NASA Technical Memorandum 89827

Strain Energy Release Rates of Composite Interlaminar End-Notch and Mixed Mode Fracture: A Subulaminate/Ply Level Analysis and a Computer Code

(NASA-TM-89827) STRAIN ENERGY RELEASE RATES OF COMPOSITE INTERLAMINAR END-NOTCH AND MIXED-MODE FRACTURE: A SUBLAMINATE/PLY LEVEL ANALYSIS AND A COMPUTER CODE (NASA) 88 p CSCL 11D G3/24 — N87-20389 Unclas 45393

R.R. Valisetty and C.C. Chamis

Lewis Research Center

Cleveland, Ohio

Prepared for the

8th Symposium on Composite Materials Testing and Design

sponsored by the American Society for Testing and Materials

Charleston, South Carolina, April 29—May 1, 1986

CONTENTS

STRAIN ENERGY RELEASE RATES OF COMPOSITE INTERLAMINAR END-NOTCH AND MIXED-MODE FRACTURE: A SUBLAMINATE/PLY LEVEL ANALYSIS AND A COMPUTER CODE

R.R. Valisetty* and C.C. Chamis
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135

E-3476

SUMMARY

A computer code is presented for the sublaminate/ply level analysis of composite structures. This code is useful for obtaining stresses in regions affected by delaminations, transverse cracks, and discontinuities related to inherent fabrication anomalies, geometric configurations, and loading conditions. Particular attention is focussed on those layers or groups of layers (sublaminates) which are immediately affected by the inherent flaws. These layers are analyzed as homogeneous bodies in equilibrium and in isolation from the rest of the laminate. The theoretical model used to analyze the individual layers allows the relevant stresses and displacements near discontinuities to be represented in the form of pure exponential-decay-type functions which are selected to eliminate the exponential-precision-related difficulties in sublaminate/ply level analysis. Thus, sublaminate analysis can be conducted without any restriction on the maximum number of layers, delaminations, transverse cracks, or other types of discontinuities. In conjunction with the strain energy release rate (SERR) concept and composite micromechanics, this computational procedure is used to model select cases of end-notch and mixed-mode fracture specimens. The computed stresses are in good agreement with those from a three-dimensional finite element analysis. Also SERRs compare well with limited available experimental data.

INTRODUCTION

Laminated composite structures exhibit a number of different failure modes. These include fiber-matrix debonding within individual layers, delamination or separation of adjacent layers, transverse cracking through one or more layers, and fiber fracture. Often interlaminar delamination initiates at sites where complex states of stress with steep gradients exist. These sites include regions near free edges, ply terminations, cutouts, voids, holes inadvertantly damaged areas, and defects resulting from fabrication processes. As a result, an accurate assessment of the interlaminar fracture toughness parameters and stress states which initiate delamination is of critical importance in a composite design.

Several methods are available to determine fracture toughness and can be classified into the following groups: experimental, numerical, and analytical. To begin with, the experimental procedures used to determine fracture toughness are edge delamination, double-cantilever-beam, cracked-lap-shear, biaxial

*National Research Council - NASA Research Associate.

interlaminar fracture, and three point bend tests for both Mode I (end-notch fracture, ENF) and mixed mode fracture (MMF) including Mode I and Mode II. Because of the test simplicity and the ability to delineate between Mode II and mixed mode fracture, the three-point bend tests are now being investigated at NASA Lewis Research Center. Results for interlaminar fracture toughness of unidirectional graphite/epoxy composites are presented in reference 1.

Next, there are numerical solutions. In general, these are developed using either finite difference, finite element, finite-element boundary integral concepts, boundary layer, energy based approximations, and extended Galerkin procedures. They require extensive idealizations to model the response variables in the critical regions and the numerical accuracy of the results is not always assured. Issues related to this topic are discussed in references 2 to 5.

Conventional laminate theories form the core of the final group, the analytical approaches. These are best suited for obtaining the total strain energy release rates. Fracture-related interlaminar stresses which are necessary for delineating between different modes of crack propagation cannot be obtained from these theories. This is because the theories are based on plane stress and global displacement assumptions. The assumption of plane stress precludes the possibility of considering the interlaminar stresses at the outset of the formulation.

This led to the development of ply/sublaminate level analysis (refs. 2 and 4) of composite laminates. Here, layers or selected groups of layers (treating each group as a homogeneous body) are analyzed with the assumption that they are in equilibrium and independent of each other. Thus, the interlaminar stresses are introduced in the formulation at an earlier stage. Initially, these stresses are assumed to be unknown. Then considering the equilibrium of each layer/sublaminate and compatibility at each interlaminar surface, a set of coupled equations in terms of global deformation variables and interlaminar stresses is generated. Subsequent to a successful solution, the fracture toughness and SERRs are easily computed. This approach was initially applied to the estimation of free-edge stresses (ref. 2), and later, successfully extended to both an edge delamination problem (ref. 6) and the analysis of compression and double-lap fracture specimens (ref. 7).

This report describes the implementation of sublaminate analysis to general composite laminates. This analytical technique has been incorporated into a computer code which is listed in appendix I. The code is written with flexibility in regards to the types of materials and loads, number of layers, and boundary conditions as well as cracks which may be in the plane of the laminate or through the thickness. The versatility of this method and the numerical economy in conducting design-related parametric studies are demonstrated through the simulation of three-point bend tests of ENF and MMF specimens. A brief description of sublaminate analysis and ply level equations are presented and an illustration of the basic steps involved in ply-modeling a given structure and setting up the governing equations is included.

SUBLAMINATE/PLY LEVEL ANALYSIS

This analysis which operates on a ply-by-ply basis may be applied to all layers in a laminate or only to those few layers which surround defects and where the stresses are adversely affected. The remaining layers can be grouped into sublaminates and may be treated as homogeneous layers. Interlaminar stresses are initially assumed to be unknown. Enforcement of displacement and traction continuity at laminar interfaces leads to the final equations, which are subsequently solved. Thus, a better engineering approximation is achieved concerning the interlaminar stresses. The success of the approach, then, depends on the engineering model that will be used to analyze each layer. The model should be capable of securing "layer equilibrium" as discussed below.

Just as it is possible to combine different layers into a single unit or a sublaminate and treat the resulting unit as a single layer, it is also possible to idealize a single layer as composed of a finite number of thin sublayers. For the purpose of sublaminate/ply level analysis, these sublayers once again can be treated as individual units. This situation is, therefore, similar to employing either a coarse mesh or a fine mesh at selective locations through the thickness in a finite element analysis. The ensuing results for interlaminar stresses depend upon the idealization in terms of sublaminates, layers, and sublayers and to what extent they are affected is discussed in reference 4.

To achieve "layer equilibrium," the model must be capable of dealing with prescribed stress/displacement conditions on transverse interlaminar surfaces and afford specification of at least five boundary conditions (three force resultants or deflections and two moment resultants or rotations) per layer edge. Two such models are available now. The first one, developed by Pagano (ref. 2), is a higher order theory based on Reissner's variational principle. The second one, developed by Valisetty (ref. 8) from an iterative procedure, is a refined homogeneous plate bending theory. The later model was selected to analyze the layers in the present program because it provides explicit distributions through the thickness for stresses and displacements.

A summary of the basic equations for the generic ply shown in figure 1 is given in appendix II. The equations are deduced from the homogeneous plate theory (ref. 8) for the case of cylindrical bending. The basis for these equations and their derivation are presented in references 6 and 8.

Suppose the analysis is to be applied to N layers or a combination of layers and sublaminates. The overall equations of equilibrium and the constitutive relationships will constitute a set of $8N$ equations in terms of like variables ($2N$ displacements, N rotations, $3N$ force resultants, and $2N$ moment resultants). This set is further supplemented by additional $2(N-1)$ equations, which are necessary for the simultaneous solution of $2(N-1)$ interlaminar stresses, when the displacement continuity is enforced at the $(N-1)$ interlaminar surfaces. The transverse shear and normal stresses at the interlaminar surfaces, if they are not prescribed explicitly (for example, as on the top and bottom surfaces of a laminate or on the cracked interlaminar surfaces), are treated as unknowns. From among this set of equations the force and moment resultant variables can be eliminated with the aid of constitutive relations. This leaves a set of $(5N-2)$ coupled ordinary differential equations to be solved for $2N$ displacement variables, N rotations, and $2(N-1)$ interlaminar stresses.

These observations are based upon two requirements: (1) the laminate is in generalized plane strain, and (2) there is no cross-sectional warping. The latter condition is satisfied when cross-ply laminates and symmetric laminates are used. However, for angle ply laminates and laminates of more general construction, the warping cannot be neglected. In this case, the number of variables increase and likewise the number of equations to be solved.

The final system of differential equations can be represented in the following form:

$$\sum_{k=0}^{4} \sum_{j=1}^{4N-2} SK_{ij}^{(k)} f_j^{(k)} = p_i; \quad i = 1,2,\ldots,4N-2 \tag{1}$$

where p represents the applied loading, f is the vector of governing variables (2N average displacements and 2N-2 interlaminar stresses), and the superscript k is the order of the in-plane derivative. The coefficient matrices ([SK]) depend on layer thickness and material properties. The total solution is constructed by obtaining: (1) a particular solution corresponding to the applied load, and (2) other solutions from the homogeneous part of the differential equations. The analysis shows that there are (8N-2) roots or eigenvalues, therefore an equal number of homogeneous or eigen solutions. An inspection of the eigenvalues reveals the nature of these solutions. There are six zero roots and the remaining are either real or complex. The solution corresponding to the zero roots is identified to be of shear deformation type and the remaining are of local type which are prominent only in the edge regions of the laminate.

There are (8N-2) integration constants which require specification of a like number of boundary conditions at the laminate edges. These are established on the basis of physical reasoning. Since global laminate equilibrium is translated into local layer equilibrium in this analysis, boundary conditions are to be specified at the layer edges insofar as possible. Among such conditions, the engineering boundary conditions appear to be a natural choice. Consequently, one can prescribe either force resultant or an average in-plane displacement, moment resultant or an average rotation and shear resultant or an average transverse displacement at each layer edge. The remaining (2N-2) conditions are specified in terms of either interlaminar transverse shear stress or its in-plane derivative at the in-plane edges of interlaminar surfaces.

By retaining the local solutions at only one laminate edge, the number of boundary conditions can be reduced to (4N+2). The edge at which the local solutions are to be retained depends on the particular problem in question. This procedure does not threaten the laminate equilibrium as it can be secured from the global perspective with the aid of a shear deformation type solution (which is controlled by the six zero roots) alone. However, for this procedure to be successful, the layer-defined force and moment resultants at the edge, where the local solutions are dropped, must be specified to be statically equivalent to the local total loading applied to the whole laminate edge. At the edge where the local solutions are retained the boundary conditions can still be specified on each individual layer.

This maneuver makes the sublaminate/ply level analysis powerful, for it allows equilibrium in any number of layers to be considered independently. Usually, the number of layers to which this type of analysis is applied is restricted by a computer precision requirement. This requirement puts a limit on the magnitude of the real roots and real part of the complex roots. When the local solutions are obtained for each laminate edge, one at a time, the mode shape of the eigen solutions can be constructed in the form of pure decay type solutions which are valid only at one edge. Thus the exponential-function-related precision problem are effectively eliminated. This is a significant contribution, because now sublaminate/ply level analysis can be extended to laminates with discontinuities in loads, materials, and geometry along the axis of the laminate and without any restriction as to the number of layers.

For cases involving axial discontinuities, the sublaminate analysis procedure essentially remains the same. The first step is to divide the structure into several regions or subunits such that each subunit is free of any discontinuity related to loads, materials, or geometry. Each subunit, then, will be in the form of a laminate loaded by continuously varying loads. Materials may differ for layers within the subunits. The concentrated loads on the original structure translate into shear loads applied to the edges of the subunits. The delaminations can be accounted for by treating the separate portions of the laminate as different sublaminates abutting the uncracked laminate which may be treated as just another sublaminate. This process is described in detail in the next section with an example of ENF and MMF specimens.

Formal solutions in terms of integration constants can then be constructed for each subunit using the aforementioned procedure. The postulation of boundary conditions and continuity conditions between different subunits provides equations for the solution of integration constants. Typical continuity conditions involve specifying axial force resultant, shear resultant, moment resultant, axial displacement, transverse displacement, and cross-sectional rotation for each layer and transverse shear stress and its derivative for each interlaminar surface.

The computer code containing these procedures is presented in appendix I. Instructions for the input and output are given in appendix III. Typical input and output for the ENF specimen are given in appendices IV and V, respectively.

IDEALIZATION OF THREE-POINT BEND TESTS

The sublaminate/ply level analysis is now applied to estimate the interlaminar fracture stresses in the three-point bend test specimens. Figure 2 presents the schematics for the three-point bend tests of the ENF and MMF specimens. For the MMF specimen, the right support is provided on the bottom surface of the top flange instead of the bottom flange. The laminate is made of 24 graphite/epoxy (AS/E) layers and five interlaminar resin layers as shown in figure 3. The manufacturing process gives rise to the resin rich region between adjacent fiber-matrix layers whose thickness usually amounts to a tenth of the fiber diameter and in general depends on the fiber volume ratio (FVR). This region is referred to as the interply resin layer and is considered in order to accurately compute those stresses which induce interply delamination. Two different FVRs, 0.3 and 0.6, are considered in the computations. At these FVRs, the interply resin layer thicknesses are 0.0001854 in. and 0.00004323 in., respectively. The graphite/epoxy layers have the following properties:

At FVR = 0.3

E_L = 11.96 msi, E_T = 0.921 msi, E_{TT} = 1.01 msi, G_{TT} = 0.245 msi,
G_{LT} = 0.368 msi, ν_{LT} = 0.348, ν_{TT} = 0.546
ply thickness = 0.005 in.

At FVR = 0.6

E_L = 21.15 msi, E_T = 1.238 msi, E_{TT} = 1.323 msi, G_{TT} = 0.362 msi,
G_{LT} = 0.623 msi, ν_{LT} = 0.282, ν_{TT} = 0.423
ply thickness = 0.005 in.

where E is the elastic modulus, G is the shear modulus and ν is Poisson's ratios, L represents a direction along the fibers and T, a direction, normal to the fibers.

As inspection reveals that in order to apply sublaminate/ply level analysis the specimen (fig. 2) must be divided into the following four regions:

$$\begin{aligned}
&\text{Region 1: } 0 &&< x_2 < L/2; \; -H/2 < z < H/2,\\
&\text{Region 2: } L/2 &&< x_2 < (L-1); \; -H/2 < z < H/2,\\
&\text{Region 3: } (L-a) &&< x_2 < L; \; 0 < z < H/2,\\
&\text{Region 4: } (L-a) &&< x_2 < L; \; -H/2 < z < 0.
\end{aligned} \tag{2}$$

where a, L and H denote the length of the crack, the length of the specimen and its thickness, respectively. The length of the specimen is 4 in. and (x_2, z) is a global coordinate system. Region 1 is identical to Region 2 and Regions 3 and 4 are identical. Now, the analysis can be applied to each of these four regions obtaining four sets of formal solutions in terms of integra tion constants. The total number of these constants will be 680.

Before proceeding to setup such a solution, it should be asked whether the analysis can be simplified any further. The answer will be obvious, if it is recognized that the objects of investigation are the stresses in the interlaminar resin layer in the region ahead of the crack. To obtain these stresses, it is not necessary to apply the analysis to all of the layers within each of the four regions. Recognizing this, it can be assumed that the 12 graphite/epoxy and two interlaminar resin layers (fig. 3) can be treated as single units or sublaminates in the third and fourth regions and in the top and bottom halves of the first and second regions. These sublaminates can further be treated as homogeneous orthotropic plates represented by their effective moduli. The resulting idealization is shown in figure 4.

Now, the analysis is applied only to the first and second regions. Since the sublaminates in the third and fourth regions are treated as homogeneous plates, the governing solutions for these regions, which are of the shear deformation type, can be obtained directly. The order of the resulting set of all equations will be 56. This can further be reduced to 32 by retaining only the local solution near the crack in Region 2. This solution yields the fracture related stresses in the resin layer ahead of the crack. The other local solutions, which are suppressed, control the stress variation near the concentrated load and at the left support.

Equations for Solution

The solution procedure for the specimen idealized in figure 4 is described here. For this purpose, the laminate in Region 1 is selected. Let "1," "2," and "3" denote, respectively, the top, middle, and bottom sublaminates. The outer sublaminates each contain 12 graphite/epoxy and two resin layers. The middle one contains just one interply resin layer. For the analysis, these sublaminates are treated as homogeneous plies. It is convenient to use local coordinate systems, (x_2,z; k+1,2; and 3), for presenting stress and displacement distributions. The ply middle surfaces are represented by $Z = 0$ and c denotes ply semithicknesses.

For each layer, stress and displacement distributions through the thickness are available. These were discussed earlier in appendix II along with relevant equilibrium and constitutive equations in terms of engineering kinematic and force variables and interlaminar stresses. As shown in figure 1, these variables are meaningful for only those plies for which they are defined. When it is necessary to indicate such ply dependency, these variables are superscripted with a ply identification number.

Herein, let U_2 and W be the components in the x_2 and z directions, respectively of the displacement of the $z = 0$ surface. Also, let σ_{2z} and σ_{zz} be the transverse shear and normal stresses. Using this notation, the unknown variables can be listed as U_2^1, U_2^2, U_2^3, W^1, W^2, W^3, $\sigma_{2z}^2(x_2,z = c)$, $\sigma_{2z}^3(x_2,z = c)$, $\sigma_{zz}^2(x_2,z = c)$, and $\sigma_{zz}^3(x_2,z = c)$. In all, these variables constitute a set of ten unknowns. An identical set of variables also govern the deformation in Region 2. For sublaminates defined in the third and fourth regions, U_2 and W are the only governing variables.

Equations I-2 and I-3 of appendix II may be combined to produce

$$M_{2,22} + c\, m_{2,2} + q_2 = 0 \tag{3}$$

where

M_2 moment resultant associated with x_2 direction,

c ply semithickness,

m_2 $\sigma_{2z}(x_2, c) + \sigma_{2z}(x_2, -c)$, and

q_2 $\sigma_{zz}(x_2, c) - \sigma_{zz}(x_2, -c)$.

The equilibrium conditions (eq. (3), and eq. (I-1) of appendix II) for each of the three plies and the following continuity requirements:

$$
\begin{aligned}
&u_2^1 = u_2^2;\ 0 < x_2 < L/2,\ z_1 = -c_1, \\
&u_2^2 = u_2^3;\ 0 < x_2 < L/2,\ z_2 = -c_2, \\
&w^1 = w^2;\ 0 < x_2 < L/2,\ z_1 = -c_1, \text{ and} \\
&w^2 = w^3;\ 0 < x_2 < L/2,\ z_2 = -c_2.
\end{aligned} \tag{4}
$$

for the displacements at the two laminar interfaces make up the desired set of equations for Region 1, where u_2 and w are the components of the displacement in the x_2 and z directions. Equation 4 can be expressed solely in terms of the generalized unknown variables of the problem with the aid of Eqs. I-16 and I-17 of appendix II which are thicknesswise displacement distributions. An identical set of equations follows for the second region. The cracked surfaces in Regions 3 and 4 are free of the transverse stresses. Therefore, the solution for the sublaminates in these regions degenerates into a shear deformation type solution with a shear correction factor of 1.

These four sets of equations are solved separately in a straightforward manner and formal solutions in terms of integration constants are constructed. There are 56 constants which may be determined through the boundary conditions at $x_2 = 0$ and $x_2 = L$ and the continuity conditions between the first and second regions at $x_2 = L/2$, between the second and third regions at $x_2 = (L-a)$, and between the second and fourth regions at $x_2 = (L-a)$. For presenting these conditions, superscripts "4," "5," and "6" are used to denote the top, middle, and bottom layers in Region 2 and "7" and "8" are for the sublaminates in the third and fourth regions. As mentioned before, superscripts "1," "2," and "3" apply similarly to the variables in the first region.

Boundary conditions at $x_2 = 0$:

$$
\begin{aligned}
&N_2^1 = 0,\ M_2^1 = 0,\ Q_2^1 = 0,\ \sigma_{2z}^1(x_2, -c_1) = 0 \\
&N_2^2 = 0,\ M_2^2 = 0,\ Q_2^2 = 0,\ \sigma_{2z}^2(x_2, -c_2) = 0 \\
&u_2^3\ (x_2, -c_3) = 0,\ w^3(x_2, -c_2) = 0\ M_2^3 = 0
\end{aligned} \tag{5}
$$

Continuity conditions at $x_2 = L/2$:

$$N_2^1 = N_2^4,\ M_2^1 = M_2^4,\ Q_2^1 = Q_2^4+60,\ \sigma_{2z}^1(x_2, -c_1) = \sigma_{2z}^4(x_2, -c_4)$$

$$U_2^1 = U_2^4,\ \phi_2^1 = \phi_2^4,\ W^1 = W^4,\ \sigma_{2z}^1(x_2, -c_1) = \sigma_{2z}^4(x_2, -c_4)_{,2}$$

$$N_2^2 = N_2^5,\ M_2^2 = M_2^5,\ Q_2^2 = Q_2^5,\ \sigma_{2z}^2(x_2, -c_2) = \sigma_{2z}^5(x_2, -c_2) \tag{6}$$

$$U_2^2 = U_2^5,\ \phi_2^2 = \phi_2^5,\ W^2 = W^5,\ \sigma_{2z}^2(x_2, -c_2) = \sigma_{2z}^5(x_2, -c_2)_{,2}$$

$$N_2^3 = N_2^6,\ M_2^3 = M_2^6,\ Q_2^3 = Q_2^6+60,$$

$$U_2^3 = U_2^6,\ \phi_2^3 = \phi_2^6,\ W^3 = W^6,$$

Continuity conditions at $x_2 = (L\text{-}a)$:

$$N_2^4 = N_2^7,\ M_2^4 = M_2^7,\ Q_2^4 = Q_2^7$$
$$U_2^4 = U_2^7,\ \phi_2^4 = \phi_2^7,\ W^4 = W^7 \tag{7}$$

Continuity conditions at $x_2 = (L\text{-}a)$:

$$N_2^6 = N_2^8,\ M_2^6 = M_2^8,\ Q_2^6 = Q_2^8$$
$$U_2^6 = U_2^8,\ \phi_2^6 = \phi_2^8,\ W^6 = W^8 \tag{8}$$

Boundary conditions at $x_2 = (L\text{-}a)$:

$$N_2^5 = 0,\ M_2^5 = 0,\ Q_2^5 = 0,$$
$$\sigma_{2z}^4(x_2, -c_4) = 0,\ \sigma_{2z}^6(x_2, -c_6) = 0 \tag{9}$$

Boundary conditions at $x_2 = L$ for ENF specimen:

$$N_2^7 = 0,\ M_2^7 = 0,\ Q_2^7 = 0,$$
$$N_2^8 = 0,\ M_2^8 = 0,\ w^8(x_2, -c_8) = 0 \tag{10}$$

Boundary conditions at $x_2 = L$ for MMF specimen:

$$N_2^7 = 0,\ M_2^7 = 0,\ w^7(x_2, -c_7) = 0,$$
$$N_2^8 = 0,\ M_2^8 = 0,\ Q_2^8 = 0. \tag{11}$$

where c_k (k = 1,2,...8) are ply semithicknesses, d_2 are cross-sectional rotations, N_2 are force resultants and M_2 are moment resultants.

The applied load of 120 lb is represented as a jump in shear resultants in the outer layers in equation 6. It is assumed that the central interlaminar resin layer is too thin to carry any net shear force and the shear stress in the outer sublaminates away from the point of application of the load is uniform through the thickness. Similarly, the conditions on the transverse shear stress in equation 6 are arrived at under the assumption that the points at which the stress is prescribed are considerably away from the point of application of the load; therefore, the stress is unaffected. Additionally, the crack is modeled as having a finite thickness, the thickness being that of the interply resin layer. The interply resin layer is destroyed as the crack advances so that the crack-tip (idealized as having finite thickness) is stressfree. This is indicated by equation 9.

The approach discussed so far makes use of the complete form of the eigen solutions to represent the response variable in the first two regions. This yields stress and displacement distributions which show not only the effect of the crack-tip but also that of the concentrated load at the left support. Since the objective is the crack-tip stress field, the form of eigen solutions can be modified to include only the decay-type solutions at the crack-tip. This degenerates formal representation of the solution in Region 1 into that of a shear deformation theory, as is also the case with the third and fourth regions. As a result, the total number of integration constants and consequently the number of boundary/continuity conditions are reduced to 36. The conditions at $x_2 = 0$ and $L/2$ are to be modified as shown below:

Boundary conditions at $x_2 = 0$:

$$u_2^2(x_2, 0) = 0 ,$$

$$u_2^2(x_2, 0) = 0 , \tag{12}$$

$$M_2^1 + M_2^2 + M_2^3 + N_2^1 \times c_1 - N_2^3 \times c_3 = 0$$

Continuity conditions at $x_2 = L/2$:

$$
\begin{aligned}
&u_2^2(x_2\,,\,0) = u_2^5(x_2\,,\,0)\\
&w^2(x_2\,,\,0) = w^5(x_2\,,\,0)\\
&d_2^2(x_2\,,\,0) = d_2^5(x_2\,,\,0)\\
&N_2^1 + N_2^2 + N_2^3 + N_2^4 + N_2^5 + N_2^6\\
&Q_2^1 + Q_2^2 + Q_2^3 + Q_2^4 + Q_2^5 + Q_2^6 + 120\\
&M_2^1 + M_2^2 + M_2^3 + N_2^1 \times c_1 - N_2^3 \times c_3 = M_2^4 + M_2^5 + M_2^6 + N_2^4 \times c_4 - N_2^6 \times c_6
\end{aligned}
\tag{13}
$$

The procedure described so far is programmed into a general purpose code. A description of this code (appendix I) together with instructions for input and output (appendix III) are included.

RESULTS AND DISCUSSION

Figures 5 and 6 display the stress field in the interply layer ahead of the crack-tip for the ENF and MMF specimens, respectively. The crack length is 1 in. and the fiber volume ration (FVR) is 0.6. The ENF specimen yields momotonically decreasing compressive normal stresses, whereas the MMF specimen produces tensile stresses. For both specimens, the transverse shear stress is positive near the crack-tip. Within a distance of 1.5 H from the crack-tip, all stresses approach engineering values. The compressive nature of the transverse normal stress for the ENF specimen indicates that this is a pure Mode II crack. For the MMF specimen, as anticipated, both transverse shear and normal stresses are tensile. Figures 7 and 8 show the behavior of peak stresses as the crack advances. Both types of specimens show monotonically increasing crack-tip stresses.

Strain energy release rates (SERR) are computed by monitoring changes in the total compliance of the structure. Figure 9, which shows the total energy release rate plotted against the crack length, indicates a stable crack growth. Both the ENF and MMF specimens predict the same total SERR. This result was also anticipated since, in the case of the MMF specimen, the unsupported delaminated flange will not pick up any load. However, in the case of the ENF specimen, the possibility for load transfer exists as both delaminated flanges come into contact as the applied load increases. Since this contact is not incorporated into the analysis, the same results are predicted.

In order to study this progressive contact, the relative displacement between the cracked surfaces should be restricted so that cracked sublaminates do not run into one another. If they do the problem becomes nonlinear and is beyond the scope of the present investigation. However, a simple idealization of the practical situation using an ENF specimen assumes complete contact between the cracked surfaces and that the resulting contact stresses are negligible. Then, the condition on the transverse displacement in equation 10, accordingly, is replaced by the following:

$$w^7(x_2, -c_7) = w^8(x_2, c_8) \tag{14}$$

This model is named ENF-II. Results for the stress field ahead of the crack-tip and total SERR using this model are presented in figures 10 and 11, respectively. The FVR for these results is 0.6. Both normal stresses are zero and in figure 10 the shear stress follows the previously observed trend. In this case, the crack plane becomes a plane of symmetry. This does not permit the normal stresses to develop on the crack plane. This total SERR is only about one-third of that which is available in the first two models. As shown in figure 11, these results compare well with engineering theory which are slightly conservative. Reference 1 presents some experimental data on the total SERR for the ENF specimen. The range of the total SERR shown in figure 11 matches with the experimentally measured range of 3.0 to 3.6 psi-in. The role of the local stress field on crack propagation can be established by measuring individual mode contributions, but that study is deferred for a future investigation.

In order to assess the present study, the interply stresses ahead of the crack-tip are compared in figure 12 with the three-dimensional finite element results of reference 1 for the ENF specimen. With the exception of small differences in peak stresses at the crack-tip, both analyses provide the same trends. The finite element results for the transverse shear stress exhibit an oscillatory behavior. This was attributed, in reference 1, to the particular fashion in which the finite element grid ahead of the crack-tip was updated as the crack length increased.

CONCLUSIONS

The analysis presented herein can be extended easily to cracks through the thickness. In this case the eigen solutions are to be computed each time the crack extends. Discontinuities related to geometry and loads can also be handled in an analogous manner. Similarly, flexibility is available to consider delaminations anywhere through the thickness and along the axis of the laminate. These features make the present analysis ideally suited for design related parametric studies when: (1) different configurations and the effect of materials are to be evaluated, and (2) fracture characteristics need to be established. The following conclusions are reached based on this study:

1. After the eigen solutions are established in each region, the order of the final system of equations representing boundary/symmetry/continuity conditions is reduced to 36 for the specimen idealizations. This results in an analysis which is highly economical.

2. The ability to represent critical local stress fields in terms of pure decay-type solutions eliminates the exponential-function-related precision problems. This effectively removes the cap on the number of layers that can be studied on a ply-by-ply basis.

3. Different types of restraints can be handled routinely, as indicated by ENF, ENF-II and MMF models, without incurring any major changes in the final systems of equations.

