Delamination Dynamics and Vibrothermographic-Thermoelastic Evaluation of Advanced Composite Materials

by

Lazarus H. Tenek

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

in

Engineering Mechanics

APPROVED:

Edmund G. Henneke, II

Dr. E.G.Henneke II, Chairman

Dr. M.D. Gunzburger

David A. Dillard

Dr. D.A. Dillard

January, 1991
Blacksburg, Virginia

ACKNOWLEDGEMENTS

The author wishes to thank Dr. Edmund G. Henneke, II for the support, assistance, advice and encouragement he received during his years at Virginia Tech.

Also appreciation and sincere thanks to:

- Dr. Max Gunzburger, for his advice, for serving on his committee and for introducing the author to the world of numerical analysis.
- Dr. David Dillard, for serving on his committee and for many valuable discussions.
- National Science Foundation Center for High Performance Polymeric Adhesives and Composites at Virginia Tech for providing support for this work.

DEDICATION

This work is dedicated to my parents Harry and Alexandra and to my best friend Cathy.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

1. INTRODUCTION

In the past twenty years composite materials have been widely used in advanced engineering structures. Because of their complex nature, refined analytical and experimental methods of analysis must be employed to study their mechanical behavior. Moreover, a number of mechanisms contributing to such a behavior must be explored and understood, both at the macro and micro structural levels. On the other hand, damage often takes a complex form and appears as matrix cracks, fiber breakage, delaminations, etc., making detection and evaluation a difficult effort. As a result, advanced nondestructive testing techniques (NDE) must be applied in order to gain information about damage presence, location, severity and type. Usually the application of one method is not sufficient to provide all the necessary information, and quite often a variety of NDE methods are combined to yield complementary data. The application of these methods depend on different factors such as the type of structure considered, equipment availability, environmental conditions and technical expertise. Often the interpretation of the results obtained is difficult.

One of the most common types of damage in anisotropic composite plates is delamination, where an in-plane crack forms as a result of the separation of adjacent layers. Impact loading is one common type of loading resulting in delaminations. At the present time it is not known exactly where the crack will first initiate. Excessive out-of-plane shearing and normal stresses may significantly contribute to the

initiation of the delamination. If after the defect appearance the structural component continues to operate under static, dynamic and environmental loads for some period of time, the crack will propagate and progressive failure will follow. It is thus essential that we detect, evaluate and control the damaged area before it will expand and contribute to structural failure.

1.1 Frequency Dependent Heat Generation During Vibrothermographic Testing

Thermography is the technique used to visualize surface temperature of an object. The thermal image is generally presented as isothermal lines on the surface. Thermal field generation can be accomplished in two ways, termed 'passive' and 'active' heating, respectively. In passive heating, the specimen is heated in-process by different transformation processes and dissipative mechanisms, while the test object is being subjected to a normal operating , testing or loading condition. In active heating, heat is produced by injection, and thermal gradients on the material surface are produced due to different thermal conductivities in the defect regions.

Vibrothermography is a term describing a nondestructive technique, where a portion of a structure's surface is mapped to indicate the temperatures on the surface - produced by passive heating - while the structure is subjected to forced mechanical oscillations. Areas of defects convert mechanical energy to heat through different dissipative mechanisms. As a result flaws and other imperfections appear as local

(usually) hot regions when the surface is mapped to indicate the temperature. The thermal image is detected by an infrared detector. The biggest advantage of this method compared with other NDE methods is that it is extremely fast in revealing damaged areas, can inspect large areas of the structure in a small period of time, can be performed 'in situ' under adverse environmental conditions and does not require advanced specialized knowledge.

As an NDE method for composite plates, vibrothermography was first initiated at Virginia Tech and it's operating principles were first discussed by Reifsnider, Henneke and Stinchcomb[1]. Henneke and Russell[2] used several NDE techniques to detect impact damage in glass-epoxy panels. Using vibrothermography they observed that heat generation patterns appeared around the defect areas. These patterns were strongly dependent on the excitation frequency. They also proposed two models to explain the frequency dependent heat generation, namely a 'local resonance model' (LRM) and a 'global resonance model' (GRM). In the local resonance model the delamination was modeled as two plates, one on either side of the delamination, free to resonate with their own dynamics. The local plates were assumed ideally clamped to the rest of the plate. Based on the Raleigh-Ritz approximation, they developed a model for local resonant frequency prediction. This model was primarily written for 0-degree unidirectional symmetric laminates. Secondly, a structural resonance model was proposed, were the natural frequencies were calculated for the whole plate. It was assumed that heating developed when global resonance occurred.

Henneke and Lin[3] expanded the above formulation and a new software

program was written to predict the natural frequencies of a symmetric, multi-angle ply anisotropic laminate with clamped boundary conditions. Experiments were performed to check the program predictions with experimental data. A finite difference computer program was also developed to simulate heat contours during vibration resonance. Heat generation was assumed proportional to the strain field.

Although some correlation between theory and experiment was achieved, not all the results were satisfactory. For example Russell[2] found that the local resonance model did not give the correct predictions when compared with experimental results obtained from impact damaged specimens. Also both Russell's and Lin's[2,3] computer programs were found to predict inaccurately higher plate natural frequencies, and no comparison with free vibration experimental results of composite plates were given to examine the validity of the analytical prediction. The local resonance model, i.e. that local areas of damage vibrate independently of the rest of the structure, was assumed and not analytically justified. Moreover, their proposed models, were able to treat only those delaminations that occured at an interphase, separating symmetric sub-plates above and below the delamination. Obviously, a delamination can occur virtually between any interphase. Finally they experimentally tested laminates over a frequency range between 10 and 25 kHz, leaving unexplored what happens in the 0-10 kHz frequency region.

In the present work a mechanism affecting the dynamic behavior of cracked composite plates has been developed and analytically justified, using the concepts of numerical analysis. The finite element method applied was based on the principles of the three dimensional theory of

linear anisotropic elasticity. Damaged material dynamic behavior was modeled, and vibrothermographic experiments were performed over a wide frequency range. Material dissipative mechanisms were identified in order to explain the mechanical energy conversion into heat. Data from other complementary NDE methods, such as the SPATE technique (Stress Pattern Analysis by Thermal Emission) were also compiled to yield additional useful information.

1.2 Objectives

The objectives of this study were as follows :

1. To simulate the dynamic behavior of anisotropic composite plates with delaminations over a broad frequency range.
2. To construct an analytical model such that a delamination can occur at any interface.
3. To prove the 'local resonance model'.
4. To apply Vibrothermography and SPATE over a wide frequency range to detect and evaluate damage in composite plates.
5. To identify heat generation mechanisms responsible for vibrothermal structural and damage heat patterns.
6. To advance Vibrothermography and SPATE for nondestructive evaluation of advanced composite materials and structures.

2. EXPERIMENTAL OBSERVATIONS

2.1 Experimental Setup

Figures 1,2 show the equipment used to perform the vibrothermographic testing. An electromagnetic/ piezoelectric shaker (Wilcoxon Research F4/F7) together with a power amplifier (Model PA7C) and a matching network (Model N7C) were used to excite the composite specimens over a wide range of frequencies, typically between 0-20 kHz. The emitted infrared thermal signal was detected by an infrared thermographic camera (AGA Thermographic 680, 780 models), and the image was displayed on a monitor with a variable temperature scale. The working principle of the infrared camera is thermal imaging, i.e. investigating the environment as it appears in the infrared and producing pictures from the invisible thermal radiation constantly being emitted, absorbed and re-emitted by all matter. The infrared spectrum covers wavelengths greater than 0.75 μm in the electromagnetic spectrum, extending to the range of microwaves.

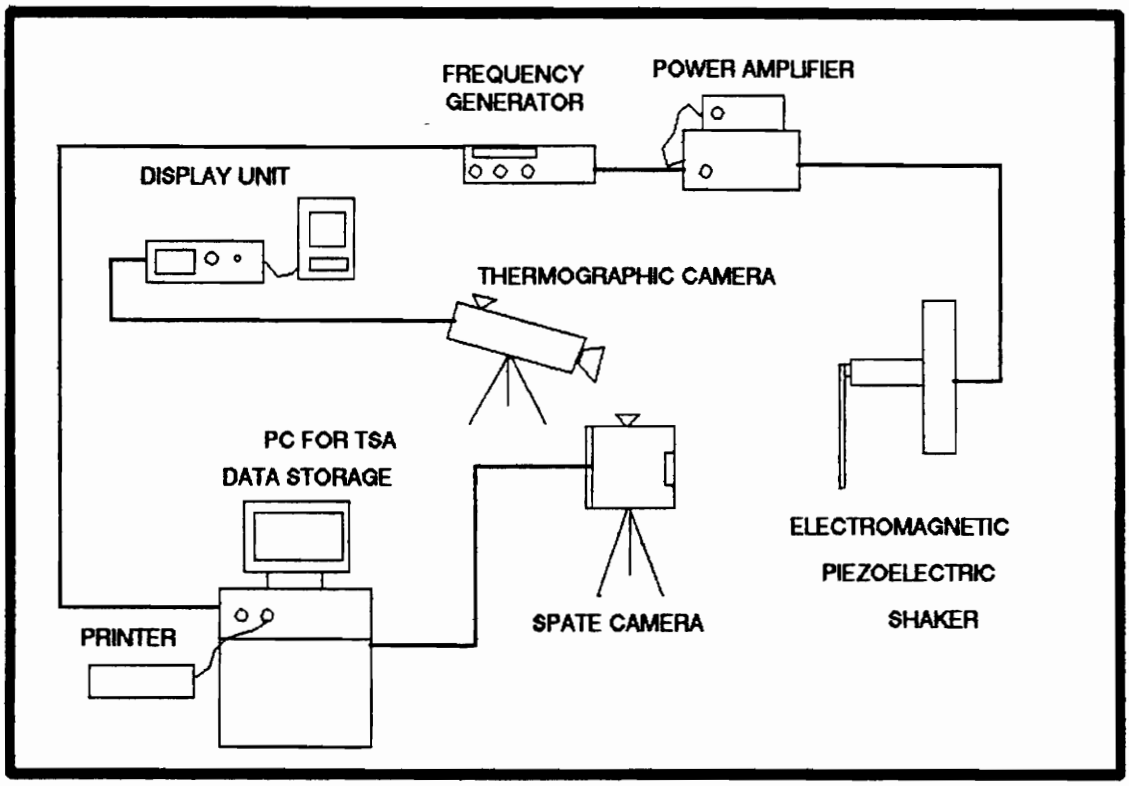

Figure 1. Vibrothermographic experimental setup. Also shown is the SPATE unit system used for thermoelastic analysis.

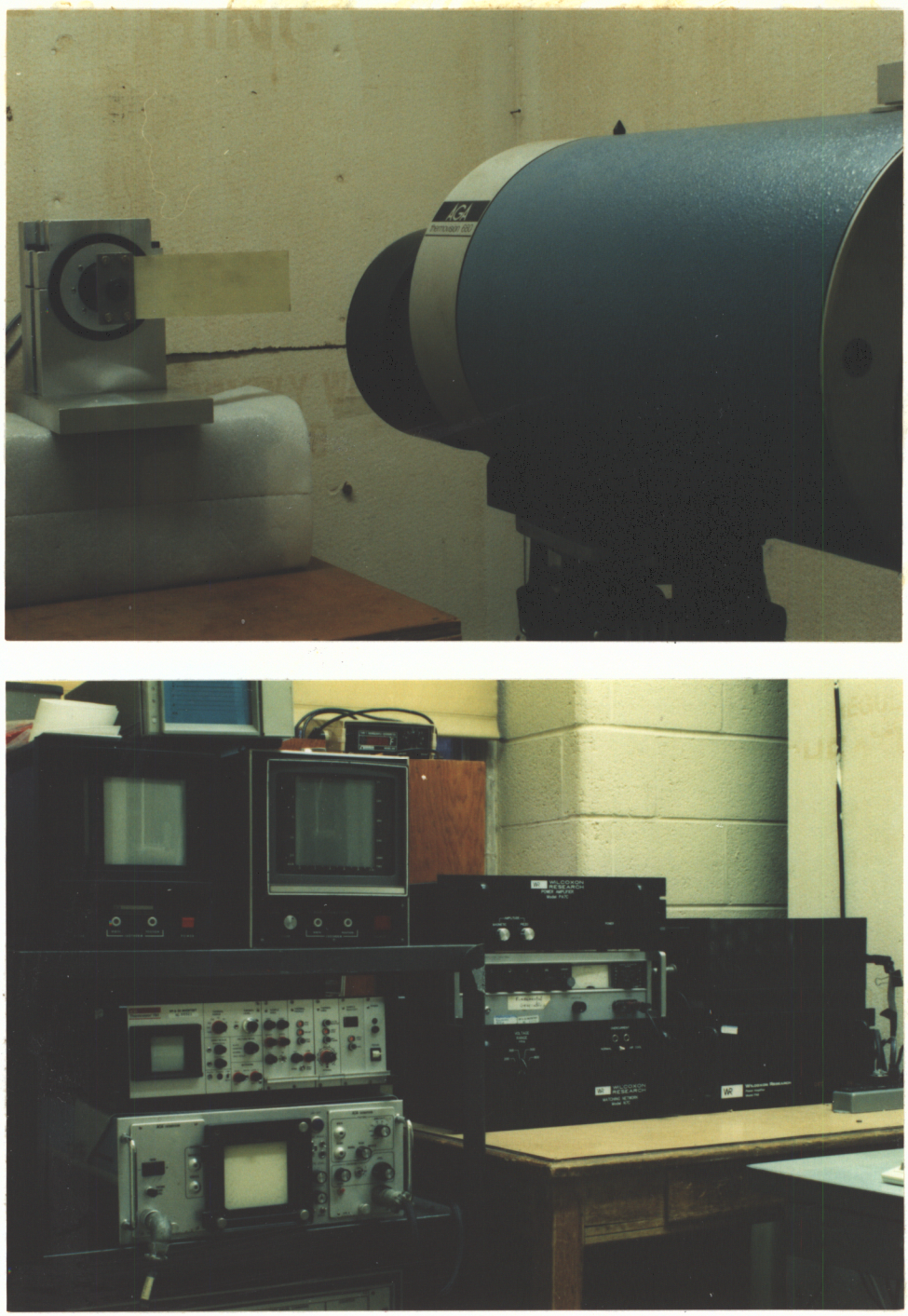

Figure 2. Equipment used for testing. (a) Vibration shaker and thermographic camera; (b) Frequency generator, power amplifiers and display monitors.

