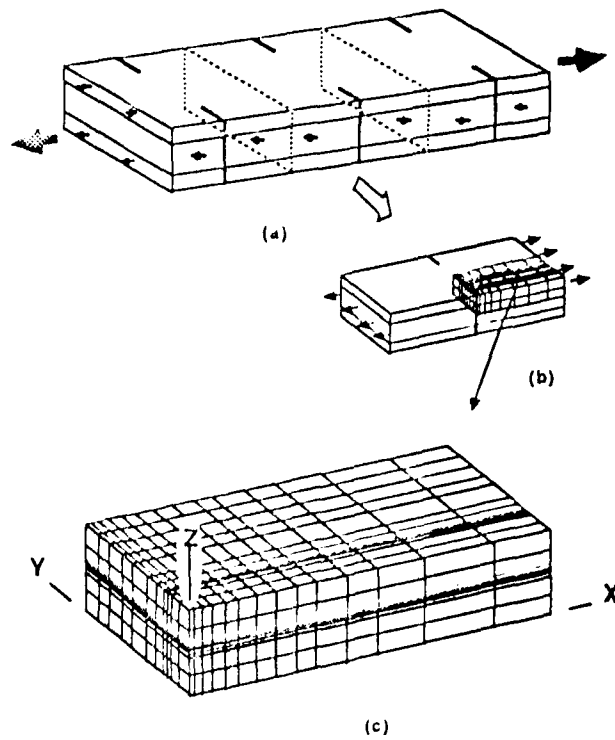
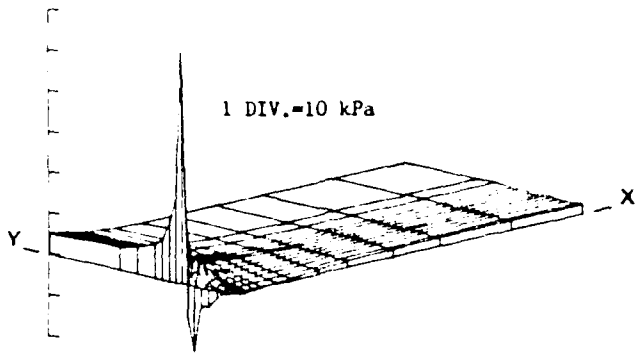
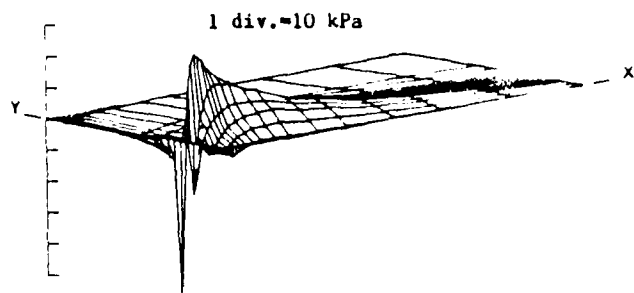


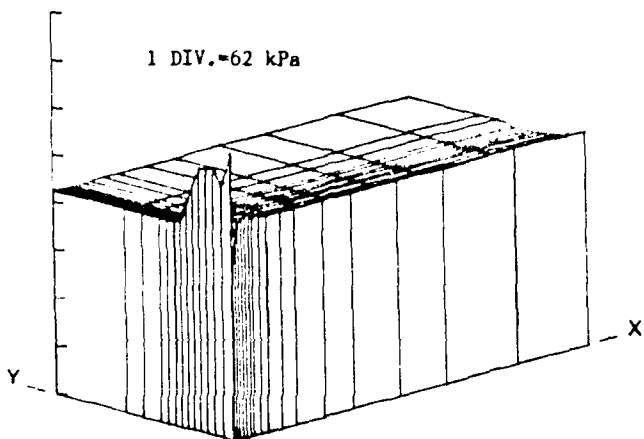
Fig. 4. Finite Element Model for the Double Side-Notched Specimen.



(a)

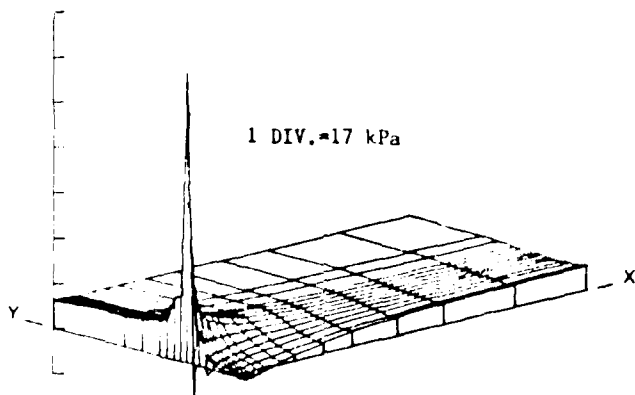


(b)

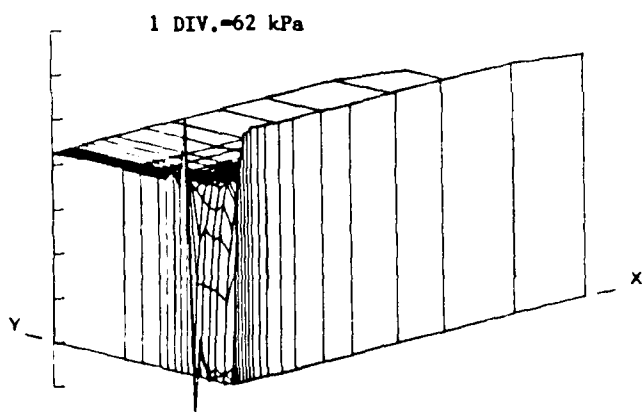


(c)

Fig 5. (a)  $\sigma_y$  Distribution in the  $0^0$ -Layer ( $e_x=10^{-6}$ )  
 (b)  $\tau_{xy}$  Distribution in the  $0^0$ -Layer ( $e_x=10^{-6}$ )  
 (c)  $\sigma_y$  Distribution in the  $0^0$ -Layer ( $\Delta T=-1^0C$ )

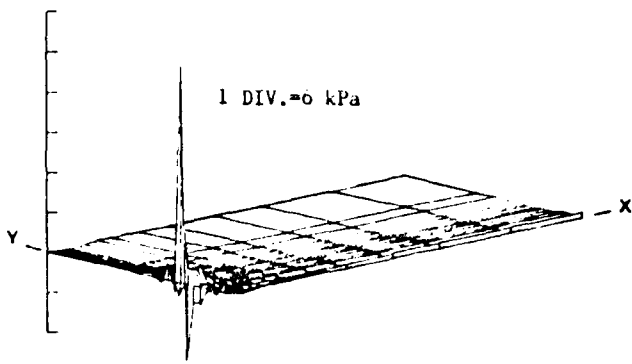


(a)

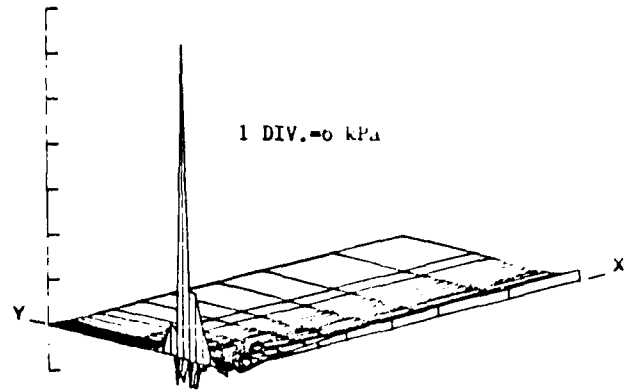


(b)

Fig 6  $\sigma_x$  Distribution in  $90^0$ -Layer due to (a)  $e_x=10^{-6}$  and (b)  $\Delta T=-1^0C$



7(a)



7(b)

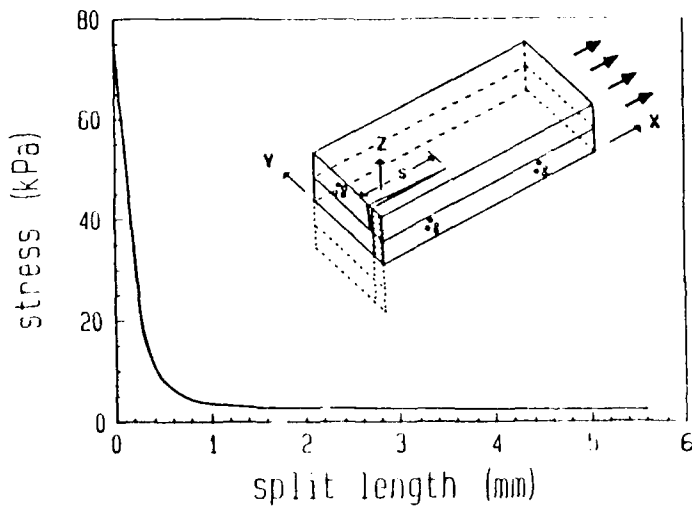
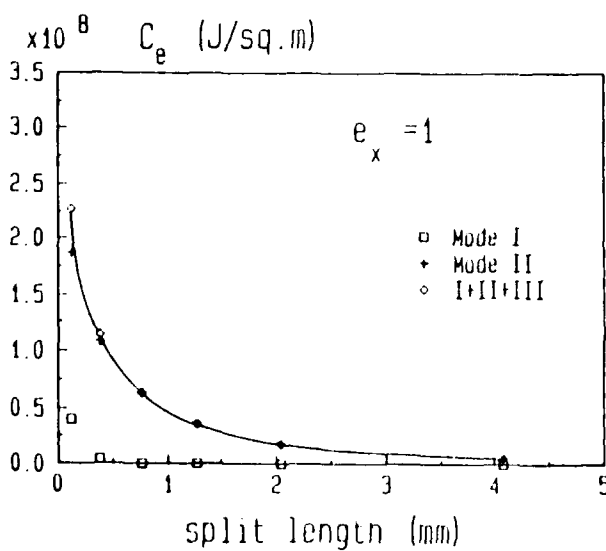


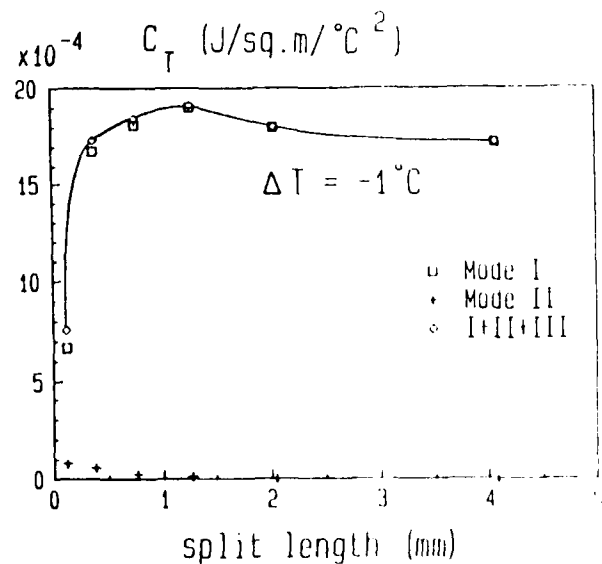
Fig. 7.  $\sigma_z$  Distribution on (a) 0/90 and (b) 90/90 Interface

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

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



9(a)



9(b)



Fig. 10. Finite Element Simulation of 0/90 interface Delamination.

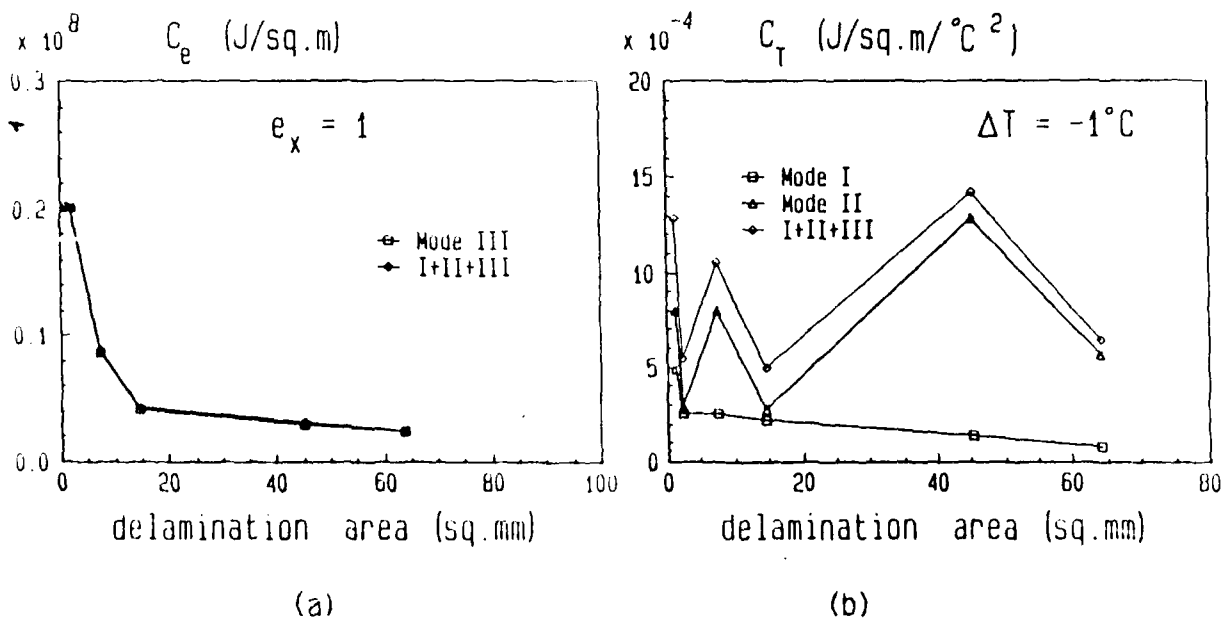
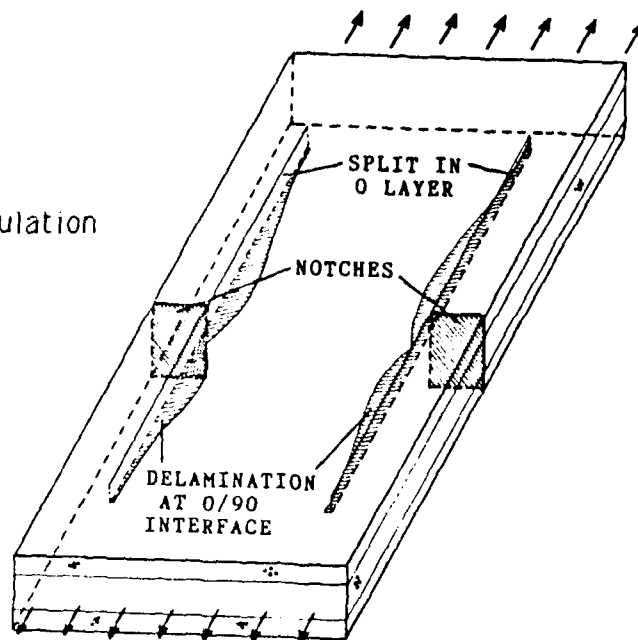


Fig. 11. Energy Release Rate at Delamination Front: (a) for  $e_x=1$ , (b)  $\Delta T=-1^\circ\text{C}$

**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;  
Also in Damage Mechanics in Composites, AD-12, ASME, 1987. pp. 83-91.

## SIMULATION OF MATRIX CRACKS IN COMPOSITE LAMINATES CONTAINING A SMALL HOLE

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

### ABSTRACT

This paper studies the matrix cracking sequence in  $[0_2/90_2]_S$  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 quantitative 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  $[0/90]_S$  type laminate under axial tension, in which the inner  $90^\circ$ -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  $[\pm 45/0/90]_S$  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  $[0_2/90_2]_S$  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  $[0_2/90_2]_S$ . 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 (intraply cracking) in the inner  $90^\circ$ -layer in the immediate region of the hole initially, and away from the hole region subsequently; (2) vertical splitting cracks (also a form of intraply cracking) in the outer  $0^\circ$ -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  $0^\circ$ -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)  $90^\circ$ -layer transverse cracks and an initial sign of  $0^\circ$ -layer splits are present at about 276 MPa. At 379 MPa, four branches of  $0^\circ$ -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  $90^\circ$ -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  $0^\circ$ -layer splits. While the delamination grows stably with load, more transverse cracks have formed and the longer the  $0^\circ$ -splits 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 observed 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  $0^\circ$ -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  $0^\circ$ -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  $[0_2/90_2]_S$  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 characterized 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  $\epsilon_x = 10^{-6}$ , and the other is by prescribing a uniform temperature change of  $\Delta T = -1^\circ\text{C}$ . 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 not 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  $0^\circ$ -layer near the  $0/90$  interface. Here, the largest stress is  $\sigma_x$  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  $\sigma_y$  and  $\tau_{xy}$ . 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  $90^\circ$ -layer near the mid-plane of the laminate. Here, we see the dominance of  $\sigma_x$  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 in order to evaluate the combined stress state. For the laminate used,  $\Delta T$  is set at  $-140^\circ\text{C}$ . 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  $30^\circ$ -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  $\sigma_y$  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  $\sigma_y$  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  $e_x$  and thermal loading  $\Delta T$  as [5]:

$$G(s)_i = \left\{ \left[ \sqrt{C_\theta} e_x + \sqrt{C_T} \Delta T \right]^2 d \right\}_i \quad i = \text{I, II, III} \quad (1)$$

where the coefficients  $C_\theta$  and  $C_T$  are functions of  $s$  and represent the strain energy release rates corresponding to  $e_x = 1$  and  $\Delta T = -1^\circ\text{C}$ , 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  $C_\theta$  and  $C_T$  versus the split length  $s$ . Note that  $C_\theta$  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:

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

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  $289 \text{ J/m}^2$ .

By using the coefficient curves in Figures 8 and 9, we can obtain from (1) and (2) the critical laminate strain  $e_x$  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 mechanics 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.

**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

- [1] A.S.D. Wang and F.W. Crossman, "Initiation and Growth of Transverse Cracks and Edge Delamination in Composite Laminates, Part 1, An Energy Method", *J. Composite Materials*, Vol. 4, pp. 71-87, 1980.
- [2] F.W. Crossman, W.J. Warren, A.S.D. Wang and G.E. Law, "Initiation and Growth of Transverse Cracks and Edge Delamination in Composite Laminates, Part 2, Experimental Correlation", *J. Composite Materials*, Vol. 14, pp. 88-106, 1980.
- [3] A.S.D. Wang, N.N. Kishore and C.A. Li, "Crack Development in Graphite-Epoxy Cross-ply Laminates Under Uniaxial Tension", *Composites Science and Technology*, Vol. 24, pp. 1-31, 1985.
- [4] A.S.D. Wang, E.S. Reddy and Yu Zhong, "Three Dimensional Simulation of Crack Growth in Notched Laminates", to be presented at Joint Symposium on Composite Materials Science and Engineering, American Society for Composites - University of Delaware Center for Composite Materials, September 23-25, University of Delaware, Newark DE, 1987.
- [5] E.F. Rybicki and D.W. Schmueser, "Effect of Stacking Sequence and Lay-up Angle on Free Edge Stresses Around a Hole in a Laminated Plate Under Tension", *J. Composite Materials*, Vol. 12, pp. 300-313, 1978.
- [6] I.M. Daniel, R.E. Rowlands and J.B. Whiteside, "Effects of Materials and Stacking Sequence on the Behavior of Composite Plates with Holes", *Experimental Mechanics*, Vol. 14, pp. 1-9, 1974.
- [7] T.K. O'Brien and I.S. Raju, "Strain Energy Release Rate Analysis of Delamination Around an Open Hole in Composite Laminates", Paper No. AIAA-84-0961, AIAA/ASME/ASCE/AHS 25th Structures, Structural Dynamics & Materials Conference, May 14-16, Palm Springs Ca., 1984.
- [8] I.S. Raju and J.H. Crews, "Three-Dimensional Analysis of  $[0/90]_S$  and  $[90/0]_S$  Laminates with a Central Circular Hole", *Composite Technology Review*, Vol. 4, No. 4, pp. 116-124, 1982.
- [9] L. Carlsson, "Interlaminar Stresses at a Hole in a Composite Member Subjected to a In-plane Loading", *J. Composite Materials*, Vol. 17, pp.241, 1983.
- [10] K. Ericson, M. Persson, L. Carlsson and A. Gustavsson, "On the Prediction of the Initiation of Delamination in a  $[0/90]_S$  Laminate with a Circular Hole", *Journal of Composite Materials*, Vol. 18, pp. 495-506, 1984.
- [11] A.S.D. Wang, W. Binienda, Yu Zhong and E.S. Reddy, "Mixed Mode Fracture of Uniaxial Fiber Reinforced Composites", NADC-TR-87133-60, Naval Air Development Center, Warminster, Pa. 18974, April 1987.
- [12] E.S. Reddy, N.N. Kishore and A.S.D. Wang, "A Three Dimensional Finite Element Code and Users' Manual for Matrix Cracking in Laminates", Drexel University Technical Report, September 1987.

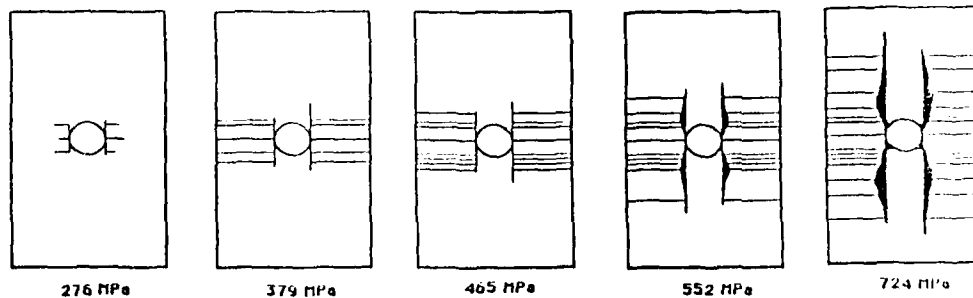


Figure 1. Progression of Matrix Cracking Under Ascending Laminate Stress.

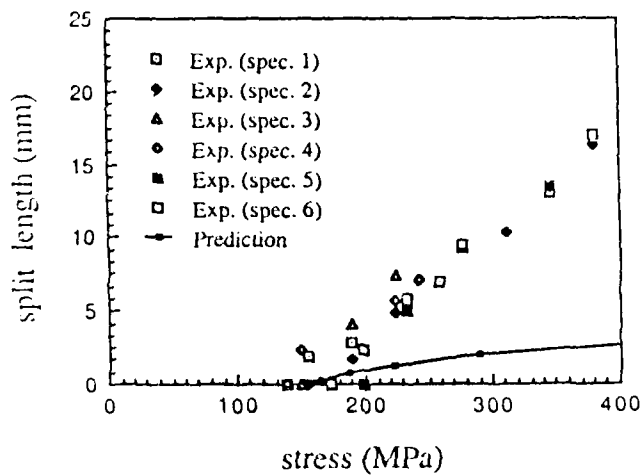


Figure 2. Splitting Length Versus Applied Laminate Stress.

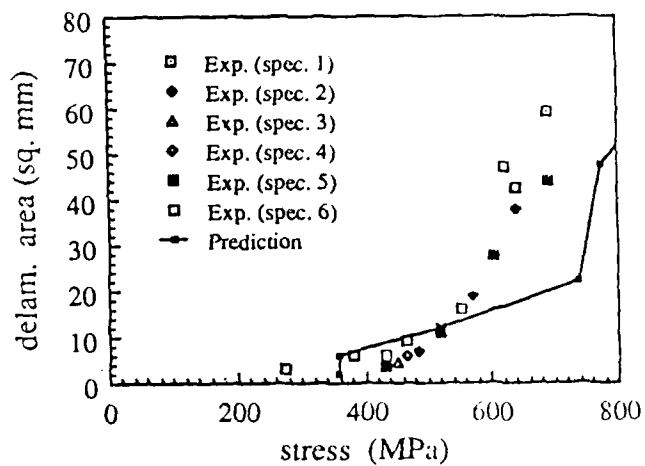


Figure 3. Delamination Area Versus Applied Laminate Stress

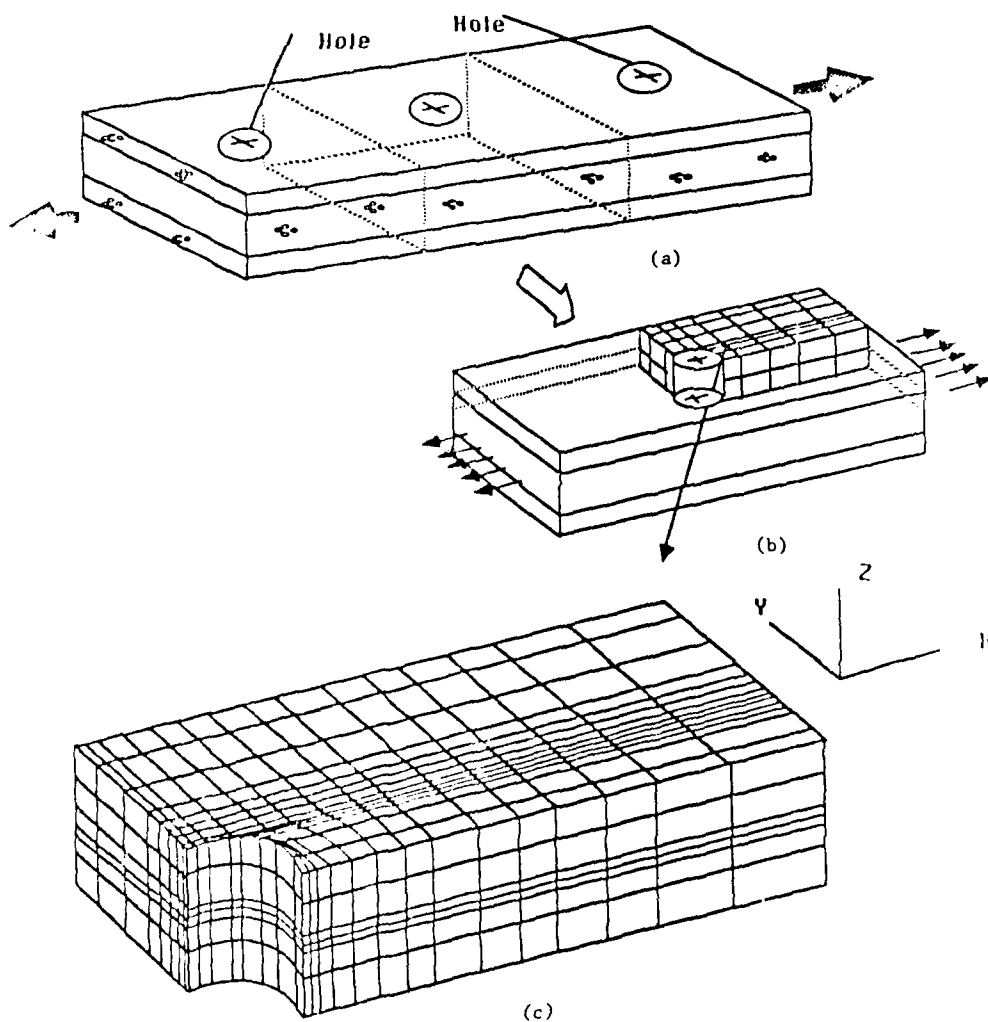


Figure 4. Finite Element Model for the Specimen Containing A Small Hole

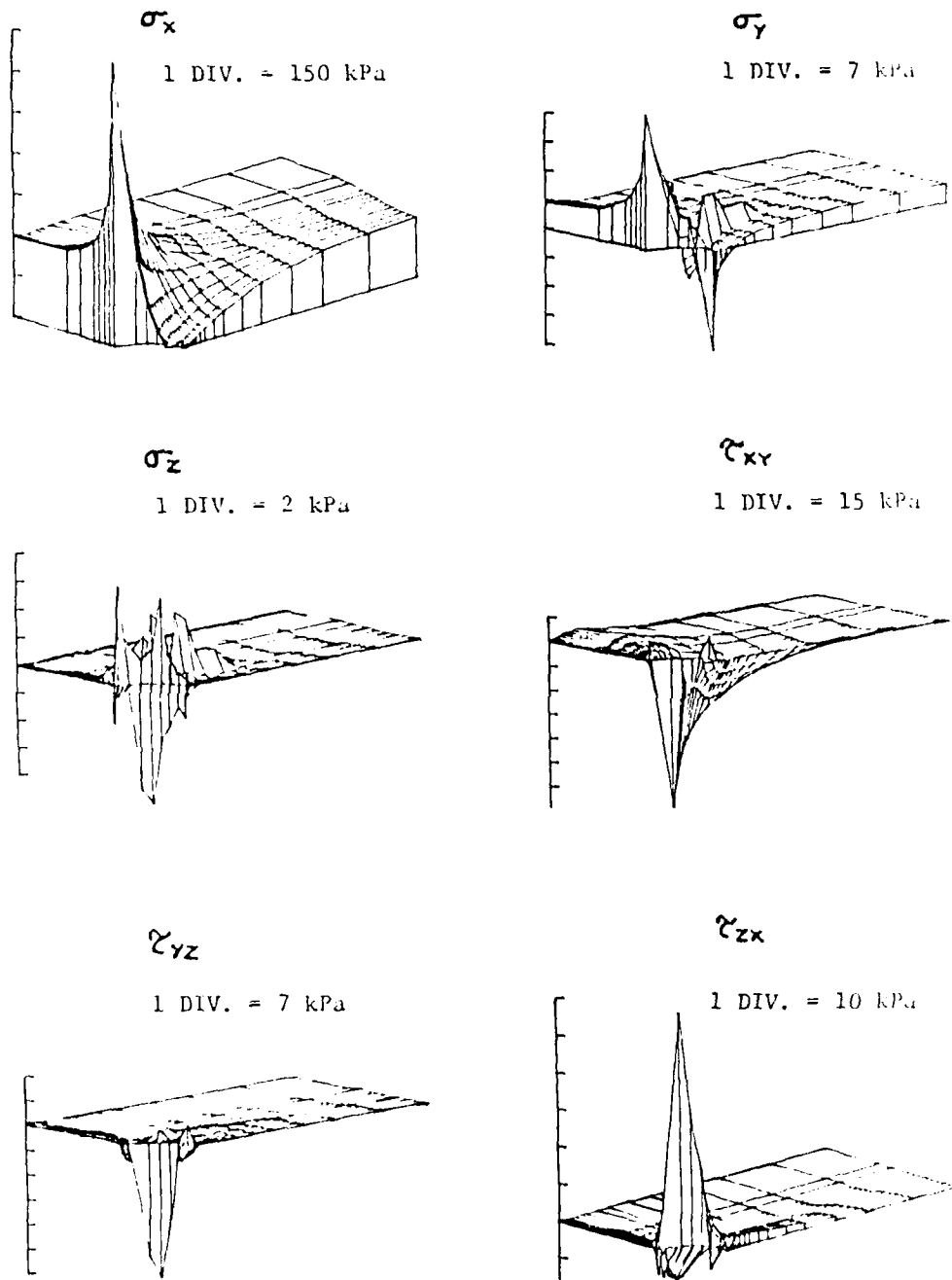


Figure 5. Distribution of Stresses in  $0^\circ$ -Layer Near  $0/90$  Interface.



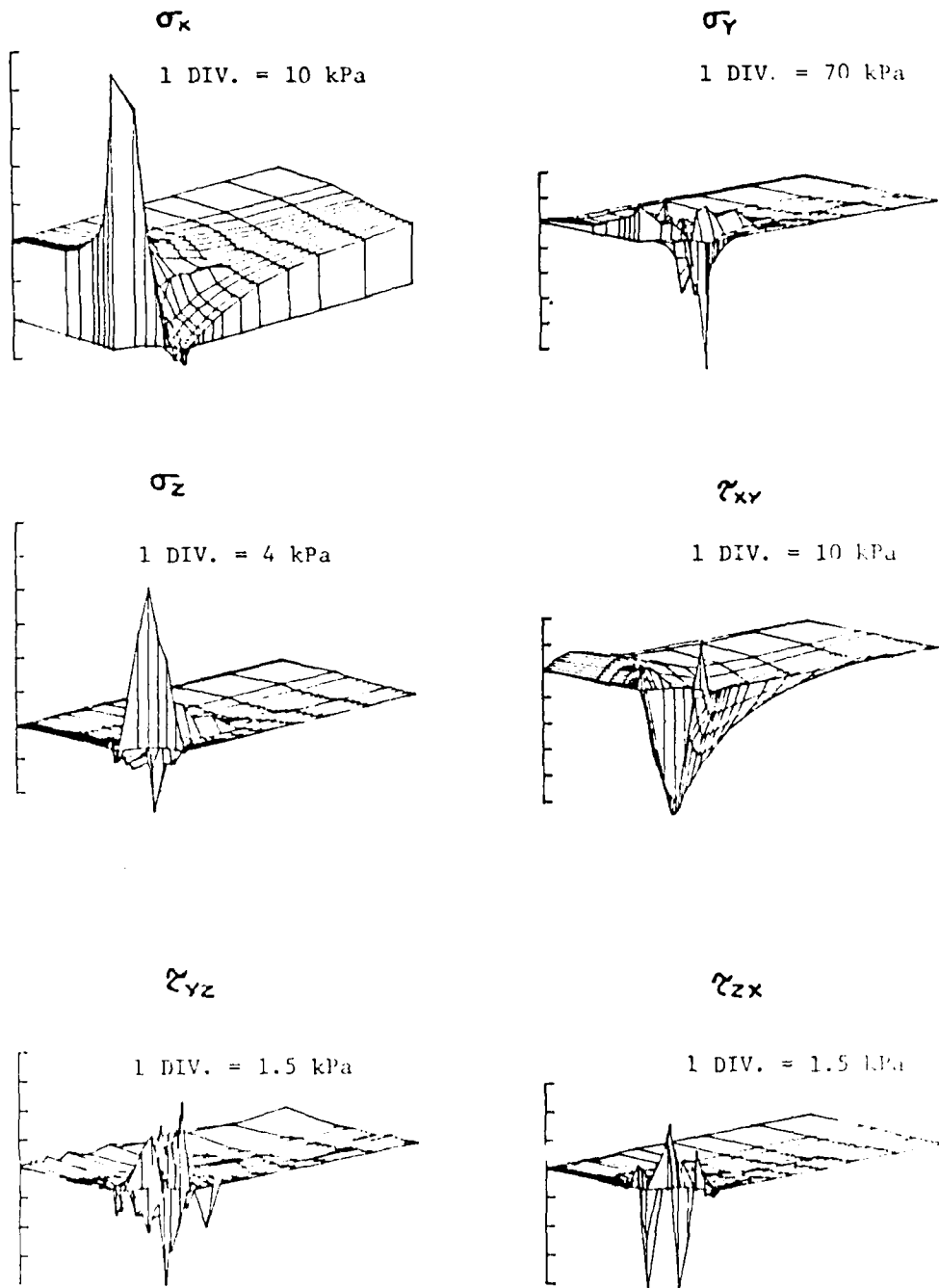


Figure 6. Distribution of Stresses in  $90^\circ$ -Layer Near  $90/90$  Interface.

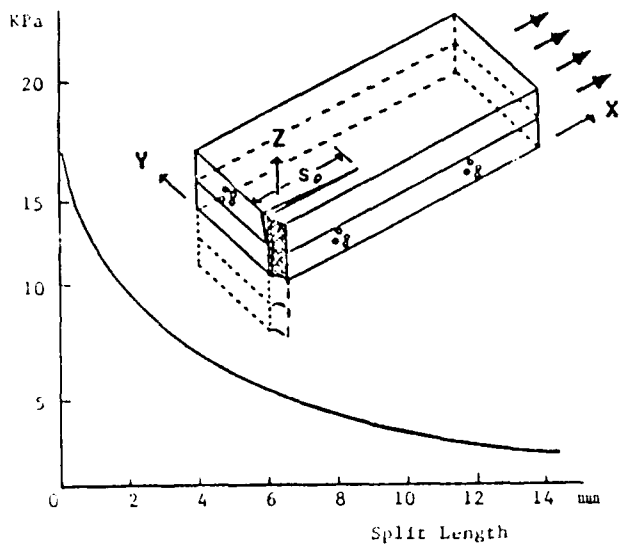


Figure 7. Split-Tip Normal (tensile) Stress Versus Split Length.

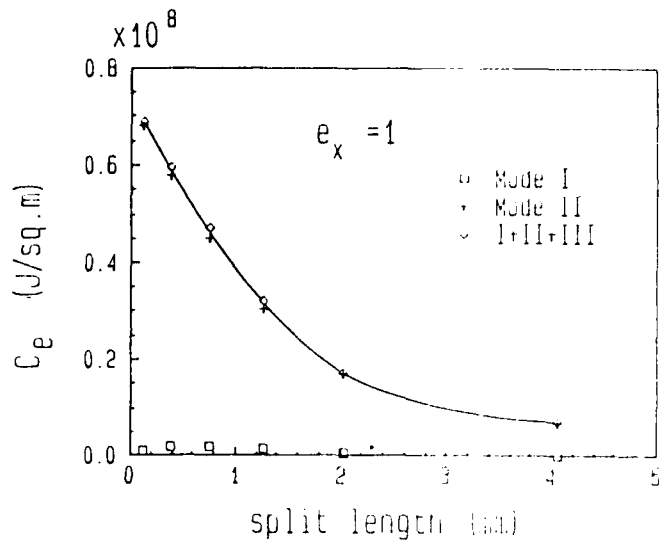


Figure 8. Energy Release Rate Coefficient  $C_e$  for  $0^\circ$ -Layer Splitting.

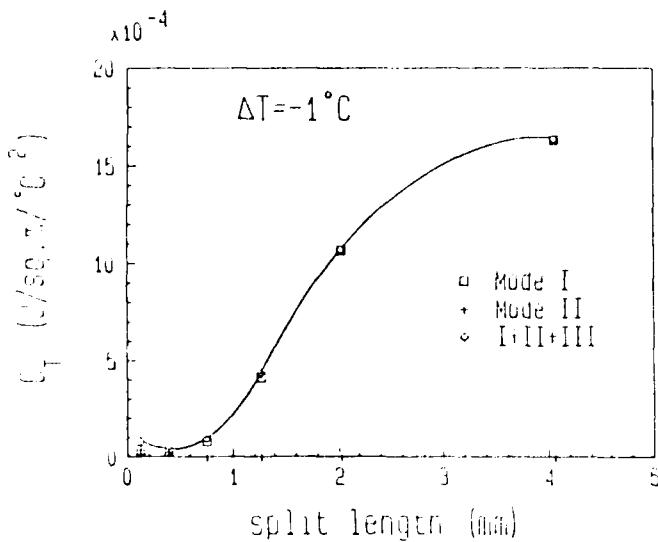


Figure 9. Energy Release Rate Coefficient  $C_T$  for  $0^\circ$ -Layer Splitting.

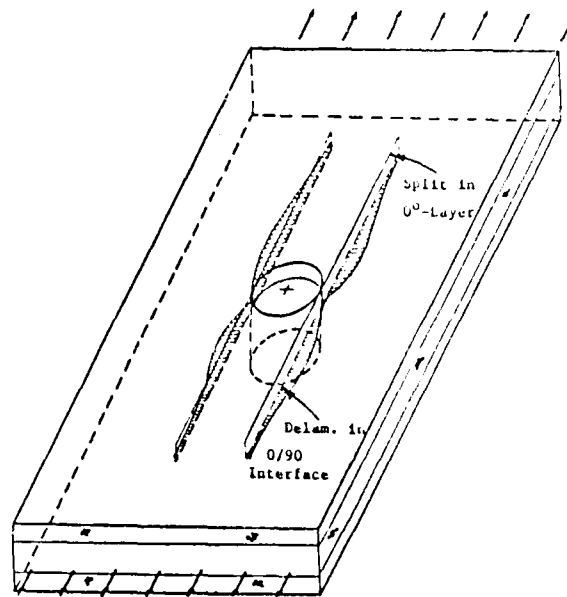


Figure 10. Simulation Model for  $0/90$  Interface Delamination.

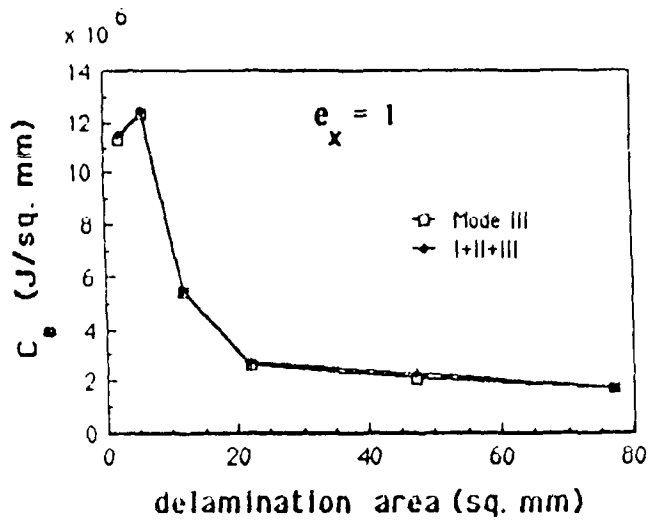


Figure 11. Energy Release Rate Coefficient  $C_e$  for 0/90 Interface Delamination.

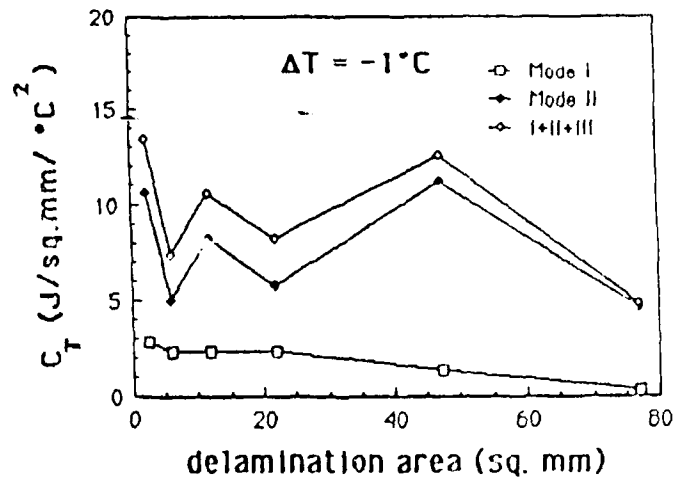


Figure 12. Energy Release Rate Coefficient  $C_T$  for 0/90 Interface Delamination.

**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 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.

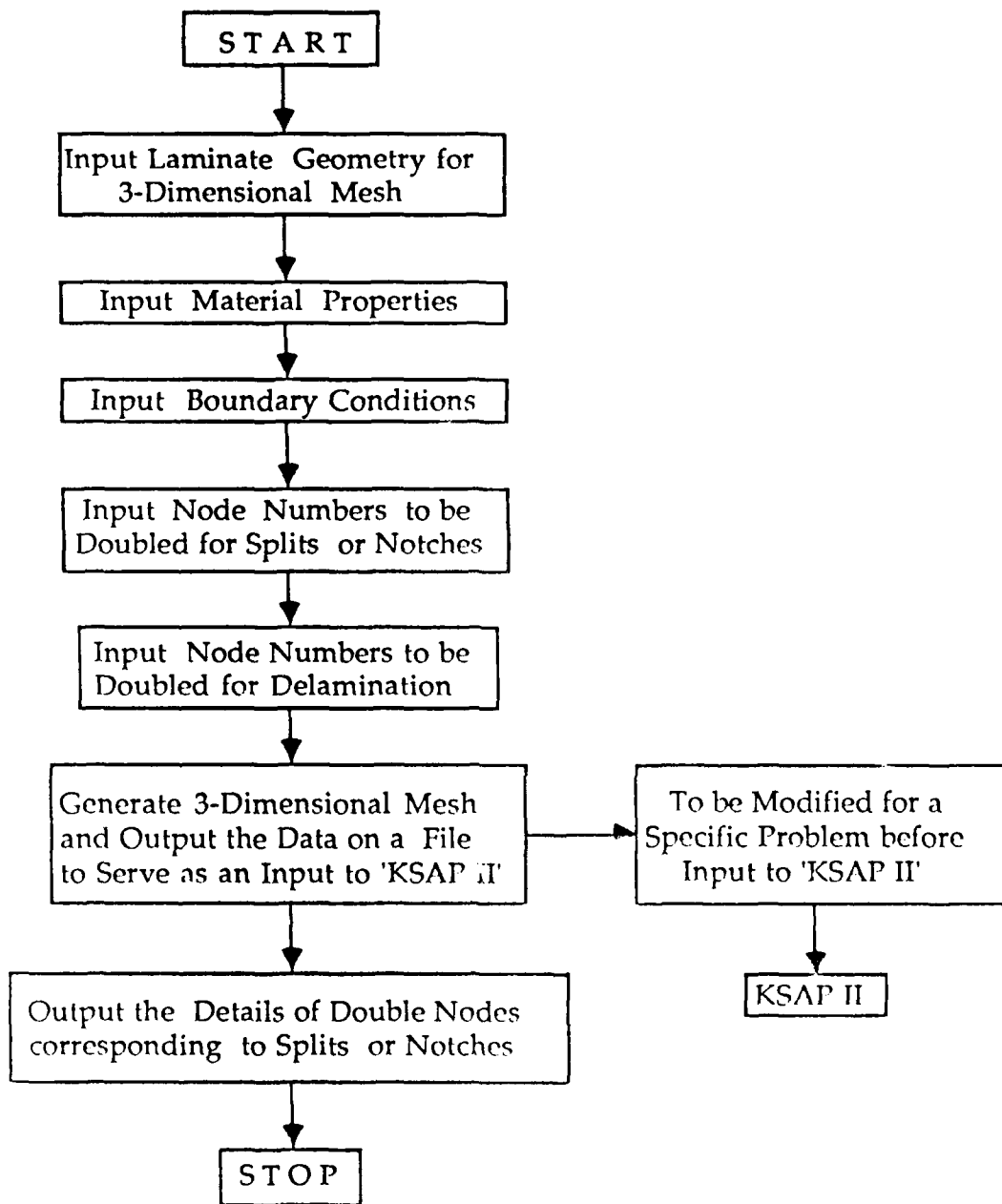
## 1.4 THE POSTPROCESSOR PROGRAM

The post processor mainly produces 3-D plots of the stresses with hidden lines removed. The input data for this programs is a modified

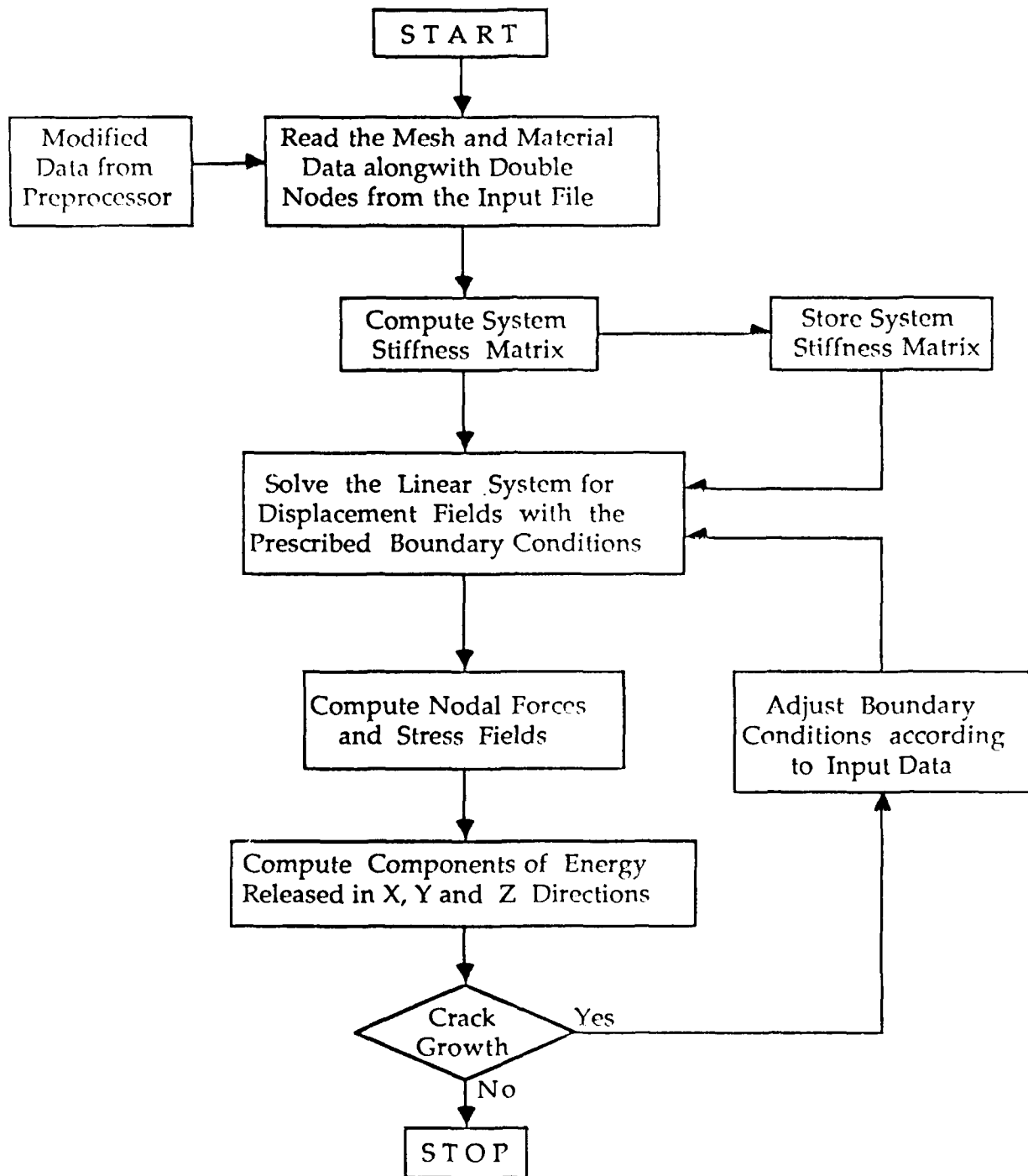


output file from the KSAP II program. Various stress distribution plots can be output along any specified plane. The three-dimensional plots can be processed at any specified viewing direction.

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



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.