4. The models also illustrate how the stress field ahead of the crack-tip responds to simple changes in the edge supports. This knowledge is important in designing fracture specimens for different modes of crack propagation.

```
      PROGRAM MAIN
C.....CDABS=CABS
C.....DCMPLX....REMOVE THIS DECLARATION
C.....REAL*8
C.....REAL*4 RCM
C.....COMPLEX*16
C   E11       S32      CB16     KQ1      CURVTR   2  NZR   12  CRTB  22
C   E22       S33      CB22     KQ2      NSREG    3  NRR   13  CRTC  23
C   E33       S44      CB26     KM1      NMT      4  NCR   14  ISGN  24
C   NU12      S45      CB66     KM2      NSL      5  NBCR  15
C   NU13      S55      SNC1     KP1      NGN      6  CRT   16
C   NU23      CB1      SNC2     KP2      NGT      7  CMODE 17
C   G44       CB2      SHM1     KN1      NMTYPE   8  SLN   18
C   G55       CB6      SHM2     KN2      THICK    9  SK    19
C   G66       CB11     SHQ1     THETA    THCK    10  RK    20
C   S31       CB12     SHQ2     EPSLON 1 NSR     11  CRTA  21
      IMPLICIT REAL   (A-H,O-Z)
      REAL   RCM
      DIMENSION RCM(150000)
      DATA IN5,IN6,IN7,IN8/5,6,5,6/
      MAXRCM=150000
      IPR=1
      CALL PZERO(RCM,MAXRCM/IPR)
      READ(IN5,2) NSREG,NMT
      WRITE(IN6,9)NSREG,NMT
9     FORMAT(/1X,2I3,'   NO. OF REGIONS AND NO. OF MATERIALS')
2     FORMAT(20I2)
92      FORMAT(10E10.3)
      N1=1+39*NMT*IPR
      N2=N1+IPR
      N3=N2+IPR
      N4=N3+1
      N5=N4+1
      N6=N5+NSREG
      N7=N6+NSREG
      N8=N7+NSREG
      N9=N8
      RCM(N3)=NSREG
      RCM(N4)=NMT
      X=1.D 00
C.....READ EPSLON, CURVTR, NSL, NGN, NGT
      CALL METPRO(RCM,IN5,IN6,IN7,IN8,NSREG,NMT,RCM(N5),RCM(N6),RCM(N7),-
     *1,X,1,1,X,IPR,RCM(N1),RCM(N2),1,1,1,1)
      DO 1 I=1,NSREG
C.....READ NMTYPE
      CALL METPRO(RCM,IN5,IN6,IN7,IN8,NSREG,NMT,RCM(N5),RCM(N6),RCM(N7),-
     *2,X,I,RCM(N9),X,IPR,X,X,1,I2,1,1)
1     N9=N9+I2
      N10=N9
      DO 3 I=1,NSREG
```

```
C.....READ THICK
      CALL METPRO(RCM,IN5,IN6,IN7,IN8,NSREG,NMT,RCM(N5),RCM(N6),RCM(N7),-
     *3,RCM(N10),I,1,X,IPR,X,X,1,I2,1,1)
3     N10=N10+I2*IPR
C.....READ THCK
      CALL METPRO(RCM,IN5,IN6,IN7,IN8,NSREG,NMT,RCM(N5),RCM(N6),RCM(N7),-
     *6,RCM(N9),1,1,RCM(N10),IPR,X,X,1,1,1,1)
      N11=N10+NSREG*IPR
C.....READ MECHANICAL PROPERTIES
      CALL METPRO(RCM,IN5,IN6,IN7,IN8,NSREG,NMT,1,1,1,4,X,1,1,X,IPR,X,X,-
     *1,1,1,1)
      N12=N11+NSREG
      N13=N12+NSREG
      N14=N13+NSREG
      N15=N14+NSREG
      N16=N15+NSREG
      N17L1=0
      N17L2=0
      N18L=0
      DO 15 I=1,NSREG
      CALL METPRO(RCM,IN5,IN6,IN7,IN8,NSREG,NMT,RCM(N5),RCM(N6),RCM(N7),-
     *5,X,1,1,X,IPR,X,X,I,I2,I,I4)
      IF(I4.EQ.0.OR.I4.EQ.I.OR.I4.GT.I) GOTO 16
      GOTO 15
16    N17L1=N17L1+8*I2-2
      IF(4*I2-2.GT.N17L2) N17L2=4*I2-2
      N18L=N18L+4*I2-2
15    CONTINUE
      N17=N16+N17L1*2*IPR
      N18=N17+N17L1*N17L2*4*2*IPR
      N19=N18+N18L*IPR
      N20=N19+N17L1*N17L1*IPR
      N21=N20+N17L1*IPR
      N22=N21+N17L1*IPR
      N23=N22+N17L1*IPR
      N24=N23+N17L1*IPR
      N25=N24+N17L1
      WRITE(IN6,17) N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15, -
     *N16,N17,N17L1,N17L2,N18,N19,N20,N21,N22,N23,N24,N25
17    FORMAT(20I5)
      CALL MPRG(IN5,IN6,IN7,IN8,NSREG,NMT,RCM(N5),RCM(N6),RCM(N7),        -
     *RCM(N8),RCM(N9),RCM(N10),RCM(N11),RCM(N12),RCM(N13),RCM(N14),       -
     *RCM(N15),RCM(N16),RCM(N17),RCM(N18),RCM(N19),RCM(N20),RCM(N21),     -
     *RCM(N22),RCM(N23),RCM(N24),N17L1,N17L2,N18L,N25,MAXRCM,RCM,         -
     *IPR)
      STOP
      END
      SUBROUTINE METPRO(RCM,IN5,IN6,IN7,IN8,NSREG,N,NSL,NGN,NGT,NXX       -
     *,THICK,NR,NMTYPE,THCK,IPR,EPSLON,CURVTR,I1,I2,I3,I4)
      IMPLICIT REAL     (A-H,O-Z)
      REAL   RCM
      DIMENSION RCM(1),NSL(1),NGN(1),NGT(1),THICK(1),NMTYPE(1),THCK(1)
      GOTO(1,2,3,4,5,6),NXX
1     READ(IN5,17)  EPSLON,CURVTR
```

```
       WRITE(IN6,19) EPSLON,CURVTR
       DO 11 NR=1,NSREG
       READ(IN5,12)    NSL(NR),NGN(NR),NGT(NR)
 11    WRITE(IN6,13)NR,NSL(NR),NGN(NR),NGT(NR) -
       RETURN
 2     NL=NSL(NR)
       I2=NL
       WRITE(IN6,14) NR
       READ(IN5,12)    (NMTYPE(I),I=1,NL)
       WRITE(IN6,12)   (NMTYPE(I),I=1,NL)
       RETURN
 3     NL=NSL(NR)
       I2=NL
       WRITE(IN6,16) NR
       READ(IN5,17) (THICK(I),I=1,NL)
       WRITE(IN6,17) (THICK(I),I=1,NL)
       RETURN
 6     J=0
       DO 18 K=1,NSREG
       NL=NSL(K)
       THCK(K)=0.D 00
       DO 18 I=1,NL
       J=J+1
       THCK(K)=THCK(K)+THICK(J)
 18    CONTINUE
       WRITE(IN6,172)
       WRITE(IN6,17) (THCK(I),I=1,NSREG)
       RETURN
 4     I=N*IPR
       CALL   MPRO1(RCM(1),RCM(I+1),RCM(2*I+1),RCM(3*I+1),RCM(4*I+1),     -
      *RCM(5*I+1),RCM(6*I+1),RCM(7*I+1),RCM(8*I+1),RCM(9*I+1),RCM(10*I+1)-
      *,RCM(11*I+1),RCM(12*I+1),RCM(13*I+1),RCM(14*I+1),RCM(15*I+1)      -
      *,RCM(16*I+1),RCM(17*I+1),RCM(18*I+1),RCM(19*I+1),RCM(20*I+1)      -
      *,RCM(21*I+1),RCM(22*I+1),RCM(23*I+1),RCM(24*I+1),RCM(25*I+1)      -
      *,RCM(26*I+1),RCM(27*I+1),RCM(28*I+1),RCM(29*I+1),RCM(30*I+1)      -
      *,RCM(31*I+1),RCM(32*I+1),RCM(33*I+1),RCM(34*I+1),RCM(35*I+1)      -
      *,RCM(36*I+1),RCM(37*I+1),RCM(38*I+1)                              -
      *,N,IN5,IN6,IN7,IN8)
       RETURN
 5     I2=NSL(I1)
       I4=NGT(I3)
       RETURN
 172   FORMAT(/1X,'SUBLAMINATE THICKNESSES')
 17    FORMAT(8E10.4)
 14    FORMAT(/1X,'REGION  =  ',I3,',  MATERIAL FOR EACH LAYER')
 19    FORMAT(/1X,2E10.4,'  EPSLON,  CURVTR'/)
 12    FORMAT(20I4)
 13    FORMAT(1X,'REGION  =  ',I3,',  NO. OF LAYERS  =  ',I3,',  NGN  =  -
      *',I3,',  NGT  =  ',I3)
 16    FORMAT(/1X,'REGION  =  ',I3,',  PLY THICKNESSES')
       END
       SUBROUTINE MPRG(IN5,IN6,IN7,IN8,NSREG,NMT,NSL,NGN,NGT,            -
      *NMTYPE,THICK,THCK,NSR,NZR,NRR,NCR,                                -
      *NBCR,CRT,CMODE,SLN,SK,RK,CRTA,                                    -
```

```
     *CRTB,CRTC,ISGN,NCN1,NCN2,NCN4,N25,MAXRCM,RCM,IPR)
      IMPLICIT REAL   (A-H,O-Z)
      REAL   RCM
      DIMENSION NSL(1),NGN(1),NGT(1),NZR(1),NRR(1),NCR(1),NBCR(1),NSR(1)-
     *,CRT(NCN1,1),CMODE(NCN1,NCN2,1),SLN(1),CRTA(1),THCK(1),          -
     *CRTB(1),CRTC(1),SK(NCN1,1),RK(1),RCM(1),ISGN(1),THICK(1),NMTYPE(1)
      CALL PZERO(CRT,NCN1*2)
      CALL PZERO(CMODE,NCN1*NCN2*4)
      CALL PZERO(SLN,NCN4)
      NG1=0
      NG2=0
      NG4=0
      DO 1244 NR=1,NSREG
      NRGN=NGN(NR)
      NZR(NR)=0
      NRR(NR)=0
      NCR(NR)=0
      NSR(NR)=0
      NL=NSL(NR)
      H=THCK(NR)
      NPLY=NL
      IF(NR.GT.1) NG1=NG1+NSL(NR-1)
      NRT=0
      NCT=0
      IF(NGT(NR).EQ.0) GOTO 12345
      IF(NGT(NR).EQ.NR) GOTO 12345
      NR1=NGT(NR)
      NZR(NR)=NZR(NR1)
      NSR(NR)=NSR(NR1)
      NRR(NR)=NRR(NR1)
      NCR(NR)=NCR(NR1)
      NBCR(NR)=NBCR(NR1)
      GOTO 1244
12345 DO 1245 NSYM=1,3
      NSTREI=5*NPLY-1
      N25L=4*NSL(NR)-2
      N26=N25+N25L**2*IPR
      N27=N26+N25L**2*IPR
      N28=N27+N25L**2*IPR
      N29=N28+N25L**2*IPR
      N30=N29+N25L**2*IPR
      N31=N30+NSTREI*NSTREI*IPR
      N32=N31+NSTREI*IPR
      N33=N32+(4*NL-2)*IPR
      N34=N33+(4*NL-2)*1
      IF(N34.GT.MAXRCM) WRITE(IN6,39846)N34,MAXRCM
      IF(N34.GT.MAXRCM) STOP
      CALL DEQMAT(NSYM,NL,NEQ,NBC,NG1,NG2,NR,NMT,NSTREI,H,             -
     *RCM(1+21*NMT),RCM(1+31*NMT),RCM(1+33*NMT),RCM(1+35*NMT),         -
     *RCM(1+37*NMT),THICK  ,SLN,CRT,CMODE,RCM(N25),                    -
     *RCM(N26),RCM(N27),RCM(N28),RCM(N29),RCM(N30),RCM(N31),RCM(N32),  -
     *RCM(N33),RCM,                                                    -
     *NCN1,NCN2,NSREG,NMTYPE,NG4,N25L,IPR)
      IF(NSYM.GT.1) GOTO 2356
```

```
      NXEWQJ=4*NSL(NR)-2
      WRITE(IN6,1459)  NR
      WRITE(IN6,1458) (SLN(NG4+I),I=1,NXEWQJ)
      IF(NRGN.EQ.0)NBCR(NR)=(8*NSL(NR)-2)
      IF(NRGN.NE.0)NBCR(NR)=(8*NSL(NR)-2-6)/2+6
      NSR(NR)=NXEWQJ
      NG4=NG4+1
      I1=NG2+1
      NZT=6
      I2=NG2+NZT
      DO 191 I=I1,I2
      CRT(I,1)=0.D 00
      CRT(I,2)=0.D 00
      WRITE(IN6,1463) I,NR
      DO 191 J=1,NXEWQJ
191   WRITE(IN6,870)(CMODE(I,J,K),K=1,4)
      NZR(NR)=NZR(NR)+NZT
      NG2=NG2+NZT
2356  IF(NPLY.EQ.1) GOTO 1245
      DO 1369 NIRT=1,2
      WRITE(IN8,2357)NR,NSYM
      READ(IN7,2) NK10
      IF(NIRT.EQ.1) WRITE(IN8,1367)
      IF(NIRT.EQ.2) WRITE(IN8,1370)
      READ(IN7,2) J
      IF(J.NE.0) GOTO 1369
      IF(NIRT.NE.1) GOTO 4884
      N25L=4*NSL(NR)-2
      N26=N25+N25L**2*IPR
      N27=N26+N25L**2*IPR
      N28=N27+N25L**2*IPR
      N29=N28+N25L**2*IPR
      N30=N29+N25L**2*IPR
      N31=N30+N25L    *IPR
      N32=N31+N25L*IPR
      N33=N32+N25L**2*IPR
      IF(N33.GT.MAXRCM) WRITE(IN6,39845)N33,MAXRCM
      IF(N33.GT.MAXRCM) STOP
39845 FORMAT(1X,'N33 = ',I7,', MAXRCM = ',I7)
      CALL      RROOT(NSYM,NEQ,NT,NG2,NRGN,NK10,NRT,NCT,IN7,IN8,
     *CRT,CMODE,RCM(N25),RCM(N26),RCM(N27),RCM(N28),RCM(N29),CRTA,
     *RCM(N30),RCM(N31),RCM(N32),NCN1,NCN2,N25L)
      GOTO 4885
4884  CONTINUE
      N25L=4*NSL(NR)-2
      N26=N25+N25L**2*IPR
      N27=N26+N25L**2*IPR
      N28=N27+N25L**2*IPR
      N29=N28+N25L**2*IPR
      N30=N29+N25L**2*IPR
      N31=N30+N25L*2*IPR
      N32=N31+N25L*2*IPR
      N33=N32+N25L**2*2*IPR
      N34=N33+N25L*IPR
```

```
      IF(N34.GT.MAXRCM) WRITE(IN6,39846)N34,MAXRCM
39846 FORMAT(1X,'N34 = ,',I7,', MAXRCM = ',I7)
      IF(N34.GT.MAXRCM) STOP
      CALL     CROOT(NSYM,NEQ,NT,NG2,NRGN,NK10,NRT,NCT,IN7,IN8,
     *CRT,CMODE,RCM(N25),RCM(N26),RCM(N27),RCM(N28),RCM(N29),CRTA,
     *RCM(N30),RCM(N31),RCM(N32),RCM(N33),NCN1,NCN2,N25L)
4885  IF(NSYM.EQ.3) GOTO 3241
      IF(NIRT.EQ.1.AND.NRT.LT.1) GOTO 3241
      IF(NIRT.EQ.2.AND.NCT.LT.2) GOTO 3241
      NEVN=(NPLY+1)/2
      N2EVN=(4*NPLY-2)/2+2
      N3EVN=(4*NPLY-2)/2+1
      N4EVN=(4*NPLY-2)
      IF(NPLY/2*2.NE.NPLY.AND.NSYM.EQ.1) I3=N3EVN-1
      IF(NPLY/2*2.NE.NPLY.AND.NSYM.EQ.2) I3=N3EVN
      IF(NPLY/2*2.EQ.NPLY.AND.NSYM.EQ.1) I3=N3EVN
      IF(NPLY/2*2.EQ.NPLY.AND.NSYM.EQ.2) I3=N3EVN-1
      DO 7119 I=1,NEVN
      J=4*(NPLY+1-I)
      ISGN(J-3)= (4*I-3)*(-1.)**(NSYM)
      ISGN(J-2)= (4*I-2)*(-1.)**(NSYM+1)
      ISGN(J-5)= (4*I-1)*(-1.)**(NSYM+1)
7119  ISGN(J-4)= (4*I-0)*(-1.)**(NSYM)
      I1=NG2+1
      IF(NIRT.EQ.1) I2=NG2+NRT
      IF(NIRT.EQ.2) I2=NG2+NCT
      DO 424 I=I1,I2
      DO 424  L =1,4
      DO 7126 J=N2EVN,N4EVN
      K=ISGN(J)
      JSGN=K/IABS(K)
      K=IABS(K)
7126  CMODE(I,J,L)=JSGN*CMODE(I,K,L)
424   CMODE(I,I3,L)=0.D 00
3241  IF(NIRT.EQ.1) WRITE(IN8,1323)
      IF(NIRT.EQ.2) WRITE(IN8,1324)
      READ (IN7,2 )J
      IF(J.NE.0) GOTO 1369
      IF(NIRT.EQ.1.AND.NRT.LT.1) GOTO 1369
      IF(NIRT.EQ.2.AND.NCT.LT.2) GOTO 1369
      I1=NG2+1
      IF(NIRT.EQ.1) I2=NG2+NRT
      IF(NIRT.EQ.2) I2=NG2+NCT
      NXEWQJ=4*NSL(NR)-2
      DO 880 I3=I1,I2
      WRITE(IN6,8610) I3,CRT(I3,1),CRT(I3,2) ,NR
      DO 880 J=1,NXEWQJ
880   WRITE(IN6,870) (CMODE(I3,J,K),K=1,4)
      IF(NIRT.EQ.1) NG2=NG2+NRT
      IF(NIRT.EQ.2) NG2=NG2+NCT
      IF(NIRT.EQ.1) NRR(NR)=NRR(NR)+NRT
      IF(NIRT.EQ.2) NCR(NR)=NCR(NR)+NCT
      NRT=0
      NCT=0
```

```
 1369 CONTINUE
      WRITE(IN8,483)
      READ(IN7,2)J
      IF(J.EQ.0) GOTO 1245
      NRTNCT=NRT+NCT
      I1=1
      I2=1
      I3=1
 6736 IF(CRT(NG2+I1,2).NE.0.) GOTO 6734
      CRTA(I2+0)=CRT(NG2+I1,1)
      CRTA(I2+1)=CRT(NG2+I1,1)
      I1=I1+2
      I2=I2+2
      GOTO 6735
 6734 DO 6847 I=1,4
      CRTB(I3+I-1)=CRT(NG2+I1,1)
 6847 CRTC(I3+I-1)=CRT(NG2+I1,2)
      I3=I3+4
      I1=I1+4
 6735 IF(I1.GT.NRTNCT) GOTO 6737
      GOTO 6736
 6737 IF(NRT.LT.2) GOTO 6589
      DO 6738 I1=1,NRT
      CRT(NG2+I1,1)=CRTA(I1)
 6738 CRT(NG2+I1,2)=0.D 00
 6589 IF(NCT.LT.4) GOTO 6588
      DO 6739 I1=1,NCT
      CRT(NG2+NRT+I1,1)=CRTB(I1)
 6739 CRT(NG2+NRT+I1,2)=CRTC(I1)
 6588 CONTINUE
      DO 1906 I=1,NRT
 1906 WRITE(IN8,1908) NRT,I,CRT(NG2+I,1),CRT(NG2+I,2)
      DO 1907 I=1,NCT
 1907 WRITE(IN8,1908) NCT,I,CRT(NG2+NRT+I,1),CRT(NG2+NRT+I,2)
      WRITE(IN8,1943)
      READ(IN7,2)NRTXL
      IF(NRTXL.LT.0) NRT=0
      IF(NRTXL.GT.0) NRT=NRTXL
      WRITE(IN8,1944)
      READ(IN7,2)NCTXL
      IF(NCTXL.LT.0) NCT=0
      IF(NCTXL.GT.0) NCT=NCTXL
      WRITE(IN8,1909)
      READ(IN7,2) J
      IF(J.EQ.0) GOTO 2356
      IF(NSYM.EQ.1) NT=((NBC-1)-NCT-NRT-4+2)/2
      IF(NSYM.EQ.2) NT=((NBC-1)-NCT-NRT-2+2)/2
      IF(NSYM.EQ.3) NT=((NBC-0)-NCT-NRT-6+2)/2
      CALL PZERO(CRTA,NCN1)
      NK10=0
      N25L=4*NSL(NR)-2
      N26=N25+N25L**2*IPR
      N27=N26+N25L**2*IPR
      N28=N27+N25L**2*IPR
```

```
      N29=N28+N25L**2*IPR
      N30=N29+N25L**2*IPR
      N31=N30+N25L**2*IPR
      CALL COEFF1(NSYM,CRTA,NEQ,NCT,NRT,NRGN,NT,NG2,RCM(N31),RCM(N25),
     *RCM(N26),RCM(N27),RCM(N28),RCM(N29),CRT,NK10,NCN1,RCM(N30),N25L)
      WRITE(IN8,44844)( (I,CRTA(I)),I=1,NT)
      GOTO 2356
 1245 CONTINUE
 1244 CONTINUE
      WRITE(IN6,1654) (NSL(I),I=1,NSREG)
      WRITE(IN6,1656) (NZR(I),I=1,NSREG)
      WRITE(IN6,1657) (NRR(I),I=1,NSREG)
      WRITE(IN6,1658) (NCR(I),I=1,NSREG)
      WRITE(IN6,1659) (NBCR(I),I=1,NSREG)
      WRITE(IN6,1652) (NSR(I),I=1,NSREG)
      NTBC=0
      DO 1411 I=1,NSREG
 1411 NTBC=NTBC+NZR(I)+NRR(I)+NCR(I)
      WRITE(IN6,8603) NTBC
      CALL PZERO(SK,NCN1*NCN1)
      CALL PZERO(RK,NCN1)
      DO 1413 M1=1,NTBC
      IF(M1.EQ.1) WRITE(IN6,8606)
      M2=1
14442 READ(IN5,8605)NBCREG,NBCPLY,NBCV,YNBC,ZNBC,PNBC,NBCRG1
C     IF(M2.EQ.1) WRITE(IN6,6667)
      WRITE(IN6,8605)NBCREG,NBCPLY,NBCV,YNBC,ZNBC,PNBC,NBCRG1
      NR1=NBCREG
      NR=NGT(NR1)
      IF(NR.EQ.0) NR=NR1
      NRGN=NGN(NR)
      NPLY=NBCPLY
      NV=NBCV
      PS=PNBC
      SSSGN=1.
      SSSGN1=1.
      IF(M2.GT.1) SSSGN=PS
      IF(M2.GT.1) SSSGN1=PS
      IF(SSSGN.EQ.0.) SSSGN=1.
      IF(SSSGN1.EQ.0.) SSSGN1=1.
      IF(M2.GT.1) PS=0.D 00
      NV=IABS(NV)
      Y=YNBC
      Z=ZNBC
      H=THCK(NR)
      NL=NSL(NR)
      NV1=NV
      IF(NRGN.NE.1) GOTO 64851
      IF(DABS(Y).LT.1.E-10) GOTO 64851
      IF(NV1.NE.1) GOTO 64852
      NPLY=1
      GOTO 64851
64852 IF(NV1.NE.3) GOTO 64854
      NPLY=1
```

```
      NV=1
      GOTO 64851
64853 NPLY=1
      NV=3
      GOTO 64851
64854 IF(NV1.NE.5) GOTO 64851
      NV=5
      NPLY=1
64851 NG1=0
      NG2=0
      NG3=0
      NG4=0
      NCT=NCR(NR)
      NZT=NZR(NR)
      NRT=NRR(NR)
      NST=NSR(NR)
      DO 1414 I=1,NR1
      IF(I.EQ.1) GOTO 1414
      IF(I.GT.NR) GOTO 1414
      NG1=NG1+NSL(I-1)
      IF(NGT(I).EQ.0.OR.NGT(I).EQ.NR)NG2=NG2+NZR(I-1)+NRR(I-1)+NCR(I-1)
      IF(NGT(I).EQ.0.OR.NGT(I).EQ.NR)NG4=NG4+NSR(I-1)
      NG3=NG3+NBCR(I-1)
1414  CONTINUE
      NG1=NG1+NPLY
      IF(NRGN.NE.1) GOTO 66710
      IF(DABS(Y).LT.1.E-10)   GOTO 66710
      IF(NV1.NE.3)  GOTO 66710
      IF(NV.EQ.3) SSSGN=SSSGN1
      IF(NV .NE.1)  GOTO 66710
      ECCN=H/2
      DO 66711 ICFG54=1,NPLY
66711 ECCN=ECCN-THICK(NG1-NPLY+ICFG54)
      ECCN=ECCN+THICK(NG1)/2
      SSSGN=SSSGN1*ECCN
66710 I111=NG3+1
      I112=NG3+NBCR(NR)
      DO 1415 N1=I111,I112
      NG2=NG2+1
      IF(CRT(NG2,1).EQ.0..AND.CRT(NG2,2).EQ.0.) MT=1
      IF(CRT(NG2,1).NE.0..AND.CRT(NG2,2).EQ.0.) MT=2
      IF(CRT(NG2,1).NE.0..AND.CRT(NG2,2).NE.0.) MT=3
      NTM=IPR*NMT
      CALL FUN(NL,NPLY,NG1,NG2,NG4,NRGN,NV,MT,NMT,H,Y,Z,F,FPS,
     *RCM(1+18*NTM),RCM(1+19*NTM),RCM(1+21*NTM),RCM(1+25*NTM),
     *RCM(1+27*NTM),RCM(1+29*NTM),RCM(1+31*NTM),RCM(1+33*NTM),
     *RCM(1+35*NTM),RCM(1+37*NTM),RCM(1+39*NTM),RCM(2+39*NTM),
     *THICK  ,NMTYPE,
     *SLN,CRT,CMODE,RCM,NCN1,NCN2,IPR)
      FPS1=FPS
      SK(M1,N1)=SK(M1,N1)+SSSGN*F
C     WRITE(IN6,6666) NR,NPLY,NG1,NG2,MT,M1,N1,NV,Y,Z,F,FPS,SSSGN,PS
6667  FORMAT(/1X,'NR,NPLY,NG1,NG2,MT,M1,N1,NV,Y,Z,F,FPS,SSSGN,PS')
6666  FORMAT(1X,8I3,6E9.3)
```

```
1415  CONTINUE
      IF(NRGN.NE.1) GOTO 14140
      IF(DABS(Y).LT.1.E-10) GOTO 14140
      IF(NV1.NE.1 ) GOTO 69941
      NPLY=NPLY+1
      RK(M1)=RK(M1)-SSSGN*FPS1
      IF(M2.EQ.1.AND.NPLY.EQ.NL) RK(M1)=RK(M1)-PS
      IF(NPLY.GT.NL) GOTO 14131
      GOTO 64851
69941 IF(NV1.NE.3) GOTO 69942
      NPLY=NPLY+1
      IF(NV.NE.1) GOTO 69943
      RK(M1)=RK(M1)-SSSGN*FPS1
      IF(M2.EQ.1.AND.NPLY.EQ.NL) RK(M1)=RK(M1)-PS
      IF(NPLY.GT.NL) GOTO  64853
      GOTO 64851
69943 RK(M1)=RK(M1)-SSSGN*FPS1
      IF(M2.EQ.1.AND.NPLY.EQ.NL) RK(M1)=RK(M1)-PS
      IF(NPLY.GT.NL) GOTO 14131
      GOTO 64851
69942 IF(NV1.NE.5) GOTO 14140
      NPLY=NPLY+1
      RK(M1)=RK(M1)-SSSGN*FPS1
      IF(M2.EQ.1.AND.NPLY.EQ.NL) RK(M1)=RK(M1)-PS
      IF(NPLY.GT.NL) GOTO 14131
      GOTO 64851
14140 RK(M1)=RK(M1)-SSSGN*FPS1
      IF(M2.EQ.1) RK(M1)=RK(M1)-PS
14131 M2=M2+1
      IF(NBCRG1.EQ.0) GOTO 1413
      GOTO 14442
1413  CONTINUE
      CALL SKINV(SK,NCN1,SK,NCN1,RK,NCN1,1,SIMUL,NTBC)
386   READ (IN5,402)NDREG,NDPLY,NDV,NDVT,YND,ZND,STPND,NPTS
      IF(NDREG.EQ.0) GOTO 387
      WRITE(IN6,339)
      Y=YND
      Z=ZND
      N=1
704   FD=0.D 00
      NR1=NDREG
      NR=NGT(NR1)
      IF(NR.EQ.0) NR=NR1
      NRGN=NGN(NR)
      NPLY=NDPLY
      NV=NDV
      H=THCK(NR)
      NL=NSL(NR)
      NG1=0
      NG2=0
      NG3=0
      NG4=0
      NCT=NCR(NR)
      NZT=NZR(NR)
```

```
      NRT=NRR(NR)
      NST=NSR(NR)
      DO 373 J=1,NR1
      IF(J.EQ.1) GOTO 373
      IF(J.GT.NR) GOTO 373
      NG1=NG1+NSL(J-1)
      IF(NGT(J).EQ.0.OR.NGT(J).EQ.NR)NG2=NG2+NZR(J-1)+NRR(J-1)+NCR(J-1)
      IF(NGT(J).EQ.0.OR.NGT(J).EQ.NR)NG4=NG4+NSR(J-1)
      NG3=NG3+NBCR(J-1)
373   CONTINUE
      NG1=NG1+NPLY
      I111=NG3+1
      I112=NG3+NBCR(NR)
      DO 703  N1=I111,I112
      NG2=NG2+1
      IF(CRT(NG2,1).EQ.0..AND.CRT(NG2,2).EQ.0.) MT=1
      IF(CRT(NG2,1).NE.0..AND.CRT(NG2,2).EQ.0.) MT=2
      IF(CRT(NG2,1).NE.0..AND.CRT(NG2,2).NE.0.) MT=3
      NTM=IPR*NMT
      CALL FUN(NL,NPLY,NG1,NG2,NG4,NRGN,NV,MT,NMT,H,Y,Z,F,FPS,          -
     *RCM(1+16*NTM),RCM(1+19*NTM),RCM(1+21*NTM),RCM(1+25*NTM),          -
     *RCM(1+27*NTM),RCM(1+29*NTM),RCM(1+31*NTM),RCM(1+33*NTM),          -
     *RCM(1+35*NTM),RCM(1+37*NTM),RCM(1+39*NTM),RCM(2+39*NTM),          -
     *THICK  ,NMTYPE,                                                   -
     *SLN,CRT,CMODE,RCM,NCN1,NCN2,IPR)
      FPS1=FPS
      IF(N1.EQ.I112) FD=FD+FPS1
703   FD=FD+RK(N1)*F
      WRITE(IN6,338) Y,FD,Z,NR,NPLY,NV
      N=N+1
      IF(N.GT.1.AND.NDVT.EQ.2) Y=Y+STPND
      IF(N.GT.1.AND.NDVT.EQ.3) Z=Z+STPND
      IF(N.LE.NPTS+1) GOTO 704
      GOTO 386
387   CONTINUE
1652  FORMAT('1',/1X,' NO. OF VARIABLES IN EACH REGION      ',8(1X,I3))
1654  FORMAT(1X,' NO. OF PLYS IN EACH REGION            ',8(1X,I3))
1656  FORMAT(1X,' NO. OF ZERO ROOTS IN EACH REGION     ',8(1X,I3))
1657  FORMAT(1X,' NO. OF REAL ROOTS IN EACH REGION     ',8(1X,I3))
1658  FORMAT(1X,' NO. OF COMP. ROOTS IN EACH REGION    ',8(1X,I3))
1659  FORMAT(1X,' NO. OF B.  CONDITIONS IN EACH REGION',8(1X,I3))
1909  FORMAT(1X,'GIVE  0  IF POLYNOMIAL IS NOT DESIRED')
1943  FORMAT(1X,'GIVE  NRT  FOR POLYNOMIAL....DEFAULT IS   AS  IS'/1X,  -
     *'A NEGATIVE   VALUE  INITIATES   NRT   TO   0')
1944  FORMAT(1X,'GIVE  NCT  FOR POLYNOMIAL....DEFAULT IS   AS  IS'/1X,  -
     *'A NEGATIVE   VALUE  INITIATES   NCT   TO   0')
339   FORMAT(/1X,'Y,F,Z,NR,NPLY,NV')
8606  FORMAT('1',/1X,'NBCREG,NBCPLY,NBCV,YNBC,ZNBC,PNBC,NBCREG')
8603  FORMAT(/1X,I4,' = NTBC')
483   FORMAT(/1X,'GIVE 0   IF NO POLYNOMIAL DESIRED')
1324  FORMAT(1X,'GIVE  0   IF COMPLEX ROOTS ARE TO BE STORED')
1370  FORMAT(1X,'GIVE  0   IF YOU WANT COMPLEX ROOTS')
8610  FORMAT(/1X,I4,1X,2D15.8,' EIGEN VALUE AND MODE FOR REGION = ',I3)
870   FORMAT(6(1X,F17.6))
```

```
 1323  FORMAT(1X,'GIVE  0   IF REAL ROOTS ARE TO BE STORED')
 1367  FORMAT(1X,'GIVE  0   IF YOU WANT REAL   ROOTS')
 2357  FORMAT(1X,'GIVE  0   IF YOU DO NOT WANT POLYNOMIAL APPROX.'/          -
      *1X,'NR =  ',I3,'  NSYM = ',I3)          -
 1463  FORMAT(/1X,I4,' ZERO MODE,    ',I3,' REGION')
 1459  FORMAT('1',/1X,'PARTICULAR SOLUTION FOR REGION = ',I2)
 8605  FORMAT(3I4,3D12.6,3I4,2D12.6)
 402   FORMAT(4I4,3D12.6,I4,I4)
 338   FORMAT(3(1X,F17.6),3(1X,I3,1X))
 1908  FORMAT(1X,2(I3,1X),2(D15.8,1X))
 44844 FORMAT(I3,2X,D23.16)
 1458  FORMAT(10(1X,D10.4))
 123   FORMAT(8(1X,D9.3))
 12    FORMAT(6(1X,E10.4),/3(1X,E10.4))
 435   FORMAT(8D10.3)
 2     FORMAT(20I2)
       RETURN
       END
       SUBROUTINE  MPRO1(E11,E22,E33,NU12,NU13,NU23,G44,G55,G66,S31,S32, -
      *S33,S44,S45,S55,CB1,CB2,CB6,CB11,CB12,CB16,CB22,CB26,CB66,SNC1,   -
      *SNC2,SHM1,SHM2,SHQ1,SHQ2,KQ1,KQ2,KM1,KM2,KP1,KP2,KN1,KN2,THETA,   -
      *N,IN5,IN6,IN7,IN8)
       IMPLICIT REAL   (A-H,O-Z)
       REAL   KM1,KM2,KN1,KN2,KQ1,KQ2,KP1,KP2,NU12,NU13,NU23
       DIMENSION E11(1),E22(1),E33(1),NU12(1),NU13(1),NU23(1),G44(1),    -
      *G55(1),G66(1),S31(1),S32(1),S33(1),S44(1),S45(1),S55(1),CB1(1),   -
      *CB2(1),CB6(1),CB11(1),CB12(1),CB16(1),CB22(1),CB26(1),CB66(1),    -
      *SNC1(1),SNC2(1),SHM1(1),SHM2(1),SHQ1(1),SHQ2(1),KQ1(1),KQ2(1),    -
      *KM1(1),KM2(1),KP1(1),KP2(1),KN1(1),KN2(1),THETA(1)                -
      *,T1(6,6),T2(6,6),T1I(6,6),ST(6,6),SF(6,6),VC6(6)
C.....LAYER FLEXIBILITI MATRICES
       PI=3.141592653589793
       WRITE(IN6,149)
 149   FORMAT('1')
       WRITE(IN6,5)
       DO 1242 K=1,N
       READ(IN5,4) E11(K),E22(K),E33(K),NU12(K),NU13(K),NU23(K),G44(K),  -
      *G55(K),G66(K),THETA(K)
 1242  WRITE(IN6,123)E11(K),E22(K),E33(K),NU12(K),NU13(K),NU23(K),G44(K),-
      *G55(K),G66(K),THETA(K)
 5     FORMAT(/1X,'E11,E22,E33,NU12,NU13,NU23,G44,G55,G66,THETA')
 4     FORMAT(10F8.4)
       DO 101 K=1,N
       CALL PZERO(ST,36)
       ST(1,1)=1./E11(K)
       ST(1,2)=1./E11(K)*NU12(K)*(-1)
       ST(1,3)=1./E11(K)*NU13(K)*(-1)
       ST(2,1)=ST(1,2)
       ST(2,2)=1./E22(K)
       ST(2,3)=1./E22(K)*NU23(K)*(-1)
       ST(3,1)=ST(1,3)
       ST(3,2)=ST(2,3)
       ST(3,3)=1./E33(K)
       ST(4,4)=1./G44(K)
```

```
      ST(5,5)=1./G55(K)
      ST(6,6)=1./G66(K)
      CALL PZERO(T1,36)
      CO=DCOS(THETA(K)*PI/180)
      SI=DSIN(THETA(K)*PI/180)
      T1(1,1)=CO*CO
      T1(1,2)=SI*SI
      T1(1,6)=-2.*CO*SI
      T1(2,1)=SI*SI
      T1(2,2)=CO*CO
      T1(2,6)=2*CO*SI
      T1(3,3)=1
      T1(4,4)=CO
      T1(4,5)=SI
      T1(5,4)=-SI
      T1(5,5)=CO
      T1(6,1)=CO*SI
      T1(6,2)=-CO*SI
      T1(6,6)=CO*CO-SI*SI
      CALL PZERO(T2,36)
      CALL PZERO(SF,36)