2.2 Vibrothermographic Experimental Observations

Using the setup described in section 2.1, different undamaged and damaged composite plates were tested. The first specimen tested was a $(0/90/90/0)_s$ 12 by 12 inches (304.8x304.8 mm) graphite/ epoxy plate. Figure 3a displays the infrared signal of the undamaged laminate oscillating at 20 kHz, with the shaker head attached to the plate's center. At this particular frequency, a global increase in temperature (approximately 0.5 C) was observed, and many local hot spots were formed, possibly following regions of high strains and stresses.

Next, a 6 by 6 inches (152.4x152.4 mm) $(0/90)_s$ plate with three simulated delaminations was forced to vibrate over a wide frequency range. Again heat contours appeared after 10 kHz. At some particular frequencies heat generation seemed to be affected by global resonance patterns (Figure 3b), being especially intense around the delaminated regions and roughly following their shape. At some other frequencies heat appeared to be localized around the damaged areas with limited global contribution (Figure 4a). Also the thermal image around the imperfection areas was found to change shape (shifted) with frequency change (Figure 4b).

By carefully examining the thermal images in Figures 3,4, the following questions arise :

1) Why do heat patterns appear only after approximately 10 kHz?

2) What is the mechanism responsible for heat generation?

3) Is surface friction present at the internal delaminations?

4) What is the structural resonance contribution to the overall thermal response?

5) Is a local phenomenon taking place at the damaged regions that is strongly frequency dependent and also influences heat generation?

6) How do the stress concentrations at the flaw areas affect the thermal patterns?

7) What is the boundary condition of the cracked areas?

It is the purpose of this study to focus, investigate and explain some of the above experimentally observed phenomena, so that the method can be further advanced and applied to nondestructively evaluate advanced composite materials and structures.

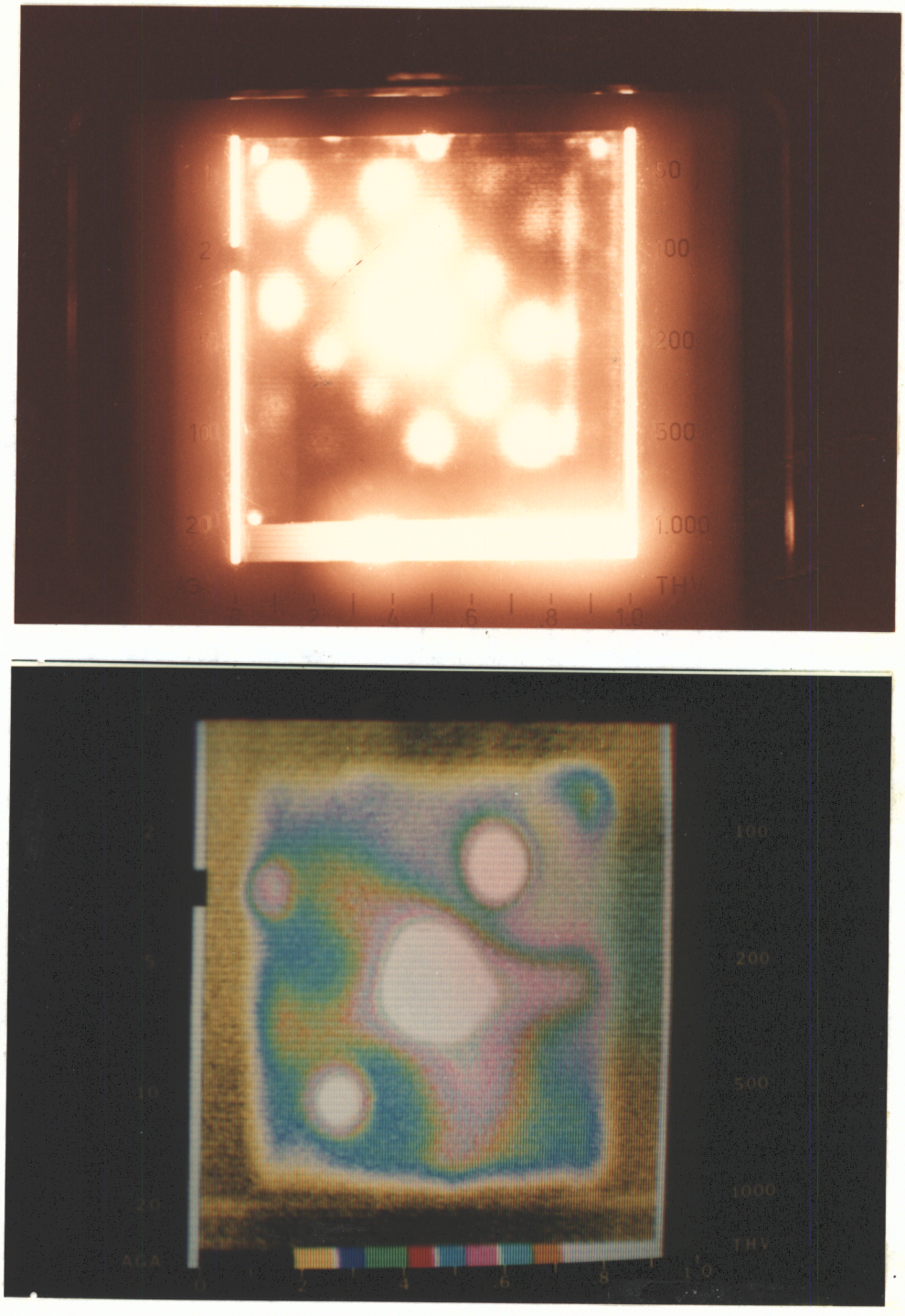

Figure 3. (a) Heat contours of a damage-free graphite epoxy plate at 20 kHz; (b) Contours of a damaged plate with the three simulated delaminations revealed.

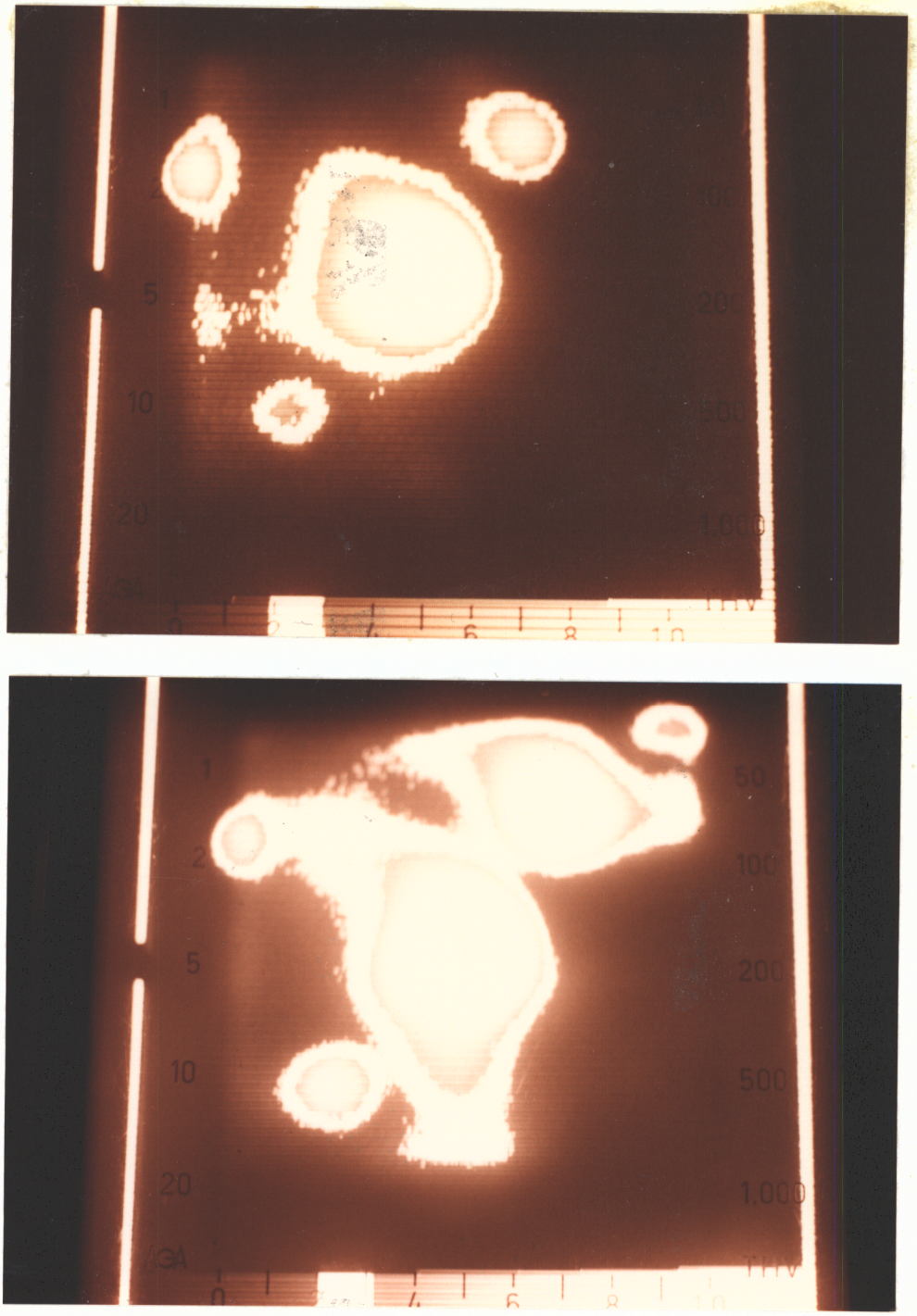

Figure 4. (a) Thermal contours localized at the defect areas at 13 kHz;

(b) Heat pattern changes with frequency (15 kHz).

3. 3D DYNAMIC FINITE ELEMENT ANALYSIS

3.1 The Delaminated Composite Plate model

In this chapter, a finite element scheme based on the three-dimensional theory of linear elasticity, for determining approximations to the natural frequencies and modes of delaminated, anisotropic composite plates is developed. The composite plate is modeled using the full equations of 3D anisotropic elasticity and the delamination is assumed to be infinitely thin. The small size of the flaw introduces the challenging computational problem of accurately determining high (in the kilohertz range) natural frequencies and the corresponding modes.

The undamaged composite plate occupies the region

$$P = \left\{ 0 < x < A,\ 0 < y < B,\ 0 < z < Z \right\}$$

in three dimensional space, where A,B, and Z denote the plate length, width, and thickness, respectively. The plate is composed of N layers $z_{n-1} < z < z_n$, $n = 1,\ldots,N$, with $z_0 = 0$ and $z_N = Z$. The delamination , which occurs between two of the layers at, say, $z = z_d$ for some d between 1 and $N - 1$, is assumed to occur over the region

$$D = \left\{ 0 < \alpha_1 < x < \alpha_2 < A,\ 0 < b_1 < y < b_2 < B,\ z = z_d \right\}.$$

Thus, $(\alpha_2 - \alpha_1)$ and $(b_2 - b_1)$ are the flaw dimensions and the damaged plate is described by the slit domain $\Omega = P \setminus D$, i.e., the domain P with the slit D. The boundary of Ω consists of six faces of P and the upper and lower surfaces of the slit D.

The equations of three-dimensional elastodynamics valid within the n-th lamina are given by

$$\sigma_{ij,j}^{(n)} + b_i^{(n)} = \rho^{(n)} \ddot{u}_i^{(n)} \quad \text{for } i=1,2,3 \text{ and } n=1,\ldots,N \tag{2.1}$$

where, for the n-th layer, $\sigma_{ij}^{(n)} = \sigma_{ji}^{(n)}$ denotes the entries of the symmetric Cauchy stress tensor, $b_i^{(n)}$ the components of the system body forces, $\rho^{(n)}$ the mass density, and $u_i^{(n)}$ the components of the displacement vector.

Invoking the standard assumptions of linear elasticity, we have the material constitutive law

$$\sigma_{ij}^{(n)} = C_{ijkl}^{(n)} \varepsilon_{kl}^{(n)} \tag{2.2}$$

with respect to the material coordinate axis system xyz. With respect to a global plate axis system XYZ, equation (2.2) becomes

$$\sigma_{ij}^{(n)} = \bar{C}_{ijkl}^{(n)} \varepsilon_{kl}^{(n)} \tag{2.3}$$

Introducing a reduced notation for ijkl

$11 \rightarrow 1$, $22 \rightarrow 2$, $33 \rightarrow 3$, $23 \rightarrow 4$, $13 \rightarrow 5$, $12 \rightarrow 6$, the matrix $\bar{C}$ is given for every (orthotropic) layer n by

$$\left[\ \bar{C}\ \right]=\begin{bmatrix} \bar{C}_{11} & \bar{C}_{12} & \bar{C}_{13} & 0 & 0 & \bar{C}_{16} \\ & \bar{C}_{22} & \bar{C}_{23} & 0 & 0 & \bar{C}_{26} \\ & & \bar{C}_{33} & 0 & 0 & \bar{C}_{36} \\ & \text{Symmetric} & & \bar{C}_{44} & \bar{C}_{45} & 0 \\ & & & & \bar{C}_{55} & 0 \\ & & & & & \bar{C}_{66} \end{bmatrix} \tag{2.4}$$

The expressions for the elements of the $\bar{C}$ matrix are

$$\begin{aligned}
\bar{C}_{11} &= m^4C_{11} + 2m^2n^2(C_{12} + 2C_{66}) + n^4C_{22} \\
\bar{C}_{12} &= n^2m^2(C_{11} + C_{22} - 4C_{66}) + (n^4+m^4)C_{12} \\
\bar{C}_{13} &= m^2C_{13} + n^2C_{23} \\
\bar{C}_{16} &= mn\ [\ m^2(C_{11} - C_{12} - 2C_{66}) + n^2(C_{12} - C_{22} - 2C_{66})] \\
\bar{C}_{22} &= n^4C_{11} + 2m^2n^2(C_{12} + 2C_{66}) + m^4C_{22} \\
\bar{C}_{23} &= n^2C_{13} + m^2C_{23} \\
\bar{C}_{26} &= mn\ [\ n^2(C_{11} - C_{12} - 2C_{66}) + m^2(C_{12} - C_{22} - 2C_{66})] \\
\bar{C}_{33} &= C_{33},\ \bar{C}_{36} = mn(C_{13} - C_{23}) \\
\bar{C}_{44} &= m^2C_{44} + n^2C_{55},\ \bar{C}_{45} = mn(C_{55} - C_{44}) \\
\bar{C}_{55} &= n^2C_{44} + m^2C_{55}, \\
\bar{C}_{66} &= m^2n^2(C_{11} - 2C_{12} + C_{22}) + C_{66}(m^2 - n^2)^2
\end{aligned} \tag{2.5}$$

where m, n are the direction cosines of the fiber angle θ with respect to global plate coordinate X and each layer is taken to be orthotropic. The elements of the stiffness matrix , C_{ij}, for an orthotropic material in terms of the engineering constants is obtained by inversion of the compliance matrix, S_{ij}

$$C_{11} = \frac{S_{22}S_{33} - S_{23}^2}{S} \qquad C_{12} = \frac{S_{13}S_{23} - S_{12}S_{33}}{S}$$

$$C_{22} = \frac{S_{33}S_{11} - S_{13}^2}{S} \qquad C_{13} = \frac{S_{12}S_{23} - S_{13}S_{22}}{S}$$

$$C_{33} = \frac{S_{11}S_{22} - S_{12}^2}{S} \qquad C_{23} = \frac{S_{12}S_{13} - S_{23}S_{11}}{S} \qquad (2.6)$$

$$C_{44} = \frac{1}{S_{44}} \qquad C_{55} = \frac{1}{S_{55}} \qquad C_{66} = \frac{1}{S_{66}}$$

where $S = S_{11}S_{22}S_{33} - S_{11}S_{23}^2 - S_{22}S_{12}^2 - S_{33}S_{12}^2 + 2\,S_{12}S_{23}S_{13}$

and $S_{11} = \frac{1}{E_1}$ $S_{12} = -\frac{\nu 12}{E_1}$ $S_{13} = -\frac{\nu 13}{E_1}$