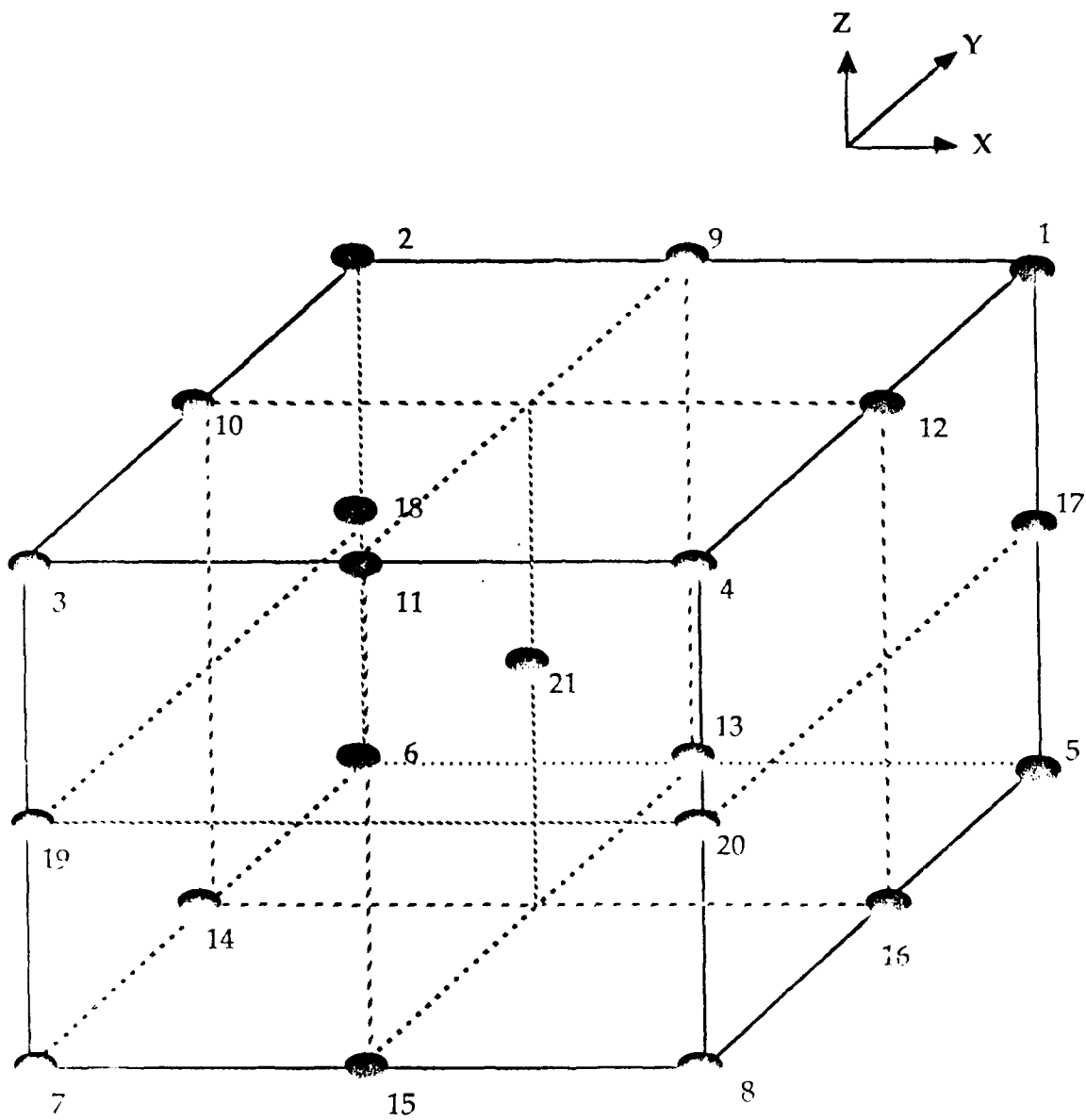


FIG. 2.1 THREE DIMENSIONAL 21 NODE ISOPARAMETRIC ELEMENT

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  $x=\text{constant}$  or  $y=\text{constant}$  or  $z=\text{constant}$ . For example: the interlaminar boundary (layer interface) corresponds to  $z=\text{constant}$ . 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 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):

HED(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.



card 5:

ZZ(1), ZZ(2), ... , ZZ(NONZ)

- z- coordinates of the nodes in the z - direction.

NOTE: For 21-node element generation, even numbered coordinates should be middle points of immediate neighboring points. i.e., for  $i=2,4,\dots$

$$xx(i)=(xx(i-1)+xx(i+1))/2.$$

$$yy(i)=(yy(i-1)+yy(i+1))/2.$$

$$zz(i)=(zz(i-1)+zz(i+1))/2.$$

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, MATER, NEND, INC

- N - starting element number
- MATER - 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 as to which material they belong to. Material properties for different identification numbers are given in later cards.

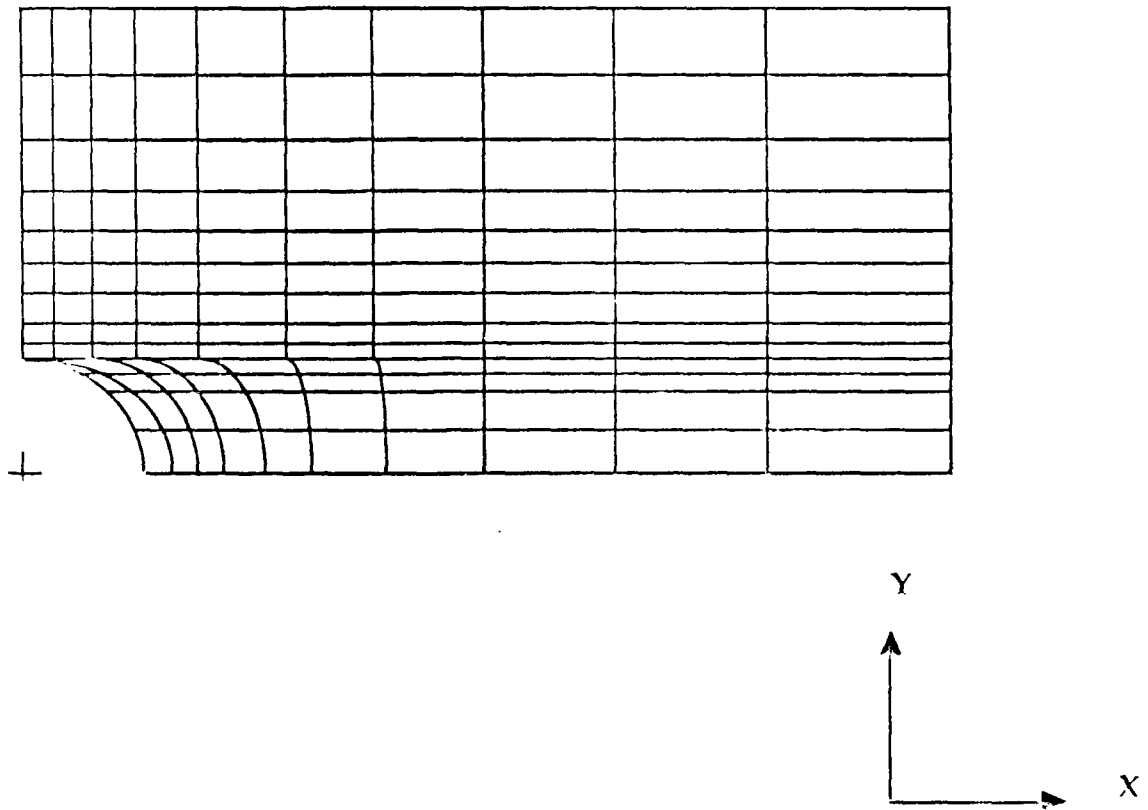


FIG. 2.2 FINITE ELEMENT MESH IN XY-PLANE IN A LAMINATE WITH A HOLE

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 upto 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 'EOP010.PAT'.

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,I4,4X,G17.7)

card 1:

<u>columns</u>	<u>entry description</u>
1- 4	bbEL
5- 8	material identification number
13-29	value of Young's modulus in L-direction

card 2:

<u>columns</u>	<u>entry description</u>
1- 4	bbET

	5- 8	material identification number
	13-29	value of Young's modulus in T-direction
card 3:		
	columns	entry description
	-----	-----
	1- 4	bbEZ
	5- 8	material identification number
	13-29	value of Young's modulus in Z-direction
card 4:		
	columns	entry description
	-----	-----
	1- 4	NULT
	5- 8	material identification number
	13-29	value of the poisson's ratio, $\nu_{lt}$
card 5:		
	columns	entry description
	-----	-----
	1- 4	NULZ
	5- 8	material identification number
	13-29	value of the poisson's ratio, $\nu_{lz}$
card 6:		
	columns	entry description
	-----	-----
	1- 4	NUTZ
	5- 8	material identification number
	13-29	value of the poisson's ratio, $\nu_{tz}$
card 7:		
	columns	entry description
	-----	-----
	1- 4	bGLT
	5- 8	material identification number
	13-29	value of the shear modulus, $G_{lt}$
card 8:		
	columns	entry description
	-----	-----
	1- 4	bGLZ
	5- 8	material identification number
	13-29	value of the shear modulus, $G_{lz}$
card 9:		
	columns	entry description
	-----	-----
	1- 4	bGTZ
	5- 8	material identification number
	13-29	value of the shear modulus, $G_{tz}$
card 10:		
	columns	entry description
	-----	-----
	1- 4	ALFL
	5- 8	material identification number
	13-29	value of the thermal expansion coeff., $\alpha_1$
card 11:		
	columns	entry description
	-----	-----
	1- 4	ALFT
	5- 8	material identification number

card 12:	13-29	value of the thermal expansion coeff., $\alpha_t$
	columns	entry description
	1- 4	ALFZ
	5- 8	material identification number
	13-29	value of the thermal expansion coeff., $\alpha_z$

NOTE: If any of these 12 cards are not supplied then that particular value will be set equal to zero.  
 The 12 constants ( $E_{11}, E_{22}, \dots, z$ ) are defined with respect to a set of axes (L,T,Z) which are the principal material directions.

(b) Data termination Card (Format A4)

columns	entry description
1-2	-1 indicates the end of material property cards.

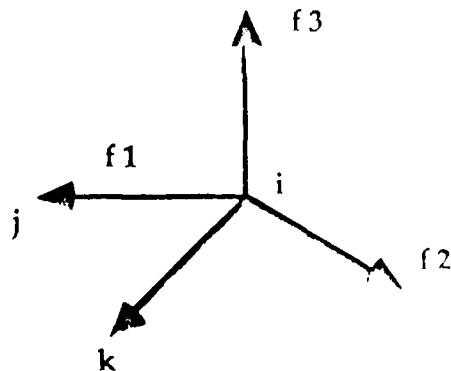
(c) Material Axes Orientation

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:

columns	variable	entry description
1 - 5	MORT	material axes orientation set number
6 - 10	NI	node number for point "i"
11- 15	NJ	node number for point "j"
16- 20	NK	node number for point "k"

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:



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 I6,1X,A4,1X,F10.0,12X,I6,I6)

<u>columns</u>	<u>variable</u>	<u>entry description</u>
1- 6	N	node at which force acts
8-11	LABEL	direction of force (in nodal coordinate system) FX, FY, or FZ
13-22	FORCE	value of the force
35-40	NEND	! 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)
41-46	INC	

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

Group VIII Displacement Boundary Conditions  
(Format I6,1X,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.

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

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.



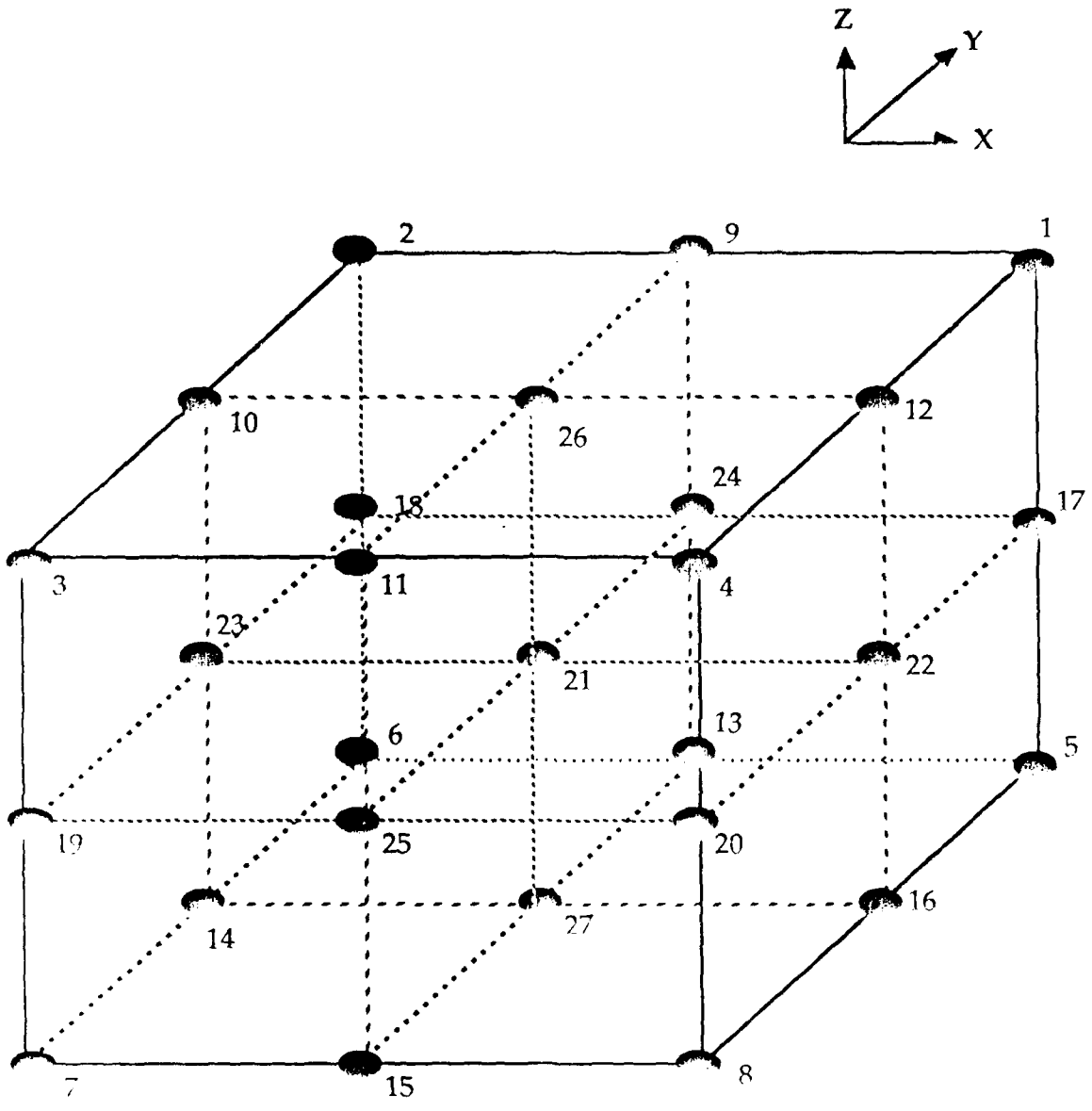


FIG. 2.3 STRESS OUTPUT LOCATION NUMBERS IN AN ELEMENT

### 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

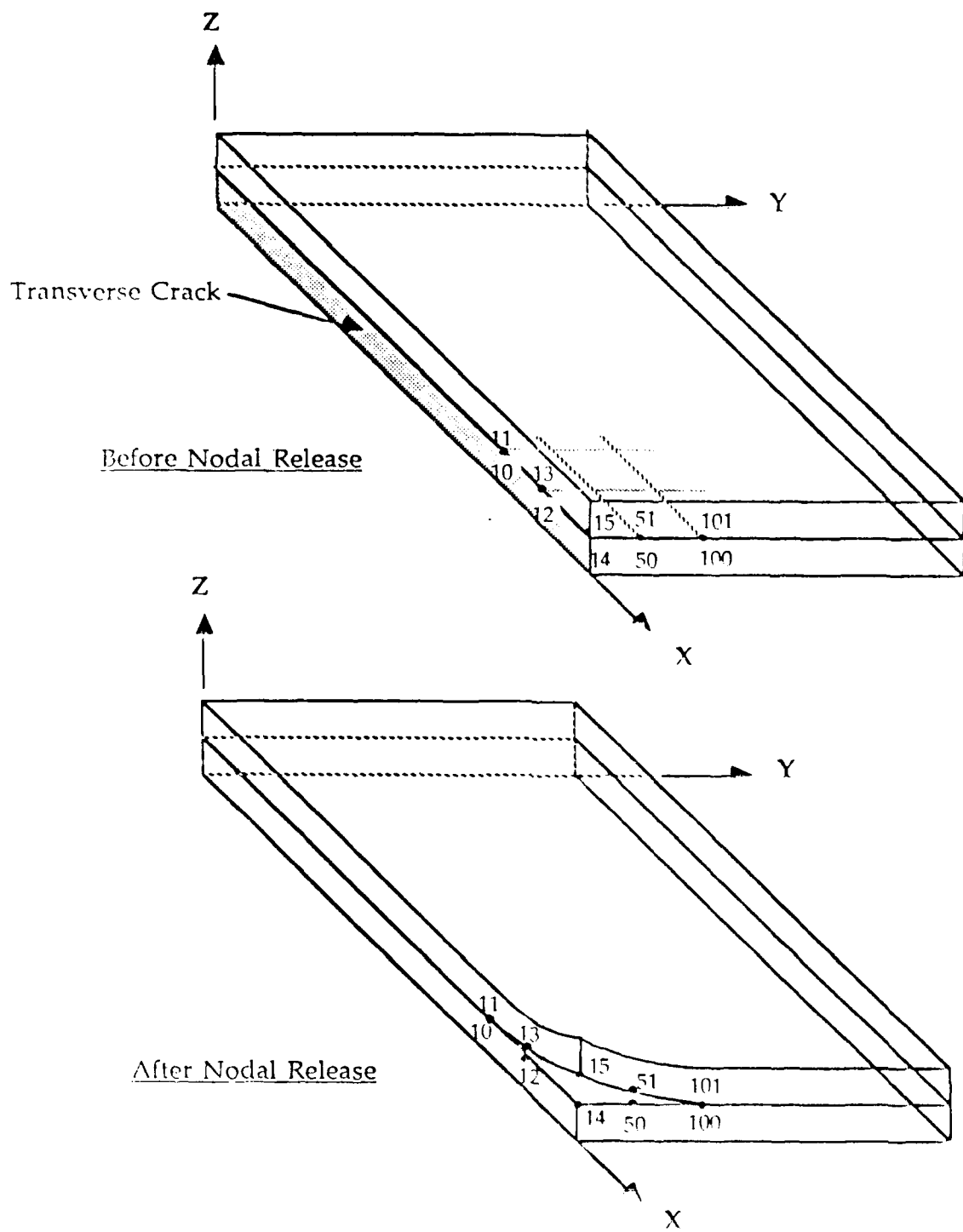


FIG. 3.1 DOUBLE NODE ARRANGEMENT FOR DELAMINATION SIMULATION

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:

#### I) 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:

NBC - node number  
LFIX - degree of freedom  
1 for x, 2 for y, 3 for z degree of freedom  
DSAVE - value of the specified displacement  
(=0, for nodes on that particular symmetric plane)

#### II) Details of the other double nodes' degrees of freedom

- a) NPAIR - number of pairs of all double nodes including those constrained on the symmetric plane
- b) There should be NPAIR following cards. Each card will give details of one pair of double nodes and the possible degrees of freedom after the nodes are open.

NPC1 - node number 1 of the pair  
NPC2 - node number 2 of the pair  
NPX - 0 or 1

NPX=1 signifies that the two nodes are constrained to have the same displacement  $U_x$ , 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-  
NPZ - 0 or 1 ! 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  $F_x$  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 this 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  
NFND ! elements from NBEG thru NFND  
This card should not be left blank. If 0,0 is entered stresses will not be printed.

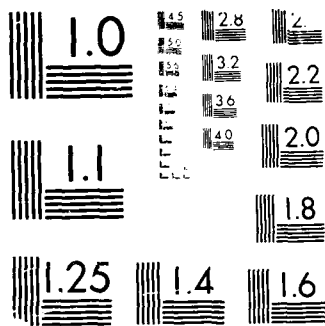
If another step of opening is desired, the above (a) and (b) will be repeated. This may be continued until all the

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.





MICROCOPY RESOLUTION TEST CHART  
ANSI AND ISO STANDARDS



## 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 upto 3000 nodes, and 100 pairs of double nodes. Should a problem involve finer mesh and 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 TNATM(\*,\*), TNAT(\*,\*), 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 KSAFOUT.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).

EYEX=30; EYEV=-30; EYEZ=30  
SCALE : 1 DIV =5000 psi

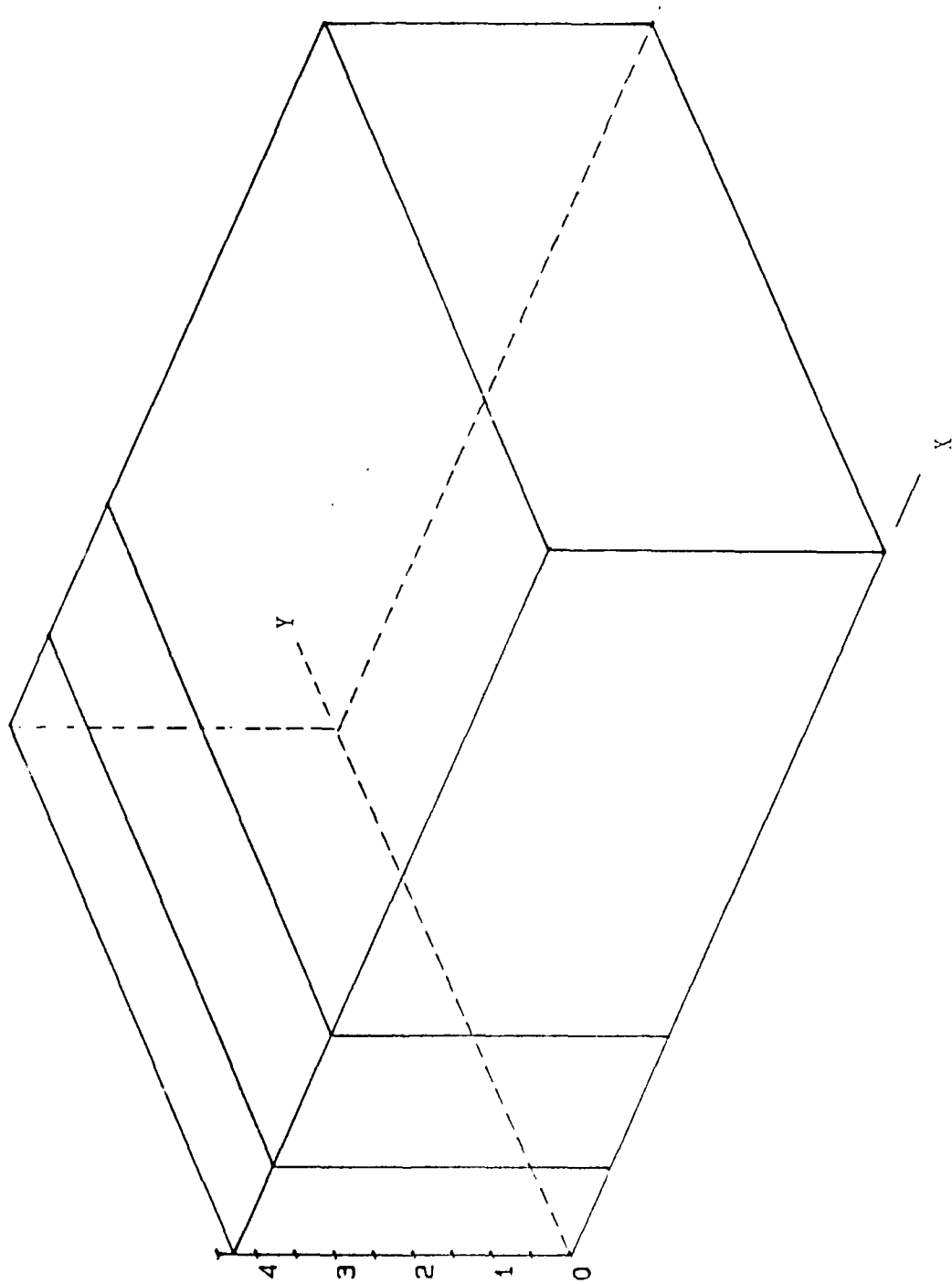


FIG. 5.1 NORMAL STRESS DISTRIBUTION ALONG Y DIRECTION IN 0° LAYER

EYEX=30; EYFY=-3-; EYEZ=30  
SCALE : 1 DIV. = 5 psi

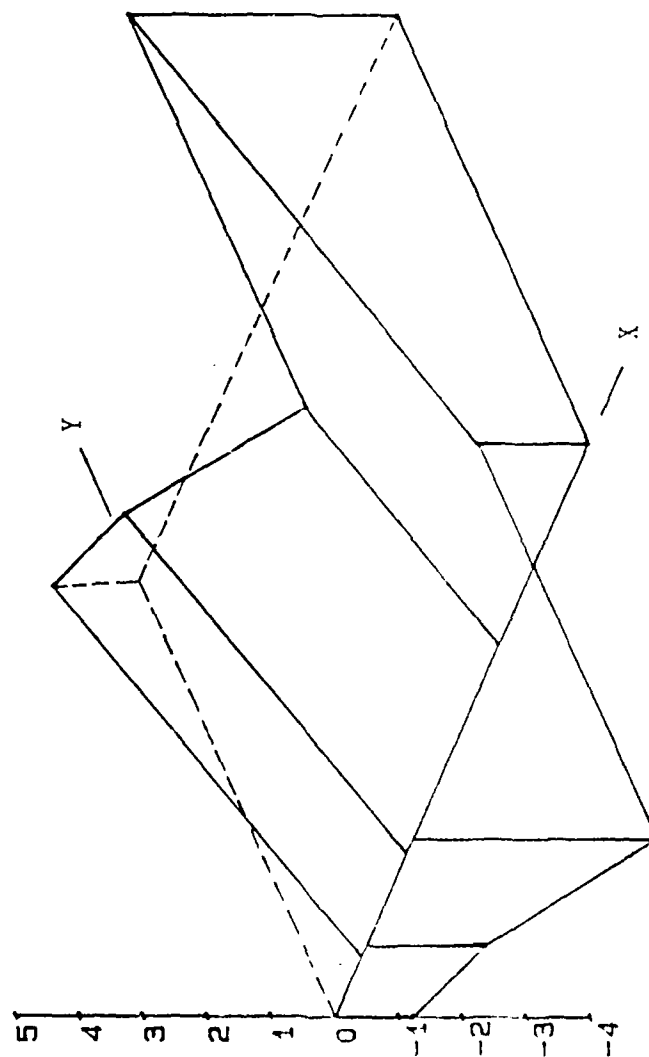


FIG. 5.2 INTERLAMINAR NORMAL STRESS DISTRIBUTION IN 0° LAYER



## 6 ILLUSTRATIVE EXAMPLE

### 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

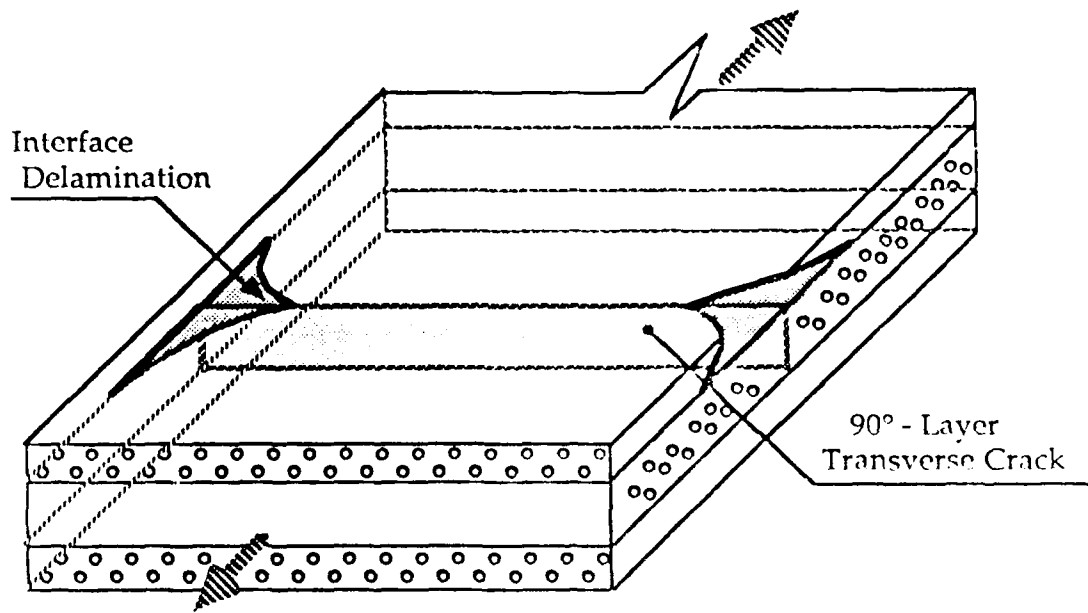
Material properties and loading information for each layer are furnished in the following manner:

Layer 1 (90° layer)

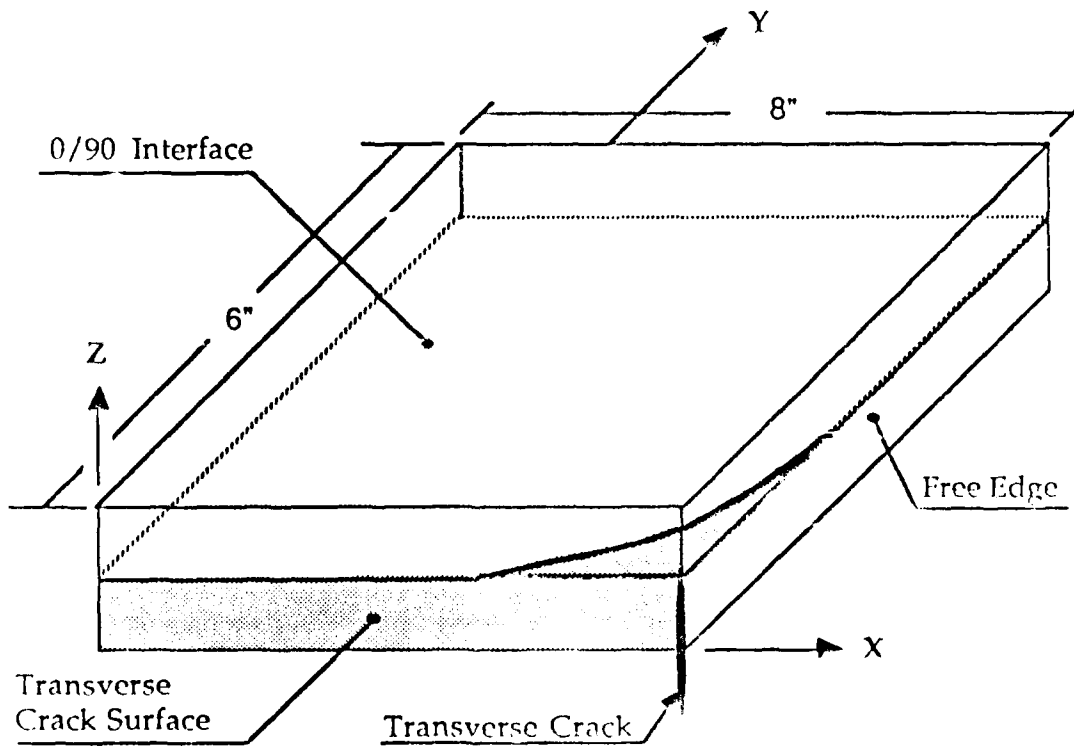
$$E_1 = 21.0 \times 10^6 \text{ psi}$$

$$E_2 = 1.7 \times 10^6 \text{ psi}$$

$$E_3 = 1.7 \times 10^6 \text{ psi}$$



(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

$$\begin{aligned}
\nu_{1t} &= 0.30 \\
\nu_{1z} &= 0.30 \\
\nu_{tz} &= 0.54 \\
G_{1t} &= 0.94 \times 10^6 \text{ psi} \\
G_{1z} &= 0.94 \times 10^6 \text{ psi} \\
G_{tz} &= 0.50 \times 10^6 \text{ psi} \\
\alpha_t &= 0.20 \times 10^{-6} / ^\circ\text{F} \\
\alpha_z &= 16.0 \times 10^{-4} / ^\circ\text{F} \\
\alpha_z &= 16.0 \times 10^{-4} / ^\circ\text{F}
\end{aligned}$$

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.

## 6.2 PREPROCESSOR INPUT DATA

Group I

---

8 node E1.[02/902]s; delam- MECH load: 5x3x5 MESH-man.inp (10/30/87)

---

In this first group the heading to be printed is given on one card.

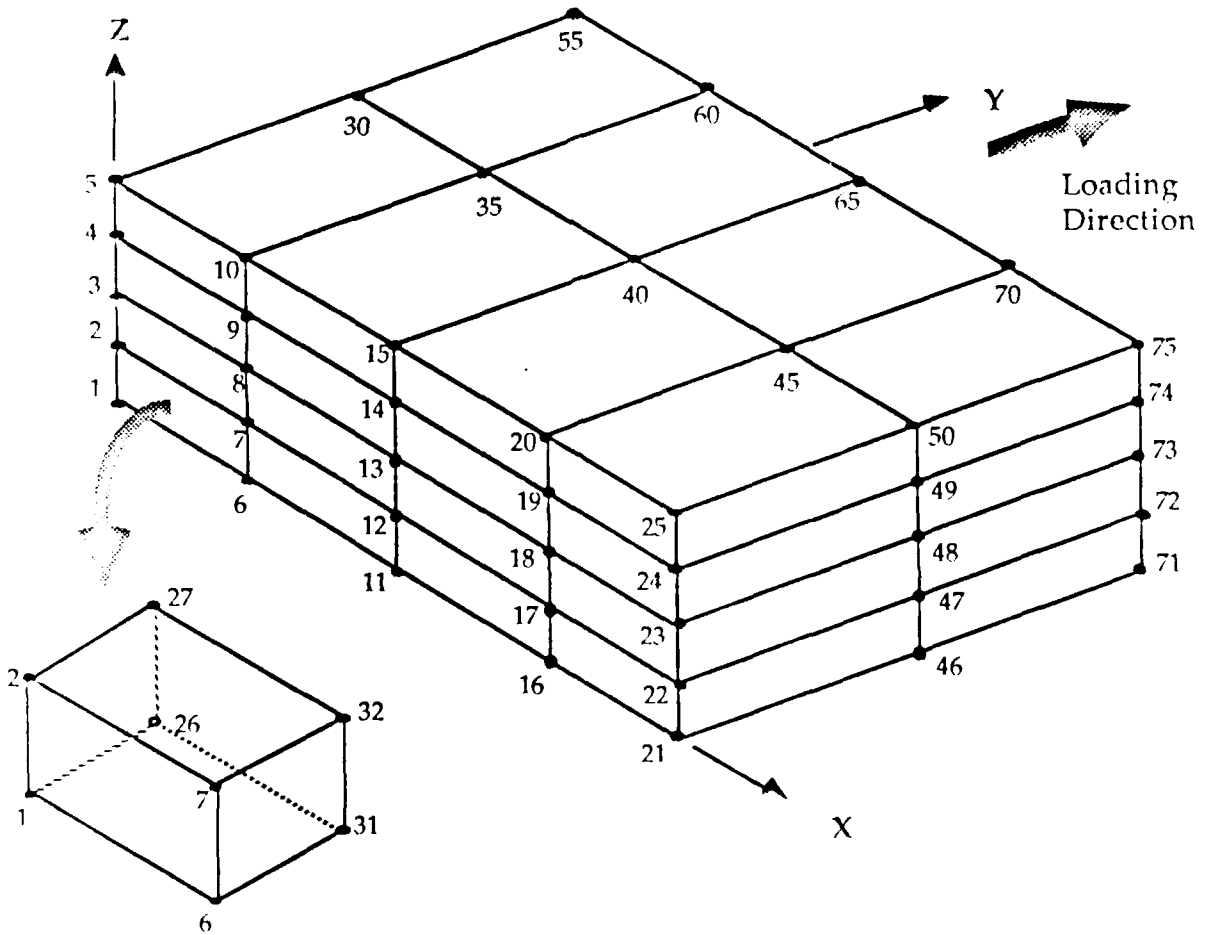


FIG. 6.2 INITIAL FINITE ELEMENT MESH WITHOUT DOUBLE NODES

Group II

---

8  
5, 3, 5, 0.0  
0.0, 2.0, 4.0, 6.0, 8.0  
0.0, 3.0, 6.0  
0.0, 0.5, 1.0, 1.5, 2.0

---

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.

Group III

---

1, 300.0, 75, 1  
-1, 0.0, 0, 0  
1, 300.0, 32, 1  
-1, 0.0, 0, 0  
1, 1, 32, 1  
-1, 0, 0, 0  
1, 1, 16, 1  
17, 2, 32, 1  
-1, 0, 0, 0  
2, 16  
18, 32  
-1, 0

---

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.

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

-----  
0,2  
-----

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

-----  
15  
3, 8, 13, 18, 23, 28, 33, 38, 43, 48, 53, 58, 63, 68, 73  
0.0, 8.0, 0.0, 6.0, 1.0, 2.0  
-----

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

---

EL	1	21.0E6
ET	1	1.7E6
EZ	1	1.7E6
NULT	1	0.3
NULZ	1	0.3
NUTZ	1	0.54
GLT	1	0.94E06
GLZ	1	0.94E06
GTZ	1	0.50E06
ALFL	1	0.2E-6
ALFT	1	0.16E-4
ALFZ	1	0.16E-4

-1

1 5 25 55  
 2 25 75 5  
 -1

---

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

---

-1

---

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

---

4	UY	0.0	24	5
5	UY	0.0	25	5
1	UX	0.0	51	25
2	UX	0.0	52	25
3	UX	0.0	53	25
4	UX	0.0	54	25
5	UX	0.0	55	25
1	UZ	0.0	21	5
26	HZ	0.0	46	5
51	UZ	0.0	71	5
51	UY	0.001	75	5

-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 ( $U_y$ ) 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
 9 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 ( $5 \times 2 - 10$ ) 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  $U_y=0$  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:

---

```

0 0 0
  1 32      ! Free format
27 0 2
27 28 1
27 28 3
  0 0 0
  1 32
21 0 2
21 22 1
21 22 3
57 58 1
57 58 2
57 58 3

```

```
0 0 0
  1 32
9999 9999 0
```

---

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 ( $F_y=0$ ). 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 ( $U_y=0$ ). 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





```

109 C
110 C      NODE GENERATION
111      READ(IRD,*) ,NONX,NONY,NONZ,RHOLE
112      READ (IRD,*)(XX(I),I=1,NONX)
113      READ (IRD,*)(YY(I),I=1,NONY)
114      READ (IRD,*)(ZZ(I),I=1,NONZ)
115
116 c--   correction for inside nodal coordinates for 21-node el.
117
118       if (ntype.eq.21) then
119         do 1444 i=1,nonx
120 1444   if (mod(i,2).eq.0) xx(i)=(xx(i-1)+xx(i+1))/2
121 C     type *,(xx(i),i=1,nonx)
122         do 1445 i=1,nony
123 1445   if (mod(i,2).eq.0) yy(i)=(yy(i-1)+yy(i+1))/2
124 C     type *,(yy(i),i=1,nony)
125         do 1446 i=1,nonz
126 1446   if (mod(i,2).eq.0) zz(i)=(zz(i-1)+zz(i+1))/2
127 C     type *,(zz(i),i=1,nonz)
128         end if
129
130       IF (RHOLE.GT.eps)CALL HOLE (RHOLE,NONX,NONY,NONZ,
131 .XX,YY,zz,X,Y,Z)
132       IF (RHOLE.GT.eps) GO TO 444
133
134       DO J=1,NONY
135       DO I=1,NONX
136       DO K=1,NONZ
137       N=(J-1)*NONX*NONZ+(I-1)*NONZ+K
138       X(N)=XX(I)
139       Y(N)=YY(J)
140       Z(N)=ZZ(K)
141       END DO
142       END DO
143       END DO
144
145 C-     END OF POLAR MESH GENERATION
146
147 444     NTON=NONX*NONY*NONZ
148
149 C--     MESH PLOTTING OPTION ON HP PLOTTER-----
150       WRITE (5,*) 'Do you need MESH plot Original nodes?..(Y/N)'
151       READ (5,55) MESH
152 55     FORMAT (A)
153       IF (MESH.EQ.'Y'.OR.MESH.EQ.'y') CALL
154 . MESHPL (X,Y,NONX,NONY,NONZ,MIP,1,ntype)
155       DO I=1,NTON
156       XMESH(I)=X(I)
157       YMESH(I)=Y(I)
158       END DO
159       IF (NN(NTOE,7).GT.NTON) THEN
160       DO 666 I=1,NTOE
161       DO 666 J=1,NITYPE
162 666     IF (NN(I,J).GT.NTON) NN(I,J)=NN(I,J)-NTON

```



```

163         END IF
164
165 148    READ(IRD,*) 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-----
175 C     ELEMENT GENERATION
176 C
177        N1=NONX
178        N2=NONY
179        N3=NONZ
180        N13=N1*N3
181
182        IF(NTYPE.EQ.8) THEN
183
184        NOS(3) =N3-1           ! third level gener.code
185        NINC(3)=1
186        NOE(3) =(N2-1)*(N1-1)
187        NOS(2) =N2-1           ! second level gener. code
188        NINC(2)=N1*N3
189        NOE(2) =N1-1
190        NOS(1) =N1-1           ! first level gener. code
191        NINC(1)=N3
192        NOE(1) =1
193
194        NRR(1)=2+  N3+  N13
195        NRR(2)=2+           N13
196        NRR(3)=2
197        NRR(4)=2+  N3
198        NRR(5)=1+  N3+  N13
199        NRR(6)=1           +  N13
200        NRR(7) =1
201        NRR(8)=1+  N3
202
203        ELSE IF(NTYPE.EQ.21) THEN
204
205        NOS(3) =(N3-1)/2       ! third level gener.code
206        NINC(3)=2
207        NOE(3) =(N2-1)*(N1-1)/4
208        NOS(2) =(N2-1)/2       ! second level gener. code
209        NINC(2)=2*N1*N3
210        NOE(2) =(N1-1)/2
211        NOS(1) =(N1-1)/2       ! first level gener. code
212        NINC(1)=2*N3
213        NOE(1) =1
214
215        NRR(1) =3+2*N3+2*N13
216        NRR(2) =3           +2*N13

```

```

217      NRR(3) =3
218      NRR(4) =3+2*N3
219      NRR(5) =1+2*N3+2*N13
220      NRR(6) =1      +2*N13
221      NRR(7) =1
222      NRR(8) =1+2*N3
223
224      NRR(9) =3+  N3+2*N13
225      NRR(10)=3      +  N13
226      NRR(11)=3+  N3
227      NRR(12)=3+2*N3+  N13
228      NRR(13)=1+  N3+2*N13
229      NRR(14)=1      +  N13
230      NRR(15)=1+  N3
231      NRR(16)=1+2*N3+  N13
232      NRR(17)=2+2*N3+2*N13
233      NRR(18)=2+2*N13
234      NRR(19)=2
235      NRR(20)=2+2*N3
236
237      NRR(21)=2+  N3+  N13
238      END IF
239      NO=0
240      DO 132 I=1,NTYPE
241 132  NN(NO+1,I)=NRR(I)
242
243 1   Perform element generation
244      DO 138 M=1,3
245      DO 138 K=2,NOS(M)
246      DO 138 I=1,NOE(M)
247      N=NO+I+NOE(M)*(K-1)
248      DO 1375 J=1,NTYPE
249 1375 NN(N,J)=NN(NO+I,J)+NINC(M)*(K-1)
250 138  continue
251      NO=N
252      NIOE=NO
253 1   DO 1415 IE=1,NIOE
254 1   TYPE *
255 1   TYPE *, ' **** ELEMENT ' ,IE
256 1415 TYPE 105, (NN(IE,J),J=1,NTYPE)
257 105  FORMAT(8I5)
258
259
260 1   Read element stress free temperature
261 158  READ(IRD,*) N,TEMP,NEND,INC
262      IF(N.EQ.-1) GO TO 159
263      DO 1585 IE=N,NEND,INC
264 1585 IZ(IE)=TEMP
265      GO TO 158
266 157  continue
267
268 1   Read element material identification number
269 1458 READ(IRD,*) N,MATRL,NEND,INC
270      IF(N.EQ.-1) GO TO 1459

```

```

271      DO 14585 IE=N,NEND,INC
272 14585 IDM(IE)=MATRL
273      GO TO 1458
274 1459 continue
275
276 C ---MAXES(I).....MATEL. AXIS ORIENT.:
277 !      Read element material axis orientation identification number
278 1658 READ(IRD,X) N,MORTI,NEND,INC
279      IF(N.EQ.-1) GO TO 1659
280      DO 16585 IE=N,NEND,INC
281 16585 MAXES(IE)=MORTI
282      GO TO 1658
283 1659 continue
284 !      Read if the stiffness matrix of an element (or a group of elements)
285 !      is same as the previous element
286
287      DO 145 I=1,NIOE
288 145  MAT(I)=0
289
290 176  READ(IRD,X) M1,M2
291 c   WRITE(111,X) 'M1,M2',M1,M2
292      IF(M1.EQ.-1)GO TO 1765
293      DO 1762 I=M1,M2
294 1762 MAT(I)=1
295      GO TO 176
296 1765 continue
297 C-----
298 C   BOUNDARY CONDITION GENERATION
299 c   Following statement are to fix all rotations when
300 c   only translational d.o.f. (eltype #8 is used)
301      DO 2010 I=1,NION
302      DO 2010 J=1,6
303      IF(J.LE.3) IX(I,J)=0
304      IF(J.GE.4) IX(I,J)=1
305 2010 continue
306
307 !      While generating 21-node elements some nodes are at the center of faces
308 !      These degrees of freedom will be removed below:
309
310 !      Nodes not used in defining any element are found
311 !      and all its d.o.f. set equal to 1 (i.e., eliminated)
312      DO IE=1,NIOE
313      DO IN=1,NTYPE
314      II=NN(IE,IN)
315      IX(II,6)=10      !to identify which nodes are used
316      end do
317      end do
318
319      DO IND=1,NION
320      IF(IX(IND,6).EQ.10)THEN
321      IX(IND,6)=1
322      ELSE
323      DO IDG=1,6
324      IX(IND,IDG)=1

```

```

325     END DO
326     END IF
327     end do
328 !!!!! FOR DOUBLING NODES !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
329     do ip=1,nton
330     mip(ip,2)=0
331     end do
332 !     For Splitting.....
333     NSD=0
334     READ(IRD,*) NSD, IDIR      ! No. of nodes to be doubled, Direction Vector
335     IF(NSD.NE.0) THEN
336     NSD1=0
337 686   READ (IRD,*) N,NEND, INC
338     IF (N.EQ.-1) GO TO 688
339     DO 687 I=N,NEND, INC
340     NSD1=NSD1+1
341 687   NONS(NSD1)=I
342     GO TO 686
343 !     READ(IRD,*) (NONS(I),I=1,NSD)      !original node # to be doubled
344 688   IF (NSD1.NE.NSD) NSD=NSD1          !check on total no. of d.n's
345 c--   to delete face center no. from double node list.....
346     IF (NTYPE.EQ.21) CALL DELETE(NONX,NONY,NONZ,NSD,NONS)
347     do i=1,nsd
348     ia=nons(i)
349     mip(ia,2)=1
350     end do
351     end if
352     NTD=0
353     READ(IRD,*) NTD      ! Number of nodes to be doubled
354
355 !     Define the zone of delamination
356 c     store coord. of the nodes to be doubled
357     IF(NTD.NE.0) THEN
358     READ(IRD,*) (NOND(I),I=1,NTD)      !original node # to be doubled
359 c--   to delete face center no. from double node list.....
360     IF (NTYPE.EQ.21) call delete(nonx,nony,nonz,ntd,nond)
361     DO I=1,NTD
362     IA=NOND(I)
363     MIP(IA,2)=1
364     END DO
365     READ(IRD,*) XL,XU,YL,YU,ZL,ZU
366     DO I=NTD,1,-1
367     XD(I)=X(NOND(I))
368     YD(I)=Y(NOND(I))
369     ZD(I)=Z(NOND(I))
370     END DO
371     END IF
372 c     if (ntd+nsd.ne.0) then
373     IP=0
374     DO I=1,NTDN
375     IP=IP+1
376     MIP(I,1)=IP
377     IF(MIP(I,2).EQ.1) THEN
378     IP=IP+1

```

```

379     MIP(I,2)=IP
380     END IF
381     END DO
382 C     end if
383     IF (NSD.NE.0)
384     .call split (NTON,nsd,idir,nons,mip,x,y,z,I,ntype,nos,NN,ntoe)
385 !     For Delamination.....
386 C     DO I=1,NTON
387 C     TYPE *,I,(MIP(I,J),J=1,2)
388 C     END DO
389
390     IF(NTD.EQ.0) GO TO 899
391 C-    CORRECTIONS FOR DOUBLE NODES IN planes normal to x, y or z -directions
392 C    FIND KND (THE # OF THE NODE CURRENTLY TO BE DOUBLED)
393
394     DO 525 I=1,NTD
395     DO J=NOND(I),NTON
396
397     DIST=SQRT((X(J)-XD(I))**2+ (Y(J)-YD(I))**2+ (Z(J)-ZD(I))**2)
398     IF(DIST.LT.0.00001) THEN
399     KND=J
400
401 C    BEGIN CHANGING NODE NUMBERS & COORD.
402
403 C    Change element node numbers if it is > knd by adding 1 to it
404     DO 346 JE=1,NTOE
405     DO 345 K=1,NTYPE
406 345     IF (NN(JE,K).GT.KND) NN(JE,K)=NN(JE,K)+1
407 346     CONTINUE
408 C    Also move coords. downstream by one slot and also assign (knd+1)th
409 !    same as (knd)th
410     DO M=NTON,KND,-1
411     X(M+1)=X(M)
412     Y(M+1)=Y(M)
413     Z(M+1)=Z(M)
414     T(M+1)=T(M)
415     END DO
416
417 C--   Allocate nodes of the pair to the appropriate elements depending
418 !    on which side of the double nodes' plane they (element) lie
419     nface=NTYPE
420     DO 3455 K=1,NTOE
421
422     IF(NTYPE.EQ.8) THEN      ! Find the coordinates of the center of the
423     XC=0.                  ! element to determine if it belongs
424     YC=0.                  ! to the delamination zone
425     ZC=0.
426     DO IA=1,8
427     III=NN(K,IA)
428     XC=XC+X(III)
429     YC=YC+Y(III)
430     ZC=ZC+Z(III)
431     END DO
432     XC=XC/8.