      CALL SKINV(T1,6,T2,6,VC6,6,-1,SIMUL,6)
      CALL BIDOT(ST,6,T2,6,SF,6)
      DO 102 I=1,6
      DO 102 J=1,6
102   T2(I,J)=T1(I,J)
      T2(6,1)=2.*CO*SI
      T2(6,2)=-2.*CO*SI
      T2(1,6)=-CO*SI
      T2(2,6)=CO*SI
      CALL BIDOT(T2,6,SF,6,ST,6)
      WRITE(IN6,105)K
105   FORMAT(/1X,'FLEXIBILITY MATRIX S FOR ',I2,' MATERIAL')
      DO 1010 I=1,6
1010  WRITE(IN6,123)ST(I,1),ST(I,2),ST(I,3),ST(I,4),ST(I,5),ST(I,6)
C.....LAYER STIFFNESS MATRICES....C
      CALL SKINV(ST,6,SF,6,VC6,6,-1,SIMUL,6)
      WRITE(IN6,112) K
112   FORMAT(/1X,'STIFFNESS MATRIX C FOR ',I2,' MATERIAL')
      DO 108 I=1,6
108   WRITE(IN6,123)SF(I,1),SF(I,2),SF(I,3),SF(I,4),SF(I,5),SF(I,6)
C.....PLANESTRESS STIFFNESS MATRIX
      CALL PZERO(T2,36)
      CALL PZERO(T1,36)
      T1(1,1)=ST(1,1)
      T1(1,2)=ST(1,2)
      T1(1,3)=ST(1,6)
      T1(2,1)=ST(2,1)
      T1(2,2)=ST(2,2)
      T1(2,3)=ST(2,6)
      T1(3,1)=ST(6,1)
      T1(3,2)=ST(6,2)
      T1(3,3)=ST(6,6)
      T1(4,4)=1.
```

```
      T1(5,5)=1.
      T1(6,6)=1.
      CALL SKINV(T1,6,T2,6,VC6,6,-1,SIMUL,6)
      CALL PZERO(T1,36)
      T1(1,1)=-ST(1,3)
      T1(2,1)=-ST(2,3)
      T1(3,1)=-ST(6,3)
      CALL BIDOT(T2,6,T1,6,T1I,6)
      WRITE(IN6,115)
115   FORMAT(/1X,'PLANESTRESS STIFFNESS MATRIX CBB, CB OF ',I2,' LAYER')
      DO 113 I=1,3
113   WRITE(IN6,123) T2(I,1),T2(I,2),T2(I,3),T1I(I,1)
C.....MATERIAL PROPERTIES FOR PLY LEVEL ANALYSIS
      S31(K)=ST(3,1)
      S32(K)=ST(3,2)
      S33(K)=ST(3,3)
      S44(K)=ST(4,4)
      S45(K)=ST(4,5)
      S55(K)=ST(5,5)
      CB1(K)=T1I(1,1)
      CB2(K)=T1I(2,1)
      CB6(K)=T1I(3,3)
      CB11(K)=T2(1,1)
      CB12(K)=T2(1,2)
      CB16(K)=T2(1,3)
      CB22(K)=T2(2,2)
      CB26(K)=T2(2,3)
      CB66(K)=T2(3,3)
      SNC1(K)=CB12(K)/CB22(K)*(S31(K)*CB12(K)+S32(K)*CB22(K))+CB12(K)*S4-
     *4(K)-CB1(K)
      SNC2(K)=CB22(K)/CB22(K)*(S31(K)*CB12(K)+S32(K)*CB22(K))+CB22(K)*S4-
     *4(K)-CB2(K)
      KP1(K)=CB1(K)/2
      KP2(K)=CB2(K)/2
      SHM2(K)=S44(K)*(-.25)
      SHQ2(K)=S44(K)*.75
      KQ1(K)=(.25*CB12(K)*(S31(K)*CB12(K)/CB22(K) +S32(K))-CB12(K)*S44(K-
     *)+CB1(K))*.6
      KQ2(K)=(.25*CB22(K)*(S31(K)*CB12(K)/CB22(K) +S32(K))-CB22(K)*S44(K-
     *)+CB2(K))*.6
      KM1(K)=(1.5*CB12(K)*(S31(K)*CB12(K)/CB22(K) +S32(K))-CB12(K)*S44(K-
     *)+CB1(K))*.1
      KM2(K)=(1.5*CB22(K)*(S31(K)*CB12(K)/CB22(K) +S32(K))-CB22(K)*S44(K-
     *)+CB2(K))*.1
      KN1(K)=(CB12(K)*(S31(K)*CB12(K)/CB22(K) +S32(K))+CB12(K)*S44(K)+2*-
     *CB1(K))/12
101   KN2(K)=(CB22(K)*(S31(K)*CB12(K)/CB22(K) +S32(K))+CB22(K)*S44(K)+2*-
     *CB2(K))/12
      WRITE(IN6,125)
125   FORMAT(/1X,'CB11,CB12,CB16,CB22,CB26,CB66,CB1,CB2,CB6')
123   FORMAT(1X,8E10.4)
      DO 126 K=1,N
126   WRITE(IN6,123)CB11(K),CB12(K),CB16(K),CB22(K),CB26(K),CB66(K),    -
     *CB1(K),CB2(K),CB6(K)
```

```
      WRITE(IN6,129)
129   FORMAT(/1X,'S31,S32,S33,S44,S45,S55,SNC1,SNC2,SHM2,SHQ2')
      DO 130 K=1,N
130   WRITE(IN6,123)S31(K),S32(K),S33(K),S44(K),S45(K),S55(K),SNC1(K),  -
     *SNC2(K),SHM2(K),SHQ2(K)
      WRITE(IN6,119)
119   FORMAT(/1X,'KN1,KN2,KP1,KP2,KM1,KM2,KQ1,KQ2')
      DO 122 K=1,N
122   WRITE(IN6,123) KN1(K),KN2(K),KP1(K),KP2(K),KM1(K),KM2(K),KQ1(K),  -
     *KQ2(K)
      RETURN
      END
      FUNCTION WVD(N,I,Z,J,RCM,IPR)
      IMPLICIT REAL   (A-H,O-Z)
      REAL   RCM
      DIMENSION RCM(1)
      M=J*IPR
      WVD= WVD1(N,I,Z,  RCM(1),RCM(1+M),RCM(1+2*M),RCM(1+3*M),RCM(1+4*M)-
     *,RCM(1+5*M),RCM(1+6*M),RCM(1+7*M),RCM(1+8*M),RCM(1+9*M)           -
     *,RCM(1+10*M),RCM(1+11*M),RCM(1+12*M),RCM(1+13*M),RCM(1+14*M)      -
     *,RCM(1+15*M),RCM(1+16*M),RCM(1+17*M),RCM(1+18*M),RCM(1+19*M)      -
     *,RCM(1+20*M),RCM(1+21*M),RCM(1+22*M),RCM(1+23*M),RCM(1+24*M)      -
     *,RCM(1+25*M),RCM(1+26*M),RCM(1+27*M),RCM(1+28*M),RCM(1+29*M)      -
     *,RCM(1+30*M),RCM(1+31*M),RCM(1+32*M),RCM(1+33*M),RCM(1+34*M)      -
     *,RCM(1+35*M),RCM(1+36*M),RCM(1+37*M),RCM(1+39*M),RCM(2+39*M))
      RETURN
      END
      FUNCTION WVD1(N,I,Z,  E11,E22,E33,NU12,NU13,NU23,G44,G55,G66,S31, -
     *S32,S33,S44,S45,S55,CB1,CB2,CB6,CB11,CB12,CB16,CB22,CB26,CB66,SNC1-
     *,SNC2,SHM1,SHM2,SHQ1,SHQ2,KQ1,KQ2,KM1,KM2,KP1,KP2,KN1,KN2,EPSLON, -
     *CURVTR)
      IMPLICIT REAL   (A-H,O-Z)
      REAL   KM1,KM2,KN1,KN2,KQ1,KQ2,KP1,KP2,NU12,NU13,NU23
      DIMENSION E11(1),E22(1),E33(1),NU12(1),NU13(1),NU23(1),G44(1),    -
     *G55(1),G66(1),S31(1),S32(1),S33(1),S44(1),S45(1),S55(1),CB1(1),   -
     *CB2(1),CB6(1),CB11(1),CB12(1),CB16(1),CB22(1),CB26(1),CB66(1),    -
     *SNC1(1),SNC2(1),SHM1(1),SHM2(1),SHQ1(1),SHQ2(1),KQ1(1),KQ2(1),    -
     *KM1(1),KM2(1),KP1(1),KP2(1),KN1(1),KN2(1)
      GOTO(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18),N
1     WVD1=1
      RETURN
2     WVD1     =-(S31(I)*CB12(I)+S32(I)*CB22(I))/2*Z*Z
      RETURN
3     WVD1    =(S31(I)*CB11(I)+S32(I)*CB12(I))*(Z*EPSLON-Z*Z/2*CURVTR)
C.....HERE Z SHOULD HAVE A DIMENSION OF A LENGTH AND W,XX SHOULD
C.....BE GIVEN AS IS WITHOUT ANY NONDIMENSIONALIZATION
      RETURN
4     WVD1     =(S31(I)*CB12(I)+S32(I)*CB22(I))*Z
      RETURN
5     WVD1     =(S31(I)*SNC1(I)+S32(I)*SNC2(I))/12*(Z*Z*Z-Z)            -
     *-S33(I)/12*(Z**3-3*Z)+(S31(I)*KN1(I)+S32(I)*KN2(I))*Z
      RETURN
6     WVD1     =(S31(I)*SNC1(I)+S32(I)*SNC2(I))/4*(Z**4/4-3./10*Z*Z)    -
     *-S33(I)/4*(Z**4/4-Z*Z/2)+(S31(I)*KM1(I)+S32(I)*KM2(I))/2*Z*Z
```

```
      RETURN
 7    WVD1    =(S31(I)*KP1(I)+S32(I)*KP2(I))*Z+S33(I)*Z/2
      RETURN
 8    WVD1 =(S31(I)*KQ1(I)+S32(I)*KQ2(I))/2*Z*Z-S33(I)/4*(Z**4/4-Z*Z/2) -
     *+(S31(I)*SNC1(I)+S32(I)*SNC2(I))/4*(Z**4/4-Z*Z*3/10)+S33(I)/4*Z*Z
      RETURN
 9    WVD1    =-Z
      RETURN
 10   WVD1    =(S31(I)*CB12(I)+S32(I)*CB22(I))/6*Z**3+S44(I)*CB22(I)/6*(-
     *Z**3-3*Z)
      RETURN
 11   WVD1    =1
      RETURN
 12   WVD1    =-S31(I)*CB12(I)/2*Z*Z
      RETURN
 13   WVD1    =(S44(I)+S32(I))/4*Z*Z
      RETURN
 14   WVD1    =-S31(I)*KN1(I)/2*Z*Z                                      -
     *-(S31(I)*SNC1(I)+S32(I)*SNC2(I))/12*(Z**4/4-Z*Z/2)                 -
     *+S33(I)/12*(Z**4/4-3./2*Z*Z)+0*S44(I)/4*SNC2(I)*(-Z**4/12+Z**2/6)
      RETURN
 15   WVD1    =S44(I)/2*Z
      RETURN
 16   WVD1    =-(S31(I)*KM1(I)+S32(I)*KM2(I))/6*Z**3                     -
     *          -S44(I)*KM2(I)/6*(Z**3-3*Z)                              -
     *          -(S31(I)*SNC1(I)+S32(I)*SNC2(I))/4*(Z**5/20-Z**3/10)     -
     *          +S33(I)/4*(Z**5/20-Z**3/6)                               -
     *        +0*S44(I)/4*SNC2(I)*(-Z**5/20+Z**3/10-Z/20)
      RETURN
 17   WVD1    =-S31(I)*KP1(I)*Z*Z/2-S33(I)*Z*Z/4
      RETURN
 18   WVD1    =-(S31(I)*KQ1(I)+S32(I)*KQ2(I))/6*Z**3                     -
     *          -S44(I)*KQ2(I)/6*(Z**3-3*Z)                              -
     *          -(S31(I)*SNC1(I)+S32(I)*SNC2(I))/4*(Z**5/20-Z**3/10)     -
     *          -S33(I)*Z**3/12                                          -
     *          +S33(I)*(Z**5/20-Z**3/6)/4                               -
     *        +0*S44(I)/4*SNC2(I)*(-Z**5/20+Z**3/10-Z/20)
      RETURN
      END
      SUBROUTINE CDET(NSYM,COEF,X,F,NSDFG,CVC30,NPIVOT,NK10,             -
     *N,NRT,NCT,CRT,NRGN,NT,NG2,SK0,SK1,SK2,SK3,SK4,A,WA,NCN1,N25L)
      IMPLICIT REAL   (A-H,O-Z)
      COMPLEX    A(N25L,1),CVC30(1),X,F,X2347,DCMPLX
      DIMENSION WA(1),COEF(1),SK0(N25L,1),SK1(N25L,1),SK2(N25L,1),       -
     *SK3(N25L,1),SK4(N25L,1),CRT(NCN1,1)
      IF(NSDFG.NE.0) GOTO 8
      IF(NK10.EQ.0) GOTO 8
      F=(0.D 00,0.D 00)
      DO 80 I=1,NT
 80   F=F+COEF(I)*X**(2*I-2)
      RETURN
 8    DO 2 I=1,N
      DO 2 J=1,N
 2    A(I,J)=SK0(I,J)+SK1(I,J)*X+SK2(I,J)*X*X +SK3(I,J)*X*X*X            -
```

```
     *+SK4(I,J)*X*X*X*X
      IF(NSDFG.NE.0) GOTO 3
      CALL LEQT1C(A,N,N25L,CVC30,1,N25L,0,WA,IER)
      F=(1.D 00,0.D 00)
      DO 1 I=1,N
      IPVT=WA(I)
      IF(IPVT.NE.I) F=-F
1     F=F*A(I,I)
      IF(NPIVOT.EQ.11000) RETURN
      Y=1.D 00
      NRTNCT=NRT+NCT
      IF(NRTNCT.LT.2) GOTO 232
      I1=1
215   IF(CRT(NG2+I1,2).NE.0.) GOTO 231
      Y=Y*(X**2-CRT(NG2+I1,1)**2)
      I1=I1+2
      GOTO 214
231   Y=Y*((X-CRT(NG2+I1,1))**2+CRT(NG2+I1,2)**2)
      Y=Y*((X+CRT(NG2+I1,1))**2+CRT(NG2+I1,2)**2)
      I1=I1+4
214   IF(I1.GT.NRTNCT) GOTO 232
      GOTO 215
232   F=F/Y
      IF(NSYM.EQ.1) F=F/X**5
      IF(NSYM.EQ.2) F=F/X**3
      IF(NSYM.EQ.3) F=F/X**6
      RETURN
3     CVC30(NPIVOT)=(1.D 00,0.D 00)
      A(NPIVOT,NPIVOT)=(1.D 00,0.D 00)
      DO 4 I=1,N
      IF(I.NE.NPIVOT) CVC30(I)=-A(I,NPIVOT)
      IF(I.NE.NPIVOT) A(NPIVOT,I)=(0.D 00,0.D 00)
4     IF(I.NE.NPIVOT) A(I,NPIVOT)=(0.D 00,0.D 00)
      CALL LEQT1C(A,N,N25L,CVC30,1,N25L,0,WA,IER)
      F=(1.D 00,0.D 00)
      DO  71 I=1,N
      IPVT=WA(I)
      IF(IPVT.NE.I) F=-F
71    F=F*A(I,I)
      RETURN
      END
      SUBROUTINE FDET(NSYM,COEF,X,F,NSDFG,VC30,NPIVOT,NK10,NCN1,N,NRT,
     *NCT,CRT,NRGN,NT,NG2,SK0,SK1,SK2,SK3,SK4,A,N25L)
      IMPLICIT REAL   (A-H,O-Z)
      DIMENSION CRT(NCN1,1),SK0(N25L,1),SK1(N25L,1),SK2(N25L,1),
     *SK3(N25L,1),SK4(N25L,1),A(N25L,1),COEF(1),VC30(1)
      IF(NSDFG.NE.0) GOTO 8
      IF(NK10.EQ.0) GOTO 8
      F=0.D 00
      DO 80 I=1,NT
80    F=F+COEF(I)*X**(2*I-2)
      RETURN
8     DO 2 I=1,N
      DO 2 J=1,N
```

```
2     A(I,J)=SK0(I,J)+SK1(I,J)*X+SK2(I,J)*X*X+SK3(I,J)*X*X*X           -
     *+SK4(I,J)*X*X*X*X
      IF(NSDFG.NE.0) GOTO 3
      CALL SKINV(A,N25L,A,N25L,VC30,1,-1,F,N)
      IF(NPIVOT.EQ.11000) RETURN
      Y=1.D 00
      NRTNCT=NRT+NCT
      IF(NRTNCT.LT.2) GOTO 232
      I1=1
215   IF(CRT(NG2+I1,2).NE.0.) GOTO 231
      Y=Y*(X**2-CRT(NG2+I1,1)**2)
      I1=I1+2
      GOTO 214
231   Y=Y*((X-CRT(NG2+I1,1))**2+CRT(NG2+I1,2)**2)
      Y=Y*((X+CRT(NG2+I1,1))**2+CRT(NG2+I1,2)**2)
      I1=I1+4
214   IF(I1.GT.NRTNCT) GOTO 232
      GOTO 215
232   F=F/Y
      IF(NSYM.EQ.1) F=F/X**5
      IF(NSYM.EQ.2) F=F/X**3
      IF(NSYM.EQ.3) F=F/X**6
      RETURN
3     VC30(NPIVOT)=1.D 00
      A(NPIVOT,NPIVOT)=1.D 00
      DO 4 I=1,N
      IF(I.NE.NPIVOT) VC30(I)=-A(I,NPIVOT)
      IF(I.NE.NPIVOT) A(NPIVOT,I)=0.D 00
4     IF(I.NE.NPIVOT) A(I,NPIVOT)=0.D 00
      CALL SKINV(A,N25L,A,N25L,VC30,1,0,F,N)
      RETURN
      END
      SUBROUTINE COEFF1(LJ,COEF,N,NCT,NRT,NRGN,NT,NG2,SK,SK0,         -
     *SK1,SK2,SK3,SK4,CRT,NK10,NCN1,A,N25L)
      IMPLICIT REAL    (A-H,O-Z)
      DIMENSION SK(N25L,1),COEF(1),SK0(N25L,1),SK1(N25L,1),SK2(N25L,1), -
     *SK3(N25L,1),SK4(N25L,1),CRT(NCN1,1),A(N25L,1)
      DO 6 I=1,NT
      Y=DFLOAT(I)
      X=1.D 00
      IF(LJ.EQ.2) X=X*Y*Y*Y
      IF(LJ.EQ.1) X=X*Y*Y*Y*Y*Y
      IF(LJ.EQ.3) X=X*Y*Y*Y*Y*Y*Y
      CALL FDET(LJ,COEF,Y,FI,0,COEF,11000,NK10,NCN1,                   -
     *N,NRT,NCT,CRT,NRGN,NT,NG2,SK0,SK1,SK2,SK3,SK4,A,N25L)
      NRTNCT=NRT+NCT
      IF(NRTNCT.LT.2) GOTO 232
      I1=1
215   IF(CRT(NG2+I1,2).NE.0.) GOTO 231
      X=X*(Y**2-CRT(NG2+I1,1)**2)
      I1=I1+2
      GOTO 214
231   X=X*((Y-CRT(NG2+I1,1))**2+CRT(NG2+I1,2)**2)
      X=X*((Y+CRT(NG2+I1,1))**2+CRT(NG2+I1,2)**2)
```

```
      I1=I1+4
214   IF(I1.GT.NRTNCT) GOTO 232
      GOTO 215
232   COEF(I)=FI/X
      DO 6 J=1,NT
6     SK(I,J)=  Y**(2*J-2)
      CALL SKINV(SK,N25L,SK,N25L,COEF,1,0,SIMUB,NT)
      RETURN
      END
      SUBROUTINE CROOT(NSYM,N,NT,NG2,NRGN,NK10,NRT,NCT,IN7,IN8,          -
     *CRT,CMODE,SK0,SK1,SK2,SK3,SK4,COEF,CVC31,CVC32,A,WA,NCN1,NCN2,N25L-
     *)
      IMPLICIT REAL   (A-H,O-Z)
      DIMENSION CRT(NCN1,1),CMODE(NCN1,NCN2,1),SK0(N25L,1),SK1(N25L,1), -
     *SK2(N25L,1),SK3(N25L,1),SK4(N25L,1),COEF(1),WA(1)
      COMPLEX    XF12,FF1,FF2,XI,XF,FI,CVC31(1),CVC32(1),XRL,XIM,       -
     *A(N25L,1), DCMPLX
C.....LIMIT MAX. NO. OF ITERATIONS TO 100, AND ROOTS TO 100
      XRL=(1.,0.)
      XIM=(0.,-1.)
      DO 4 I=1,100
4011  WRITE(IN8,5)
5     FORMAT(1X,'2F10.6....XI..INPUT INITIAL COMPLEX ROOT VALUE')
      READ(IN7,6) XI
      IF(CABS(XI).LE.1.D-20) GOTO 14
6     FORMAT(10F10.6)
      NITER=0
9     CALL CDET(NSYM,COEF,XI,FI,0,CVC31,1,NK10,                         -
     *N,NRT,NCT,CRT,NRGN,NT,NG2,SK0,SK1,SK2,SK3,SK4,A,WA,NCN1,N25L)
      RFI=FI*XRL
      RGI=FI*XIM
      UI=XI*XRL
      VI=XI*XIM
      XI=CMPLX(   1.00000000001*UI,VI)
      CALL CDET(NSYM,COEF,XI,FI,0,CVC31,1,NK10,                         -
     *N,NRT,NCT,CRT,NRGN,NT,NG2,SK0,SK1,SK2,SK3,SK4,A,WA,NCN1,N25L)
      RFU=FI*XRL
      RGU=FI*XIM
      RFU=(RFU-RFI)/.00000000001/UI
      RGU=(RGU-RGI)/.00000000001/UI
      XI=CMPLX(UI,1.00000000001*VI)
      CALL CDET(NSYM,COEF,XI,FI,0,CVC31,1,NK10,                         -
     *N,NRT,NCT,CRT,NRGN,NT,NG2,SK0,SK1,SK2,SK3,SK4,A,WA,NCN1,N25L)
      RFV=FI*XRL
      RGV=FI*XIM
      RFV=(RFV-RFI)/.00000000001/VI
      RGV=(RGV-RGI)/.00000000001/VI
      UF=UI-(RFI*RGV-RFV*RGI)/(RFU*RGV-RFV*RGU)
      VF=VI-(RFU*RGI-RFI*RGU)/(RFU*RGV-RFV*RGU)
      NITER=NITER+1
      WRITE(IN8,12)NITER,XI,FI
      IF(CABS(XI/CMPLX(UF,VF)-1.).LE..000000001) GOTO 7
      IF(NITER.GT.100) GOTO 4011
      XI=CMPLX(UF,VF)
```

```
      GOTO 9
7     WRITE(IN8,12)I,XI,FI
      WRITE(IN8,76)
      READ(IN7,17) J
17    FORMAT(20I2)
      IF(J.NE.0) GOTO 77
      GOTO 4011
12    FORMAT(1X,I3,1X,4D15.8,' I,XI,FI')
77    DO 7712 IIII=1,4
      CRT(NG2+NRT+NCT+IIII,1)=UF
7712  CRT(NG2+NRT+NCT+IIII,2)=VF
      NCT=NCT+4
76    FORMAT(1X,'I2...INPUT...J=1 TO STORE ROOT')
4     CONTINUE
14    WRITE(IN8,21)
21    FORMAT(1X,'I2...INPUT J=0 TO FIND COMPLEX EIGEN MODES')
      READ(IN7,17) J
      IF(J.NE.0) GOTO 86
      III=0
      DO 82 I=1,NCT,4
      III=III+1
      X1=CRT(NG2+I,1)
      X2=CRT(NG2+I,2)
      DO 821 J1=1,4,2
      IF(J1.NE.1) GOTO 832
      XI=CMPLX( X1, X2)
      CALL CDET(NSYM,COEF,XI,FI,1,CVC31,1,NK10,
     *N,NRT,NCT,CRT,NRGN,NT,NG2,SK0,SK1,SK2,SK3,SK4,A,WA,NCN1,N25L)
861   FORMAT(/3(1X,I4),1X,2D15.8,1X,'COMPLEX EIGEN VALUE AND MODE')
832   IF(J1.NE.3) GOTO 821
      XI=CMPLX(-X1, X2)
      CALL CDET(NSYM,COEF,XI,FI,1,CVC32,1,NK10,
     *N,NRT,NCT,CRT,NRGN,NT,NG2,SK0,SK1,SK2,SK3,SK4,A,WA,NCN1,N25L)
87    FORMAT(10(1X,D10.4))
821   CONTINUE
      NCT12=NCT/2
      IF(NRGN.EQ.0) WRITE(IN8,861) NG2,NCT,I,XI
      IF(NRGN.EQ.1) WRITE(IN8,861) NG2,NCT12,III,XI
      IF(NRGN.EQ.1) CRT(NG2+2*III-1,1)=X1
      IF(NRGN.EQ.1) CRT(NG2+2*III-1,2)=X2
      IF(NRGN.EQ.1) CRT(NG2+2*III-0,1)=X1
      IF(NRGN.EQ.1) CRT(NG2+2*III-0,2)=X2
      DO 7634 J=1,N
7634  WRITE(IN8,87) CVC31(J),CVC32(J)
      DO 823 J=1,N
      RCVC31=CVC31(J)*XRL
      ACVC31=CVC31(J)*XIM
      RCVC32=CVC32(J)*XRL
      ACVC32=CVC32(J)*XIM
      IF(NRGN.EQ.1) GOTO 76341
      CMODE(NG2+I+0,J,1)=( ACVC31+ACVC32)/2.
      CMODE(NG2+I+0,J,2)=( RCVC31-RCVC32)/2.
      CMODE(NG2+I+0,J,3)=( ACVC31-ACVC32)/2.
      CMODE(NG2+I+0,J,4)=( RCVC31+RCVC32)/2.
```

```
      CMODE(NG2+I+1,J,1)=( ACVC31-ACVC32)/2.
      CMODE(NG2+I+1,J,2)=( RCVC31+RCVC32)/2.
      CMODE(NG2+I+1,J,3)=( ACVC31+ACVC32)/2.
      CMODE(NG2+I+1,J,4)=( RCVC31-RCVC32)/2.
      CMODE(NG2+I+2,J,1)=( RCVC31+RCVC32)/2.
      CMODE(NG2+I+2,J,2)=(-ACVC31+ACVC32)/2.
      CMODE(NG2+I+2,J,3)=( RCVC31-RCVC32)/2.
      CMODE(NG2+I+2,J,4)=(-ACVC31-ACVC32)/2.
      CMODE(NG2+I+3,J,1)=( RCVC31-RCVC32)/2.
      CMODE(NG2+I+3,J,2)=(-ACVC31-ACVC32)/2.
      CMODE(NG2+I+3,J,3)=( RCVC31+RCVC32)/2.
      CMODE(NG2+I+3,J,4)=(-ACVC31+ACVC32)/2.
      GOTO 823
76341 CONTINUE
      CMODE(NG2+2*III-1,J,1)= RCVC31
      CMODE(NG2+2*III-1,J,2)=-ACVC31
      CMODE(NG2+2*III-1,J,3)=0.D 00
      CMODE(NG2+2*III-1,J,4)=0.D 00
      CMODE(NG2+2*III-0,J,1)= ACVC31
      CMODE(NG2+2*III-0,J,2)= RCVC31
      CMODE(NG2+2*III-0,J,3)=0.D 00
      CMODE(NG2+2*III-0,J,4)=0.D 00
823   CONTINUE
82    CONTINUE
      IF(NRGN.EQ.1) NCT=NCT/2
86    RETURN
      END
      SUBROUTINE RROOT(NSYM,N,NT,NG2,NRGN,NK10,NRT,NCT,IN7,IN8,
     *CRT,CMODE,SK0,SK1,SK2,SK3,SK4,COEF,VC31,VC32,A,NCN1.NCN2,N25L)
      IMPLICIT REAL   (A-H,O-Z)
      DIMENSION CRT(NCN1,1),CMODE(NCN1,NCN2,1),SK0(N25L,1),SK1(N25L,1),
     *SK2(N25L,1),SK3(N25L,1),SK4(N25L,1),COEF(1),VC31(1),VC32(1)
     *,A(N25L,1)
C6390  FORMAT(1X,I2,14F9.4)
C      NQQ=N
C      DO 6391 I=1,NQQ
C6391  WRITE(6,6390) I,(SK0(I,J),J=1,NQQ)
C      DO 6392 I=1,NQQ
C6392  WRITE(6,6390) I,(SK1(I,J),J=1,NQQ)
C      DO 6393 I=1,NQQ
C6393  WRITE(6,6390) I,(SK2(I,J),J=1,NQQ)
C      DO 6394 I=1,NQQ
C6394  WRITE(6,6390) I,(SK3(I,J),J=1,NQQ)
C      DO 6395 I=1,NQQ
C6395  WRITE(6,6390) I,(SK4(I,J),J=1,NQQ)
C.....LIMIT MAX. NO. OF ROOTS TO 100, AND ITERATIONS TO 100
      DO 4 I=1,100
4011  WRITE(IN8,5)
5     FORMAT(1X,'10F10.6....XI..INPUT INITIAL ROOT VALUE')
      READ(IN7,6) XI
      IF(DABS(XI).LE.1.D-20) GOTO 14
6     FORMAT(10F10.6)
      NITER=0
9     CALL FDET(NSYM,COEF,XI,FI,0,VC31,1,NK10,NCN1,
```

```
     *N,NRT,NCT,CRT,NRGN,NT,NG2,SK0,SK1,SK2,SK3,SK4,A,N25L)
      XF1=(1.0+0.00000000001)*XI
      CALL FDET(NSYM,COEF,XF1,FF1,0,VC31,1,NK10,NCN1,
     *N,NRT,NCT,CRT,NRGN,NT,NG2,SK0,SK1,SK2,SK3,SK4,A,N25L)
      XF2=(1.0-0.00000000001)*XI
      CALL FDET(NSYM,COEF,XF2,FF2,0,VC31,1,NK10,NCN1,
     *N,NRT,NCT,CRT,NRGN,NT,NG2,SK0,SK1,SK2,SK3,SK4,A,N25L)
      DERXI=(FF1-FF2)/(XF1-XF2)
      XF=XI-FI/DERXI
      NITER=NITER+1
      WRITE(IN8,12)NITER,XI,FI
      IF(DABS(XI/XF-1.).LE..000000001) GOTO 7
      IF(NITER.GT.100) GOTO 4011
      XI=XF
      GOTO 9
7     WRITE(IN8,12)I,XI,FI
12    FORMAT(1X,I3,1X,2(D15.8,1X),' I,XI,FI')
      WRITE(IN8,76)
      READ(IN7,17)  J
17    FORMAT(20I2)
      IF(J.NE.0) GOTO 77
      GOTO 4011
77    DO 7712 IIII=1,2
      CRT(NG2+NRT+NCT+IIII,2)=0.D 00
7712  CRT(NG2+NRT+NCT+IIII,1)=XI
      NRT=NRT+2
76    FORMAT(1X,'I2...INPUT...J=1 TO STORE ROOT')
4     CONTINUE
14    WRITE(IN8,21)
21    FORMAT(1X,'I2...INPUT J=0 TO FIND REAL   EIGEN MODES')
      READ(IN7,17) J
      IF(J.NE.0) GOTO 86
      III=0
      DO 82 I=1,NRT,2
      III=III+1
      XI=-CRT(NG2+I,1)
      CALL FDET(NSYM,COEF, XI,FI,1,VC32,1,NK10,NCN1,
     *N,NRT,NCT,CRT,NRGN,NT,NG2,SK0,SK1,SK2,SK3,SK4,A,N25L)
      XI=CRT(NG2+I,1)
      CALL FDET(NSYM,COEF,XI,FI,1,VC31,1,NK10,NCN1,
     *N,NRT,NCT,CRT,NRGN,NT,NG2,SK0,SK1,SK2,SK3,SK4,A,N25L)
      DO 83 I1=1,N
      IF(NRGN.EQ.1) GOTO 24850
      CMODE(NG2+I  ,I1,1)=(VC31(I1)+VC32(I1))/2
      CMODE(NG2+I  ,I1,2)=(VC31(I1)-VC32(I1))/2
      CMODE(NG2+I+1,I1,1)=(VC31(I1)-VC32(I1))/2
      CMODE(NG2+I+1,I1,2)=(VC31(I1)+VC32(I1))/2
      GOTO 83
24850 CMODE(NG2+III,I1,1)=VC31(I1)
      CMODE(NG2+III,I1,2)=0.D 00
83    CONTINUE
      IF(NRGN.NE.1) GOTO 8612
      CRT(NG2+III,1)=XI
      NRT12=NRT/2
```

```
      WRITE(IN8,861) NG2,NRT12,III,CRT(NG2+III,1)
      WRITE(IN8,87) (CMODE(NG2+III,J,1),J=1,N)
      WRITE(IN8,87) (CMODE(NG2+III,J,2),J=1,N)
8612  IF(NRGN.NE.0) GOTO 82
861   FORMAT(/3(1X,I4),1X,D15.8,1X,'REAL   EIGEN VALUE AND MODE')
      WRITE(IN8,861) NG2,NRT, I,CRT(NG2+I,1)
      WRITE(IN8,87) (CMODE(NG2+I  ,J,1),J=1,N)
      WRITE(IN8,87) (CMODE(NG2+I  ,J,2),J=1,N)
      WRITE(IN8,87) (CMODE(NG2+I+1,J,1),J=1,N)
      WRITE(IN8,87) (CMODE(NG2+I+1,J,2),J=1,N)
87    FORMAT(10(1X,D10.4))
82    CONTINUE
      IF(NRGN.EQ.1) NRT=NRT/2
86    RETURN
      END
      FUNCTION HPX(N,I,P,A,B,YN,Y,MT,NRGN)
      IMPLICIT REAL     (A-H,O-Z)
      DIMENSION P(6,4)
      CH0(X)=DCOSH(X)*(1-NRGN)+DEXP(A*X)*NRGN
      SH0(X)=DSINH(X)*(1-NRGN)+DEXP(A*X)*NRGN
      CH1(X)=SH0(X)*A
      SH1(X)=CH0(X)*A
      CH2(X)=CH0(X)*A*A
      SH2(X)=SH0(X)*A*A
      CH3(X)=SH0(X)*A*A*A
      SH3(X)=CH0(X)*A*A*A
      CH4(X)=CH0(X)*A*A*A*A
      SH4(X)=SH0(X)*A*A*A*A
      CH5(X)=SH0(X)*A*A*A*A*A
      SH5(X)=CH0(X)*A*A*A*A*A
      CH6(X)=CH0(X)*A*A*A*A*A*A
      SH6(X)=SH0(X)*A*A*A*A*A*A
      SHC0(X)=SH0(X)*DCOS(B*X)
      SHS0(X)=SH0(X)*DSIN(B*X)
      CHS0(X)=CH0(X)*DSIN(B*X)
      CHC0(X)=CH0(X)*DCOS(B*X)
      SHC1(X)=A*CHC0(X)-B*SHS0(X)
      SHS1(X)=A*CHS0(X)+B*SHC0(X)
      CHS1(X)=A*SHS0(X)+B*CHC0(X)
      CHC1(X)=A*SHC0(X)-B*CHS0(X)
      SHC2(X)=(A*A-B*B)*SHC0(X)-2*A*B*CHS0(X)
      SHS2(X)=(A*A-B*B)*SHS0(X)+2*A*B*CHC0(X)
      CHS2(X)=(A*A-B*B)*CHS0(X)+2*A*B*SHC0(X)
      CHC2(X)=(A*A-B*B)*CHC0(X)-2*A*B*SHS0(X)
      SHC3(X)=(A**3-3*A*B*B)*CHC0(X)-(3*A*A*B-B**3)*SHS0(X)
      SHS3(X)=(A**3-3*A*B*B)*CHS0(X)+(3*A*A*B-B**3)*SHC0(X)
      CHS3(X)=(A**3-3*A*B*B)*SHS0(X)+(3*A*A*B-B**3)*CHC0(X)
      CHC3(X)=(A**3-3*A*B*B)*SHC0(X)-(3*A*A*B-B**3)*CHS0(X)
      SHC4(X)=(A**4-6*A*A*B*B+B**4)*SHC0(X)-(4*A**3*B-4*A*B**3)*CHS0(X)
      SHS4(X)=(A**4-6*A*A*B*B+B**4)*SHS0(X)+(4*A**3*B-4*A*B**3)*CHC0(X)
      CHS4(X)=(A**4-6*A*A*B*B+B**4)*CHS0(X)+(4*A**3*B-4*A*B**3)*SHC0(X)
      CHC4(X)=(A**4-6*A*A*B*B+B**4)*CHC0(X)-(4*A**3*B-4*A*B**3)*SHS0(X)
      SHC5(X)=(A**4-6*A*A*B*B+B**4)*SHC1(X)-(4*A**3*B-4*A*B**3)*CHS1(X)
      SHS5(X)=(A**4-6*A*A*B*B+B**4)*SHS1(X)+(4*A**3*B-4*A*B**3)*CHC1(X)
```

```
      CHS5(X)=(A**4-6*A*A*B*B+B**4)*CHS1(X)+(4*A**3*B-4*A*B**3)*SHC1(X)
      CHC5(X)=(A**4-6*A*A*B*B+B**4)*CHC1(X)-(4*A**3*B-4*A*B**3)*SHS1(X)
      SHC6(X)=(A**4-6*A*A*B*B+B**4)*SHC2(X)-(4*A**3*B-4*A*B**3)*CHS2(X)
      SHS6(X)=(A**4-6*A*A*B*B+B**4)*SHS2(X)+(4*A**3*B-4*A*B**3)*CHC2(X)
      CHS6(X)=(A**4-6*A*A*B*B+B**4)*CHS2(X)+(4*A**3*B-4*A*B**3)*SHC2(X)
      CHC6(X)=(A**4-6*A*A*B*B+B**4)*CHC2(X)-(4*A**3*B-4*A*B**3)*SHS2(X)
      N1=N+1
      GOTO(1,2,3,4),N1
1     IF(MT.EQ.1)HPX=(P(I,1)+P(I,2)*Y+P(I,3)*Y*Y+P(I,4)*Y*Y*Y)
      IF(MT.EQ.2)HPX=(P(I,1)*CH0(Y)+P(I,2)*SH0(Y))
      IF(MT.EQ.3)HPX=(P(I,1)*CHC0(Y)+P(I,2)*CHS0(Y)+P(I,3)*SHC0(Y)+P(I,4-
     *)*SHS0(Y))
      RETURN
2     IF(MT.EQ.1)HPX=(P(I,2)*1   +P(I,3)*2*Y+P(I,4)*3*Y*Y)*YN
      IF(MT.EQ.2)HPX=(P(I,1)*CH1(Y)+P(I,2)*SH1(Y))*YN
      IF(MT.EQ.3)HPX=(P(I,1)*CHC1(Y)+P(I,2)*CHS1(Y)+P(I,3)*SHC1(Y)+P(I,4-
     *)*SHS1(Y))*YN
      RETURN
3     IF(MT.EQ.1)HPX=(      P(I,3)*2  +P(I,4)*3*2*Y)*YN**2
      IF(MT.EQ.2)HPX=(P(I,1)*CH2(Y)+P(I,2)*SH2(Y))*YN**2
      IF(MT.EQ.3)HPX=(P(I,1)*CHC2(Y)+P(I,2)*CHS2(Y)+P(I,3)*SHC2(Y)+P(I,4-
     *)*SHS2(Y))*YN**2
      RETURN
4     IF(MT.EQ.1)HPX=(           +P(I,4)*3*2*1)*YN**3
      IF(MT.EQ.2)HPX=(P(I,1)*CH3(Y)+P(I,2)*SH3(Y))*YN**3
      IF(MT.EQ.3)HPX=(P(I,1)*CHC3(Y)+P(I,2)*CHS3(Y)+P(I,3)*SHC3(Y)+P(I,4-
     *)*SHS3(Y))*YN**3
      RETURN
      END
      SUBROUTINE FUN(NL,N,I1,N2,N4,NRGN,NV,MT,NMT,H,Y1,Z1,F,FPS,        -