$S_{21} = -\frac{\nu 12}{E_1}$ $S_{22} = \frac{1}{E_2}$ $S_{23} = -\frac{\nu 23}{E_2}$

$S_{31} = -\frac{\nu 13}{E_1}$ $S_{32} = -\frac{\nu 23}{E_2}$ $S_{33} = \frac{1}{E_3}$

$S_{44} = \frac{1}{G_{23}}$ $S_{55} = \frac{1}{G_{13}}$ $S_{66} = \frac{1}{G_{12}}$

$$S_{14}=S_{15}=S_{16}=S_{24}=S_{25}=S_{26}=S_{34}=S_{35}=S_{36}=S_{41}=S_{42}=S_{43}=$$

$$S_{45}=S_{46}=S_{51}=S_{52}=S_{53}=S_{54}=S_{56}=S_{61}=S_{62}=S_{63}=S_{64}=S_{65}= 0$$

E_1, E_2, E_3 = Young's moduli in 1,2, and 3 directions

ν_{ij} = Poisson's ratio for transverse strain in the j-direction when stressed in the i-direction,

that is, $\nu_{ij} = -\dfrac{\varepsilon_j}{\varepsilon_i}$

G_{23}, G_{13}, G_{12} = shear moduli in the 2-3, 1-3 and 1-2 planes

Also the Maxwell - Betti reciprocal relation holds i.e.,

$$\frac{\nu_{ij}}{E_i} = \frac{\nu_{ji}}{E_j} \qquad i,j = 1,2,3$$

We also recall the strain - displacement relations of linear elasticity

$$\varepsilon_{kl}^{(n)} = \frac{1}{2}\left(u_{k,l}^{(n)} + u_{l,k}^{(n)} \right) \text{ for } k,l = 1,2,3 \tag{2.7}$$

Within each layer the material is assumed to be homogeneous so that for each value of n, the elements of the elasticity tensor $C^{(n)}$ are constant. Of course, (2.3) and (2.7) may be combined to yield, taking into account the symmetries of the elasticity tensor,

$$\sigma_{ij}^{(n)} = \bar{C}_{ijkl}^{(n)}\, u_{k,l}^{(n)} \tag{2.8}$$

If we assume that the body forces vanish, the substitution of (2.8) into (2.1) yields

$$(\bar{C}_{ijkl}^{(n)}\, u_{k,l}^{(n)})_{,j} = \rho^{(n)}\, \ddot{u}_i^{(n)} \text{ for } i = 1,2,3 \text{ and } n = 1,\ldots,N, \tag{2.9}$$

We will focus on the natural frequencies and modes of the plate, so that we examine the equations

$$-(\bar{C}^{(n)}_{ijkl}\, u^{(n)}_{k,l})_{,j} = (2\pi\omega)^2\, \rho^{(n)}\, \ddot{u}^{(n)}_i \quad \text{for } i = 1,2,3,\ n = 1,..,N, \tag{2.10}$$

where ω denotes the frequency. Of course (2.10) may be derived from (2.9) by using an appropriate transform, or through the assumption that the displacement is proportional to $e^{2\pi\sqrt{-1}\omega t}$.

On different segments of the boundary Γ of Ω, one may impose the displacement $\mathbf{u}$, or the traction $\mathbf{t} = \sigma\cdot\mathbf{n}$, or some compatible combination of these. The boundary condition along any clamped side of the plate is that the displacement vanishes. On the other hand, along any side that is free, the tractions, or stresses, vanish. Thus for example, for a cantilever plate clamped at x = 0, the appropriate boundary conditions for the components of the displacement vector are

$$u^{(n)}_i = 0 \text{ for } i = 1,2,3,\ n=1,\ldots,N, \text{ and } x = 0,\ 0 < y < B,\ z_{n-1} < z < z_n, \tag{2.11}$$

$$\sigma^{(n)}_{ij}.n = \bar{C}^{(n)}_{ijkl}\, u^{(n)}_{k,l}.n = 0 \text{ for } \begin{cases} i,j = 1,2,3, \\ n = 1,\ldots,N \end{cases} \text{ and}$$

$$\begin{cases} x = A, & 0 < y < B, & z_{n-1} < z < z_n \\ 0 < x < A, & y = 0, & z_{n-1} < z < z_n \\ 0 < x < A, & y = B, & z_{n-1} < z < z_n \\ 0 < x < A, & 0 < y < B, & z = z_{n-1} \\ 0 < x < A, & 0 < y < B, & z = z_n \end{cases} \tag{2.12}$$

Along the delamination, we assume that the tractions vanish, so that

$$\sigma_{ij}^{(n)}.n=\bar{C}_{ijkl}^{(n)}u_{k,l}^{(n)}.n=0 \text{ for } \begin{cases} i,j = 1,2,3, \\ n = d,d+1 \end{cases} \text{ and } \alpha_1 < x < \alpha_2,\ b_1 < y < b_2,\ z = z_d \tag{2.13}$$

Except at the delamination, it is assumed that the displacement is continuous along the interfaces between layers, i.e.,

$$u_i^{(n)} = u_{i+1}^{(n+1)} \text{ for } \begin{cases} i = 1,2,3, \\ n = 1,\ldots,N-1 \end{cases} \tag{2.14}$$

$$\text{and} \begin{cases} 0 < x < A,\ 0 < y < B, & z = z_n \\ \text{except for} & \\ \alpha_1 < x < \alpha_2,\ b_1 < y < b_2, & z = z_d \end{cases}$$

Thus the natural frequencies and modes of the delaminated, composite, cantilever plate are to be determined as solutions of the mathematical model (2.10)-(2.14).

3.2 A Weak Formulation

In order to define finite element approximations to the solution of the eigenvalue problem (2.10)-(2.14), we need to recast that problem into a weak form. To this end, the following spaces are introduced

$H^1(\Omega) = \{\ \phi \in L^2(\Omega) \mid \phi_{,1} \in L^2(\Omega) \text{ for } i = 1,2,3\ \},$

$\mathbf{L}^2(\Omega) = [L^2(\Omega)]^3 = \{\ \mathbf{v}=(v_1,v_2,v_3) \mid v_i \in L^2(\Omega) \text{ for } i=1,2,3\},$

$\mathbf{H}^1(\Omega) = [H^1(\Omega)]^3 = \{\ \mathbf{v}=(v_1,v_2,v_3) \mid v_i \in H^2(\Omega) \text{ for } i=1,2,3\},$

$\mathbf{H}_c^1(\Omega) = \{\ \mathbf{v} \in \mathbf{H}^1(\Omega) \mid \mathbf{v} = 0 \text{ on the clamped end } x = 0\ \},$

The space $\mathbf{H}_c^1(\Omega)$ will be used in defining a weak formulation for the cantilever problem.

Associated with the above spaces are the following norms and seminorms:

$$\| \phi \|_0 = \left(\int_\Omega \phi^2 \, d\Omega \right)^{1/2} \quad \text{for } \phi \in L^2(\Omega),$$

$$\| \mathbf{v} \|_0 = \left(\sum_{i=1}^{3} \| v_i \|_0^2 \right)^{1/2} \quad \text{for } \mathbf{v} \in \mathbf{L}^2(\Omega),$$

$$\|\phi\|_1 = \left(\| \phi \|_0^2 + \sum_{i=1}^{3} \| \phi_{,i} \|_0^2 \right)^{1/2} \quad \text{for } \phi \in H^1(\Omega),$$

$$\| \mathbf{v} \|_1 = \left(\sum_{i=1}^{3} \| v_i \|_1^2 \right)^{1/2} \quad \text{for } \mathbf{v} \in \mathbf{H}^1(\Omega),$$

$$| \phi |_1 = \left(\sum_{i=1}^{3} \| \phi_{,i} \|_0^2 \right)^{1/2} \quad \text{for } \phi \in H^1(\Omega),$$

and

$$| \mathbf{v} |_1 = \left(\sum_{i=1}^{3} | v_i |_1^2 \right)^{1/2} \quad \text{for } \mathbf{v} \in \mathbf{H}^1(\Omega),$$

Note that on $\mathbf{H}_c^1(\Omega)$, the seminorm $| \cdot |_1$ defines a norm equivalent to $\| \cdot \|_1$, i.e., there exist positive constants C_1 and C_2 such that

$$C_1 |\mathbf{v}|_1 \le \|\mathbf{v}\|_1 \le C_2 |\mathbf{v}|_1 \quad \text{for all } \mathbf{v} \in \mathbf{H}_c^1(\Omega)$$

A **Galerkin** weak formulation is obtained by defining the bilinear forms

$$\alpha(\mathbf{u}, \mathbf{v}) = \int_\Omega \bar{C}_{ijkl} u_{k,l} v_{i,j} \, d\Omega \quad \text{for all } \mathbf{u}, \mathbf{v} \in \mathbf{H}^1(\Omega)$$

$$b(\mathbf{u}, \mathbf{v}) = \rho \int u_i v_i d\Omega \quad \text{for all } \mathbf{u}, \mathbf{v} \in L^2(\Omega)$$

It is well known[4] that these bilinear forms are *symmetric, continuous, and coercive, i.e.* symmetry ⇒

$$\alpha(\mathbf{u},\mathbf{v}) = \alpha(\mathbf{v},\mathbf{u}) \quad \text{for all } \mathbf{u},\mathbf{v} \in H^1(\Omega)$$

$$b(\mathbf{u},\mathbf{v}) = b(\mathbf{v},\mathbf{u}) \quad \text{for all } \mathbf{u},\mathbf{v} \in L^1(\Omega);$$

continuity ⇒ there exist positive constants α_1 and β_1 such that

$$\alpha(\mathbf{u},\mathbf{v}) \le \alpha_1 \|\mathbf{u}\|_1 \|\mathbf{v}\|_1 \quad \text{for all } \mathbf{u},\mathbf{v} \in H^1(\Omega)$$

$$\text{and } b(\mathbf{u},\mathbf{v}) \le \beta \|\mathbf{u}\|_1 \|\mathbf{v}\|_0 \quad \text{for all } \mathbf{u},\mathbf{v} \in L(\Omega)^2$$

coercivity ⇒ there exists a positive constant α_2 such that

$$\alpha(\mathbf{u},\mathbf{u}) \ge \alpha_2 \|\mathbf{u}\|_1^2 \quad \text{for all } \mathbf{u} \in H^1(\Omega)$$

$$\text{and} \qquad b(\mathbf{u},\mathbf{u}) \ge \rho \|\mathbf{u}\|_0^2 \quad \text{for all } \mathbf{u} \in L^2(\Omega)$$

The weak formulation of the problem (2.10)-(2.14) for the cantilever plate is then given as follows: We seek $\lambda = (2\pi\omega)^2 \in \mathbb{R}$ and $\mathbf{u} = (u_1, u_2, u_3) \in H_c^1(\Omega)$ such that

$$\alpha(\mathbf{u},\mathbf{v}) = \lambda\, b(\mathbf{u},\mathbf{v}) \quad \text{for all } \mathbf{v} \in H_c^1(\Omega). \tag{2.15}$$

Integrating (2.15) by parts shows that sufficiently smooth solutions of (2.15) satisfy the partial differential equations (2.10), and that the boundary conditions (2.12) along the free ends of the plate and (2.13) along the delamination are natural for this weak formulation, i.e., are automatically satisfied by any $\mathbf{u}$ that satisfies (2.15). The boundary conditions (2.11) along the clamped end of the plate, as well as the continuity conditions (2.14) along non-damaged interfaces between layers, are essential conditions, i.e., they are imposed on the test functions $\mathbf{v}$ and trial functions $\mathbf{u}$.

Note that due to symmetry of the forms $\alpha(.,.)$ and $b(.,.)$, the

eigenvalues λ_m are real, and the associated eigenfunctions $\{\mathbf{u}_m\}$ may be chosen to be real-valued and to form an orthonormal set. Moreover, as a result of the coercivity of these forms, the eigenvalues λ are positive.

3.3 The Finite Element Scheme

Next a finite element approximation scheme for the solution of the eigen-problem (2.15) is defined. Let T_h denote a subdivision of Ω into boxes. The associated requirement is that interlayer boundaries do not cut through the interior of any of the boxes that make-up the subdivision T_h.

A finite element space $\mathbf{S}^h$ is defined as follows: $\mathbf{v}^h \in \mathbf{S}^h$ if

(i) $\mathbf{v}^h|_{\Omega_k}$, i.e., $\mathbf{v}^h$ restricted to an element $\Omega_h \in T_h$, is such that each of its components v_i^h, i = 1,2,3, is a trilinear polynomial whose value in Ω_h is determined by its value at the eight vertices of Ω_h; and

(ii) along the delamination region, no continuity of $\mathbf{v}^h$ across the delamination is enforced.

One may easily conclude that $\mathbf{S}^h \subset \mathbf{H}^1(\Omega)$. A practical method for satisfying (ii) is to define a double layer of nodes along the delamination, one associated with the elements below the delamination, and one with those above. (Figure 5). These nodes occupy the same geometric position in space, but could vibrate independently. At an undamaged point on the interface, the continuity of the displacement vector is imposed, i.e.,

$$u^{(+)} = u^{(-)},$$

where the superscripts + and - refer to interface nodes belonging to the upper and lower layers, respectively. Thus, at an undamaged interface point, the two layers meeting at an interface are not allowed to vibrate independently.

At an interface point corresponding to a delamination, two nodes corresponding to the same geometric point but to different layers in the structure are allowed to vibrate independently. In this case, the Galerkin formulation (2.15) implies that stresses vanish at the delamination. Since the nodes corresponding to the upper and lower layers occupy the same point in space, it is effectively assumed that the crack is infinitely thin.