```

```

433      YC=YC/8.
434      ZC=ZC/8.
435      ELSE
436      III=NN(K,21)
437      XC=X(III)
438      YC=Y(III)
439      ZC=Z(III)
440      END IF
441      IF(K.EQ.32) THEN
442      TYPE *, ' ... ELEMENT # ... ', K
443      TYPE *, ' - CENTER:', XC,YC,ZC
444      TYPE *, ' .. BEFORE ...'
445      TYPE *, (NN(K,M),M=1,NEACE)
446      END IF
447      DO 3453 M=1,nface
448      IF (NN(K,M).EQ.KND
449      .and.(XC.GT.XL.AND.XC.LT.XU)
450      .and.(YC.GT.YL.AND.YC.LT.YU)
451      .and.(ZC.GT.ZL.AND.ZC.LT.ZU))
452
453      . NN(K,M)=KND+1
454 3453      END DO
455 3455      END DO
456
457
458      NTON=NTON+1
459
460      go to 525      ! so that the modifications are done only once
461      ! for each double node
462      END IF      ! (DIST.LT.0.00001)
463      END DO      !J - LOOP
464 525      END DO      !I - loop
465
466      899 CONTINUE      ! SKIP IF NTD=0
467
468      write(5,*)'...Do You Need Mesh with Double Nodes (Y/N).. '
469      READ (5,S5) MESH
470      IF (MESH.EQ.'Y'.OR.MESH.EQ.'y') CALL
471      . MESHPL (XMESH,YMESH,NONX,NONY,NONZ,MIP,2,otype)
472
473      IF (MESH.EQ.'Y'.OR.MESH.EQ.'y')
474      .write(5,*)'...Do You Want to Continue (Y/N).. '
475      IF (MESH.EQ.'Y'.OR.MESH.EQ.'y')
476      .READ (5,S5) MESH
477      IF (MESH.EQ.'Y'.and.MESH.EQ.'y') stop
478
479      !!!!!!! ABOVE ARE FOR DOUBLING NODES !!!!!!!
480
481      Material properties
482      OPEN (UNIT=9,FILE='KSAPIN.DAT',STATUS='NEW')      !!!!!
483
484      nummat=0
485 500      read(IRD,S50) A,I,U
486      if(i.gt.nummat) nummat=i

```

```

487 550  FORMAT(A4,I4,4X,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.'NULT') ANU12(I)=V*E22(I)/E11(I)
492      IF(A.EQ.'NUTZ') ANU23(I)=V*E33(I)/E22(I)
493      IF(A.EQ.'NULZ') ANU13(I)=V*E33(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.'ALFL') ALP1(I)=V
498      IF(A.EQ.'ALFT') ALP2(I)=V
499      IF(A.EQ.'ALFZ') ALP3(I)=V
500      IF(A.EQ.'-1') 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      MORT(I)=NR(1)
508      NI(I)=NR(2)
509      NJ(I)=NR(3)
510      NK(I)=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 *, '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(X(LLL))-ABS(X(I))).GT.1.E-06) GO TO 994
537     IF (ABS(X(LLL)-X(I)).GT.1.E-06) GO TO 994
538     IF ((ABS(Y(LLL))-ABS(Y(I))).GT.1.E-06) GO TO 994
539     IF (ABS(Y(LLL)-Y(I)).GT.1.E-06) GO TO 994
540     IF ((Z(LLL)-Z(I)).LT.0.0) GO TO 994

```

```

541      LQ=I      !y & z coord. same as J, xcoord. diff.
542      LX=J      !y & z coord. same as I, xcoord. diff.
543      LY=K      !.ne.I or J, x & z coord. same as I and y is diff.
544      LZ=LLL    !.ne. I,J or K, x & y coord. same as I and z diff.
545      GO TO 995
546 994      CONTINUE
547 993      CONTINUE
548 992      CONTINUE
549 991      CONTINUE
550 995      CONTINUE
551
552 C-- READ & GENERATE CONCENTRATED LOAD DATA:
553
554          NTBF=0
555          NINCC=1
556 640      READ(IRD,710) N1,A,V,N2,'INCC
557          IF(N1.EQ.-1) GO TO 699
558
559          DO I=N1,N2,NINCC
560          NTBF=NTBF+1
561          NBF(NTBF)=I
562          IF(A.EQ.' EX') EX(NTBF)=V
563          IF(A.EQ.' FY') FY(NTBF)=V
564          IF(A.EQ.' FZ') FZ(NTBF)=V
565          END DO
566          GO TO 640
567 699      CONTINUE
568 C
569 C      READ & GENERATE DISPL. B.C.E. DATA
570 C
571          NTBD=0
572          NINCC=1
573 700      READ(IRD,710) N1,A,V,N2,NINCC
574 710      FORMAT(16,1X,A4,1X,F10.0,12X,2I6)
575          IF(N1.EQ.-1) GO TO 799
576
577
578          IF(V.EQ.0) THEN
579          DO I=N1,N2,NINCC
580          IF(A.EQ.' UX') IX(I,1)=1
581          IF(A.EQ.' UY') IX(I,2)=1
582          IF(A.EQ.' UZ') IX(I,3)=1
583          END DO
584
585          ELSE
586          NELTYP=2
587          V1=V
588          V2=0
589          JD=1
590          JR=0
591          IF(A.EQ.' UX') then
592          L(1)=L0
593          L(2)=LY !L0 -> LY ALONG Y DIR.
594          L(3)=L0 !L0 -> LZ ALONG Z DIR.

```



```

595      L(4)=LZ
596      else if(A.EQ.' UZ') then
597      L(1)=L0
598      L(2)=LX
599      L(3)=L0
600      L(4)=LY
601      else if(A.EQ.' UY') then
602      L(1)=L0
603      L(2)=LZ
604      L(3)=L0
605      L(4)=LX
606      END IF
607
608      DO I=N1,N2,NINCC
609      NTBD=NTBD+1
610      NBD(NTBD)=I
611      DO J=1,4
612      ND(NTBD,J)=L(J)
613      END DO
614      VD(NTBD)=V1
615      VR(NTBD)=V2
616      KD(NTBD)=JD
617      KR(NTBD)=JR
618      END DO
619      end if
620      GO TO 700
621 799 CONTINUE
622
623 C writing double nodes to for010.dat, and renumbering displacement
624 c boundary elements and force boundary node number
625
626      do i=nton,1,-1
627      do ic=1,2
628      ip=mip(i,ic)
629      if(ip.ne.0) then
630      do id=1,6
631      ix(ip,id)=ix(i,id)
632      end do
633      end if
634      end do
635      end do
636
637      if(ntd+nsd.ge.1) then
638      DO K=1,NTBD
639
640      NBDK1=MIP(NBD(K),1)
641      NBDK2=MIP(NBD(K),2)
642      NBD(K)=NBDK1
643      KR(K)=MIP(KR(K),1)
644      KD(K)=MIP(KD(K),1)
645
646      IF(NBDK2.NE.0) THEN
647      ntbd=ntbd+1
648      NBD(NTBD)=NBDK2

```

```

649      KR(NTBD)=KR(K)
650      KD(NTBD)=KD(K)
651      VD(NTBD)=VD(K)
652      VR(NTBD)=VR(K)
653      DO M=1,4
654      ND(NTBD,M)=ND(K,M)
655      END DO
656      END IF
657      END DO
658
659 C    RENUMBER NI,NJ,NK (ORIENT. DEFINITION)
660      DO K=1,NORTHO
661      NI(K)=MIP(NI(K),1)
662      NJ(K)=MIP(NJ(K),1)
663      NK(K)=MIP(NK(K),1)
664      END DO
665
666      end if      ! ntd.ge.1
667
668
669 C    Renumbering the force boundary conditions and adding
670 C    new force boundary conditions of double nodes if necessary
671
672      DO K=1,NTBF
673      if(mip(nbf(k),2).ne.0) THEN
674      WRITE (5,* ) ' ....D.N at force b.c; NODE no..', NBF(K)
675      STOP  'STOPPING due to double node at force boundary cond.'
676      END IF
677      NBF(K)=mip(NBF(K),1)
678      END DO
679 C
680 C    OUTPUT NODE DATA
681 C
682      WRITE(9,1100) (HED(I),I=1,18)
683 1100  FORMAT(18A4)
684      IADOF=NTD*6
685      IADOF==ADDITIONAL D.O.F. DUE TO DOUBLE NODES & DISP.BCS.
686      WRITE(9,1101) NTON,NELTYP,IADOF
687 1101  FORMAT(15,15,4X,'1',14X,'0',15X,15)
688
689
690      WRITE(9,11015) (IX(1,J),J=1,6),X(1),Y(1),Z(1),I(1)
691 11015  FORMAT(4X,'1C',14,5I5,3F10.4,5X,F10.0)
692      KN=0
693      WRITE(9,1102) 2,(IX(2,J),J=1,6),X(2),Y(2),Z(2),KN,I(2)
694      KNH=0
695      DO I=3,NTON-1
696      IXM1=ix(i,1)-ix(i-1,1)
697      IXM2=ix(i,2)-ix(i-1,2)
698      IXM3=ix(i,3)-ix(i-1,3)
699      IXM4=ix(i,4)-ix(i-1,4)
700      IXM5=ix(i,5)-ix(i-1,5)
701      IXM6=ix(i,6)-ix(i-1,6)
702      DXM=X(I)-X(I-1)

```

```

703     DYM=Y(I)-Y(I-1)
704     DZM=Z(I)-Z(I-1)
705     DTM=T(I)-T(I-1)
706     IXP1=ix(i+1,1)-ix(i,1)
707     IXP2=ix(i+1,2)-ix(i,2)
708     IXP3=ix(i+1,3)-ix(i,3)
709     IXP4=ix(i+1,4)-ix(i,4)
710     IXP5=ix(i+1,5)-ix(i,5)
711     IXP6=ix(i+1,6)-ix(i,6)
712     DXP=X(I+1)-X(I)
713     DYP=Y(I+1)-Y(I)
714     DZP=Z(I+1)-Z(I)
715     DTP=T(I+1)-T(I)
716     KN=0
717     IF (IXM1.EQ. IXP1.AND. IXM2.EQ. IXP2.AND. IXM3.EQ. IXP3.AND.
718     .   IXM4.EQ. IXP4.AND. IXM5.EQ. IXP5.AND. IXM6.EQ. IXP6.AND.
719     .   DXM.EQ. DXP.AND. DYM.EQ. DYP.AND. DZM.EQ. DZP.AND.
720     .   DTM.EQ. DTP) KN=1
721     IF (KN.EQ.0)
722     .WRITE(9,1102) I, (IX(I,J),J=1,6),X(I),Y(I),Z(I),KNM,T(I)
723     .KNM=KN
724 1102  FORMAT(15,6I5,3F10.4,15,F10.0)
725     .END DO
726     .i=nton
727     .WRITE(9,1102) I, (IX(I,J),J=1,6),X(I),Y(I),Z(I),KN,T(I)
728 C
729     .if (neltyp.gt.1) then      !      OUTPUT FOR B.C.E. #7
730 C
731     .WRITE(9,1201) NTBD
732 1201  FORMAT(4X,'7',15/8X,'1.')
733     .DO I=1,NTBD
734     .WRITE(9,1202) NBD(I), (ND(I,J),J=1,4),KD(I),KR(I),VD(I),VR(I)
735 1202  FORMAT(7I5,5X,2F10.7,' 0.100E+21 ')
736     .END DO
737     .end if
738 C
739 C     OUTPUT ELEMENT DATA
740 C
741     .MAXNOD=NTYPE
742     .NOPSET=1      !Number of sets of data requesting stress output
743     .WRITE(9,1300) NTOE,NUMMAT,NORTHO,MAXNOD,NOPSET,INTRS,INIT
744 1300  FORMAT(4X,'8',15,15,4X,'0',15,5X,15,3I5)
745
746 c     I=1
747     .do i=1,nummat
748     .WRITE(9,1301) I,TZ(I),E11(I),E22(I),E33(I),ANU12(I),ANU13(I),
749     .ANU23(I),G12(I),G13(I),G23(I),ALP1(I),ALP2(I),ALP3(I)
750 1301  FORMAT(15,4X,'1',20X,'AXIS#1==0-LAYER; AXIS#2==90-LAYER.' /
751     .*f10.0,3f10.0,3f10.4/3i10.0,3f10.7)
752 ccc     .WRITE(9,13010) T(I),E11(I),E22(I),E33(I),ANU12(I),ANU13(I),
753 ccc     .ANU23(I),G12(I),G13(I),G23(I),ALP1(I),ALP2(I),ALP3(I)
754 13010  FORMAT(F10.0,3F10.0,3F10.4/3F10.0,3F10.7)
755     .END DO
756

```

```

757      DO I=1,NORTHO
758      WRITE(9,13011) MORT(I),NI(I),NJ(I),NK(I)
759 13011 FORMAT(4I5)
760      END DO
761
762 13005 FORMAT(4I5/4I5/3I5)
763
764      READ(IRD,*) LOC1,LOC2,LOC3,LOC4,LOC5,LOC6,LOC7
765      WRITE(9,13008) LOC1,LOC2,LOC3,LOC4,LOC5,LOC6,LOC7
766 13008 FORMAT(7I5)
767      TA=1.0
768      IF(T(1).EQ.TZ(1)) TA=0.0
769      WRITE (9,13012) TA
770 13012 FORMAT(///F10.0/)
771      IOP=1      ! I.D.# OF STRESS OUTPUT LOCATION SET
772      ISKIP0=0
773
774      DO I=1,NTOE
775      KGM=0
776      if(i.gt.1.and.
777      .IDM(I).EQ.IDM(i-1).AND.MAXES(I).EQ.MAXES(i-1).AND.IOP.EQ.IOP
778      .AND.TZ(I).EQ.TZ(i-1).AND.MAT(I).EQ.MAT(i-1)) then
779      KGM1=NN(I,7)-NN(I-1,7)
780      KGM2=NN(I,8)-NN(I-1,8)
781      KGM3=NN(I,5)-NN(I-1,5)
782      KGM4=NN(I,6)-NN(I-1,6)
783      KGM5=NN(I,3)-NN(I-1,3)
784      KGM6=NN(I,4)-NN(I-1,4)
785      KGM7=NN(I,1)-NN(I-1,1)
786      KGM8=NN(I,2)-NN(I-1,2)
787      KGMX=MAX0(KGM1,KGM2,KGM3,KGM4,KGM5,KGM6,KGM7,KGM8)
788      KGMMN=MINO(KGM1,KGM2,KGM3,KGM4,KGM5,KGM6,KGM7,KGM8)
789      IF(KGMX.EQ.KGMMN) KGM=KGMX
790      end if
791
792      KGP=0
793      if(i.lt.ntoe.and.
794      .IDM(I).EQ.IDM(I+1).AND.MAXES(I).EQ.MAXES(I+1).AND.IOP.EQ.IOP
795      .AND.TZ(I).EQ.TZ(I+1).AND.MAT(I).EQ.MAT(I+1)) then
796      KGP1=NN(I+1,7)-NN(I,7)
797      KGP2=NN(I+1,8)-NN(I,8)
798      KGP3=NN(I+1,5)-NN(I,5)
799      KGP4=NN(I+1,6)-NN(I,6)
800      KGP5=NN(I+1,3)-NN(I,3)
801      KGP6=NN(I+1,4)-NN(I,4)
802      KGP7=NN(I+1,1)-NN(I,1)
803      KGP8=NN(I+1,2)-NN(I,2)
804      KGPMX=MAX0(KGP1,KGP2,KGP3,KGP4,KGP5,KGP6,KGP7,KGP8)
805      KGPMN=MINO(KGP1,KGP2,KGP3,KGP4,KGP5,KGP6,KGP7,KGP8)
806      IF(KGPMX.EQ.KGPMN) KGP=KGPMX
807      end if
808      ISKIP=0
809 cc  ISKIP=1
810      kgz=0

```

```

811         if(iskipo.eq.0.and.kgp.gt.0) kgz=kgp
812         il=i
813 c       WRITE(9,* ) 1,ISKIPO,ISKIP,KGM,KGP
814         IF(ISKIP.EQ.0)THEN
815         WRITE(9,13025) I1, IDM(I1),MAXES(I1), IOP,IZ(I1),kgz
816         .,MAT(I1),(NN(I1,JX),JX=1,NTYPE)
817         END IF
818 13025  FORMAT(15,I15,2I5,F10.0,15,10X,15/16I5/8I5)
819         ISKIP=ISKIP
820         KGM=KGM
821         END DO
822 1302  FORMAT(15,I15,2I5,F10.0,15X,15/8I5)
823
824         NBMN=0
825         DO J=1,NTBF
826         NBM=99999
827         DO I=1,NTBF
828         IF(NBF(I).LT.NBM.AND.NBF(I).GT.NBMN) THEN
829         NBM=NBF(I)
830         IM=I
831         END IF
832         END DO
833         ID1(J)=IM
834         NBMN=NBM
835         END DO
836
837         DO J=1,NTBF
838         I=ID1(J)
839         WRITE(9,1303) NBF(I),EX(I),FY(I),FZ(I)
840         END DO
841 1303  FORMAT(15,4X,'1',3F10.4)
842
843         WRITE(9,1305)
844 1305  FORMAT(/8X,'1.')
845
846         WRITE (9,1306)
847 1306  FORMAT(1X,'0'//1X,'0'//1X,'0 0 0'//1X,'9999 9999 0')
848         WRITE (10,*) '
849         WRITE (10,*) ' ....For Crack Simulation...'
850         WRITE (10,*) ' ORIG. NODES  DOUBLE NODES '
851         WRITE (10,*) ' -----'
852         ntono=nton-ntd-nsd
853         write (9,8128) ntd
854         do 850 i=1,ntono
855         if (mip(i,2).ne.0) then
856         do 860 j=1,nsd
857         if (i.eq.nons(j)) write(10,8126) I,mip(i,1),mip(i,2)
858         if (i.eq.nons(j)) go to 850
859 860   continue
860         write(9,8125) mip(i,1),mip(i,2),I
861         end if
862 850   continue
863 8125  format(2i5,7h 1 1 1, ' ! ',i5)
864 8126  format(2X,15,3X,2I5)

```

```

865 c      do i=1,ntono
866 c      if(mip(i,2).ne.0) then
867 c--    writing in the order they are given in the input file----
868      do i=1,ntd
869      ii=nond(i)
870      do j=1,3
871      write(9,8128) mip(ii,1),mip(ii,2),j,ii
872      end do
873 8128 format(2i5,i3,5X,'      !      ',15)
874 c      end if
875      end do
876
877      CLOSE (UNIT=9)
878      STOP
879      END
880 c
881 c{=====
882      subroutine split (NTON,ntd,idir,nond,mip,x,y,z,I,
883      .ntype,nos,NN,ntoe)
884      dimension nond(1),mip(4000,2),iface(9),jface(9),nos(1),I(1)
885      .,xd(400),yd(400),zd(400),x(1),y(1),z(1),nn(3000,21)
886      NFF=4
887      IF (NTYPE.EQ.21) NFF=9
888 C      STORE COORD. OF THE NODES TO BE DOUBLED
889 C-  CORRECTIONS FOR DOUBLE NODES IN x or y or z -direction--
890
891      DO 819 I=1,4
892      IFACE(I)=I+4
893 819 JFACE(I)=I
894      IFACE(5)=13
895      IFACE(6)=14
896      IFACE(7)=27
897      IFACE(8)=16
898      IFACE(9)=15
899      JFACE(5)=9
900      JFACE(6)=10
901      JFACE(7)=26
902      JFACE(8)=12
903      JFACE(9)=11
904      NIELR=NOS(1)*NOS(2) !!NIELR--TOTAL NUMBER OF ELEM. IN AN ELE-LAYER
905      IF (IDIR.LT.1.OR.IDIR.GT.3) IDIP=3
906      IF (IDIR=2) 901,902,903
907 901 IFACE(1)=2
908      IFACE(2)=3
909      IFACE(3)=6
910      IFACE(4)=7
911      JFACE(1)=1
912      JFACE(2)=4
913      JFACE(3)=5
914      JFACE(4)=8
915      IFACE(5)=10
916      IFACE(6)=18
917      IFACE(7)=23
918      IFACE(8)=19

```

```

919     IFACE(9)=14
920     JFACE(5)=12
921     JFACE(6)=17
922     JFACE(7)=22
923     JFACE(8)=20
924     JFACE(9)=16
925     NTELR=1
926     GO TO 903
927 902   IFACE(1)=3
928       IFACE(2)=4
929       JFACE(1)=2
930       JFACE(2)=1
931       JFACE(3)=6
932       JFACE(4)=5
933       IFACE(5)=11
934       IFACE(6)=19
935       IFACE(7)=25
936       IFACE(8)=20
937 C     IFACE(9)=15
938 C     JFACE(5)=9
939       JFACE(6)=18
940       JFACE(7)=24
941       JFACE(8)=17
942       JFACE(9)=13
943       NTELR=NOS(1)
944 903   CONTINUE
945
946     DO 701 I=1,NTD-1
947       ICHANGE=0
948     DO 702 J=1,NTD-I
949       JJ=J+1
950       IF (NOND(J).LT.NOND(JJ)) GO TO 702
951       ICHANGE=1
952       AA=NOND(J)
953       NOND(J)=NOND(JJ)
954       NOND(JJ)=AA
955 702   CONTINUE
956       IF (ICHANGE.EQ.0) GO TO 703
957 701   CONTINUE
958
959 703   CONTINUE
960     DO I=1,NTD
961       XD(I)=X(NOND(I))
962       YD(I)=Y(NOND(I))
963       ZD(I)=Z(NOND(I))
964     END DO
965 C-- CORRECTIONS DOUB. NODES. ENDS.--
966
967 C     FIND KND (THE # OF THE NODE CURRENTLY TO BE DOUBLED)
968     DO 830 I=1,NTD
969       DO 820 J=NOND(I),NTON
970         DIF=SQRT((X(J)-XD(I))**2+(Y(J)-YD(I))**2+
971             .(Z(J)-ZD(I))**2)
972 C     IF(X(J).NE.XD(I)) GO TO 820

```

```

973 C      IF(Y(J).NE.YD(I)) GO TO 820
974 C      IF(Z(J).NE.ZD(I)) GO TO 820
975      IF (DIF.GT.1.0E-13) GO TO 820
976      KND=J
977      GO TO 825      ! GET OUT OF J LOOP
978 820 CONTINUE
979 825 CONTINUE
980 C      BEGIN CHANGING NODE NUMBERS & COORD.
981 C      CHANGE NODE NUMBERS
982      DO J=1,NTOE
983      DO K=1,ntype
984      IF (NN(J,K).GT.KND) THEN
985      NN(J,K)=NN(J,K)+1
986      END IF
987      END DO
988      END DO
989
990 C-- substitution FOR DOUB. NODES x or y or z dir.--
991
992      DO K=1,NTOE
993      DO M=1,NEE
994      MM=JFACE(M)
995      IF (NN(K,MM).EQ.KND) THEN
996      MJ=IFACE(M)
997      NN(K+NTELR,MJ)=KND+1
998      END IF
999      END DO
1000     END DO
1001
1002 c-- corrections for doub. nodes ends.--
1003
1004 C      CHANGING COORD.
1005      DO J=NTON,KND,-1
1006      X(J+1)=X(J)
1007      Y(J+1)=Y(J)
1008      Z(J+1)=Z(J)
1009      T(J+1)=T(J)
1010      END DO
1011      NTON=NTON+1
1012 830 CONTINUE
1013      return
1014      end
1015 c=====
1016      subroutine delete(nonx,nony,nonz,ntd,nond)
1017      dimension nond(1)
1018      ix=nonz
1019      iyy=nonx*nonz
1020      IFNO=0
1021      DO 422 I=1,NONZ
1022      DO 422 J=1,NONY
1023      DO 422 K=1,NONX
1024      IF (MOD(I,2).EQ.0) GO TO 411
1025      IF (MOD(J,2).NE.0.OR.MOD(K,2).NE.0) GO TO 422
1026 c      I-odd, j,k-even.....

```



```

1027      go to 421
1028 411  if(mod((j+k),2).eq.0) go to 422
1029 421  i3=iyy*(j-1)+ixxx*(k-1)+1
1030      do 425 l=1,ntd
1031      if (i3.ne.nond(1)) go to 425
1032      ifno=ifno+1
1033      do 426 m=1+1,ntd
1034 426  nond(m-1)=nond(m)
1035      go to 427
1036 425  continue
1037      go to 422
1038 427  ntd=ntd-1
1039 422  continue
1040 C    type *,ntd
1041 C    type *,(nond(1),i=1,ntd)
1042      return
1043      end
1044
1045 c&=====
1046      SUBROUTINE MESHPL (XX,YY,NONX,NONY,NONZ,MIP,IPLOT,ntype)
1047 c-----
1048 c--   Program 'MESH.SOR' to be used with 'PREPRO214.FOR'
1049 c--   This routine plots the mesh in xy-direction..
1050 c--   This routine is based on HP-GL language. The
1051 c--   plot cannot be displayed on ITY. Before using
1052 c--   this routine ASSIGN a HP plotter to FOR090
1053 c--   IPLOT = 1 original nodes
1054 c--   IPLOT = 2 after double nodes
1055 c-----
1056      LOGICAL*1 ETX,ESC
1057      CHARACTER *1 JUNK,MESH
1058      DATA ETX/'3',ESC/'33'/
1059      DIMENSION X(4000),Y(4000),XX(1),YY(1)
1060      .,mip(4000,2)
1061      NTOT=NONX*NONY*NONZ
1062      DO 2 I=1,NTOT
1063      X(I)=XX(I)
1064 2    Y(I)=YY(I)
1065
1066      IX=NONZ
1067      IY=NONX*NONZ
1068
1069 C--   SCALING THE COORDINATES-----
1070
1071      XMIN=5000.
1072      XMAX=-5000.
1073      YMIN=5000.
1074      YMAX=-5000.
1075      DO 50 I=1,NTOT
1076      IF (X(I).GT.XMAX) XMAX=X(I)
1077      IF (X(I).LT.XMIN) XMIN=X(I)
1078      IF (Y(I).GT.YMAX) YMAX=Y(I)
1079 50   IF (Y(I).LT.YMIN) YMIN=Y(I)
1080      WRITE (5,*) 'TYPE LEVEL no. to plotted..'

```

```

1081      READ (5,*) LEVEL
1082      N=LEVEL
1083      XLL=XMIN
1084      YLL=YMIN
1085      XMM=XMAX
1086      YMM=YMAX
1087      XL=XMM-XLL
1088      YL=YMM-YLL
1089      WRITE (5,*) ' DO YOU WANT TO GIVE X,Y LIMITS..? (Y/N)'
1090      READ (5,33) MESH
1091      IF (MESH.EQ.'Y'.OR.MESH.EQ.'y') THEN
1092      WRITE (5,*) ' ENTER X- LIMITS'
1093      READ (5,*) XLL,XMM
1094      WRITE (5,*) ' ENTER Y- LIMITS'
1095      READ (5,*) YLL,YMM
1096      XL=XMM-XLL
1097      YL=YMM-YLL
1098      END IF
1099      XO=650.0
1100      YO=1246.0
1101      Xc= 8800.0/XL
1102      Yc=6000.0/YL
1103      SC=xc
1104      IF (XC*yl.gt.6000.0) SC=yc
1105
1106      IF (XL.LT.YL) THEN
1107      XO=1246.0
1108      YO=650.0
1109      Xc= 6000.0/XL
1110      Yc=8800.0/YL
1111      SC=xc
1112      IF (XC*yl.gt.8800.0) SC=yc
1113      END IF
1114 C      type *, 'x1,y1',x1,y1
1115      DO 1 I=1,NTOT
1116      X(I)=(X(I)-XLL)*SC+XO
1117 1      Y(I)=(Y(I)-YLL)*SC+YO
1118
1119 C      type *,x(1),x(ntot)
1120 C      type *,y(1),y(ntot)
1121 9999 WRITE (90,9999) ESC,ESC,ESC,ESC
1122 9999 FORMAT (' ',A1,'.(',/,',',A1,'.0:0:',/,',',A1,'.140:;17:',/,',',
1123 . A1,'.N;19:',/,',', IN;DF;',/)
1124      WRITE (90,*) 'SP1;'
1125      WRITE (90,*) ' VS15.0;'
1126      IF (XL.LT.YL) THEN
1127      WRITE (90,*) ' R090;'
1128      WRITE (90,*) ' IP;IW;'
1129      END IF
1130
1131      XMM=(XMM-XLL)*SC+XO
1132      YMM=(YMM-YLL)*SC+YO
1133      XLL=XO
1134      YLL=YO

```

```

1135
1136      J1=level
1137      J2=(nony-1)*iy+j1
1138 C      DO 111 I=1,NOBY
1139 C111    TYPE *,Y(I),YLL
1140 C      CALL LIMIT (Y,YLL,j1,j2,iy,J1)
1141 C      CALL LIMIT (Y,YMM,J1,j2,iy,J2)
1142 C      J2=IF2(Y,YMM,J1,NOBY,1)
1143 C      TYPE *,'20 LOOP',J1,J2
1144      DO 20 J=J1,J2,iy
1145      NX1=j
1146      NX2=nx1+IY-IX
1147      CALL LIMIT (X,XLL,NX1,NX2,IX,NX1)
1148      CALL LIMIT (X,XMM,NX1,NX2,IX,NX2)
1149 C      NX1=IF1(X,XLL,NX1,NX2,IX)
1150 C      NX2=IF2(X,XMM,NX1,NX2,IX)
1151 C      TYPE *,' 10 LOOP ',NX1,NX2
1152 C      TYPE *,X(NX1),Y(NX1),X(NX2),Y(NX2)
1153      DO 10 I=NX1,NX2,IX
1154      IF (i.eq.nx1) WRITE (90,101), X(I),Y(I)
1155      WRITE (90,104), X(I),Y(I)
1156 104    FORMAT (' PD',2F11.3,')')
1157 10      continue
1158 20      WRITE (90,*) ' PU;'
1159
1160      I1=level
1161      I2=(NONX-1)*ix+i1
1162 C      CALL LIMIT (X,XLL,i1,i2,IX,i1)
1163 C      CALL LIMIT (X,XMM,i1,i2,IX,i2)
1164 C      I1=IF1(X,XLL,1,NOBX,1)
1165 C      I2=IF2(X,XMM,1,NOBX,1)
1166 C      TYPE *,' 30 LOOP ',I1,I2
1167      DO 30 I=I1,I2,IX
1168      NY1=i
1169      NY2=NY1+(nony-1)*IY
1170 C      CALL LIMIT (Y,YLL,NY1,NY2,iy,NY1)
1171 C      CALL LIMIT (Y,YMM,NY1,NY2,iy,NY2)
1172 C      NY1=IF1(Y,YLL,NY1,NY2,iy)
1173 C      NY2=IF2(Y,YMM,NY1,NY2,iy)
1174 C      TYPE *,' 40 LOOP ',NY1,NY2
1175      DO 40 J=NY1,NY2,iy
1176      IF (J.EQ.NY1) WRITE (90,101), X(J),Y(J)
1177      WRITE (90,102), X(J),Y(J)
1178 40      CONTINUE
1179 101    FORMAT (' PU;PA',2F11.3,')')
1180 102    FORMAT (' PD',2F11.3,')')
1181 30      WRITE (90,*) ' PU;'
1182
1183      WRITE (5,*) 'DO YOU NEED NODE NOS. (Y/N).. '
1184      READ (5,33) JUNK
1185 33      FORMAT (A)
1186      IF (JUNK.EQ.'Y'.OR.JUNK.EQ.'y') GO TO 44
1187      GO TO 66
1188 !      WRITE (5,*) '..ENTER SSX,SSY...'

```

```

1189 !      READ (5,*) SSX,SSY
1190 44      SSX=0.1
1191      SSY=0.15
1192 !--    WRITING THE NODE NUMBERS-----
1193      WRITE (90,*) 'SIO.1,0.15;'
1194      J1=level
1195      J2=(nony-1)*iy+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 *,(KK,KK=NX1,NX2,IX)
1205      DO 100 IN=NX1,NX2,IX
1206      NXX=(IN-NX1)/IX+1      !skipping the face nos.
1207      IF (MOD(J,2).EQ.0.AND.MOD(NXX,2).EQ.0.and.
1208      . ntype.eq.21)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 (IPLOT.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 100      WRITE (90,*) ' FU;'
1222 66      WRITE (90,*) 'SPO;'
1223      WRITE (5,*) 'TYPE another LEVEL no. to plotted..'
1224      READ (5,*) LEVEL
1225      if (level.ne.0) go to 888
1226      RETURN
1227      END
1228
1229      SUBROUTINE SYMB (X,Y,SPL,I)
1230      LOGICAL*1 ETX,ESC
1231      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+0.5*(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

```

```

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(I),AL
1261      IF (A(I).GE.(AL-1.0E-03)) THEN
1262      IF1=I
1263      RETURN
1264      END IF
1265 10     CONTINUE
1266      END
1267
1268      INTEGER FUNCTION IF2(A,AL,I1,I2,IN)
1269 c--    to find the lower limit of do loop...
1270      DIMENSION A(1)
1271      IF2=I2
1272      DO 10 I=I1,I2,IN
1273 C      type *,a(i),al
1274      IF (A(I).GE.(AL-1.0E-03)) THEN
1275      IF1=I
1276      RETURN
1277      END IF
1278 10     CONTINUE
1279      END
1280
1281 c&=====
1282      SUBROUTINE HOLE(RX,NONX,NONY,NONZ,XX,YY,ZZ,X,Y,Z)
1283 C    GENERATING POLAR COORDINATES AND POLAR MESH--
1284      DIMENSION XX(1),YY(1),ZZ(1),X(1),Y(1),R(100),NC(4)
1285      . ,Z(1)
1286 c      TYPE *,(XX(I),I=1,NONX)
1287 c      TYPE *,(YY(I),I=1,NONY)
1288 c      TYPE *,(ZZ(I),I=1,NONZ)
1289      pi=3.14159265
1290      EPS=1.0E-05
1291
1292      IX=NONZ
1293      IY=NONX*NONZ
1294      DO 31 I=1,NONX
1295      rad=rx
1296      if (xx(i).ne.0.0)rad=sqrt(xx(i)**2+rx**2)

```

```

1297      DO 31 J=1,NONY
1298
1299      xxx=xx(i)
1300      if (yy(j).le.(rx-eps)) then
1301      xxx=rad
1302      if (yy(j).ne.0.)xxx=sqrt(rad**2-yy(j)**2)
1303      nbk=(j-1)*iy+(i-2)*ix+1
1304      TYPE *,NBK
1305      if ((xxx-x(nbk)).lt.(xx(2)-xx(1)))xxx=(nbk)+xx(2)-xx(1)
1306      end if
1307
1308      nxy=(J-1)*IY+(I-1)*IX
1309      DO 31 K=1,NONZ
1310      N=nxy+K
1311      X(N)=xxx
1312      IF (I.EQ.NONX) X(N)=XX(NONX)
1313      Y(N)=YY(J)
1314      Z(N)=ZZ(K)
1315 31      continue
1316 C--      correcting for center nodes for 31 node element....
1317
1318      DO 20 K=1,NONZ
1319      DO 20 J=1,NONY
1320      DO 20 I=1,NONX
1321      IF (MOD(I,2).NE.0) GO TO 20
1322      N=(J-1)*IY+(I-1)*IX+K
1323      X(N)=(X(N-IX)+X(N+IX))/2.0
1324 20      CONTINUE
1325      do 30 j=1,nony
1326      do 30 i=1,nonx
1327      n=(j-1)*iy+(i-1)*ix+1
1328      type *,n,x(n),y(n)
1329 30      CONTINUE
1330      RETURN
1331      END

```

APPENDIX - B  
LISTING OF THE MAIN CODE 'KSAP II'

```

1 C  **  **  **  **  **  **  **  **  **  **  **  **  **  **  **  **  **  **
2 C
3 C          KSAP 11
4 C      SIMPLIFIED VERSION OF SAP4 FOR
5 C      USING ELEMENT TYPE 8 ONLY
6 C
7 C          September 1987
8 C
9 C
10 C
11 C  **  **  **  **  **  **  **  **  **  **  **  **  **  **  **  **  **  **
12 C
13      IMPLICIT REAL*8(A-H,O-Z)
14      REAL*4 T,IT
15      COMMON /JUNK / HED(12),JUK(406)
16      COMMON /ELPAR/ NPAR(14),NUMNP,MBAND,NELTYP,N1,N2,N3,N4,N5,MTOT,NEQ
17      COMMON /EM/   QQQ(2846)
18      COMMON /DYN/  IDUS(11),NDYN
19      COMMON /TAPES/ NQQ(6)
20      COMMON /EXTRA/ MDEX,NT8,N10SV,NT10,KEQB,NUMEL,T(10)
21      COMMON /SOL/  NBLOCK,NEQB,LL,NE, IDUM,NEIG,NAD,NVV,ANORM,NEQ
22 C
23 C      PROGRAM CAPACITY CONTROLLED BY THE FOLLOWING TWO STATEMENTS ...
24 C
25      COMMON A(650001)      !CHANGE MTOT ALSO
26 C
27 C  -- OPEN SCRATCH FILES
28 C      OPEN (UNIT=1,STATUS='UNKNOWN',FORM='UNFORMATTED')
29 C  -----
30      open(unit=1,FILE='SCR:casw.eman11',status='scratch',
31      .form='unformatted')
32 !      open(unit=2,status='scratch',form='unformatted')      !dra0:
33      open (unit=2,file='SCR:casw.eman11',status='new',blocksize=4800,
34      .form='unformatted')      !msa0:
35 !      open(unit=3,status='scratch',form='unformatted')      !dra0:
36      open (unit=3,file='UTL:casw.eman11',status='new',blocksize=4800,
37      .form='unformatted')      !msa0:
38 !      open (unit=3,file='MSA0:[ESK]',status='new',blocksize=4800,
39 !      .form='unformatted')      !msa0:
40      OPEN (UNIT=4,FILE='SCR:casw.eman11',STATUS='NEW',BLOCKSIZE=4800,
41      .FORM='UNFORMATTED')      !dra0:
42 !      open(unit=55,status='UNKNOWN',form='unformatted')      !dra0:
43      OPEN (UNIT=55,FILE='SCR:casw.eman11',STATUS='NEW',BLOCKSIZE=4800,
44      .FORM='UNFORMATTED')      !DMA0:
45      open(unit=8,FILE='SCR:casw.eman11',status='scratch',
46      .FORM='UNFORMATTED')      !dra0:
47      open(unit=9,FILE='SCR:casw.eman11',status='scratch',
48      .FORM='UNFORMATTED')      !dra0:
49 C      OPEN (UNIT=15,STATUS='SCRATCH',FORM='UNFORMATTED')      !DRA0:
50      open(unit=16,FILE='SCR:casw.eman11',status='scratch',
51      .FORM='UNFORMATTED')      !dra0:
52      open(unit=18,FILE='SCR:casw.eman11',status='scratch',
53      .FORM='UNFORMATTED')      !dra0:
54      open (unit=19,file='workdone.wok',status='new')      !dra0:

```



```

55      open (unit=33,file='SCR:LASW.EMANI1DISP.dat',status='new')
56      open (unit=34,file='ksapout.dat',status='new')
57      OPER (UNIT=15,FILE='SCR:LASW.EMANI1',STATUS='NEW',BLOCKSIZE=4800,
58      .FORM='UNFORMATTED')
59
60 *-----
61
62 !     THE following should be 1 less than A() dimension
63      MTOT= 650000      ! 300000
64
65 C     USE THE IBM FORTRAN EXTENDED ERROR HANDLING FACILITY TO
66 C     ELIMINATE PRINTOUT OF UNDERFLOW ERROR MESSAGE (ERROR NUMBER 208)
67 C
68 C     CALL ERRSET (208,256,-1,1)
69 C
70 C
71 C     CALL STIME
72 C
73      NT8 = 8
74      REWIND NT8
75      NT10= 10
76      REWIND NT10
77      N1=1
78 C
79 C     P R O G R A M   C O N T R O L   D A T A
80 C
81      5 CALL TTIME(T(1))
82      READ (5,100,END=990) HED,NUMNP,NELTYP,LL,NF,NDYN,MODEX,NAD,
83      1      KEQB,N1OSV,NDOF
84      IF(MODEX.GT.0) MODEX = 1
85      IF (NUMNP.EQ.0) STOP
86      WRITE (33,200) HED,NUMNP,NELTYP,LL,NF,NDYN,MODEX,NAD,KEQB,N1OSV
87      WRITE (19,299) HED !WORKDONE.WOK FILE TILTE.....
88      WRITE (34,299) HED !KSAPOUT.DAT (STRESSES) FILE TILTE.....
89      IF(KEQB.LT.2) KEQB = 99999
90      IF (NDYN.NE.0) LL=1
91      IF(LL.GE.1) GO TO 10
92      WRITE (33,300)
93      STOP
94 C*** DATA PORTHOLE SAVE
95      10 IF(MODEX.EQ.1)
96      *WRITE (NT8)      HED,NUMNP,NELTYP,LL,NF,NDYN
97 C
98      KDYN = IABS(NDYN) +1
99      IF(KDYN.LE.5) GO TO 14
100     WRITE (33,310) NDYN
101     STOP
102 C
103 C     RE-START MODE ACTIVATED IF NDYN.EQ.-2 OR NDYN.EQ.-3
104 C
105     14 IF(NDYN.LT.0) GO TO 20
106 C
107 C     I N P U T   J O I N T   D A T A
108 C

```

```

109      N2=N1+6*NUMNP
110      N3=N2+NUMNP
111      N4=N3+NUMNP
112      N5=N4+NUMNP
113      N6=N5+NUMNP
114      IF(N6.GT.MTOT) CALL ERROR(N6-MTOT)
115 C
116      CALL INPUTJ(A(N1),A(N2),A(N3),A(N4),A(N5).NUMNP,NEQ)
117 C
118 C      F O R M   E L E M E N T   S T I F F N E S S E S
119 C
120      CALL TTIME(T(2))
121 C
122      MBAND=0
123      NUMEL=0
124      REWIND 1
125      REWIND 2
126 C
127      DO 900 M=1,NELTYP
128          READ  (5,1001) NPAR
129 C***    DATA PORTHOLE SAVE
130          IF(MODEX.EQ.1) WRITE (NT8) NPAR
131          WRITE (1) NPAR
132          NUMEL=NUMEL+NPAR(2)
133          MTYPE=NPAR(1)
134 C
135          CALL ELTYPE(MTYPE)
136 C
137      900 CONTINUE
138 C
139 C      D E T E R M I N E   B L O C K S I Z E
140 C
141 C      ADDSTF
142 C
143      LL1=LL+NDOF      ! IN the following LL is replaced with LL1
144      NEQB=(MTOT - 4*LL)/(MBAND + LL1 + 1)/2      !modified with ndof
145 C
146 C      OVER-RIDE THE SYSTEM MATRIX BLOCKSIZE WITH THE INPUT (NON-ZERO)
147 C      VALUE, KEQB.
148 C      THIS OVER-RIDE ENTRY IS TO ALLOW PROGRAM CHECKING OF MULTI-
149 C      BLOCK ALGORITHMS WITH WHAT WOULD NORMALLY BE ONE BLOCK DATA.
150 C
151      IF(KEQB.LT.NEQB) NEQB = KEQB
152 C
153      GO TO  (690,700,700,700,730), KDYN
154 C
155 C      STATIC SOLUTION
156 C
157      690 CONTINUE
158      NEQB1=(MTOT - MBAND)/(2*(MBAND+LL1) + 1)
159      NEQB2=(MTOT - MBAND - LL1 *(MBAND-2))/(3*LL1 + MBAND + 1)
160      IF (NEQB1.LT.NEQB) NEQB=NEQB1
161      IF (NEQB2.LT.NEQB) NEQB=NEQB2
162      NBLOCK = (NEQ-1)/NEQB +1

```

```

163      IF(NEQB.GT.NEQ) NEQB=NEQ
164      GO TO 790
165 C
166 C      EIGENSOLUTION
167 C
168 C          1. DETERMINANT SEARCH ALGORITHM
169 C
170      700 IF (NEQB.LT.NEQ) GO TO 710
171          NIM=3
172          NC=NE + NIM
173          NV=6
174          NCA=NEQ*MAX0(MBAND,NC)
175          NTOT=NCA + 4*NEQ + 2*NV*NEQ + 5*NC
176          NEIG=0
177          IF(NTOT.LE.MTOT) GO TO 720
178 C
179 C          2. SUBSPACE ITERATION ALGORITHM
180 C
181      710 NV=MIN0(2*NE,NE+6)
182          IF (NAD.NE.0) NV=NAD
183          NEQB1=(MTOT - MBAND)/(2*MBAND + 1)
184          NEQB2=(MTOT - MBAND - 2*NV - NV*(MBAND-2))/(3*NV + MBAND + 1)
185          NEQB3=(MTOT - 3*NV*NV - 3*NV)/(2*NV + 1)
186          NEQB4=(MTOT - 6*NV)/(1 + MBAND)
187          IF (NEQB1.LT.NEQB) NEQB=NEQB1
188          IF (NEQB2.LT.NEQB) NEQB=NEQB2
189          IF (NEQB3.LT.NEQB) NEQB=NEQB3
190          IF (NEQB4.LT.NEQB) NEQB=NEQB4
191          NEIG=1
192 C
193      720 CONTINUE
194          NBLOCK = (NEQ-1)/NEQB + 1
195          IF (NEQB.GE.NEQ) NEQB=NEQ
196 C
197 C      HISTORY OR SPECTRUM ANALYSIS
198 C
199          KREM = 1000
200          NTOT = NBLOCK*NEQB*NE + KREM
201          IF(MTOT.LT.NTOT)
202      *WRITE (33,320)
203          GO TO 790
204 C
205 C      STEP-BY-STEP DIRECT INTEGRATION
206 C
207      730 CONTINUE
208 C      DISPLACEMENT COMPONENTS FOR DIRECT OUTPUT (*NSD*)
209          NN2 = NEQ
210 C      DISPLACEMENT COMPONENTS REQUIRED FOR RECOVERY OF ALL OF THE
211 C      REQUESTED ELEMENT STRESS COMPONENTS (*NSS*)
212          NN3 = NEQ
213 C
214 C          1. DECOMPOSITION
215 C
216          NEQB1 = (MTOT-NN2-NN3-NEQ-MBAND)/(2*MBAND+1)

```

```

217 C
218 C      2. TIME INTEGRATION PHASE
219 C
220      mcs1= MBAND+2*(NN2+NN3)+5*NEQ +(2*MBAND+1)
221
222
223      write (33,555) mcs1
224 555      format(/5x,' Minimum dimension, MTOT, required for array A( )
225      = ',I8/5X,50(1H+)//)
226      if(mtot.le.mcs1) STOP      ! Abnormal stop as dim. of A is insufficient
227
228      NEQB2 = (MTOT-MBAND-2*(NN2+NN3)-5*NEQ)/(MBAND+1)
229 C
230      IF(NEQB1.LT.NEQB) NEQB = NEQB1
231      IF(NEQB2.LT.NEQB) NEQB = NEQB2
232      IF(NEQB.GT.NEQ) NEQB = NEQ
233      NBLOCK = (NEQ-1)/NEQB +1
234 C
235 C      3. INPUT PHASE
236 C
237 C      NUMBER OF TIME FUNCTIONS (*NEN*)
238      NN2 = 10
239 C      MAXIMUM NUMBER OF FUNCTION DEFINITION POINTS (*MXLP*)
240      NN3 = 40
241 C
242      NN4 = 6*NUMNP + 2*NN2*NEQ
243      IF(NN4.GT.MTOT)
244      *WRITE (33,320)
245      NN4 = NEQ*2*(NN2+1) + NN2*(1+2*NN3)
246      IF(NN4.GT.MTOT)
247      *WRITE (33,320)
248 C
249 790 CONTINUE
250 C
251 C      INPUT NODAL LOADS
252 C
253      N3=N2+NEQB*ALL
254      N4=N3+6*LL
255      WRITE (33,201) NEQ,MBAND,NEQB,NBLOCK
256 C
257      CALL TTIME(T(3))
258 C
259      CALL INL(A(N1),A(N2),A(N3),A(N4),NUMNP,NEQB,LL)
260 C
261      CALL TTIME(T(4))
262 C
263 C      FORM TOTAL STIFFNESS
264 C
265      NE2B=2*NEQB
266      N2=N1+NEQB*MBAND
267      N3=N2+NEQB*ALL
268      N4=N3+4*LL
269      NN2=N1+NE2B*MBAND
270      NN3=NN2+NE2B*ALL

```

```

271      NN4=NN3+4ALL
272 C
273      CALL ADDSTF (A(N1),A(NN2),A(NN3),A(NN4),NUMEL,NBLOCK,NE2B,LL,MBAND
274      1,ANORM,NUV)
275 C
276      CALL TTIME(T(5))
277 C
278 C      S O L U T I O N   P H A S E
279 C
280      20 GO TO (30,40,50,60,70), KDYN
281 C
282 C      STATIC SOLUTION
283 C
284      30 IF(MODEX.EQ.0) GO TO 32
285      DO 31 I=6,10
286      31 T(I) = T(5)
287      GO TO 90
288 C
289      32 CALL SOLEQ
290      CALL TTIME(T(6))
291      DO 33 I=7,10
292      33 T(I) = T(6)
293      GO TO 90
294 C
295 C      EIGENVALUE EXTRACTION
296 C
297      40 T(6) = T(5)
298 C      CALL SOLEIG
299      CALL TTIME(T(7))
300      T(8) = T(7)
301      T(9) = T(7)
302      T(10)= T(7)
303      GO TO 90
304 C
305 C      FORCED DYNAMIC RESPONSE ANALYSIS
306 C
307      50 T(6) = T(5)
308      IF(NDYN.LT.0) GO TO 52
309 C      CALL SOLEIG
310      CALL TTIME (T(7))
311      GO TO 54
312      52 DO 53 I=1,6
313      53 T(I+1)=T(I)
314      REWIND 2
315      READ (2) NEQ,NBLOCK,NEQB,MBAND,N1,NE,(QQQ(I),I=1,NE)
316      REWIND 55
317      IMAX=NEQBANE
318      READ (55) (A(I),I=1,NE)
319      DO 56 L=1,NBLOCK
320      56 READ (55) (A(I),I=1,IMAX)
321      54 CONTINUE
322 C      54 CALL HISTRY
323      CALL TTIME (T(8))
324      T(9) = T(8)