     *CB11,CB12,CB22,SNC2,SHM2,SHQ2,KQ2,KM2,KP2,KN2,EPSLON,CURVTR,THICK,-
     *NMTYPE,SLN,CRT,CMODE,RCM,NCN1,NCN2,IPR)
      IMPLICIT REAL   (A-H,O-Z)
      REAL   RCM
      REAL   KM1,KM2,KN1,KN2,KQ1,KQ2,KP1,KP2,NU12,NU13,NU23
      DIMENSION CB11(1),CB12(1),CB22(1),SNC2(1),SHM2(1),SHQ2(1),KQ2(1), -
     *KM2(1),KP2(1),KN2(1),THICK(1),NMTYPE(1),                          -
     *SLN(1),CRT(NCN1,1),CMODE(NCN1,NCN2,1),RCM(1),P(6,4)
      DATA Z01,Z02,Z03,Z04,Z05,Z06/0.,0.,0.,0.,0.,0./
C.....NV=1   N2 (X1,X2)     NV=2    U2(X1,X2,Z)
C.....NV=3   M2 (X1,X2)     NV=4   PHI(X1,X2,0)
C.....NV=5   Q2 (X1,X2)     NV=6    W (X1,X2,Z)
C.....NV=7   SYY(X1,X2,Z)   NV=8   SYZ(X1,X2,Z)      NV=9   SZZ(X1,X2,Z)
C.....NV=10  WB(X1,X2)  THICKNESS STRETCH PARAMETER
C.....NV=11  SYZ(X1,X2,-1),X2   SHEAR STRESS DERIVATIVE AT PLY BOTTOM
      N1=NMTYPE(I1)
      CALL PZERO(P,24)
      CN=THICK(I1)/2.
      YN=CN/(H/2.)
      Y=Y1/(H/2.)
      Z=Z1/CN
      A=CRT(N2,1)
      B=CRT(N2,2)
      IF(4*N-5.LT.3.OR.4*N-5.GT.4*NL-2) GOTO 8101
```

```
      Z05=SLN(N4+4*N-5)
      Z06=SLN(N4+4*N-4)
8101  Z01=SLN(N4+4*N-3)
      Z02=SLN(N4+4*N-2)
      IF(4*N-1.LT.3.OR.4*N-1.GT.4*NL-2) GOTO 8201
      Z03=SLN(N4+4*N-1)
      Z04=SLN(N4+4*N-0)
8201  IF(4*N-5.LT.3.OR.4*N-5.GT.4*NL-2) GOTO 9905
      DO 8945 I=1,4
      P(5,I)=CMODE(N2,4*N-5,I)
8945  P(6,I)=CMODE(N2,4*N-4,I)
9905  DO 8946 I=1,4
      P(1,I)=CMODE(N2,4*N-3,I)
8946  P(2,I)=CMODE(N2,4*N-2,I)
      IF(4*N-1.LT.3.OR.4*N-1.GT.4*NL-2) GOTO 9904
      DO 8947 I=1,4
      P(3,I)=CMODE(N2,4*N-1,I)
8947  P(4,I)=CMODE(N2,4*N-0,I)
9904  GOTO(1,2,3,5,5,6,7,5,9,6,11),NV
1     FPS=2*CN*(KP2(N1)*(Z06+Z04)+CB12(N1)*EPSLON)
      H11=HPX(1,1,P,A,B,YN,Y,MT,NRGN)
      H15=HPX(1,5,P,A,B,YN,Y,MT,NRGN)
      H13=HPX(1,3,P,A,B,YN,Y,MT,NRGN)
      H06=HPX(0,6,P,A,B,YN,Y,MT,NRGN)
      H04=HPX(0,4,P,A,B,YN,Y,MT,NRGN)
      F=2*CN*(H11*CB22(N1)+(H15-H13)*KN2(N1)+(H06+H04)*KP2(N1))
      RETURN
3     FPS=(KQ2(N1)/CN*(Z06-Z04)-CB12(N1)*CURVTR)*2./3*CN**3
      H22=HPX(2,2,P,A,B,YN,Y,MT,NRGN)
      H15=HPX(1,5,P,A,B,YN,Y,MT,NRGN)
      H13=HPX(1,3,P,A,B,YN,Y,MT,NRGN)
      H06=HPX(0,6,P,A,B,YN,Y,MT,NRGN)
      H04=HPX(0,4,P,A,B,YN,Y,MT,NRGN)
      F=2./3*CN*CN*(-H22*CB22(N1)+(H15+H13)*KM2(N1)+(H06-H04)*KQ2(N1))
      RETURN
5     FPS=CN*(Z05+Z03)
      H32=HPX(3,2,P,A,B,YN,Y,MT,NRGN)
      H25=HPX(2,5,P,A,B,YN,Y,MT,NRGN)
      H23=HPX(2,3,P,A,B,YN,Y,MT,NRGN)
      H16=HPX(1,6,P,A,B,YN,Y,MT,NRGN)
      H14=HPX(1,4,P,A,B,YN,Y,MT,NRGN)
      H05=HPX(0,5,P,A,B,YN,Y,MT,NRGN)
      H03=HPX(0,3,P,A,B,YN,Y,MT,NRGN)
      F=2./3*CN*(-H32*CB22(N1)+(H25+H23)*KM2(N1)+(H16-H14)*KQ2(N1))
     *+(H05+H03)*CN
      IF(NV.EQ.4) GOTO 4
      IF(NV.EQ.8) GOTO 8
      RETURN
4     FPS=(SHM2(N1)+SHQ2(N1))*(Z05+Z03)
      H12=HPX(1,2,P,A,B,YN,Y,MT,NRGN)
      F=SHQ2(N1)/CN*F-H12+SHM2(N1)*(H05+H03)
      RETURN
6     ZZZZ=.000000001D 00
      IF(NV.EQ.10) Z=ZZZZ
```

```
 61   FPS=WVD(1,N1,Z,NMT,RCM,IPR)*CN*Z02+WVD(3,N1,Z*CN,NMT,RCM,IPR)          -
     *+((WVD(7,N1,Z,NMT,RCM,IPR)+WVD(8,N1,Z,NMT,RCM,IPR))*Z06                 -
     * +(WVD(7,N1,Z,NMT,RCM,IPR)-WVD(8,N1,Z,NMT,RCM,IPR))*Z04)*CN
      H02=HPX(0,2,P,A,B,YN,Y,MT,NRGN)
      H22=HPX(2,2,P,A,B,YN,Y,MT,NRGN)
      H11=HPX(1,1,P,A,B,YN,Y,MT,NRGN)
      H15=HPX(1,5,P,A,B,YN,Y,MT,NRGN)
      H13=HPX(1,3,P,A,B,YN,Y,MT,NRGN)
      H06=HPX(0,6,P,A,B,YN,Y,MT,NRGN)
      H04=HPX(0,4,P,A,B,YN,Y,MT,NRGN)
      F=CN*(H02*WVD(1,N1,Z,NMT,RCM,IPR)+H22*WVD(2,N1,Z,NMT,RCM,IPR)+        -
     *        H11*WVD(4,N1,Z,NMT,RCM,IPR)+                                    -
     *       H15*(WVD(5,N1,Z,NMT,RCM,IPR)+WVD(6,N1,Z,NMT,RCM,IPR))+          -
     *      H13*(-WVD(5,N1,Z,NMT,RCM,IPR)+WVD(6,N1,Z,NMT,RCM,IPR))+          -
     *       H06*(WVD(7,N1,Z,NMT,RCM,IPR)+WVD(8,N1,Z,NMT,RCM,IPR))+          -
     *       H04*(WVD(7,N1,Z,NMT,RCM,IPR)-WVD(8,N1,Z,NMT,RCM,IPR)))
      IF(NV.EQ.6) RETURN
      IF(Z.NE.ZZZZ) GOTO 62
      FPS1=FPS
      F1=F
      Z=-ZZZZ
      GOTO 61
 62   FPS=(FPS1-FPS)/2/CN/ZZZZ
      F=(F1-F)/2/CN/ZZZZ
      RETURN
 1    FPS=0
      H13=HPX(1,3,P,A,B,YN,Y,MT,NRGN)
      F=H13/CN
      RETURN
 2    FPS=Z01*CN*WVD(11,N1,Z,NMT,RCM,IPR)+Z05*CN*(WVD(13,N1,Z,NMT,         -
     *RCM,IPR)+WVD(15,N1,Z,NMT,RCM,IPR))+                                     -
     *Z03*CN*(-WVD(13,N1,Z,NMT,RCM,IPR)+WVD(15,N1,ZNMT,RCM,IPR))
      H12=HPX(1,2,P,A,B,YN,Y,MT,NRGN)
      H32=HPX(3,2,P,A,B,YN,Y,MT,NRGN)
      H01=HPX(0,1,P,A,B,YN,Y,MT,NRGN)
      H21=HPX(2,1,P,A,B,YN,Y,MT,NRGN)
      H05=HPX(0,5,P,A,B,YN,Y,MT,NRGN)
      H25=HPX(2,5,P,A,B,YN,Y,MT,NRGN)
      H03=HPX(0,3,P,A,B,YN,Y,MT,NRGN)
      H23=HPX(2,3,P,A,B,YN,Y,MT,NRGN)
      H16=HPX(1,6,P,A,B,YN,Y,MT,NRGN)
      H14=HPX(1,4,P,A,B,YN,Y,MT,NRGN)
      F=(H12*WVD(9,N1,Z,NMT,RCM,IPR)+H32*WVD(10,N1,Z,NMT,RCM,IPR)+          -
     *    H01*WVD(11,N1,Z,NMT,RCM,IPR)+H21*WVD(12,N1,Z,NMT,RCM,IPR)+         -
     *  H05*( WVD(13,N1,Z,NMT,RCM,IPR)+WVD(15,N1,Z,NMT,RCM,IPR))+            -
     *  H03*(-WVD(13,N1,Z,NMT,RCM,IPR)+WVD(15,N1,Z,NMT,RCM,IPR))+            -
     *  H25*( WVD(14,N1,Z,NMT,RCM,IPR)+WVD(16,N1,Z,NMT,RCM,IPR))+            -
     *  H23*(-WVD(14,N1,Z,NMT,RCM,IPR)+WVD(16,N1,Z,NMT,RCM,IPR))+            -
     *  H16*( WVD(17,N1,Z,NMT,RCM,IPR)+WVD(18,N1,Z,NMT,RCM,IPR))+            -
     *  H14*( WVD(17,N1,Z,NMT,RCM,IPR)-WVD(18,N1,Z,NMT,RCM,IPR)))*CN
      RETURN
 8    FPS=Z05*(+Z/2+.5)+Z03*(-Z/2+.5)
      H05=HPX(0,5,P,A,B,YN,Y,MT,NRGN)
      H03=HPX(0,3,P,A,B,YN,Y,MT,NRGN)
```

```
      F=H05*(Z/2+.5)+H03*(-Z/2+.5)+(1-Z**2)/2*F
      RETURN
9     FPS=( .5+Z/2-(Z**3-Z)/4)*Z06+( .5-Z/2+(Z**3-Z)/4)*Z04
      H15=HPX(1,5,P,A,B,YN,Y,MT,NRGN)
      H13=HPX(1,3,P,A,B,YN,Y,MT,NRGN)
      H06=HPX(0,6,P,A,B,YN,Y,MT,NRGN)
      H04=HPX(0,4,P,A,B,YN,Y,MT,NRGN)
      F=H06*(.5+Z/2-(Z**3-Z)/4)+H04*(.5-Z/2+(Z**3-Z)/4)+                -
     *H15*(-(Z**2-1)/4-(Z**3-Z)/4)+H13*((Z**2-1)/4-(Z**3-Z)/4)
      RETURN
7     FPS=Z06*(KP2(N1)+Z*KQ2(N1)+(Z**3-.6*Z)/4*SNC2(N1))+Z04*(KP2(N1)-Z*-
     *KQ2(N1)-(Z**3-.6*Z)/4*SNC2(N1))+CB12(N1)*(EPSLON-CURVTR*CN*Z)
      H15=HPX(1,5,P,A,B,YN,Y,MT,NRGN)
      H13=HPX(1,3,P,A,B,YN,Y,MT,NRGN)
      H06=HPX(0,6,P,A,B,YN,Y,MT,NRGN)
      H04=HPX(0,4,P,A,B,YN,Y,MT,NRGN)
      H11=HPX(1,1,P,A,B,YN,Y,MT,NRGN)
      H22=HPX(2,2,P,A,B,YN,Y,MT,NRGN)
      F= H11      *CB22(N1)-H22      *CB22(N1)*Z                        -
     *+H15      *( KN2(N1)+Z*KM2(N1)+SNC2(N1)*( Z**2-1./3+Z**3-.6*Z)/4) -
     *+H13      *(-KN2(N1)+Z*KM2(N1)+SNC2(N1)*(-Z**2+1./3+Z**3-.6*Z)/4) -
     *+H06      *( KP2(N1)+Z*KQ2(N1)+.25*SNC2(N1)*(+(Z**3-.6*Z)))        -
     *+H04      *( KP2(N1)-Z*KQ2(N1)+.25*SNC2(N1)*(-(Z**3-.6*Z)))
      RETURN
      END
      SUBROUTINE DEQMAT(NSYM2,NPLY,NEQ,NBC,NG1,NG2,NR,NMT, NSTREI,H,    -
     *CB22,KQ2,KM2,KP2,KN2,THICK,      SLN,CRT,CMODE,                   -
     *SK0,SK1,SK2,SK3,SK4,SX,VC45,FSLN,ISGN,RCM,                        -
     *NCN1,NCN2,NSREG,NMTYPE,NG4,N25L,IPR)
      IMPLICIT REAL   (A-H,O-Z)
      REAL   RCM
      REAL   KM1,KM2,KN1,KN2,KQ1,KQ2,KP1,KP2,NU12,NU13,NU23
      DIMENSION CB22(1),KQ2(1),KM2(1),KP2(1),KN2(1),THICK(1),SLN(1),    -
     *CRT(NCN1,1),CMODE(NCN1,NCN2,1),SX(NSTREI,1),                      -
     *VC45(1),FSLN(1),ISGN(1),NMTYPE(1),SK0(N25L ,1),                   -
     *SK1(N25L ,1),SK2(N25L ,1),SK3(N25L ,1),SK4(N25L ,1),RCM(1)
C.....U1,U2,W AND Z, ARE NONDIMENSIONALIZED W.R.T CN
C.....X1,X2 ARE NONDIMENSIONALIZED W.R.T.  LAMINATE SEMITHICKNESS, H/2
C.....EPSLN=UNIFORM STRAIN IN X1-DIRECTION, U1,1
C.....CURVTR=NONDIMENSIONALIZED CURVATURE IN X1-DIRECTION, W,XX
      CALL PZERO(SK0,N25L*N25L)
      CALL PZERO(SK1,N25L*N25L)
      CALL PZERO(SK2,N25L*N25L)
      CALL PZERO(SK3,N25L*N25L)
      CALL PZERO(SK4,N25L*N25L)
      CALL PZERO(FSLN,N25L)
C.....CREATING OF THE DIFFERENTIAL EQUATION OPERATOR MATRIX
C.....  SK4 *LM**4+ SK3 *LM**3+ SK2 *LM**2+ SK1 *LM1+ SK0
      DO 101 N=1,NPLY
      L=NMTYPE(N+NG1)
      YN=THICK(N+NG1)/2/(H/2)
      CN=THICK(N+NG1)/2
      IF(N.LT.NPLY) L1=NMTYPE(N+NG1+1)
      IF(N.LT.NPLY) YN1=THICK(N+NG1+1)/2/(H/2)
```

```
      IF(N.LT.NPLY) CN1=THICK(N+NG1+1)/2
      SK2(4*N-3,4*N-3)=2.*YN*YN*CB22(L)
      IF(4*N-5.GT.2.AND.4*N-5.LT.4*NPLY-4) SK2(4*N-3,4*N-5)=2.*YN*YN*KN2-
     *(L)
      IF(4*N-1.GT.2.AND.4*N-1.LT.4*NPLY-4) SK2(4*N-3,4*N-1)=-2*YN*YN*KN2-
     *(L)
      IF(4*N-5.GT.2.AND.4*N-5.LT.4*NPLY-4) SK0(4*N-3,4*N-5)=1
      IF(4*N-1.GT.2.AND.4*N-1.LT.4*NPLY-4) SK0(4*N-3,4*N-1)=-1.
      IF(4*N-4.GT.3.AND.4*N-4.LT.4*NPLY-3) SK1(4*N-3,4*N-4)=2.*YN*KP2(L)
      IF(4*N  .GT.3.AND.4*N  .LT.4*NPLY-3) SK1(4*N-3,4*N)=2.*YN*KP2(L)
      SK4(4*N-2,4*N-2)=-2./3*CB22(L)*YN*YN*YN*YN
      IF(4*N-5.GT.2.AND.4*N-5.LT.4*NPLY-4) SK3(4*N-2,4*N-5)=2./3*KM2(L)*-
     *YN*YN*YN
      IF(4*N-1.GT.2.AND.4*N-1.LT.4*NPLY-4) SK3(4*N-2,4*N-1)=2./3*KM2(L)*-
     *YN*YN*YN
      IF(4*N-4.GT.3.AND.4*N-4.LT.4*NPLY-3) SK2(4*N-2,4*N-4)=2./3*KQ2(L)*-
     *YN*YN
      IF(4*N  .GT.3.AND.4*N  .LT.4*NPLY-3) SK2(4*N-2,4*N  )=-2./3*KQ2(L)-
     **YN*YN
      IF(4*N-5.GT.2.AND.4*N-5.LT.4*NPLY-4) SK1(4*N-2,4*N-5)=YN
      IF(4*N-1.GT.2.AND.4*N-1.LT.4*NPLY-4) SK1(4*N-2,4*N-1)=YN
      IF(4*N-4.GT.3.AND.4*N-4.LT.4*NPLY-3) SK0(4*N-2,4*N-4)=1
      IF(4*N  .GT.3.AND.4*N  .LT.4*NPLY-3) SK0(4*N-2,4*N  )=-1
      IF(N.EQ.NPLY) GOTO 101
      SK0(4*N-1,4*N-2)=WVD(1,L,-1.,NMT,RCM,IPR)*CN
      SK2(4*N-1,4*N-2)=WVD(2,L,-1.,NMT,RCM,IPR)*YN*YN*CN
      SK0(4*N-1,4*N+2)=-WVD(1,L1,1.0,NMT,RCM,IPR)*CN1
      SK2(4*N-1,4*N+2)=-WVD(2,L1,1.0,NMT,RCM,IPR)*YN1*YN1*CN1
      SK1(4*N-1,4*N-3)=WVD(4,L,-1.,NMT,RCM,IPR)*YN*CN
      SK1(4*N-1,4*N+1)=-WVD(4,L1,1.0,NMT,RCM,IPR)*YN1*CN1
      IF(4*N-5.GT.2.AND.4*N-5.LT.4*NPLY-4) SK1(4*N-1,4*N-5)=WVD(5,L,-1.,-
     *NMT,RCM,IPR)*YN*CN+ WVD(6,L,-1.,NMT,RCM,IPR)*CN*YN
      IF(4*N-1.GT.2.AND.4*N-1.LT.4*NPLY-4) SK1(4*N-1,4*N-1)=            -
     *- WVD(5,L,-1.,NMT,RCM,IPR)*CN*YN- WVD(5,L1,1.,NMT,RCM,IPR)*CN1*YN1-
     *+ WVD(6,L,-1.,NMT,RCM,IPR)*CN*YN- WVD(6,L1,1.,NMT,RCM,IPR)*CN1*YN1
      IF(4*N+3.GT.2.AND.4*N+3.LT.4*NPLY-4) SK1(4*N-1,4*N+3)= WVD(5,L1,1.-
     *0,NMT,RCM,IPR)*CN1*YN1- WVD(6,L1,1.0,NMT,RCM,IPR)*CN1*YN1
      IF(4*N-4.GT.3.AND.4*N-4.LT.4*NPLY-3) SK0(4*N-1,4*N-4)= WVD(7,L,-1.-
     *0,NMT,RCM,IPR)*CN+ WVD(8,L,-1.,NMT,RCM,IPR)*CN
      IF(4*N  .GT.3.AND.4*N  .LT.4*NPLY-3) SK0(4*N-1,4*N  )= WVD(7,L,-1.-
     *0,NMT,RCM,IPR)*CN- WVD(8,L,-1.,NMT,RCM,IPR)*CN- WVD(7,L1,1.0,NMT, -
     *RCM,IPR)*CN1- WVD(8,L1,1.0,NMT,RCM,IPR)*CN1
      IF(4*N+4.GT.3.AND.4*N+4.LT.4*NPLY-3)SK0(4*N-1,4*N+4)=- WVD(7,L1,1.-
     *0,NMT,RCM,IPR)*CN1+ WVD(8,L1,1.0,NMT,RCM,IPR)*CN1
      FSLN(4*N-1)=- WVD(3,L,-CN,NMT,RCM,IPR)+WVD(3,L1,CN1,NMT,RCM,IPR)
      SK1(4*N,4*N-2)= WVD(9,L,-1.,NMT,RCM,IPR)*YN*CN
      SK3(4*N,4*N-2)= WVD(10,L,-1.,NMT,RCM,IPR)*YN*YN*YN*CN
      SK1(4*N,4*N+2)=- WVD(9,L1,1.0,NMT,RCM,IPR)*YN1*CN1
      SK3(4*N,4*N+2)=- WVD(10,L1,1.0,NMT,RCM,IPR)*YN1*YN1*YN1*CN1
      SK0(4*N,4*N-3)= WVD(11,L,-1.,NMT,RCM,IPR)*CN
      SK2(4*N,4*N-3)= WVD(12,L,-1.,NMT,RCM,IPR)*YN*YN*CN
      SK0(4*N,4*N+1)=- WVD(11,L1,1.0,NMT,RCM,IPR)*CN1
      SK2(4*N,4*N+1)=- WVD(12,L1,1.0,NMT,RCM,IPR)*YN1*YN1*CN1
      IF(4*N-5.GT.2.AND.4*N-5.LT.4*NPLY-4) SK0(4*N,4*N-5)= WVD(13,L,-1.,-
```

```
     *NMT,RCM,IPR)*CN+ WVD(15,L,-1.,NMT,RCM,IPR)*CN
      IF(4*N-1.GT.2.AND.4*N-1.LT.4*NPLY-4) SK0(4*N,4*N-1)=- WVD(13,L,-1.-
     *,NMT,RCM,IPR)*CN+ WVD(15,L,-1.,NMT,RCM,IPR)*CN- WVD(13,L1,1.0,NMT,-
     *RCM,IPR)*CN1-WVD(15,L1,1.0,NMT,RCM,IPR)*CN1
      IF(4*N+3.GT.2.AND.4*N+3.LT.4*NPLY-4) SK0(4*N,4*N+3)= WVD(13,L1,1.0-
     *,NMT,RCM,IPR)*CN1- WVD(15,L1,1.0,NMT,RCM,IPR)*CN1
      IF(4*N-5.GT.2.AND.4*N-5.LT.4*NPLY-4) SK2(4*N,4*N-5)=( WVD(14,L,-1.-
     *,NMT,RCM,IPR)+WVD(16,L,-1.,NMT,RCM,IPR))*YN*YN*CN
      IF(4*N-1.GT.2.AND.4*N-1.LT.4*NPLY-4) SK2(4*N,4*N-1)=               -
     *(- WVD(14,L,-1.,NMT,RCM,IPR)+ WVD(16,L,-1.,NMT,RCM,IPR))*YN*YN*CN--
     *(WVD(14,L1,1.0,NMT,RCM,IPR)+WVD(16,L1,1.,NMT,RCM,IPR))*YN1*YN1*CN1
      IF(4*N+3.GT.2.AND.4*N+3.LT.4*NPLY-4) SK2(4*N,4*N+3)=( WVD(14,L1,1.-
     *0,NMT,RCM,IPR)-WVD(16,L1,1.0,NMT,RCM,IPR))*CN1*YN1*YN1
      IF(4*N-4.GT.3.AND.4*N-4.LT.4*NPLY-3) SK1(4*N,4*N-4)=( WVD(17,L,-1.-
     *,NMT,RCM,IPR)+WVD(18,L,-1.,NMT,RCM,IPR))*YN*CN
      IF(4*N  .GT.3.AND.4*N  .LT.4*NPLY-3) SK1(4*N,4*N  )=               -
     *( WVD(17,L,-1.,NMT,RCM,IPR)- WVD(18,L,-1.,NMT,RCM,IPR))*YN*CN-(   -
     *WVD(17,L1,1.0,NMT,RCM,IPR)+WVD(18,L1,1.0,NMT,RCM,IPR))*YN1*CN1
      IF(4*N+4.GT.3.AND.4*N+4.LT.4*NPLY-3) SK1(4*N,4*N+4)=-(WVD(17,L1,1.-
     *0,NMT,RCM,IPR)-WVD(18,L1,1.0,NMT,RCM,IPR))*YN1*CN1
  101 CONTINUE
C       NQQ=8
C       DO 6491 I=1,NQQ
C6491   WRITE(6,6390) I,(SK0(I,J),J=1,NQQ)
C       DO 6492 I=1,NQQ
C6492   WRITE(6,6390) I,(SK1(I,J),J=1,NQQ)
C       DO 6493 I=1,NQQ
C6493   WRITE(6,6390) I,(SK2(I,J),J=1,NQQ)
C       DO 6494 I=1,NQQ
C6494   WRITE(6,6390) I,(SK3(I,J),J=1,NQQ)
C       DO 6495 I=1,NQQ
C6495   WRITE(6,6390) I,(SK4(I,J),J=1,NQQ)
      IF(NSYM2.GT.1.AND.NPLY.EQ.1) GOTO 142
      IF(NSYM2.GT.1) GOTO 7104
      SLN(NG4+2)=0.D 00
      DO 5534 I=2,NPLY
      IF(NPLY.LT.2) GOTO 5534
      SLN(NG4+4*I-2)=(FSLN(4*I-5)-SK0(4*I-5,4*I-6)*SLN(NG4+    4*I-6))/ -
     *SK0(4*I-5,4*I-2)
 5534 CONTINUE
      DO 5536 I=1,NPLY
 5536 SLN(NG4+   4*I-2)=SLN(NG4+   4*I-2)/(THICK(NG1+I)/2.)
      IF(NPLY.GT.1) GOTO 7113
      CMODE(NG2+5,1,1)=1
      CMODE(NG2+2,2,1)=1
      CMODE(NG2+1,2,2)=1
      CMODE(NG2+6,1,2)=1
      CMODE(NG2+4,2,3)=1
      CMODE(NG2+3,2,4)=1
 7113 IF(NPLY.LE.1) GOTO 142
      CMODE(NG2+5,1,1)=1
      CMODE(NG2+2,2,1)=1
      CMODE(NG2+1,2,2)=1
      CMODE(NG2+6,1,2)=1
```

```
      CMODE(NG2+4,2,3)=1
      DO 7114 I=2,NPLY
      CMODE(NG2+5,4*I-3,1)=1
      CMODE(NG2+2,4*I-2,1)=1
      CMODE(NG2+1,4*I-2,2)=-1./SK0(4*I-5,4*I-2)*(
     *                         SK0(4*I-5,4*I-6)*CMODE(NG2+1,4*I-6,2))
      CMODE(NG2+1,4*I-3,1)=-1./SK0(4*I-4,4*I-3)*(
     *                         SK0(4*I-4,4*I-7)*CMODE(NG2+1,4*I-7,1)+
     *                         SK1(4*I-4,4*I-6)*CMODE(NG2+1,4*I-6,2)+
     *                         SK1(4*I-4,4*I-2)*CMODE(NG2+1,4*I-2,2))
      CMODE(NG2+6,4*I-3,2)=-1./SK0(4*I-4,4*I-3)*(
     *                         SK0(4*I-4,4*I-7)*CMODE(NG2+6,4*I-7,2))
      CMODE(NG2+6,4*I-2,1)=-1./SK0(4*I-5,4*I-2)*(
     *                         SK1(4*I-5,4*I-7)*CMODE(NG2+6,4*I-7,2)+
     *                         SK0(4*I-5,4*I-6)*CMODE(NG2+6,4*I-6,1)+
     *                         SK1(4*I-5,4*I-3)*CMODE(NG2+6,4*I-3,2))
      CMODE(NG2+4,4*I-2,3)=-1./SK0(4*I-5,4*I-2)*(
     *                         SK0(4*I-5,4*I-6)*CMODE(NG2+4,4*I-6,3))
      CMODE(NG2+4,4*I-3,2)=-1./SK0(4*I-4,4*I-3)*(
     *                       2.*SK1(4*I-4,4*I-6)*CMODE(NG2+4,4*I-6,3)+
     *                       2.*SK1(4*I-4,4*I-2)*CMODE(NG2+4,4*I-2,3)+
     *                         SK0(4*I-4,4*I-7)*CMODE(NG2+4,4*I-7,2))
 7114 CMODE(NG2+4,4*I-2,1)=-1./SK0(4*I-5,4*I-2)*(
     *                         SK0(4*I-5,4*I-6)*CMODE(NG2+4,4*I-6,1)+
     *                       2.*SK2(4*I-5,4*I-6)*CMODE(NG2+4,4*I-6,3)+
     *                       2.*SK2(4*I-5,4*I-2)*CMODE(NG2+4,4*I-2,3)+
     *                         SK1(4*I-5,4*I-7)*CMODE(NG2+4,4*I-7,2)+
     *                         SK1(4*I-5,4*I-3)*CMODE(NG2+4,4*I-3,2))
      CALL PZERO(SX,NSTREI)
      CALL PZERO(VC45,NSTREI)
      SX(1,1)=1
      SX(2,3)=1
      SX(3,4)=1
      VC45(3)=1
      DO 7116 I=1,NPLY
      SX(3+5*I-4,5*I-3)=2.*SK2(4*I-3,4*I-3)
      IF(5*I-5.GT.4.AND.5*I-5.LT.NSTREI+1)
     *SX(3+5*I-4,5*I-5)=SK0(4*I-3,4*I-5)
      IF(5*I  .GT.4.AND.5*I  .LT.NSTREI+1)
     *SX(3+5*I-4,5*I  )=SK0(4*I-3,4*I-1)
      IF(I.EQ.NPLY) GOTO 7116
      SX(3+5*I-3,5*I-1)=SK0(4*I-1,4*I-2)
      SX(3+5*I-3,5*I+4)=SK0(4*I-1,4*I+2)
      SX(3+5*I-2,5*I-1)=6.*SK2(4*I-1,4*I-2)
      SX(3+5*I-2,5*I-2)=   SK0(4*I-1,4*I-2)
      SX(3+5*I-2,5*I-3)=2.*SK1(4*I-1,4*I-3)
      SX(3+5*I-2,5*I+4)=6.*SK2(4*I-1,4*I+2)
      SX(3+5*I-2,5*I+3)=   SK0(4*I-1,4*I+2)
      SX(3+5*I-2,5*I+2)=2.*SK1(4*I-1,4*I+1)
      SX(3+5*I-1,5*I-1)=3.*SK1(4*I-0,4*I-2)
      SX(3+5*I-1,5*I-3)=   SK0(4*I-0,4*I-3)
      SX(3+5*I-1,5*I+4)=3.*SK1(4*I-0,4*I+2)
      SX(3+5*I-1,5*I+2)=   SK0(4*I-0,4*I+1)
      SX(3+5*I-0,5*I-1)=6.*SK3(4*I-0,4*I-2)
```

```
      SX(3+5*I-0,5*I-2)=   SK1(4*I-0,4*I-2)
      SX(3+5*I-0,5*I-3)=2.*SK2(4*I-0,4*I-3)
      SX(3+5*I-0,5*I-4)=   SK0(4*I-0,4*I-3)
      SX(3+5*I-0,5*I+4)=6.*SK3(4*I-0,4*I+2)
      SX(3+5*I-0,5*I+3)=   SK1(4*I-0,4*I+2)
      SX(3+5*I-0,5*I+2)=2.*SK2(4*I-0,4*I+1)
      SX(3+5*I-0,5*I+1)=   SK0(4*I-0,4*I+1)
      IF(5*I-5.GT.4.AND.5*I-5.LT.NSTREI+1)
     *SX(3+5*I-0,5*I-5)=   SK0(4*I-0,4*I-5)
      IF(5*I  .GT.4.AND.5*I  .LT.NSTREI+1)
     *SX(3+5*I-0,5*I  )=   SK0(4*I-0,4*I-1)
      IF(5*I+5.GT.4.AND.5*I+5.LT.NSTREI+1)
     *SX(3+5*I-0,5*I+5)=   SK0(4*I-0,4*I+3)
 7116 CONTINUE
      CALL SKINV(SX,NSTREI,SX,NSTREI,VC45,NSTREI,1,SIMUB,NSTREI)
      DO 7117 I=1,NPLY
      CMODE(NG2+3,4*I-3,1)=VC45(5*I-4)
      CMODE(NG2+3,4*I-3,3)=VC45(5*I-3)
      CMODE(NG2+3,4*I-2,2)=VC45(5*I-2)
      CMODE(NG2+3,4*I-2,4)=VC45(5*I-1)
      IF(I.EQ.NPLY) GOTO 7117
      CMODE(NG2+3,4*I-1,1)=VC45(5*I  )
 7117 CONTINUE
 7104 IF(NSYM2.EQ.3) GOTO 142
      NEVN=(NPLY+1)/2
      NEQ=4*NEVN-2
      IF(NPLY/2*2.EQ.NPLY) NEQ=4*NEVN
      N2EVN=(4*NPLY-2)/2+2
      N3EVN=(4*NPLY-2)/2+1
      N4EVN=(4*NPLY-2)
      IF(NPLY/2*2.NE.NPLY.AND.NSYM2.EQ.1) I1=N3EVN-1
      IF(NPLY/2*2.NE.NPLY.AND.NSYM2.EQ.1) I2=N3EVN-1
      IF(NPLY/2*2.NE.NPLY.AND.NSYM2.EQ.2) I1=N3EVN
      IF(NPLY/2*2.NE.NPLY.AND.NSYM2.EQ.2) I2=N3EVN
      IF(NPLY/2*2.EQ.NPLY.AND.NSYM2.EQ.1) I1=N3EVN
      IF(NPLY/2*2.EQ.NPLY.AND.NSYM2.EQ.1) I2=N3EVN-1
      IF(NPLY/2*2.EQ.NPLY.AND.NSYM2.EQ.2) I1=N3EVN-1
      IF(NPLY/2*2.EQ.NPLY.AND.NSYM2.EQ.2) I2=N3EVN
      IF(NPLY/2*2.NE.NPLY.AND.NSYM2.EQ.1) NBC = 8*(NEVN-1)+1+4
      IF(NPLY/2*2.EQ.NPLY.AND.NSYM2.EQ.1) NBC = 8*(NEVN-1)+1+4+2+2
      IF(NPLY/2*2.NE.NPLY.AND.NSYM2.EQ.2) NBC = 8*(NEVN-1)+1+2
      IF(NPLY/2*2.EQ.NPLY.AND.NSYM2.EQ.2) NBC = 8*(NEVN-1)+1+4+2
      DO 7119 I=1,NEVN
      J=4*(NPLY+1-I)
      ISGN(J-3)= (4*I-3)*(-1.)**(NSYM2)
      ISGN(J-2)= (4*I-2)*(-1.)**(NSYM2+1)
      ISGN(J-5)= (4*I-1)*(-1.)**(NSYM2+1)
 7119 ISGN(J-4)= (4*I-0)*(-1.)**(NSYM2)
      DO 7121 I=1,N3EVN
      IF(I.EQ.I2) GOTO 7121
      DO 71212 J=N2EVN,N4EVN
      K=ISGN(J)
      JSGN=K/IABS(K)
      K=IABS(K)
```

```
      SK0(I,K)=SK0(I,K)+SK0(I,J)*JSGN
      SK1(I,K)=SK1(I,K)+SK1(I,J)*JSGN
      SK2(I,K)=SK2(I,K)+SK2(I,J)*JSGN
      SK3(I,K)=SK3(I,K)+SK3(I,J)*JSGN
71212 SK4(I,K)=SK4(I,K)+SK4(I,J)*JSGN
7121  CONTINUE
      DO 7129 J=1,N3EVN
      SK0(I2,J)=0.D 00
      SK1(I2,J)=0.D 00
      SK2(I2,J)=0.D 00
      SK3(I2,J)=0.D 00
7129  SK4(I2,J)=0.D 00
      CALL PZERO(SK0(1,I1),N3EVN)
      CALL PZERO(SK1(1,I1),N3EVN)
      CALL PZERO(SK2(1,I1),N3EVN)
      CALL PZERO(SK3(1,I1),N3EVN)
      CALL PZERO(SK4(1,I1),N3EVN)
      SK1(I2,I1)=1.
6390  FORMAT(1X,I2,14F9.4)
C      NQQ=8
C      DO 6391 I=1,NQQ
C6391  WRITE(6,6390) I,(SK0(I,J),J=1,NQQ)
C      DO 6392 I=1,NQQ
C6392  WRITE(6,6390) I,(SK1(I,J),J=1,NQQ)
C      DO 6393 I=1,NQQ
C6393  WRITE(6,6390) I,(SK2(I,J),J=1,NQQ)
C      DO 6394 I=1,NQQ
C6394  WRITE(6,6390) I,(SK3(I,J),J=1,NQQ)
C      DO 6395 I=1,NQQ
C6395  WRITE(6,6390) I,(SK4(I,J),J=1,NQQ)
      RETURN
142   NEQ=4*NPLY-2
      NBC=8*NPLY-2
      RETURN
      END
      SUBROUTINE PZERO(A,N)
      IMPLICIT REAL   (A-H,O-Z)
      DIMENSION A(1)
      DO 1 I=1,N
1     A(I)=0.D 00
      RETURN
      END
      SUBROUTINE BIDOT(A,I1,B,I2,C,I3)
      IMPLICIT REAL   (A-H,O-Z)
      DIMENSION A(I1,1),B(I2,1),C(I1,1)
      DO 1 I=1,I1
      DO 1 K=1,I3
      C(I,K)=0.D 00
      DO 1 J=1,I2
1     C(I,K)=C(I,K)+A(I,J)*B(J,K)
      RETURN
      END
      SUBROUTINE SKINV(SK,NSK,SKI,NSKI,X,NX,INDIC,SIMUL,N)
      IMPLICIT REAL   (A-H,O-Z)
```

```
C.....INDIC  =0 GIVES DETE. (SIMUL), INVERSE (SKI) AND SOLUTION(X)
C.....INDIC -VE GIVES DETERMINANT AND INVERSE
C.....INDIC +VE GIVES DETERMINANT AND SOLUTION
      DIMENSION IROW(251),JCOL(251),JORD(251),Y(251),A(251,251),SK(NSK,1-
     *),SKI(NSKI,1),X(NX)
      EPS=1.D-70
      NRC=251
      MAX=N
      IF(INDIC.GE.0) MAX=N+1
      DO 101 I=1,N
      DO 101 J=1,N
101   A(I,J)=SK(I,J)
      IF(INDIC.LT.0) GO TO 103
      DO 102 I=1,N
102   A(I,MAX)=X(I)
103   CONTINUE
      IF(N.LT.NRC) GOTO 5
      WRITE(6,200)
200   FORMAT(/1X,'MATRIX IS TOO BIG IN "SKINV" '/)
      RETURN
5     CONTINUE
      DETER=1.
      DO 18 K=1,N
      KM1=K-1
      PIVOT=0.
      DO 11 I=1,N
      DO 11 J=1,N
      IF(K.EQ.1) GOTO 9
      DO 8 ISCAN=1,KM1
      DO 8 JSCAN=1,KM1
      IF(I.EQ.IROW(ISCAN)) GOTO 11
      IF(J.EQ.JCOL(JSCAN)) GOTO 11
8     CONTINUE
9     IF(DABS(A(I,J)).LE.DABS(PIVOT)) GOTO 11
      PIVOT=A(I,J)
      IROW(K)=I
      JCOL(K)=J
11    CONTINUE
      IF(DABS(PIVOT).GT.EPS) GOTO 13
      SIMUL=0.
      WRITE(6,1)
1     FORMAT(1X,'STOPPED IN SUBROUTINE SKINV..... DETERMINANT = 0')
      RETURN
13    IROWK=IROW(K)
      JCOLK=JCOL(K)
      DETER=DETER*PIVOT
      DO 14 J=1,MAX
14    A(IROWK,J)=A(IROWK,J)/PIVOT
      A(IROWK,JCOLK)=1./PIVOT
      DO 18 I=1,N
      AIJCK=A(I,JCOLK)
      IF(I.EQ.IROWK) GOTO 18
      A(I,JCOLK)=-AIJCK/PIVOT
      DO 17 J=1,MAX
```

```
17    IF(J.NE.JCOLK) A(I,J)=A(I,J)-AIJCK*A(IROWK,J)
18    CONTINUE
      DO 20 I=1,N
      IROWI=IROW(I)
      JCOLI=JCOL(I)
      JORD(IROWI)=JCOLI
20    IF(INDIC.GE.0) X(JCOLI)=A(IROWI,MAX)
      INTCH=0
      NM1=N-1
      DO 22 I=1,NM1
      IP1=I+1
      DO 22 J=IP1,N
      IF(JORD(J).GE.JORD(I)) GOTO 22
      JTEMP=JORD(J)
      JORD(J)=JORD(I)
      JORD(I)=JTEMP
      INTCH=INTCH+1
22    CONTINUE
      IF(INTCH/2*2.NE.INTCH) DETER=-DETER
      IF(INDIC.LE.0) GOTO 26
      SIMUL=DETER
      RETURN
26    DO 28 J=1,N
      DO 27 I=1,N
      IROWI=IROW(I)
      JCOLI=JCOL(I)
27    Y(JCOLI)=A(IROWI,J)
      DO 28 I=1,N
28    A(I,J)=Y(I)
      DO 30 I=1,N
      DO 29 J=1,N
      IROWJ=IROW(J)
      JCOLJ=JCOL(J)
29    Y(IROWJ)=A(I,JCOLJ)
      DO 30 J=1,N
30    A(I,J)=Y(J)
      SIMUL=DETER
      DO 209 I=1,N
      DO 209 J=1,N
209   SKI(I,J)=A(I,J)
      RETURN
      END
```