The following subspace is also defined

$$S_c^h = \{\, v^h \in S^h \mid v^h = 0 \text{ at the clamped end } x = 0 \,\} = S^h \cap H_c^1(W) \subset H_c^1(\Omega)$$

For the cantilever plate, the approximate problem is given as follows: seek $\lambda^h \in \mathbb{R}$ and $u^h \in S_c^h$ such that

$$a(u^h, v^h) = \lambda^h\, b(u^h, v^h) \quad \text{for all } v^h \in S_c^h. \tag{2.16}$$

Again the coercivity, continuity, and symmetry properties of the bilinear forms imply that the eigenvalues λ_m^h are real and positive, and that the corresponding eigenvectors u_m^h are real and can be composed into an orthonormal set.

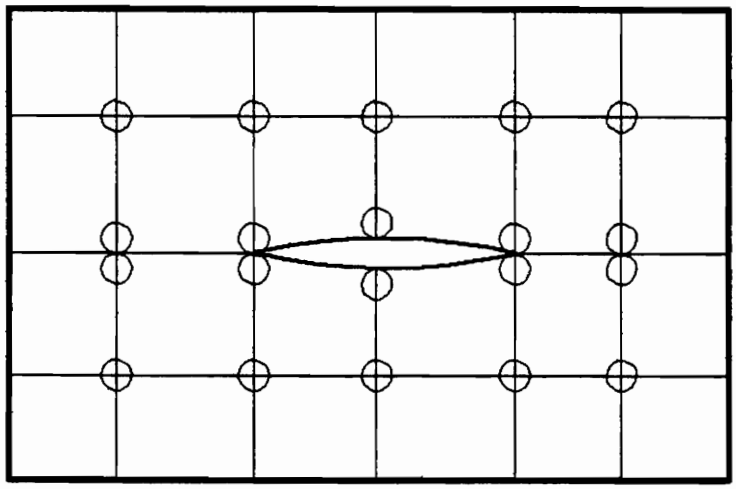

Figure 5. Interlayer crack model.

In the usual manner, i.e., by choosing a basis for $\mathbf{S}_c^h$, (2.16) can be shown to be equivalent to an algebraic eigenvalue problem. The size of this problem is determined as follows. At each node of T_h we have three displacement components. At the clamped end, we set these displacement components to zero, while at the delamination, two values for each component are allowed.

Whenever the plate is thin (in the z-direction), the aspect ratio of the boxes used in any practical computations will be rather large, i.e., the height of the boxes (in the z direction) are in general much smaller than the length and depth (in the x and y directions). (In the present calculations, the aspect ratio was as large as 120). This gives rise to the well known phenomenon of *shear-locking* within the computations. A common remedy for this problem is to use *reduced integration* techniques[5,6] wherein low-order quadrature rules are used to evaluate the integrals involving, in our case, out-of-plane shears. Thus, a one-point (centroid) rule is used to evaluate the integrals of terms in (2.16) with (i,j) = (1,3), (2,3), (3,1) or (3,2). The effect of the reduced integration is to make the plate less stiff. All other terms are integrated using an eight-point Gauss rule within each element.

The accuracy of the numerical approximation is limited by at least two factors. First, the use of reduced integration techniques to alleviate the shear-locking problem introduces errors in the computation; here we have the classic trade-off of enhancing stability at the expense of accuracy. Secondly, more serious is the fact that the

regularity of the solutions of (2.16) is severely limited because the domain Ω contains a slit, i.e. the delamination region, and the material properties are piecewise constant, and therefore discontinuous, in Ω. This lack of regularity severely limits the attainable accuracy of finite element approximations, and is one of the reasons for the use of simple piecewise trilinear test and trial functions. Using higher degree polynomials will not necessarily yield better accuracy.

The assembly of the element stiffness and mass matrices yields the algebraic eigenvalue problem

$$K\,\mathbf{U} = \omega^2\, M\,\mathbf{U}. \tag{2.17}$$

The components of $\mathbf{U}$ are the approximate values of the nodal displacements. The algebraic eigenvalue problem (2.17) is solved using the subspace iteration method of reference 7, with the stiffness and mass matrices stored in compacted form using the skyline[7] technique.

3.4 Computational Experiments at Low Frequencies

The numerical scheme developed is used to study the dynamic behavior of delaminated composite plates over a wide frequency range. All the computations were performed on an IBM mainframe computer with 250 Mb available memory. We will examine first the response of undamaged cantilever composite plates at lower frequencies, for which the experimental determination of the natural frequencies and mode shapes has been reported[8]. In that work the panels were constructed from a 5245C/G40-600 graphite Prepreg System made by BASF Structural Materials, Inc. The material properties, which are also used in this computational model are given in Table 1 (δ denotes the ply thickness).

The comparison of the first experimental and analytical natural frequency for five different configurations for length to width ratio of 2.5 is given in Table 2. Table 3 presents the comparison of higher experimental and numerical natural frequencies for three eight-ply laminates with the same aspect ratio as Table 2 . The frequencies are normalized with respect to the first one.

Considering the fact that the material properties are only known approximately, and taking into account other factors such as the uncertainty of the experimental modeling of boundary conditions, the agreement is found to be very good.

Table 1. Material properties used in the present study.

E_1	E_2	E_3	G_{12}	G_{13}
127.5 GPa	7.3 GPa	7.3 Gpa	4.2 GPa	4.2 GPa

G_{23}	ν_{12}	ν_{13}	ν_{23}	ρ	δ
4.7 GPa	0.33	0.33	0.40	0.00166 kg/cm^3	0.15367 mm

Table 2. Comparison of the experimental and analytical first natural frequency for five cantilever composite laminates.

Layups (L/W=2.5)	Experimental first frequency (Hz)	Finite element analysis (Hz)
(0/90/0/90)s	25.129	28.952
(0/45/0/-45)s	29.851	29.784
(0/30/0/-30)s	34.909	31.149
(0/60/0/-60)s	30.321	29.134
(90/-45/90/45)s	11.224	10.856

Table 3. Comparison of six natural frequencies for three plates. Frequencies are normalized with respect to the first one.

	(0/45/0/-45)s		(90/-45/90/45)s		(0/30/0/-30)s	
	EXP	**FEM**	**EXP**	**FEM**	**EXP**	**FEM**
MODE 1	1	1	1	1	1	1
MODE 2	3.13	2.94	6.11	6.03	2.92	2.83
MODE 3	6.38	6.26	8.68	8.34	6.24	6.29
MODE 4	10.49	10.00	17.02	17.26	9.74	9.62
MODE 5	17.35	16.94	26.18	25.95	13.63	13.47
MODE 6	19.05	18.62	32.83	34.29	17.69	18.09

Next we proceed to study the vibration behavior of a delaminated composite plate at low frequencies. First a flaw was embedded at a midplane of a (0/90/90/0) 8 x 4 in (203.2x101.6 mm) composite plate, i.e., between two layers with the same fiber orientation. The size of the flaw is then varied and the effect on the natural frequencies is monitored. Table 4 presents these results; and for a better visual comparison, they are also given in Figure 6a. The first column of Table 4 shows the corresponding natural frequencies of the undamaged panel. It is clear from the data that for large flaw sizes, even when the delamination is half the plate size, the effect on the first five natural frequencies is not significant. A drop in the natural frequencies was observed at the higher modes, with the magnitude of the drop increasing with increases in the flaw size. It is also noticeable that the plate barely "feels" the impact of small cracks.

Next, the delamination was placed between two layers with different fiber orientation, namely at the second interface of $a(90/45/-45)_s$ laminate. Table 5 and Figure 6b display the corresponding dynamic behavior for different flaw sizes. The results are similar to the ones obtained for the previous configuration. Now the first four natural frequencies were not seriously affected, even for large flaw sizes. Thus, the same dynamic behavior was observed when two layers having the same or different fiber orientations are partially delaminated. For more computational experiments see reference 9.

The computational results obtained before indicate that the lowest natural frequencies of a composite plate are little affected by delaminations. The vibrational modes corresponding to these frequencies are such that the upper and lower surfaces of the delamination vibrate in concert, i.e. in phase. At higher frequencies these surfaces tend to vibrate out of phase, and effects due to the delamination become noticeable. No local flaw behavior is observed at the lower natural frequencies.

3.5 Higher Frequencies and Local Resonance

Next the (0/90/90/0) graphite epoxy plate is allowed to vibrate at higher frequencies. For this case a new mesh is defined. Figure 7 shows a finite element discretization of the above plate consisting of a mesh of 1936 trilinear brick elements. Computations were made, for a cantilever plate, with a flaw size of 1.09 x 1.09 inches (27.6 x 27.6 mm), inserted either between the first two layers or at the middle plane. To check for any local behavior and for the effect of boundary conditions, the delamination was separately set at different locations, as indicated in Figure 7. Also, the flaw size was increased and the effect on any local behavior was monitored.

For the case of a flaw of dimension 1.09 x 1.09 inches, the above discretization resulted in a total of 3174 nodes and 7260 degrees of freedom. Relative convergence of many modes was achieved, this being necessary due to the small flaw size. The solution of one hundred eigenvalues and corresponding eigenvectors was obtained in less than 7.5

Table 4. The effect of delamination size on the natural frequencies of a (0/90/90/0) composite plate. The crack is inserted at the midplane.

MODE	UNDAMAGED Hz	.73x.73 in	1x0.73	1x1	1.45x1.45	2.18x2.18	4x2.37
1	20.4	20.4	20.4	20.4	20.4	20.2	19.7
2	37.0	37.0	37.0	37.0	37.0	37.0	36.9
3	128.7	128.7	128.7	128.7	128.6	127.8	121.5
4	154.6	154.6	154.6	154.6	154.6	154.3	153.0
5	241.8	241.8	241.8	241.7	242.0	240.8	237.0
6	319.4	319.4	319.4	319.1	318.4	291.4	258.5
7	366.1	365.1	364.4	361.9	349.0	315.0	302.5
8	397.6	397.3	397.2	395.5	393.7	384.4	376.3
9	543.1	542.3	541.8	539.7	533.2	512.7	422.1
10	728.5	727.7	725.7	721.9	692.0	602.0	436.8

Table 5. The effect of delamination size on the natural frequencies of a $(90/45/-45)_s$ composite plate. The crack is inserted at the second interface.

	UNDAMAGED	0.73x0.73in (18.5x18.5mm)	1x1 (25.4x25.4)	1.45x1.45 (36.8x36.8)	2.18x2.18 (55.3x55.3)	4x2.37 (101x60)
MODE NO 1	10.74	10.74	10.74	10.74	10.74	10.72
2	60.58	60.58	60.58	60.57	60.50	60.32
3	74.60	74.60	74.59	74.57	74.50	74.02
4	176.42	176.41	176.36	176.26	175.71	174.69
5	226.11	226.00	227.79	226.72	219.93	211.01
6	339.18	339.15	339.01	338.56	335.63	329.06
7	429.74	429.47	427.95	425.60	419.77	373.86
8	549.68	549.59	549.25	548.20	538.27	489.89
9	693.09	692.31	691.29	684.00	640.54	512.87
10	766.70	766.24	764.44	785.00	664.62	575.74
11	809.63	809.19	807.35	809.39	781.79	654.21
12	846.97	846.69	844.89	867.00	845.82	726.95

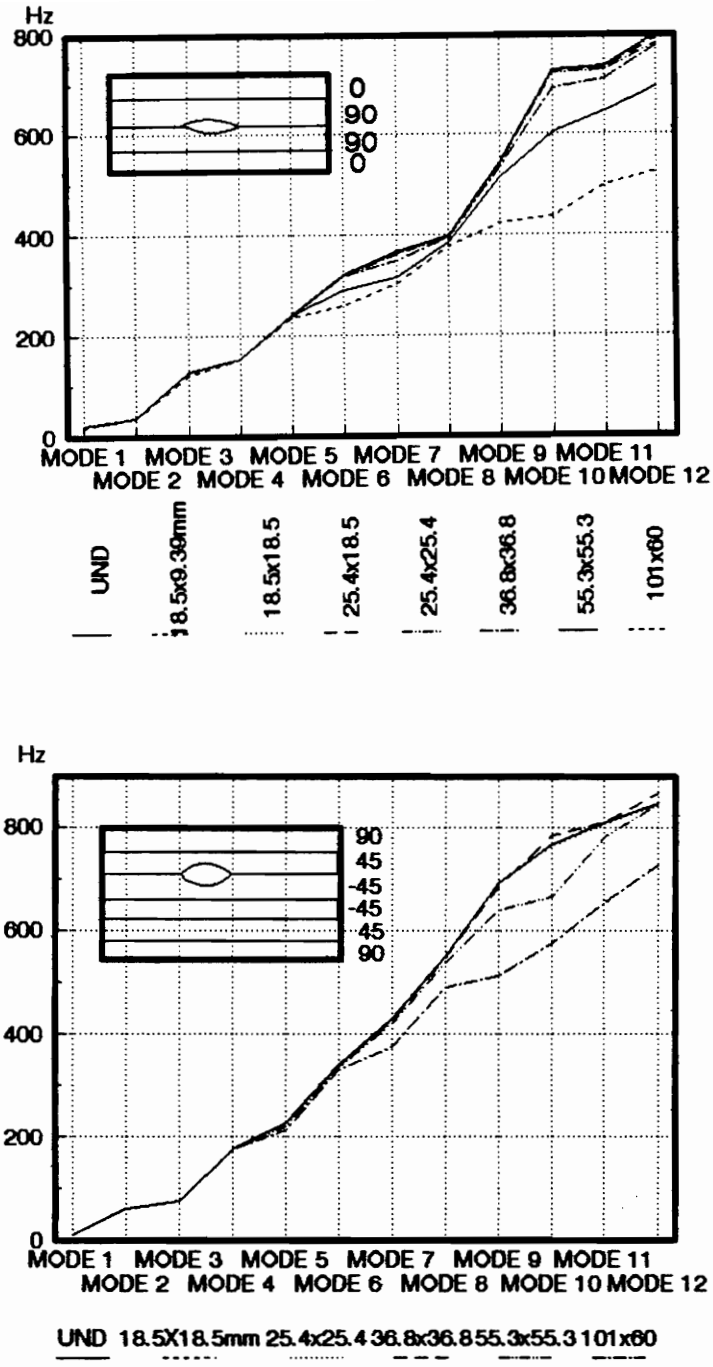

Figure 6. (a) Natural frequencies of a $(0/90)_s$ plate with different size flaws at the midplane; (b) Natural frequencies of a $(90/45/-45)_s$ plate with different size flaws at the 45/-45 interface.