```

```

325     T(10)= T(8)
326     GO TO 90
327 C
328 C     RESPONSE SPECTRUM ANALYSIS
329 C
330     60 T(6) = T(5)
331     IF(NDYN.LT.0) GO TO 62
332 C     CALL SOLEIG
333     CALL TTIME (T(7))
334     T(8) = T(7)
335     GO TO 64
336     62 DO 63 I=1,7
337     63 T(I+1)=T(I)
338     REWIND 2
339     READ (2) NEQ,NBLOCK,NEQB,MBAND,N1,NE
340     REWIND 55
341     IMAX=NEQBANE
342     READ (55) (A(I),I=1,NE)
343     DO 66 L=1,NBLOCK
344     66 READ (55) (A(I),I=1,IMAX)
345     64 CONTINUE
346 C     64 CALL RESPEC
347     CALL TTIME (T(9))
348     T(10)= T(9)
349     GO TO 90
350 C
351 C     STEP-BY-STEP (DIRECT INTEGRATION) ANALYSIS
352 C
353     70 DO 71 I=6,9
354     71 T(I) = T(5)
355 C     CALL STEP
356     CALL TTIME(T(10))
357 C
358 C     COMPUTE AND PRINT OVERALL TIME LOG
359 C
360     90 TT = 0.0
361     DO 95 I=1,9
362     T(I) = T(I+1)-T(I)
363     TT = TT + T(I)
364     95 CONTINUE
365 C
366     WRITE (33,203) (T(K),K=1,9),TT
367 C
368     GO TO 5
369     990 CONTINUE
370 C -- CLOSE ALL SCRATCH FILES
371     CLOSE (UNIT=1)
372     CLOSE (UNIT=2)
373     CLOSE (UNIT=3)
374     CLOSE (UNIT=4)
375     CLOSE (UNIT=55)
376     CLOSE (UNIT=8)
377     CLOSE (UNIT=9)
378     close (unit=15)

```

```

379      close (unit=16)
380      close (unit=18)
381      close (unit=19)
382 C
383      STOP
384 C
385      100 FORMAT (12A6/10I5)
386 299      FFORMAT(/2X,12A6/)
387 200      FFORMAT(1H1,12A6///
388          1 38H C O N T R O L   I N F O R M A T I O N, // 4X,
389          2 27H NUMBER OF NODAL POINTS   =, 15 / 4X,
390          3 27H NUMBER OF ELEMENT TYPES   =, 15 / 4X,
391          4 27H NUMBER OF LOAD CASES      =, 15 / 4X,
392          5 27H NUMBER OF FREQUENCIES     =, 15 / 4X,
393          6 27H ANALYSIS CODE (NDYN)      =, 15 / 4X,
394          7 16H   EQ.0,  STATIC,          / 4X,
395          8 26H   EQ.1,  MODAL EXTRACTION, / 4X,
396          9 25H   EQ.2,  FORCED RESPONSE, / 4X,
397          A 27H   EQ.3,  RESPONSE SPECTRUM, / 4X,
398          * 28H   EQ.4,  DIRECT INTEGRATION, / 4X,
399          B 27H SOLUTION MODE (MODEX)     =, 15 / 4X,
400          C 19H   EQ.0,  EXECUTION,        / 4X,
401          D 20H   EQ.1,  DATA CHECK,      / 4X,
402          E 19H NUMBER OF SUBSPACE,        / 4X,
403          F 27H ITERATION VECTORS (NAD)   =, 15 / 4X,
404          G 27H EQUATIONS PER BLOCK       =, 15 / 4X,
405          H 27H TAPE10 SAVE FLAG (N10SV) =, 15 / 4X)
406      201 FORMAT (38H1E Q U A T I O N   P A R A M E T E R S, //
407          *      34H TOTAL NUMBER OF EQUATIONS   =,15,
408          1      /34H BANDWIDTH                 =,15,
409          2      /34H NUMBER OF EQUATIONS IN A BLOCK =,15,
410          3      /34H NUMBER OF BLOCKS          =,15)
411      203 FORMAT (1H1,31H O V E R A L L   T I M E   L O G, //
412          1 5X,30HNODAL POINT INPUT           =, F8.2 /
413          2 5X,30HELEMENT STIFFNESS FORMATION =, F8.2 /
414          3 5X,30HNODAL LOAD INPUT            =, F8.2 /
415          4 5X,30HTOTAL STIFFNESS FORMATION   =, F8.2 /
416          5 5X,30HSTATIC ANALYSIS            =, F8.2 /
417          6 5X,30HEIGENVALUE EXTRACTION      =, F8.2 /
418          7 5X,30HEFORCED RESPONSE ANALYSIS  =, F8.2 /
419          8 5X,30HKRESPONSE SPECTRUM ANALYSIS =, F8.2 /
420          * 5X,30HSTEP-BY-STEP INTEGRATION    =, F8.2 //
421          9 5X,30HTOTAL SOLUTION TIME        =, F8.2 /)
422 C
423      300 FORMAT (// 48H ** ERROR. (AT LEAST ONE LOAD CASE IS REQUIRED) )
424      310 FORMAT (// 33H ** ERROR. ANALYSIS CODE (NDYN =,13,9H) IS BAD. )
425      320 FORMAT (// 47H ** WARNING. ESTIMATE OF STORAGE FOR A DYNAMIC,
426          1      32H ANALYSIS EXCEEDS AVAILABLE CORE, // 1X)
427 C
428      1001 FORMAT (14I5)
429      END
430 C=====
431      SUBROUTINE ADDSTF (A,B,STR,IMASS,NUMEL,NBLOCK,NE2B,LL,MBAND,ANORM,
432          INVV)

```

```

433 C
434     IMPLICIT REAL*8(A-H,O-Z)
435 C
436 C     CALLED BY:  MAIN
437 C
438 C     FORMS GLOBAL EQUILIBRIUM EQUATIONS IN BLOCKS
439 C
440     DIMENSION A(NE2B,MBAND),B(NE2B,LL),STR(4,LL),TMASS(NE2B)
441 C
442     COMMON /DYN/  NT,NOT,ALFA,DT,BETA,NEN,NGM,NAT,NDYN
443     COMMON /EM/  LRD,ND,LR(63),IPAD,SS(2331)
444     COMMON /EXTRA/  MDEX,NTS,IFILL(14)
445 C
446     NEQB=NE2B/2
447     K=NEQB+1
448     X=NBLOCK
449     MB=DSQRT(X)
450     MB=MB/2+1
451     NEBB=MB*NE2B
452     MM=1
453     NDEG=0
454     NVV=0
455     ANORM=0.
456     NSHIFT=0
457     REWIND 3
458     REWIND 4
459     REWIND 9
460 C
461 C     READ ELEMENT LOAD MULTIPLIERS
462 C
463     WRITE (33,2000)
464     DO 50 L=1,LL
465     READ (5,1002) (STR(I,L),I=1,4)
466 50 WRITE (33,2002) L,(STR(I,L),I=1,4)
467     IF(MDEX.EQ.0) WRITE (8) STR
468 C
469 C     FOR A STEP-BY-STEP ANALYSIS (NDYN.EQ.4) READ THE SOLUTION
470 C     CONTROL CARD.  THE TIME STEP (DT) AND THE DAMPING COEFFICIENTS
471 C     (ALFA/BETA) ARE REQUIRED FOR THE ASSEMBLY OF THE EFFECTIVE
472 C     SYSTEM STIFFNESS MATRIX IN THIS ROUTINE.
473 C
474     IF(NDYN.NE.4) GO TO 65
475 C
476     READ (5,1004) NEN,NGM,NAT,NT,NOT,DT,ALFA,BETA
477     WRITE (33,2004) NEN,NGM,NAT,NT,NOT,DT,ALFA,BETA
478     IF(NAT.EQ.0) NAT = 1
479     IF(NOT.EQ.0) NOT = 1
480     IF(DT.GT.1.0E-12) GO TO 55
481     WRITE (33,3000)
482     STOP
483 C
484 C     COMPUTE INTEGRATION COEFFICIENTS FOR ASSEMBLY OF EFFECTIVE
485 C     SYSTEM STIFFNESS (STEP-BY-STEP ANALYSIS ONLY)
486 C

```



```

487 55 TETA = 1.4
488 DT1 = TETA*DT
489 DT2 = DT1**2
490 A0 = (6.+3.*ALEA*DT1)/(DT2+3.*BETA*DT1)
491 C
492 65 IF(MODEX.EQ.1) RETURN
493 C
494 C FORM EQUATIONS IN BLOCKS ( 2 BLOCKS AT A TIME)
495 C
496 DO 1000 M=1,NBLOCK ,2
497 DO 100 I=1,NE2B
498 DO 100 J=1,MBAND
499 100 A(I,J)=0.
500 READ (3) ((B(I,L),I=1,NEQB),L=1,LL),(TMASS(I),I=1,NEQB)
501 IF (M.EQ.NBLOCK) GO TO 200
502 READ (3) ((B(I,L),I=K,NE2B),L=1,LL),(TMASS(I),I=K,NE2B)
503 200 CONTINUE
504 C
505 REWIND 55
506 REWIND 2
507 NA=55
508 NUME=NUM7
509 IF (MM.NE.1) GO TO 75
510 NA=2
511 NUME=NUMEL
512 NUM7 =0
513 C
514 75 DO 700 N=1,NUME
515 READ (NA) LRD,ND,(LM(I),I=1,ND),(SS(I),I=1,LRD)
516 MSHFT = ND * (ND+1)/2 + 4 *ND
517 DO 600 I=1,ND
518 LMN=1-LM(I)
519 II=LM(I)-NSHIFT
520 IF (II.LE.0.OR.II.GT.NE2B) GO TO 600
521 IMS=I+MSHFT
522 TMASS(II)=TMASS(II)+ SS(IMS)
523 DO 300 L=1,LL
524 DO 300 J=1,4
525 KK = ND *(ND+1)/2 + ND*(J-1)
526 300 B(II,L)=B(II,L)+SS(I+KK)*STR(J,L)
527 DO 500 J=1,ND
528 JJ=LM(J)+LMN
529 IF(JJ) 500,500,390
530 390 IF(J-I) 396,394,394
531 394 KK = ND*I-(I-1)*I/2 + J-ND
532 GO TO 400
533 396 KK =ND*J -(J-1)*J/2+I-ND
534 400 A(II,JJ)=A(II,JJ)+SS( KK)
535 500 CONTINUE
536 600 CONTINUE
537 C
538 C DETERMINE IF STIFFNESS IS TO BE PLACED ON TAPE 55
539 C
540 IF (MM.GT.1) GO TO 700

```

```

541      DO 650 I=1,ND
542      II=LM(I) -NSHIFT
543      IF(II.GT.NE2B.AND.II.LE.NE8B) 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 C      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) + A0*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) + A0*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.NB) 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 (5I5,3F10.0)

```

```

595 1010 FORMAT (51HSTRUCTURE WITH NO DEGREES OF FREEDOM CHECK DATA )
596 2000 FORMAT (/// 10H STRUCTURE,13X,7HELEMENT,4X,4HLOAD,4X,
597 1 11HMULTIPLIERS,/ 10H LOAD CASE,12X,1HA,9X,1HB,9X,1HC,9X,1HD,/ 1X)
598 2002 FORMAT (1G,7X,4F10.3)
599 2004 FORMAT (45HSTEP - BY - STEP SOLUTION ,
600 1 37HC O N T R O L I N F O R M A T I O N , ///
601 2 5X, 35HNUMBER OF TIME VARYING FUNCTIONS =, 15 //
602 3 5X, 35HGROUND MOTION INDICATOR =, 15 /
603 4 8X, 10HEQ.0, NONE, /
604 5 8X, 29HGT.0, READ ACCELERATION INPUT, //
605 6 5X, 35HNUMBER OF ARRIVAL TIMES =, 15 /
606 7 8X, 26HEQ.0, ALL FUNCTIONS ARRIVE, /
607 8 8X, 18H AT TIME ZERO, //
608 9 5X, 35HNUMBER OF SOLUTION TIME STEPS =, 15 //
609 A 5X, 35HOUTPUT (PRINT) INTERVAL =, 15 //
610 B 5X, 35HSOLUTION TIME INCREMENT =, E14.4 //
611 C 5X, 30HMASS- PROPORTIONAL DAMPING, /
612 D 5X, 35HCOEFFICIENT (ALPHA) =, E14.4 //
613 E 5X, 30HSTIFFNESS-PROPORTIONAL DAMPING, /
614 F 5X, 35HCOEFFICIENT (BETA) =, E14.4 /// 1X)
615 3000 FORMAT (27H0*** ERROR ZERO TIME STEP, / 1X)
616 END
617 C
618 C
619 C=====
620 SUBROUTINE BOUND
621 IMPLICIT REAL*8(A-H,O-Z)
622 COMMON A(1)
623 COMMON /ELPAR/ NPAR(14),NUMNP,MBAND,NELTYP,N1,N2,N3,N4,N5,MTOT,NEQ
624 COMMON /JUNK/ LT,LH,L,IPAD,SIG(20),IFILL(386)
625 COMMON /EXTRA/ MODEX,NT8,N10SV,NT10,IFILL2(12)
626 IF (NPAR(1).EQ.0) GO TO 500
627 CALL CLAMP (NPAR(2),A(N1),A(N2),A(N3),A(N4),NUMNP,MBAND)
628 RETURN
629 500 continue
630 c-- WRITE (33,2002)
631 NUME=NPAR(2)
632 DO 800 MM=1,NUME
633 CALL STRSC (A(N1),A(N3),NEQ,0)
634 c-- WRITE (33,2001)
635 DO 800 L=LT,LH
636 CALL STRSC (A(N1),A(N3),NEQ,1)
637 c-- WRITE (33,3002) MM,L,(SIG(1),I=1,2) !printing suppressed
638 IF(N10SV.EQ.1)
639 *WRITE (NT10) MM,L,SIG(1),SIG(2)
640 800 CONTINUE
641 RETURN
642 2001 FORMAT (/)
643 2002 FORMAT (48H1B O U N D A R Y E L E M E N T F O R C E S /,
644 1 14H M O M E N T S, // 8H ELEMENT,3X,4HLOAD,14X,5HFORCE,
645 2 9X,6HMOMENT, / 8H NUMBER,3X,4HCASE, // 1X)
646 3002 FORMAT (18,17,4X,2E15.5)
647 END
648 C

```

```

649 C
650 C&=====
651     SUBROUTINE CALBAN (MBAND,NDIF,LM,XM,S,P,ND,NDM,NS)
652     IMPLICIT REAL*8(A-H,O-Z)
653 C
654 C     CALLED BY:  RUSS,TEAM,PLNAX,BRICK8,IPLATE,CLAMP,ELST3D,PIPEK
655 C
656 C-----CALCULATES BAND WIDTH AND WRITES STIFFNESS MATRIX ON TAPE 2
657     DIMENSION LM(1),XM(1),S(NDM,NDM),P(NDM,4)
658     COMMON /EXTRA/ MODEX,NT8,IFILL(14)
659     MIN=100000
660     MAX=0
661     DO 800 L=1,ND
662     IF (LM(L).EQ.0) GO TO 800
663     IF (LM(L).GT.MAX) MAX=LM(L)
664     IF (LM(L).LT.MIN) MIN=LM(L)
665 800 CONTINUE
666     NDIF=MAX-MIN+1
667     IF (NDIF.GT.MBAND) MBAND=NDIF
668     IF(MODEX.EQ.1) GO TO 810
669 C
670     LRD=MDA(ND+1)/2+5*ND
671     WRITE(2) LRD,ND,(LM(I),I=1,ND),((S(I,J),J=I,ND),I=1,ND),
672     1 ((P(I,J),I=1,ND),J=1,4),(XM(I),I=1,ND)
673     RETURN
674 C
675 810 WRITE (1) ND,NS,(LM(I),I=1,ND)
676     RETURN
677 C
678     END
679 C
680 C
681 C&=====
682     SUBROUTINE CLAMP (NUMEL, ID,X,Y,Z,NUMNP,MBAND)
683     IMPLICIT REAL*8(A-H,O-Z)
684     COMMON/EM/LM(24),ND,NS,S(24,24),P(24,4),XM(24),SA(12,24),TT(12,4),
685     1 IFILL1(3048)
686     DIMENSION X(1),Y(1),Z(1),ID(NUMNP,1)
687     COMMON / JUNK / R(6),RM(4),IFILL2(410)
688     COMMON /EXTRA/ MODEX,NT8,IFILL3(14)
689     WRITE (33,2000) NUMEL
690     NS=2
691     ND=6
692     READ(5,1005) RM
693     WRITE (33,2005) RM
694     IF(MODEX.EQ.1)
695     *WRITE (NT8) RM
696     DO 30 NI=1,ND
697     XM(NI) = 0.0
698     DO 20 NJ=1,ND
699     20 S(NI,NJ)=0.0
700     30 CONTINUE
701     DO 50 NK=1,NS
702     DO 40 NL=1,ND

```

```

595 1010 FORMAT (51H0STRUCTURE WITH NO DEGREES OF FREEDOM CHECK DATA )
596 2000 FORMAT (/// 10H STRUCTURE,13X,7HELEMENT,4X,4HLOAD,4X,
597 1 11HMULTIPLIERS,/ 10H LOAD CASE,12X,1HA,9X,1HB,9X,1HC,9X,1HD,/ 1X)
598 2002 FORMAT (IG,7X,4F10.3)
599 2004 FORMAT (45H1S T E P - B Y - S T E P S O L U T I O N ,
600 1 37H C O N T R O L I N F O R M A T I O N , ///
601 2 5X, 35HNUMBER OF TIME VARYING FUNCTIONS =, 15 //
602 3 5X, 35HCONTINUING MOTION INDICATOR =, 15 /

```

```

703 40 SA(NK,NL) = 0.0
704 DO 50 NI=1,4
705 TI(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) NF,NI,NJ,NK,NL,KD,KR,KN,SD,SR,TRACE
712 IF (TRACE.EQ.0.) TRACE=1.0E+10
713 IF (KG.GT.0) GO TO 550
714 KG=KN
715 IF(MODEX.EQ.1) GO TO 530
716 IF(NJ.EQ.0)GO TO 250
717 X1=X(NJ)-X(NI)
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 T1=X(NI)-X(NP)
728 T2=Y(NI)-Y(NP)
729 T3=Z(NI)-Z(NP)
730 260 XL=T1*T1+T2*T2+T3*T3
731 XL=DSQRT(XL)
732 IF(XL.GT.1.0E-6) GO TO 270
733 WRITE (33,3000)
734 3000 FORMAT (32H0AAA ERROR ZERO ELEMENT LENGTH, / 1X)
735 STOP
736 270 CONTINUE
737 T1=T1/XL
738 T2=T2/XL
739 T3=T3/XL
740 IF (KD.EQ.0) GO TO 300
741 SA(1,1)=T1*TRACE
742 SA(1,2)=T2*TRACE
743 SA(1,3)=T3*TRACE
744 S(1,1)=T1*T1*TRACE
745 S(1,2)=T1*T2*TRACE
746 S(1,3)=T1*T3*TRACE
747 S(2,2)=T2*T2*TRACE
748 S(2,3)=T2*T3*TRACE
749 S(3,3)=T3*T3*TRACE
750 PP=TRACE*SD
751 R(1)=T1*PP
752 R(2)=T2*PP
753 R(3)=T3*PP
754 GO TO 350
755 300 DO 310 I=1,3
756 R(I)=0.0

```

```

757      SA(1,I)=0.0
758      DO 310 J=1,3
759 310  S(I,J)=0.0
760 350  IF (KR.EQ.0) GO TO 400
761      SA(2,5)=T3*TRACE
762      SA(2,4)=T1*TRACE
763      SA(2,6)=T3*TRACE
764      S(4,4)=T1*T1*TRACE
765      S(4,5)=T1*T2*TRACE
766      S(4,6)=T1*T3*TRACE
767      S(5,5)=T2*T2*TRACE
768      S(5,6)=T2*T3*TRACE
769      S(6,6)=T3*T3*TRACE
770      PP=TRACE*SR
771      R(4)=T1*PP
772      R(5)=T2*PP
773      R(6)=T3*PP
774      GO TO 450
775 400  DO 410 I=4,6
776      R(I)=0.0
777      SA(2,I)=0.0
778      DO 410 J=1,6
779 410  S(I,J)=0.0
780 450  DO 500 I=1,ND
781      DO 500 J=1,ND
782 500  S(J,I)=S(I,J)
783      DO 520 I=1,ND
784      DO 520 J=1,4
785 520  P(I,J)=R(I)*RM(J)
786 530  NN=NP
787      NNI=NI
788      NNJ=NJ
789      NNK=NK
790      NNL=NL
791      NKD=KD
792      NKR=KR
793      SSD=SD
794      SSR=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,NKD,NKR,KN,SSD,SSR,TTR
802      NE=NE+1
803      IF (MODEX.EQ.1)
804      *WRITE (NTB) NE,NN,NNI,NNJ,NNK,NNL,NKD,NKR,SSD,SSR,TTR
805      DO 600 I=1,ND
806 600  LM(I)=ID(NN,I)
807      NDM=24
808      CALL CALBAN (MBAND,NDIF,LM,XM,S,P,ND,NDM,NS)
809      IF (MODEX.EQ.1) GO TO 650
810      WRITE (1) ND,NS,(LM(L),L=1,ND),((SA(L,K),L=1,NS),K=1,ND),

```

```

811      1 ((TT(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 (8I5,3F10.0)
818      1005 FORMAT (4F10.0)
819      2000 FORMAT (34H1B O U N D A R Y E L E M E N T S, ///
820      1          27H ELEMENT TYPE          =          7, /
821      2          21H NUMBER OF ELEMENTS =,I6      ///1X)
822      2005 FORMAT (30H ELEMENT LOAD CASE MULTIPLIERS, // 8X,7HCASE(A),8X,
823      1          7HCASE(B),8X,7HCASE(C),8X,7HCASE(D),/ 4F15.4 /// 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 KD KR CODE (KN) DISPLACEMENT,6X,
829      5          22H ROTATION RATE, / 1X)
830      2010 FORMAT (1X,2(2X, I5),2X,4(4X, I5),2(2X, I5),7X, I5,2E15.4,E13.4)
831      END
832 C
833 C&=====
834      SUBROUTINE CROSS2 (A,B,C,IERR)
835 C
836 C      CALLED BY : INP21
837 C
838      IMPLICIT REAL*8(A-H,O-Z)
839 C
840 C      THIS ROUTINE FORMS THE VECTOR PRODUCT C = A*B WHERE *C*
841 C      IS NORMALIZED TO UNIT LENGTH
842 C
843      DIMENSION A(3),B(3),C(3)
844 C
845      X = A(2) * B(3) - A(3) * B(2)
846      Y = A(3) * B(1) - A(1) * B(3)
847      Z = A(1) * B(2) - A(2) * B(1)
848      XLN =DSQRT(X*X+Y*Y+Z*Z)
849      IERR = 1
850      IF(XLN.LE.1.0E-08) RETURN
851      XLN = 1.0 /XLN
852      C(3) = Z * XLN
853      C(2) = Y * XLN
854      C(1) = X * XLN
855      IERR = 0
856      RETURN
857      END
858 C&=====
859      SUBROUTINE DER3DS (NEL,XX,B,DET,R,S,T,NOD9,H,P,IELD,IELX)
860 C
861 C      CALLED BY : THDFE
862 C
863      IMPLICIT REAL*8(A-H,O-Z)
864 C

```

```

865 C
866 C
867 C . . . . .
868 C .
869 C .   P R O G R A M
870 C .
871 C .   EVALUATES STRAIN-DISPLACEMENT MATRIX B AT POINT (R,S,T)
872 C .
873 C .   CURVILINEAR HEXAHEDRON   8 TO 21 NODES
874 C .
875 C . . . . .
876 C
877 C
878 C
879   DIMENSION  XX(3,1),B(6,1),NOD9(1),H(1),P(3,1)
880   DIMENSION  XJ(3,3),XJI(3,3)
881 C
882 C
883 C   FIND INTERPOLATION FUNCTIONS AND THEIR DERIVATIVES
884 C   EVALUATE JACOBIAN MATRIX AT POINT (R,S,T)
885 C   COMPUTE DETERMINANT OF JACOBIAN MATRIX AT POINT (R,S,T)
886 C
887 C
888   CALL FNCT (R,S,T,H,P,NOD9,XJ,DET,XX,IELD,IELX,NEL)
889 C
890 C
891 C   COMPUTE INVERSE OF JACOBIAN MATRIX
892 C
893 C
894   DUM=1.0/DET
895   XJI(1,1)=DUM*( XJ(2,2)*XJ(3,3) - XJ(2,3)*XJ(3,2))
896   XJI(2,1)=DUM*(-XJ(2,1)*XJ(3,3) + XJ(2,3)*XJ(3,1))
897   XJI(3,1)=DUM*( XJ(2,1)*XJ(3,2) - XJ(2,2)*XJ(3,1))
898   XJI(1,2)=DUM*(-XJ(1,2)*XJ(3,3) + XJ(1,3)*XJ(3,2))
899   XJI(2,2)=DUM*( XJ(1,1)*XJ(3,3) - XJ(1,3)*XJ(3,1))
900   XJI(3,2)=DUM*(-XJ(1,1)*XJ(3,2) + XJ(1,2)*XJ(3,1))
901   XJI(1,3)=DUM*( XJ(1,2)*XJ(2,3) - XJ(1,3)*XJ(2,2))
902   XJI(2,3)=DUM*(-XJ(1,1)*XJ(2,3) + XJ(1,3)*XJ(2,1))
903   XJI(3,3)=DUM*( XJ(1,1)*XJ(2,2) - XJ(1,2)*XJ(2,1))
904 C
905 C
906 C   EVALUATE B MATRIX IN GLOBAL (X,Y,Z) COORDINATES
907 C
908 C
909   DO 130 K=1, IELD
910     K2=K*3
911     DO 115 L=1,3
912       B(L,K2-2) = 0.0
913       B(L,K2-1) = 0.0
914   115 B(L,K2  ) = 0.0
915 C
916 C   DIRECT STRAINS (1=EXX, 2=EYY, 3=EZZ)
917 C
918   DO 120 I=1,3

```



```

919      B(1,K2-2) = B(1,K2-2) + XJI(1,I)* P(I,K)
920      B(2,K2-1) = B(2,K2-1) + XJI(2,I)* P(I,K)
921 120  B(3,K2 ) = B(3,K2 ) + XJI(3,I)* P(I,K)
922 C
923 C      SHEAR STRAINS (4=EXY, 5=EYZ, 6=EZX)
924 C
925      B(4,K2-2) = B(2,K2-1)
926      B(4,K2-1) = B(1,K2-2)
927      B(5,K2-1) = B(3,K2 )
928      B(5,K2 ) = B(2,K2-1)
929      B(6,K2-2) = B(3,K2 )
930 130  B(6,K2 ) = B(1,K2-2)
931 C
932 C
933      RETURN
934 C
935      END
936 C=====
937      SUBROUTINE ELTYPE(MTYPE)
938 C
939      IMPLICIT REAL8(A-H,O-Z)
940 C
941 C      CALLED BY:  MAIN,STRESS
942 C
943      GO TO (1,2,3,4,5,6,7,8,9,10,11,12),MTYPE
944 C
945 C      THREE DIMENSIONAL TRUSS ELEMENTS
946 C
947      1 CONTINUE
948 C      1 CALL TRUSS
949      GO TO 900
950 C
951 C      THREE DIMENSIONAL BEAM ELEMENTS
952 C
953      2 CONTINUE
954 C      2 CALL BEAM
955      GO TO 900
956 C
957 C      PLANE STRESS ELEMENTS
958 C
959      3 CONTINUE
960 C      3 CALL PLANE
961      GO TO 900
962 C
963 C      AXISYMMETRIC SOLID ELEMENTS
964 C
965      4 CONTINUE
966 C      4 CALL PLANE
967      GO TO 900
968 C
969 C      THREE DIMENSIONAL SOLID ELEMENTS
970 C
971      5 CONTINUE
972 C      5 CALL THREED

```

```

973      GO TO 900
974 C
975 C      PLATE BENDING ELEMENTS
976 C
977      6 CONTINUE
978 C      6 CALL SHELL
979      GO TO 900
980 C
981 C
982      7 CALL BOUND
983      GO TO 900
984 C
985 C      THICK SHELL ELEMENTS
986 C
987      8 CALL SOL21
988      GO TO 900
989 C
990      9 WRITE (33,100) MTYPE
991      GO TO 900
992 C
993      10 WRITE (33,100) MTYPE
994      GO TO 900
995 C
996      11 WRITE (33,100) MTYPE
997      GO TO 900
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 ('ELEMENT',I4,' IS NOT IMPLEMENTED YET')
1007      END
1008 C=====
1009      SUBROUTINE ERROR(N)
1010      WRITE (33,2000) N
1011      2000 FORMAT ('// 20M STORAGE EXCEEDED BY  I6)
1012      STOP
1013      END
1014 C=====
1015      SUBROUTINE FACEPR (NEL,KDIS,KXYZ,XX,NOD9,H,P,PL,NEACE,LT,PWA,KLS)
1016 C
1017 C      CALLED BY : THDFE
1018 C      CALLS : FNCT
1019 C
1020      IMPLICIT REAL*8(A-H,O-Z)
1021 C
1022 C
1023 C      THIS ROUTINE COMPUTES NODE FORCES DUE TO APPLIED ELEMENT FACE
1024 C      PRESSURE DISTRIBUTIONS
1025 C
1026 C

```

```

1027     DIMENSION   XX(3,1),NOD9(1),H(1),P(3,1),PL(1),PWA(1)
1028     DIMENSION   XJ(3,3),ETA(3),KEACE(6,8),KCRD(6),EVAL(6),IPRM(3),
1029     1           PR(8),NODES(8),IPR4(4)
1030     COMMON /GAUSS/ XK(4,4),WGT(4,4)
1031 C
1032     DATA KEACE / 1, 2, 1, 4, 1, 5,
1033     1           4, 3, 5, 8, 2, 6,
1034     2           8, 7, 6, 7, 3, 7,
1035     3           5, 6, 2, 3, 4, 8,
1036     4           12, 10, 17, 20, 9, 13,
1037     5           20, 19, 13, 15, 10, 14,
1038     6           16, 14, 18, 19, 11, 15,
1039     7           17, 18, 9, 11, 12, 16/
1040 C
1041     DATA KCRD / 1, 1, 2, 2, 3, 3/
1042     DATA EVAL / 1.,-1., 1.,-1., 1.,-1./
1043     DATA IPRM / 2, 3, 1/
1044     DATA IPR4 / 2, 3, 4, 1/
1045 C
1046 C     DETERMINE THE ELEMENT NODES CONTRIBUTING TO FORCE CALCULATIONS
1047 C     ON THIS FACE
1048 C
1049     DO 2 I=1,4
1050     NODES(I) = KEACE(NFACE,I)
1051     NODES(I+4) = 0
1052 2 CONTINUE
1053 C
1054     IF(KDIS.LT.9) GO TO 9
1055 C
1056     NN9 = KDIS-8
1057 C
1058     DO 8 K=5,8
1059     DO 4 I=1,NN9
1060 C
1061     J = I
1062     IF(KEACE(NFACE,K).EQ.NOD9(I)) GO TO 6
1063 C
1064     4 CONTINUE
1065     GO TO 8
1066 C
1067     6 NODES(K) = J
1068     8 CONTINUE
1069 C
1070     9 CONTINUE
1071 C
1072 C     SET UP THE PRESSURE VECTOR FOR THE FOUR FACE CORNER NODES
1073 C
1074 C     1. ADJUST THE SIGN OF THE PRESSURES SO THAT POSITIVE
1075 C     PRESSURE ALWAYS COMPRESSES THE ELEMENT
1076 C
1077     FACT = -EVAL(NFACE)
1078 C
1079     GO TO (10,30), LT
1080 C

```

```

1081 C           2. DISTRIBUTED PRESSURE GIVEN AT THE CORNER NODES
1082 C
1083   10 DO 25 K=1,8
1084 C
1085           IF(NODES(K).EQ.0) GO TO 25
1086 C
1087           IF(K.GT.4) GO TO 15
1088 C
1089           PR(K) = PWA(K) * FACT
1090           GO TO 25
1091 C
1092   15 J = K-4
1093           L = IPR4(J)
1094           PR(K) = (PWA(J) + PWA(L)) * 0.5 * FACT
1095 C
1096   25 CONTINUE
1097           GO TO 75
1098 C
1099 C           3. ELEMENT FACE EXPOSED TO HYDROSTATIC PRESSURE
1100 C
1101   30 GAMMA = PWA(1) * FACT
1102 C
1103           XLN = 0.0
1104           DO 35 K=1,3
1105           ETA(K) = PWA(K+4) - PWA(K+1)
1106   35 XLN = XLN + ETA(K)**2
1107           XLN =DSQRT(XLN)
1108 C
1109           IF(XLN.GT.1.0E-6) GO TO 40
1110 C
1111           WRITE (33,3000) KLS,NEL
1112   3000 FORMAT (31HOERROR*** PRESSURE LOAD SET (,I3,15H) FOR ELEMENT (,
1113           1      15,43H) HAS UNDEFINED HYDROSTATIC SURFACE NORMAL., / 1X)
1114           STOP
1115 C
1116   40 DO 45 K=1,3
1117   45 ETA(K) = ETA(K) / XLN
1118 C
1119           DO 70 N=1,8
1120 C
1121           IF(NODES(N).EQ.0) GO TO 70
1122 C
1123           XLN = 0.0
1124           NOD = NODES(N)
1125           IF(N.GT.4) NOD = NOD + 8
1126 C
1127           DO 50 I=1,3
1128   50 XLN = XLN + (XX(I,NOD) - PWA(I+1)) * ETA(I)
1129 C
1130           PR(N) = XLN * GAMMA
1131 C
1132           IF(XLN.LT.0.0) PR(N) = 0.0
1133 C
1134   70 CONTINUE

```

```

1135 75 CONTINUE
1136 C
1137 C   SET UP VARIABLES FOR THE SURFACE INTEGRATION
1138 C
1139     ML = KCRD(NFACE)
1140     MM = IPKM(ML)
1141     MN = IPKM(MM)
1142 C
1143 C   SURFACE INTEGRATION LOOP
1144 C
1145     ETA(ML) = EVAL(NFACE)
1146 C
1147     DO 300 LX=1,3
1148 C
1149       ETA(MM) = AK(LX,3)
1150 C
1151     DO 300 LY=1,3
1152 C
1153       ETA(MN) = AK(LY,3)
1154 C
1155       WT = WGT(LX,3)*WGT(LY,3)
1156 C
1157 C   EVALUATE THE INTERPOLATION FUNCTIONS AND JACOBIAN MATRIX
1158 C
1159     CALL FNCT (ETA(1),ETA(2),ETA(3),H,P,MOD9,XJ,DET,XX,KDIS,KX12,NEL)
1160 C
1161 C   COMPUTE THE DIRECTION COSINES OF THE UNIT SURFACE NORMAL VECTOR
1162 C   AT THIS SAMPLE POINT
1163 C
1164     A1 = XJ(MM,2)*XJ(MN,3)-XJ(MM,3)*XJ(MN,2)
1165     A2 = XJ(MM,3)*XJ(MN,1)-XJ(MM,1)*XJ(MN,3)
1166     A3 = XJ(MM,1)*XJ(MN,2)-XJ(MM,2)*XJ(MN,1)
1167 C
1168     AA = DSQRT(A1**2 + A2**2 + A3**2)
1169     IF(AA.GT.1.0E-8) GO TO 100
1170 C
1171     WRITE (33,3010) NFACE,NEL
1172 3010 FORMAT (38H0ERRRRA** UNDEFINED NORMAL IN FACE (,I1,5H) FOR,
1173 1          10H ELEMENT (,I5,2H)., / 1X)
1174     STOP
1175 C
1176 100 FACT = 1.0/AA
1177     A1 = A1*FACT
1178     A2 = A2*FACT
1179     A3 = A3*FACT
1180 C
1181 C   COMPUTE THE FIRST FUNDAMENTAL FORM (AREA DIFFERENTIAL)
1182 C
1183     AA = 0.0
1184     BB = 0.0
1185     CC = 0.0
1186     DO 120 I=1,3
1187     AA = AA + XJ(MM,I)**2
1188     CC = CC + XJ(MN,I)**2

```

```

1189 120 BB = BB + XJ(MN,1)*XJ(MN,1)
1190      C = DSQRT(AAACC - BB**2)
1191 C
1192 C   INTERPOLATE FOR THE PRESSURE AT THIS SAMPLE POINT
1193 C
1194      PRESS = 0.0
1195 C
1196      DO 130 K=1,8
1197 C
1198      IF(NODES(K).EQ.0) GO TO 130
1199 C
1200          NOD = NODES(K)
1201      IF(K.GT.4) NOD = NOD + 8
1202 C
1203      PRESS = PRESS + H(NOD)*PR(K)
1204 130 CONTINUE
1205 C
1206      FACT = WT*CA*PRESS
1207 C
1208 C   ASSEMBLE THE NODE FORCE CONTRIBUTION
1209 C
1210      DO 160 L=1,8
1211 C
1212      IF(NODES(L).EQ.0) GO TO 160
1213 C
1214      IF(L.GT.4) GO TO 140
1215 C
1216 C       1. CORNER NODES
1217 C
1218      N = NODES(L)
1219      K = 3*N
1220      GO TO 150
1221 C
1222 C       2. SIDE NODES
1223 C
1224 140 J = NODES(L)
1225      N = J+8
1226      K = 3*NOD9(J)
1227 C
1228 150 QU = FACT*H(N)
1229 C
1230      PL(K-2) = PL(K-2) + QU*A1
1231      PL(K-1) = PL(K-1) + QU*A2
1232      PL(K ) = PL(K ) + QU*A3
1233 160 CONTINUE
1234 C
1235 300 CONTINUE
1236 C
1237      RETURN
1238      END
1239 C&=====
1240      SUBROUTINE FNCT (K,S,T,H,P,NOD9,XJ,DET,XX,IELD,IELX,NEL)
1241 C
1242 C   CALLED BY : FACEPR

```

```

1243 C
1244     IMPLICIT REAL*8(A-H,O-Z)
1245 C
1246 C
1247 C . . . . .
1248 C .
1249 C .   P R O G R A M
1250 C .
1251 C .     TO FIND INTERPOLATION FUNCTIONS ( H )
1252 C .     AND DERIVATIVES ( P ) CORRESPONDING TO THE NODAL
1253 C .     POINTS OF A CURVILINEAR ISOPARAMETRIC HEXAHEDRON
1254 C .     OR SUBPARAMETRIC HEXAHEDRON ( 8 TO 21 NODES )
1255 C .
1256 C .     TO FIND JACOBIAN ( XJ ) AND ITS DETERMINANT ( DET )
1257 C .
1258 C . . . . .
1259 C
1260 C
1261     DIMENSION H(1),P(3,1),NOD9(1),IPERM(8),XJ(3,3),XX(3,1)
1262 C
1263     DATA IPERM / 2,3,4,1,6,7,8,5 /
1264 C
1265     IEL = IELD
1266     NND9= IELD-8
1267 C
1268     RP=1.0 + R
1269     SP=1.0 + S
1270     TP=1.0 + T
1271     RM=1.0 - R
1272     SM=1.0 - S
1273     TM=1.0 - T
1274     RR=1.0 - R*R
1275     SS=1.0 - S*S
1276     TT=1.0 - T*T
1277 C
1278 C
1279 C     INTERPOLATION FUNCTIONS AND THEIR DERIVATIVES
1280 C
1281 C
1282 C     8-NODE BRICK
1283 C
1284     H(1)=0.125*RP*SP*TP
1285     H(2) =0.125*RM*SP*TP
1286     H(3)=0.125*RM*SM*TP
1287     H(4)=0.125*RP*SM*TP
1288     H(5)=0.125*RP*SP*TM
1289     H(6)=0.125*RM*SP*TM
1290     H(7)=0.125*RM*SM*TM
1291     H(8)=0.125*RP*SM*TM
1292 C
1293     P(1,1)=0.125*SP*TP
1294     P(1,2)=-P(1,1)
1295     P(1,3)=-0.125*SM*TP
1296     P(1,4)=-P(1,3)

```

```

1297      P(1,5)=0.125*SP*TM
1298      P(1,6)=-P(1,5)
1299      P(1,7)=-0.125*SM*TM
1300      P(1,8)=-P(1,7)
1301 C
1302      P(2,1)=0.125*RP*TP
1303      P(2,2)=0.125*RM*TP
1304      P(2,3)=-P(2,2)
1305      P(2,4)=-P(2,1)
1306      P(2,5)=0.125*RP*TM
1307      P(2,6)=0.125*RM*TM
1308      P(2,7)=-P(2,6)
1309      P(2,8)=-P(2,5)
1310 C
1311      P(3,1)=0.125*RP*SP
1312      P(3,2)=0.125*RM*SP
1313      P(3,3)=0.125*RM*SM
1314      P(3,4)=0.125*RP*SM
1315      P(3,5)=-P(3,1)
1316      P(3,6)=-P(3,2)
1317      P(3,7)=-P(3,3)
1318      P(3,8)=-P(3,4)
1319 C
1320      IF(IEL.EQ.8) GO TO 50
1321 C
1322 C
1323 C      ADD DEGREES OF FREEDOM IN EXCESS OF 8
1324 C
1325      I=0
1326      2 I=I + 1
1327      IF(I.GT.NND9) GO TO 40
1328      NN=NDD9(I) - 8
1329      GO TO (9,10,11,12,13,14,15,16,17,18,19,20,21) ,NN
1330 C
1331      9 H(9) = 0.25*RR*SP*TP
1332      P(1,9) = -0.50*RR*SP*TP
1333      P(2,9) = 0.25*RR*TP
1334      P(3,9) = 0.25*RR*SP
1335      GO TO 2
1336      10 H(10)=0.25*RM*SS*TP
1337      P(1,10)=-0.25*SS*TP
1338      P(2,10)=-0.50*RM*SS*TP
1339      P(3,10)= 0.25*RM*SS
1340      GO TO 2
1341      11 H(11)=0.25*RR*SM*TP
1342      P(1,11)=-0.50*RR*SM*TP
1343      P(2,11)=-0.25*RR*TP
1344      P(3,11)= 0.25*RR*SM
1345      GO TO 2
1346      12 H(12)=0.25*RP*SS*TP
1347      P(1,12)= 0.25*SS*TP
1348      P(2,12)=-0.50*RP*SS*TP
1349      P(3,12)= 0.25*RP*SS
1350      GO TO 2

```



```

1351      13 H(13)=0.25*RR*SP*TM
1352      P(1,13)=-0.50*RR*SP*TM
1353      P(2,13)= 0.25*RR*TM
1354      P(3,13)=-0.25*RR*SP
1355      GO TO 2
1356      14 H(14)=0.25*RM*SS*TM
1357      P(1,14)=-0.25*SS*TM
1358      P(2,14)=-0.50*RM*S*TM
1359      P(3,14)=-0.25*RM*SS
1360      GO TO 2
1361      15 H(15)=0.25*RR*S*TM
1362      P(1,15)=-0.50*RR*S*TM
1363      P(2,15)=-0.25*RR*TM
1364      P(3,15)=-0.25*RR*S*TM
1365      GO TO 2
1366      16 H(16)=0.25*RP*SS*TM
1367      P(1,16)= 0.25*SS*TM
1368      P(2,16)=-0.50*RP*S*TM
1369      P(3,16)=-0.25*RP*SS
1370      GO TO 2
1371      17 H(17)=0.25*RP*SP*TT
1372      P(1,17)=0.25*SP*TT
1373      P(2,17)=0.25*RP*TT
1374      P(3,17)=-0.50*RP*SP*TT
1375      GO TO 2
1376      18 H(18)=0.25*RM*SP*TT
1377      P(1,18)=-0.25*SP*TT
1378      P(2,18)= 0.25*RM*TT
1379      P(3,18)=-0.50*RM*SP*TT
1380      GO TO 2
1381      19 H(19)=0.25*RM*S*TT
1382      P(1,19)=-0.25*S*TT
1383      P(2,19)=-0.25*RM*TT
1384      P(3,19)=-0.50*RM*S*TT
1385      GO TO 2
1386      20 H(20)=0.25*RP*S*TT
1387      P(1,20)= 0.25*S*TT
1388      P(2,20)=-0.25*RP*TT
1389      P(3,20)=-0.50*RP*S*TT
1390      GO TO 2
1391      21 H(21)=RR*SS*TT
1392      P(1,21)=-2.0*RR*SS*TT
1393      P(2,21)=-2.0*S*RR*TT
1394      P(3,21)=-2.0*T*RR*SS
1395      GO TO 2
1396 C
1397 C      MODIET FIRST 8 FUNCTIONS IF 9 OR MORE NODES IN ELEMENT
1398 C
1399      40 IH=0
1400      41 IH=IH + 1
1401      IF (IH.GT.NND9) GO TO 50
1402      II=IH + 7
1403      IF (II.EQ.IELX) GO TO 51
1404      42 IN=NOD9(IH)

```

```

1405     IF (IN.GT.16) GO TO 46
1406     I1=IN -8
1407     I2=IPERM(I1)
1408     H(I1)=H(I1) - 0.5*H(IN)
1409     H(I2)=H(I2) - 0.5*H(IN)
1410     H(IH+8)=H(IN)
1411     DO 45 J=1,3
1412     P(J,I1)=P(J,I1) - 0.5*P(J,IN)
1413     P(J,I2)=P(J,I2) - 0.5*P(J,IN)
1414     45 P(J,IH+8)=P(J,IN)
1415     GO TO 41
1416     46 IF (IN.EQ.21) GO TO 30
1417     I1=IN -16
1418     I2=I1 + 4
1419     H(I1)=H(I1) - 0.5*H(IN)
1420     H(I2)=H(I2) - 0.5*H(IN)
1421     H(IH+8)=H(IN)
1422     DO 47 J=1,3
1423     P(J,I1)=P(J,I1) - 0.5*P(J,IN)
1424     P(J,I2)=P(J,I2) - 0.5*P(J,IN)
1425     47 P(J,IH+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=NOD9(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*H(21)
1438     H(I2)=H(I2) + 0.125*H(21)
1439     DO 32 J=1,3
1440     P(J,I1)=P(J,I1) + 0.125*P(J,21)
1441     32 P(J,I2)=P(J,I2) + 0.125*P(J,21)
1442     GO TO 31
1443     33 I1=IN - 16
1444     I2=I1 + 4
1445     H(I1)=H(I1) + 0.125*H(21)
1446     H(I2)=H(I2) + 0.125*H(21)
1447     DO 34 J=1,3
1448     P(J,I1)=P(J,I1) + 0.125*P(J,21)
1449     34 P(J,I2)=P(J,I2) + 0.125*P(J,21)
1450     GO TO 31
1451     35 DO 36 I=1,8
1452     H(I)=H(I) - 0.125*H(21)
1453     DO 36 J=1,3
1454     36 P(J,I)=P(J,I) - 0.125*P(J,21)
1455     NN=RND9 + 7
1456     IF (NN.EQ.8) GO TO 50
1457     DO 38 I=9,NN
1458     H(I)=H(I) - 0.25*H(21)

```

```

1459      DO 38 J=1,3
1460      38 P(J,I)=P(J,I) - 0.25*P(J,21)
1461      H(NND9+8)=H(21)
1462      DO 39 J =1,3
1463      39 P(J,NND9+8)=P(J,21)
1464 C
1465 C
1466 C      EVALUATE JACOBIAN MATRIX AT POINT (R,S,T)
1467 C
1468 C
1469      50 IF (IELX.LT.IELD) RETURN
1470      51 DO 100 I=1,3
1471      DO 100 J=1,3
1472      DUM=0.0
1473      DO 90 K=1, IELX
1474      90 DUM=DUM + P(I,K)*XX(J,K)
1475      100 XJ(I,J)=DUM
1476 C
1477 C
1478 C      COMPUTE DETERMINANT OF JACOBIAN MATRIX AT POINT (R,S,T)
1479 C
1480 C
1481      DET = XJ(1,1)*XJ(2,2)*XJ(3,3)
1482      1   + XJ(1,2)*XJ(2,3)*XJ(3,1)
1483      2   + XJ(1,3)*XJ(2,1)*XJ(3,2)
1484      3   - XJ(1,3)*XJ(2,2)*XJ(3,1)
1485      4   - XJ(1,2)*XJ(2,1)*XJ(3,3)
1486      5   - XJ(1,1)*XJ(2,3)*XJ(3,2)
1487      IF(DET.GT.1.0E-8) GO TO 110
1488      WRITE (33,2000) NEL,R,S,T
1489      STOP
1490      110 IF (IELX.LT.IELD) GO TO 42
1491 C
1492 C
1493      RETURN
1494 C
1495 C
1496 C
1497      2000 FORMAT (49H0ERROR***  NEGATIVE OR ZERO JACOBIAN DETERMINANT,
1498      1          23H COMPUTED FOR ELEMENT (,IS,1H), /
1499      2          12X, 3HR =, F10.5 /
1500      3          12X, 3HS =, F10.5 /
1501      4          12X, 3HT =, F10.5 / 1X)
1502 C
1503 C
1504      END
1505 C&=====
1506      SUBROUTINE INL(ID,B,TR,IMASS,NUMNP,NEQB,LL)
1507 C
1508      IMPLICIT REAL*8(A-H,O-Z)
1509 C
1510 C      CALLED BY:  MAIN
1511 C
1512 C      INPUT NODAL LOADS AND MASSES

```

```

1513 C
1514     DIMENSION ID(NUMNP,6),B(NEQB,LL),TR(6,LL),TMASS(NEQB)
1515     COMMON / JUNK / R(6),TXM(6),IFILL1(406)
1516     COMMON /EXTRA/ MODEX,NTB,IFILL2(14)
1517 C
1518     NT=3
1519     REWIND NTB
1520     KSHE=0
1521     WRITE (33,2002)
1522     IF(MODEX.EQ.1) GO TO 50
1523     DO 750 I=1,NEQB
1524     TMASS(I)=0.
1525     DO 750 K=1,LL
1526 750 B(I,K)=0.0
1527 C
1528     50 DO 900 NN=1,NUMNP
1529 C
1530     DO 100 I=1,6
1531     TXM(I)=0.
1532     DO 100 J=1,LL
1533 100 TR(I,J)=0.0
1534 C
1535     IF(NN.EQ.1) GO TO 300
1536 150 IF(N.NE.NN) GO TO 400
1537     DO 200 I=1,6
1538     IF (L) 180,180,190
1539 180 TXM(I)=R(I)
1540     GO TO 200
1541 190 TR(I,L)=R(I)
1542 200 CONTINUE
1543 300 READ (5,1001) N,L,R
1544     IF (N.EQ.0) GO TO 150
1545     WRITE(33,2001) N,L,R
1546     GO TO 150
1547 C
1548 400 IF(MODEX.EQ.1) GO TO 900
1549     DO 300 J=1,6
1550     II=ID(NN,J)-KSHE
1551     IF (II) 800,800,500
1552 500 DO 600 K=1,LL
1553 600 B(II,K)=TR(J,K)
1554     TMASS(II)=TXM(J)
1555 610 IF(II.NE.NEQB) GO TO 800
1556     WRITE (NT) B,TMASS
1557     KSHE=KSHE+NEQB
1558     DO 700 I=1,NEQB
1559     TMASS(I)=0.
1560     DO 700 K=1,LL
1561 700 B(I,K)=0.0
1562 800 CONTINUE
1563 900 CONTINUE
1564 C
1565     IF(MODEX.EQ.1) RETURN
1566 C