APPENDIX II: PLY EQUATIONS

The derivation of these equations is reported in reference 8. These equations encompass the following main categories: overall equilibrium, constitutive relations, stress distribution, and displacement distribution.

Overall Equilibrium:

$$N^k_{2,2} + n^k_2 = 0 \tag{I-1}$$

$$M^k_{2,2} + c_k\, m^k_2 - Q^k_2 = 0 \tag{I-2}$$

$$Q^k_{2,2} + q^k = 0 \tag{I-3}$$

where

$$n^k_2 = \sigma^k_{2z}(x_2,\ c_k) - \sigma^k_{2z}(x_2,\ -c_k) \tag{I-4}$$

$$m^k_2 = \sigma^k_{2z}(x_2,\ c_k) - \sigma^k_{2z}(x_2,\ -c_k) \tag{I-5}$$

$$q^k = \sigma^k_{zz}(x_2,\ c_k) - \sigma^k_{zz}(x_2,\ -c_k) \tag{I-6}$$

where N_2, M_2 and Q_2 are, respectively, the force, moment, and shear resultants associated with the x_2-coordinate direction, and n_2, m_2, and q are related to the interfacial stresses. The semithickness of the k-th ply is c_k. Superscript k which identifies the generic ply will be dropped in the subsequent equations for convenience.

Constitutive Relations:

$$N_\beta/h = \bar{C}_{\beta 1}\varepsilon + \bar{C}_{\beta 2}\, U_{2,2} + K_{n\beta}\, c\, n_{2,2} + K_{p\beta}\, p \ ; \ \beta = 1,2 \tag{1-7}$$

$$M_\beta/I = -\bar{C}_{\beta 1}W_{,11} - \bar{C}_{\beta 2}W_{,22} + K_{m\beta}\, m_{2,2} + K_{q\beta}q/c \tag{I-8}$$

$$\Phi_2 + W_{,2} = (Q_2 - c\, m_2/3)\, S_{44}c^2/(2I) \tag{I-9}$$

where

$$K_{m\beta} = (3\ \bar{C}_{\beta 2}\ S_{3\alpha}\ \bar{C}_{\alpha 2}/\bar{C}_{22} - 2\ \bar{C}_{\beta 2}\ S_{44} + 2\bar{C}_\beta)/20 \ ; \ \alpha,\beta = 1,2$$

$$K_{q\beta} = (3\ \bar{C}_{\beta 2}\ S_{3\alpha}\ \bar{C}_{\alpha 2}/\bar{C}_{22} - 12\ \bar{C}_{\beta 2}\ S_{44} + 12\bar{C}_\beta)/20$$

$$K_{n\beta} = (\bar{C}_{\beta 2}\ S_{3\alpha}\ \bar{C}_{\alpha 2}/\bar{C}_{22} + \bar{C}_{\beta 2}\ S_{44} + 2\bar{C}_\beta)/12$$

$$K_{p\beta}\ \bar{C}_\beta/2$$

$$p = \sigma_{zz}(x_2,\ c) + \sigma_{zz}(x_2,\ -c)\ ,\ h = 2c\ ,\ I = 2\ c^3/3 \tag{I-10}$$

where ε is the applied strain in the x_1-direction and $W_{,11}$ is the curvature about x_2-axis. [C] is the plane stress stiffness matrix and [S] is the general flexibility matrix. The constants $\bar{C}_1$, $\bar{C}_2$, and $\bar{C}_6$ introduce the influence of the transverse normal stress in the computation on the refined surface parallel stresses. For a given lamina with arbitrary orientation of the fibers σ_1 and σ_2 are surface parallel normal stresses, σ_3 is transverse normal stress and σ_6 the in-plane shear stress. The corresponding engineering strains are denoted by ε_1, ε_2, ε_3, and T_6. Using the generalized Hooke's law, the equations for ε_1, ε_2 and T_6 can be inverted to obtain

$$\sigma_i = \bar{C}_{ij}\varepsilon_j + \bar{C}_i\sigma_3 \quad ; \quad i,j = 1,\ 2,\ \text{and}\ 6 \tag{I-11}$$

where ε_6 is an alternative notation for T_6. The expressions for $\bar{C}_1$, $\bar{C}_2$, and $\bar{C}_6$ can be obtained in terms of flexibilities and later in terms of lamina elastic moduli with relative ease.

The axial displacement U_2 and the transverse displacement W are associated with reference surface $z = 0$. Here ϕ_2 is the rotation of a normal to the reference surface.

Stress Distribution:

$$\sigma_\beta = N_\beta/h + M_\beta z/I + K_\beta n_{2,2}(z^2 - c^2/3)/(2h) + K_\beta(c\ m_{2,2} + q)(z^3 - 3\ c^2z/5)/(6I) \tag{I-12}$$

$$\sigma_{2z} = n_2z/I + c\ m_2(3\ z^2 - c^2)/(6I) - Q_2(z^2 - c^2)/(2I) \tag{I-13}$$

$$\sigma_{zz} = p/2 - n_{2,2}(z^2 - c^2)/(2h) + q\ z/h - (c\ m_{2,2} + q)(z^3 - c^2z)/(6I) \tag{I-14}$$

where

$$K_\beta = \bar{C}_{\beta 2}S_{3\alpha}\bar{C}_{\alpha 2}/\bar{C}_{22} + \bar{C}_{\beta 2}S_{44} - \bar{C}_\beta \quad ; \quad \alpha,\beta = 1 \text{ and } 2 \tag{I-15}$$

σ_1 = surface parallel stress in x_1 direction,

σ_2 = surface parallel stress in x_2 direction,

σ_{2z} = shear stress in x_2-z plane, and

σ_{zz} = transverse normal stress in z direction.

Displacement Distribution:

$$\begin{aligned} w = W &+ S_{3\alpha}(N_\alpha z/h + M_\alpha z^2/(2I)) + S_{3\alpha}K_\alpha n_{2,2}(z^3 - c^2z)/(6h) \\ &+ S_{3\alpha}K_\alpha(c\ m_{2,2} + q)(z^4/4 - 3\ c^2z^2/10)/(6I) \\ &+ S_{33}(p\ z/2 + q\ z^2/(2h)) - S_{33}n_{2,2}(z^3 - 3\ c^2z)/(6h) \\ &+ S_{33}(c\ m_{2,2} + q)(z^4/4 - c^2z^2/2)/(6I) \end{aligned} \tag{I-16}$$

$$
\begin{aligned}
u = U_2 - z\, W_{,2} - S_{3\alpha}(N_\alpha z^2/(2h) + M_\alpha z^3/(6I))_{,2} \\
- S_{3\alpha}K_\alpha\{n_{2,22}(z^4/4 - c^2z^2/2)/(6h) \\
+ (c\, m_{2,22} + q_{,2})(z^5/20 - c^2z^3/10)/(6I)\} + S_{33}\{p_{,2}z^2/4 + q_{,2}z^3/(6h) \\
- n_{2,22}(z^4/4 - 3\, c^2z^2/2)/(6h) - (c\, m_{2,22} + q_{,2})(z^5/20 - c^2z^3/6)/6I)\} \\
+ S_{44}\{z^2n_2/(2h) + c\, m_2(z^3 - c^2z)/(6I) - Q_2(z^3 - 3\, c^2z)/(6I)\} \qquad \text{(I-17)}
\end{aligned}
$$

The Greek subscripts in the foregoing expressions assume values of 1 and 2. The repeated Greek indicies indicate a summation over the appropriate range. Finally, u_2 and w are the displacement components in the x_2 and z coordinate directions.

APPENDIX III: INPUT AND OUTPUT INSTRUCTIONS FOR THE CODE

This code containa a **MAIN** program and 16 subroutines. Additionally, it also needs a subroutine to fine the determinant and inverse of a complex matrix. The IMSL mathematical subroutine library is used for this purpose. The program is restricted to cylindrical bending problems in x_2-z plane. The laminate can be analyzed in stretching in x_1-direction or in bending in x_1-z plane.

The **MAIN** program controls input. The subroutine **MPRG** controls construction of eigen solutions for each region, construction and solution of boundary/continuity conditions matrix, and finally computation of response variables through the thickness and along the axis of the laminate. The core memory can be controlled by the dimension of RCM vector in MAIN program. Also, set the value of the variable MAXRCM to the same value as the dimension of RCM vector.

The input is divided into three parts. The first part given by Data cards 1 through 5 describes the sublaminates, ply layup, and material properties. The second part generates the eigenvalues and eigen modes. Finally, boundary/continuity conditions are input through Data card 6. The distribution of the response variables is controlled by Data card 7. The first part of the input is explained below:

Data 1: NSREG, NMT

Format(I2)...These variables, respectively, indicate the number of sublaminates which the laminate is divided into and the number of materials used. A different ply angle indicates a different material.

Data 2: EPSLON, CURVTR

Format(8E10.4)...EPSLON is the uniform strain in x_1-direction and CURVTR is the quotient of curvature in x_1-z plane.

Data 3: NL, NGN, NGT

Format(20I4)...NL is the number of layers independently analyzed in a sublaminate. A "0" for NGN constructs full solutions. A "1" gives decay type solutions for the current sublaminate. If NGT is either "0" or the current sublaminate identification number, then the full eigenvalue problem is solved for the current sublaminate. If NGT is initiated to a number less than the current sublaminate identification number, then its eigen solutions are not computed. Instead these are assumed to be the same as those of the NGT sublaminate. This means that the present sublaminate is identically the same as the NGT sublaminate. The data for NSREG sublaminates is given here.

Data 4: (THICK(K), K=1,NL)

Format(8E10.4)...Thicknesses for the NL layers of the sublaminate. The data for all the NSREG sublaminates are entered here.

Data 5: E_1, E_2, E_3, ν_{12}, ν_{13}, ν_{23}, G_{23}, G_{13}, G_{12}, θ

Format(10f8.4)...Material properties along ply principal directions. θ is the angle of orientation. The data for all the NMT materials is entered here.

The second part of the data, to compute eigen modes, is to be given interactively. This procedure can be automated if a reasonable approximation of the eigen values is known. It begins with the following series of prompts:

Prompt 1: "GIVE 0 IF YOU DO NOT WANT POLYNOMINAL APPROX.
NR = xx, nsym = xxx"

A default ("0") uses complete eigen matrix to evaluate real roots. Any other number uses a polynominal representation. Control goes to Prompt 2. NR indicates the current sublaminate for which real roots are sought. There is a do loop on NR covering all sublaminates. Sublaminates are skipped for which NGN is not initiated to "0" or their proper identification number. There is a do loop on NSYM. Pure stretching modes are constructed if NGN is 1. Pure bending modes are computed when it is 2. Both are computed when NSYM = 3. These do loops end when the reply to Prompt 5 is a default.

Prompt 2: "GIVE 0 IF YOU WANT REAL ROOTS"

A default computes real roots by going to Prompt 9. Any other number transfers to Prompt 3.

Prompt 3: "GIVE 0 IF YOU DO NOT WANT POLYNOMIAL APPROX.
NR = xx, nsym = xxx"

A default ("0") used complete eigen matrix to evaluate complex roots. Any other number uses a polynomial representation. Control goes to Prompt 4.

Prompt 4: "GIVE 0 IF YOU WANT COMPLEX ROOTS"

A default computes complex roots by going to Prompt 13. Any other number transfers to Prompt 5.

Prompt 5: "GIVE 0 IF POLYNOMIAL IS NOT DESIRED"

A default ends do loops on NR and NSYM. Any other number initiates the process for construction of a polynomial suppressing the real and complex roots constructed thus far by transferring control to Prompt 6. Here, it also shows the roots computed thus far.

Prompt 6: "GIVE NRT FOR POLYNOMIAL...DEFAULT IS AS IS A NEGATIVE VALUE
INITIATES NRT TO 0"

NRT indicates the number of real roots computed thus far for the particular cases of NR and NSYM. A negative value initiates NRT to 0. Any other number less than the current NRT initiates NRT to that value. This helps controlling the number of real roots in the polynomial evaluation. Control next goes to Prompt 7.

Prompt 7: "GIVE NCT FOR POLYNOMIAL...DEFAULT IS AS IS A NEGATIVE VALUE
INITIATES NCT TO 0"

NCT indicates the number of complex roots computed thus far for the particular cases of NR and NSYM. A negative value initiates NCT to 0. Any other number less than the current NCT initiates NCT to that value. This helps controlling the number of complex roots in the polynomial evaluation. Control goes to Prompt 8.

Prompt 8: "GIVE 0 IF POLYNOMIAL IS NOT DESIRED"

A default transfers the control to Prompt 1 without changing either NR or NSYM. Any other number computes the polynomial. Control again shifts to Prompt 1.

Prompt 9: "10F10.6...XI...INITIAL ROOT VALUE"

This commences the process for finding real roots. This is based on Newton-Raphson technique. XI is the initial root value. A default transfers control to Prompt 11. If there is convergence, the root is displayed and the control goes to Prompt 10. If there is no convergence after 100 iterations the control goes to Prompt 9 displaying the fact.

Prompt 10: "12...INPUT...J=1 TO STORE ROOT"

After the converged root is displayed, it asks whether the root is to be saved (input "1") or not (default) for the purpose of a later storage. If allowed, it stores two values for the root, +xxx and -xxx, and appends NRT by 2. If NGN is not initiated to "0" only one root is stored and NRT is increased by 1. The choice x_2-origin in relation to the edge where decay type solutions are required determines whether to store positive or negative roots. Control goes to Prompt 9. NRT starts at 0 for each NR and NSYM. At the behest of Prompt 6 this parameter can be changed to drop unwanted roots.

Prompt 11: "12...INPUT...J=0 TO FIND REAL EIGEN MODES"

A default computes the eigen modes. Next is Prompt 12.