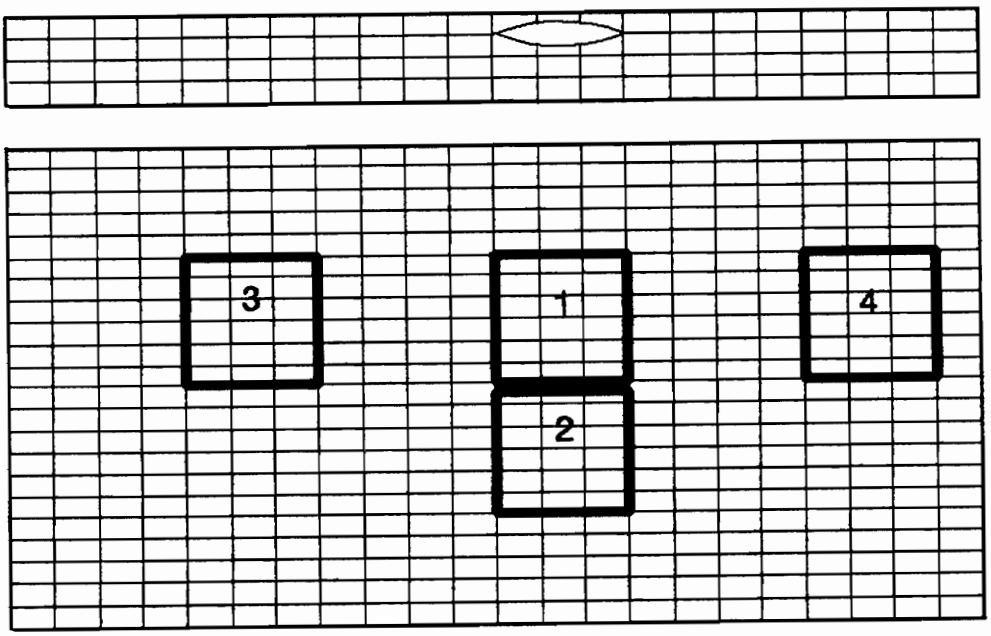

Figure 7. Finite element discretization of an 8x4x0.0242 in (0/90/90/0) delaminated plate with a damage of size 1.09x1.09 in. Four different flaw locations are indicated.

minutes of CPU. This attests to the speed and effectiveness of the subspace iteration solver. As a result , the structural and local dynamic behavior could be studied over a broad range of frequencies to nearly 20 kilohertz.

The discrete eigenvectors are normalized by the root mean square of their nodal values, i.e., so that their l^2 norm is unity. This facilitated, for any particular mode, the estimation of the energy concentrated at the delamination nodes as a percentage of the total energy of the whole structure.

Figure 8 displays (as a function of the frequency ω), the nodal values of the normalized eigenvectors at nodes near the middle of the delamination region. For this figure, the flaw is inserted between the two layers at position 1. (see Figure 7). The cantilever plate is clamped at the left end, and is free at the other boundaries. This plot indicates that at certain particular frequencies, a large portion of the energy of the structure is being concentrated at the delamination. At these frequencies, the local subplates in the vicinity of the flaw are excited and vibrate out of phase with the rest of the structure. The surface displacement is plotted in Figure 9. It is clear from the figure (the peak in Figure 9 is at the delamination) that a local vibration is taking place at the delamination; it appears that the upper subplate (above the delamination) resonates in its third mode.

For Figure 10, the delamination is inserted at the mid-plane of the (0/90/90/0) laminate, i.e., between the second and third layer, and at position 1. Now two identical subplates are created above and below the delamination, each being an unsymmetric (0/90) plate. Due to symmetry

considerations, we now expect that if a local excitation takes place, then the two subplates will become excited at the same frequencies. This is indeed the case, as can be seen in Figure 10, where the opposite motions of the damaged areas at 3 kHz are evident.

Finally, for Figure 11. the flaw dimensions were increased to 2.1 x 1.8 in (53.3 x 45.7 mm). This flaw is placed between the first two layers of a cantilever plate. The first local natural frequency is now approximately 400 Hz. This reduction is to be expected since now the flaw area, and thus the size of the independently vibrating subplates, is more than tripled. For more computational trials for flaw positions 2,3,4 (Figure 7) see reference 10.

The computational results reported above all indicate that at certain mechanical excitation frequencies, delaminations exhibit independent dynamic plate behavior and vibrate out of phase with the rest of the structure. This is what we call the **local resonance model.** The independence of the vibrations of the subplates above or below the flaw is not affected by the location of the flaw. However, the detailed structure of those vibrations is affected by flaw location, e.g., the natural frequency distribution depends on the proximity of the flaw to the boundaries of the structure. Also the larger the flaw size, i.e., the larger the subplate dimensions, the lower the excitation frequencies, as would be expected from independently vibrating subplates.

The computational results suggest that the surface above or below the delamination will actually "pop out" when an input forcing frequency coincides with one of the natural frequencies of the local subplates.

Recall from Chapter 2 that experimentally it was found that strong heat generation will follow from the above phenomenon. Furthermore, the computational results indicate that the values of the resonating forcing frequencies can yield quantitative information, e.g., as to the flaw size and depth.

3.6 Computation of Stresses.

The six components of the stress tensor can be easily estimated after the displacement components have been determined. The program first computes the strains at the element center, as a function of the displacement vector at the element nodes. The strain matrix is then multiplied by the stiffness matrix for the individual element to yield the stress matrix. Thus the stresses depend only on the element nodal displacements.

Figures 12,13,14 display individual stress contours occurring in the middle of the top layer of the (0/90/90/0) cantilever composite plate discussed previously. They are plotted when a flaw of size 2x1 in (50x25.4 mm) resonates independently of the rest of the structure. It can be seen from the contour plots that high normal stress concentrations exist at the flawed region. No shearing stress concentrations are observed. This was expected since for thin plates, a plane stress situation prevails. It is expected that shearing stress contribution will be significant for thicker plates .

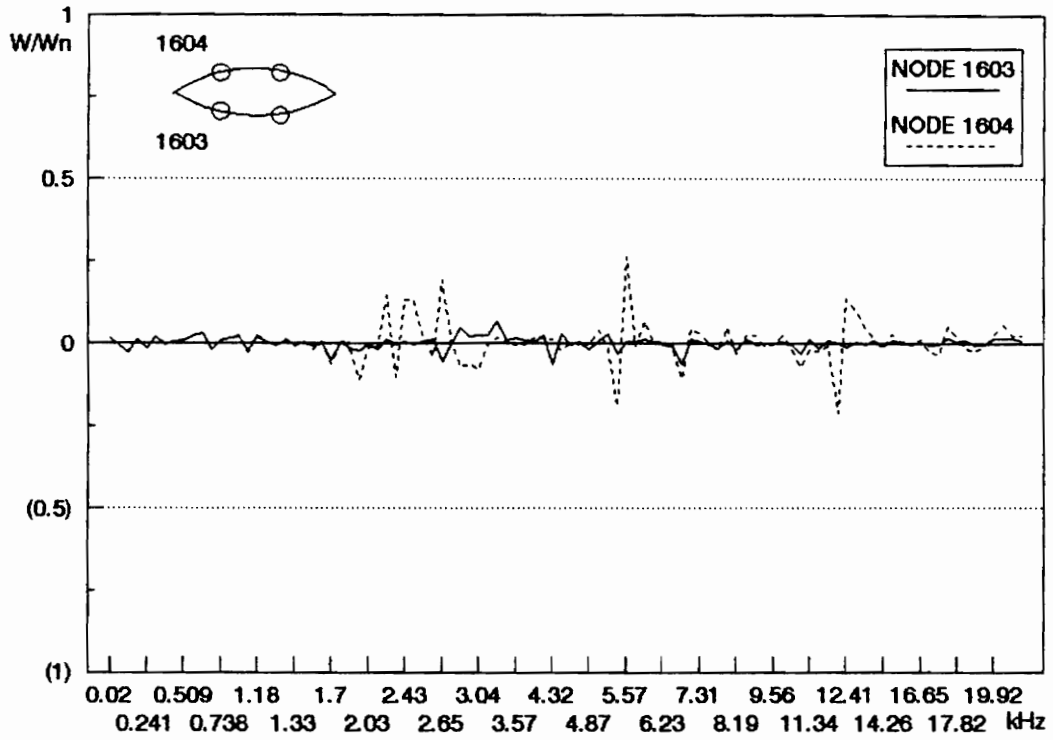

Figure 8. Displacement across delamination nodes (normalized by the root mean square of the displacements at all the nodes) versus natural frequency. The flaw is at location 1 (Figure 7) at interface 1.

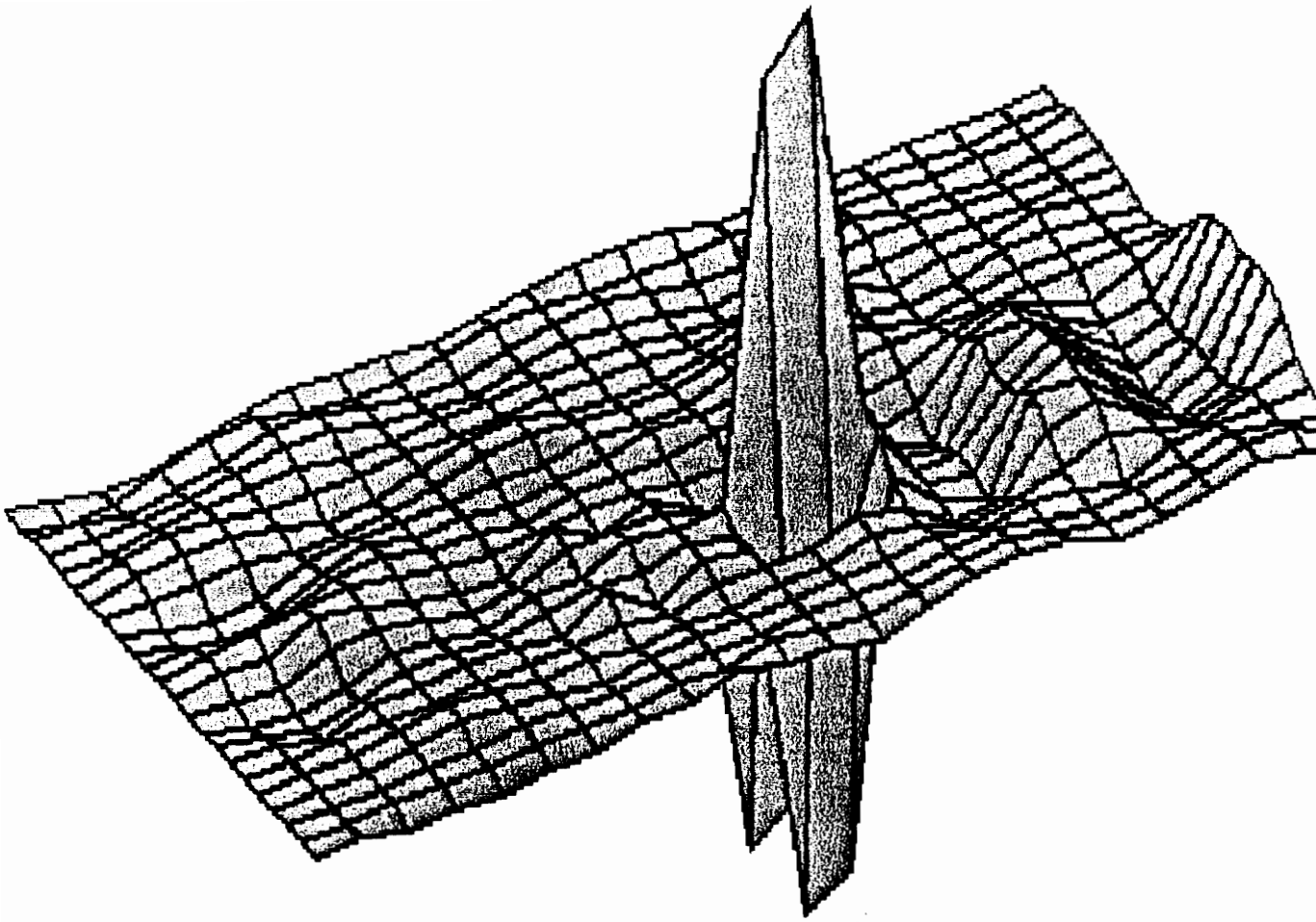

Figure 9. Top surface of the natural mode corresponding to 5.6 kHz for a cantilever plate with the flaw positioned as for Figure 8.

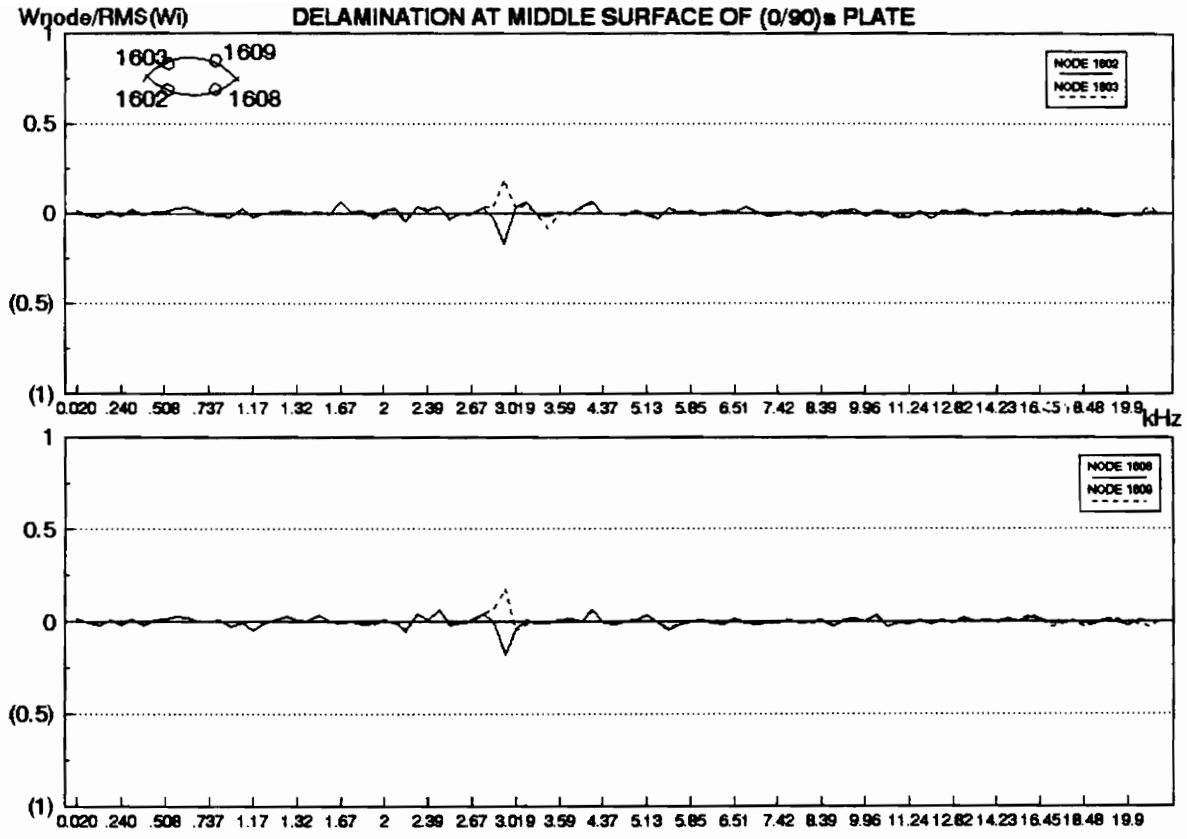

Figure 10. Displacement across delamination nodes (normalized by the root mean square of the displacements at all the nodes) versus frequency. The flaw is inserted at location 1 (Figure 7) at the middle interface.