```

```

1567      WRITE (NT) B, TMASS
1568 C
1569      RETURN
1570 1001 FORMAT (2I5,7F10.4)
1571 2001 FORMAT (2(3X,I4),6E15.5)
1572 2002 FORMAT (47H1N O D A L   L O A D S   ( S T A T I C )   O R   ,
1573      A       29HM A S S E S   ( D Y N A M I C ) ,   / / /
1574      B       3X,4HNODE,2X,4HLOAD,
1575      1 2(9X,6HX-AXIS,9X,6HY-AXIS,9X,6HZ-AXIS), / 7H NUMBER,3X,4HCASE,
1576      2 3(10X,5HFORCE), 3(9X,6HMOMENT), / 1X)
1577      END
1578 C=====
1579      SUBROUTINE INPUTJ(ID,X,Y,Z,T,NUMNP,NEQ)
1580 C
1581      IMPLICIT REAL*8(A-H,O-Z)
1582 C
1583 C      CALLED BY:  MAIN
1584 C
1585      DIMENSION X(1),Y(1),Z(1),ID(NUMNP,6),T(1)
1586 C
1587      COMMON /EXTRA/ MDEX,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. , 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 *** AND *B*
1598 C
1599      DIMENSION IPRC(4)
1600 C
1601      DATA IPRC/1H ,1HA,1HB,1HC/
1602 C
1603      IPR = IPRC(1)
1604      RAD = DATAN(1.0D0)/45.0D0
1605 C
1606 C
1607 C---- READ OR GENERATE NODAL POINT DATA-----
1608      WRITE (33,2000)
1609      WRITE (33,2001)
1610      NOLD=0
1611      10 READ  (5,1000) IT,N,JPR,(ID(N,I),I=1,6),X(N),Y(N),Z(N),KN,T(N)
1612 C
1613 C
1614 C** ** ** NEXT LINE IS DELETED FOR NOT PRINTING NODAL INPUT DATA
1615 C      WRITE (33,2002) IT,N,JPR,(ID(N,I),I=1,6),X(N),Y(N),Z(N),KN,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)* RAD

```

```

1621      Z(N) = X(N)*DCOS(DUM)
1622      X(N) = X(N)*DSIN(DUM)
1623      15 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(N,I).EQ.0.AND.ID(NOLD,I).LT.0) ID(N,I)=ID(NOLD,I)
1628      20 CONTINUE
1629      IF(KN.EQ.0) GO TO 50
1630      NUM=(N-NOLD)/KN
1631      NUMN=NUM-1
1632      IF(NUMN.LT.1) GO TO 50
1633      XNUM=NUM
1634      DX=(X(N)-X(NOLD))/XNUM
1635      DY=(Y(N)-Y(NOLD))/XNUM
1636      DZ=(Z(N)-Z(NOLD))/XNUM
1637      DT=(T(N)-T(NOLD))/XNUM
1638      K=NOLD
1639      DO 30 J=1,NUMN
1640      KK=K
1641      K=K+KN
1642      X(K)=X(KK)+DX
1643      Y(K)=Y(KK)+DY
1644      Z(K)=Z(KK)+DZ
1645      T(K)=T(KK)+DT
1646      DO 30 I=1,6
1647      ID(K,I)=ID(KK,I)
1648      IF (ID(K,I).GT.1) ID(K,I)=ID(KK,I)+KN
1649      30 CONTINUE
1650 C
1651      50 NOLD=N
1652      IF(N.NE.NUMNP) GO TO 10
1653 C
1654 C---- PRINT ALL NODE/L POINT DATA-----
1655 C
1656      IF(IPR.EQ.IPRC(2) .OR. IPR.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      52 CONTINUE
1661 C
1662 C-----NUMBER UNKNOWN AND SET MASTER NODES NEGATIVE-----
1663 C
1664      NEQ=0
1665      DO 60 N=1,NUMNP
1666      DO 60 I=1,6
1667      ID(N,I)=IABS(ID(N,I))
1668      IF(ID(N,I)-1) 57,58,59
1669      57 NEQ=NEQ+1
1670      ID(N,I)=NEG
1671      GO TO 60
1672      58 ID(N,I)=0
1673      GO TO 60
1674      59 ID(N,I)=-ID(N,I)

```

```

1675     60 CONTINUE
1676 C
1677 C---- PRINT MASTER INDEX ARRAY
1678 C
1679     IF(IPR.EQ.IPRC(3) .OR. IPR.EQ.IPRC(4)) GO TO 62
1680     WRITE (33,2004) (N,(ID(N,I),I=1,6),N=1,NUMNP)
1681     62 CONTINUE
1682     IF(MODEX.EQ.0) GO TO 70
1683 C*** DATA PORTHOLE SAVE
1684     WRITE (NT8) ((ID(N,I),I=1,6),N=1,NUMNP)
1685     WRITE (NT8) (X(N),N=1,NUMNP)
1686     WRITE (NT8) (Y(N),N=1,NUMNP)
1687     WRITE (NT8) (Z(N),N=1,NUMNP)
1688     WRITE (NT8) (T(N),N=1,NUMNP)
1689     ENDFILE NT8
1690 C
1691     REWIND 2
1692     WRITE (2) ID
1693 C
1694     RETURN
1695 C
1696     70 CONTINUE
1697     REWIND 8
1698     WRITE (8) ID
1699 C
1700     RETURN
1701 C
1702 1000 FORMAT (2(A1,I4),5I5,3F10.0,I5,F10.0)
1703 2000 FORMAT (//23H NODAL POINT INPUT DATA )
1704 2001 FORMAT (5HONODE 3X 24HBOUNDARY CONDITION CODES 11X
1705     . 23HNODAL POINT COORDINATES / 7H NUMBER 2X 1HX 4X 1HY 4X 1HZ 3X
1706     . 2HXX 3X 2HYY 3X 2HZZ12X 1HX 12X 1HY 12X 1HZ 12X 1HT )
1707 C
1708 C
1709 C** ** ** NEXT LINE IS IGNORED WITH LINE #31600
1710 C2002 FORMAT (1X,A1,I4,A1,I3,5I5,3F13.3,I5,F13.3)
1711 C** ** **
1712 C
1713 C
1714 2003 FORMAT (//21H1GENERATED NODAL DATA)
1715 2004 FORMAT (//17H1EQUATION NUMBERS/
1716     1 35H   N   X   Y   Z   XX   YY   ZZ /(7I5))
1717 2005 FORMAT (I5,6I5,4F13.3)
1718     END
1719 C&=====
1720     SUBROUTINE INP21 (NUMMAT,MAXTP,NORTHO,NDLS,NOPSET,NT8SV,NUMNP,X,
1721     1           Y,Z,DEN,RHO,NTP,EE,DCA,NEACE,LT,PWA,LOC,MAXPTS)
1722 C
1723 C     CALLED BY : THDFE
1724 C     CALLS : VECTR2,CROSS2
1725 C
1726     IMPLICIT REAL8(A-H,O-Z)
1727 C
1728 C

```

```

1729 C THIS ROUTINE READS AND PRINTS ALL 21-NODE SOLID ELEMENT DATA
1730 C BETWEEN THE CONTROL CARD AND THE ELEMENT DATA CARDS
1731 C
1732 C
1733 COMMON / JUNK/ XLF(4),YLE(4),ZLF(4),TLF(4),PLF(4),FILL1(22),V2(3)
1734 COMMON /EXTRA/ MODEX,NT8
1735 C
1736 DIMENSION X(1),Y(1),Z(1),DEN(1),RHO(1),NTP(1),EE(MAXTP,13,1),
1737 1 DCA(3,3,1),NEFACE(1),LT(1),PWA(7,1),LOC(7,1),
1738 2 MAXPTS(1)
1739 DIMENSION HED(6)
1740 C
1741 C READ AND PRINT OF MATERIAL PROPERTIES
1742 C
1743 WRITE (33,3000)
1744 C
1745 DO 10 I=1,NUMMAT
1746 C
1747 READ (5,1001) m,NTP(I),DEN(I),RHO(I),(HED(N),N=1,6)
1748 C
1749 C SET DEFAULT VALUES IF REQUIRED AND CHECK FOR INPUT ERRORS
1750 C
1751 IF(RHO(I).EQ.0.0) RHO(I) = DEN(I) / 386.4
1752 IF(NTP(I).EQ.0) NTP(I) = 1
1753 C
1754 WRITE (33,3002) m,NTP(I),DEN(I),RHO(I),(HED(N),N=1,6)
1755 C
1756 IF(I.EQ.M) GO TO 2
1757 WRITE (33,4001)
1758 STOP
1759 C
1760 2 IF(NTP(M).LE.MAXTP) GO TO 4
1761 WRITE (33,4002) MAXTP
1762 STOP
1763 4 NT = NTP(M)
1764 C
1765 C READ PROPERTIES FOR EACH TEMPERATURE
1766 C
1767 DO 6 K=1,NT
1768 READ (5,1002) (EE(K,L,m),L=1,13)
1769 WRITE (33,3003) (EE(K,L,m),L=1,13)
1770 6 CONTINUE
1771 C
1772 C TEMPERATURE CARDS MUST BE ASCENDING ORDER
1773 C
1774 IF(NT.EQ.1) GO TO 10
1775 DO 8 J=2,NT
1776 IF(EE(J,1,m).GT.EE(J-1,1,m)) GO TO 8
1777 WRITE (33,4003)
1778 STOP
1779 8 CONTINUE
1780 10 CONTINUE
1781 C*** DATA PORTHOLE SAVE
1782 IF(NT8SV.EQ.0) GO TO 12

```



```

1783      DO 11 M=1,NUMMAT
1784      WRITE (NT8) M,NIP(M),DEN(M),RHO(M)
1785      NT = NIP(M)
1786      WRITE (NT8) ((EE(K,L,M),L=1,13),K=1,NT)
1787      11 CONTINUE
1788 C***
1789 C
1790 C      MATERIAL AXIS ORIENTATION SETS
1791 C
1792      12 IF(NORTH0.EQ.0) GO TO 21
1793 C
1794      WRITE (33,3004)
1795 C
1796      DO 20 M=1,NORTH0
1797      READ (5,1003) N,NI,NJ,NK
1798      WRITE (33,3005) N,NI,NJ,NK
1799 C
1800 C*** DATA PORTHOLE SAVE
1801      IF(NT8SV.EQ.1)
1802      *WRITE (NT8)      N,NI,NJ,NK
1803 C***
1804 C      CHECK FOR ADMISSABILITY OF DATA
1805 C
1806      IF(N.EQ.M) GO TO 13
1807      WRITE (33,4004)
1808      STOP
1809 C
1810      13 IF(NI.GT.0 .AND. NI.LE.NUMNP) GO TO 5015
1811      L = NI
1812      5014 WRITE (33,4005) L
1813      STOP
1814      5015 IF(NJ.GT.0 .AND. NJ.LE.NUMNP) GO TO 5016
1815      L = NJ
1816      GO TO 5014
1817      5016 IF(NK.GT.0 .AND. NK.LE.NUMNP) GO TO 14
1818      L = NK
1819      GO TO 5014
1820      14 CONTINUE
1821 C
1822 C      GENERATE DIRECTION COSINE ARRAY FOR THIS DATA SET
1823 C
1824      CALL VECTR2 (DCA(1,1,M),X(NI),Y(NI),Z(NI),X(NJ),Y(NJ),Z(NJ),IERR)
1825      IF(IERR.EQ.0) GO TO 16
1826      WRITE (33,4006)
1827      STOP
1828      16 CALL VECTR2 (V2,X(NI),Y(NI),Z(NI),X(NK),Y(NK),Z(NK),IERR)
1829      IF(IERR.EQ.0) GO TO 17
1830      WRITE (33,4007)
1831      STOP
1832      17 CALL CROSS2 (DCA(1,1,M),V2,DCA(1,3,M),IERR)
1833      IF(IERR.EQ.0) GO TO 18
1834      WRITE (33,4008)
1835      STOP
1836      18 CALL CROSS2 (DCA(1,3,M),DCA(1,1,M),DCA(1,2,M),IERR)

```

```

1837     IF(IERR.EQ.0) GO TO 20
1838     WRITE (33,4009)
1839     STOP
1840 20 CONTINUE
1841 C
1842 C     READ AND PRINT DISTRIBUTED SURFACE LOAD DATA
1843 C
1844 21 IF(NDLS.EQ.0) GO TO 31
1845 C
1846     WRITE (33,3006)
1847 C
1848     DO 30 M=1,NDLS
1849 C
1850     READ (5,1004) N,NFACE(M),LT(M)
1851     WRITE (33,3007) N,NFACE(M),LT(M)
1852 C
1853 C     CHECK FOR DATA ADMISSABILITY
1854 C
1855     IF(N.EQ.M) GO TO 22
1856     WRITE (33,4010)
1857     STOP
1858 22 IF(NFACE(M).GE.1 .AND. NFACE(M).LE.6) GO TO 23
1859     WRITE (33,4011)
1860     STOP
1861 23 IF(LT(M).EQ.0) LT(M) = 1
1862     IF(LT(M).GE.1 .AND. LT(M).LE.2) GO TO 24
1863     WRITE (33,4012)
1864     STOP
1865 24 IF(LT(M).EQ.2) GO TO 26
1866     READ (5,1005) (PWA(I,M),I=1,4)
1867     DO 25 I=3,4
1868 25 IF(PWA(I,M).EQ.0.0) PWA(I,M) = PWA(1,M)
1869     WRITE (33,3008) (PWA(I,M),I=1,4)
1870     GO TO 30
1871 26 READ (5,1005) (PWA(I,M),I=1,7)
1872     WRITE (33,3009) (PWA(I,M),I=1,7)
1873 30 CONTINUE
1874 C
1875 CAAA DATA PORTHOLE SAVE
1876     IF(NTBSV.EQ.0) GO TO 5031
1877     DO 5030 M=1,NDLS
1878     WRITE (NTS) NFACE(M),LT(M),(PWA(I,M),I=1,7)
1879 5030 CONTINUE
1880 5031 CONTINUE
1881 CAAA
1882 C
1883 C     READ AND PRINT OF STRESS OUTPUT REQUEST LOCATION SETS
1884 C
1885 31 IF(NOPSET.EQ.0) GO TO 40
1886 C
1887     WRITE (33,3010) (I,I=1,7)
1888     WRITE (34,*) '---STRESS OUTPUT LOCATIONS---'
1889 C
1890     DO 40 M=1,NOPSET

```

```

1184      BB = 0.0
1185      CC = 0.0
1186      DO 120 I=1,3
1187      AA = AA + XJ(MM,I)**2
1188      CC = CC + XJ(MN,I)**2

```

B-22

```

1891      READ (5,1006) (LOC(I,M),I=1,7)
1892      WRITE (33,3011) M,(LOC(I,M),I=1,7)
1893      WRITE (34,3011) M,(LOC(I,M),I=1,7)
1894 C
1895      L = 0
1896      DO 35 J=1,7
1897      IF(LOC(J,M).EQ.0) GO TO 36
1898      L = L + 1
1899      IF(LOC(J,M).GE.1 .AND. LOC(J,M).LE.27) GO TO 35
1900      WRITE (33,4013) J
1901      MDEX = 1
1902      GO TO 36
1903 35 CONTINUE
1904 C
1905 36 IF(L.GT.0) GO TO 37
1906      L = 1
1907      LOC(1,M) = 21
1908 37 MAXPTS(M) = L
1909 C
1910 40 CONTINUE
1911 C*** DATA PORTHOLE SAVE
1912      IF(NTBSV.EQ.1)
1913      *WRITE (NT8) ((LOC(I,J),I=1,7),J=1,NOPSET)
1914 C***
1915 C
1916 C      ELEMENT LOAD CASE MULTIPLIERS
1917 C
1918 49 WRITE (33,2012)
1919 C
1920      READ (5,1007) XLE,YLE,ZLF,TLF,PLF
1921      WRITE (33,3013) XLE,YLE,ZLF,TLF,PLF
1922 C*** DATA PORTHOLE SAVE
1923      IF(NTBSV.EQ.1)
1924      *WRITE (NT8) XLE,YLE,ZLF,TLF,PLF
1925 C***
1926 C
1927      RETURN
1928 C
1929 C      FORMATS
1930 C
1931 1001 FORMAT (2I5,2F10.0,6A6)
1932 1002 FORMAT (2F10.0/6E10.0)
1933 1003 FORMAT (4I5)
1934 1004 FORMAT(3I5)
1935 1005 FORMAT (2F10.0)
1936 1006 FORMAT (2I5)
1937 1007 FORMAT (4F10.0)
1938 C
1939 3000 FORMAT (//,3SH MATERIAL PROPERTY TABLES
1940 3002 FORMAT (//,2SH MATERIAL NUMBER = (,I3,1H),/
1941 1      1      1SH NUMBER OF, /
1942 2      2      2SH TEMPERATURE POINTS = (,I3,1H),/
1943 3      3      3SH WEIGHT DENSITY = (,E12.4,1H),/
1944 4      4      4SH MASS DENSITY = (,E12.4,1H),/

```

B-30

```

1945      5      23H IDENTIFICATION      = (,6A6,1H),//
1946      6 1X,11HTEMPERATURE,9X,3HE11,9X,3HE22,9X,3HE33,4X,3HV12,4X,3HV13,
1947      7 4X,3HV23,6X,3HG12,8X,3HG13,8X,3HG23,3X,7HALPHA-1,3X,7HALPHA-2,
1948      8 3X,7HALPHA-3,/1X)
1949 3003 FORMAT (F12.2,3F12.1,3F7.3,3F11.1,3E10.3)
1950 3004 FORMAT (//50H M A T E R I A L A X I S O R I E N T A T I O N
1951      1 3X,9HT A B L E      ,//
1952      2 28H      SET      NODE      NODE      NODE      ,/
1953      3 23H NUMBER      NI      NJ      NK, / 1X)
1954 3005 FORMAT (4I7)
1955 3006 FORMAT(//51H D I S T R I B U T E D S U R F A C E L O A D
1956      1      11HT A B L E      ,//,1X)
1957 3007 FORMAT (//7X,27HLOAD SET NUMBER      = ,16 /
1958      1      7X,27HLOAD SURFACE ELEMENT FACE = ,16 /
1959      1      7X,27HLOAD TYPE CODE      = ,16/1X)
1960 3008 FORMAT (12H DISTRIBUTED, 11X,4HP(1),11X,4HP(2),11X,4HP(3),11X,
1961      1      4HP(4), / 4X,8MPRESSURE,4F15.3)
1962 3009 FORMAT (12H HYDROSTATIC,10X,5HGAMMA,11X,4HX(S),11X,4HY(S),11X,
1963      1      4HZ(S),11X,4HX(N),11X,4HY(N),11X,4HZ(N), /
1964      2      4X,8MPRESSURE, 7F15.3)
1965 3010 FORMAT (//51H S T R E S S O U T P U T R E Q U E S T T A B L E ,
1966      * //
1967      4EH      SET ,7(2X,5HPPOINT), / 8H NUMBER ,7(4X,1H(,11,1H)), / 1X)
1968 3011 FORMAT (18,7I7)
1969 3012 FORMAT (//34H E L E M E N T L O A D C A S E      ,3X,
1970      1 21HM U L T I P L I E R S      ,//
1971      *      31X,6HCASE A,4X,6HCASE B,4X,6HCASE C,
1972      2 4X,6HCASE D,/1X)
1973 3013 FORMAT (
1974      1 27H X-DIRECTION GRAVITY =      ,4F10.2/
1975      2 27H Y-DIRECTION GRAVITY =      ,4F10.2/
1976      3 27H Z-DIRECTION GRAVITY =      ,4F10.2/
1977      4 27H THERMAL LOADING      =      ,4F10.2/
1978      5 27H PRESSURE LOADING      =      ,4F10.2 //1X)
1979 C
1980 4001 FORMAT (40HOERRORXXX MATERIAL CARDS OUT OF ORDER.,/1X)
1981 4002 FORMAT (52HOERRORXXX NUMBER OF TEMPERATURE CARDS EXCEEDS USER,
1982      1 10H MAXIMUM (,14,2H)., / 1X)
1983 4003 FORMAT (51HOERRORXXX TEMPERATURES MUST BE INPUT IN ASCENDING
1984      1 7H ORDER., / 1X)
1985 4004 FORMAT (47HOERRORXXX AXIS ORIENTATION CARD OUT OF ORDER., 1X)
1986 4005 FORMAT (38HOERRORXXX UNDEFINED NODE NUMBER = ,15 / 1X)
1987 4006 FORMAT (38HOERRORXXX VECTOR IJ HAS ZERO LENGTH.,/1X)
1988 4007 FORMAT (38HOERRORXXX VECTOR IK HAS ZERO LENGTH.,/1X)
1989 4008 FORMAT (43HOERRORXXX IJ AND IK VECTORS ARE PARALLEL.,/1X)
1990 4009 FORMAT (43HOERRORXXX I3 AND I1 VECTORS ARE PARALLEL.,/1X)
1991 4010 FORMAT (50HOERRORXXX SET NUMBERS MUST BE IN ASCENDING ORDER., 1X)
1992 4011 FORMAT (40HOERRORXXX INVALID SURFACE FACE NUMBER.,/1X)
1993 4012 FORMAT (38HOERRORXXX INVALID LOAD TYPE.,/1X)
1994 4013 FORMAT (42HOERRORXXX INVALID OUTPUT POINT NUMBER = ,15 / 1X)
1995 C
1996 C
1997      END
1998 C3=====

```

```

1999      SUBROUTINE PRINTD (ID,D,B,NEQB,NUMNP,LL,NBLOCK,NEQ,NT,MU)
2000      IMPLICIT REAL*8(A-H,O-Z)
2001 C
2002 C      CALLED BY: SOLEQ,SOLEIG,RESPEC
2003 C
2004      DIMENSION ID(NUMNP,6),B(NEQB,LL),D(6,LL)
2005      DATA Q11,Q21,Q12,Q22,Q13,Q23/' LOAD',' CASE',' EIGEN-',' VECTOR',
2006      1 ' MODE ',' NUMBER'/
2007 C
2008      REWIND 8
2009      READ (8) ID
2010      M=NEQ
2011      NN=NEQB*NBLOCK
2012 C
2013      IF (MQ.EQ.2) GO TO 50
2014      IF (MQ.EQ.3) GO TO 55
2015      REWIND NT
2016      Q1=Q11
2017      Q2=Q21
2018      GO TO 60
2019      50 Q1=Q12
2020      Q2=Q22
2021      GO TO 60
2022      55 Q1=Q13
2023      Q2=Q23
2024      REWIND NT
2025      READ (NT)
2026      60 continue
2027 c-- WRITE (33,2003) Q1,Q2      'removed as there is a print in SOLEQ
2028 C
2029      N=NUMNP
2030      rewind nt      !*****
2031 C
2032      DO 500 KK=1,NUMNP
2033 C
2034      I=6
2035      DO 250 II=1,6
2036      DO 100 L=1,LL
2037      100 D(I,L)=0.
2038      IF(M.GT.NN) GO TO 150
2039      IF (M.EQ.0) GO TO 150
2040      READ (NT) B
2041      NN=NN-NEQB
2042      150 IF(ID(N,I).LT.1) GO TO 250
2043      K=M-NN
2044      M=M-1
2045 C
2046      DO 200 L=1,LL
2047      200 D(I,L)=B(K,L)
2048      250 I=I-1
2049 C
2050 cc- WRITE (33,2004) N,(L,(D(I,L),I=1,6),L=1,LL)
2051 C
2052      500 N=N-1

```

```

2053 C
2054     RETURN
2055 C
2056     2003 FORMAT (1H1,38HNODE DISPLACEMENTS / ,
2057     1       17HR OUT PUT I D N S, // 3X,4HNODE,2X,A6,2(12X,2HX-,1X,
2058     2       2HY-,12X,2HZ-), / 7H NUMBER,2X,A6,3(3X,11HTRANSLATION),
2059     3       3(6X,9HROTATION), / 1X)
2060     2004 FORMAT (16,18,6E14.5, : / (7X,18,6E14.5) )
2061 C-- 2004 FORMAT (1H0,16,18,6E14.5 / (7X,18,6E14.5) )
2062 C
2063     END
2064 C=====
2065     SUBROUTINE SOL31
2066 C
2067 C     CALLED BY : ELTYPE
2068 C     CALLS : STR3C
2069 C
2070     IMPLICIT REAL8(A-H,O-Z)
2071 C
2072 C     3 / D 8 TO 21 NODE SOLID ELEMENTS
2073 C
2074     COMMON /ELPAR/ NPAR(14),NUMNP,MBAND,NELTYP,N1,N2,N3,N4,N5,MTOT,REQ
2075     COMMON /EM/ NS,ND,LM(63)
2076     COMMON/JUNK/ LT,LH,L,N6,SIG(42),N7,N8,N9,N10,N11,N12,N13,N14,
2077     1       N15,N16,N17
2078     COMMON /EXTRA/ MODEX,NT8,N10SV,NT10
2079 C
2080     COMMON   A(1)
2081 C
2082 C
2083     IF(NPAR(1).EQ.0) GO TO 500
2084 C
2085 C     ERROR CHECKS AND SET DEFAULT VALUES IF REQUIRED
2086 C
2087     WRITE (33,1000)
2088     IF(NPAR(2).GT.0) GO TO 10
2089     WRITE (33,1001) (NPAR(K),K=1,10)
2090     WRITE (33,1002)
2091     STOP
2092     10 IF(NPAR(3).GT.0) GO TO 20
2093     WRITE (33,1001) (NPAR(K),K=1,10)
2094     WRITE (33,1003)
2095     STOP
2096     20 IF(NPAR(4).EQ.0) NPAR(4) = 1
2097     IF(NPAR(7).EQ.0) NPAR(7) = 21
2098     IF(NPAR(7).GE.8 .AND. NPAR(7).LE.21) GO TO 30
2099     WRITE (33,1001) (NPAR(K),K=1,10)
2100     WRITE (33,1004)
2101     STOP
2102     30 IF(NPAR(9).EQ.0) NPAR(9) = 2
2103     IF(NPAR(9).GE.2 .AND. NPAR(9).LE.4) GO TO 40
2104     WRITE (33,1001) (NPAR(K),K=1,10)
2105     WRITE (33,1005)
2106     STOP

```

```

2107 40 IF(NPAR(10).EQ.0) NPAK(10) = 2
2108 IF(NPAR(10).GE.2 .AND. NPAR(10).LE.4) GO TO 50
2109 WRITE (33,1001) (NPAR(K),K=1,10)
2110 WRITE (33,1005)
2111 STOP
2112 C
2113 C STORAGE ALLOCATION
2114 C
2115 C A(N6) = STARTING LOCATION OF WEIGHT DENSITY
2116 C A(N7) = STARTING LOCATION OF MASS DENSITY
2117 C A(N8) = STARTING LOCATION OF VECTOR CONTAINING THE ACTUAL
2118 C NUMBER OF TEMPERATURE POINTS FOR EACH MATERIAL TABLE
2119 C A(N9) = STARTING LOCATION OF MATERIAL PROPERTY TABLE
2120 C A(N10) = STARTING LOCATION OF DIRECTION COSINE ARRAYS FOR
2121 C MATERIAL ORIENTATION AXIS
2122 C A(N11) = STARTING LOCATION OF SURFACE LOAD FACE NUMBERS
2123 C A(N12) = STARTING LOCATION OF SURFACE LOAD CODE TYPES
2124 C A(N13) = STARTING LOCATION OF PRESSURE WORKING ARRAY
2125 C A(N14) = STARTING LOCATION OF OUTPUT REQUEST LOCATION SETS
2126 C A(N15) = STARTING LOCATION OF VECTOR CONTAINING THE ACTUAL
2127 C NUMBER OF REQUESTED OUTPUT LOCATION IN EACH OUTPUT SET
2128 C A(N16) = STARTING LOCATION OF ELEMENT STIFFNESS MATRIX
2129 C
2130 50 NG = N5 + NUMNP
2131 N7 = N6 + NPAR(3)
2132 N8 = N7 + NPAR(3)
2133 N9 = N8 + NPAR(3)
2134 N10 = N9 + NPAR(3) * NPAR(4) * 13
2135 N11 = N10 + NPAR(5) * 9
2136 N12 = N11 + NPAR(6)
2137 N13 = N12 + NPAR(6)
2138 N14 = N13 + NPAR(6) * 7
2139 N15 = N14 + NPAR(8) * 7
2140 N16 = N15 + NPAR(8)
2141 N17 = N16 + NPAR(7) * 189
2142 C
2143 IF(N17.GT.MTOT) CALL ERROR(N17-MTOT)
2144 C
2145 C PROCESS ELEMENT DATA, AND GENERATE ELEMENT MATRICES
2146 C
2147 CALL THDFE (A(N1),A(N2),A(N3),A(N4),A(N5),A(N6),A(N7),A(N8),A(N9),
2148 1 A(N10),A(N11),A(N12),A(N13),A(N14),A(N15),A(N16),
2149 2 NPAR(2),NPAR(3),NPAR(4),NPAR(5),NPAR(6),NPAR(7),
2150 3 NPAR(8),NPAR(9),NPAR(10),NUMNP)
2151 C
2152 RETURN
2153 C
2154 C RECOVER ELEMENT STRESSES (STATIC CASES ONLY)
2155 C
2156 500 WRITE (34,2001)
2157 NUME = NPAR(2)
2158 C
2159 read (5,*), all, new
2160 501 format(3i5)

```

```

2161      if(n11.le.0) n11=1
2162      if(nuu.le.0) nuu=nuue
2163      DO 800 mm=1,NUME
2164 C
2165 C
2166 C*** STRESS PORTHOLE
2167      CALL STRSC (A(N1),A(N3),NEQ,0)
2168      IF(N10SV.EQ.1)
2169      *WRITE (NT10) NS
2170 C***
2171 C
2172      IF(NS.EQ.1) GO TO 800
2173 C
2174 C-   WRITE (34,5000)
2175 C
2176      DO 700 L=LT,LH
2177 C
2178 C
2179      CALL STRSC (A(N1),A(N3),NEQ,1)
2180      LOC = NS/6
2181      K1 = -5
2182 C
2183      DO 600 M=1,LOC
2184      K1 = K1 + 6
2185      K2 = K1 + 5
2186 C
2187      if(mm.ge.n11.and.mm.le.nuu) then
2188      IF(N.EQ.1) WRITE (34,3001) mm,L,N,(SIG(I),I=K1,K2)
2189      IF(N.GT.1) WRITE (34,4001) N,(SIG(I),I=K1,K2)
2190      end if
2191 C
2192 C*** STRESS PORTHOLE
2193      IF(N10SV.EQ.1)
2194      *WRITE (NT10) mm,L,N,(SIG(I),I=K1,K2)
2195 C***
2196      600 CONTINUE
2197 C
2198 C-   WRITE (34,5000)
2199 C
2200      700 CONTINUE
2201      800 CONTINUE
2202      RETURN
2203 C
2204 C   FORMATS
2205 C
2206 1000 FORMAT (53H121 - N O D E S O L I D E L E M E N T I N P U T
2207      1 10HD A T A      //38HCNTR0L INFORMATION      /1X)
2208 1001 FORMAT (48HOEKKR DETECTED WHILE PROCESSING MASTER ELEMENT ,
2209      1 12HCNTR0L CARD, //16X,1H(,10IS,1H),/1X)
2210 1002 FORMAT (32H NO 3-D SOLID ELEMENTS SPECIFIED,/1X)
2211 1003 FORMAT (23H NO MATERIALS REQUESTED, / 1X)
2212 1004 FORMAT (49H MAXIMUM NUMBER OF NODES MUST BE GE.8 .AND. LE.21, 1X)
2213 1005 FORMAT (42H INTEGRATION ORDER MUST BE GE.2 .AND. LE.4,/1X)
2214 2001 FORMAT (54H121 - N O D E S O L I D E L E M E N T S T R E S S

```



```

2215      * //
2216      *23H ELEMENT LOAD LOCATION,9X,6HSIG-XX,9X,6HSIG-YY,9X,6HSIG-ZZ,
2217      3 9X,6HSIG-XY,9X,6HSIG-YZ,9X,6HSIG-ZX, //1X)
2218      3001 FORMAT (I8,I6,I9,6E15.6)
2219      4001 FORMAT ( 14X,I9,6E15.6)
2220      5000 FORMAT ( / )
2221 C
2222      END
2223 C=====
2224      SUBROUTINE SSLAW (D,E,TEMP,DCA,KAXES,KMAT,NEL,DUM,ALPHA)
2225 C
2226 C      CALLED BY : THDFE
2227 C
2228      IMPLICIT REAL*8(A-H,O-Z)
2229 C
2230 C      THIS ROUTINE FORMS THE STRESS-STRAIN LAW IN MATERIAL COORDINATES
2231 C      (X1,X2,X3) AND TRANSFORMS THE MATERIAL SYSTEM LAW TO GLOBAL
2232 C      COORDINATES (X,Y,Z).
2233 C
2234      DIMENSION D(6,6),E(12),TEMP(6,6),DCA(3,3),IPRM(3),DUM(6,6),
2235      1          ALPHA(6)
2236 C
2237      DATA IPRM / 2,3,1 /
2238 C
2239 C      FORM THE DIRECT STRAIN PARTITION OF THE STRAIN-STRESS LAW IN
2240 C      MATERIAL COORDINATES (X1,X2,X3)
2241 C
2242      DO 20 I=1,3
2243      ALPHA(I) = E(I+9)
2244      ALPHA(I+3) = 0.0
2245      IF(E(I).GT.1.0E-08) GO TO 15
2246      WRITE (33,3000) I,I,KMAT,NEL
2247      STOP
2248      3000 FORMAT (23HOERROR*** MODULUS E(,2I1,16H) FOR MATERIAL (,I3,
2249      1          14H) IN ELEMENT (,I5,10H) IS ZERO., / 1X)
2250      15 TEMP(I,I) = 1.0/E(I)
2251      20 CONTINUE
2252 C
2253      TEMP(1,2) = -E(4)* TEMP(2,2)
2254      TEMP(2,1) =          TEMP(1,2)
2255      TEMP(1,3) = -E(5)* TEMP(3,3)
2256      TEMP(3,1) =          TEMP(1,3)
2257      TEMP(2,3) = -E(6)* TEMP(3,3)
2258      TEMP(3,2) =          TEMP(2,3)
2259 C
2260 C      INVERT THE DIRECT STRAIN PARTITION
2261 C
2262      DO 60 N=1,3
2263      X = 1.0/TEMP(N,N)
2264      DO 30 J=1,3
2265      30 TEMP(N,J) = - TEMP(N,J)* X
2266 C
2267      DO 50 I=1,3
2268      IF(N.EQ.I) GO TO 50

```

```

2269      DO 40 J=1,3
2270      IF(N.EQ.J) GO TO 40
2271      TEMP(I,J) = TEMP(I,J) + TEMP(I,N) * TEMP(N,J)
2272  40  CONTINUE
2273  50  TEMP(I,N) = TEMP(I,N) * X
2274 C
2275      TEMP(N,N) = X
2276  60  CONTINUE
2277 C
2278 C      FORM THE COMPLETE STRESS-STRAIN LAW IN MATERIAL COORDINATES
2279 C
2280      DO 70 I=1,6
2281      DO 70 J=1,6
2282  70  D(I,J) = 0.0
2283 C
2284      DO 80 I=1,3
2285      DO 80 J=1,3
2286  80  D(I,J) = TEMP(I,J)
2287 C
2288      D(4,4) = E(7)
2289      D(5,5) = E(8)
2290      D(6,6) = E(8)
2291 C
2292 C      TRANSFORM THE MATERIAL LAW TO GLOBAL COORDINATES (X,Y,Z)
2293 C
2294      IF(KAXES.LT.1) RETURN
2295 C
2296 C      TRANSFORMATION BETWEEN MATERIAL STRAINS AND GLOBAL STRAINS
2297 C
2298      DO 100 I1=1,3
2299      I2 = IPRM(I1)
2300      DO 90 J1 = 1,3
2301      J2 = IPRM(J1)
2302      TEMP(I1 ,J1 ) = DCA(J1,I1)*DCA(J1,I1)
2303      TEMP(I1+3,J1 ) = DCA(J1,I1)*DCA(J1,I2) * 2.0
2304      TEMP(I1 ,J1+3) = DCA(J1,I1)*DCA(J2,I1)
2305      TEMP(I1+3,J1+3) = DCA(J1,I1)*DCA(J2,I2) +
2306  1      DCA(J2,I1)*DCA(J1,I2)
2307  90  CONTINUE
2308 100  CONTINUE
2309 C
2310 C      ROTATE THE MATERIAL LAW TO THE GLOBAL SYSTEM
2311 C
2312      DO 130 I=1,6
2313      DO 120 J=1,6
2314      X = 0.0
2315      DO 110 K=1,6
2316 110  X = X + D(I,K)*TEMP(K,J)
2317 120  DUM(I,J) = X
2318 130  CONTINUE
2319 C
2320      DO 160 I=1,6
2321      DO 150 J=1,6
2322      X = 0.0

```

```

2323      DO 140 K=1,6
2324 140 X = X + TEMP(K,I)*DUM(K,J)
2325      D(I,J) = X
2326      D(J,I) = X
2327 150 CONTINUE
2328 160 CONTINUE
2329 C
2330 C      TRANSFORM THE EXPANSION COEFFICIENTS FROM MATERIAL COORDINATES
2331 C      TO GLOBAL COORDINATES
2332 C
2333 C
2334      DO 200 I=1,6
2335      X = 0.0
2336      DO 190 K=1,3
2337 190 X = X + TEMP(K,I)*E(K+9)
2338      IF(I.GT.3) X =X*2.0
2339 200 ALPHA(I) = X
2340 C
2341      RETURN
2342      END
2343 C=====
2344      SUBROUTINE STRESS(STR,B,D,NEQB,LB,LL,NEQ,NBLOCK)
2345      IMPLICIT REAL*8(A-H,O-Z)
2346 C
2347 C      CALLS: ELTYPE
2348 C      CALLED BY: SOLEQ
2349 C
2350      DIMENSION D(NEQ,LB),B(NEQB,LL),STR(4,LL)
2351      COMMON /ELPAR/ NPAR(14),NUMNP,MBAND,NELTYP,N1,N2,N3,N4,N5,MTOT,MEQ
2352      COMMON /JUNK/ LT,LH,IFILL(428)
2353      COMMON /EXTRA/ MODEX,NT8,N10SV,NT10,IFILL2(12)
2354 C
2355      READ (8) STR
2356      NT=(LL-1)/LB +1
2357      LH=0
2358 C*** STRESS PORTHOLE
2359      IF(N10SV.EQ.1)
2360      *WRITE (NT10) NELTYP,NT
2361 C
2362      DO 1000 I1=1,NT
2363 C
2364      LT =LH+1
2365      LLT=1-LT
2366      LH=LT+LB-1
2367      IF(LH.GT.LL) LH=LL
2368 C
2369 C      MOVE DISPLACEMENTS INTO CORE FOR LB LOAD CONDITIONS
2370 C
2371      REWIND 2
2372 C*** STRESS PORTHOLE
2373      IF(N10SV.EQ.1)
2374      *WRITE (NT10) LT,LH
2375      NQ=NEQB*NBLOCK
2376      DO 200 NN=1,NBLOCK

```

```

2377      READ (2) B
2378      N=NEQB
2379      IF (NN.EQ.1) N=NEQ-NQ+NEQB
2380      NQ=NQ-NEQB
2381      DO 200 J=1,N
2382      I=NQ+J
2383      DO 200 L=LT,LH
2384      K=L+LLT
2385      200 D(I,K)=B(I,L)
2386      LK=LH-LT+1
2387      C
2388      C      CALCULATE STRESSES FOR ALL ELEMENTS FOR LB LOAD CONDITIONS
2389      C
2390      REWIND 1
2391      DO 1000 M=1,NELTIP
2392      READ (1) NPAR
2393      C*** STRESS PORTHOLE
2394      IF(NIOSV.EQ.1)
2395      *WRITE (NT10) NPAR
2396      MTYPE=NPAR(1)
2397      NPAR(1)=0
2398      CALL ELTYPE(MTYPE)
2399      1000 CONTINUE
2400      C
2401      RETURN
2402      END
2403      C=====
2404      SUBROUTINE STRSC(STR,D,NEQ,NTAG)
2405      IMPLICIT REAL*8(A-H,O-Z)
2406      C
2407      C      CALLED BY: TRUSS,BEAM,PLANE,THREED,SHELL,BOUND,PIPE
2408      C
2409      DIMENSION STR(4,1),D(NEQ,1)
2410      COMMON /JUNK/ LT,LH,L,IPAD,SG(20),SIG(7),EXTRA(186)
2411      COMMON /EM/ NS,ND,B(42,63),TI(42,4),LM(63)
2412      C
2413      IF (NTAG.EQ.0) GO TO 800
2414      LL=L-LT+1
2415      DO 300 I=1,NS
2416      SG(I)=0.0
2417      DO 300 J=1,4
2418      300 SG(I)=SG(I)+TI(I,J)*STR(I,L)
2419      DO 500 J=1,ND
2420      JJ=LM(J)
2421      IF(JJ.EQ.0) GO TO 500
2422      DO 400 I=1,NS
2423      400 SG(I)=SG(I)+B(I,J)*D(JJ,LL)
2424      C
2425      500 CONTINUE
2426      GO TO 900
2427      800 READ (1) ND,NS,(LM(I),I=1,ND),(( B(I,J),I=1,NS),J=1,ND),
2428      1 (( TI(I,J),I=1,NS),J=1,4)
2429      900 RETURN
2430      END

```

```

2431 C=====
2432 SUBROUTINE ST8R31 (E,B,S,XX,NOD9,H,P,SIGDT,DELT,FT,DL,XM,NEL,ND,
2433 1 IELD,IELX,KTL,KGL,KMS,NINT,NINTZ,WIDEN,MSDEN)
2434 C
2435 C CALLED BY : THREE
2436 C CALLS : DER3DS
2437 C
2438 IMPLICIT REAL*8(A-H,O-Z)
2439 C
2440 C
2441 C . . . . .
2442 C .
2443 C .
2444 C . HEXAHEDRAL CURVILINEAR THREE-DIMENSIONAL ELEMENTS .
2445 C .
2446 C . ISOPARAMETRIC OR SUBPARAMETRIC .
2447 C .
2448 C .
2449 C . . . . .
2450 C
2451 C
2452 C
2453 DIMENSION E(6,1),B(6,1),S(63,1),XX(3,1),NOD9(1),H(1),P(3,1)
2454 1 SIGDT(1),DELT(1),FT(1),DL(1),XM(1),D(9),SDT(6),EVL(63),
2455 2 W(3,3),IPERM(3,3),KDX(3),LDX(3)
2456 C
2457 COMMON /GAUSS/ XG(4,4),WGT(4,4)
2458 C REAL MSDEN
2459 REAL*8 MSDEN
2460 C
2461 DATA IPERM / 1,4,6, 4,2,5, 6,5,3 /
2462 C
2463 VOL = 0.0
2464 C
2465 C DETERMINE IF THE MATERIAL IS ORTHOTROPIC (ISO.EQ.1, ISOTROPIC)
2466 C
2467 DUM = 0.0
2468 DO 20 I=4,6
2469 J = I-1
2470 DO 20 K=1,J
2471 20 DUM = DUM +DABS(E(K,I))
2472 ISO = 1
2473 IF(DUM.GT.1.0E-6) ISO = 0
2474 IF(ISO.EQ.0) GO TO 24
2475 DO 22 I=2,3
2476 DUM = DUM +DABS(E(I ,1 ) -E(I-1,I-1))
2477 22 DUM = DUM +DABS(E(I+3,I+3) -E(I+2,I+2))
2478 DUM = DUM +DABS(E(1 ,2 ) - E(2 ,3 ))
2479 DUM = DUM +DABS(E(2 ,3 ) - E(3 ,1 ))
2480 IF ( DUM.GT.1.0E-6 ) ISO=0
2481 24 CONTINUE
2482 C
2483 C
2484 C VOLUME INTEGRATION LOOP

```

```

2485 C
2486 C
2487     DO 10 LX=1,NINT
2488     DO 10 LY=1,NINT
2489     E1=XG(LX,NINT)
2490     E2=XG(LY,NINT)
2491     DO 10 LZ=1,NINTZ
2492     E3=XG(LZ,NINTZ)
2493 C
2494     WT=WGT(LX,NINT)*WGT(LY,NINT)*WGT(LZ,NINTZ)
2495 C
2496 C     EVALUATE STRAIN-DISPLACEMENT MATRIX B AND JACOBIAN DETERMINANT
2497 C
2498     CALL SUBRDS (REL,XX,B,DET,E1,E2,E3,NOD9,H,P,IELD,IELX)
2499 C
2500     ADD CONTRIBUTION TO ELEMENT STIFFNESS
2501 C
2502     FACT = WT*DET
2503     FACT2 =DSQRT(FACT)
2504 C
2505     DO 25 I=1,IELD
2506     K3 = 3*I
2507     K2 = K3-1
2508     K1 = K2-1
2509     BV(K1) = B(1,K1)*FACT2
2510     BV(K2) = B(2,K2)*FACT2
2511     BV(K3) = B(3,K3)*FACT2
2512     25 CONTINUE
2513 C
2514     DO 30 I=1,ND
2515     DO 30 J=1,ND
2516     30 S(I,J) = S(I,J) + BV(I)*BV(J)
2517 C
2518     ACCUMULATE ELEMENT VOLUME
2519 C
2520     VOL = VOL + FACT
2521 C
2522     COMPUTE GRAVITY LOADS
2523 C
2524     IF(KGL.EQ.0) GO TO 150
2525     DO 130 K=1,IELD
2526     130 BL(K) = BL(K) + HCK)*FACT)* WTDEN
2527 C
2528     COMPUTE THERMAL LOADING NODE FORCE VECTOR
2529 C
2530     150 IF(KTL.EQ.0) GO TO 190
2531 C
2532     1. ELEMENT TEMPERATURE DIFFERENCE AT THIS INTEGRATION POINT
2533     (R,S,T)
2534 C
2535     DT = 0.0
2536     DO 160 R=1,IELD
2537     160 DT = DT + HCR)* (BET K)
2538     DT = DT*FACT

```

```

2539 C
2540 C      2. INITIAL STRESSES AT (R,S,T)
2541 C
2542      DO 170 K=1,6
2543 170 SDT(K) = SIGDT(K)*DT
2544 C
2545 C      3. NODE FORCES
2546 C
2547      DO 180 K=1,ND
2548      DO 175 I=1,6
2549 175 FT(K) = ET(K) + B(I,K)*SDT(I)
2550 180 CONTINUE
2551 C
2552 C      WRITE(28,*) ' DT, DELT', DT, (DELT(K), K=1, IELD)
2553 C      WRITE(28,*) ' SIGDT ', (SIGDT(K), K=1, 6)
2554 C      WRITE(28,*) ' FT -----'
2555 C      WRITE(28,*) ' (ET(K), K=1, 6)
2556 190 CONTINUE
2557 10 CONTINUE
2558 C
2559      DO 35 I=1,2
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      PPLICATIONS OF THE INTEGRATED B(I)*B(J) ARRAY.
2567 C
2568 C      1. TEST FOR MATERIAL TYPE
2569 C
2570      IF(ISO.EQ.0) GO TO 75
2571 C
2572 C      A. ISOTROPIC MATERIAL
2573 C
2574      D1 = E(1,1)
2575      D2 = E(1,2)
2576      D3 = E(4,4)
2577 C
2578      DO 60 I=1, IELD
2579      K3 = 3*I
2580      K2 = K3-1
2581      K1 = K2-1
2582      K0 = K1-1
2583      DO 60 J=1, IELD
2584      L3 = 3*J
2585      L2 = L3-1
2586      L1 = L2-1
2587      L0 = L1-1
2588 C
2589      IC = 0
2590      DO 40 II=1, 3
2591      M = II+ IC
2592      DO 40 JJ=1, 3

```

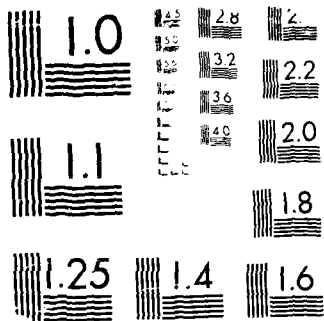
```

2593      N = JJ+ L0
2594      IC = IC+ 1
2595      D(IC) = S(m,N)
2596      40 CONTINUE
2597 C
2598      S(K1,L1) = D(1)* D1 + (D(5) + D(9))* D3
2599      S(K2,L2) = D(5)* D1 + (D(1) + D(9))* D3
2600      S(K3,L3) = D(9)* D1 + (D(5) + D(1))* D3
2601      S(K1,L2) = D(2)* D2 + D(4)* D3
2602      S(K2,L1) = D(4)* D2 + D(2)* D3
2603      S(K2,L3) = D(6)* D2 + D(8)* D3
2604      S(K3,L2) = D(8)* D2 + D(6)* D3
2605      S(K3,L1) = D(7)* D2 + D(3)* D3
2606      S(K1,L3) = D(3)* D2 + D(7)* D3
2607 C
2608      60 CONTINUE
2609 C
2610      GO TO 110
2611 C
2612 C          B. ANISOTROPIC MATERIAL
2613 C
2614      75 DO 100 I=1, IELD
2615          KO = 3*I-3
2616          DO 100 J=1, IELD
2617              LO = 3*J-3
2618 C
2619              DO 80 II=1, 3
2620                  M = II+KO
2621                  DO 80 JJ=1, 3
2622                      N = JJ+L0
2623                      W(II,JJ) = S(M,N)
2624              80 CONTINUE
2625 C
2626              DO 100 K=1, 3
2627                  II = KO+K
2628                  DO 82 IJ=1, 3
2629              82 KDX(IJ)=IPERM(II,K)
2630                  DO 95 L=1, 3
2631                  I2 = LO+L
2632                  DO 83 IJ=1, 3
2633              83 LDX(IJ)=IPERM(IJ,L)
2634 C
2635                  SUM=0.0
2636 C
2637                  DO 90 II=1, 3
2638                      K1 = KDX(II)
2639                      DO 85 JJ=1, 3
2640                      K2 = LDX(JJ)
2641 C
2642                      85 SUM = SUM + W(II, JJ)*E(K1,K2)
2643                  90 CONTINUE
2644 C
2645                  S(II, I2) = SUM
2646 C