Prompt 12: "GIVE 0 IF REAL ROOTS ARE TO BE STORED"

A default stores real roots and real modes and sets NRT back to zero. Once the roots are stored, then the polynomial at Prompt 5 cannot be constructed by suppressing these roots. Control goes to Prompt 3.

Prompts 13 through 16:

The "in" control comes from Prompt 4 and the "out" control goes to Prompt 5. These prompts do the same job as described for Prompts 9 through 12, but for complex roots. NCT keeps track of the number of complex roots to be stored. Four roots...positive, positive complex conjugate, negative, and negative complex conjugate versions will be saved and NCT will be appended by 4 if NGN is "0." Otherwise, only positive or negative roots (depends on the discretion of the user) will be saved and NCT will be padded up by two.

The logic outlined by these prompts is elementary and is useful for finding the eigen solutions. User ends this process after determining that all roots are found and stored. Once a root is stored at the instance of either Prompt 10 or 14, it will be suppressed so that the next iteration will not converge on to this root. The next set of data concerns the boundary/continuity conditions and printing of response variables.

Data 6: NBCREG, NBCPLY, NBCV, YNBC, ZNBC, PNBC, NBCREGL

Format(3I4,3E12.6,I4)...Repeat this data to cover all boundary and continuity conditions. NBCREG is the sublaminate, NBCPLY is the ply, NBCV is the variable, YNBC is the local x_2-coordinate ZNBC is the local z coordinate. The origin for the local z axis is always in the local ply middle plane. Origin for x_2-coordinate can be placed anywhere but care should be taken when using pure decay type solutions. For each sublaminate, the plys are numbered 1, 2, 3,...locally from top to bottom.

NBCV = 1	indicates	$N_2(x_2)$
NBCV = 2	indicates	$u_2(x_2, z)$
NBCV = 3	indicates	$M_2(x_2)$
NBCV = 4	indicates	$\phi_2(x_2)$
NBCV = 5	indicates	$Q_2(x_2)$
NBCV = 6	indicates	$w(x_2, z)$
NBCV = 7	indicates	$\sigma_{22}(x_2, z)$
NBCV = 8	indicates	$\sigma_{2z}(x_2, z)$
NBCV = 9	indicates	$\sigma_{zz}(x_2, z)$
NBCV = 10	indicates	transverse stretch parameter
NBCV = 11	indicates	$\sigma_{2z,2}(x_2, z)$

The tenth variable, transverse thickness stretch parameter, denotes the coefficient of z in the thicknesswise distribution of the transverse displacement. N_2, M_2, and Q_2 are defined layerwise when full solutions are used. At edges where decay type solutions are present, this definition is again retained. At far edges, however, these are defined over entire sublaminate thickness preserving local force equilibrium. For conditions involving several variables (NV's), this data continues on to the next card and is terminated by specifying NBCREGL = 0 on the last card.

For example,

```
   2   1   1 .100000d 01 .000000d 00 .120000d 03   8
   1   3   3-.100000d 01 .000000d 00-.100000d 01   8
   2   2   2 .200000d 01-.200000d 00 .900000d 01   8
   4   7   5-.100000d 01-.200000d 00 .200000d 01   0
```

indicates the following:

$$N_2 - M_2 + 9U_2 + 2Q_2 + 120 = 0$$

where,

N_2 is defined for the second sublaminate and first ply at $x_2 = 1$,
M_2 is defined for the first sublaminate and third ply at $x_2 = -1$,
u_2 is defined for the second sublaminate and second ply at $x_2 = 2$, $z = -2$,
Q_2 is defined for the fourth sublaminate and seventh ply at $x_2 = -1$.
PS in the first card carries the inhomogeneous part of the condition. In subsequent cards, PS's carry the coefficients of the second, third, and fourth variables, respectively. NBCREGL is an arbitrary number in the first three cards.

Data 7: NDREG, NDPLY, NDV, NDVT, YND, ZND, STPND, NDP

Format(4I4,3d12.6,I4)...A value "2" for NDVT gives distribution of NDV variable of the NDPLY ply in the NDREG sublaminate with respect to x_2-coordinate starting at x_2 = YND computed NDP times at in interval of STPND. A value "3" gives the distribution with respect to z coordinate. Repeat Data 7 as many times as is required. A blank card terminates the computations of all distributions.

The source code is given in appendix I. Typical input and output for a ENF specimen are shown in the next two appendices. The output is self-explanatory. The number of plys, real and complex roots, eigen modes and boundary conditions for each region are printed. The input boundary/continuity conditions are also printed.

[CBB] is the plane stress stiffness matrix. CB11, CB12,...are its elements. $\{CB\}$ is the vector referred to in equation I-11 of appendix II. SNC1 and SNC2 denote K_{n1} and K_{n2} of equation I-12. SHM2 and SHQ2 denote the coefficients of Q_2 and m_2 in equation I-9. KN1, KN2, KM1, KM2, KQ1, KQ2, KP1, and KP2 are defined in equation I-11.

The amplitudes of the governing variables are given as rows in four columns under ZERO MODE, REGION = xx and EIGENVALUE and MODE FOR REGION = xx. The governing variables in each region are assumed in the order of U_2, W, $\sigma_{2z}(x_2\ ,\ z = -c)$ and $\sigma_{zz}(x_2\ ,\ z = -c)$ for each layer in that region starting from top layer to the bottom layer. The different columns indicate the mode shape. In the case of zero modes, the first, second, third, and fourth columns denote, respectively, the coefficients of x_2^0, x_2^1, x_2^2 and x_2^3. In the case of real roots only the first two columns are meaningful. They represent the coefficients of cosh(a*x_2*2/H) and sinh(a*x_2*2/h), in that order. If NRGN is not "0," then the second column will also become zero and the first column represents the coefficient of exp(a*x_2*2/H). Symbols, a and H, denote the real root and the thickness of the sublaminate respectively. In the case of complex roots, these columns represent the coefficients of cosh(a*x_2*2/H)*cos(b*x_2*2/H), cosh(a*x_2*2/H)*sin(b*x_2*2/H), sinh(a*x_2*2/H)*cos(b*x_2*2/H) and sinh(a*x_2*2/H)*sin(b*x_2*2/H), respectively. The first and second columns would represent the coefficients of exp(a*x_2*2/H)*cos(b*x_2*2/H) and exp(a*x_2*2/H)*sin(b*x_2*2/H) if NRGN is not initiated to zero. a+i*b denotes a complex root.

The distributions are output with Y, F, Z, NR, NPLY, and NV denoting x_2-coordinate, value of the variable NV, z-coordinate, sublaminate, ply, and the variable NV, in that order.

APPENDIX IV: TYPICAL INUT FOR ENF SPECIMEN

```
4 2
 0.0000  0.0000
  1
  3   1
  1   0   0
  1
  1
  1   2   1
  1
  1
1.200D-01
0.600D-01 0.400D-04 0.600D-01
0.600D-01
0.600D-01
18.8000  1.1930  1.1930  0.2600  0.2600  0.4230  0.3620  0.6230  0.6230 90.0000
 0.5000  0.5000  0.5000  0.3500  0.3500  0.3500  0.1852  0.1852  0.1852 90.0000
0NK10
0REAL
3.75
1STORE
1.05
1STORE
0          EXIT
0 EIGEN
0STORE
0NK10
0COMPLEX
808.000000764.
1STORE
                  EXIT
 EIGEN
 STORE
0POLY
0NK10
0REAL
2.5
1STORE
.59
1STORE
0        EXIT
0EIGEN
0DO NOT STORE
0NK10
0COMPLEX
23.200000023.3
1STORE
                  EXIT
0EIGEN
0DO NOT STRORE
0POLY
0NK10
1REAL
0NK10
1COMPLEX
```

```
0POLY
 1    1    2-.100000D 01     0.000000     0.000000   0
 1    1    3-.100000D 01     0.000000     0.000000   0
 1    1    6-.100000D 01     0.000000     0.000000   0
 1    1    10.100000D 01     0.000000     0.000000   2
 2    1    1-.100000D 01     0.000000    -1.000000   0
 1    1    20.100000D 01     0.000000     0.000000   2
 2    1    2-.100000D 01-.300000D-01    -1.000000   0
 1    1    30.100000D 01     0.000000     0.000000   2
 2    1    3-.100000D 01     0.000000    -1.000000   0
 1    1    40.100000D 01     0.000000     0.000000   2
 2    1    4-.100000D 01-.300000D-01    -1.000000   0
 1    1    50.100000D 01     0.000000  .120000D 03   2
 2    1    5-.100000D 01     0.000000    -1.000000   0
 1    1    60.100000D 01     0.000000     0.000000   2
 2    1    6-.100000D 01-.300000D-01    -1.000000   0
 2    1    1      0.000000     0.000000     0.000000   3
 3    1    1      0.000000     0.000000     0.000000   0
 2    1    2      0.000000     0.000000     0.000000   3
 3    1    2      0.000000     0.000000     0.000000   0
 2    1    3      0.000000     0.000000     0.000000   3
 3    1    3      0.000000     0.000000     0.000000   0
 2    1    4      0.000000     0.000000     0.000000   3
 3    1    4      0.000000     0.000000     0.000000   0
 2    1    5      0.000000     0.000000     0.000000   3
 3    1    5      0.000000     0.000000     0.000000   0
 2    1    6      0.000000     0.000000     0.000000   3
 3    1    6      0.000000     0.000000     0.000000   0
 2    1   10      0.000000-.300000D-01     0.000000   3
 3    1   10      0.000000-.300000D-01     0.000000   0
 2    3   10      0.000000 .300000D-01     0.000000   4
 4    1   10      0.000000 .300000D-01     0.000000   0
 2    2    1      0.000000     0.000000     0.000000   0
 2    2    3      0.000000     0.000000     0.000000   0
 2    2    5      0.000000     0.000000     0.000000   0
 2    3    1      0.000000     0.000000     0.000000   4
 4    1    1      0.000000     0.000000     0.000000   0
 2    3    2      0.000000     0.000000     0.000000   4
 4    1    2      0.000000     0.000000     0.000000   0
 2    3    3      0.000000     0.000000     0.000000   4
 4    1    3      0.000000     0.000000     0.000000   0
 2    3    4      0.000000     0.000000     0.000000   4
 4    1    4      0.000000     0.000000     0.000000   0
 2    3    5      0.000000     0.000000     0.000000   4
 4    1    5      0.000000     0.000000     0.000000   0
 2    3    6      0.000000     0.000000     0.000000   4
 4    1    6      0.000000     0.000000     0.000000   0
 3    1    11.000000D 00     0.000000     0.000000   0
 3    1    31.000000D 00     0.000000     0.000000   0
 3    1    51.000000D 00     0.000000     0.000000   0
 4    1    11.000000D 00     0.000000     0.000000   0
 4    1    31.000000D 00     0.000000     0.000000   0
 4    1    61.000000D 00     0.000000     0.000000   0
 1    1    6    2-.100000D 01-.600000D-011.000000D 00    2
```

```
1   1   1   2-.100000D 01-.000000D-011.000000D 00    2
1   1   3   2-.100000D 01-.000000D-011.000000D 00    2
1   1   5   2-.100000D 01-.000000D-011.000000D 00    2
2   1   1   2-.100000D 01-.000000D-010.500000D 00    2
2   1   3   2-.100000D 01-.000000D-010.500000D 00    2
2   1   5   2-.100000D 01-.000000D-010.500000D 00    2
2   2   1   2-.100000D 01-.000000D-010.500000D 00    2
2   2   3   2-.100000D 01-.000000D-010.500000D 00    2
2   2   5   2-.100000D 01-.000000D-010.500000D 00    2
2   3   1   2-.100000D 01-.000000D-010.500000D 00    2
2   3   3   2-.100000D 01-.000000D-010.500000D 00    2
2   3   5   2-.100000D 01-.000000D-010.500000D 00    2
3   1   1   2-.000000D 01-.000000D-010.500000D 00    2
3   1   3   2-.000000D 01-.000000D-010.500000D 00    2
3   1   5   2-.000000D 01-.000000D-010.500000D 00    2
4   1   1   2-.000000D 01-.000000D-010.500000D 00    2
4   1   3   2-.000000D 01-.000000D-010.500000D 00    2
4   1   5   2-.000000D 01-.000000D-010.500000D 00    2
2   2   7   2     0.000000     0.000000-.625000D-02 160
2   2   8   2     0.000000     0.000000-.625000D-02 160
2   2   9   2     0.000000     0.000000-.625000D-02 160

/EOF
```

APPENDIX V: TYPICAL OUTPUT FOR ENF SPECIMEN

```
 4  2   NO. OF REGIONS AND NO. OF MATERIALS

 0.0000E+000.0000E+00  EPSLON,  CURVTR

 REGION  =    1,  NO. OF LAYERS  =    1,  NGN  =    0,  NGT  =    0
 REGION  =    2,  NO. OF LAYERS  =    3,  NGN  =    1,  NGT  =    0
 REGION  =    3,  NO. OF LAYERS  =    1,  NGN  =    0,  NGT  =    0
 REGION  =    4,  NO. OF LAYERS  =    1,  NGN  =    0,  NGT  =    0

 REGION  =    1,  MATERIAL FOR EACH LAYER
   1

 REGION  =    2,  MATERIAL FOR EACH LAYER
   1   2   1

 REGION  =    3,  MATERIAL FOR EACH LAYER
   1

 REGION  =    4,  MATERIAL FOR EACH LAYER
   1

 REGION  =    1,  PLY THICKNESSES
0.1200E+00

 REGION  =    2,  PLY THICKNESSES
0.6000E-010.4000E-040.6000E-01

 REGION  =    3,  PLY THICKNESSES
0.6000E-01

 REGION  =    4,  PLY THICKNESSES
0.6000E-01

 SUBLAMINATE THICKNESSES
0.1200E+000.1200E+000.6000E-010.6000E-01
1

 E11,E22,E33,NU12,NU13,NU23,G44,G55,G66,THETA
 0.1880E+020.1193E+010.1193E+010.2600E+000.2600E+000.4230E+000.3620E+000.6230E+00
 0.6230E+000.9000E+02
 0.5000E+000.5000E+000.5000E+000.3500E+000.3500E+000.3500E+000.1852E+000.1852E+00
 0.1852E+000.9000E+02

 FLEXIBILITY MATRIX S FOR  1 MATERIAL
 0.8382E+00-.1383E-01-.3546E+000.0000E+000.0000E+000.1697E-15
```

```
-.1383E-010.5319E-01-.1383E-010.0000E+000.0000E+000.2523E-14
-.3546E+00-.1383E-010.8382E+000.0000E+000.0000E+00-.1169E-14
0.0000E+000.0000E+000.0000E+000.1605E+010.1985E-140.0000E+00
0.0000E+000.0000E+000.0000E+000.1985E-140.2762E+010.0000E+00
0.1697E-150.2523E-14-.1169E-140.0000E+000.0000E+000.1605E+01

STIFFNESS MATRIX C FOR  1 MATERIAL
0.1469E+010.5457E+000.6302E+000.0000E+000.0000E+00-.5542E-15
0.5457E+000.1908E+020.5457E+000.0000E+000.0000E+00-.2966E-13
0.6302E+000.5457E+000.1469E+010.0000E+000.0000E+000.1450E-15
0.0000E+000.0000E+000.0000E+000.6230E+00-.4476E-150.0000E+00
0.0000E+000.0000E+000.0000E+00-.4476E-150.3620E+000.0000E+00
-.5542E-15-.2966E-130.1450E-150.0000E+000.0000E+000.6230E+00

PLANESTRESS STIFFNESS MATRIX CBB, CB OF
0.1198E+010.3115E+00-.6164E-150.4291E+00
0.3115E+000.1888E+02-.2971E-130.3716E+00
-.6164E-15-.2971E-130.6230E+000.9872E-16

FLEXIBILITY MATRIX S FOR  2 MATERIAL
0.2000E+01-.7000E+00-.7000E+000.0000E+000.0000E+000.7409E-18
-.7000E+000.2000E+01-.7000E+000.0000E+000.0000E+00-.7409E-18
-.7000E+00-.7000E+000.2000E+010.0000E+000.0000E+000.0000E+00
0.0000E+000.0000E+000.0000E+000.5400E+010.0000E+000.0000E+00
0.0000E+000.0000E+000.0000E+000.0000E+000.5400E+010.0000E+00
0.7409E-18-.7409E-180.0000E+000.0000E+000.0000E+000.5400E+01

STIFFNESS MATRIX C FOR  2 MATERIAL
0.8025E+000.4321E+000.4321E+000.0000E+000.0000E+00-.5082E-19
0.4321E+000.8025E+000.4321E+000.0000E+000.0000E+000.5082E-19
0.4321E+000.4321E+000.8025E+000.0000E+000.0000E+00-.4636E-33
0.0000E+000.0000E+000.0000E+000.1852E+000.0000E+000.0000E+00
0.0000E+000.0000E+000.0000E+000.0000E+000.1852E+000.0000E+00
-.5082E-190.5082E-19-.4636E-330.0000E+000.0000E+000.1852E+00

PLANESTRESS STIFFNESS MATRIX CBB, CB OF
0.5698E+000.1994E+00-.5082E-190.5385E+00
0.1994E+000.5698E+000.5082E-190.5385E+00
-.5082E-190.5082E-190.1852E+00-.5778E-33

CB11,CB12,CB16,CB22,CB26,CB66,CB1,CB2,CB6
0.1198E+010.3115E+00-.6164E-150.1888E+02-.2971E-130.6230E+000.4291E+000.3716E+00
0.0000E+00
0.5698E+000.1994E+00-.5082E-190.5698E+000.5082E-190.1852E+000.5385E+000.5385E+00
0.0000E+00

S31,S32,S33,S44,S45,S55,SNC1,SNC2,SHM2,SHQ2
-.3546E+00-.1383E-010.8382E+000.1605E+010.1985E-140.2762E+010.6477E-010.2956E+02
-.4013E+000.1204E+01
-.7000E+00-.7000E+000.2000E+010.5400E+010.0000E+000.5400E+010.3499E+000.2000E+01
-.1350E+010.4050E+01

KN1,KN2,KP1,KP2,KM1,KM2,KQ1,KQ2
0.1127E+000.2557E+010.2146E+000.1858E+00-.8009E-02-.3049E+01-.4346E-01-.1802E+02
```

```
0.1638E+000.3013E+000.2692E+000.2692E+00-.8211E-01-.3346E+00-.3513E+00-.1604E+01
  79   80   81   82   83   87   91   95  101  107  111  115  119  123  127  131  211   40   10 3411
3427 5027 5067 5107 5147 5187 5227
1
 PARTICULAR SOLUTION FOR REGION =  1
 0.0000E+00 0.0000E+00

    1 ZERO MODE,     1 REGION
         0.000000         0.000000         0.000000         0.000000
         0.000000         1.000000         0.000000         0.000000

    2 ZERO MODE,     1 REGION
         0.000000         0.000000         0.000000         0.000000
         1.000000         0.000000         0.000000         0.000000

    3 ZERO MODE,     1 REGION
         0.000000         0.000000         0.000000         0.000000
         0.000000         0.000000         0.000000         1.000000

    4 ZERO MODE,     1 REGION
         0.000000         0.000000         0.000000         0.000000
         0.000000         0.000000         1.000000         0.000000

    5 ZERO MODE,     1 REGION
         1.000000         0.000000         0.000000         0.000000
         0.000000         0.000000         0.000000         0.000000

    6 ZERO MODE,     1 REGION
         0.000000         1.000000         0.000000         0.000000
         0.000000         0.000000         0.000000         0.000000
1
 PARTICULAR SOLUTION FOR REGION =  2
 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00

    7 ZERO MODE,     2 REGION
         0.000000         0.000000         0.000000         0.000000
         0.000000         1.000000         0.000000         0.000000
         0.000000         0.000000         0.000000         0.000000
         0.000000         0.000000         0.000000         0.000000
       750.249917         0.000000         0.000000         0.000000
         0.000000      1500.000000         0.000000         0.000000
         0.000000         0.000000         0.000000         0.000000
         0.000000         0.000000         0.000000         0.000000
         1.000333         0.000000         0.000000         0.000000
         0.000000         1.000000         0.000000         0.000000

    8 ZERO MODE,     2 REGION
         0.000000         0.000000         0.000000         0.000000
         1.000000         0.000000         0.000000         0.000000
         0.000000         0.000000         0.000000         0.000000
         0.000000         0.000000         0.000000         0.000000
         0.000000         0.000000         0.000000         0.000000
         1.000000         0.000000         0.000000         0.000000
         0.000000         0.000000         0.000000         0.000000
```

```
        0.000000          0.000000          0.000000          0.000000
        0.000000          0.000000          0.000000          0.000000
        1.000000          0.000000          0.000000          0.000000

   9 ZERO MODE,     2 REGION
        0.000000          0.000000         -1.500500          0.000000
        0.000000          0.000000          0.000000          1.000000
      -28.312051          0.000000          0.000000          0.000000
        0.000000          0.000000          0.000000          0.000000
    62492.292815          0.000000          0.000000          0.000000
        0.000000       -418.299411          0.000000       1500.000000
      -28.312051          0.000000          0.000000          0.000000
        0.000000          0.000000          0.000000          0.000000
       83.323057          0.000000          1.500500          0.000000
        0.000000          0.000000          0.000000          1.000000

  10 ZERO MODE,     2 REGION
        0.000000          0.000000          0.000000          0.000000
        0.000000          0.000000          1.000000          0.000000
        0.000000          0.000000          0.000000          0.000000
        0.000000          0.000000          0.000000          0.000000
        0.000000       1500.499833          0.000000          0.000000
      139.516525          0.000000       1500.000000          0.000000
        0.000000          0.000000          0.000000          0.000000
        0.000000          0.000000          0.000000          0.000000
        0.000000          2.000666          0.000000          0.000000
        0.371933          0.000000          1.000000          0.000000

  11 ZERO MODE,     2 REGION
        1.000000          0.000000          0.000000          0.000000
        0.000000          0.000000          0.000000          0.000000
        0.000000          0.000000          0.000000          0.000000
        0.000000          0.000000          0.000000          0.000000
        1.000000          0.000000          0.000000          0.000000
        0.000000          0.000000          0.000000          0.000000
        0.000000          0.000000          0.000000          0.000000
        0.000000          0.000000          0.000000          0.000000
        1.000000          0.000000          0.000000          0.000000
        0.000000          0.000000          0.000000          0.000000

  12 ZERO MODE,     2 REGION
        0.000000          1.000000          0.000000          0.000000
        0.000000          0.000000          0.000000          0.000000
        0.000000          0.000000          0.000000          0.000000
        0.000000          0.000000          0.000000          0.000000
        0.000000       1500.000000          0.000000          0.000000
      278.856741          0.000000          0.000000          0.000000
        0.030000          0.000000          0.000000          0.000000
        0.000000          0.000000          0.000000          0.000000
        0.000000          1.000000          0.000000          0.000000
        0.371809          0.000000          0.000000          0.000000
GIVE  0   IF YOU DO NOT WANT POLYNOMIAL APPROX.
NR =    2  NSYM =    1
GIVE  0   IF YOU WANT REAL   ROOTS
```

```
10F10.6....XI..INPUT INITIAL ROOT VALUE
  1  0.37500000E+01  0.47319991E-07  I,XI,FI
  2  0.37480371E+01  0.22840498E-10  I,XI,FI
  3  0.37480362E+01  0.96024904E-13  I,XI,FI
  4  0.37480362E+01  0.96513838E-17  I,XI,FI
  1  0.37480362E+01  0.96513838E-17  I,XI,FI
I2...INPUT...J=1 TO STORE ROOT
10F10.6....XI..INPUT INITIAL ROOT VALUE
  1  0.10500000E+01  0.42971090E-08  I,XI,FI
  2  0.10417340E+01  0.15428486E-10  I,XI,FI
  3  0.10417041E+01 -0.86109741E-14  I,XI,FI
  4  0.10417041E+01 -0.19605414E-17  I,XI,FI
  2  0.10417041E+01 -0.19605414E-17  I,XI,FI
I2...INPUT...J=1 TO STORE ROOT
10F10.6....XI..INPUT INITIAL ROOT VALUE
I2...INPUT J=0 TO FIND REAL   EIGEN MODES

  12    2    1  0.37480362E+01 REAL   EIGEN VALUE AND MODE
0.1000E+01 -.5208E+00 0.6995E+01 -.8737E-02 0.0000E+00 0.2607E+04
0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00

  12    2    2  0.10417041E+01 REAL   EIGEN VALUE AND MODE
0.1000E+01 0.1080E+01 0.4290E+01 -.1489E-02 0.0000E+00 0.2479E+04
0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
GIVE  0   IF REAL ROOTS ARE TO BE STORED

  13  0.37480362E+01 0.00000000E+00 EIGEN VALUE AND MODE FOR REGION =   2
         1.000000          0.000000           0.000000           0.000000
        -0.520829          0.000000           0.000000           0.000000
         6.995287          0.000000           0.000000           0.000000
        -0.008737          0.000000           0.000000           0.000000
         0.000000          0.000000           0.000000           0.000000
      2607.393034          0.000000           0.000000           0.000000
         6.995287          0.000000           0.000000           0.000000
         0.008737          0.000000           0.000000           0.000000
        -1.000000          0.000000           0.000000           0.000000
        -0.520829          0.000000           0.000000           0.000000

  14  0.10417041E+01 0.00000000E+00 EIGEN VALUE AND MODE FOR REGION =   2
         1.000000          0.000000           0.000000           0.000000
         1.080237          0.000000           0.000000           0.000000
         4.290226          0.000000           0.000000           0.000000
        -0.001489          0.000000           0.000000           0.000000
         0.000000          0.000000           0.000000           0.000000
      2478.639427          0.000000           0.000000           0.000000
         4.290226          0.000000           0.000000           0.000000
         0.001489          0.000000           0.000000           0.000000
        -1.000000          0.000000           0.000000           0.000000
         1.080237          0.000000           0.000000           0.000000
GIVE  0   IF YOU DO NOT WANT POLYNOMIAL APPROX.
NR =    2  NSYM =    1
GIVE  0   IF YOU WANT COMPLEX ROOTS
2F10.6....XI..INPUT INITIAL COMPLEX ROOT VALUE
  1  0.80800000E+03 0.76400000E+03-0.12036548E+04-0.61901902E+03 I,XI,FI
```

```
  2  0.80895690E+03 0.76561962E+03 0.13470563E+04-0.33421342E+03 I,XI,FI
  3  0.80590462E+03 0.76837678E+03 0.17262976E+04 0.53075139E+04 I,XI,FI
  4  0.80557780E+03 0.76824207E+03 0.12990180E+04 0.55361773E+04 I,XI,FI
  5  0.80799898E+03 0.76479830E+03-0.33968396E+03 0.32976972E+02 I,XI,FI
  6  0.80814428E+03 0.76491454E+03-0.94234759E+02-0.29204353E+02 I,XI,FI
*** TERMINAL ERROR          (IER = 129) FROM IMSL ROUTINE LEQT1C
*** TERMINAL ERROR          (IER = 129) FROM IMSL ROUTINE LEQT1C
*** TERMINAL ERROR          (IER = 129) FROM IMSL ROUTINE LEQT1C
  7  0.80814522E+03 0.76503220E+03-0.34344867E+02-0.66588807E+02 I,XI,FI
*** TERMINAL ERROR          (IER = 129) FROM IMSL ROUTINE LEQT1C
*** TERMINAL ERROR          (IER = 129) FROM IMSL ROUTINE LEQT1C
*** TERMINAL ERROR          (IER = 129) FROM IMSL ROUTINE LEQT1C
  8  0.80816312E+03 0.76498867E+03-0.17958008E+01-0.11299457E+02 I,XI,FI
*** TERMINAL ERROR          (IER = 129) FROM IMSL ROUTINE LEQT1C
*** TERMINAL ERROR          (IER = 129) FROM IMSL ROUTINE LEQT1C
*** TERMINAL ERROR          (IER = 129) FROM IMSL ROUTINE LEQT1C
  9  0.80816635E+03 0.76497798E+03 0.71629596E+01 0.10026871E+01 I,XI,FI
*** TERMINAL ERROR          (IER = 129) FROM IMSL ROUTINE LEQT1C
*** TERMINAL ERROR          (IER = 129) FROM IMSL ROUTINE LEQT1C
*** TERMINAL ERROR          (IER = 129) FROM IMSL ROUTINE LEQT1C
 10  0.80816691E+03 0.764[illegible]84E+03 0.14287519E+01-0.23833712E+01 I,XI,FI
*** TERMINAL ERROR          (IER = 129) FROM IMSL ROUTINE LEQT1C
*** TERMINAL ERROR          (IER = 129) FROM IMSL ROUTINE LEQT1C
*** TERMINAL ERROR          (IER = 129) FROM IMSL ROUTINE LEQT1C
 11  0.80816893E+03 0.76498343E+03-0.88597704E+00-0.67733064E+00 I,XI,FI
*** TERMINAL ERROR          (IER = 129) FROM IMSL ROUTINE LEQT1C
*** TERMINAL ERROR          (IER = 129) FROM IMSL ROUTINE LEQT1C
*** TERMINAL ERROR          (IER = 129) FROM IMSL ROUTINE LEQT1C
 12  0.80816902E+03 0.76498272E+03-0.17799826E+00 0.67669535E-02 I,XI,FI
*** TERMINAL ERROR          (IER = 129) FROM IMSL ROUTINE LEQT1C
*** TERMINAL ERROR          (IER = 129) FROM IMSL ROUTINE LEQT1C
*** TERMINAL ERROR          (IER = 129) FROM IMSL ROUTINE LEQT1C
 13  0.80816889E+03 0.76498256E+03 0.96409432E-01-0.25189725E-02 I,XI,FI
*** TERMINAL ERROR          (IER = 129) FROM IMSL ROUTINE LEQT1C
*** TERMINAL ERROR          (IER = 129) FROM IMSL ROUTINE LEQT1C
*** TERMINAL ERROR          (IER = 129) FROM IMSL ROUTINE LEQT1C
 14  0.80816897E+03 0.76498263E+03-0.42219723E-01 0.19949809E-01 I,XI,FI
*** TERMINAL ERROR          (IER = 129) FROM IMSL ROUTINE LEQT1C
*** TERMINAL ERROR          (IER = 129) FROM IMSL ROUTINE LEQT1C
*** TERMINAL ERROR          (IER = 129) FROM IMSL ROUTINE LEQT1C
 15  0.80816890E+03 0.76498256E+03 0.82382746E-01 0.68086943E-02 I,XI,FI
*** TERMINAL ERROR          (IER = 129) FROM IMSL ROUTINE LEQT1C
*** TERMINAL ERROR          (IER = 129) FROM IMSL ROUTINE LEQT1C
*** TERMINAL ERROR          (IER = 129) FROM IMSL ROUTINE LEQT1C
 16  0.80816893E+03 0.76498261E+03 0.64987188E-02-0.94443887E-03 I,XI,FI
*** TERMINAL ERROR          (IER = 129) FROM IMSL ROUTINE LEQT1C
*** TERMINAL ERROR          (IER = 129) FROM IMSL ROUTINE LEQT1C
*** TERMINAL ERROR          (IER = 129) FROM IMSL ROUTINE LEQT1C
 17  0.80816893E+03 0.76498261E+03 0.29928882E-02-0.18379697E-02 I,XI,FI
  1  0.80816893E+03 0.76498261E+03 0.29928882E-02-0.18379697E-02 I,XI,FI
I2...INPUT...J=1 TO STORE ROOT
2F10.6....XI..INPUT INITIAL COMPLEX ROOT VALUE
I2...INPUT J=0 TO FIND COMPLEX EIGEN MODES
```

```
  14    2    1 -0.80816893E+03 0.76498261E+03 COMPLEX EIGEN VALUE AND MODE
0.1000E+01 0.0000E+00 0.1000E+01 0.0000E+00
0.2503E-02 -.2370E-02 -.2503E-02 -.2370E-02
0.7625E+01 -.6344E-05 0.7625E+01 0.6344E-05
0.1333E+04 0.1262E+04 -.1333E+04 0.1262E+04
0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
-.3856E+06 -.3388E+06 0.3856E+06 -.3388E+06
GIVE  0   IF COMPLEX ROOTS ARE TO BE STORED

  15  0.80816893E+03 0.76498261E+03 EIGEN VALUE AND MODE FOR REGION =   2
          1.000000            0.000000            0.000000            0.000000
          0.002503            0.002370            0.000000            0.000000
          7.625240            0.000006            0.000000            0.000000
       1332.934249        -1261.588821            0.000000            0.000000
          0.000000            0.000000            0.000000            0.000000
    -385552.250163       338805.724230            0.000000            0.000000
          7.625240            0.000006            0.000000            0.000000
      -1332.934249         1261.588821            0.000000            0.000000
         -1.000000            0.000000            0.000000            0.000000
          0.002503            0.002370            0.000000            0.000000

  16  0.80816893E+03 0.76498261E+03 EIGEN VALUE AND MODE FOR REGION =   2
          0.000000            1.000000            0.000000            0.000000
         -0.002370            0.002503            0.000000            0.000000
         -0.000006            7.625240            0.000000            0.000000
       1261.588821         1332.934249            0.000000            0.000000
          0.000000            0.000000            0.000000            0.000000
    -338805.724230      -385552.250163            0.000000            0.000000
         -0.000006            7.625240            0.000000            0.000000
      -1261.588821        -1332.934249            0.000000            0.000000
          0.000000           -1.000000            0.000000            0.000000
         -0.002370            0.002503            0.000000            0.000000

GIVE 0   IF NO POLYNOMIAL DESIRED
GIVE  0   IF YOU DO NOT WANT POLYNOMIAL APPROX.
NR =    2  NSYM =    2
GIVE  0   IF YOU WANT REAL   ROOTS
10F10.6....XI..INPUT INITIAL ROOT VALUE
  1  0.25000000E+01  0.56359421E-06  I,XI,FI
  2  0.25038045E+01 -0.26547414E-08  I,XI,FI
  3  0.25037867E+01  0.68529545E-12  I,XI,FI
  4  0.25037867E+01  0.27345368E-15  I,XI,FI
  1  0.25037867E+01  0.27345368E-15  I,XI,FI
I2...INPUT...J=1 TO STORE ROOT
10F10.6....XI..INPUT INITIAL ROOT VALUE
  1  0.59000000E+00  0.48992205E-07  I,XI,FI
  2  0.59822567E+00 -0.33901159E-09  I,XI,FI
  3  0.59816954E+00 -0.77504131E-13  I,XI,FI
  4  0.59816953E+00  0.10187731E-16  I,XI,FI
  2  0.59816953E+00  0.10187731E-16  I,XI,FI
I2...INPUT...J=1 TO STORE ROOT
10F10.6....XI..INPUT INITIAL ROOT VALUE
I2...INPUT J=0 TO FIND REAL   EIGEN MODES
```

```
  16    2    1  0.25037867E+01 REAL   EIGEN VALUE AND MODE
0.1000E+01 -.7295E+02 0.3782E-01 -.1265E+03 0.4788E+05 0.0000E+00
0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00

  16    2    2  0.59816953E+00 REAL   EIGEN VALUE AND MODE
0.1000E+01 -.2226E+02 0.5873E-03 -.3038E+02 0.1181E+06 0.0000E+00
0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
GIVE  0   IF REAL ROOTS ARE TO BE STORED

  17  0.25037867E+01 0.00000000E+00 EIGEN VALUE AND MODE FOR REGION =   2
          1.000000            0.000000            0.000000            0.000000
        -72.950252            0.000000            0.000000            0.000000
          0.037818            0.000000            0.000000            0.000000
       -126.451326            0.000000            0.000000            0.000000
      47877.904548            0.000000            0.000000            0.000000
          0.000000            0.000000            0.000000            0.000000
         -0.037818            0.000000            0.000000            0.000000
       -126.451326            0.000000            0.000000            0.000000
          1.000000            0.000000            0.000000            0.000000
         72.950252            0.000000            0.000000            0.000000

  18  0.59816953E+00 0.00000000E+00 EIGEN VALUE AND MODE FOR REGION =   2
          1.000000            0.000000            0.000000            0.000000
        -22.257227            0.000000            0.000000            0.000000
          0.000587            0.000000            0.000000            0.000000
        -30.377292            0.000000            0.000000            0.000000
     118078.602044            0.000000            0.000000            0.000000
          0.000000            0.000000            0.000000            0.000000
         -0.000587            0.000000            0.000000            0.000000
        -30.377292            0.000000            0.000000            0.000000
          1.000000            0.000000            0.000000            0.000000
         22.257227            0.000000            0.000000            0.000000
GIVE  0   IF YOU DO NOT WANT POLYNOMIAL APPROX.
NR =     2  NSYM =    2
GIVE  0   IF YOU WANT COMPLEX ROOTS
2F10.6....XI..INPUT INITIAL COMPLEX ROOT VALUE
  1  0.23200000E+02 0.23300000E+02-0.76523440E-01 0.10615611E+00 I,XI,FI
  2  0.23027721E+02 0.23327779E+02-0.12220121E-02 0.48153943E-02 I,XI,FI
  3  0.23021458E+02 0.23331900E+02 0.55482352E-04-0.36123335E-03 I,XI,FI
*** TERMINAL ERROR            (IER = 129) FROM IMSL ROUTINE LEQT1C
*** TERMINAL ERROR            (IER = 129) FROM IMSL ROUTINE LEQT1C
*** TERMINAL ERROR            (IER = 129) FROM IMSL ROUTINE LEQT1C
  4  0.23021859E+02 0.23331572E+02 0.38146148E-05 0.37013959E-05 I,XI,FI
*** TERMINAL ERROR            (IER = 129) FROM IMSL ROUTINE LEQT1C
*** TERMINAL ERROR            (IER = 129) FROM IMSL ROUTINE LEQT1C
*** TERMINAL ERROR            (IER = 129) FROM IMSL ROUTINE LEQT1C
  5  0.23021859E+02 0.23331580E+02 0.59870490E-06-0.52050829E-06 I,XI,FI
*** TERMINAL ERROR            (IER = 129) FROM IMSL ROUTINE LEQT1C
*** TERMINAL ERROR            (IER = 129) FROM IMSL ROUTINE LEQT1C
*** TERMINAL ERROR            (IER = 129) FROM IMSL ROUTINE LEQT1C
  6  0.23021860E+02 0.23331580E+02-0.81667991E-08-0.25195860E-09 I,XI,FI
  1  0.23021860E+02 0.23331580E+02-0.81667991E-08-0.25195860E-09 I,XI,FI
I2...INPUT...J=1 TO STORE ROOT
2F10.6....XI..INPUT INITIAL COMPLEX ROOT VALUE
```

```
I2...INPUT J=0 TO FIND COMPLEX EIGEN MODES

  18    2    1 -0.23021860E+02 0.23331580E+02 COMPLEX EIGEN VALUE AND MODE
0.1000E+01 0.0000E+00 0.1000E+01 0.0000E+00
0.3914E-01 -.3640E-01 -.3914E-01 -.3640E-01
0.7545E+01 0.8589E-02 0.7545E+01 -.8589E-02
0.2475E+02 0.2607E+02 -.2475E+02 0.2607E+02
-.1761E+04 0.1109E+06 -.1761E+04 -.1109E+06
0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
GIVE  0   IF COMPLEX ROOTS ARE TO BE STORED

  19  0.23021860E+02 0.23331580E+02 EIGEN VALUE AND MODE FOR REGION =    2
          1.000000          0.000000            0.000000            0.000000
          0.039140          0.036398            0.000000            0.000000
          7.544946         -0.008589            0.000000            0.000000
         24.748159        -26.073365            0.000000            0.000000
      -1760.793992    -110928.639935            0.000000            0.000000
          0.000000          0.000000            0.000000            0.000000
         -7.544946          0.008589            0.000000            0.000000
         24.748159        -26.073365            0.000000            0.000000
          1.000000          0.000000            0.000000            0.000000
         -0.039140         -0.036398            0.000000            0.000000

  20  0.23021860E+02 0.23331580E+02 EIGEN VALUE AND MODE FOR REGION =    2
          0.000000          1.000000            0.000000            0.000000
         -0.036398          0.039140            0.000000            0.000000
          0.008589          7.544946            0.000000            0.000000
         26.073365         24.748159            0.000000            0.000000
     110928.639935      -1760.793992            0.000000            0.000000
          0.000000          0.000000            0.000000            0.000000
         -0.008589         -7.544946            0.000000            0.000000
         26.073365         24.748159            0.000000            0.000000
          0.000000          1.000000            0.000000            0.000000
          0.036398         -0.039140            0.000000            0.000000
GIVE 0   IF NO POLYNOMIAL DESIRED
GIVE  0   IF YOU DO NOT WANT POLYNOMIAL APPROX.
NR =     2  NSYM =    3
GIVE  0   IF YOU WANT REAL   ROOTS
GIVE  0   IF YOU DO NOT WANT POLYNOMIAL APPROX.
NR =     2  NSYM =    3
GIVE  0   IF YOU WANT COMPLEX ROOTS

GIVE 0   IF NO POLYNOMIAL DESIRED
1
PARTICULAR SOLUTION FOR REGION =  3
0.0000E+00 0.0000E+00

  21 ZERO MODE,      3 REGION
          0.000000          0.000000            0.000000            0.000000
          0.000000          1.000000            0.000000            0.000000

  22 ZERO MODE,      3 REGION
          0.000000          0.000000            0.000000            0.000000
```

```
          1.000000          0.000000          0.000000          0.000000

     23 ZERO MODE,      3 REGION
          0.000000          0.000000          0.000000          0.000000
          0.000000          0.000000          0.000000          1.000000

     24 ZERO MODE,      3 REGION
          0.000000          0.000000          0.000000          0.000000
          0.000000          0.000000          1.000000          0.000000

     25 ZERO MODE,      3 REGION
          1.000000          0.000000          0.000000          0.000000
          0.000000          0.000000          0.000000          0.000000

     26 ZERO MODE,      3 REGION
          0.000000          1.000000          0.000000          0.000000
          0.000000          0.000000          0.000000          0.000000
1
 PARTICULAR SOLUTION FOR REGION =   4
 0.0000E+00 0.0000E+00

     27 ZERO MODE,      4 REGION
          0.000000          0.000000          0.000000          0.000000
          0.000000          1.000000          0.000000          0.000000

     28 ZERO MODE,      4 REGION
          0.000000          0.000000          0.000000          0.000000
          1.000000          0.000000          0.000000          0.000000

     29 ZERO MODE,      4 REGION
          0.000000          0.000000          0.000000          0.000000
          0.000000          0.000000          0.000000          1.000000

     30 ZERO MODE,      4 REGION
          0.000000          0.000000          0.000000          0.000000
          0.000000          0.000000          1.000000          0.000000

     31 ZERO MODE,      4 REGION
          1.000000          0.000000          0.000000          0.000000
          0.000000          0.000000          0.000000          0.000000

     32 ZERO MODE,      4 REGION
          0.000000          1.000000          0.000000          0.000000
          0.000000          0.000000          0.000000          0.000000
   NO. OF PLYS IN EACH REGION             1    3    1    1
   NO. OF ZERO ROOTS IN EACH REGION       6    6    6    6
   NO. OF REAL ROOTS IN EACH REGION       0    4    0    0
   NO. OF COMP. ROOTS IN EACH REGION      0    4    0    0
   NO. OF B.  CONDITIONS IN EACH REGION   6   14    6    6
1
   NO. OF VARIABLES IN EACH REGION        2   10    2    2

    32 = NTBC
```

```
NBCREG,NBCPLY,NBCV,YNBC,ZNBC,PNBC,NBCREG
    1    1    2-.100000E+010.000000E+000.000000E+00    0
    1    1    3-.100000E+010.000000E+000.000000E+00    0
    1    1    6-.100000E+010.000000E+000.000000E+00    0
    1    1    10.100000E+010.000000E+000.000000E+30    2
    2    1    1-.100000E+010.000000E+00-.100000E+01    0
    1    1    20.100000E+010.000000E+000.000000E+00    2
    2    1    2-.100000E+01-.300000E-01-.100000E+01    0
    1    1    30.100000E+010.000000E+000.000000E+00    2
    2    1    3-.100000E+010.000000E+00-.100000E+01    0
    1    1    40.100000E+010.000000E+000.000000E+00    2
    2    1    4-.100000E+01-.300000E-01-.100000E+01    0
    1    1    50.100000E+010.000000E+000.120000E+03    2
    2    1    5-.100000E+010.000000E+00-.100000E+01    0
    1    1    60.100000E+010.000000E+000.000000E+00    2
    2    1    6-.100000E+01-.300000E-01-.100000E+01    0
    2    1    10.000000E+000.000000E+000.000000E+00    3
    3    1    10.000000E+000.000000E+000.000000E+00    0
    2    1    20.000000E+000.000000E+000.000000E+00    3
    3    1    20.000000E+000.000000E+000.000000E+00    0
    2    1    30.000000E+000.000000E+000.000000E+00    3
    3    1    30.000000E+000.000000E+000.000000E+00    0
    2    1    40.000000E+000.000000E+000.000000E+00    3
    3    1    40.000000E+000.000000E+000.000000E+00    0
    2    1    50.000000E+000.000000E+000.000000E+00    3
    3    1    50.000000E+000.000000E+000.000000E+00    0
    2    1    60.000000E+000.000000E+000.000000E+00    3
    3    1    60.000000E+000.000000E+000.000000E+00    0
    2    1   100.000000E+00-.300000E-010.000000E+00    3
    3    1   100.000000E+00-.300000E-010.000000E+00    0
    2    3   100.000000E+000.300000E-010.000000E+00    4
    4    1   100.000000E+000.300000E-010.000000E+00    0
    2    2    10.000000E+000.000000E+000.000000E+00    0
    2    2    30.000000E+000.000000E+000.000000E+00    0
    2    2    50.000000E+000.000000E+000.000000E+00    0
    2    3    10.000000E+000.000000E+000.000000E+00    4
    4    1    10.000000E+000.000000E+000.000000E+00    0
    2    3    20.000000E+000.000000E+000.000000E+00    4
    4    1    20.000000E+000.000000E+000.000000E+00    0
    2    3    30.000000E+000.000000E+000.000000E+00    4
    4    1    30.000000E+000.000000E+000.000000E+00    0
    2    3    40.000000E+000.000000E+000.000000E+00    4
    4    1    40.000000E+000.000000E+000.000000E+00    0
    2    3    50.000000E+000.000000E+000.000000E+00    4
    4    1    50.000000E+000.000000E+000.000000E+00    0
    2    3    60.000000E+000.000000E+000.000000E+00    4
    4    1    60.000000E+000.000000E+000.000000E+00    0
    3    1    10.100000E+010.000000E+000.000000E+00    0
    3    1    30.100000E+010.000000E+000.000000E+00    0
    3    1    50.100000E+010.000000E+000.000000E+00    0
    4    1    10.100000E+010.000000E+000.000000E+00    0
    4    1    30.100000E+010.000000E+000.000000E+00    0
    4    1    60.100000E+010.000000E+000.000000E+00    0
```

```
Y,F,Z,NR,NPLY,NV
      -1.000000          0.000000        -0.060000   1    1    6
       0.000000     -59187.849166        -0.060000   1    1    6
       1.000000     -96307.651585        -0.060000   1    1    6

Y,F,Z,NR,NPLY,NV
      -1.000000         -0.000001         0.000000   1    1    1
       0.000000         -0.000001         0.000000   1    1    1
       1.000000         -0.000001         0.000000   1    1    1

Y,F,Z,NR,NPLY,NV
      -1.000000          0.000000         0.000000   1    1    3
       0.000000        -60.000000         0.000000   1    1    3
       1.000000       -119.999999         0.000000   1    1    3

Y,F,Z,NR,NPLY,NV
      -1.000000        -60.000000         0.000000   1    1    5
       0.000000        -60.000000         0.000000   1    1    5
       1.000000        -60.000000         0.000000   1    1    5

Y,F,Z,NR,NPLY,NV
      -1.000000      -1499.499966         0.000000   2    1    1
      -0.500000      -1124.480712         0.000000   2    1    1
       0.000000          0.000000         0.000000   2    1    1

Y,F,Z,NR,NPLY,NV
      -1.000000        -14.983137         0.000000   2    1    3
      -0.500000        -10.969712         0.000000   2    1    3
       0.000000          0.000000         0.000000   2    1    3

Y,F,Z,NR,NPLY,NV
      -1.000000         30.003676         0.000000   2    1    5
      -0.500000         32.709472         0.000000   2    1    5
       0.000000          0.000000         0.000000   2    1    5

Y,F,Z,NR,NPLY,NV
      -1.000000          0.000001         0.000000   2    2    1
      -0.500000          0.000105         0.000000   2    2    1
       0.000000          0.000000         0.000000   2    2    1

Y,F,Z,NR,NPLY,NV
      -1.000000          0.000000         0.000000   2    2    3
      -0.500000          0.000000         0.000000   2    2    3
       0.000000          0.000000         0.000000   2    2    3

Y,F,Z,NR,NPLY,NV
      -1.000000          0.029990         0.000000   2    2    5
      -0.500000          0.030090         0.000000   2    2    5
       0.000000          0.000000         0.000000   2    2    5

Y,F,Z,NR,NPLY,NV
      -1.000000       1499.499963         0.000000   2    3    1
      -0.500000       1124.480605         0.000000   2    3    1
       0.000000          0.000000         0.000000   2    3    1
```

```
Y,F,Z,NR,NPLY,NV
      -1.000000        -14.986884        0.000000   2    3    3
      -0.500000        -11.516468        0.000000   2    3    3
       0.000000        -60.000000        0.000000   2    3    3

Y,F,Z,NR,NPLY,NV
      -1.000000         29.966334        0.000000   2    3    5
      -0.500000         27.260438        0.000000   2    3    5
       0.000000         60.000000        0.000000   2    3    5

Y,F,Z,NR,NPLY,NV
       0.000000          0.000000        0.000000   3    1    1
       0.500000          0.000000        0.000000   3    1    1
       1.000000          0.000000        0.000000   3    1    1

Y,F,Z,NR,NPLY,NV
       0.000000          0.000000        0.000000   3    1    3
       0.500000          0.000000        0.000000   3    1    3
       1.000000          0.000000        0.000000   3    1    3

Y,F,Z,NR,NPLY,NV
       0.000000          0.000000        0.000000   3    1    5
       0 500000          0.000000        0.000000   3    1    5
       1.000000          0.000000        0.000000   3    1    5

Y,F,Z,NR,NPLY,NV
       0.000000          0.000000        0.000000   4    1    1
       0.500000          0.000000        0.000000   4    1    1
       1.000000          0.000000        0.000000   4    1    1

Y,F,Z,NR,NPLY,NV
       0.000000         60.000000        0.000000   4    1    3
       0.500000         30.000000        0.000000   4    1    3
       1.000000          0.000000        0.000000   4    1    3

Y,F,Z,NR,NPLY,NV
       0.000000        -60.000000        0.000000   4    1    5
       0.500000        -60.000000        0.000000   4    1    5
       1.000000        -60.000000        0.000000   4    1    5

Y,F,Z,NR,NPLY,NV
       0.000000          8.404113        0.000000   2    2    7
      -0.006250     -17361.632794        0.000000   2    2    7
      -0.012500      -1585.236528        0.000000   2    2    7
      -0.018750      -2635.387718        0.000000   2    2    7
      -0.025000      -1871.055926        0.000000   2    2    7
      -0.031250      -1393.344180        0.000000   2    2    7
      -0.037500      -1026.510766        0.000000   2    2    7
      -0.043750       -746.099838        0.000000   2    2    7
      -0.050000       -532.849492        0.000000   2    2    7
      -0.056250       -371.070326        0.000000   2    2    7
      -0.062500       -248.805433        0.000000   2    2    7
      -0.068750       -156.841677        0.000000   2    2    7
```