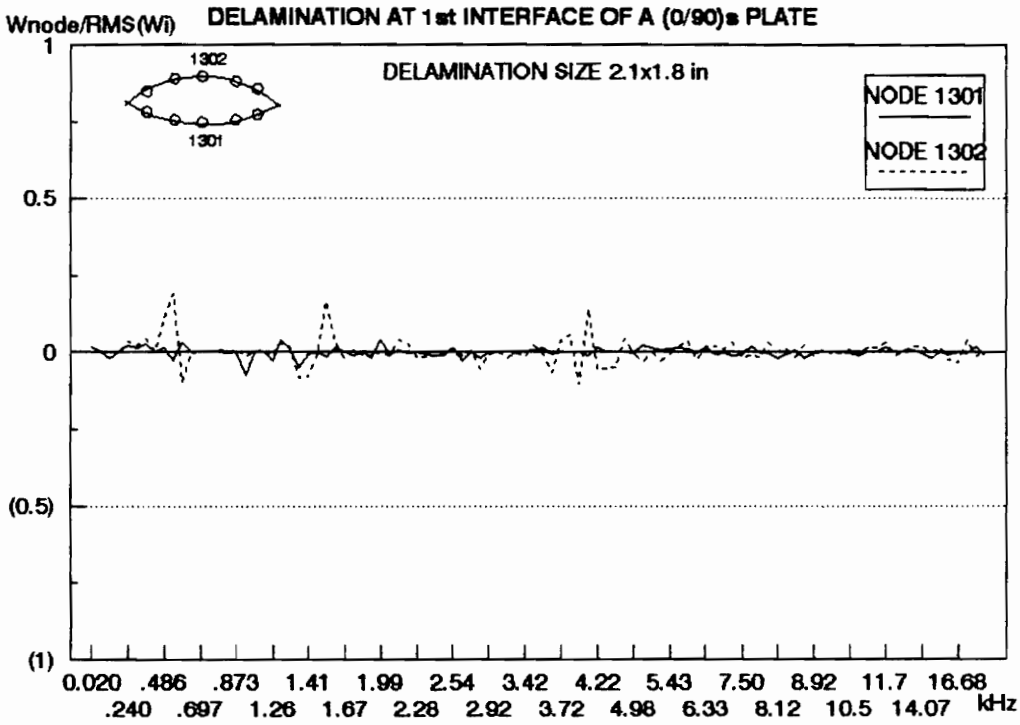

Figure 11. Normalized displacement across delamination nodes versus frequency. The flaw is of larger size and positioned at location 1 at interface 1.

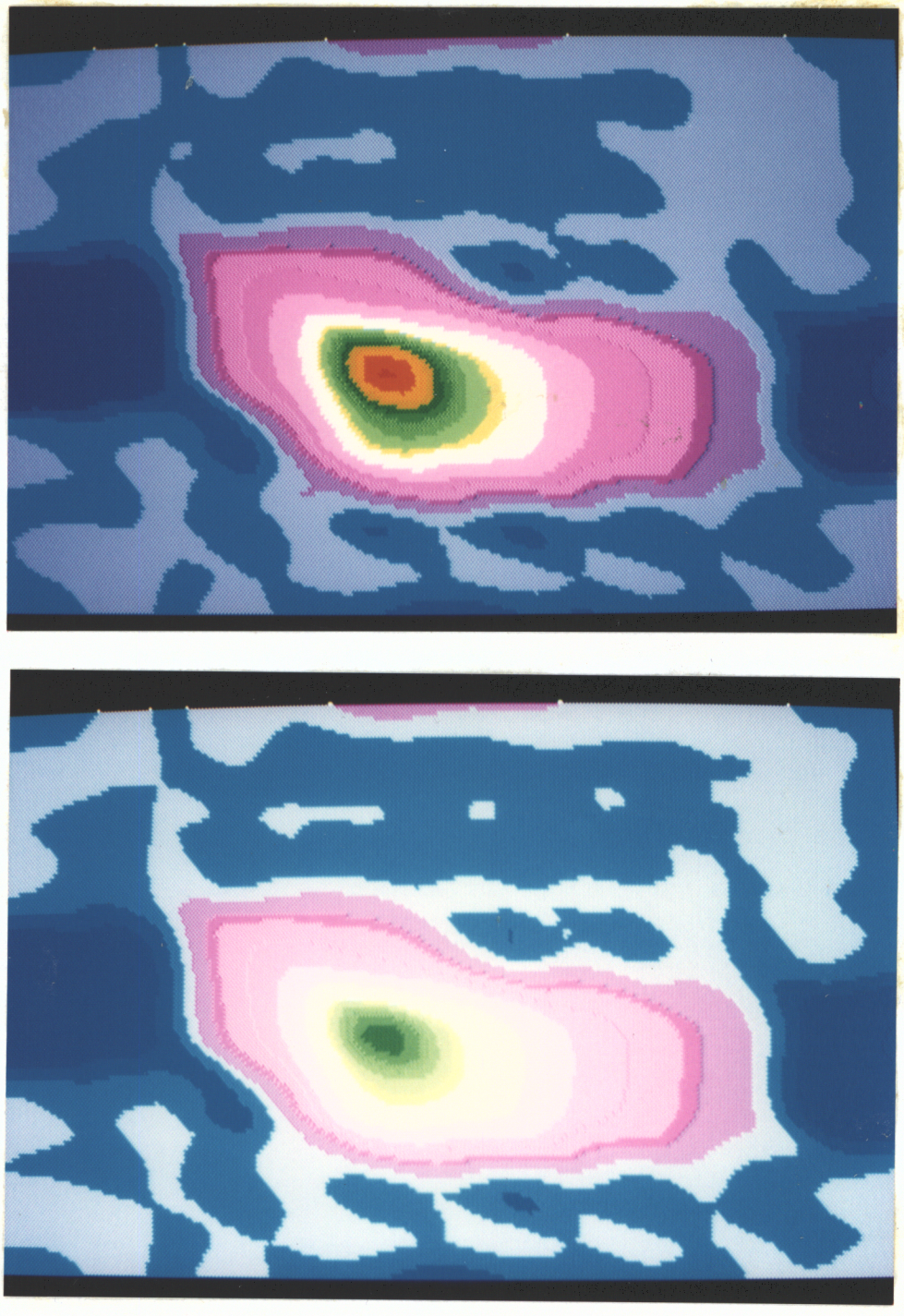

Figure 12. (a) Normal stress contours (σ_x) at local resonance;
(b) Normal stress contours (σ_y).

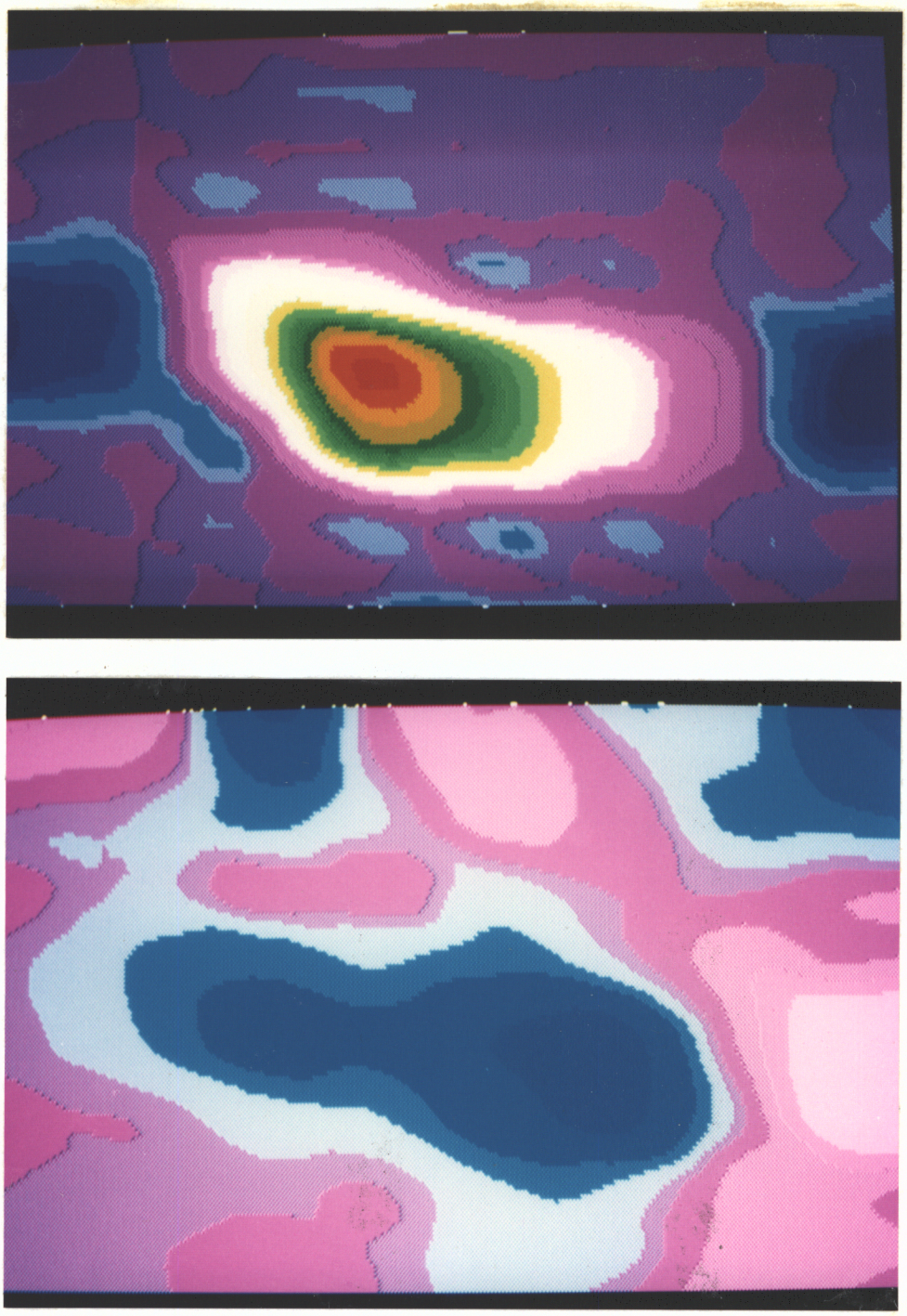

Figure 13. (a) Normal stress contours (σ_z) at local resonance.

(b) Shear stress contours (τ_{yz}).

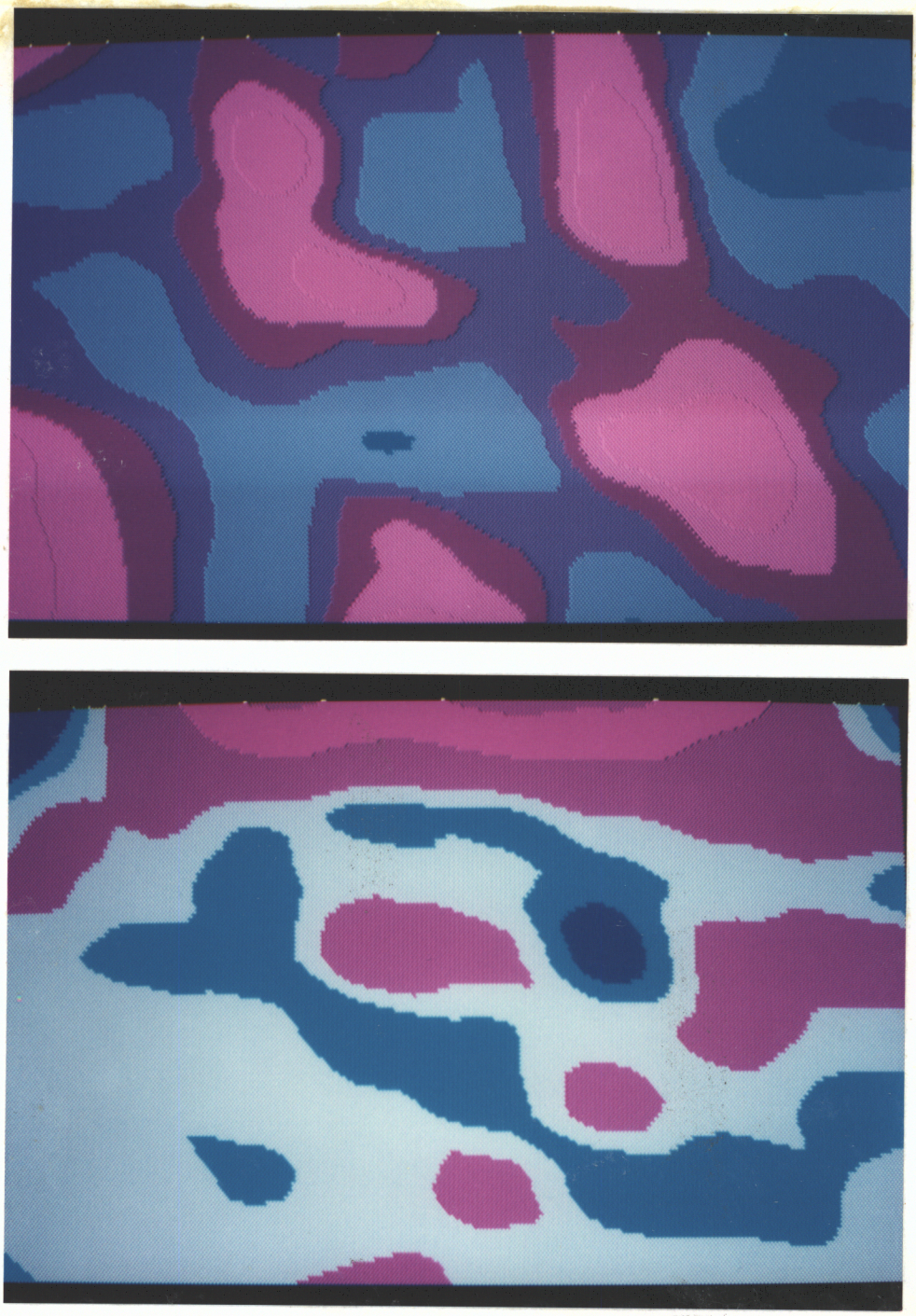

Figure 14. (a) Shear stress contours (τ_{xz}) at local resonance.