```







MICROCOPY RESOLUTION TEST CHART  
1963-A

```

2647     95 CONTINUE
2648    100 CONTINUE
2649    110 CONTINUE
2650 C
2651 C
2652 C     REFLECT FOR SYMMETRY
2653 C
2654     DO 200 I=1,ND
2655     DO 200 J=I,ND
2656     200 S(J,I) = S(I,J)
2657 C
2658 C     CONSTRUCT THE LUMPED MASS MATRIX
2659 C
2660     IF(KMS.EQ.0) RETURN
2661 C
2662     FACT = VOLX * MSDEN / IELD
2663     DO 220 K=1,ND
2664     220 XM(K) = FACT
2665 C
2666 C
2667     RETURN
2668     END
2669 C3=====
2670     SUBROUTINE THDFE (ID,X,Y,Z,T,DEN,RHO,NTP,EE,
2671     1     DCA,NEFACE,LT,PWA,LOC,MAXPTS,SS,
2672     2     NUME,NUMMAT,MAXTP,NORTH0,NDLS,MAXNOD,
2673     3     NOPSET,INTRS,INTT,NUMNP)
2674 C
2675 C     CALLED BY : SOL21
2676 C     CALLS : INF21,CALBAN,SSLAW,DER3DS,STR21,FACEPR
2677 C
2678     IMPLICIT REAL*8(A-H,O-Z)
2679 C
2680 C     ROUTINE FOR THE STIFFNESS, MASS AND STRESS MATRIX GENERATION
2681 C     FOR THE 8-TO-21 NODE ISO-(OR SUB)-PARAMETRIC ORTHOTROPIC
2682 C     HEXAHEDRON.
2683 C
2684     COMMON /JUNK/ XLF(4),YLF(4),ZLF(4),TLF(4),PLF(4),FILL1(22),V2(3),
2685     1     FILL2(12),LS(4),KLS(4),NOD(21),NOD9M(13),KOD(21),
2686     2     NREAD,TAG,E(12)
2687     COMMON /ELPAK/ IFILL3(15),MBAND
2688     COMMON /EM/ SDT(42,63),SF(42,4),NS,ND,LM(63)
2689     DIMENSION RE(63,4),XM(63),D(6,6),TEMP(6,6),DUM(6,6),
2690     *     ALPHA(6),XX(3,21),B(6,63),H(21),P(3,21),SIGDT(6),
2691     *     DELT(21),FT(63),BL(21),PL(63),LOCOP(7),VIS(6)
2692 C
2693     COMMON /GAUSS/ XG(4,4),WGT(4,4),STPTS(27,3)
2694     COMMON /DYN / IFILL4(11),NDYN
2695     COMMON /EXTRA/ MODEX,NT8
2696 C
2697     DIMENSION ID(NUMNP,1),X(1),Y(1),Z(1),T(1),DEN(1),RHO(1),
2698     1     NTP(1),EE(MAXTP,13,1),DCA(3,3,1),NEFACE(1),LT(1),
2699     2     PWA(7,1),LOC(7,1),MAXPTS(1),SS(63,1)
2700 C

```



2755	STPTS(26,3)=0.
2756	STPTS(27,3)=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) = -.5773502691896D0
2789	XG(2,2) = .5773502691896D0
2790	XG(3,2) = 0.
2791	XG(4,2) = 0.
2792	XG(1,3) = -.7745966692415D0
2793	XG(2,3) = 0.
2794	XG(3,3) = .7745966692415D0
2795	XG(4,3) = 0.
2796	XG(1,4) = -.8611363115941D0
2797	XG(2,4) = -.3399810435849D0
2798	XG(3,4) = .3399810435849D0
2799	XG(4,4) = .8611363115941D0
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

```

2809      WGT(2,3) = .38888888888889   D0
2810      WGT(3,3) = .59999999999996   D0
2811      WGT(4,3) = 0.0
2812      WGT(1,4) = .3478548451375   D0
2813      WGT(2,4) = .6521451548625   D0
2814      WGT(3,4) = .6521451548625   D0
2815      WGT(4,4) = .3478548451375   D0
2816 C
2817      NTBSV = MDEX
2818      DO 10 I=4,6
2819      DO 10 J=1,6
2820 10  B(I,J) = 0.0
2821      DO 14 I=1,42
2822      DO 14 J=1,4
2823 14  SF(I,J)=0.0
2824 C
2825 C      PRINT ELEMENT CONTROL VARIABLES
2826 C
2827      WRITE (33,3001) NOME,NUMMAT,MAXTP,NORTHO,NDLS,MAXNOD,NOPSET,INTRS,
2828 1      INTT
2829 C
2830 C      READ AND CHECK INPUT UP TO THE ELEMENT DATA CARDS
2831 C
2832      CALL INP21      (NUMMAT,MAXTP,NORTHO,NDLS,NOPSET,NTBSV,NUMNP,X,
2833 1      Y,Z,BEN,RHO,NTP,EE,DCA,NFACE,LT,PWA,LOC,MAXPTS)
2834 C
2835 C      READ ELEMENT DATA CARDS
2836 C
2837      NREAD = 8
2838      IF(MAXNOD.GT.8) NREAD = 21
2839 C
2840      WRITE (33,3014) (I,I=1,8)
2841      IF(MAXNOD.GT.8)
2842 *WRITE (33,3016) (I,I=9,21)
2843 C
2844      NEL = 0
2845 C
2846 C      CARD FOR ELEMENT NUMBER ONE ONLY
2847 C
2848      READ (5,1008) INEL,NDIS,NXYZ,NMAT,MAXES,IOP,TZ,KG,NRSINT,RTINT
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 (5,1008) INEL,NDIS,NXYZ,NMAT,MAXES,IOP,TZ,KG,NRSINT,RTINT
2860 1,IREUSE,(LS(I),I=1,4)
2861      READ (5,1009) (NOD(I),I=1,NREAD)
2862 C

```

```

2863 C      DATA ADMISSIBILITY CHECK
2864 C
2865      51 IF(NDIS.EQ.0) NDIS = MAXNOD
2866          IF(NDIS.LE.MAXNOD) GO TO 5051
2867          WRITE (33,3015) INEL,NDIS,NXYZ,NMAT,MAXES,IOP,TZ,KG,NRSINT,NTINT
2868      1, IREUSE,(LS(I),I=1,4)
2869          WRITE (33,4015) NDIS,MAXNOD
2870          STOP
2871      5051 IF(NDIS.GE.8) GO TO 52
2872          WRITE (33,4023) NDIS
2873          STOP
2874      52 IF(NXYZ.EQ.0) NXYZ = NDIS
2875          IF(NXYZ.LE.NDIS) GO TO 5052
2876          WRITE (33,4016) NXYZ,NDIS
2877          WRITE (33,4099)
2878          MODEX = 1
2879          GO TO 53
2880      5052 IF(NXYZ.GE.8) GO TO 53
2881          WRITE (33,4024) NXYZ
2882          WRITE (33,4099)
2883          MODEX = 1
2884      53 IF(NMAT.GE.1 .AND. NMAT.LE.NUMMAT) GO TO 54
2885          WRITE (33,3015) INEL,NDIS,NXYZ,NMAT,MAXES,IOP,TZ,KG,NRSINT,NTINT
2886      1, IREUSE,(LS(I),I=1,4)
2887          WRITE (33,4017)
2888          WRITE (33,4099)
2889          MODEX = 1
2890      54 IF(MAXES.LE.NORTHO) GO TO 55
2891          WRITE (33,3015) INEL,NDIS,NXYZ,NMAT,MAXES,IOP,TZ,KG,NRSINT,NTINT
2892      1, IREUSE,(LS(I),I=1,4)
2893          WRITE (33,4018)
2894          WRITE (33,4099)
2895          MODEX = 1
2896      55 IF(IOP.GE.0 .AND. IOP.LE.NOPSET) GO TO 56
2897          WRITE (33,3015) INEL,NDIS,NXYZ,NMAT,MAXES,IOP,TZ,KG,NRSINT,NTINT
2898      1, IREUSE,(LS(I),I=1,4)
2899          WRITE (33,4019)
2900          WRITE (33,4099)
2901          MODEX = 1
2902      56 DO 57 I=1,4
2903          IF(LS(I).GE.0 .AND. LS(I).LE.NDLS) GO TO 57
2904          WRITE (33,3015) INEL,NDIS,NXYZ,NMAT,MAXES,IOP,TZ,KG,NRSINT,NTINT
2905      1, IREUSE,(LS(J),J=1,4)
2906          WRITE (33,4020) LS(I)
2907          WRITE (33,4099)
2908          MODEX = 1
2909      57 CONTINUE
2910 C
2911 C      DEFAULT VALUES IF REQUIRED
2912 C
2913      IF(KG.EQ.0) KG = 1
2914      IF(NRSINT.EQ.0) NRSINT = INTRS
2915      IF(NTINT.EQ.0) NTINT = INTT
2916 C

```

```

2917      DO 58 I=1,8
2918      IF(NOD(I).GE.1 .AND. NOD(I).LE.NUMNP) GO TO 58
2919      WRITE (33,3015) INEL,NDIS,NXYZ,NMAT,MAXES,IOP,TZ,KG,NRSINT,NTINT
2920      1,IREUSE,(LS(J),J=1,4)
2921      WRITE (33,4021) I,NOD(I)
2922      STOP
2923 58 CONTINUE
2924      IF(MAXNOD.LT.9) GO TO 60
2925      II = 0
2926      DO 59 I=9,21
2927      IF(NOD(I).EQ.0) GO TO 59
2928      II = II + 1
2929      NOD9M(II) = I
2930      IF(NOD(I).LE.NUMNP) GO TO 59
2931      WRITE (33,3015) INEL,NDIS,NXYZ,NMAT,MAXES,IOP,TZ,KG,NRSINT,NTINT
2932      1,IREUSE,(LS(J),J=1,4)
2933      WRITE (33,4021) I,NOD(I)
2934      STOP
2935 59 CONTINUE
2936 C
2937      I = II + 8
2938      IF(I.EQ.NDIS) GO TO 60
2939      WRITE (33,4025) I,NDIS
2940      STOP
2941 C
2942 60 NEL = NEL + 1
2943      ML = INEL - NEL
2944      IF(ML) 65,70,80
2945 65 WRITE (33,4022) INEL
2946      STOP
2947 C
2948 C      SAVE THE DATA FOR ELEMENT NUMBER *INEL* FOR POSSIBLE USE IN
2949 C      DATA GENERATION
2950 C
2951 C
2952 70 KDIS = NDIS
2953      KXYZ = NXYZ
2954      KMAT = NMAT
2955      KAXES = MAXES
2956      KIOP = IOP
2957      KTZ = TZ
2958      KKG = KG
2959      KRSINT = NRSINT
2960      KNTINT = NTINT
2961      KREUSE = IREUSE
2962      DO 72 I=1,4
2963 72 KLS(I) = LS(I)
2964      DO 74 I=1,NREAD
2965 74 KOD(I) = NOD(I)
2966      TAG = TG1
2967 C
2968      GO TO 90
2969 C
2970 C      INCREMENT THE NON-ZERO NODE NUMBERS FROM THE PRECEDING ELEMENT

```



```

2863 C      DATA ADMISSIBILITY CHECK
2864 C
2865      51 IF(NDIS.EQ.0) NDIS = MAXNOD

```

```

2971 C
2972      80 DO 85 I=1,NKREAD
2973          IF(KOD(I).LT.1) GO TO 85
2974          KOD(I) = KOD(I) + KR6
2975      85 CONTINUE
2976          TAG = T62
2977 C
2978      90 ND = 3 * NDIS
2979 C
2980 C      COMPUTE THE AVERAGE ELEMENT TEMPERATURE USING COORDINATE NODES
2981 C
2982          TAV = 0.0
2983          DO 95 K=1,KXYZ
2984              I = KOD(K)
2985          95 TAV = TAV + T(I)
2986          TAV = TAV / KXYZ
2987 C
2988 C      PERFORM TEMPERATURE INTERPOLATION FOR THE PROPERTY SET
2989 C
2990          NT = NIP(KMAT)
2991          IF(NT.GT.1) GO TO 100
2992      97 DO 98 I=1,12
2993          98 E(I) = EE(1,I+1,KMAT)
2994          GO TO 112
2995      100 IF(TAV.GE.EE(1,1,KMAT)) GO TO 104
2996      102 WRITE (33,4030) TAV,NEL,KMAT
2997          STOP
2998      104 IF(TAV.GT.EE(NT,1,KMAT)) GO TO 103
2999          IF(TAV.EQ.EE(1,1,KMAT)) GO TO 97
3000 C
3001          IF(MODEX.EQ.1) GO TO 112
3002 C
3003          DO 106 K=C,NT
3004              K2 = K
3005              K1 = K-1
3006          IF(TAV.GT.EE(K1,1,KMAT) .AND. TAV.LE.EE(K2,1,KMAT)) GO TO 108
3007      106 CONTINUE
3008      108 DT = EE(K2,1,KMAT) - EE(K1,1,KMAT)
3009          RATIO = (TAV - EE(K1,1,KMAT)) / DT
3010          DO 110 I=1,12
3011          110 E(I) = EE(K1,I+1,KMAT) + RATIO *(EE(K2,I+1,KMAT)-EE(K1,I+1,KMAT))
3012 C
3013      112 CONTINUE
3014 C
3015 C      FORM THE STRESS-STRAIN LAW IN MATERIAL COORDINATES AND TRANSFORM
3016 C      TO GLOBAL (X,Y,Z) COORDINATES
3017 C
3018          IF(MODEX.EQ.0)
3019          *CALL SSLAW (D,E,TEMP,DCA(1,1,KAXES),KAXES,KMAT,NEL,DUM,ALPHA)
3020 C
3021 C      STORE THE NODE COORDINATES FOR THIS ELEMENT
3022 C
3023          IF(MODEX.EQ.1) GO TO 410
3024 C

```

```

3025      DO 130 I=1,KDIS
3026      II = KOD(I)
3027      IF(I.LT.9) GO TO 125
3028      JJ = NOD9M(I-8)
3029      II = KOD(JJ)
3030 125  XX(1,I) = X(II)
3031      XX(2,I) = Y(II)
3032      XX(3,I) = Z(II)
3033 130  CONTINUE
3034 C
3035 C      COMPUTE THE ELEMENT STIFENESS, MASS, THERMAL AND GRAVITY LOAD
3036 C      MATRICES
3037 C
3038      DO 170 I=1,63
3039      DO 170 J=1,4
3040 170  RF(I,J)=0.0
3041 C
3042      IF(KREUSE.EQ.1) GO TO 300
3043 C
3044      DO 180 I=1,KDIS
3045 180  DL(I)=0.0
3046      DO 190 I=1,ND
3047 C
3048 C
3049 C      1. THERMAL LOADS
3050 C
3051 190  FT(I)=0.0
3052      KTL = 0
3053      DUX = 0.0
3054      DO 200 I=1,4
3055 200  DUX = DUX +DABS(TLF(I))
3056      IF(DUX.GT.1.0E-06) KTL = 1
3057      IF(KTL.EQ.1) THEN      !!!!!
3058      WRITE(99,*) '##### ktl==1'      !!!!!
3059      END IF      !!!!!
3060      IF (NDYN.GT.0) KTL=0
3061      IF(KTL.EQ.0 .OR. NDYN.GT.0) GO TO 235
3062 C
3063 C      A. INITIAL STRESS CONSTANTS
3064 C
3065      DO 210 I=1,6
3066      SIGDT(I) = 0.0
3067      DO 205 K=1,6
3068 205  SIGDT(I) = SIGDT(I) + D(I,K)* ALPHA(K) ! 1 changed to I (first)
3069 210  CONTINUE
3070 C      WRITE(28,*) ' sigdt in THDFE'
3071 C      WRITE(28,*) (sigdt(k),k=1,6)
3072 C
3073 C      B. VECTOR OF NODE TEMPERATURE DIFFERENCES
3074 C
3075      DO 230 I=1,KDIS
3076      II = KOD(I)
3077      IF(I.LT.9) GO TO 220
3078      J = NOD9M(I-8)

```

```

3079      II = KOD(J)
3080      220 DELT(I) = T(II) - TTZ
3081      230 CONTINUE
3082 C
3083 C          C. CLEAR THE THERMAL LOAD NODE FORCE VECTOR
3084 C
3085 C          2. GRAVITY LOADS
3086 C
3087      235 DUX=0.0
3088          DO 250 I=1,4
3089      250 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
3093 C
3094 C
3095 C          3. MASS MATRIX
3096          KMS = 0
3097          IF(NDYN.GT.0) KMS = 1
3098 C
3099          DO 270 K=1,ND
3100 C
3101 C          4. STIFFNESS MATRIX
3102 C
3103      270 XM(K) = 0.0
3104          DO 280 I=1,ND
3105          DO 280 K=1,ND
3106      280 SS(I,K) = 0.0
3107 C
3108 C
3109          CALL ST8K21 (D,B,SS,XX,NOD9M,H,P,SIGDT,DELT,FT,DL,XM,NEL,ND,KDIS,
3110      1          KXYZ,KTL,KGL,KMS,KRSINT,KTINT,DEN(KMAT),RHO(KMAT))
3111 C
3112 C
3113 C          NODE FORCES DUE TO THERMAL DISTORTION
3114 C
3115      300 IF (KTL.EQ.0) GO TO 325
3116          DO 320 I=1,ND
3117          DO 310 K=1,4
3118      310 RE(I,K) = FT(I)* TLF(K)
3119      320 CONTINUE
3120 C
3121 C          NODE FORCES DUE TO STATIC ACCELERATIONS
3122 C
3123 C
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          RE(K1,L) = RE(K1,L) + XLF(L)*DL(I)
3131          RE(K2,L) = RE(K2,L) + YLF(L)* DL(I)
3132          330 RE(K3,L) = RE(K3,L) + ZLF(L)* DL(I)

```

```

3133 340 CONTINUE
3134 C
3135 C COMPUTE NODE FORCES DUE TO ELEMENT SURFACE LOADINGS
3136 C
3137 350 IF(NDLS.LT.1.OR.NDYN.LT.0) GO TO 405
3138 C
3139 DO 400 L=1,4
3140 IF(PLS(L).EQ.0.0) GO TO 400
3141 M = KLS(L)
3142 IF(M.LT.1) GO TO 400
3143 DO 360 K=1,ND
3144 C
3145 360 PL(K) = 0.0
3146 CALL FACEPR (ANEL,NDIS,KOZ,XX,NOB9M,H,P,PL,NEFACE(M),LT,M,
3147 1 PWA(1,M),M)
3148 C
3149 DO 370 I=1,ND
3150 C
3151 370 RE(I,L) = RE(I,L) + PL(I)*PLE(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 = NOB9M(I-8)
3162 II = KOD(JJ)
3163 415 K = K+3
3164 DO 420 L=1,3
3165 M = K+L
3166 420 LM(M) = ID(II,L)
3167 C
3168 IF(KIOP.GT.0) NS = G*MAXPTS(KIOP)
3169 IF(KIOP.EQ.0) NS = 6
3170 IF (NDYN.GT.0) NS=42
3171 C
3172 C SAVE STIFFNESS AND LOAD MATRICES
3173 C
3174 CALL CALBAN (MBAND,NDIF,LM,XM,SS,RE,ND,63,NS)
3175 C
3176 C COMPUTE STRESS RECOVERY MATRICES
3177 C
3178 IF (NDYN.LT.1) GO TO 425
3179 NOP=7
3180 DO 422 I=1,7
3181 422 LOCOP(I)=I + 20
3182 GO TO 450
3183 425 IF (KIOP.EQ.0) GO TO 440
3184 NOP = MAXPTS(KIOP)
3185 DO 430 I=1,NOP
3186 430 LOCOP(I) = LOC(I,KIOP)

```

```

3187      GO TO 450
3188  440  NOP = 1
3189      LOCOP(1) = 21
3190 C
3191  450  IF(MODEX.EQ.1) GO TO 510
3192 C
3193 C      CONSIDER EACH OUTPUT LOCATION
3194 C
3195      DO 500 L=1,NOP
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 DER3DS (NEL,XX,L,DET,E1,E2,E3,NOD9M,H,P,RDIS,KXYZ)
3205 C
3206      DO 470 I=1,6
3207      N= 6*(L-1)+I
3208      DO 465 J=1,ND
3209      Q = 0.0
3210      DO 460 K=1,6
3211  460  Q = Q + D(I,K)* B(K,J)
3212  465  SDI(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. NDYN.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,NDIS
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 * SIGDT(K)
3230 C
3231      DO 490 I=1,6
3232      N = 6*(L-1)+I
3233 C
3234      DO 490 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 1WRITE (1) ND,NS,(LM(I),I=1,ND),((SDT(I,J),I=1,NS),J=1,ND),
3246 2 ((SF(I,J),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,KIOP,TTZ,KRS,KRSINT,RTINT,
3251 1 KREUSE,KLS
3252 WRITE (33,3017) (KOD(I),I=1,NREAD)
3253 C
3254 C*** DATA PORTHOLE SAVE
3255 IF(NT8SV.EQ.1)
3256 1WRITE (NT8) NEL,KDIS,KXYZ,KMAT,KAXES,KIOP,TTZ, KRSINT,RTINT,
3257 2 KREUSE,KLS,NREAD,
3258 3 (KOD(I),I=1,NREAD)
3259 C***
3260 C
3261 C CHECK FOR THE LAST ELEMENT
3262 C
3263 IF(NUML-NEL) 530,600,530
3264 530 IF(ML) 50,50,60
3265 C
3266 600 RETURN
3267 C
3268 C FORMATS
3269 C
3270 1008 FORMAT (G15,F10.0,4I5,4I2)
3271 1009 FORMAT (16I5)
3272 C
3273 3001 FORMAT ( 7X,34HNUMBER OF 21-NODE ELEMENTS = 16//
3274 1 7X,34HNUMBER OF MATERIAL SETS = 16//
3275 2 7X,26HMAXIMUM NUMBER OF MATERIAL, /
3276 3 7X,34HTEMPERATURE INPUT POINTS = 16 //
3277 4 7X,19HNUMBER OF MATERIAL, /
3278 5 7X,34HAXIS ORIENTATION SETS = 16//
3279 * 7X,34HNUMBER OF DISTRIBUTED LOAD SETS = 16//
3280 6 7X,34HMAXIMUM NUMBER OF ELEMENT NODES = 16 //
3281 7 7X,34HNUMBER OF STRESS OUTPUT SETS = 16 //
3282 8 7X,34HR,S COORDINATE INTEGRATION ORDER = 16 //
3283 9 7X,34HT COORDINATE INTEGRATION ORDER =16 // 16
3284 3014 FORMAT (52H13 / D 3 T O 2 1 N O D E S O L I D E L E ,
3285 1 18H M E N T D A T A , // 8H ELEMENT 2(2X,5HNODES),2(2X,
3286 2 5HMAIL.),2X,6HSTRESS,4X,6HSTRESS,2X,4HNODE,2(2X,5HGAUSS),2X,
3287 3 2HK-,5X,3HLSA,3X,3HLSB,3X,3HLSC,3X,3HLSD, /
3288 4 8H NUMBER,7H -NDIS-,7H -NXYZ-,2X,5HTABLE,3X,4HAXES,2X,4HOUTPUT,
3289 5 6X,4HFKEE,2X,4HINC.,2(3X,4HPTS.),2X,6HMATRIX,2X,4(2X,4H-OR-), /
3290 6 26X,3HNO.,4X,3HSET,5X,3HSET,5X,5HTEMP.,2X,4H-KG-,2X,5H-R,9-,4X,
3291 7 3H-I-,2X,6HRE-USE,2X, 3(2X,2HN-,I2) )
3292 3015 FORMAT (18,4I7,18,F10.1,10,2I7,18,2X,4I6)
3293 3016 FORMAT (84X,8(2X,2HN-,I2),:/ 84X,5(2X,2HN-,I2) )
3294 3017 FORMAT (84X,8I6,:/ 84X,8I6,:/ 84X,5I6)

```

```

3295 C
3296 4014 FORMAT (33HOERROR*** ENCOUNTED ELEMENT (,15,13H), BUT EXPECT,
3297 1 21H TO READ ELEMENT ONE., / 1X)
3298 4015 FORMAT (42HOERROR*** NUMBER OF DISPLACEMENT NODES (,15,4H) IS,
3299 1 30H LARGER THAN MAXIMUM ALLOWED (,15,2H)., / 1X)
3300 4016 FORMAT (40HOERROR*** NUMBER OF COORDINATE NODES (,15,6H) MUST,
3301 1 39H BE .LE. NUMBER OF DISPLACEMENT NODES (,15,2H).)
3302 4017 FORMAT (36HOERROR*** ILLEGAL MATERIAL NUMBER. )
3303 4018 FORMAT (44HOERROR*** ILLEGAL MATERIAL AXIS REFERENCE. )
3304 4019 FORMAT (41HOERROR*** ILLEGAL OUTPUT SET REFERENCE. )
3305 4020 FORMAT (41HOERROR*** PRESSURE LOAD SET REFERENCE (,15,4H) IS,
3306 1 9H ILLEGAL. )
3307 4021 FORMAT (16HOERROR*** THE ,12,18H-TH ELEMENT NODE (,15,4H) IS,
3308 1 9H ILLEGAL., / 1X)
3309 4022 FORMAT (28HOERROR*** ELEMENT NUMBER (,15,11H) IS OUT OF,
3310 1 10H SEQUENCE., / 1X)
3311 4023 FORMAT (42HOERROR*** NUMBER OF DISPLACEMENT NODES (,15,
3312 1 25H) MUST BE AT LEAST EIGHT. )
3313 4024 FORMAT (40HOERROR*** NUMBER OF COORDINATE NODES (,15,
3314 1 25H) MUST BE AT LEAST EIGHT. )
3315 4025 FORMAT (38HOERROR*** NUMBER OF NON-ZERO NODES (,13,6H) READ,
3316 1 50H DOES NOT EQUAL THE NUMBER OF DISPLACEMENT NODES (,
3317 2 13,2H)., / 1X)
3318 4030 FORMAT (33HOERROR*** AVERAGE TEMPERATURE (,F10.2,5H) FOR,
3319 1 10H ELEMENT (,15,29H) OUT OF RANGE FOR MATERIAL (,13,
3320 2 2H)., / 1X)
3321 4099 FORMAT (12X,31HPROCEED IN DATA CHECK ONLY MODE, / 1X)

```

3322 C

3323 END

3324 C=====

3325 SUBROUTINE VECTR2 (V,XI,YI,ZI,XJ,YJ,ZJ,IERR)

3326 C

3327 C CALLED BY : INP21

3328 C

3329 C IMPLICIT REAL\*8(A-H,O-Z)

3330 C

3331 C THIS ROUTINE FORMS A UNIT LENGTH VECTOR AVA FROM POINT AIA

3332 C TO POINT AJA IN X,Y,Z SPACE

3333 C

3334 C DIMENSION V(3)

3335 C

3336 C IERR = 1

3337 C X = XJ - XI

3338 C Y = YJ - YI

3339 C Z = ZJ - ZI

3340 C XLN =DSQRT(XAX+YAY+ZAZ)

3341 C IF(XLN.LE.1.0E-08) RETURN

3342 C XLN = 1.0 / XLN

3343 C IERR = 0

3344 C V(3) = Z \* XLN

3345 C V(2) = Y \* XLN

3346 C V(1) = X \* XLN

3347 C RETURN

3348 C END

```

3349 C1=====
3350      SUBROUTINE STIME
3351      TS=0.0
3352      RETURN
3353      END
3354 C
3355 C
3356 C1=====
3357 C      SUBROUTINE TTIME
3358 C          T - CUMULATIVE TASK TIME, RETURNED IN UNITS OF SECONDS
3359      SUBROUTINE TTIME(T)
3360      INTEGER*4 get_time, time
3361      DATA get_time /2/
3362      CALL LIB$STAT TIMEK(get_time,time)
3363      T = time / 100.0
3364      RETURN
3365      END
3366 C1=====
3367
3368      SUBROUTINE SOLEQ
3369      IMPLICIT REAL*8(A-H,O-Z)
3370 C
3371 C      CALLS:  SESOL,PKINTU,STRESS
3372 C      CALLED BY:  MAIN
3373 C
3374 C      STATIC SOLUTION PHASE
3375 C
3376      COMMON A(1)          ! ?????????
3377      COMMON /ELPAR/ NP(14),NUMNP,MBAND,NELTYP,N1,N2,N3,N4,N5,MTOT,NEQ
3378      COMMON /SOL / NBLOCK,NEQB,LL,NE,IFILL(7)
3379      DIMENSION IB(3000,6),B(40000),NX(3,200)
3380      COMMON/CRK. NCKRD, ICR(9000)
3381      DATA NCKRD/4/      ! ?????????
3382
3383 !xxxx MIN. DIMENSION = ( nb + 6*npair) : also check in SESOL routine
3384      DIMENSION TMAT(600,600)
3385      COMMON/INT. TMAT(600,600),TCOL(600),TCOL2(600),TCOLM(600),
3386      .IST(600),K(9000)
3387
3388      dimension nbc(200),ifix(200),dsave(200),npt(2,200),hcol,
3389      . FORC1(3,200),FORC2(3,200),FORC3(3,200),ENERG(3)
3390
3391 C
3392      REAL*4 TT(4),CCSUB(10)
3393      INTR= 100          ! INTERMEDIATE PRINTING
3394
3395      rewind 5
3396      read (8) ((idn,i,n=1,numnp),i=1,6)      ! degrees of freedom
3397 c      WRITE(23,*)      '*** id = array in soleq *****'
3398 c      do n=1,numnp
3399 c          WRITE(23,1028) (idn,i),i=1,6)
3400 c      end do
3401 1028 format (2i,20i5)
3402

```



```

3403 *** TO read data regarding
3404 c nodes where forces to be found whose disp. are specified (NB)
3405 c & double nodes along the crack propagation (NPAIR)
3406 read (5,*),nb ! no of boundary nodes where forces are to found
3407 IF(INTPR.LE.2) WRITE(33,*),nb
3408 c -- input node, degree of freedom, displacement
3409 do ij=1,nb
3410 read(5,*),nbc(ij),ifix(ij),dsave(ij)
3411 ! if(INTPR.LE.2)
3412 WRITE(33,*),nbc(ij),ifix(ij),dsave(ij)
3413 end do
3414 ncrkd=0
3415 if(INTPR.LE.2)WRITE(33,*),node,ix,ndof,ncrkd,icr(ndof),ist(ncrkd)
3416 if(INTPR.LE.2)WRITE(33,*),ij,(id(ij,ix),ix=1,6),ij=1,169
3417
3418 do ib=1,nb
3419 node=nbc(ib)
3420 ix=ifix(ib)
3421 ndof=id(node,ix)
3422 ncrkd=ncrkd+1
3423 if(INTPR.LE.2)
3424 . WRITE(33,1029),node,ix,ndof,ncrkd
3425 icr(ndof)=ncrkd
3426 ist(ncrkd)=-ncrkd
3427 if(INTPR.LE.2)
3428 . WRITE(33,1029),node,ix,ndof,ncrkd,icr(ndof),ist(ncrkd)
3429 end do
3430 1029 format (2x,5i8,5i8)
3431 if(INTPR.LE.1) WRITE(33,1028) (ICR(IJ),IJ=1,NEQ)
3432 if (INTPR.LE.2) WRITE(33,1028) IST
3433 1030 FORMAT (2x,20F6.3)
3434
3435
3436 read (5,*),npair ! no of double nodes & the double nodes
3437 ! if(INTPR.LE.2)
3438 WRITE(33,*),npair
3439 do ipair=1,npair
3440 read(5,*),npc(1,ipair),i=1,2),(NX(IJ,ipair),IJ=1,3)
3441 ! if(INTPR.LE.2)
3442 WRITE(33,*),npc(1,ipair),i=1,2),(nx(ij,ipair),ij=1,3)
3443 do ix=1,2
3444 node=npc(ix,ipair)
3445 do iy=1,3 ! d.o.f. at each node
3446 ndof=id(node,iy)
3447 if(ndof.ne.0.and.icr(ndof).eq.0.and.nx(iy,ipair).eq.1) then
3448 ncrkd=ncrkd+1
3449 icr(ndof)=ncrkd
3450 ist(ncrkd)=ncrkd
3451 end if
3452 end do
3453 end do
3454 end do
3455
3456

```

```

3457 C
3458 C   SOLVE FOR THE DISPLACEMENT VECTORS
3459 C
3460   CALL TTIME(TT(1))
3461
3462 C-   N1=1
3463 C-   NCRKD - TOTAL NO. OF ADDITIONAL COLUMNS
3464   LL=1+NCRKD
3465
3466   NSB0=(MBAND+1)*ANEQB
3467   NSB=(MBAND+LL)*ANEQB
3468 C   N3=NSB+1
3469   N3=N3+LL*ANEQB
3470   NSBB=NEQB*LL*(2+MBAND-1)/NEQB
3471   IF(NSBB.LT.NSB) NSBB=NSB
3472   N4=N3+NSBB
3473   M1 = MBAND + NEQB -1
3474   IF(INTPR.LE.2)
3475     .WRITE(33,x) LL,MBAND,NSB,N3,NSBB,N4,M1,N1,N2,N3
3476   IF(INTPR.LE.2)
3477     .WRITE(33,10301) LL,MBAND,NSB,N3,NSBB,N4,M1,N1,N2,N3
3478 10301   format(2x,12i8)
3479   rewind 15
3480   rewind 4
3481   do ij=1,nblock
3482     read(4)(A(IK),IK=1,NSB0)
3483     WRITE(15)(A(IJK),IJK=1,NSB0)
3484   end do
3485   CALL SESOL(A(N1),A(N3),A(N4),LL,NBLOCK,NEQB,NSB,M1,4,3,2,55)
3486   CALL TTIME(TT(2))
3487   NL=2
3488   NL1=18
3489   NWV=LL*ANEQB
3490   REWIND NL
3491   REWIND NL1
3492   DO NJ=1,NBLOCK
3493     READ(NL)(A(IJ),IJ=1,NWV)
3494     WRITE(NL1)(A(IJ),IJ=1,NWV)
3495   END DO
3496
3497   WRITE(16) TMAT,tool
3498   do ipair=1,npair
3499     node1=npc(1,ipair)
3500     node2=npc(2,ipair)
3501     do idf=1,6
3502       ndof1=id(node1,idf)
3503       ndof2=id(node2,idf)
3504       if(ndof1.ne.0.and.ndof2.ne.0) then
3505         icr1=icr(ndof1)
3506         icr2=icr(ndof2)
3507         if(ist(icr1).gt.0.and.ist(icr2).gt.0) ist(icr2)=ist(icr1)
3508       end if
3509     end do
3510   end do

```

```

3511
3512
3513      write (33,*), ' ---      NODES      ----      D.O.F. RELEASED ----'
3514      do itr=1,100
3515      call ttime(ttsub(1))
3516      read (5,*), ip1,ip2,ix
3517      write (33,*), ip1,ip2,ix
3518      if(ip1.eq.9999.and.ip2.eq.9999) go to 1995
3519      do while (ip1.ne.0)
3520      if (ip2.eq.0) then
3521          node=ip1
3522          ndof=id(node,ix)
3523          if(ndof.ne.0) then
3524              icr1=icr(ndof)
3525              ist(icr1)=icr1
3526          end if
3527          else
3528              node1=ip1
3529              node2=ip2
3530              idf=ix
3531              ndof1=id(node1,idf)
3532              ndof2=id(node2,idf)
3533              if(ndof1.ne.0.and.ndof2.ne.0) then
3534                  icr1=icr(ndof1)
3535                  icr2=icr(ndof2)
3536                  ist(icr2)=icr2
3537              end if
3538          end if
3539      read (5,*), ip1,ip2,ix
3540      write (33,*), ip1,ip2,ix
3541      end do
3542
3543      if(INIPR.LE.2) then
3544      WRITE(33,*) ' ----- IST -----'
3545      WRITE(33,1028) (ist(ij),ij=1,nerkd)
3546      end if
3547
3548      REWIND 16
3549      READ (16) TMAT,tcol
3550
3551      do i=1,neq
3552      icri=icr(i)
3553      isti=ist(icri)
3554      if(icri.le.nb.and.isti.gt.0) tcol(icri)=tcol(icri)+tcol2(icri)
3555      end do
3556
3557      do i=1,nerkd
3558      isti=ist(i)
3559      do j=1,nerkd
3560      istj=ist(j)
3561      if(isti.gt.0.and.istj.le.0) tcol(i)=tcol(i)-TMAT(i,j)*dsave(j) !disp
3562      if(isti.le.0.or.istj.le.0) TMAT(i,j)=0.
3563      if(isti.le.0.and.istj.le.0.AND.isti.eq.istj) TMAT(i,j)=1.
3564      end do

```

```

3565     end do
3566     IF(INTPR.LE.1) WRITE(33,* )
3567     ' ---- TMAT,TCOL after IST manipulation-----'
3568     DO I=1,NCRKD
3569     IF(INTPR.LE.1) WRITE(33,1091) (TMAT(I,J),J=1,NCRKD),TCOL(I)
3570     END DO
3571     do i=1,norkd
3572     tcolm(i)=0.
3573     do j=1,norkd
3574     tmatm(i,j)=0.
3575     end do
3576     end do
3577     do i=1,norkd
3578     ist=iabs(ist(i))
3579     tcolm(ist)=tcolm(ist)+tcol(i)
3580     do j=1,norkd
3581     istj=iabs(ist(j))
3582     tmatm(ist,istj)=tmatm(ist,istj)+TMAT(i,j)
3583     end do
3584     end do
3585     do i=1,norkd
3586     if(tmatm(i,i).eq.0) tmatm(i,i)=1.0
3587     end do
3588     IF(INTPR.LE.1) WRITE(33,* ) ' ---- TMATm,TCOLm before matin-----'
3589     DO I=1,NCRKD
3590     IF(INTPR.LE.1) WRITE(33,1091) (TMATm(I,J),J=1,NCRKD),TCOLm(I)
3591     end do
3592
3593     call ttime(ttsub(2))
3594
3595     call MATIN(tmatm,Norkd,tcolm,1,DETERM)
3596
3597     call ttime(ttsub(3))
3598     IF(INTPR.LE.1) WRITE(33,* ) ' ---- TMATm,TCOLm after matin-----'
3599     DO I=1,NCRKD
3600     IF(INTPR.LE.1) WRITE(33,1091) (TMATm(I,J),J=1,NCRKD),TCOLm(I)
3601     end do
3602     do i=1,norkd
3603     ist=iabs(ist(i))
3604     tcol(i)=tcolm(ist)
3605     if(ist(i).le.0) tcol(i)=dsave(i)      ! disp
3606     end do
3607     NWV=LL*NEQB
3608     Rewind nll
3609     do nj=1,nblock-1
3610     read (nll) (c(ij),ij=1,nwv)
3611     end do
3612     do nj=1,nblock
3613     nconst=(nj-1)*neqb
3614     read (nll) (c(ij),ij=1,nwv)
3615     backspace nll
3616     backspace nll
3617     do i=1,neqb
3618     b(i)=nconst+c(i)

```

```

3619      do k=1,nckd
3620      nk=neqb+(k-1)*neqb+1
3621      b(i+nconst)=b(i+nconst)-a(nk)*tcol(k)
3622      end do
3623      end do
3624
3625      IF(INIPR.LE.1) WRITE(33,*) ' --- intermediate solution ---'
3626      IF(INIPR.LE.1) WRITE(33,1091) (b(ij),ij=1,neq)
3627
3628      end do !nj
3629      do i=1,neq
3630      icl=icr(i)
3631      if(icl.ne.0) b(i)=tcol(icl)
3632      end do
3633 c      WRITE(33,*) ' --- final displacement solution ---'
3634 c      WRITE(33,1091) (b(ij),ij=1,neq)
3635      REWIND NL
3636      DO NJ=NBLOCK,1,-1
3637      NCONST=(NJ-1)*NEQB+1
3638      NU=NCONST+NEQB-1
3639      WRITE(NL) (B(IJ),IJ=NCONST,NU)
3640      END DO
3641
3642      call ttime(ttsub(4))
3643
3644      do i=1,neq
3645      r(i)=0.
3646      end do
3647      rewind 15
3648      do nj=1,nblock
3649      DO IJK=NSBO,NSB
3650      A(IJK)=0.0
3651      END DO
3652      read (15) (a(IJK),IJK=1,NSBO)
3653      nconst=(nj-1)*neqb
3654      ij=0
3655      J1=1+NCONST
3656      do i=1,neqb
3657      ij=ij+1
3658      in=i+nconst
3659      r(in)=r(in)+a(ij)*b(in)
3660      end do
3661      do j=2,M&AND
3662      do i=1,neqb
3663      in=i+nconst
3664      jn=j+nconst+i-1
3665      ij=ij+1
3666      r(in)=r(in)+a(ij)*b(jn)
3667      r(jn)=r(jn)+a(ij)*b(in)
3668      end do
3669      end do
3670      end do
3671 c      WRITE(33,*) ' ----- r vector -----'
3672 c      WRITE(33,1091) (r(ij),ij=1,neq)

```

```

3673
3674 C
3675 C      Correction for thermal case --
3676 c      To find the mechanical loads subtract the thermal loads
3677 c      from R(*)
3678 c      ( It is assumed no external loads are applied at double nodes)
3679      Rewind 15
3680      nsbl=neqbt*mband
3681      do nj=1,nblock
3682      read (15) (a(ij),ij=1,nbte)
3683      nconst=(nj-1)*neqb
3684      do ij=1,neqb
3685      r(ij+nconst)=r(ij+nconst)-a(nsbl+ij)
3686      end do
3687      end do
3688
3689      N=NUMNP
3690      ITR1=ITR-1
3691      WRITE(33,582) ITR1
3692 582  FORMAT (1H1, ' ***** STEP # ', I4, ' *****'/1X,40(1H_))
3693      WRITE(33,*) ' *****'
3694      WRITE(33,*) ' ---- GLOBAL DISPLACEMENTS AND FORCES IN SOLID ----'
3695      WRITE(33,*) ' *****'
3696 CC  WRITE(33,*) ' (no mech. loads at the double nodes)'
3697      WRITE(33,20034)
3698      WRITE(33,20035)
3699 20034 FORMAT(/2X,75(1H-))
3700
3701      NAUX=1
3702      DO 500 N=1,NUMNP
3703      IFLAG=0
3704      DO 250 I=1,3
3705      D(I)=0.
3706      D(I+3)=0.0
3707 150 IF (ID(N,I).LT.1) GO TO 250
3708      IDNI=ID(N,I)
3709      IF (.NOT.(IDNI).NE.0) IFLAG=1
3710 200 D(I)=R(NAUX)
3711      D(I+3)=R(NAUX)
3712      NAUX=NAUX+1
3713 250 continue
3714 C
3715 C      IF (.NOT.(IFLAG.EQ.0)) GO TO 500
3716      WRITE (33,2004) N,(D(I),I=1,6)
3717 C
3718 500 CONTINUE
3719
3720 1091 format (2X,6G12.5)
3721 2004 FORMAT(2X,15,6G12.5)
3722      WRITE(33,20034)
3723 20035 FORMAT(2X,' NODE',3X,' U',11X,' V',11X,' W',10X,' Fx',10X,' Fy',
3724      . 10X,' Fz'/2X,75(1H-))
3725
3726      IF (ITR.NE.1) THEN

```

```

3727      ENERG(1)=0.0
3728      ENERG(2)=0.0
3729      ENERG(3)=0.0
3730      DO IPAIR=1,NPAIR
3731      NP1=NPC(1,IPAIR)
3732      NP2=NPC(2,IPAIR)
3733      DO IDG=1,3
3734      ND1=ID(NP1,IDG)
3735      ND2=ID(NP2,IDG)
3736 C-    WRITE(33,*) 'NP1,NP2,ND1,ND2',NP1,NP2,ND1,ND2
3737 C-    WRITE(33,*) 'FORC1,B s',FORC1(IDG,IPAIR),B(ND1),B(ND2)
3738      NAUX=NX(IDG,IPAIR)
3739      IF(ND1.GT.0.AND.ND2.GT.0.and.naux.ne.0)
3740      . ENERG(IDG)=ENERG(IDG)-(FORC2(IDG,IPAIR)*B(ND2)+
3741      . FORC1(IDG,IPAIR)*B(ND1))*0.50
3742      END DO
3743      END DO
3744
3745
3746      DO IB=1,NB
3747      NPO=NBC(IB)
3748 C-    DO IDG=1,3
3749      IDG=ifix(ib)
3750      NDO=ID(NPO,IDG)
3751      IF(NDO.GT.0) ENERG(IDG)=ENERG(IDG)-FORCIB(IDG,IB)*B(NDO)*0.50
3752 C-    END DO
3753      END DO
3754
3755      WRITE(34,*)
3756      . '----- ENERGY RELEASED in ( x, y, z ) directions -----'
3757      WRITE(34,1048) ENERG
3758      IF (ITR.EQ.2) WRITE(19,1049)
3759      WRITE(19,1048) ENERG
3760 1048  format (5x,3(g15.8,3x))
3761 1049  format (5x,7X,'X',18X,'Y',18X,'Z')
3762      WRITE(34,*) ' ====='
3763      END IF
3764      DO IPAIR=1,NPAIR
3765      NP1=NPC(1,IPAIR)
3766      NP2=NPC(2,IPAIR)
3767      DO IDG=1,3
3768      ND1=ID(NP1,IDG)
3769      ND2=ID(NP2,IDG)
3770      FORC1(IDG,IPAIR)=R(ND1)
3771      FORC2(IDG,IPAIR)=R(ND2)
3772 C-    WRITE(33,*) 'NP1,NP2,ND1,ND2', NP1,NP2,ND1,ND2
3773 C-    WRITE(33,*) ' IDG,IPAIR, FORC1,FORC2',IDG,IPAIR,FORC1(IDG,IPAIR),
3774 C-    . FORC2(IDG,IPAIR)
3775      END DO
3776      END DO
3777
3778      DO IB=1,NB
3779      NPO=NBC(IB)
3780      DO IDG=1,3

```

```

3781      NDO=ID(NPO, IDG)
3782      FORCNB(IDG, IB)=K(NDO)
3783      END DO
3784      END DO
3785
3786      call time(ttsub(5))
3787
3788 C      PRINT DISPLACEMENTS
3789 C
3790      N2=N1+NUMDFAG
3791      N3=N2+6*LL
3792
3793      LL1=1          !***** REASSIGNED *****
3794
3795      CALL PRINTD (A(N1),A(N2),A(N3),NEQB,NUMNP,LL1,NBLOCK,NEQ,3,1)
3796      CALL TIME=TT(3)
3797 C
3798 C      COMPUTE AND PRINT ELEMENT STRESSES
3799 C
3800      N2=N1+4*LL1
3801      N3=N2+NEQB*LL1
3802      LB=(MTOT-N3)/(NEQ +12)
3803      CALL STRESS(A(N1),A(N2),A(N3),NEQB,LB,LL1,NEQ,NBLOCK)
3804
3805 C      COMPUTE TIME LOG FOR THE DOUBLE NODES SOLUTION PHASE
3806 C
3807      DO K=1,4
3808      ttsub(K) = ttsub(K+1)-ttsub(K)
3809      end do
3810      IF(INTPR.LE.2) WRITE (35,1985) (ttsub(L),L=1,4)
3811 1985   format(5X,' time for tmatm formation      =',f8.2,/)
3812   .      5X,' time for matin                    =',f8.2,/)
3813   .      5X,' time to find global disp.         =',f8.2,/)
3814   .      5X,' time to find global nodal forces =',f8.2//)
3815 C
3816      end do      !itr
3817 1995   continue
3818      CALL TIME=TT(4)
3819 C
3820 C      COMPUTE TIME LOG FOR THE STATIC SOLUTION PHASE
3821 C
3822      DO 50 K=1,3
3823 50     TT(K) = TT(K+1)-TT(K)
3824      WRITE (34,2000) (TT(L),L=1,3)
3825 C
3826 2000  FORMAT (//) 48H S T A T I C   S O L U T I O N   T I M E   L O G ,
3827   1      4SA,31HEQUATION SOLUTION      =, F8.2 /
3828   2      5SA,31HDISPLACEMENT OUTPUT    =, F8.2 /
3829   3      5SA,31HSTRESS RECOVERY        =, F8.2 //)
3830 C
3831 C      RETURN
3832 C      RETURN
3833 C      END
3834 C!=====

```



```

3835
3836      SUBROUTINE SESOL
3837      .(A,B,MAXA,NV,NBLOCK,NEQB,NAV,MI,NSTIF,NRED,NL,NR)
3838      IMPLICIT REAL*8(A-H,O-Z)
3839
3840      real*4 tt(10)
3841 C      CALLED BY: SOLEQ
3842
3843      COMMON /ELPAR/ NF(14),NUMNP,MA,NELTYP,NZ1,NZ2,NZ3,NZ4,N5,NTOT,NEQ
3844      COMMON/CRK/NCRKD,ICR(9000)          ! NEQ
3845 c-      DATA ICR/0,1,2,0,0,3,0,4/      !change CVT,CVT2 line also
3846 c-      DATA IST/-1,-2,-3,-4/         ! IST(NCRKD)
3847 c-      DATA DISP/2.0,3.0,2.0,2.0,6*0.0/ ! DISP(NEQ)
3848      COMMON/TMT/TMT(600,600),TCOL(600),TCOL2(600),TCOLM(600),
3849      .IST(600),R(9000)
3850      DIMENSION A(NAV),B(NAV),MAXA(MI)
3851      call ttime(tt(1))
3852      INTPR=100
3853      if (INTPR.LE.2) WRITE(33,*)'NV,NBLOCK,NEQB,NAV,MI,NSTIF,NRED,NL,NR'
3854      if (INTPR.LE.2) WRITE(33,1029),NV,NBLOCK,NEQB,NAV,MI,NSTIF,NRED,NL,NR
3855 1029      format(2X,12I8)
3856
3857      if (INTPR.LE.2) WRITE(33,1028) (ICR(IJ),IJ=1,NEQ)
3858      if (INTPR.LE.2) WRITE(33,1028) (IST(IJK),IJK=1,NCRKD)
3859 1028      FORMAT( 2X,20I4)
3860      if (INTPR.LE.2) WRITE(33,1030) DISP
3861 1030      FORMAT (2X,20F6.3)
3862      MM=1
3863      MA2=MA - 2
3864      IF(MA2.EQ.0) MA2=1
3865      INC=NEQB - 1
3866      NWA=NEQB*MA
3867      NTB=(MA-2)/NEQB + 1
3868      NEB=NTB*NEQB
3869      NEBT=NEB + NEQB
3870      NWV=NEQB*NV
3871      NWVV=NEBT*NV
3872
3873      N1=NL
3874      N2=NR
3875      if (INTPR.LE.2)
3876      .WRITE(33,*)', mm,ma2,inc,neqb,nwa,ntb,neb,nebt,nwv,nwvv'
3877      if (INTPR.LE.2)
3878      .WRITE(33, 1029), mm,ma2,inc,neqb,nwa,ntb,neb,nebt,nwv,nwvv
3879      REWIND NSTIF
3880      REWIND NRED
3881      REWIND N1
3882      REWIND N2
3883
3884      if (INTPR.LE.2) WRITE(33,*) ' NAV =',NAV          !***
3885      DO IJ=1,NAV          !***
3886      A(IJ)=0.          !***
3887      B(IJ)=0.          !***
3888      END DO          !***

```

```

3889
3890
3891 *** Taking the apair coeffs. out and placing in a vertical matrix
3892 C- NCRKD - TOTAL NO. OF ADDITIONAL COLUMNS
3893 IF(INTR.LE.1) WRITE(20,*) ' IB,JB,I,J,ICI,ICJ,IJ,ICJ,JICI,A(IJ)'
3894 NO=NAV-NCRKD*NEQB
3895 DO I=1,MI
3896 DO J=1,NCRKD
3897 IJ=I+(J-1)*MI
3898 B(IJ)=0.
3899 END DO
3900 END DO
3901 DO NJ=1,NBLOCK
3902
3903 DO J=1,NCRKD
3904 IJ=(J-1)*MI
3905 INJ=NEQB+(J-1)*MI
3906 DO I=1,MA-1
3907 IJ=IJ+1
3908 INJ=INJ+1
3909 B(IJ)=B(IJ)
3910 END DO
3911 END DO
3912
3913 do j=1,ncrkd
3914 ij=ma+(j-1)*mi
3915 do i=ma,mi
3916 b(ij)=0.
3917 ij=ij+1
3918 end do
3919 end do
3920
3921
3922 READ (UNIT) (A(IJ),IJ=1,NO)
3923 NCONST=NEQB*(NJ-1)
3924 DO IB=1,NEQB
3925 DO JB=1,MA
3926 IJ=(JB-1)*NEQB+IB
3927 I=IB+NCONST
3928 J=JB+NCONST+IB-1
3929 IF(I.LE.NEQ.AND.J.LE.NEQ) THEN
3930 ICI=ICR(I)
3931 ICJ=ICR(J)
3932 J2ICI=JB+(ICI-1)*MI
3933 IBICJ=IB+(ICJ-1)*MI
3934 IF(ICI.NE.0) B(J2ICI)=A(IJ)
3935 IF(ICJ.NE.0) B(IBICJ)=A(IJ)
3936
3937
3938 IF(ICI.NE.0.AND.ICJ.NE.0) THEN
3939 TMAT(ICI,ICJ)=A(IJ)
3940 TMAT(ICJ,ICI)=A(IJ)
3941 J2ICI=JB+(ICI-1)*MI
3942 B(J2ICI)=0.