-0.075000	-88.089417	0.000000	2	2	7
-0.081250	-37.094339	0.000000	2	2	7
-0.087500	0.338403	0.000000	2	2	7
-0.093750	27.433693	0.000000	2	2	7
-0.100000	46.669506	0.000000	2	2	7
-0.106250	59.948738	0.000000	2	2	7
-0.112500	68.731571	0.000000	2	2	7
-0.118750	74.137446	0.000000	2	2	7
-0.125000	77.023599	0.000000	2	2	7
-0.131250	78.045562	0.000000	2	2	7
-0.137500	77.703754	0.000000	2	2	7
-0.143750	76.379361	0.000000	2	2	7
-0.150000	74.361966	0.000000	2	2	7
-0.156250	71.870824	0.000000	2	2	7
-0.162500	69.071230	0.000000	2	2	7
-0.168750	66.087130	0.000000	2	2	7
-0.175000	63.010811	0.000000	2	2	7
-0.181250	59.910367	0.000000	2	2	7
-0.187500	56.835426	0.000000	2	2	7
-0.193750	53.821567	0.000000	2	2	7
-0.200000	50.893698	0.000000	2	2	7
-0.206250	48.068663	0.000000	2	2	7
-0.212500	45.357230	0.000000	2	2	7
-0.218750	42.765619	0.000000	2	2	7
-0.225000	40.296674	0.000000	2	2	7
-0.231250	37.950753	0.000000	2	2	7
-0.237500	35.726410	0.000000	2	2	7
-0.243750	33.620914	0.000000	2	2	7
-0.250000	31.630640	0.000000	2	2	7
-0.256250	29.751367	0.000000	2	2	7
-0.262500	27.978500	0.000000	2	2	7
-0.268750	26.307239	0.000000	2	2	7
-0.275000	24.732699	0.000000	2	2	7
-0.281250	23.250005	0.000000	2	2	7
-0.287500	21.854354	0.000000	2	2	7
-0.293750	20.541061	0.000000	2	2	7
-0.300000	19.305594	0.000000	2	2	7
-0.306250	18.143593	0.000000	2	2	7
-0.312500	17.050885	0.000000	2	2	7
-0.318750	16.023486	0.000000	2	2	7
-0.325000	15.057607	0.000000	2	2	7
-0.331250	14.149655	0.000000	2	2	7
-0.337500	13.296221	0.000000	2	2	7
-0.343750	12.494087	0.000000	2	2	7
-0.350000	11.740208	0.000000	2	2	7
-0.356250	11.031713	0.000000	2	2	7
-0.362500	10.365893	0.000000	2	2	7
-0.368750	9.740197	0.000000	2	2	7
-0.375000	9.152221	0.000000	2	2	7
-0.381250	8.599702	0.000000	2	2	7
-0.387500	8.080510	0.000000	2	2	7
-0.393750	7.592641	0.000000	2	2	7
-0.400000	7.134211	0.000000	2	2	7
-0.406250	6.703447	0.000000	2	2	7

-0.412500	6.298683	0.000000	2	2	7
-0.418750	5.918352	0.000000	2	2	7
-0.425000	5.560980	0.000000	2	2	7
-0.431250	5.225182	0.000000	2	2	7
-0.437500	4.909658	0.000000	2	2	7
-0.443750	4.613185	0.000000	2	2	7
-0.450000	4.334612	0.000000	2	2	7
-0.456250	4.072860	0.000000	2	2	7
-0.462500	3.826912	0.000000	2	2	7
-0.468750	3.595816	0.000000	2	2	7
-0.475000	3.378674	0.000000	2	2	7
-0.481250	3.174644	0.000000	2	2	7
-0.487500	2.982935	0.000000	2	2	7
-0.493750	2.802802	0.000000	2	2	7
-0.500000	2.633547	0.000000	2	2	7
-0.506250	2.474512	0.000000	2	2	7
-0.512500	2.325081	0.000000	2	2	7
-0.518750	2.184674	0.000000	2	2	7
-0.525000	2.052746	0.000000	2	2	7
-0.531250	1.928784	0.000000	2	2	7
-0.537500	1.812309	0.000000	2	2	7
-0.543750	1.702867	0.000000	2	2	7
-0.550000	1.600034	0.000000	2	2	7
-0.556250	1.503410	0.000000	2	2	7
-0.562500	1.412622	0.000000	2	2	7
-0.568750	1.327316	0.000000	2	2	7
-0.575000	1.247162	0.000000	2	2	7
-0.581250	1.171848	0.000000	2	2	7
-0.587500	1.101082	0.000000	2	2	7
-0.593750	1.034590	0.000000	2	2	7
-0.600000	0.972113	0.000000	2	2	7
-0.606250	0.913408	0.000000	2	2	7
-0.612500	0.858249	0.000000	2	2	7
-0.618750	0.806421	0.000000	2	2	7
-0.625000	0.757723	0.000000	2	2	7
-0.631250	0.711965	0.000000	2	2	7
-0.637500	0.668971	0.000000	2	2	7
-0.643750	0.628573	0.000000	2	2	7
-0.650000	0.590614	0.000000	2	2	7
-0.656250	0.554948	0.000000	2	2	7
-0.662500	0.521436	0.000000	2	2	7
-0.668750	0.489947	0.000000	2	2	7
-0.675000	0.460360	0.000000	2	2	7
-0.681250	0.432560	0.000000	2	2	7
-0.687500	0.406438	0.000000	2	2	7
-0.693750	0.381894	0.000000	2	2	7
-0.700000	0.358832	0.000000	2	2	7
-0.706250	0.337163	0.000000	2	2	7
-0.712500	0.316802	0.000000	2	2	7
-0.718750	0.297671	0.000000	2	2	7
-0.725000	0.279695	0.000000	2	2	7
-0.731250	0.262805	0.000000	2	2	7
-0.737500	0.246934	0.000000	2	2	7
-0.743750	0.232022	0.000000	2	2	7

-0.750000	0.218011	0.000000	2	2	7
-0.756250	0.204846	0.000000	2	2	7
-0.762500	0.192475	0.000000	2	2	7
-0.768750	0.180852	0.000000	2	2	7
-0.775000	0.169931	0.000000	2	2	7
-0.781250	0.159669	0.000000	2	2	7
-0.787500	0.150027	0.000000	2	2	7
-0.793750	0.140967	0.000000	2	2	7
-0.800000	0.132454	0.000000	2	2	7
-0.806250	0.124455	0.000000	2	2	7
-0.812500	0.116940	0.000000	2	2	7
-0.818750	0.109878	0.000000	2	2	7
-0.825000	0.103242	0.000000	2	2	7
-0.831250	0.097008	0.000000	2	2	7
-0.837500	0.091150	0.000000	2	2	7
-0.843750	0.085645	0.000000	2	2	7
-0.850000	0.080473	0.000000	2	2	7
-0.856250	0.075614	0.000000	2	2	7
-0.862500	0.071047	0.000000	2	2	7
-0.868750	0.066757	0.000000	2	2	7
-0.875000	0.062726	0.000000	2	2	7
-0.881250	0.058938	0.000000	2	2	7
-0.887500	0.055379	0.000000	2	2	7
-0.893750	0.052034	0.000000	2	2	7
-0.900000	0.048892	0.000000	2	2	7
-0.906250	0.045939	0.000000	2	2	7
-0.912500	0.043165	0.000000	2	2	7
-0.918750	0.040559	0.000000	2	2	7
-0.925000	0.038109	0.000000	2	2	7
-0.931250	0.035808	0.000000	2	2	7
-0.937500	0.033646	0.000000	2	2	7
-0.943750	0.031614	0.000000	2	2	7
-0.950000	0.029705	0.000000	2	2	7
-0.956250	0.027911	0.000000	2	2	7
-0.962500	0.026225	0.000000	2	2	7
-0.968750	0.024642	0.000000	2	2	7
-0.975000	0.023153	0.000000	2	2	7
-0.981250	0.021755	0.000000	2	2	7
-0.987500	0.020441	0.000000	2	2	7
-0.993750	0.019207	0.000000	2	2	7
-1.000000	0.018047	0.000000	2	2	7

Y,F,Z,NR,NPLY,NV

0.000000	9368.650250	0.000000	2	2	8
-0.006250	9815.065479	0.000000	2	2	8
-0.012500	9793.669874	0.000000	2	2	8
-0.018750	9480.254145	0.000000	2	2	8
-0.025000	8999.911505	0.000000	2	2	8
-0.031250	8434.153598	0.000000	2	2	8
-0.037500	7835.318175	0.000000	2	2	8
-0.043750	7236.287692	0.000000	2	2	8
-0.050000	6657.037491	0.000000	2	2	8
-0.056250	6109.041135	0.000000	2	2	8
-0.062500	5598.228094	0.000000	2	2	8

-0.068750	5126.964053	0.000000	2	2	8
-0.075000	4695.371897	0.000000	2	[illegible]	8
-0.081250	4302.208422	0.000000	2	2	8
-0.087500	3945.442122	0.000000	2	2	8
-0.093750	3622.630253	0.000000	2	2	8
-0.100000	3331.161479	0.000000	2	2	8
-0.106250	3068.408824	0.000000	2	2	8
-0.112500	2831.823081	0.000000	2	2	8
-0.118750	2618.986945	0.000000	2	2	8
-0.125000	2427.643517	0.000000	2	2	8
-0.131250	2255.708276	0.000000	2	2	8
-0.137500	2101.270629	0.000000	2	2	8
-0.143750	1962.589070	0.000000	2	2	8
-0.150000	1838.082628	0.000000	2	2	8
-0.156250	1726.320339	0.000000	2	2	8
-0.162500	1626.009867	0.000000	2	2	8
-0.168750	1535.985994	0.000000	2	2	8
-0.175000	1455.199404	0.000000	2	2	8
-0.181250	1382.706020	0.000000	2	2	8
-0.187500	1317.657039	0.000000	2	2	8
-0.193750	1259.289709	0.000000	2	2	8
-0.200000	1206.918873	0.000000	2	2	8
-0.206250	1159.929255	0.000000	2	2	8
-0.212500	1117.768458	0.000000	2	2	8
-0.218750	1079.940614	0.000000	2	2	8
-0.225000	1046.000662	0.000000	2	2	8
-0.231250	1015.549175	0.000000	2	2	8
-0.237500	988.227708	0.000000	2	2	8
-0.243750	963.714612	0.000000	2	2	8
-0.250000	941.721265	0.000000	2	2	8
-0.256250	921.988692	0.000000	2	2	8
-0.262500	904.284524	0.000000	2	2	8
-0.268750	888.400267	0.000000	2	2	8
-0.275000	874.148854	0.000000	2	2	8
-0.281250	861.362440	0.000000	2	2	8
-0.287500	849.890434	0.000000	2	2	8
-0.293750	839.597722	0.000000	2	2	8
-0.300000	830.363077	0.000000	2	2	8
-0.306250	822.077734	0.000000	2	2	8
-0.312500	814.644107	0.000000	2	2	8
-0.318750	807.974644	0.000000	2	2	8
-0.325000	801.990789	0.000000	2	2	8
-0.331250	796.622064	0.000000	2	2	8
-0.337500	791.805234	0.000000	2	2	8
-0.343750	787.483567	0.000000	2	2	8
-0.350000	783.606159	0.000000	2	2	8
-0.356250	780.127343	0.000000	2	2	8
-0.362500	777.006143	0.000000	2	2	8
-0.368750	774.205798	0.000000	2	2	8
-0.375000	771.693323	0.000000	2	2	8
-0.381250	769.439127	0.000000	2	2	8
-0.387500	767.416658	0.000000	2	2	8
-0.393750	765.602096	0.000000	2	2	8
-0.400000	763.974068	0.000000	2	2	8

-0.406250	762.513399	0.000000	2	2	8
-0.412500	761.202884	0.000000	2	2	8
-0.418750	760.027088	0.000000	2	2	8
-0.425000	758.972162	0.000000	2	2	8
-0.431250	758.025680	0.000000	2	2	8
-0.437500	757.176496	0.000000	2	2	8
-0.443750	756.414606	0.000000	2	2	8
-0.450000	755.731037	0.000000	2	2	8
-0.456250	755.117738	0.000000	2	2	8
-0.462500	754.567485	0.000000	2	2	8
-0.468750	754.073798	0.000000	2	2	8
-0.475000	753.630860	0.000000	2	2	8
-0.481250	753.233456	0.000000	2	2	8
-0.487500	752.876904	0.000000	2	2	8
-0.493750	752.557006	0.000000	2	2	8
-0.500000	752.269992	0.000000	2	2	8
-0.506250	752.012483	0.000000	2	2	8
-0.512500	751.781445	0.000000	2	2	8
-0.518750	751.574158	0.000000	2	2	8
-0.525000	751.388180	0.000000	2	2	8
-0.531250	751.221319	0.000000	2	2	8
-0.537500	751.071612	0.000000	2	2	8
-0.543750	750.937295	0.000000	2	2	8
-0.550000	750.816785	0.000000	2	2	8
-0.556250	750.708663	0.000000	2	2	8
-0.562500	750.611656	0.000000	2	2	8
-0.568750	750.524621	0.000000	2	2	8
-0.575000	750.446533	0.000000	2	2	8
-0.581250	750.376473	0.000000	2	2	8
-0.587500	750.313615	0.000000	2	2	8
-0.593750	750.257218	0.000000	2	2	8
-0.600000	750.206619	0.000000	2	2	8
-0.606250	750.161221	0.000000	2	2	8
-0.612500	750.120490	0.000000	2	2	8
-0.618750	750.083947	0.000000	2	2	8
-0.625000	750.051159	0.000000	2	2	8
-0.631250	750.021743	0.000000	2	2	8
-0.637500	749.995350	0.000000	2	2	8
-0.643750	749.971671	0.000000	2	2	8
-0.650000	749.950425	0.000000	2	2	8
-0.656250	749.931364	0.000000	2	2	8
-0.662500	749.914262	0.000000	2	2	8
-0.668750	749.898918	0.000000	2	2	8
-0.675000	749.885152	0.000000	2	2	8
-0.681250	749.872800	0.000000	2	2	8
-0.687500	749.861719	0.000000	2	2	8
-0.693750	749.851776	0.000000	2	2	8
-0.700000	749.842856	0.000000	2	2	8
-0.706250	749.834852	0.000000	2	2	8
-0.712500	749.827672	0.000000	2	2	8
-0.718750	749.821229	0.000000	2	2	8
-0.725000	749.815449	0.000000	2	2	8
-0.731250	749.810263	0.000000	2	2	8
-0.737500	749.805610	0.000000	2	2	8