(b) Shear stress contours (τ_{xy}).

3.7 The Local Boundary Condition

An attempt was made to find if the local subplates resonate according to a perfect model with the appropriate boundary condition. Different computations were carried out, where the little plates were modeled either as simply supported or fully clamped. It was found that the local dynamic behavior was closer to the clamped model, but did not match exactly the behavior of a fully clamped plate. This was expected since the fiber orientation around the flawed areas will determine how stiff the actual boundary is. It was thus concluded that in general, the local delamination boundary was geometry dependent and closer to a clamped condition.

3.8 Limitations of the Finite Element Analysis

The finite element analysis presented did not take into account the contact between the delamination surfaces, which is expected to affect the motion and the natural frequencies of the subplates above or below the delamination. The model allowed the interference and self penetration of the delamination surfaces.

4. INTERPRETATION OF EXPERIMENTAL RESULTS

4.1 Vibrothermographic Testing of Undamaged Specimens

In Chapter 2, the frequency dependent heat generation was observed. It was also found to be particularly intense around the defect regions. In Chapter 3 the local resonance phenomenon was analytically justified. In order to determine the frequency that thermal patterns first appear, different *undamaged* composite specimens were forced to vibrate from 0 to nearly 20 kHz. Thus, the best frequency range for vibrothermographic evaluation can be defined. This chapter will also focus on the mechanisms responsible for the observed thermal behavior.

Damage-free specimens made from three different matrix and fiber materials were tested. The first one was a (90/0/90) 11 by 0.5 in (279.4x12.7 mm) graphite-epoxy beam. The second was a 6 ply unidirectional 1002 Scotchply E-glass epoxy 8 by 3 in (203.2x76.2 mm) panel. The third one was a Nicalon/Cas (SiC) ceramic $(0/90/0/90)_s$ beam of dimensions 2.9x0.21x0.1 in (7.36x0.53x0.25 mm) composed of Glass-Ceramic matrix and Silicon Carbide Fibers.

For all three specimens it was found that heat contours appeared only after the mechanical excitation exceeded 10 kHz. The material was thermally inactive below that frequency. Figure 15 shows thermal

patterns in the undamaged (90/0/90) graphite-epoxy beam at 13 kHz. Figure 16 presents heat generated in the glass-epoxy plate at 14 kHz. Figure 17 displays a slight increase in temperature inside the ceramic beam at 14 kHz. Note that for both the graphite-epoxy and the glass-epoxy panels, thermal contours appeared rather fast and steady state situation was rapidly developed. The ceramic beam was forced to vibrate for approximately ten minutes before the thermographic camera was able to detect any heat. It appears that for the ceramic beam, a different mechanism is responsible for heat generation, or else the mechanism occurs at a different frequency range for the ceramic beam.

4.2 Vibrothermographic Testing of a Damaged Specimen

Delamination in real composite panels can be simulated by folding mylar, teflon or halar tapes and sealing them with plain scotch tape[3]. Two delaminations were embedded in the two interphases of the (90/0/90) graphite-epoxy beam that was tested in an undamaged form in Chapter 4.1. The bigger flaw size was approximately 0.4x0.3 in (10.1x7.62 mm). The smaller was 0.3x0.3 in (7.62x7.62 mm). Figure 18 shows the finite element discretization (containing only one flaw) of this cracked anisotropic domain using trilinear brick elements. The dynamic behavior of the beam was studied over the frequency range from 0 to nearly 35 kHz. The displacement at the center delamination nodes divided by the root mean square of the displacement at all other nodes, is plotted in Figure 19. It is evident from the plot that at 13.5 and 14.5 kHz a

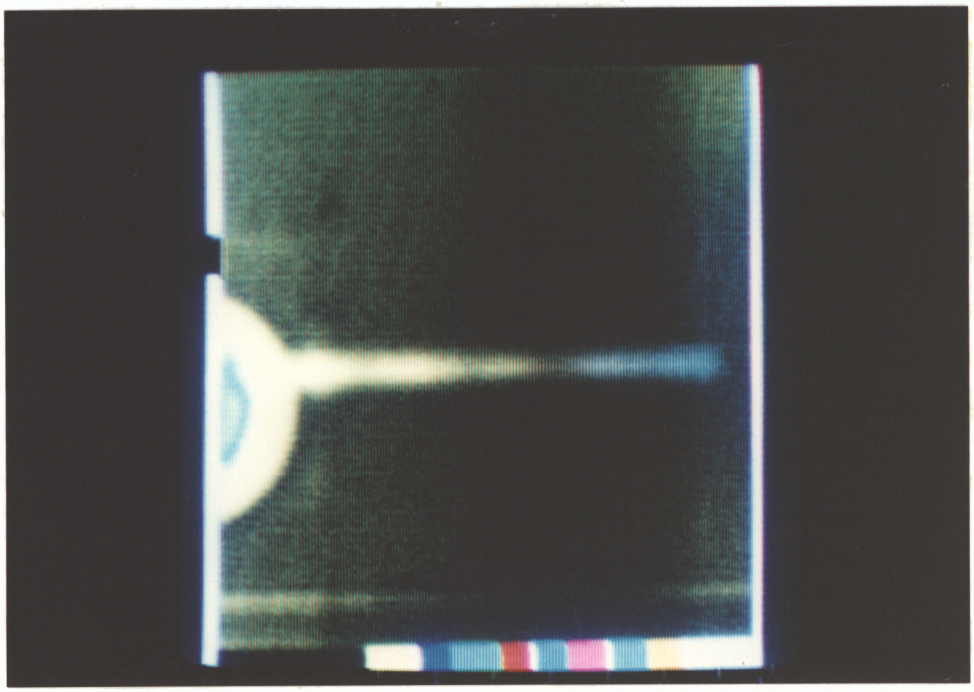

Figure 15. Thermal patterns on a (90/0/90) undamaged graphite-epoxy beam at 13 kHz.

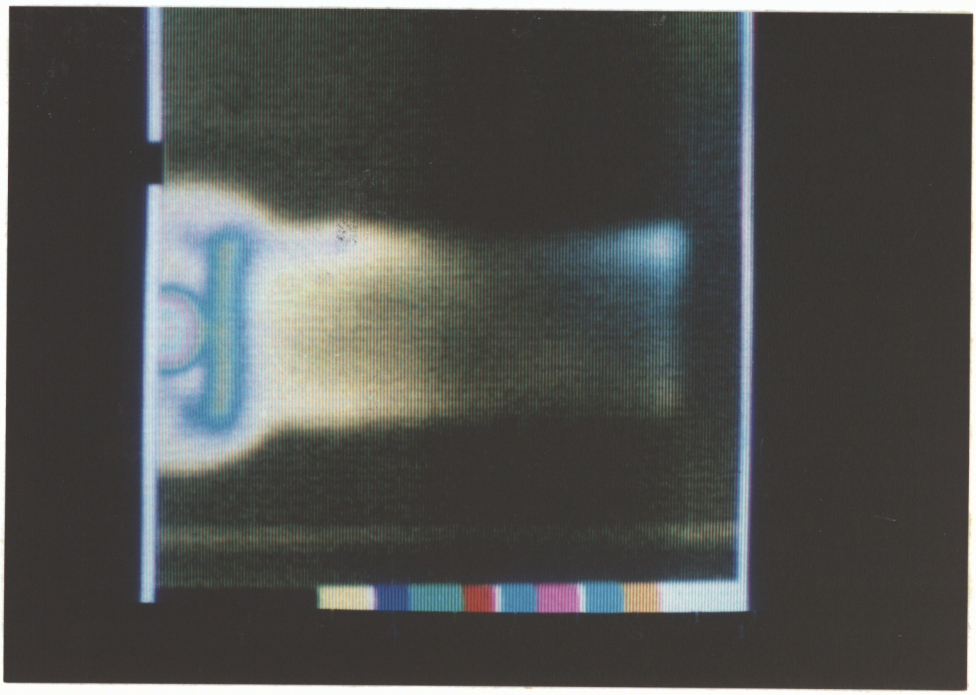

Figure 16. Thermal patterns on a unidirectional undamaged glass-epoxy panel at 14 kHz.

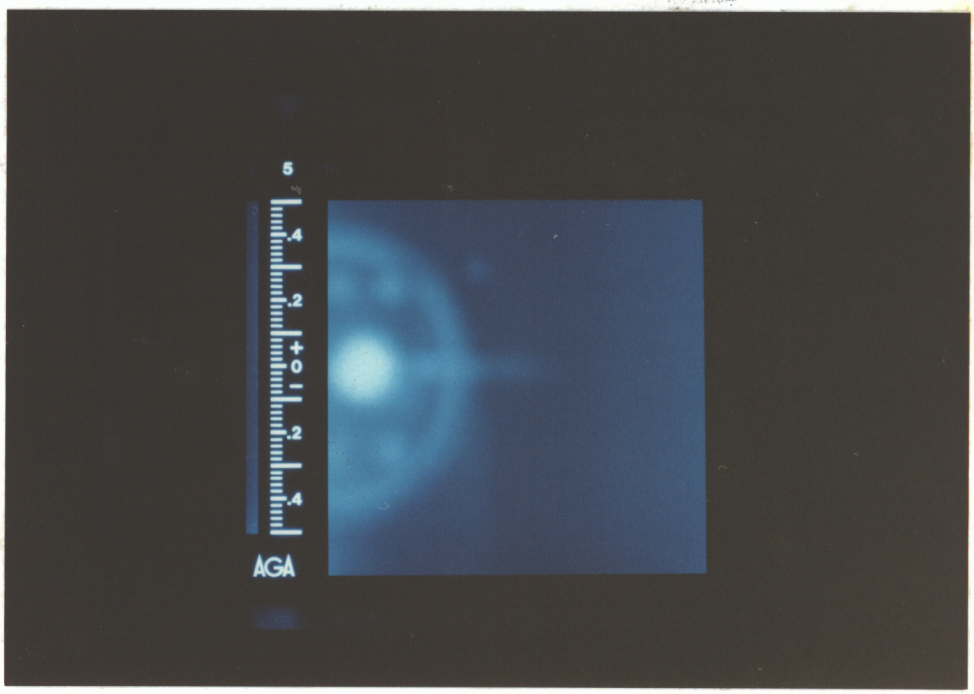

Figure 17. Heat generation in a $(0/90/0/90)_s$ undamaged Nicalon/Cas ceramic beam at 14 kHz.

large part of the energy absorbed by the composite beam in this mode, is concentrated at the flaw. At these particular frequencies, some local, independent vibration is taking place. Figure 20 displays the vibrothermographic signal of the damaged (90/0/90) beam specimen at 13.5 kHz excited as a cantilever structure. As can be seen from the figure, heat generation is particularly intense at the flaw area. The delaminated region appears to the thermographic camera as a hot spot and is approximately located. A similar experimental result was obtained for exciting frequencies between 13.5 and 15.0 kHz. No heat was detected at any other frequency from 0 to 25 kHz.

4.3 Thermoelastic (SPATE) NDE Evaluation.

SPATE (thermoelastic emission) analysis stands for Stress Pattern Analysis by thermal emission or simply Thermographic Stress Analysis (TSA). It is a powerful relatively new, noncontacting stress analysis technique which is based on the thermoelastic effect, which relates dynamic changes in strains to the temperature changes they induce. These small temperature changes (typically at most a few tenths of a degree) are monitored with an infrared photodetector. A dynamic loading must be applied to the component. For adiabatic conditions, the thermoelastic equation describing the change in temperature resulting for a given change in stress for isotropic materials can be written as[15]

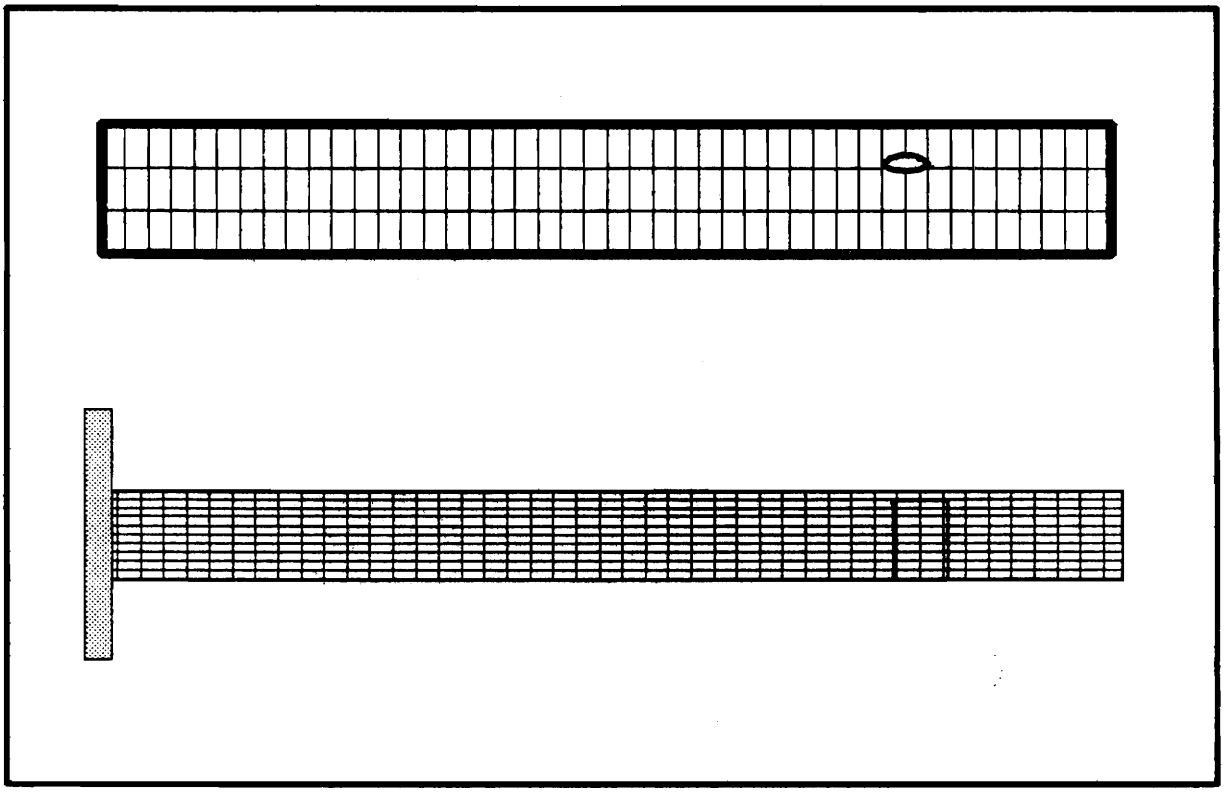

Figure 18. Finite element discretization of the damaged (90/0/90) beam.