```

```

3943      IBICJ=IB+(ICJ-1)*MI
3944      B(IBICJ)=0.
3945      END IF
3946
3947      IF (INTPR.LE.1) WRITE(33,1018) IB,JB,I,J,ICI,ICJ,II,IIICJ,JIICJ,A(IJ)
3948 1018    FORMAT(2X,9I5,8F10.3)
3949      IF (ICJ.NE.0.OR.ICI.NE.0) THEN
3950          A(IJ)=0.
3951          IF (I.EQ.J) A(IJ)=1.
3952      END IF
3953      END IF
3954      END DO
3955      END DO
3956
3957      NIJ=NEQB*NA
3958      DO I=1,NEQB
3959          NIJ=NIJ+1
3960          ICI=ICR(I+NEQB*NA)
3961          IF (ICI.NE.0) THEN
3962              TCOL(ICI)=A(NIJ)
3963              A(NIJ)=0.
3964          END IF
3965      END DO
3966
3967      NIJ=NO
3968      DO J=1,NCRKD
3969          DO I=1,NEQB
3970              NIJ=NIJ+1
3971              IJ=I+(J-1)*MI
3972              A(NIJ)=B(IJ)
3973          END DO
3974      END DO
3975
3976 c-      WRITE(33,*) 'NJ=',NJ,' Reordered A -MATRIX'
3977      IF (INTPR.LE.1) THEN
3978          DO I=1,NEQB
3979              IA=(NJ-1)*NEQB+I
3980              WRITE(33,1019) (A(IJ),IJ=I,NAV,NEQB)
3981 1019    FORMAT(2X,11F9.2)
3982          END DO
3983      END IF
3984
3985
3986      WRITE(N1) A
3987
3988      END DO      NJ = 1000
3989
3990      REWIND N1
3991      REWIND NSTIF
3992      DO NJ=1,NELDCK
3993          READ (N1) A
3994          WRITE(NSTIF) A
3995      END DO
3996

```

```

3997         IF (INTPR.LE.1) THEN
3998         WRITE(33,X) '---- TMat ----'
3999         DO I=1,NCRKD
4000         WRITE(33,1019) (TMat(I,J),J=1,NCRKD),TCOL(I)
4001         END DO
4002         end if
4003
4004
4005 C--      STOP
4006
4007 call time(0.2)
4008 **** main loop over all blocks
4009 99      REWIND NETIE
4010
4011      DO 600 NI=1,NBLOCK
4012
4013 c-      WRITE(33,X) '----- MAIN LOOP ----- NJ = ,NJ      ....
4014
4015         IF (NI.NE.1) GO TO 10
4016         READ (NETIE) (A(IJ),IJ=1,NAV)
4017         IF (INTPR.LE.1) WRITE(33,X) 'NJ=',NJ,'      A - MATRIX'
4018         IF (INTPR.LE.1) WRITE(33,1020) (A(IJ),IJ=1,NAV)
4019 1020    FORMAT(LX,11G11.4)
4020         IF (NEQ.GT.1) GO TO 100
4021         MAXA(1)=1
4022         WRITE(NREQ) A,MAXA
4023         IF (A(1) 1,174,3)
4024         1 KK=1
4025 C--      IF (INTPR.LE.1) WRITE(33,1010) KK,A(1)
4026         3 DO 5 L=1,NV
4027         5 A(1+L)=A(1+L)+A(1)
4028         KK=1+NV
4029         WRITE(NL) (A(KK),KK=2,KN)
4030         RETURN
4031 10      IF (NIE.EQ.1) GO TO 100
4032         REWIND NI
4033         REWIND N2
4034         READ (N1) (A(IJ),IJ=1,NAV)
4035         IF (INTPR.LE.1) WRITE(33,X) ' NJ=',NJ,'      A - MATRIX'
4036         IF (INTPR.LE.1) WRITE(33,1020) (A(IJ),IJ=1,NAV)
4037
4038 100      continue
4039         IF (INTPR.LE.1) WRITE(33,X) ' BEFORE FINDING COLUMN HEIGHTS
4040
4041         IMX=NEQB*(MA+NV)
4042         IF (INTPR.LE.1) WRITE(33,X) 'NEQB,MA,NV,IMX',NEQB,MA,NV,IMX
4043         DO I=1,NEQB
4044         IF (INTPR.LE.1) WRITE(33,1020) (A(IJ),IJ=1,IMX,NEQB)
4045         IMX=IMX+1
4046         END DO
4047 **** Find column heights
4048         KU=1
4049         IM=MING(MA,NEQB)
4050         MAXA(1)=1

```

```

4051      DO 110 N=2,M1
4052      IF (N.LE.MA) THEN          ****
4053      KU=KU + NEQB
4054      KK=KU
4055      MM=MINO(N,KM)
4056      ELSE                          ****
4057      KU=KU + 1
4058      KK=KU
4059      IF (N.LE.NEQB) GO TO 140
4060      MM=MM - 1
4061      END IF                          ****
4062  140      DO 160 K=1,MM
4063      IF (A(KK)) 110,160,110
4064  160      KK=KK - INC
4065  110      MAXA(N)=KK
4066
4067      IF (A(1)) 172,174,176
4068  174      KK=(NJ-1)*NEQB + 1
4069      IF (KK.GT.NEQ) GO TO 590
4070      IF(INTPR.LE.1) WRITE (33,1000) KK
4071      STOP
4072  172      KK=(NJ-1)*NEQB + 1
4073 C--      IF(INTPR.LE.1) WRITE (33,1010) KK,A(1)
4074
4075
4076 ****      Factorize leading block
4077
4078  176      DO 200 N=2,NEQB
4079      NH=MAXA(N)
4080      IF (NH-N) 200,200,210
4081  210      KL=N + INC
4082      K=N
4083      D=0.
4084      DO 220 KK=KL,NH,INC
4085      K=K - 1
4086      AKK=A(KK)
4087      C=AKK/A(K)
4088      D=D + C*AKK
4089  220      A(KK)=C
4090      A(N)=A(N) - D
4091
4092      IF (A(N)) 222,224,230
4093  224      KK=(NJ-1)*NEQB + N
4094      IF (KK.GT.NEQ) GO TO 590
4095      IF(INTPR.LE.1) WRITE (33,1000) KK
4096      STOP
4097  222      KK=(NJ-1)*NEQB + N
4098 C--      IF(INTPR.LE.1) WRITE (33,1010) KK,A(N)
4099
4100  230      IC=NEQB
4101      DO 240 J=1,MA2
4102      MJ=MAXA(N+J) - IC
4103      IF (MJ.LE.N) GO TO 240          ****
4104      KU=MINO(MJ,NH)                ****

```

```

4105      KN=N + IC
4106      C=0.
4107      DO 300 KK=KL,RU,INC
4108 300    C=C + A(KK)*A(KK+IC)
4109      A(KN)=A(KN) - C
4110 240    IC=IC + NEQB
4111
4112      K=N + NWA
4113      DO 450 L=1,NV
4114      KJ=K
4115      C=0.
4116      DO 440 KK=KL,RH,INC
4117      KJ=KJ - 1
4118 440    C=C + A(KK)/A(KJ)
4119      A(K)=A(K) - C
4120 450    K=K + NEQB
4121
4122 200    CONTINUE
4123      IF(INTPR.LE.1) WRITE(33,A) ' -- AFTER FACTORIZING LEADING BLOCK
4124      IMX=NEQB*(MA+NV)
4125      DO I=1,NEQB
4126 C-    IF(INTPR.LE.1) WRITE(33,1020) (A(IJ),IJ=I,IMX,NEQB)
4127      IMX=IMX+1
4128      END DO
4129 1205   FORMAT(2X,1069.2)
4130
4131
4132 ****   Carry over into trailing blocks
4133
4134      DO 400 NK=1,NTB
4135      IF(INTPR.LE.1) WRITE(33,X) 'NJ,NK ',NJ,NK,'      B- MAT'
4136      IF ((NK+NJ).GT.NBLOCK) GO TO 400
4137      NI=NI
4138      IF ((NJ.EQ.1).OR.(NK.EQ.NTB)) NI=NSTIF
4139      READ (NI) (B(IJ),IJ=1,NAV)
4140
4141
4142 C-    IF(INTPR.LE.1) WRITE(33,1020) (B(IJ),IJ=1,NAV)
4143      ML=NK*NEQB + 1
4144      MR=MIND((NK+1)*NEQB,MI)
4145      IF(MA.EQ.1) ML=MR
4146      MD=MI - ML
4147      KL=NEQB + (NK-1)*NEQB*NEQB
4148      N=1
4149
4150      DO 500 m=ML,MR
4151      NH=MAXA(m)
4152      RL=KL + NEQB
4153      IF (NH.LT.RL) GO TO 505      !AAA
4154      R=NEQB
4155      D=0.
4156      DO 520 KK=RL,RH,INC
4157      C=A(KK)/A(K)
4158      D=D + C*A(KK)

```

```

4159      A(KK)=C
4160  520  K=K - 1
4161      B(N)=B(N) - D
4162      IF (MD.LE.0) GO TO 530      ***
4163      IC=NEQB
4164      DO 540 J=1,MD
4165      MJ=MAXA(M+J) - IC
4166      IF (MJ.LT.KL) GO TO 540      ***
4167      KU=MINO(MJ,NH)
4168      KN=N + IC
4169      C=0.
4170      DO 575 KK=KL,KU,INC
4171  575  C=C + A(KK)X(KK+IC)
4172      B(KN)=B(KN) - C
4173  540  IC=IC + NEQB
4174
4175  580  KN=N + NWA
4176      K=NEQB + NWA
4177      DO 610 L=1,NV
4178      KJ=K
4179      C=0.
4180      DO 620 KK=KL,NH,INC
4181      C=C + A(KK)X(KJ)
4182  620  KJ=KJ - 1
4183      B(KN)=B(KN) - C
4184      KN=KN + NEQB
4185  610  K=K + NEQB
4186
4187  505  MD=MD - 1
4188  500  N=N + 1
4189
4190      IF (NTB.NE.1) GO TO 560
4191      WRITE (NRED) A,MAXA
4192      DO 570 I=1,NAV
4193  570  A(I)=B(I)
4194      GO TO 600
4195  560  WRITE (N2) J
4196
4197  400  CONTINUE
4198
4199      M=N1
4200      N1=N2
4201      N2=M
4202  590  WRITE (NRED) A,MAXA
4203
4204  600  CONTINUE
4205
4206      call ttime(tt(5))
4207
4208  ***  Vector back substitution
4209
4210      DO 700 K=1,NWWW
4211  700  B(K)=0.
4212      REWIND NL

```

```

4213
4214 DO 800 NJ=1,NBLOCK
4215 BACKSPACE NRED
4216
4217 READ (NRED) (A(IJ),IJ=1,NAV),(MAXA(IJ),IJ=1,MI)
4218 c-- WRITE(33,x) ' Vector back sub. NJ=',NJ,' A= MAT'
4219
4220 c- IF (INTPR.LE.1) WRITE(33,1020) (A(IJ),IJ=1,NAV)
4221 BACKSPACE NRED
4222 K=NEBT
4223 DO 810 L=1,NV
4224 DO 820 I=1,NEB
4225 B(K)=B(K)-NEQB
4226 820 K=K - 1
4227 310 K=K + NEBT + NEB
4228 KN=0
4229 KK=NWA
4230 NDIF=NEQB
4231 IF (NJ.EQ.1) NDIF=NEQB - (NBLOCK*NEQB - NEQ)
4232 DO 855 L=1,NV
4233 DO 850 K=1,NDIF
4234 850 B(KN+K)=A(KK+K).A(K)
4235 KK=KK + NEQB
4236 855 KN=KN + NEBT
4237 IF (MA.EQ.1) GO TO 915
4238 ML=NEQB + 1
4239 KL=NEQB
4240 DO 860 M=mL,MI
4241 KL=KL + NEQB
4242 KU=MAXA(M)
4243 IF (KU-KL) 870,870,870
4244 870 K=NEQB
4245 KM=M
4246 DO 880 L=1,NV
4247 KJ=K
4248 DO 890 KK=KL,KU,INC
4249 B(KJ)=B(KJ) - A(KK)*B(KM)
4250 890 KJ=KJ - 1
4251 KM=KM + NEBT
4252 880 K=K + NEBT
4253 860 CONTINUE
4254 N=NEQB
4255 DO 910 I=2,NEQB
4256 KL=N + INC
4257 KU=MAXA(N)
4258 IF (KU-KL) 910,920,920
4259 920 K=N
4260 DO 930 L=1,NV
4261 KJ=K
4262 DO 940 KK=KL,KU,INC
4263 KJ=KJ - 1
4264 940 B(KJ)=B(KJ) - A(KK)*B(K)
4265 930 K=K + NEBT
4266 910 N=N - 1

```



```

4267
4268 915 KK=0
4269     KN=0
4270     DO 950 L=1,NV
4271     DO 960 K=1,NEUB
4272     KK=KK + 1
4273 960  A(KK)=B(KN+K)
4274 950  KN=KN + NEBT
4275
4276     WRITE (NL) (A(K),K=1,NWV)
4277     IF(INTPR.LE.1) WRITE(33,*) ' Solution --'
4278     IF(INTPR.LE.1) WRITE (33,1020) (A(K),K=1,NWV)
4279 300  CONTINUE
4280
4281     call ttime:tt(4))
4282
4283 ****  To find y - vector
4284     do i=1,nckrd
4285     tcol2(i)=tcol(i)
4286     if(ist(i).le.0) tcol(i)=0.
4287     end do
4288     rewind nstif
4289     backspace nl
4290     nc2=neqb*na
4291     do nj=1,nblock
4292
4293     read (nstif) a
4294     nij=no
4295
4296
4297
4298     read (nl) (B(ij),ij=1,nwv)
4299     backspace nl
4300     backspace nl
4301     nij=neqb*(nv-nckrd)
4302
4303
4304     NIJ10=NO
4305     NIJ20=NEQB*(NV-NCKRD)
4306
4307     do J=1,nckrd
4308     do I=1,j
4309     tauX=0.0
4310     NIJ0=NIJ10+(I-1)*NEQB
4311     NIJ=NIJ20+(J-1)*NEQB
4312     do k=1,neqb
4313     NIJ0=NIJ0+1
4314     NIJ=NIJ+1
4315     tauX=tauX-A(NIJ0)*B(NIJ)
4316     end do
4317     tmat(i,j)=tmat(i,j)+tauX
4318     end do
4319     end do
4320     do j=1,nckrd

```

```

4321      do i=1,j-1
4322          tmat(j,i)=tmat(i,j)
4323      end do
4324  end do
4325
4326      do i=1,nckkd
4327          tau=0.0
4328          NIJO=NIJ10+(1-1)*NEQB
4329          do k=1,nckq
4330              NIJO=NIJO+1
4331              tau=tau-A*(NIJO)*B(k)
4332          end do
4333          tcol(i)=tcol(i)+tau
4334      end do
4335      IF(INTPR.LE.1) then
4336          WRITE(33,*) ' ---- TMAT,TCOL towards end ----'
4337          DO I=1,NCKKD
4338              WRITE(33,1091) (TMAT(I,J),J=1,NCKKD),TCOL(I)
4339 1091      format (2X,8gl2.5)
4340          END DO
4341      end if
4342
4343      end do      i=nj
4344      call ttime(tt(5))
4345      if(INTPR.LE.2) WRITE(33,*) ' TIME LOG IN SESDL -----'
4346      do k=1,4
4347          tt(k)=tt(k+1)-tt(k)
4348      end do
4349      if(INTPR.LE.3) WRITE(33,995) (tt(1),1=1,4)
4350 995      format(10X, Time to form matrix for double nodes etc.=,f8.2,
4351          .      /10X, Time to decompose A - matrix           =,f8.2,
4352          .      /10X, Time for vector back substitution      =,f8.2,
4353          .      /10X, Time to form TMAT                      =,f8.2)
4354 1000 FORMAT (/ 4GH  STOP *** ZERO DIAGONAL ENCOUNTERED DURING,
4355      1      18H EQUATION SOLUTION, /
4356      2      18X,18H EQUATION NUMBER =, I6 )
4357 1010 FORMAT (/ 50H WARNING *** NEGATIVE DIAGONAL ENCOUNTERED DURING,
4358      1      18H EQUATION SOLUTION, /
4359      2      18X,18H EQUATION NUMBER =, I6, 5X, THVALUE =, E20.5 )
4360
4361      RETURN
4362      END
4363 C3=====
4364      SUBROUTINE MATIN(A,N,S,m,DETERM)
4365      IMPLICIT REAL8 (A-H,O-Z)
4366      DIMENSION A(600,600),B(600,1),IPIVOT(600),INDEX(600,2),BT(600)
4367      EQUIVALENCE (IROW,JROW),(ICOLU,JCOLU),(AMAX,T,SWAP)
4368      DETERM=1.0
4369      do 20 I=1,n
4370 20      IPIVOT(I)=0.0
4371      do 550 I=1,n
4372          AMAX=0.0
4373
4374      do 105 J=1,n

```

```

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(AMAX -DABS(A(J,K))) 25,100,100
4379 35    IROW=J
4280      ICOLUM=K
4381      AMAX=DABS(A(J,K))
4382 100    CONTINUE
4383 105    CONTINUE
4384
4385      IPIVOT(ICOLUM)=IPIVOT(ICOLUM)+1
4386      IF(IROW-ICOLUM) 140,260,140
4387 140    DETERM=-DETERM
4388      do 200 L=1,N
4389      SWAP=A(IROW,L)
4390      A(IROW,L)=A(ICOLUM,L)
4391 200    A(ICOLUM,L)=SWAP
4392      IF(M) 260,260,210
4393 210    do 250 L=1,m
4394      SWAP=B(IROW,L)
4395      B(IROW,L)=B(ICOLUM,L)
4396 250    B(ICOLUM,L)=SWAP
4397 260    INDEX(I,1)=IROW
4398      INDEX(I,2)=ICOLUM
4399      PIVOT=A(ICOLUM,ICOLUM)
4400      DT(I)=PIVOT
4401      A(ICOLUM,ICOLUM)=1.0
4402      do 350 L=1,N
4403 350    A(ICOLUM,L)=A(ICOLUM,L)/PIVOT
4404      IF(m) 380,380,360
4405 360    do 370 L=1,m
4406 370    B(ICOLUM,L)=B(ICOLUM,L)/PIVOT
4407 380    do 550 L1=1,N
4408      IF(L1-ICOLUM) 400,550,400
4409 400    T=A(L1,ICOLUM)
4410      A(L1,ICOLUM)=0.0
4411      do 450 L=1,N
4412 450    A(L1,L)=A(L1,L)-A(ICOLUM,L)*T
4413      IF(M) 550,550,460
4414 460    do 500 L=1,M
4415 500    B(L1,L)=B(L1,L)-B(ICOLUM,L)*T
4416 550    CONTINUE
4417      do 710 I=1,N
4418      L=N+1-I
4419 C--    DETERM=DETERM*DT(L)
4420      IF(INDEX(L,1)-INDEX(L,2)) 630,710,630
4421 630    JROW=INDEX(L,1)
4422      JCOLUM=INDEX(L,2)
4423      do 705 K=1,N
4424      SWAP=A(K,JROW)
4425      A(K,JROW)=A(K,JCOLUM)
4426      A(K,JCOLUM)=SWAP
4427 705    CONTINUE
4428 710    CONTINUE

```

```
4429      GO 11 K=1,N
4430      IF(PIVOTR(LINE,1) GO TO 12
4431 11     CONTINUE
4432      RETURN
4433 12     WRITE(33,991)
4434 991    FORMAT(10X,'MATRIX IS DSINGULAR')
4435 740    RETURN
4436      END
```

APPENDIX - C  
LISTING OF THE POSTPROCESSOR, 'PLOT'

```

1
2
3  |-----
4  |--- THIS IS PROGRAM FOR PLOTTING 3-D GRAPHS USING TEMPLATE
5  |--- ROUTINES. FROM THE OUT OF KSAP II, THIS PROGRAM CAN
6  |--- SORT OUT STRESSES AND CORRESPONDING COORDINATE LOCATIONS.
7  |--- THE STRESSES MAY BE SCALED CONVENIENTLY AND EYE
8  |--- COORDINATES CAN BE CHOSEN TO OBTAIN DIFFERENT SIZES OF
9  |--- THE SAME 3-D PLOT.
10 |-----
11 DIMENSION AX(40),YY(40),Z(40,40),STRESS(15,15,6,10),
12 .WORK(3200),STRE(1000),PENS(6),WIV(8)
13 DIMENSION HEAD(7)
14 CHARACTER*12 FILNAM
15 INTEGER SIZ
16
17 DATA IO/57,IOB/7,IORE/3.07,OUTFIL/8.07,FONTEFIL/11.07
18 DATA PENS/1,2,3,4,1,1,1
19 DATA IORTHO,PPD,EX,SY/1,30.0,8.0,8.07
20
21 WRITE (5,X) 'ENTER FILENAME'
22 READ (5,555) FILNAM
23 555 FORMAT (A)
24 OPEN(UNIT=100,FILE=FILNAM,STATUS='OLD')
25 WRITE(10,X) '0:SCREEN,1:PRINTER,2:PLOTTER'
26 READ(10,X) IDEV
27 READ (100,22), (HEAD(I), I=1,6)
28 22 FORMAT(1X,8A4)
29
30 READ (100,X) NNODES,NLOC
31 IF (NLOC.NE.1) CALL SORT21(RAD,100,XX,YY,NX,NY,STRE,nnodes,nloc)
32 IF (NNODES.EQ.21) GO TO 899
33 IF (nloc.ne.1) go to 899
34 READ(100,X) NX,NY,NL
35 WRITE(5,X) NX,NY,NL
36 READ(100,X) (XX(I), I=1,NX)
37 WRITE(5,X) (XX(I), I=1,NX)
38 READ(100,X) (YY(I), I=1,NY)
39 WRITE(5,X) (YY(I), I=1,NY)
40 NX=NX-1
41 NY=NY-1
42 READ(100,7000) ((STRESS(I,J,K,L),K=1,6), I=1,NX), J=1,NY), L=1,NL)
43 7000 FORMAT(25X,6E15.6)
44 WRITE(5,7000) ((STRESS(I,J,K,L),K=1,6), I=1,NX),
45 .J=1,NY), L=1,NL)
46 20 WRITE(10,X) 'ENTER STRESS & LAYER #'
47 READ(10,X) S11,LR
48 WRITE(5,X) S11,LR
49 K=0
50 DO J=1,NY
51 DO I=1,NX
52 K=K+1
53 STRE(K)=STRESS(I,J,S11,LR)
54 END DO
55 END DO

```

```

55
56 c-- finding stress location coordinates..
57 c-- from nodal coordinates for 8 node element.
58     do i=1,nx
59     if (i.gt.1) xx(i)=(xx(i)+xx(i-1))/2.
60     end do
61     xx(nx)=xx(nx+1)
62     do i=1,ny
63     if (i.gt.1) yy(i)=(yy(i)+yy(i-1))/2.
64     end do
65     yy(ny)=yy(ny+1)
66
67 899  CONTINUE
68     NXNY=NX*NY
69     SMAX=-1.0E+30
70     SMIN=1.0E+30
71     DO I=1,NXNY
72     IF (STRE(I).GT.SMAX) SMAX=STRE(I)
73     IF (STRE(I).LE.SMIN) SMIN=STRE(I)
74     END DO
75     WRITE(5,*)'SMIN= ',SMIN,' SMAX= ',SMAX
76 10  WRITE(10,*)'EYEX,EY,EY,EZ'
77     READ(10,*)EX,EY,EZ
78     WRITE(10,*)'TYPE SCALE FACTOR'
79     READ(10,*) FACT
80     DO I=1,NXNY
81     STRE(I)=STRE(I)/FACT
82     END DO
83     DO J=1,NY
84     I1=1+(J-1)*NX
85     I2=I1+NX-1
86     II=0
87     DO I=I1,I2
88     II=II+1
89     Z(II,J)=STRE(I)
90     END DO
91     WRITE (85,*) (STRE(I),I=II,I2)
92     WRITE (85,*) '-----'
93     END DO
94     DO J=1,NY
95     WRITE(5,*)J,':',(Z(I,J),I=1,NX)
96     END DO
97     WRITE(5,7000) (STRE(I),I=1,NXNY)
98     WRITE (5,*) 'BEFORE SCALING..'
99 c-- SCALING THE COORDINATES----
100 c     CMAX=XX(NX)
101 c     IF (CMAX.LT.YY(NY)) CMAX=YY(NY)
102 c     WRITE (5,*) 'CMAX',CMAX
103 c     CMAX=8.0/CMAX
104 c     DO 444 I=1,NX
105 444  XX(I)=XX(I)*CMAX
106 c     DO 445 J=1,Ny
107 445  YY(I)=YY(I)*CMAX
108     FX=N2

```

```

109      FY=NY
110      IF (IDEV.EQ.2) CALL UCONFG(3.0)
111
112      CALL USTART
113      IF(IDEV.EQ.1) CALL UPSET('OUTP',OUTFIL)
114      IF (IDEV.EQ.0) CALL UERRAS
115      CALL UPSET ('ENTFILE',ENTFILE)
116 C      CALL USET ('NLSAX15')
117      CALL USET ('ZAX15')
118      CALL USET ('SKATE')
119      CALL USET ('PERC')
120      CALL USET ('GRTA')
121 C      CALL USTUD ('W19')
122      CALL UVWPRT (0.0,100.0, 0.0,100.0)
123      CALL UPVSRF (S1R2,KA,PI,WCR,EX,ET,EX,EY,EZ,PENS)
124      WRITE (5,X) '.....'
125 666   IF (IDEV.EQ.0) CALL UMODE
126 7     FORMAT (A)
127      CALL UEND
128 999   STOP
129      END
130
131      subroutine sort21 (KAD,IOB,XP,YP,NX,NY,ZP,nnodes,nloc)
132 C-----
133 C--    THE FOLLOWING ARE THE STRESS LOCATIONS USED IN KSAP II
134 C--    PROGRAM. DEPENDING ON THE REQUIRED STRESS PLANE LOCATION
135 C--    SORTING WILL BE DONE. AVERAGING IS ALSO CARRIED OUT
136 C--    BETWEEN ADJACENT ELEMENTS.
137 C--    LOC - STRESS OUTPUT LOCATIONS IN SAP....
138 C--
139 c--      6--13-- 5      18--24--12      3-- 9-- 1      v
140 c--      | | |      | | |      | | |      |
141 c--     14--27--10     23--21--22     10--26--12     |
142 c--      | | |      | | |      | | |      |-----> x
143 c--      7--15-- 8      19--25--20      3--11-- 4
144 c--      bottom      middle      top
145 C-----
146      DIMENSION XP(1),YP(1),ZP(1),LOC(7)
147      ,,X(50),Y(50),Z(50),IEMT(2,5,6),ID(3),JD(2),STRES(300,7,6)
148      DATA IEMT / 2,19,3,14,25,16,6,18,3, 15,25,11,27,21,26 /
149      , 13,24,9, 8,20,4,16,22,12,5,17,17 /
150      WRITE (5,X) 'ENTER STRESS NO.(1,2,3,4,5,6)'
151      READ (5,X) NLT
152      READ (IOB,X) NONX,NONY,NONZ
153      READ (IOB,*) ILOC(1,1-1,NLOC)
154      READ (IOB,X) (X(I),I=1,NONX)
155      READ (IOB,X) (Y(I),I=1,NONY)
156      READ (IOB,X) (Z(I),I=1,NONZ)
157      IF (NNOBES.EQ.3) THEN
158          NONX=2*NONX-1
159          NONY=2*NONY-1
160          NONZ=2*NONZ-1
161          DO 111 I=NONX,1,-1
162              J=I/2

```



```

163      IF (MOD(I,2).NE.0) X(I)=X(J+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)=Y(J+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)=Z(J+1)
173 333    IF (MOD(I,2).EQ.0) Z(I)=(Z(I+1)+Z(J))/2.
174      END IF
175
176      WRITE (5,*) ' SELECT LEVEL (z-coord. No.) of xy-plane'
177      DO 41 I=1,NONZ
178 41     write (5,51) I,Z(I)
179 51     FORMAT (2X,I2,2X,F8.4)
180      READ (5,*) nlev
181      JS=0
182      IF (MOD(NLEV,2).NE.0) GO TO 310
183 C--    MIDDLE.....
184      JS=2
185      DO 20 J=1,3
186      DO 20 I=1,3
187      IF (LOC(1).EQ.IBMT(2,J,1)) GO TO 21
188 20     CONTINUE
189      GO TO 26
190 210    IF (NLEV.EQ.1) GO TO 23
191      JS=3
192      DO 10 J=1,3
193      DO 10 I=1,3
194      IF (LOC(1).EQ.IBMT(3,J,1)) GO TO 21
195 10     CONTINUE
196      IF (NLEV.EQ.NON2) GO TO 26
197 23     JS=1
198      DO 11 J=1,3
199      DO 11 I=1,3
200      IF (LOC(1).EQ.IBMT(1,J,1)) GO TO 21
201 11     CONTINUE
202 25     WRITE (5,*) ' .LEVEL NO. DOESN T MATCH WITH LOC.
203         NOS.
204         STOP
205 21     CONTINUE
206      DO 40 J=1,3
207      ID(J)=0
208      DO 50 I=1,3
209      DO 50 K=1,NLOC
210      IF (LOC(K).EQ.IBMT(3,I,J)) THEN
211      ID(J)=1
212      GO TO 40
213      ENDIF
214 40     CONTINUE
215 40     CONTINUE
216      DO 45 I=1,3

```

```

217      JD(I)=0
218      DO 55 J=1,3
219      DO 55 K=1,NLOC
220      IF (LOC(K).EQ.1) THEN
221      JD(I)=1
222      GO TO 45
223      ENDIF
224 55    CONTINUE
225 45    CONTINUE
226      NEX=(NOMX-1)/2
227      NX=0
228      DO 70 J=1,NOMX
229      IF (ID(1).EQ.0) GO TO 70
230      IF (ID(2).EQ.0) GO TO 70
231      IF (ID(2).EQ.0) GO TO 70
232      NX=NX+1
233      XP(NX)=X(J)
234 70    CONTINUE
235      XP(NX)=X(NOMX) !TO KEEP SIZE....
236      XP(1)=X(1) !TO KEEP SIZE....
237      NEY=(NOMY-1)/2
238      NY=0
239      DO 80 J=1,NOMY
240      IF (JD(1).EQ.0) GO TO 80
241      IF (JD(2).EQ.0) GO TO 80
242      IF (JD(2).EQ.0) GO TO 80
243      NY=NY+1
244      YP(NY)=Y(J)
245 80    CONTINUE
246      YP(NY)=Y(NOMY) !TO KEEP SIZE....
247      YP(1)=Y(1) !TO KEEP SIZE....
248      NEZ=(NOME-1)/2
249      IF (NNOBES.EQ.8) THEN
250      NEX=(NOMX+1)/2-1
251      NEY=(NOMY+1)/2-1
252      NEZ=(NOME+1)/2-1
253      END IF
254      NEXY=NEX*NEY
255      NEZ=NEX*NEZ
256 --- Reading stresses i-jel. no.; j-loc. no.
257      DO 91 I=1,NE
258      DO 91 J=1,NLOC
259      READ (100,10) (S(I,J,K),K=1,8)
260 91    CONTINUE
261 92    FORMAT (3A1,3B10,10)
262
263      NN=(NLE-1)/2
264      IF (JS.EQ.1) NN=NN+1
265      NEE=NN*NEZ
266 --- stresses of elements nnel onwards are required..
267      K=0
268      DO 90 J=1,NEZ
269      NI=NEE+NEZ*(J-1)
270      NJ=NI+NEZ

```

```

271      DO 88 JJ=1,3
272      IF (JD(JJ).EQ.0) GO TO 88
273      DO 90 I=N1,N2
274      DO 91 II=1,3
275      IF (ID(II).EQ.0) GO TO 91
276      IL=IBMT(JS,JJ,II)
277      LL=0
278      DO 92 L=1,NLOC
279 88    IF (IL.EQ.LDC(L)) LL=L
280      K=K+1
281      ZP(K)=STRES(I,LL,NST)
282      IF (II.NE.1.OR.I.EQ.N1) GO TO 91
283      IF (ID(3).EQ.0.OR.ID(1).EQ.0) GO TO 91
284      K=K-1
285      ZP(K)=(ZP(K)+ZP(K+1))/2.
286 91    CONTINUE
287 90    CONTINUE
288      IJ=K-NX+1
289      IF (JJ.NE.1.OR.J.EQ.1) GO TO 88
290      IF (JD(3).EQ.0) GO TO 88
291      K=K-2*NX
292      DO 93 KK=1,NX
293      K=K+1
294 93    ZP(K)=(ZP(K)+ZP(K+NX))/2.
295 89    continue
296 89    continue
297      do i=1,nv
298      j1=1+(i-1)*nx
299      j2=j1+nx-1
300 85    format (2x,10cell,4)
301      end do
302      RETURN
303      end

```

APPENDIX - D  
LISTING OF THE EXAMPLE RESULTS

INPUT DATA for 'PFEKROCESSOR' : man.inp

```

1 1 node El.102/9021s; delam= none load: S3S5 MESH-man.inp (10/30/8
2 4          !type of the element (8 OR 21)
3 5,3,5,0          !NO. of coord. in x,y & z dir.. Rad. of hole at node 1
4 0.0, 2.0,4.0,6.0,8.0 !x - coordinates          !Rad. =0 means : no hole
5 0.0, 3.0,6.0          !y - coordinates
6 0.0, 0.5,1.0,1.5,2.0 !z - coordinates
7 1. 300.0, 75, 1      !from node no.,node1 temp.,to node no., increm.
8 -1, 0.0, 0, 0      !data termination indicator....
9 1. 300.0, 32, 1      !from el. no., stress free temp., to el. no., increm.
10 -1, 0.0, 0, 0      !data termination indicator....
11 1. 1. 32, 1         !from element no.,mat. no.,to node no.,increm.
12 -1, 0, 0, 0        !data termination indicator....
13 1. 1. 16, 1         !from el. no.,mat.axis orient set no.,to node no.,increm.
14 17, 2, 32, 1       !from el. no.,mat.axis orient set no.,to node no.,increm.
15 -1, 0, 0, 0        !data termination indicator....
16 2,16               !from el. no., to el. no., for same stiffness
17 18,32              !from el. no., to el. no., for same stiffness
18 -1,0               !data termination indicator.. for same stiffness elements
19 0,2                !no. of nodes to simulate split, dir. normal to the plane
20 15                !no. of double nodes for Delamination Region
21 3,8,13,18,23,28,33,38,43,48,53,58,63,68,73 !double nodes
22 0.0,3.0, 0.0,6.0, 1.0,2.0 !x,y,z limits of the solid that has delam.
23 EL 1 21.0E06          !Elastic constant...
24 ET 1 1.7E06           !Elastic constant...
25 EZ 1 1.7E06           !Elastic constant...
26 RULT 1 0.3            !Elastic constant... as per elasticity notation
27 RULZ 1 0.3            !Elastic constant... as per elasticity notation
28 RUTZ 1 0.54           !Elastic constant...
29 GLT 1 0.94E06         !Elastic constant...
30 GLZ 1 0.94E06         !Elastic constant...
31 GTZ 1 0.50E06         !Elastic constant...
32 ALFL 1 0.2E-06        !Thermal expansion coefficient in L- dir.
33 ALFT 1 0.16E-04       !Thermal expansion coefficient in T- dir.
34 ALFZ 1 0.16E-04       !Thermal expansion coefficient in Z- dir.
35 -1                  !data termination indicator for Mat. constant
36 1 5 25 55           !mat. axis orientation for element set =1
37 2 25 25 5          !mat. axis orientation for element set =2
38 -1                  !data termination indicator....
39 -1                  !data termination for force b.c.
40 4 UY 0.0            24 5 !from,type,value,to,element
41 5 UY 0.             25 5 ! " " " " " "
42 1 UX 0.             51 25 ! disp. boundary condition
43 2 UX 0.             52 25
44 3 UX 0.             53 25
45 4 UX 0.             54 25
46 5 UX 0.             55 25
47 1 UZ 0.0            71 5
48 51 UY 0.006         75 1 !nonzero disp. b.c.
49 -1                  !data termination for disp. b.c.
50 21,0,0,0,0,0        !stress loc. nos.

```

OUTPUT FROM 'PREPROCESSOR' : 22/01/87

1 0 Node El.(02/902)s; delay= 1000; 0.000 0.000 0.000 (10/30/87)

2	30	2	1	0	0	0	0	0.0000	0.0000	0.0000		300.
3	10	1	0	1	1	1	1	0.0000	0.0000	0.5000	0	300.
4	2	1	0	0	1	1	1	0.0000	0.0000	1.0000	0	300.
5	3	1	0	0	1	1	1	0.0000	0.0000	1.0000	0	300.
6	4	1	0	0	1	1	1	0.0000	0.0000	1.0000	0	300.
7	5	1	1	0	1	1	1	0.0000	0.0000	1.5000	0	300.
8	6	1	1	0	1	1	1	0.0000	0.0000	2.0000	0	300.
9	7	0	0	1	1	1	1	0.0000	0.0000	0.0000	0	300.
10	8	0	0	0	1	1	1	0.0000	0.0000	0.5000	0	300.
11	9	0	0	0	1	1	1	0.0000	0.0000	1.0000	0	300.
12	10	0	0	0	1	1	1	0.0000	0.0000	1.0000	0	300.
13	11	0	1	0	1	1	1	0.0000	0.0000	1.5000	0	300.
14	12	0	1	0	1	1	1	0.0000	0.0000	2.0000	0	300.
15	13	0	0	1	1	1	1	0.0000	0.0000	0.0000	0	300.
16	14	0	0	0	1	1	1	0.0000	0.0000	0.5000	0	300.
17	15	0	0	0	1	1	1	0.0000	0.0000	1.0000	0	300.
18	16	0	0	0	1	1	1	0.0000	0.0000	1.0000	0	300.
19	17	0	1	0	1	1	1	0.0000	0.0000	1.5000	0	300.
20	18	0	1	0	1	1	1	0.0000	0.0000	2.0000	0	300.
21	19	0	0	1	1	1	1	0.0000	0.0000	0.0000	0	300.
22	20	0	0	0	1	1	1	0.0000	0.0000	0.5000	0	300.
23	21	0	0	0	1	1	1	0.0000	0.0000	1.0000	0	300.
24	22	0	0	0	1	1	1	0.0000	0.0000	1.0000	0	300.
25	23	0	1	0	1	1	1	0.0000	0.0000	1.5000	0	300.
26	24	0	1	0	1	1	1	0.0000	0.0000	2.0000	0	300.
27	25	0	0	1	1	1	1	0.0000	0.0000	0.0000	0	300.
28	26	0	0	0	1	1	1	0.0000	0.0000	0.5000	0	300.
29	27	0	0	0	1	1	1	0.0000	0.0000	1.0000	0	300.
30	28	0	0	0	1	1	1	0.0000	0.0000	1.0000	0	300.
31	29	0	1	0	1	1	1	0.0000	0.0000	1.5000	0	300.
32	30	0	1	0	1	1	1	0.0000	0.0000	2.0000	0	300.
33	31	1	0	1	1	1	1	0.0000	0.0000	0.0000	0	300.
34	32	1	0	0	1	1	1	0.0000	0.0000	0.5000	0	300.
35	33	1	0	0	1	1	1	0.0000	0.0000	1.0000	0	300.
36	34	1	0	0	1	1	1	0.0000	0.0000	1.0000	0	300.
37	36	1	0	0	1	1	1	0.0000	0.0000	2.0000	1	300.
38	37	0	0	1	1	1	1	0.0000	0.0000	0.0000	0	300.
39	38	0	0	0	1	1	1	0.0000	0.0000	0.5000	0	300.
40	39	0	0	0	1	1	1	0.0000	0.0000	1.0000	0	300.
41	40	0	0	0	1	1	1	0.0000	0.0000	1.0000	0	300.
42	42	0	0	0	1	1	1	0.0000	0.0000	2.0000	1	300.
43	43	0	0	1	1	1	1	0.0000	0.0000	0.0000	0	300.
44	44	0	0	0	1	1	1	0.0000	0.0000	0.5000	0	300.
45	45	0	0	0	1	1	1	0.0000	0.0000	1.0000	0	300.
46	46	0	0	0	1	1	1	0.0000	0.0000	1.0000	0	300.
47	48	0	0	0	1	1	1	0.0000	0.0000	2.0000	1	300.
48	49	0	0	1	1	1	1	0.0000	0.0000	0.0000	0	300.
49	50	0	0	0	1	1	1	0.0000	0.0000	0.5000	0	300.
50	51	0	0	0	1	1	1	0.0000	0.0000	1.0000	0	300.
51	52	0	0	0	1	1	1	0.0000	0.0000	1.0000	0	300.
52	54	0	0	0	1	1	1	0.0000	0.0000	2.0000	1	300.
53	55	0	0	1	1	1	1	0.0000	0.0000	0.0000	0	300.
54	56	0	0	0	1	1	1	0.0000	0.0000	0.5000	0	300.