-0.743750	749.801436	0.000000	2	2	8
-0.750000	749.797690	0.000000	2	2	8
-0.756250	749.794330	0.000000	2	2	8
-0.762500	749.791315	0.000000	2	2	8
-0.768750	749.788610	0.000000	2	2	8
-0.775000	749.786183	0.000000	2	2	8
-0.781250	749.784005	0.000000	2	2	8
-0.787500	749.782052	0.000000	2	2	8
-0.793750	749.780299	0.000000	2	2	8
-0.800000	749.778726	0.000000	2	2	8
-0.806250	749.777315	0.000000	2	2	8
-0.812500	749.776049	0.000000	2	2	8
-0.818750	749.774913	0.000000	2	2	8
-0.825000	749.773894	0.000000	2	2	8
-0.831250	749.772980	0.000000	2	2	8
-0.837500	749.772160	0.000000	2	2	8
-0.843750	749.771424	0.000000	2	2	8
-0.850000	749.770764	0.000000	2	2	8
-0.856250	749.770171	0.000000	2	2	8
-0.862500	749.769640	0.000000	2	2	8
-0.868750	749.769163	0.000000	2	2	8
-0.875000	749.768735	0.000000	2	2	8
-0.881250	749.768351	0.000000	2	2	8
-0.887500	749.768007	0.000000	2	2	8
-0.893750	749.767698	0.000000	2	2	8
-0.900000	749.767420	0.000000	2	2	8
-0.906250	749.767172	0.000000	2	2	8
-0.912500	749.766948	0.000000	2	2	8
-0.918750	749.766748	0.000000	2	2	8
-0.925000	749.766569	0.000000	2	2	8
-0.931250	749.766407	0.000000	2	2	8
-0.937500	749.766263	0.000000	2	2	8
-0.943750	749.766133	0.000000	2	2	8
-0.950000	749.766017	0.000000	2	2	8
-0.956250	749.765912	0.000000	2	2	8
-0.962500	749.765818	0.000000	2	2	8
-0.968750	749.765734	0.000000	2	2	8
-0.975000	749.765659	0.000000	2	2	8
-0.981250	749.765591	0.000000	2	2	8
-0.987500	749.765531	0.000000	2	2	8
-0.993750	749.765476	0.000000	2	2	8
-1.000000	749.765427	0.000000	2	2	8

Y,F,Z,NR,NPLY,NV

0.000000	-24253.389799	0.000000	2	2	9
-0.006250	-8819.215527	0.000000	2	2	9
-0.012500	-6727.635173	0.000000	2	2	9
-0.018750	-4590.680653	0.000000	2	2	9
-0.025000	-2976.740149	0.000000	2	2	9
-0.031250	-1771.874917	0.000000	2	2	9
-0.037500	-874.399832	0.000000	2	2	9
-0.043750	-212.588805	0.000000	2	2	9
-0.050000	269.486656	0.000000	2	2	9
-0.056250	614.757490	0.000000	2	2	9

-0.062500	856.201076	0.000000	2	2	9
-0.068750	1019.131162	0.000000	2	2	9
-0.075000	1122.962117	0.000000	2	2	9
-0.081250	1182.568348	0.000000	2	2	9
-0.087500	1209.331329	0.000000	2	2	9
-0.093750	1211.946039	0.000000	2	2	9
-0.100000	1197.042016	0.000000	2	2	9
-0.106250	1169.661612	0.000000	2	2	9
-0.112500	1133.628231	0.000000	2	2	9
-0.118750	1091.829804	0.000000	2	2	9
-0.125000	1046.436994	0.000000	2	2	9
-0.131250	999.071100	0.000000	2	2	9
-0.137500	950.933232	0.000000	2	2	9
-0.143750	902.903648	0.000000	2	2	9
-0.150000	855.618126	0.000000	2	2	9
-0.156250	809.526632	0.000000	2	2	9
-0.162500	764.938377	0.000000	2	2	9
-0.168750	722.056386	0.000000	2	2	9
-0.175000	681.003994	0.000000	2	2	9
-0.181250	641.845131	0.000000	2	2	9
-0.187500	604.599841	0.000000	2	2	9
-0.193750	569.256110	0.000000	2	2	9
-0.200000	535.778888	0.000000	2	2	9
-0.206250	504.116930	0.000000	2	2	9
-0.212500	474.207978	0.000000	2	2	9
-0.218750	445.982665	0.000000	2	2	9
-0.225000	419.367435	0.000000	2	2	9
-0.231250	394.286722	0.000000	2	2	9
-0.237500	370.664553	0.000000	2	2	9
-0.243750	348.425708	0.000000	2	2	9
-0.250000	327.496561	0.000000	2	2	9
-0.256250	307.805658	0.000000	2	2	9
-0.262500	289.284104	0.000000	2	2	9
-0.268750	271.865818	0.000000	2	2	9
-0.275000	255.487668	0.000000	2	2	9
-0.281250	240.089533	0.000000	2	2	9
-0.287500	225.614304	0.000000	2	2	9
-0.293750	212.007846	0.000000	2	2	9
-0.300000	199.218928	0.000000	2	2	9
-0.306250	187.199126	0.000000	2	2	9
-0.312500	175.902726	0.000000	2	2	9
-0.318750	165.286607	0.000000	2	2	9
-0.325000	155.310120	0.000000	2	2	9
-0.331250	145.934975	0.000000	2	2	9
-0.337500	137.125114	0.000000	2	2	9
-0.343750	128.846599	0.000000	2	2	9
-0.350000	121.067496	0.000000	2	2	9
-0.356250	113.757765	0.000000	2	2	9
-0.362500	106.889151	0.000000	2	2	9
-0.368750	100.435086	0.000000	2	2	9
-0.375000	94.370590	0.000000	2	2	9
-0.381250	88.672180	0.000000	2	2	9
-0.387500	83.317779	0.000000	2	2	9
-0.393750	78.286639	0.000000	2	2	9

-0.400000	73.559256	0.000000	2	2	9
-0.406250	69.117304	0.000000	2	2	9
-0.412500	64.943555	0.000000	2	2	9
-0.418750	61.021823	0.000000	2	2	9
-0.425000	57.336895	0.000000	2	2	9
-0.431250	53.874476	0.000000	2	2	9
-0.437500	50.621133	0.000000	2	2	9
-0.443750	47.564244	0.000000	2	2	9
-0.450000	44.691947	0.000000	2	2	9
-0.456250	41.993097	0.000000	2	2	9
-0.462500	39.457222	0.000000	2	2	9
-0.468750	37.074480	0.000000	2	2	9
-0.475000	34.835625	0.000000	2	2	9
-0.481250	32.731968	0.000000	2	2	9
-0.487500	30.755346	0.000000	2	2	9
-0.493750	28.898088	0.000000	2	2	9
-0.500000	27.152985	0.000000	2	2	9
-0.506250	25.513265	0.000000	2	2	9
-0.512500	23.972565	0.000000	2	2	9
-0.518750	22.524904	0.000000	2	2	9
-0.525000	21.164665	0.000000	2	2	9
-0.531250	19.886568	0.000000	2	2	9
-0.537500	18.685653	0.000000	2	2	9
-0.543750	17.557259	0.000000	2	2	9
-0.550000	16.497006	0.000000	2	2	9
-0.556250	15.500780	0.000000	2	2	9
-0.562500	14.564715	0.000000	2	2	9
-0.568750	13.685177	0.000000	2	2	9
-0.575000	12.858753	0.000000	2	2	9
-0.581250	12.082235	0.000000	2	2	9
-0.587500	11.352609	0.000000	2	2	9
-0.593750	10.667045	0.000000	2	2	9
-0.600000	10.022880	0.000000	2	2	9
-0.606250	9.417616	0.000000	2	2	9
-0.612500	8.848902	0.000000	2	2	9
-0.618750	8.314532	0.000000	2	2	9
-0.625000	7.812432	0.000000	2	2	9
-0.631250	7.340652	0.000000	2	2	9
-0.637500	6.897363	0.000000	2	2	9
-0.643750	6.480843	0.000000	2	2	9
-0.650000	6.089476	0.000000	2	2	9
-0.656250	5.721743	0.000000	2	2	9
-0.662500	5.376217	0.000000	2	2	9
-0.668750	5.051556	0.000000	2	2	9
-0.675000	4.746501	0.000000	2	2	9
-0.681250	4.459868	0.000000	2	2	9
-0.687500	4.190544	0.000000	2	2	9
-0.693750	3.937485	0.000000	2	2	9
-0.700000	3.699707	0.000000	2	2	9
-0.706250	3.476288	0.000000	2	2	9
-0.712500	3.266361	0.000000	2	2	9
-0.718750	3.069111	0.000000	2	2	9
-0.725000	2.883772	0.000000	2	2	9
-0.731250	2.709626	0.000000	2	2	9

-0.737500	2.545997	0.000000	2	2	9
-0.743750	2.392248	0.000000	2	2	9
-0.750000	2.247784	0.000000	2	2	9
-0.756250	2.112045	0.000000	2	2	9
-0.762500	1.984502	0.000000	2	2	9
-0.768750	1.864661	0.000000	2	2	9
-0.775000	1.752057	0.000000	2	2	9
-0.781250	1.646254	0.000000	2	2	9
-0.787500	1.546839	0.000000	2	2	9
-0.793750	1.453428	0.000000	2	2	9
-0.800000	1.365658	0.000000	2	2	9
-0.806250	1.283188	0.000000	2	2	9
-0.812500	1.205699	0.000000	2	2	9
-0.818750	1.132889	0.000000	2	2	9
-0.825000	1.064476	0.000000	2	2	9
-0.831250	1.000194	0.000000	2	2	9
-0.837500	0.939794	0.000000	2	2	9
-0.843750	0.883041	0.000000	2	2	9
-0.850000	0.829716	0.000000	2	2	9
-0.856250	0.779611	0.000000	2	2	9
-0.862500	0.732531	0.000000	2	2	9
-0.868750	0.688295	0.000000	2	2	9
-0.875000	0.646730	0.000000	2	2	9
-0.881250	0.607675	0.000000	2	2	9
-0.887500	0.570979	0.000000	2	2	9
-0.893750	0.536498	0.000000	2	2	9
-0.900000	0.504100	0.000000	2	2	9
-0.906250	0.473658	0.000000	2	2	9
-0.912500	0.445055	0.000000	2	2	9
-0.918750	0.418179	0.000000	2	2	9
-0.925000	0.392926	0.000000	2	2	9
-0.931250	0.369198	0.000000	2	2	9
-0.937500	0.346902	0.000000	2	2	9
-0.943750	0.325954	0.000000	2	2	9
-0.950000	0.306270	0.000000	2	2	9
-0.956250	0.287775	0.000000	2	2	9
-0.962500	0.270396	0.000000	2	2	9
-0.968750	0.254068	0.000000	2	2	9
-0.975000	0.238725	0.000000	2	2	9
-0.981250	0.224309	0.000000	2	2	9
-0.987500	0.210763	0.000000	2	2	9
-0.993750	0.198036	0.000000	2	2	9
-1.000000	0.186076	0.000000	2	2	9

REFERENCES

1. Murthy, P.L.N., and Chamis, C.C., "Interlaminar Fracture Toughness: Three-Dimensional Finite-Element Modeling for End-Notch and Mixed-Mode Flexure," NASA TM-87138, National Aeronautics and Space Administration, Washington, D.C., 1985.

2. Pagano, N.J., International Journal of Solids and Structures, Vol. 14, No. 5, 1978, pp. 385-400.

3. Raju, I.S., Whitcomb, J.D., and Goree, J.G., "A New Look at the Numerical Analyses of Free-Edge Stresses in Composite Laminates," NASA TP-1751, National Aeronautics and Space Administration, Washington, D.C., 1980.

4. Pagano, N.J. and Soni, S.R., International Journal of Solids and Structures, Vol. 19, No. 3, 1983, pp. 207-228.

5. Salamon, N.J., Journal of Composite Materials Supplement, Vol. 14, No. 1, 1980, pp. 177-194.

6. Valisetty, R.R. and Rehfield, L.W., in Delamination and Debonding of Materials, ASTM STP-876, W.S. Johnson, ED., American Society for Testing and Materials, Philadelphia, 1985, pp. 52-68.

7. Rehfield, L.W., Erian, A.A., and Valisetty, R.R., Computers and Structures, Vol. 20, No. 1-3, 1985, pp. 401-411.

8. Valisetty, R.R., "Bending of Beams, Plates, and Laminates: Refined Theories and Comparative Studies," Ph.D. Theis, Georgia Institute of Technology, Atlanta, GA., 1983.

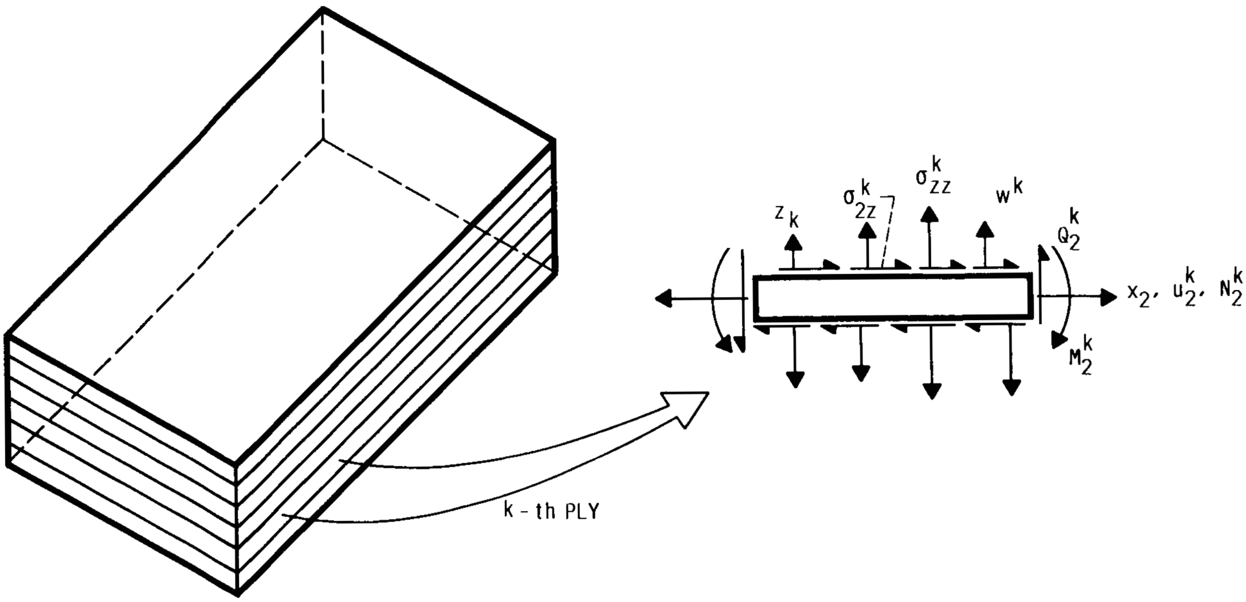

FIGURE 1. - EQUILIBRIUM OF A PLY UNDER INTERLAMINAR STRESSES AND EDGE FORCES.

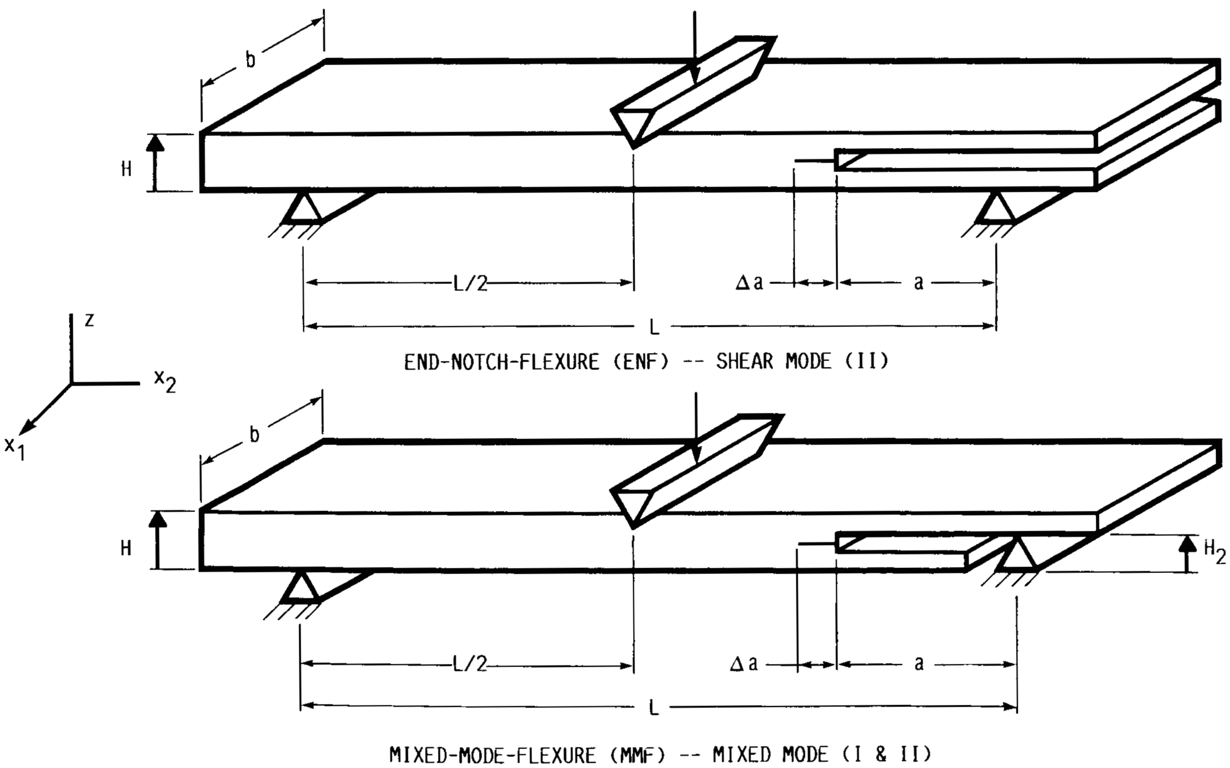

FIGURE 2. - SCHEMATIC OF FLEXURAL TEST FOR INTERLAMINAR FRACTURE MODE TOUGHNESS.

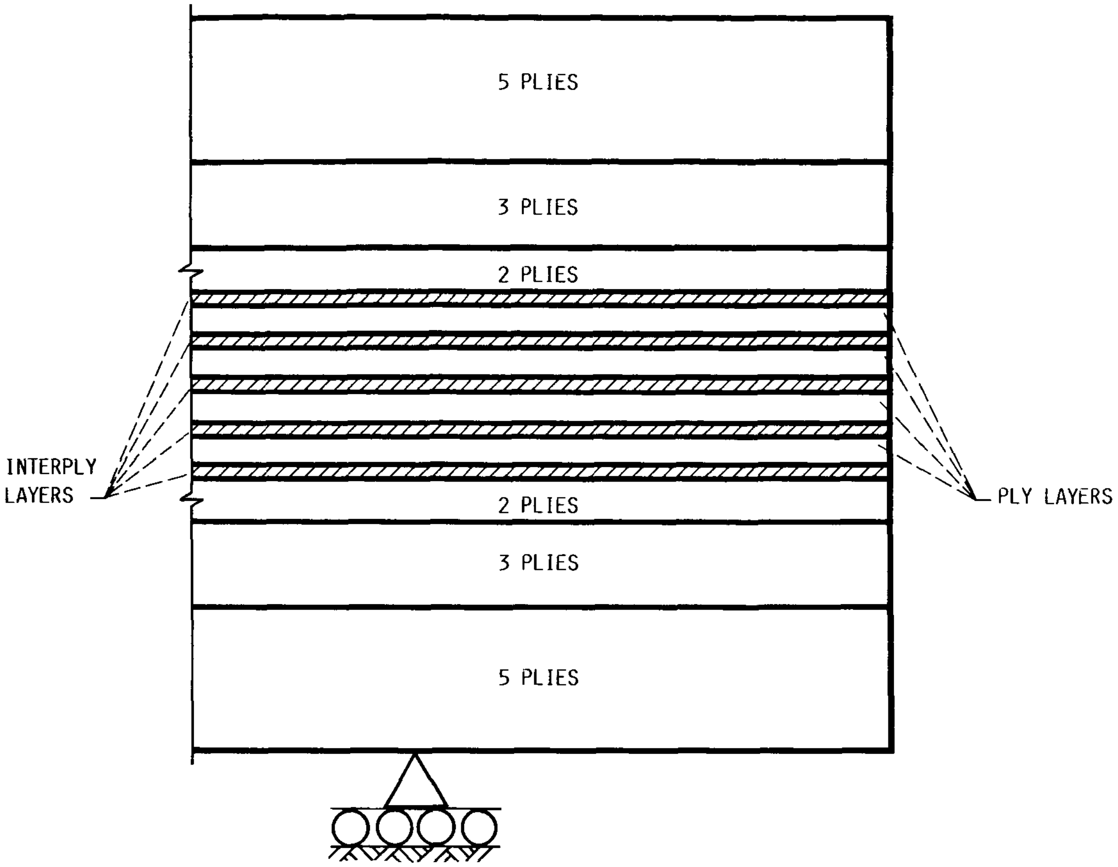

FIGURE 3. - PLY STACKING SEQUENCE.

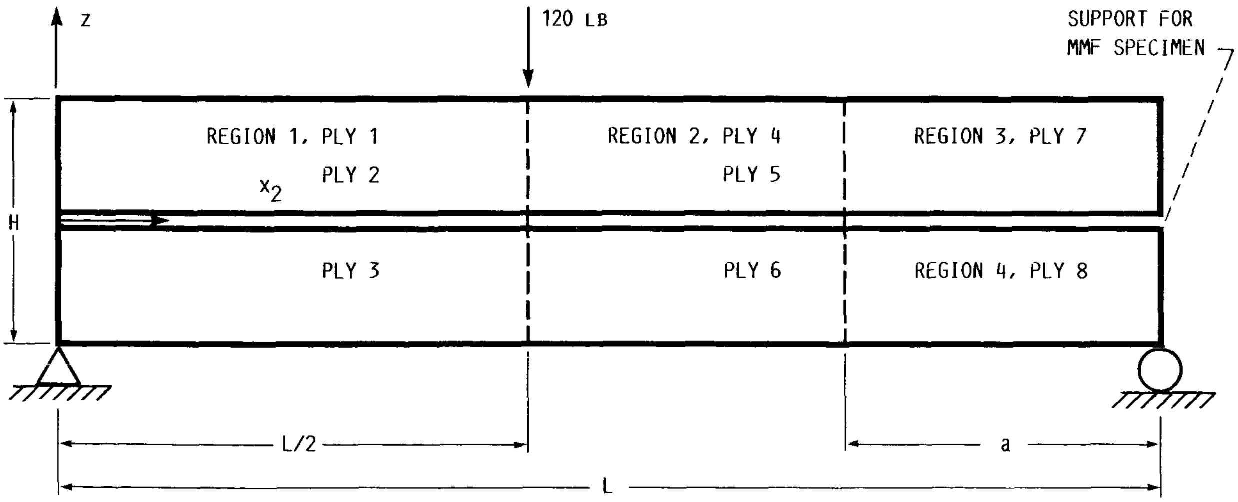

FIGURE 4. - IDEALIZATION OF THE ENF SPECIMEN FOR SUBLAMINATE ANALYSIS.

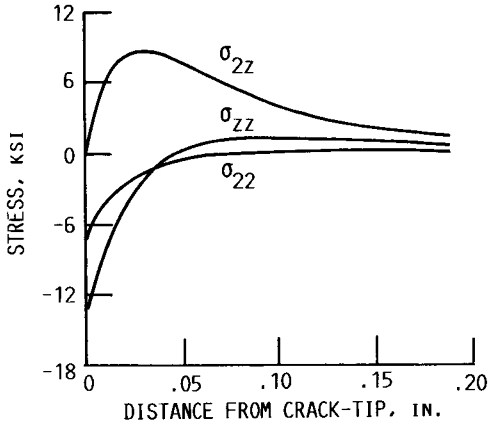

FIGURE 5. - STRESS DISTRIBUTION IN THE INTERPLY LAYER AHEAD OF THE CRACK-TIP FOR THE ENF SPECIMEN, CRACK LENGTH, a = 1 IN.

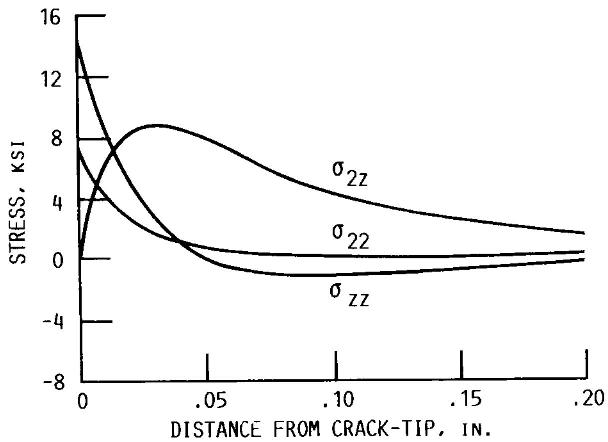

FIGURE 6. - STRESS DISTRIBUTION IN THE INTERPLY LAYER AHEAD OF THE CRACK-TIP FOR THE MMF SPECIMEN, CRACK LENGTH, a = 1 IN.

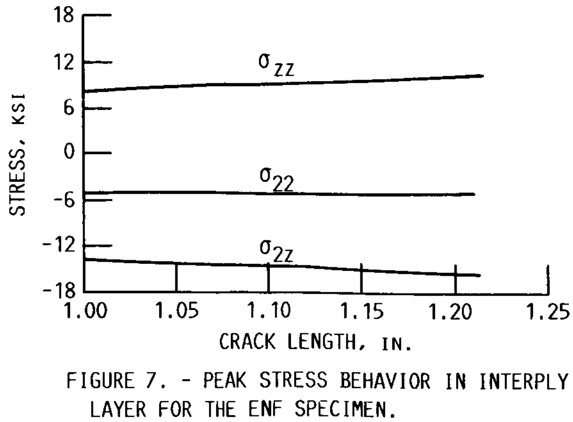

FIGURE 7. - PEAK STRESS BEHAVIOR IN INTERPLY LAYER FOR THE ENF SPECIMEN.

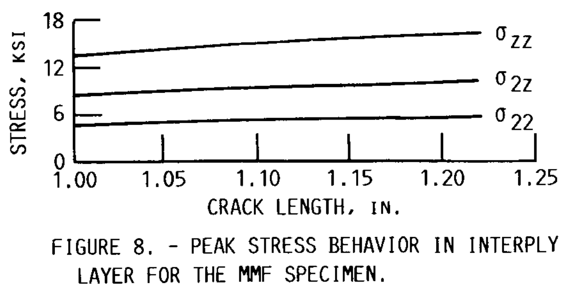

FIGURE 8. - PEAK STRESS BEHAVIOR IN INTERPLY LAYER FOR THE MMF SPECIMEN.

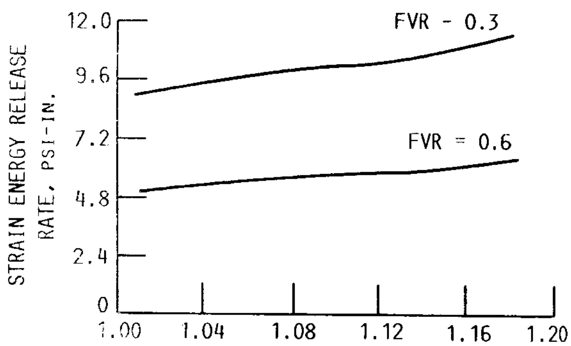

FIGURE 9. - TOTAL STRAIN ENERGY RELEASE RATES FOR ENF AND MMF SPECIMENS.

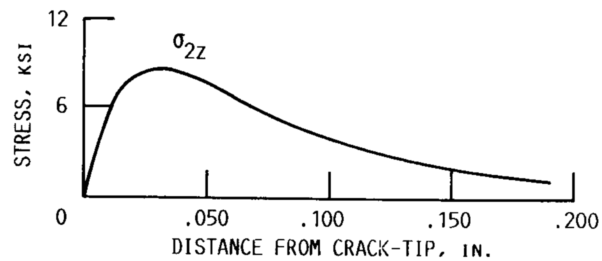

FIGURE 10. - STRESS FIELD IN INTERPLY LAYER AHEAD OF CRACK-TIP FOR ENF-II IDEALIZATION.

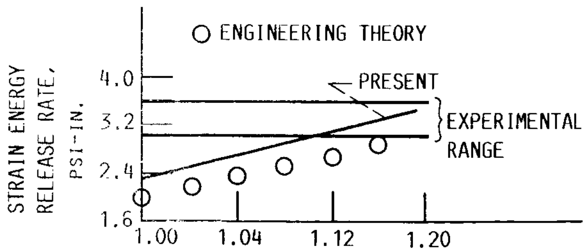

FIGURE 11. - TOTAL STRAIN ENERGY RELEASE RATE FOR ENF-11 IDEALIZATION.

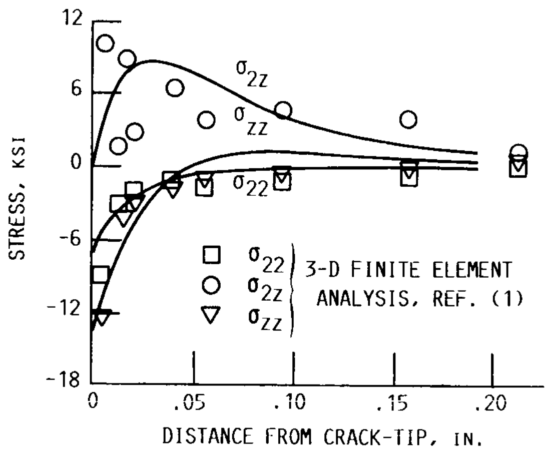

FIGURE 12. - STRESS DISTRIBUTION IN THE INTERPLY LAYER AHEAD OF THE CRACK-TIP FOR THE ENF SPECIMEN, CRACK LENGTH, a= 1 IN.

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TM-89827		

4. Title and Subtitle	5. Report Date
Strain Energy Release Rates of Composite Interlaminar End-Notch and Mixed-Mode Fracture: A Sublaminate/Ply Level Analysis and a Computer Code	April 1987
	6. Performing Organization Code 505-63-11
7. Author(s) R.R. Valisetty and C.C. Chamis	8. Performing Organization Report No. E-3476
	10. Work Unit No.
9. Performing Organization Name and Address National Aeronautics and Space Administration Lewis Research Center Cleveland, Ohio 44135	11. Contract or Grant No.
	13. Type of Report and Period Covered Technical Memorandum
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546	14. Sponsoring Agency Code

15. Supplementary Notes

Prepared for the 8th Symposium on Composite Materials Testing and Design, sponsored by the American Society for Testing and Materials, Charleston, South Carolina, April 29 - May 1, 1986. R.R. Valisetty, National Research Council - NASA Research Associate; C.C. Chamis, NASA Lewis Research Center.

16. Abstract

A computer code is presented for the sublaminate/ply level analysis of composite structures. This code is useful for obtaining stresses in regions affected by delaminations, transverse cracks, and discontinuities related to inherent fabrication anomalies, geometric configurations, and loading conditions. Particular attention is focussed on those layers or groups of layers (sublaminates) which are immediately affected by the inherent flaws. These layers are analyzed as homogeneous bodies in equilibrium and in isolation from the rest of the laminate. The theoretical model used to analyze the individual layers allows the relevant stresses and displacements near discontinuities to be represented in the form of pure exponential-decay-type functions which are selected to eliminate the exponential-precision-related difficulties in sublaminate/ply level analysis. Thus, sublaminate analysis can be conducted without any restriction on the maximum number of layers, delaminations, transverse cracks, or other types of discontinuities. In conjunction with the strain energy release rate (SERR) concept and composite micromechanics, this computational procedure is used to model select cases of end-notch and mixed-mode fracture specimens. The computed stresses are in good agreement with those from a three-dimensional finite element analysis. Also SERRs compare well with limited available experimental data.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Sublaminate analysis; Ply level analysis; Composite laminate analysis; Strain energy release rate; Interlaminar delamination; Mode I fracture; Mode II fracture; Interply layer; Fracture toughness; Graphite fiber; Epoxy matrix; Crack closure; Interlaminar fracture; Interply layer; Mixed mode fracture; Graphite/epoxy laminates	Unclassified - unlimited STAR Category 24

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price*
Unclassified	Unclassified	86	A05

*For sale by the National Technical Information Service, Springfield, Virginia 22161