55	57	0	0	0	1	1	1	8.0000	3.0000	1.0000	0	300.
56	58	0	0	0	1	1	1	8.0000	3.0000	1.0000	0	300.
57	60	0	0	0	1	1	1	8.0000	3.0000	2.0000	1	300.
58	61	1	0	1	1	1	1	0.0000	6.0000	0.0000	0	300.
59	62	1	0	0	1	1	1	0.0000	6.0000	0.5000	0	300.
60	63	1	0	0	1	1	1	0.0000	6.0000	1.0000	0	300.
61	64	1	0	0	1	1	1	0.0000	6.0000	1.0000	0	300.
62	66	1	0	0	1	1	1	0.0000	6.0000	2.0000	1	300.
63	67	0	0	1	1	1	1	2.0000	6.0000	0.0000	0	300.
64	68	0	0	0	1	1	1	2.0000	6.0000	0.5000	0	300.
65	69	0	0	0	1	1	1	2.0000	6.0000	1.0000	0	300.
66	70	0	0	0	1	1	1	2.0000	6.0000	1.0000	0	300.
67	72	0	0	0	1	1	1	2.0000	6.0000	2.0000	1	300.
68	73	0	0	1	1	1	1	4.0000	6.0000	0.0000	0	300.
69	74	0	0	0	1	1	1	4.0000	6.0000	0.5000	0	300.
70	75	0	0	0	1	1	1	4.0000	6.0000	1.0000	0	300.
71	76	0	0	0	1	1	1	4.0000	6.0000	1.0000	0	300.
72	78	0	0	0	1	1	1	4.0000	6.0000	2.0000	1	300.
73	79	0	0	1	1	1	1	6.0000	6.0000	0.0000	0	300.
74	80	0	0	0	1	1	1	6.0000	6.0000	0.5000	0	300.
75	81	0	0	0	1	1	1	6.0000	6.0000	1.0000	0	300.
76	82	0	0	0	1	1	1	6.0000	6.0000	1.0000	0	300.
77	84	0	0	0	1	1	1	6.0000	6.0000	2.0000	1	300.
78	85	0	0	1	1	1	1	6.0000	6.0000	0.0000	0	300.
79	86	0	0	0	1	1	1	6.0000	6.0000	0.5000	0	300.
80	87	0	0	0	1	1	1	8.0000	6.0000	1.0000	0	300.
81	88	0	0	0	1	1	1	8.0000	6.0000	1.0000	0	300.
82	90	0	0	0	1	1	1	8.0000	6.0000	2.0000	1	300.
85	90	0	0	0	1	1	1	8.0000	6.0000	2.0000	1	300.
84	91	1	2	1	7	1	0	0.0000000	0.0000000	0.100E+21		
86	92	1	2	1	7	1	0	0.0000000	0.0000000	0.100E+21		
87	93	1	2	1	7	1	0	0.0000000	0.0000000	0.100E+21		
88	95	1	2	1	7	1	0	0.0000000	0.0000000	0.100E+21		
89	96	1	2	1	7	1	0	0.0000000	0.0000000	0.100E+21		
90	97	1	2	1	7	1	0	0.0000000	0.0000000	0.100E+21		
91	99	1	2	1	7	1	0	0.0000000	0.0000000	0.100E+21		
92	99	1	2	1	7	1	0	0.0000000	0.0000000	0.100E+21		
93	99	1	2	1	7	1	0	0.0000000	0.0000000	0.100E+21		
94	99	1	2	1	7	1	0	0.0000000	0.0000000	0.100E+21		
95	99	1	2	1	7	1	0	0.0000000	0.0000000	0.100E+21		
96	99	1	2	1	7	1	0	0.0000000	0.0000000	0.100E+21		
97	99	1	2	1	7	1	0	0.0000000	0.0000000	0.100E+21		
98	99	1	2	1	7	1	0	0.0000000	0.0000000	0.100E+21		
99	99	1	2	1	7	1	0	0.0000000	0.0000000	0.100E+21		
100	99	1	2	1	7	1	0	0.0000000	0.0000000	0.100E+21		
101	80	1	2	1	7	1	0	0.0000000	0.0000000	0.100E+21		
102	81	1	2	1	7	1	0	0.0000000	0.0000000	0.100E+21		
103	93	1	2	1	7	1	0	0.0000000	0.0000000	0.100E+21		
104	94	1	2	1	7	1	0	0.0000000	0.0000000	0.100E+21		
105	85	1	2	1	7	1	0	0.0000000	0.0000000	0.100E+21		
106	86	1	2	1	7	1	0	0.0000000	0.0000000	0.100E+21		
107	87	1	2	1	7	1	0	0.0000000	0.0000000	0.100E+21		
108	88	1	2	1	7	1	0	0.0000000	0.0000000	0.100E+21		

105	90	1	2	1	7	1	0	0.0000000	0.0000000	0.100E+21
110	64	1	2	1	7	1	0	0.0000000	0.0000000	0.100E+21
111	70	1	2	1	7	1	0	0.0000000	0.0000000	0.100E+21
112	76	1	2	1	7	1	0	0.0000000	0.0000000	0.100E+21
113	82	1	2	1	7	1	0	0.0000000	0.0000000	0.100E+21
114	68	1	2	1	7	1	0	0.0000000	0.0000000	0.100E+21
115	8	32	1	0	2	0	0	0.0000000	0.0000000	0.100E+21
116	1	1								
LAYER1==0-LAYER; LAYER2==90-LAYER.										
117	300.	21000000.	1700000.	1000000.	0.0243	0.0243	0.5400			
118	940000.	940000.	500000.	11000000	0.0000150	0.0000150				
119	1	6	30	66						
120	2	30	90	6						
121	31	0	0	0	0	0	0			
122										
123										
124										
125	0.									
126										
127	1			1	1	1	300.	0		0
128	38	32	2	8	37	31	1	0		
129	2			1	1	1	100.	0		1
130	44	38	8	14	43	31	7	13		
131	3			1	1	1	100.	0		1
132	50	44	14	20	49	48	13	19		
133	4			1	1	1	100.	12		1
134	56	50	20	26	55	49	19	23		
135	5			1	1	1	100.	0		1
136	68	62	32	38	67	61	31	37		
137	6			1	1	1	100.	0		1
138	74	68	38	44	73	67	37	43		
139	7			1	1	1	100.	0		1
140	60	74	44	50	79	73	43	49		
141	8			1	1	1	100.	0		1
142	86	80	50	56	85	79	49	55		
143	9			1	1	1	100.	0		1
144	39	33	3	9	38	32	2	7		
145	10			1	1	1	100.	0		1
146	45	39	9	15	44	38	6	14		
147	11			1	1	1	100.	0		1
148	51	45	15	21	50	44	14	20		
149	12			1	1	1	100.	12		1
150	57	51	21	27	56	50	20	26		
151	13			1	1	1	100.	0		1
152	69	63	33	39	68	62	32	38		
153	14			1	1	1	100.	0		1
154	75	69	39	45	74	68	38	44		
155	15			1	1	1	100.	0		1
156	31	75	45	51	80	74	48	50		
157	16			1	1	1	100.	0		1
158	87	81	51	57	86	80	50	56		
159	17			1	1	1	100.	0		0
160	41	35	5	11	40	34	4	10		
161	18			1	1	1	100.	0		1
162	47	41	11	17	46	40	11	16		



163	19			1	2	1	300.	0		1
164	53	47	17	23	52	46	16	22		
165	20			1	2	1	300.	10		1
166	59	53	23	29	58	52	22	38		
167	21			1	2	1	300.	0		1
168	71	65	35	41	70	64	34	40		
169	22			1	2	1	300.	0		1
170	77	71	41	47	76	70	40	46		
171	23			1	2	1	300.	0		1
172	83	77	47	53	82	76	48	52		
173	24			1	2	1	300.	0		1
174	89	83	53	59	88	82	52	58		
175	25			1	2	1	300.	0		1
176	42	36	6	12	41	35	8	11		
177	26			1	2	1	300.	0		1
178	48	42	12	18	47	41	11	17		
179	27			1	2	1	300.	0		1
180	54	48	18	24	53	47	17	23		
181	28			1	2	1	300.	10		1
182	60	54	24	30	59	53	23	29		
183	29			1	2	1	300.	0		1
184	72	66	36	42	71	65	35	41		
185	30			1	2	1	300.	0		1
186	73	72	42	48	72	71	41	47		
187	31			1	2	1	300.	6		1
188	84	78	48	54	83	77	47	53		
189	32			1	2	1	300.	0		1
190	90	84	54	60	89	83	53	59		
191										
192		1.								
193	10									
194	0	2	0.0							
195	4	2	0.0							
196	0	2	0.0							
197	10	2	0.0							
198	15	2	0.0							
199	15	2	0.0							
200	21	2	0.0							
201	22	2	0.0							
202	27	2	0.0							
203	28	2	0.0							
204	15									
205	0	4	1 0 1							
206	0	10	1 0 1							
207	15	16	1 0 1							13
208	21	22	1 0 1							18
209	27	28	1 0 1							23
210	33	34	1 1 1							28
211	39	40	1 1 1							33
212	45	46	1 1 1							38
213	51	52	1 1 1							43
214	57	58	1 1 1							48
215	63	64	1 1 1							53
216	69	70	1 1 1							58

217	75	76	1 1 1	!	63
218	81	82	1 1 1	!	68
219	87	88	1 1 1	!	73
220	0	0	0		
221	1	32			
222	27	28	1	!	23
223	27	28	2	!	23
224	27	28	3	!	23
225	0	0	0		
226	1	32			
227	31	32	1	!	13
228	31	32	2	!	13
229	31	32	3	!	13
230	57	58	1	!	43
231	57	58	2	!	43
232	57	58	3	!	43
233	0	0	0		
234	1	32			
235	5999	5999	0		
236					

MODIFICATIONS FOR CRACK OPENING SEQUENCE

1	10					
2	3	2	0.0			
3	4	2	0.0			
4	9	2	0.0			
5	10	2	0.0			
6	15	2	0.0			
7	16	2	0.0			
8	21	2	0.0			
9	22	2	0.6			
10	27	2	0.0			
11	28	2	0.0			
12	15					
13	3	4	1 0 1	!		3
14	9	10	1 0 1	!		3
15	15	16	1 0 1	!		13
16	21	22	1 0 1	!		18
17	27	28	1 0 1	!		23
18	33	34	1 1 1	!		28
19	39	40	1 1 1	!		33
20	45	46	1 1 1	!		38
21	51	52	1 1 1	!		43
22	57	58	1 1 1	!		48
23	63	64	1 1 1	!		53
24	69	70	1 1 1	!		58
25	75	76	1 1 1	!		63
26	81	82	1 1 1	!		68
27	87	88	1 1 1	!		73
28	0	0	0			
29	1	32				
30	27	28	1	!		23
31	27	28	2	!		23
32	27	28	3	!		23
33	0	0	0			
34	1	32				
35	21	22	1	!		18
36	21	22	2	!		16
37	21	22	3	!		18
38	57	58	1	!		48
39	57	58	2	!		48
40	57	58	3	!		48
41	0	0	0			
42	1	32				
43	9999	9999	0			
44						

OUTPUT FROM KRAP II : 01/17/87

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27

8 NODE EL102/9020a; delimit node 1001; 8002 mesh-wal.inp (10/30/87)

CONTROL INFORMATION

NUMBER OF NODAL POINTS = 67  
 NUMBER OF ELEMENT TYPES = 1  
 NUMBER OF LOAD CASES = 4  
 NUMBER OF FREQUENCIES = 0  
 ANALYSIS CODE (NDYN) = 0  
 EQ.0, STATIC  
 EQ.1, MODAL EXTRACTION  
 EQ.2, FORCED RESPONSE  
 EQ.3, RESPONSE SPECTRUM  
 EQ.4, DIRECT INTEGRATION  
 SOLUTION MODE (MODEX) = 0  
 EQ.0, EXECUTION  
 EQ.1, DATA CHECK  
 NUMBER OF SUBSPACE  
 ITERATION VECTORS (IND) = 0  
 EQUATIONS PER BLOCK = 0  
 TAPEIO SAVE FLAG (NIOSV) = 0

MODAL POINT INPUT DATA

28	BOUNDARY CONDITION CODES				MODAL POINT COORDINATES	
29	NUMBER	X	Y	Z	XX	YY
30	18	0	0	0	0	0

ELEMENT TYPE = 7,  
 NUMBER OF ELEMENTS = 30

ELEMENT LOAD CASE MULTIPLIERS

41	CASE(A)	CASE(B)	CASE(C)	CASE(D)
42	1.0000	0.0000	0.0000	0.0000

46	ELEMENT	NODE	NODES DEFINING CONSTRAINT DIRECTION		CODE	CODE	GENERATION
47	NUMBER	(N)	(N1)	(N2)	(N1)	(N2)	CODE (KN)
49	1	61	1	2	7	1	0
50	2	62	1	2	7	1	0
51	3	63	1	2	7	1	0
52	4	65	1	2	7	1	0
53	5	66	1	2	7	1	0
54	6	67	1	2	7	1	0

55	7	68	1	2	1	7	1	0	0
56	8	69	1	2	1	7	1	0	0
57	9	71	1	2	1	7	1	0	0
58	10	72	1	2	1	7	1	0	0
59	11	73	1	2	1	7	1	0	0
60	12	74	1	2	1	7	1	0	0
61	13	75	1	2	1	7	1	0	0
62	14	77	1	2	1	7	1	0	0
63	15	78	1	2	1	7	1	0	0
64	16	79	1	2	1	7	1	0	0
65	17	80	1	2	1	7	1	0	0
66	18	81	1	2	1	7	1	0	0
67	19	83	1	2	1	7	1	0	0
68	20	84	1	2	1	7	1	0	0
69	21	85	1	2	1	7	1	0	0
70	22	86	1	2	1	7	1	0	0
71	23	87	1	2	1	7	1	0	0
72	24	89	1	2	1	7	1	0	0
73	25	90	1	2	1	7	1	0	0
74	26	64	1	2	1	7	1	0	0
75	27	70	1	2	1	7	1	0	0
76	28	76	1	2	1	7	1	0	0
77	29	82	1	2	1	7	1	0	0
78	30	88	1	2	1	7	1	0	0

79 21-NODE SOLID ELEMENT INPUT DATA

80

81 CONTROL INFORMATION

82

83 NUMBER OF 21-NODE ELEMENTS = 32

84

85 NUMBER OF MATERIAL SETS = 1

86

87 MAXIMUM NUMBER OF MATERIAL

88 TEMPERATURE INPUT POINTS = 1

89

90 NUMBER OF MATERIAL

91 AXIS ORIENTATION SETS = 2

92

93 NUMBER OF DISTRIBUTED LOAD SETS = 0

94

95 MAXIMUM NUMBER OF ELEMENT NODES = 3

96

97 NUMBER OF STRESS OUTPUT SETS = 1

98

99 R,S COORDINATE INTEGRATION ORDER = 2

100

101 T COORDINATE INTEGRATION ORDER = 2

102

103

104

105

106 MATERIAL PROPERTY TABLES

107

108

```

109 CHARACTER NUMBER = ( 1 )
110 NUMBER OF
111 TEMPERATURE POINTS = ( 1 )
112 WEIGHT DENSITY = ( 0.00000000 )
113 MASS DENSITY = ( 0.00000000 )
114 IDENTIFICATION = (MATERIAL=0-LAYER=0-MATERIAL= )
115
116 TEMPERATURE E11 E12 E13 V12 V13 V23 G12
117
118 500.00 21000000.0 1.000000 1.000000 0.024 0.024 0.540 940000.0
119
120

```

121 MATERIAL AXIS CORRELATION TABLE

```

122
123 SET NODE NODE NODE
124 NUMBER NI NJ NI
125
126 1 6 30 30
127 2 30 90 90
128
129

```

130 STRESS OUTPUT REQUEST TABLE

```

131
132 SET POINT POINT POINT POINT POINT POINT POINT
133 NUMBER (1) (2) (3) (4) (5) (6) (7)
134
135 1 21 0 0 0 0 0
136
137

```

138 ELEMENT LOAD CASE MULTIPLIERS

```

139
140
141 CASE A CASE B CASE C CASE D
142
143 X-DIRECTION GRAVITY = 0.00 0.00 0.00 0.00
144 Y-DIRECTION GRAVITY = 0.00 0.00 0.00 0.00
145 Z-DIRECTION GRAVITY = 0.00 0.00 0.00 0.00
146 THERMAL LOADING = 0.00 0.00 0.00 0.00
147 PRESSURE LOADING = 0.00 0.00 0.00 0.00
148
149

```

150 B Y D 2 1 NODES SELECTED ELEMENT DATA

```

151
152 ELEMENT NODES NODES MATL. MATL. MATL. STRESS NODE GAUSS GAUSS
153 NUMBER -NUS- -XYZ- INTC. NUS. OUTPUT FREE INC. PTS. PTS. MATR
154 NO. SET SET TEMP. -NC- -R,S- -I- NL-U
155 1 8 8 1 1 1 500.0 1 2 0
156
157 2 8 8 1 1 1 500.0 6 2 0
158
159 3 8 8 1 1 1 500.0 6 2 0
160
161 4 8 8 1 1 1 500.0 12 2 0
162

```

163	5	3	8	1	1	1	300.0	6	2	0
164										
165	6	8	8	1	1	1	300.0	6	2	0
166										
167	7	8	8	1	1	1	300.0	6	2	0
168										
169	8	6	8	1	1	1	300.0	1	2	0
170										
171	9	8	8	1	1	1	300.0	6	2	0
172										
173	10	3	8	1	1	1	300.0	6	2	0
174										
175	11	8	8	1	1	1	300.0	6	2	0
176										
177	12	8	8	1	1	1	300.0	12	2	0
178										
179	13	8	8	1	1	1	300.0	6	2	0
180										
181	14	8	8	1	1	1	300.0	6	2	0
182										
183	15	8	8	1	1	1	300.0	6	2	0
184										
185	16	8	8	1	1	1	300.0	1	2	0
186										
187	17	8	8	1	2	1	300.0	1	2	0
188										
189	18	8	8	1	2	1	300.0	6	2	0
190										
191	19	8	8	1	2	1	300.0	6	2	0
192										
193	20	8	8	1	2	1	300.0	12	2	0
194										
195	21	8	8	1	2	1	300.0	6	2	0
196										
197	22	8	8	1	2	1	300.0	6	2	0
198										
199	23	8	8	1	2	1	300.0	6	2	0
200										
201	24	8	8	1	2	1	300.0	6	2	0
202										
203	25	8	8	1	2	1	300.0	6	2	0
204										
205	26	8	8	1	2	1	300.0	6	2	0
206										
207	27	8	8	1	2	1	300.0	6	2	0
208										
209	28	8	8	1	2	1	300.0	12	2	0
210										
211	29	8	8	1	2	1	300.0	6	2	0
212										
213	30	8	8	1	2	1	300.0	6	2	0
214										
215	31	8	8	1	2	1	300.0	6	2	0
216										





271 ---- NODAL DISPLACEMENTS AND FORCES IN COLEU----

272 \*\*\*\*\*

273

274

275

276

277

NODE	U	V	W	Fx	Fy	Fz
278	1 0.00000E+00	0.11463E-02	0.00000E+00	0.00000E+00	0.54179E-13	0.00000E+00
279	2 0.00000E+00	0.66914E-03	-0.23028E-03	0.00000E+00	0.31974E-13	-0.21316E-13
280	3 0.00000E+00	0.00000E+00	-0.48447E-03	0.00000E+00	-1254.5	-12.449
281	4 0.00000E+00	0.00000E+00	-0.48447E-03	0.00000E+00	-5360.6	12.449
282	5 0.00000E+00	0.00000E+00	-0.70911E-03	0.00000E+00	0.00000E+00	-0.99476E-13
283	6 0.00000E+00	0.00000E+00	-0.92034E-03	0.00000E+00	0.00000E+00	-0.99476E-13
284	7-0.73276E-04	0.11457E-02	0.00000E+00	-0.12434E-13	0.25892E-12	0.00000E+00
285	8-0.75126E-04	0.86872E-03	-0.23072E-03	0.53391E-13	-0.92371E-13	-0.92371E-13
286	9-0.78116E-04	0.00000E+00	-0.48515E-03	-14.750	-2505.8	-27.415
287	10-0.78116E-04	0.00000E+00	-0.48515E-03	14.750	-10721.	27.415
288	11-0.80760E-04	0.00000E+00	-0.70685E-03	0.49738E-13	0.00000E+00	-0.56343E-13
289	12-0.82216E-04	0.00000E+00	-0.92999E-03	-0.71054E-14	0.00000E+00	-0.85265E-13
290	13-0.13991E-03	0.11383E-02	0.00000E+00	-0.29310E-13	0.30509E-12	0.00000E+00
291	14-0.14765E-03	0.98319E-03	-0.23431E-03	0.88818E-13	-0.87041E-13	-0.37659E-12
292	15-0.15902E-03	0.00000E+00	-0.43065E-03	-74.970	-2492.4	-64.931
293	16-0.15902E-03	0.00000E+00	-0.43065E-03	74.970	-10718.	64.931
294	17-0.17076E-03	0.00000E+00	-0.71501E-03	-0.71054E-14	0.00000E+00	-0.14211E-13
295	18-0.16825E-03	0.00000E+00	-0.93503E-03	0.71054E-14	0.00000E+00	-0.54001E-12
296	19-0.18363E-03	0.11306E-02	0.00000E+00	0.10658E-13	0.78160E-13	0.00000E+00
297	20-0.20636E-03	0.88057E-03	-0.25799E-03	-0.65725E-13	0.18296E-12	0.00000E+00
298	21-0.25376E-03	0.00000E+00	-0.53085E-03	-224.03	-2509.2	-90.916
299	22-0.25376E-03	0.00000E+00	-0.53085E-03	224.03	-10719.	90.916
300	23-0.30927E-03	0.00000E+00	-0.74951E-03	-0.10658E-12	0.00000E+00	0.55422E-12
301	24-0.35406E-03	0.00000E+00	-0.94472E-03	-0.39080E-13	0.00000E+00	-0.22737E-12
302	25-0.20854E-03	0.12250E-02	0.00000E+00	0.16376E-13	-0.55067E-13	0.00000E+00
303	26-0.23900E-03	0.94963E-03	-0.17625E-03	0.35527E-14	0.81712E-13	-0.28422E-13
304	27-0.37387E-03	0.00000E+00	-0.39000E-03	-339.31	-1305.8	104.43
305	28-0.37287E-03	0.00000E+00	-0.39000E-03	339.81	-5368.9	-104.43
306	29-0.55838E-03	0.00000E+00	-0.57126E-03	-0.97700E-13	0.00000E+00	0.22737E-12
307	30-0.77311E-03	0.00000E+00	-0.74140E-03	-0.46185E-13	0.00000E+00	-0.14211E-12
308	31 0.00000E+00	0.30734E-02	0.00000E+00	0.00000E+00	0.23093E-13	0.00000E+00
309	32 0.00000E+00	0.30562E-02	-0.18195E-03	0.00000E+00	0.17941E-12	-0.16342E-12
310	33 0.00000E+00	0.30773E-02	-0.41630E-03	0.00000E+00	-444.89	-53.583
311	34 0.00000E+00	0.30773E-02	-0.41630E-03	0.00000E+00	444.89	53.583
312	35 0.00000E+00	0.30191E-02	-0.64568E-03	0.00000E+00	0.17935E-11	0.11369E-12
313	36 0.00000E+00	0.30104E-02	-0.86908E-03	0.00000E+00	-0.11990E-12	-0.24158E-12
314	37-0.84036E-04	0.30738E-02	0.00000E+00	0.71054E-14	0.41656E-12	0.00000E+00
315	38-0.85401E-04	0.30559E-02	-0.18028E-03	0.74607E-13	-0.52935E-12	-0.41922E-12
316	39-0.87090E-04	0.30771E-02	-0.41734E-03	-19.334	-888.27	-112.18
317	40-0.87090E-04	0.30771E-02	-0.41734E-03	19.334	888.27	112.18
318	41-0.87638E-04	0.30190E-02	-0.64533E-03	-0.36818E-13	0.18510E-11	0.17053E-12
319	42-0.88506E-04	0.30104E-02	-0.86809E-03	-0.71054E-14	0.11431E-11	-0.52580E-12
320	43-0.16394E-03	0.30731E-02	0.00000E+00	-0.29527E-13	0.29488E-12	0.00000E+00
321	44-0.17014E-03	0.30560E-02	-0.18640E-03	0.11724E-12	-0.31442E-12	-0.83844E-12
322	45-0.17822E-03	0.30769E-02	-0.42031E-03	-125.37	-835.34	-186.00
323	46-0.17822E-03	0.30769E-02	-0.42031E-03	125.37	885.34	186.00
324	47-0.18532E-03	0.30190E-02	-0.65904E-03	0.31974E-13	-0.91357E-12	0.15632E-12

325	48-0.18212E-03	0.30165E-02	0.30165E-02	0.158020E-13	0.22280E-11	-0.63949E-12
326	49-0.22847E-03	0.30765E-02	0.30765E-02	0.15721E-12	0.98588E-13	0.00000E+00
327	50-0.24583E-03	0.30089E-02	0.30089E-02	0.20466E-12	-0.43165E-12	-0.20606E-12
328	51-0.28415E-03	0.30775E-02	0.45075E-03	-408.01	-900.19	-243.72
329	52-0.28415E-03	0.30775E-02	0.45075E-03	408.01	900.19	243.72
330	53-0.33013E-03	0.30190E-02	0.10004E-03	0.14931E-12	0.13074E-11	0.48317E-12
331	54-0.37055E-03	0.30099E-02	0.07690E-03	0.14211E-12	0.15103E-11	-0.44054E-12
332	55-0.26533E-03	0.30814E-02	0.00000E+00	0.10630E-12	-0.88818E-15	0.00000E+00
333	56-0.30213E-03	0.30695E-02	0.13292E-03	0.21054E-14	0.12923E-12	0.48850E-13
334	57-0.40790E-03	0.30051E-02	0.32047E-03	-741.90	-469.95	173.82
335	58-0.40790E-03	0.30051E-02	0.32047E-03	741.90	469.95	-173.82
336	59-0.67171E-03	0.30222E-02	0.21392E-03	0.70166E-13	0.15967E-11	0.14744E-12
337	60-0.78089E-03	0.30100E-02	0.08772E-03	0.12312E-12	0.25624E-12	-0.16375E-12
338	61	0.00000E+00	0.00000E+00	0.00000E+00	-16.000	0.00000E+00
339	62	0.00000E+00	0.00000E+00	0.00000E+00	32.000	-0.85265E-13
340	63	0.00000E+00	0.00000E+00	0.00000E+00	-2416.0	63.375
341	64	0.00000E+00	0.00000E+00	0.00000E+00	2416.0	-63.375
342	65	0.00000E+00	0.00000E+00	0.00000E+00	-16.000	0.39790E-12
343	66	0.00000E+00	0.00000E+00	0.00000E+00	32.000	-0.56843E-13
344	67-0.86973E-04	0.60000E-02	0.00000E+00	0.14211E-13	32.000	0.00000E+00
345	68-0.90275E-04	0.60000E-02	0.00000E+00	0.10658E-13	80.000	-0.29843E-12
346	69-0.91885E-04	0.60000E-02	0.54214E-03	-7.6169	-4848.0	124.28
347	70-0.91885E-04	0.60000E-02	0.54214E-03	7.6169	4848.0	-124.28
348	71-0.92281E-04	0.60000E-02	0.74309E-03	-0.39080E-13	48.000	-0.71054E-13
349	72-0.92165E-04	0.60000E-02	0.95005E-03	0.42633E-13	16.000	-0.56843E-12
350	73-0.17348E-03	0.60000E-02	0.00000E+00	0.09278E-13	0.00000E+00	0.00000E+00
351	74-0.17961E-03	0.60000E-02	0.20722E-03	0.30198E-13	32.000	-0.32552E-12
352	75-0.18760E-03	0.60000E-02	0.54005E-03	-58.135	-4928.0	87.420
353	76-0.18760E-03	0.60000E-02	0.54005E-03	58.135	4916.0	-87.420
354	77-0.19428E-03	0.60000E-02	0.74810E-03	0.12479E-12	32.000	0.10747E-12
355	78-0.19082E-03	0.60000E-02	0.90151E-03	0.10214E-13	-32.000	0.13189E-12
356	79-0.24146E-03	0.60000E-02	0.00000E+00	0.14368E-12	32.000	0.00000E+00
357	80-0.25882E-03	0.60000E-02	0.00000E+00	0.00396E-13	32.000	-0.15610E-12
358	81-0.29749E-03	0.60000E-02	0.50050E-03	-200.15	-4848.0	58.284
359	82-0.29749E-03	0.60000E-02	0.50050E-03	200.15	4848.0	-58.284
360	83-0.34226E-03	0.60000E-02	0.77819E-03	0.40650E-13	16.000	-0.13569E-12
361	84-0.38280E-03	0.60000E-02	0.90023E-03	0.55955E-13	48.000	0.41478E-12
362	85-0.30196E-03	0.60000E-02	0.00000E+00	0.10658E-13	32.000	0.00000E+00
363	86-0.31903E-03	0.60000E-02	0.24017E-03	0.03949E-13	48.000	-0.14211E-12
364	87-0.42430E-03	0.60000E-02	0.45004E-03	-374.35	-2432.0	179.58
365	88-0.42430E-03	0.60000E-02	0.45004E-03	374.35	2432.0	-179.58
366	89-0.69559E-03	0.60000E-02	0.00000E+00	0.00343E-13	-16.000	0.86818E-13
367	90-0.80112E-03	0.60000E-02	0.76230E-03	0.00396E-13	-32.000	0.12079E-12

---

370			
371	27	28	1
372	27	28	2
373	27	28	3
374	0	0	0
375	1	STEP	1
376			
377			
378			



433	47-0.18275E-03	0.30000E-02	-0.07415E-03	-0.00205E-13	0.93792E-12	-0.63949E-12
434	46-0.17762E-03	0.30000E-02	-0.08297E-03	-0.11569E-12	0.23170E-11	0.36948E-12
435	49-0.24254E-03	0.32165E-02	-0.00000E+00	-0.13923E-13	-0.13145E-12	0.00000E+00
436	50-0.26493E-03	0.31994E-02	-0.10154E-03	-0.23981E-12	0.40945E-12	-0.99476E-13
437	51-0.32906E-03	0.31915E-02	-0.44950E-03	-0.00000E+00	-808.58	-347.65
438	52-0.32906E-03	0.31915E-02	-0.44950E-03	-0.00000E+00	808.58	347.65
439	53-0.36051E-03	0.30619E-02	-0.07692E-03	-0.19240E-13	0.20961E-12	0.24158E-12
440	54-0.37399E-03	0.29663E-02	-0.37057E-03	-0.10345E-14	0.18332E-11	-0.36948E-12
441	55-0.30339E-03	0.33490E-02	-0.00000E+00	-0.13076E-12	0.34417E-13	0.00000E+00
442	56-0.32779E-03	0.33772E-02	-0.01875E-04	-0.19302E-12	-0.65281E-13	0.18874E-12
443	57-0.45422E-03	0.36043E-02	-0.25509E-03	-0.00000E+00	-79.197	368.55
444	58-0.45422E-03	0.36043E-02	-0.25509E-03	-0.00000E+00	79.197	-368.55
445	59-0.67718E-03	0.35075E-02	-0.41501E-03	-0.13095E-12	-0.12574E-11	-0.23093E-12
446	60-0.72348E-03	0.32694E-02	-0.60040E-03	-0.42166E-14	0.13501E-11	-0.50981E-12
447	61	0.00000E+00	0.60000E-02	-0.00000E+00	0.00000E+00	-16.000
448	62	0.00000E+00	0.60000E-02	-0.00000E+00	0.00000E+00	32.000
449	63	0.00000E+00	0.60000E-02	-0.00000E+00	0.00000E+00	-2416.0
450	64	0.00000E+00	0.60000E-02	-0.00000E+00	0.00000E+00	2416.0
451	65	0.00000E+00	0.60000E-02	-0.00000E+00	0.00000E+00	-16.000
452	66	0.00000E+00	0.60000E-02	-0.00000E+00	0.00000E+00	32.000
453	67-0.10239E-03	0.60000E-02	-0.00000E+00	0.71054E-14	32.000	0.00000E+00
454	68-0.10480E-03	0.60000E-02	-0.26521E-03	-0.63949E-13	48.000	-0.34106E-12
455	69-0.10772E-03	0.60000E-02	-0.54300E-03	-4.8732	-4864.0	121.46
456	70-0.10772E-03	0.60000E-02	-0.54300E-03	-4.8732	4896.0	-121.46
457	71-0.10817E-03	0.60000E-02	-0.74230E-03	0.24369E-13	16.000	-0.12221E-11
458	72-0.10861E-03	0.60000E-02	-0.95260E-03	-0.42633E-13	48.000	0.14211E-12
459	73-0.19891E-03	0.60000E-02	-0.00000E+00	0.31974E-13	-32.000	0.00000E+00
460	74-0.20703E-03	0.60000E-02	-0.28553E-03	0.51514E-13	64.000	-0.24514E-12
461	75-0.22297E-03	0.60000E-02	-0.54440E-03	-59.615	-4928.0	100.92
462	76-0.22297E-03	0.60000E-02	-0.54440E-03	-59.615	4832.0	-100.92
463	77-0.24100E-03	0.60000E-02	-0.74532E-03	-0.49513E-13	64.000	-0.82645E-12
464	78-0.24622E-03	0.60000E-02	-0.95450E-03	0.73375E-13	-16.000	0.41256E-12
465	79-0.27400E-03	0.60000E-02	-0.00000E+00	0.55307E-13	-16.000	0.00000E+00
466	80-0.29576E-03	0.60000E-02	-0.27050E-03	-0.15145E-12	64.000	-0.77272E-13
467	81-0.34303E-03	0.60000E-02	-0.52275E-03	-221.86	-4640.0	206.30
468	82-0.34303E-03	0.60000E-02	-0.52275E-03	-221.86	4688.0	-206.30
469	83-0.42417E-03	0.60000E-02	-0.70000E-03	-0.15099E-13	80.000	0.33573E-12
470	84-0.50061E-03	0.60000E-02	-0.87791E-03	-0.41744E-13	16.000	0.22826E-12
471	85-0.33558E-03	0.60000E-02	-0.00000E+00	0.03949E-13	16.000	0.00000E+00
472	86-0.35725E-03	0.60000E-02	-0.18971E-03	-0.10236E-12	0.00000E+00	-0.29310E-13
473	87-0.47481E-03	0.60000E-02	-0.32297E-03	-593.50	-2096.0	354.50
474	88-0.47481E-03	0.60000E-02	-0.32297E-03	-593.50	2112.0	-354.50
475	89-0.31700E-03	0.60000E-02	-0.03231E-03	-0.21310E-13	0.00000E+00	0.10658E-12
476	90-0.10120E-02	0.60000E-02	-0.42711E-03	-0.40850E-13	16.000	0.26645E-13

---

480	21	22	1
481	21	22	2
482	21	22	3
483	57	58	1
484	57	58	2
485	57	58	3
486	0	0	0

487 1 \*\*\*\*\* STEP # 2 \*\*\*\*\*

488

489

490

491

492

493

494

495

496

497

498

499

500

501

502

503

504

505

506

507

508

509

510

511

512

513

514

515

516

517

518

519

520

521

522

523

524

525

526

527

528

529

530

531

532

533

534

535

536

537

538

539

540

\*\*\*\*\*  
---- NODAL DISPLACEMENTS AND FORCES IN SOLEO ---  
\*\*\*\*\*

NODE	U	V	W	Fx	Fy	Fz
1	0.00000E+00	0.11479E-02	0.00000E+00	0.00000E+00	0.47519E-13	0.00000E+00
2	0.00000E+00	0.86565E-03	-0.22519E-03	0.00000E+00	-0.17764E-12	-0.11369E-12
3	0.00000E+00	0.00000E+00	-0.48940E-03	0.00000E+00	-1257.5	19.659
4	0.00000E+00	0.00000E+00	-0.48940E-03	0.00000E+00	-5347.2	-19.659
5	0.00000E+00	0.00000E+00	-0.72839E-03	0.00000E+00	0.00000E+00	-0.39476E-13
6	0.00000E+00	0.00000E+00	-0.95189E-03	0.00000E+00	0.00000E+00	-0.11369E-12
7	-0.43660E-04	0.11561E-02	0.00000E+00	0.14211E-13	0.11102E-12	0.00000E+00
8	-0.51390E-04	0.89754E-03	-0.21430E-03	0.35527E-14	0.22560E-12	-0.16342E-12
9	-0.26748E-04	0.00000E+00	-0.43272E-03	95.531	-2546.6	77.612
10	-0.26748E-04	0.00000E+00	-0.43272E-03	-95.531	-10733.	-77.612
11	0.24181E-04	0.00000E+00	-0.64642E-03	-0.49738E-13	0.00000E+00	0.85265E-13
12	0.17142E-04	0.00000E+00	-0.90242E-03	-0.10658E-13	0.00000E+00	-0.56843E-13
13	-0.66606E-04	0.11869E-02	0.00000E+00	-0.91432E-13	0.58620E-13	0.00000E+00
14	-0.66621E-04	0.91686E-03	-0.34559E-03	0.00000E+00	0.17764E-14	-0.19185E-12
15	-0.14033E-03	0.00000E+00	-0.72966E-03	-606.80	-2564.5	-666.75
16	-0.14033E-03	0.00000E+00	-0.72966E-03	606.80	-9015.7	666.75
17	0.17186E-03	0.00000E+00	-0.10019E-02	-0.17408E-12	0.00000E+00	-0.48317E-12
18	0.67653E-03	0.00000E+00	-0.12980E-02	-0.13500E-12	0.00000E+00	0.21316E-12
19	-0.84855E-04	0.13739E-02	0.00000E+00	0.39080E-13	0.36415E-12	0.00000E+00
20	-0.96202E-04	0.10742E-02	-0.25259E-03	-0.17764E-13	-0.22915E-12	0.24158E-12
21	-0.70549E-04	0.00000E+00	-0.51319E-03	-0.29310E-13	-2778.4	-0.10658E-12
22	-0.13701E-02	0.22522E-02	-0.57535E-02	0.14211E-12	0.31974E-12	-0.10232E-11
23	-0.22839E-03	0.00000E+00	-0.56503E-02	-0.55422E-12	0.00000E+00	-0.26432E-11
24	-0.59570E-03	0.00000E+00	-0.57034E-02	-0.71054E-14	0.00000E+00	0.36380E-11
25	-0.11507E-03	0.14984E-02	0.00000E+00	0.29921E-13	0.47962E-13	0.00000E+00
26	-0.12385E-02	0.11694E-02	-0.27357E-03	-0.39706E-13	0.27303E-13	0.11269E-12
27	-0.15967E-03	0.00000E+00	-0.60236E-03	0.10125E-12	-1498.4	-0.71054E-13
28	-0.16199E-02	0.28741E-02	-0.88058E-02	0.67502E-13	0.29843E-12	0.68212E-12
29	-0.10713E-02	0.00000E+00	-0.89213E-02	0.14211E-13	0.00000E+00	-0.41496E-11
30	-0.29968E-03	0.30090E+00	-0.89224E-02	0.39790E-12	0.00000E+00	0.30127E-11
31	0.00000E+00	0.30553E-02	0.00000E+00	0.00000E+00	0.26823E-12	0.00000E+00
32	0.00000E+00	0.30423E-02	-0.17931E-03	0.00000E+00	-0.17941E-12	-0.49027E-12
33	0.00000E+00	0.30750E-02	-0.41112E-03	0.00000E+00	-433.39	-31.994
34	0.00000E+00	0.30750E-02	-0.41112E-03	0.00000E+00	433.39	31.994
35	0.00000E+00	0.30189E-02	-0.64004E-03	0.00000E+00	0.21174E-11	0.42633E-13
36	0.00000E+00	0.30085E-02	-0.86547E-03	0.00000E+00	0.21272E-12	-0.32685E-12
37	-0.93072E-04	0.30743E-02	0.00000E+00	0.39080E-13	-0.24336E-12	0.00000E+00
38	-0.97256E-04	0.30510E-02	-0.18087E-03	0.10481E-12	0.44764E-12	-0.67502E-12
39	-0.96425E-04	0.30616E-02	-0.41379E-03	79.605	-958.20	-46.563
40	-0.96425E-04	0.30616E-02	-0.41379E-03	-79.605	958.20	46.563
41	-0.83153E-04	0.30059E-02	-0.63399E-03	-0.85265E-13	0.14992E-11	-0.18474E-12
42	-0.77701E-04	0.30000E-02	-0.86052E-03	-0.15987E-13	0.13709E-11	-0.46896E-12

541	43-0.17673E-03	0.32280E-03	0.00000E+00	0.15731E-12	-0.25580E-12	0.00000E+00
542	44-0.19037E-03	0.32062E-02	-0.15322E-03	0.15820E-13	0.59686E-13	-0.41922E-12
543	45-0.22271E-03	0.31866E-02	-0.42200E-03	-579.02	-819.24	-408.11
544	46-0.22271E-03	0.31866E-02	-0.42200E-03	579.02	819.24	408.11
545	47-0.20486E-03	0.30612E-02	-0.54854E-03	-0.10303E-12	-0.44942E-12	-0.59686E-12
546	48-0.18848E-03	0.29922E-02	-0.50920E-03	0.35527E-14	0.32807E-11	0.15632E-12
547	49-0.24670E-03	0.33942E-02	0.00000E+00	-0.78100E-13	0.44231E-12	0.00000E+00
548	50-0.25784E-03	0.34314E-02	-0.15282E-03	0.20901E-12	-0.87574E-12	0.60396E-13
549	51-0.35287E-03	0.36820E-02	-0.43007E-03	-1021.9	72.417	-185.66
550	52-0.35287E-03	0.36820E-02	-0.43007E-03	1021.9	-72.417	185.66
551	53-0.38131E-03	0.34879E-02	-0.55100E-03	-0.79902E-12	0.18048E-11	-0.14140E-11
552	54-0.22594E-03	0.26038E-02	-0.94200E-03	-0.10039E-12	-0.11637E-11	0.16129E-11
553	55-0.25875E-03	0.35344E-02	0.10000E+00	-0.24009E-13	0.12068E-12	0.00000E+00
554	56-0.32107E-03	0.35647E-02	-0.15000E-03	-0.10000E-12	-0.56621E-14	0.25846E-12
555	57-0.31717E-03	0.37435E-02	-0.38855E-03	-0.51514E-13	0.19318E-13	-0.45119E-12
556	58-0.11466E-02	0.39825E-02	-0.17455E-02	-0.35527E-13	0.10107E-11	0.44764E-12
557	59-0.71109E-03	0.34814E-02	-0.19112E-02	0.35527E-13	0.20410E-11	-0.30198E-11
558	60-0.38445E-03	0.25762E-02	-0.21901E-02	0.35527E-12	-0.37104E-12	0.13705E-11
559	61	0.00000E+00	0.60000E-02	0.00000E+00	0.00000E+00	0.00000E+00
560	62	0.00000E+00	0.60000E-02	-0.25070E-03	0.00000E+00	48.000
561	63	0.00000E+00	0.60000E-02	-0.54430E-03	0.00000E+00	-2448.0
562	64	0.00000E+00	0.60000E-02	-0.54430E-03	0.00000E+00	2432.0
563	65	0.00000E+00	0.60000E-02	-0.74325E-03	0.00000E+00	-32.000
564	66	0.00000E+00	0.60000E-02	-0.95227E-03	0.00000E+00	32.000
565	67-0.10561E-03	0.60000E-02	0.00000E+00	0.71054E-14	16.000	0.00000E+00
566	68-0.10813E-03	0.60000E-02	-0.29179E-03	0.35422E-13	48.000	-0.30553E-12
567	69-0.11717E-03	0.60000E-02	-0.59992E-03	-1.1518	-4896.0	142.11
568	70-0.11717E-03	0.60000E-02	-0.59992E-03	1.1518	4848.0	-142.11
569	71-0.13113E-03	0.60000E-02	-0.74081E-03	-0.67902E-13	32.000	-0.42633E-13
570	72-0.14211E-03	0.60000E-02	-0.95039E-03	-0.21310E-13	32.000	-0.54001E-12
571	73-0.20521E-03	0.60000E-02	0.00000E+00	-0.39903E-13	16.000	0.00000E+00
572	74-0.21400E-03	0.60000E-02	-0.24712E-03	0.44409E-13	64.000	-0.20384E-12
573	75-0.22667E-03	0.60000E-02	-0.40879E-03	-49.501	-4656.0	270.56
574	76-0.22667E-03	0.60000E-02	-0.40879E-03	49.501	4608.0	-270.56
575	77-0.25849E-03	0.60000E-02	-0.64762E-03	-0.74007E-13	-16.000	-0.16875E-12
576	78-0.29616E-03	0.60000E-02	-0.85435E-03	-0.30198E-13	48.000	0.47740E-12
577	79-0.28233E-03	0.60000E-02	0.00000E+00	-0.12454E-13	16.000	0.00000E+00
578	80-0.30470E-03	0.60000E-02	-0.24014E-03	-0.39080E-13	48.000	-0.13767E-12
579	81-0.34939E-03	0.60000E-02	-0.41262E-03	-357.03	-3872.0	420.80
580	82-0.34939E-03	0.60000E-02	-0.41262E-03	357.03	3904.0	-420.80
581	83-0.40527E-03	0.60000E-02	-0.49564E-03	-0.95923E-13	64.000	-0.64926E-12
582	84-0.46693E-03	0.60000E-02	-0.60819E-03	-0.75495E-13	48.000	0.80203E-12
583	85-0.34344E-03	0.60000E-02	0.00000E+00	0.24007E-13	16.000	0.00000E+00
584	86-0.36462E-03	0.60000E-02	-0.14797E-03	-0.67902E-13	16.000	0.35527E-13
585	87-0.49756E-03	0.60000E-02	-0.24049E-03	-332.14	-1680.0	347.53
586	88-0.49756E-03	0.60000E-02	-0.24049E-03	332.14	1664.0	-347.53
587	89-0.80121E-03	0.60000E-02	-0.37001E-03	-0.60990E-13	-32.000	-0.28422E-12
588	90-0.92550E-03	0.60000E-02	-0.37100E-03	0.24809E-13	48.000	0.35705E-12

591 -----  
592 9999 9999  
593 10 V E R A L L T I M E C O U  
594

585	NODAL POINT INPUT	=	0.00
586	ELEMENT STIFFNESS FORMATION	=	0.00
587	NODAL LOAD INPUT	=	0.00
588	TOTAL STIFFNESS FORMATION	=	0.00
589	STATIC ANALYSIS	=	0.00
590	EIGENVALUE EXTRACTION	=	0.00
601	FORCED RESPONSE ANALYSIS	=	0.00
602	RESPONSE SPECTRUM ANALYSIS	=	0.00
603	STEP-BY-STEP INTEGRATION	=	0.00
604			
605	TOTAL SOLUTION TIME	=	0.00
606			

OUTPUT FROM 'KSAFE II' : RESULTS.DAT

1  
2 5 Node EL102/9020s; 32125 MESH 1000; 50005 MESH-200.100 (10/30/87)

4 ---STRESS OUTPUT LOCATIONS---

5 1 21 0 0 0 0 0 0  
6 121 - N O D E S O L I D E L E M E N T S T R E S S

7	8 ELEMENT	LOAD LOCATION	SIG-XX	SIG-YY	SIG-ZZ	SIG
11	1	1	0.000000E+00	0.107370E+04	-0.135072E+03	-0.183894E
12	2	1	-0.013450E+03	0.107329E+04	-0.142729E+03	-0.680461E
13	3	1	-0.013600E+03	0.105201E+04	-0.192748E+03	-0.100091E
14	4	1	-0.013600E+03	0.110929E+04	-0.617359E+02	0.305791E
15	5	1	-0.013614E+03	0.112531E+04	0.130826E+03	-0.847874E
16	6	1	-0.013499E+03	0.112149E+04	0.122555E+03	-0.324551E
17	7	1	-0.013600E+03	0.109723E+04	0.742566E+02	-0.278600E
18	8	1	-0.013639E+03	0.112703E+04	0.204648E+03	-0.346053E
19	9	1	-0.044321E+03	0.144401E+04	-0.648239E+02	-0.160910E
20	10	1	-0.043323E+03	0.143353E+04	-0.681815E+02	-0.545370E
21	11	1	-0.046755E+03	0.141654E+04	-0.110922E+03	-0.864481E
22	12	1	-0.047112E+03	0.142475E+04	-0.106091E+03	-0.352275E
23	13	1	-0.046643E+03	0.108814E+04	0.625469E+02	-0.809481E
24	14	1	-0.042447E+03	0.108391E+04	0.590736E+02	-0.224869E
25	15	1	-0.043903E+03	0.105933E+04	0.164611E+03	-0.312321E
26	16	1	-0.041455E+03	0.112343E+04	0.116455E+03	-0.322742E
27	17	1	0.044756E+03	0.214096E+05	-0.603122E+01	-0.126553E
28	18	1	0.044014E+03	0.214056E+05	-0.926818E+01	-0.392549E
29	19	1	0.043828E+03	0.214536E+05	-0.189085E+02	-0.660276E
30	20	1	0.022746E+03	0.214591E+05	0.876635E+01	-0.637688E
31	21	1	0.043292E+03	0.207950E+05	0.692551E+01	-0.763006E
32	22	1	0.042190E+03	0.207926E+05	0.388937E+01	-0.219859E
33	23	1	0.039362E+03	0.207794E+05	-0.648407E+01	-0.335985E
34	24	1	0.031456E+03	0.207430E+05	0.211307E+02	-0.384381E
35	25	1	0.044406E+03	0.212235E+05	0.184790E+00	-0.104566E
36	26	1	0.043733E+03	0.212341E+05	-0.135654E+00	-0.326547E
37	27	1	0.023675E+03	0.212204E+05	0.124554E+02	-0.518971E
38	28	1	0.013400E+03	0.211650E+05	-0.132877E+02	-0.369496E
39	29	1	0.0424514E+03	0.210275E+05	0.709911E+00	-0.742767E
40	30	1	0.042193E+03	0.210253E+05	0.245991E+00	-0.212575E
41	31	1	0.027449E+03	0.210144E+05	0.127670E+02	-0.333745E
42	32	1	0.110990E+03	0.209958E+05	-0.129398E+02	-0.451993E

43 ----- ENERGY RELEASED IN X, Y, Z Directions -----

44 0.24492099 9.6706281 -9.23363803

46 121 - N O D E S O L I D E L E M E N T S T R E S S

47	48 ELEMENT	LOAD LOCATION	SIG-XX	SIG-YY	SIG-ZZ	SIG
51	1	1	-0.043945E+03	0.107430E+04	-0.131187E+03	-0.677179E
52	2	1	-0.043530E+03	0.108008E+04	-0.118742E+03	-0.147760E
53	3	1	-0.025807E+03	0.105071E+04	-0.244673E+03	0.945992E
54	4	1	-0.018020E+03	0.114190E+04	-0.112547E+03	0.384709E



55	5	1	1	-0.461091E+03	0.172685E+04	0.130259E+03	-0.313154E
56	6	1	1	-0.455280E+03	0.172218E+04	0.122723E+03	-0.363426E
57	7	1	1	-0.298799E+03	0.165999E+04	0.921526E+02	0.256255E
58	8	1	1	-0.595150E+02	0.168531E+04	0.272012E+03	0.265112E
59	9	1	1	-0.330255E+03	0.144035E+04	-0.657823E+02	-0.634614E
60	10	1	1	-0.339405E+03	0.145545E+04	-0.319520E+02	-0.265199E
61	11	1	1	-0.547364E+03	0.140766E+04	-0.172534E+03	0.148727E
62	12	1	1	-0.444362E+02	0.157722E+04	-0.523599E+02	0.355020E
63	13	1	1	-0.509716E+03	0.166601E+04	0.623993E+02	-0.264511E
64	14	1	1	-0.550907E+03	0.163559E+04	0.598138E+02	-0.325761E
65	15	1	1	-0.631592E+03	0.162664E+04	0.232740E+02	0.246812E
66	16	1	1	-0.497861E+03	0.158764E+04	0.179141E+03	0.628181E
67	17	1	1	0.454919E+02	0.214599E+05	-1.940930E-01	-0.491934E
68	18	1	1	0.477818E+03	0.214093E+05	0.136786E+02	-0.257382E
69	19	1	1	0.445136E+03	0.212855E+05	-0.246355E+02	-0.365693E
70	20	1	1	-0.599666E+02	0.172619E+05	-0.645275E+02	0.520656E
71	21	1	1	0.425392E+03	0.209036E+05	0.678539E+01	-0.255128E
72	22	1	1	0.415775E+03	0.208448E+05	0.596236E+01	-0.134598E
73	23	1	1	0.365666E+03	0.205495E+05	-0.344725E+01	0.873985E
74	24	1	1	0.285974E+03	0.187183E+05	0.716737E+02	0.930356E
75	25	1	1	0.451999E+03	0.212285E+05	-0.208313E+00	-0.390673E
76	26	1	1	0.471023E+02	0.212024E+05	-0.923305E+01	-0.246273E
77	27	1	1	0.562284E+03	0.212840E+05	0.285149E+02	-0.109388E
78	28	1	1	0.316123E+03	0.218731E+05	-0.228633E+01	-0.175313E
79	29	1	1	0.423236E+03	0.210360E+05	0.883838E+00	-0.308121E
80	30	1	1	0.416479E+03	0.210620E+05	0.254047E+01	-0.152305E
81	31	1	1	0.343974E+03	0.210059E+05	0.111353E+03	-0.202870E
82	32	1	1	0.141543E+03	0.202542E+05	-0.312466E+02	-0.747290E

83 ----- ENERGY RELEASED in (in. ) z / directions -----

84 0.64558025 14.140609 1.3794686

85 =====

86 121 - NODE SOLID ELEMENT STRESS

87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108
ELEMENT	LOAD	LOCATION		SIG-XX	SIG-YY	SIG-ZZ	SIG														
1	1	1	1	-0.455977E+03	0.108435E+04	-0.104790E+03	-0.231523E														
2	1	1	1	-0.312685E+03	0.106184E+04	-0.223837E+03	0.154030E														
2	1	1	1	-0.214170E+03	0.111562E+04	-0.208921E+03	0.427663E														
4	1	1	1	-0.108979E+03	0.121222E+04	-0.612078E+02	0.279814E														
5	1	1	1	-0.502761E+03	0.172726E+04	0.131473E+03	0.142089E														
6	1	1	1	-0.458382E+03	0.168180E+04	0.133735E+03	0.303849E														
7	1	1	1	-0.293367E+03	0.157673E+04	0.115431E+03	0.354324E														
8	1	1	1	-0.916770E+02	0.154290E+04	0.218118E+03	0.187686E														
9	1	1	1	-0.262870E+03	0.145925E+04	-0.245710E+02	-0.867483E														
10	1	1	1	-0.536140E+03	0.141648E+04	-0.157500E+03	0.999497E														
11	1	1	1	0.151097E+02	0.155991E+04	-0.132885E+03	0.522352E														
12	1	1	1	0.169432E+03	0.173573E+04	0.118579E+02	-0.291787E														
13	1	1	1	-0.574305E+03	0.168757E+04	0.631362E+02	-0.302754E														
14	1	1	1	-0.623795E+03	0.165204E+04	0.718584E+02	0.282842E														
15	1	1	1	-0.614966E+03	0.149768E+04	0.481470E+02	0.791060E														
16	1	1	1	-0.156371E+03	0.141790E+04	0.119957E+03	0.196302E														
17	1	1	1	0.499080E+03	0.214341E+05	0.147009E+02	-0.167694E														
18	1	1	1	0.474299E+03	0.216301E+05	-0.125472E+02	-0.286492E														

109	19	1	1	-0.435933E+02	0.177421E+05	-0.115167E+03	0.522225E
110	20	1	1	-0.852782E+02	0.148759E+05	0.447546E+02	0.123125E
111	21	1	1	0.419800E+03	0.208458E+05	0.784515E+01	-0.848678E
112	22	1	1	0.402690E+03	0.205723E+05	0.107676E+02	0.112853E
113	23	1	1	0.359939E+03	0.186384E+05	0.434208E+02	0.102231E
114	24	1	1	0.422418E+02	0.164075E+05	0.722564E+01	0.767311E
115	25	1	1	0.483780E+03	0.212003E+05	-0.110830E+02	-0.183663E
116	26	1	1	0.659448E+03	0.213080E+05	0.260463E+02	-0.107525E
117	27	1	1	0.411701E+03	0.217331E+05	0.185557E+02	-0.153453E
118	28	1	1	0.498017E+02	0.216361E+05	0.233374E+02	-0.900981E
119	29	1	1	0.421236E+03	0.216689E+05	0.288468E+01	-0.113321E
120	30	1	1	0.400005E+03	0.210154E+05	-0.260228E+01	-0.158576E
121	31	1	1	0.384507E+03	0.205108E+05	-0.206248E+01	-0.540002E
122	32	1	1	0.125790E+03	0.204220E+05	-0.650713E+02	-0.976947E

123

124

125

126

127 STATIC SOLUTION TIME LOG

128

129 EQUATION SOLUTION = 1.00

130 DISPLACEMENT OUTPUT = 0.00

131 STRESS RECOVERY = 0.00

132

END

DATE

FILMED

DTIC

JULY 88