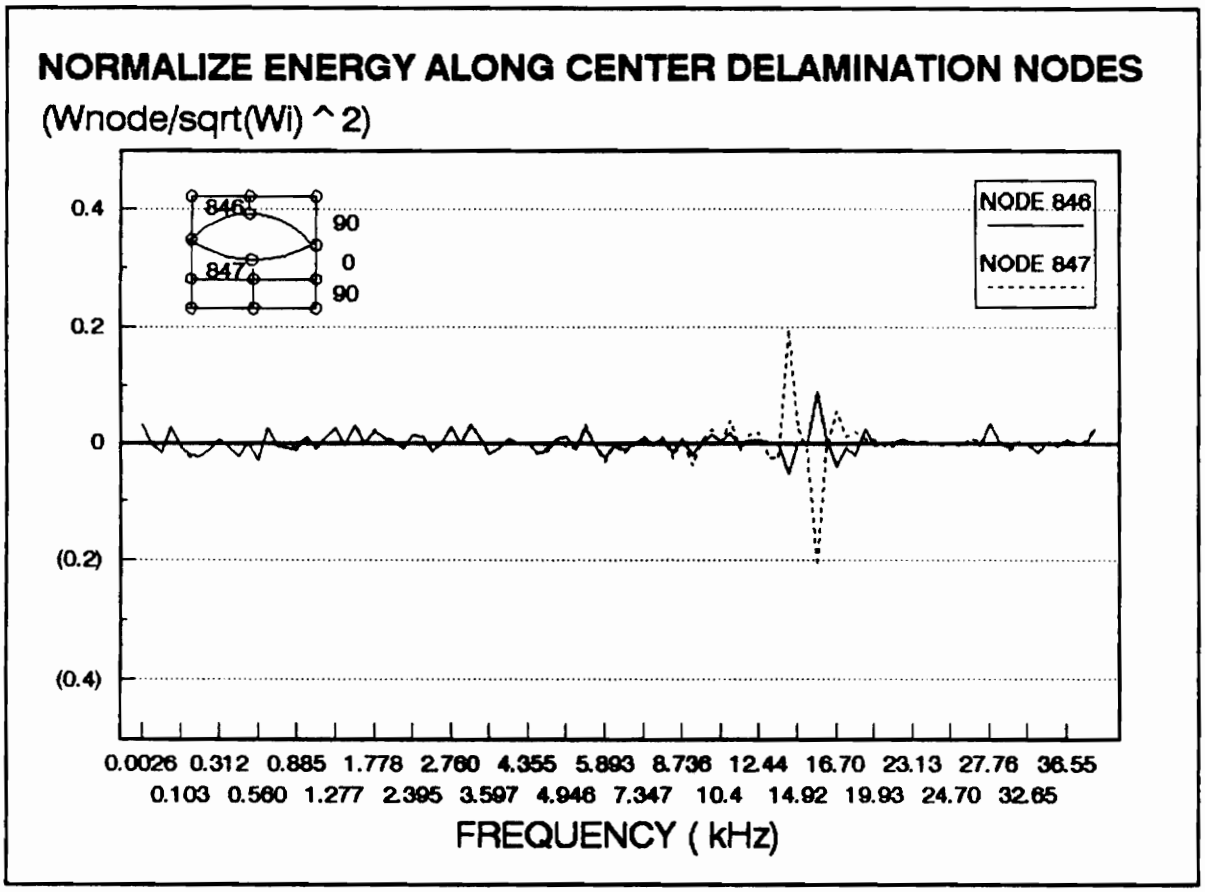

Figure 19. Displacement at delamination nodes (normalized by the root mean square of the displacements at all the other nodes) versus natural frequency.

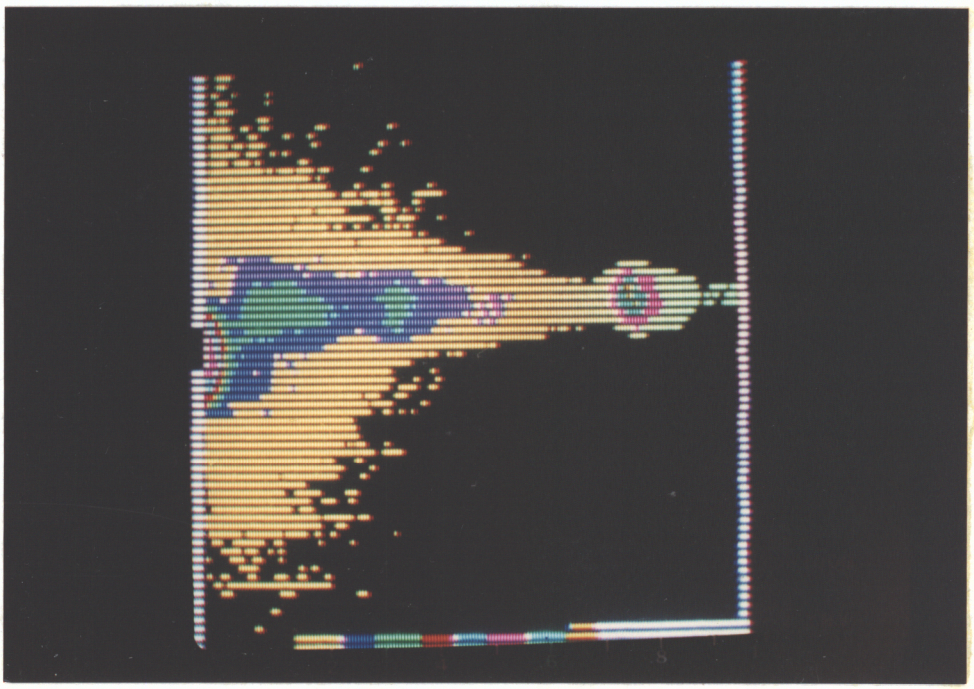

Figure 20. Vibrothermographic signal of the damaged beam specimen at 13.5 kilohertz.

$$\Delta T = \frac{-\alpha T}{\rho C_p} dI_1 \qquad (4.4)$$

For a unidirectional composite under uniaxial loading in the fiber direction the thermoelastic equation becomes[16]

$$\Delta T = -\frac{T}{\rho C_p}\left(\alpha_1 - \frac{1}{E_1^2}\frac{\partial E_1}{\partial T}\sigma_m\right)\Delta\sigma_1 \qquad (4.5)$$

where ΔT is the change of temperature, ρ the density, α the thermal conductivity, $\Delta\sigma$ the mean stress and C_p the specific heat at constant temperature.

The SPATE system detects the infrared flux emitted from points on an observable surface as a result of the minute temperature changes in a cyclically stressed structure or component. The Ometron SPATE 8000 equipment consists of an infrared-sensitive camera coupled to a correlator and a computer. The sensitivity of the system can detect a temperature discrimination typically better than 0.001^0 K. The measurement is taken synchronized with peak to peak amplitude of the periodic change in the sum of the principal stresses at the measuring point. The points are scanned in a raster-like manner under adjustable computer control and can be identified simultaneously on the surface by means of a visual channel incorporating an eyepiece or a projected light spot with cross wires. Therefore, the image obtained on the screen,

which results from adiabatic temperature changes, is point-wise related to the stress field or strain field induced by cyclic loading. One limitation is that the TSA signal only provides information on the *sum of the principle strains* I_1. Both area and line scans can be accomplished by using the above method over a large structural area in a short period of time. The output of the SPATE is the thermoelastic emission from the material under cyclic loading.

Figure 21 displays a typical thermoelastic experimental setup. The SPATE camera shown receives the thermoelastic signal of the vibrating composite specimens and displays it on a personal computer color monitor. The computer has special software to further manipulate the thermoelastic signal and store it for later imaging.

Figure 22 presents a SPATE area scan of the damaged beam resonating at 283.55 Hz in one of it's lower resonance modes. Stress intensities are represented by different colors, and areas of zero stress can be located. A normalized plot of the principle stresses along a beam line is displayed at the lower part of the plot. Note that there was no flaw detection at this exciting frequency and no irregular flaw behavior was observed.

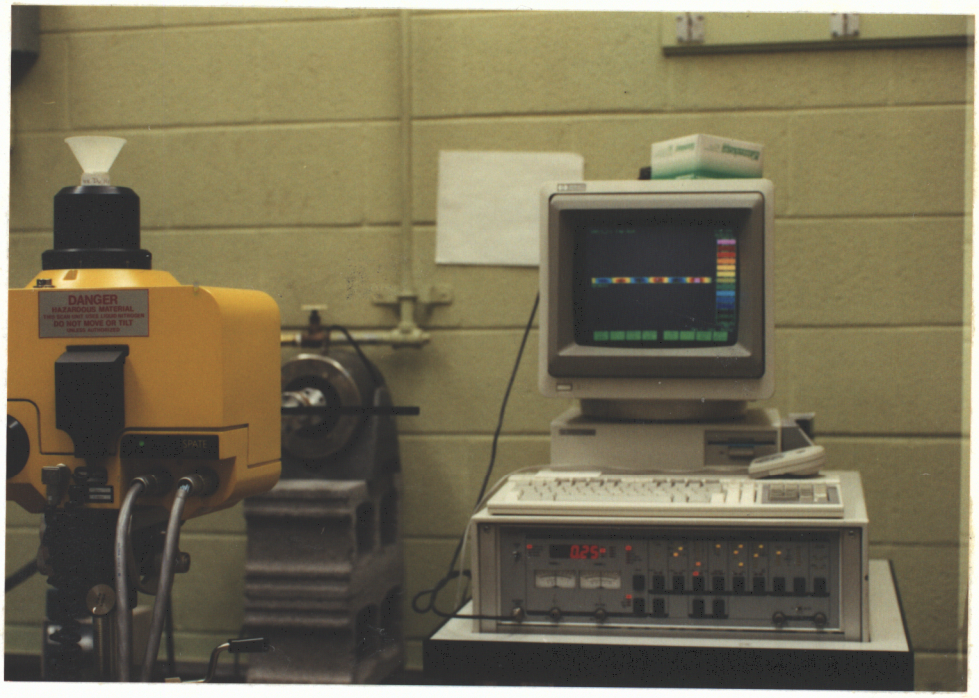

Figure 21. Thermoelastic SPATE experimental setup.

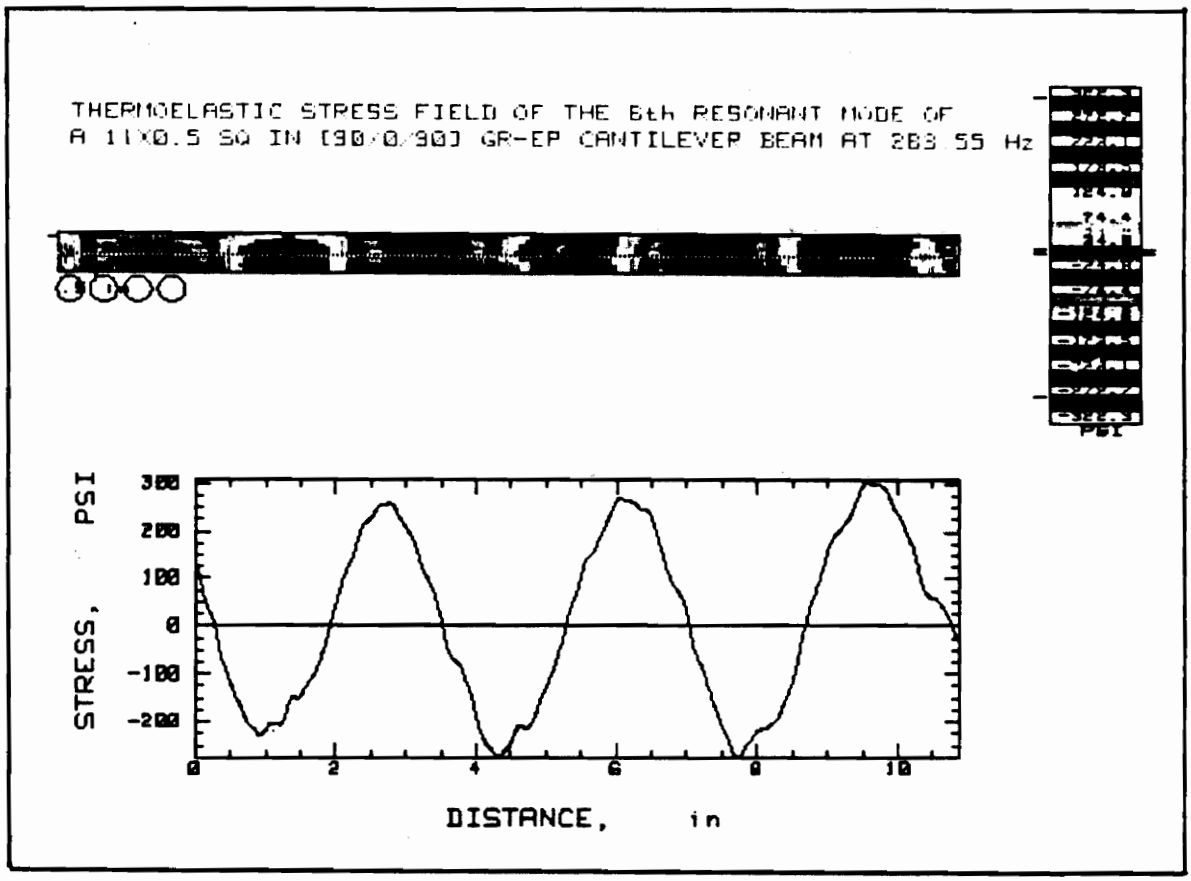

Figure 22. A SPATE line scan along the damaged (90/0/90) beam specimen at 283.55 Hz.

Figure 23 presents a SPATE line scan at 13.5 kHz. The bottom part of the figure shows the infrared signal received from the undamaged (90/0/90) beam, while the top part displays the corresponding signal of the damaged one. The first scan indicates that near the flaw location, the SPATE camera detected a response which was out of phase with the rest of the beam, suggesting that either one of the local subplates is resonating independently of the rest of the structure[17].

Figure 24 shows the thermoelastic signal of the damaged beam at two different frequencies, namely 13.5 and 14.5 kHz. At 14.5 kHz (Figure 24.a), the second simulated delamination is excited and again an out of phase signal was detected by the SPATE camera. The response also indicates that the second delamination is located approximately 4 inches to the right of the clamped end and is smaller in size than the one being closer to the right free end. This was indeed the case. Note that all of the above line scans were obtained in less than 5 seconds.

The 'local resonance model' suggests than even when the structural component is relatively thick, heat will be generated if the local subplates -formed as a result of delamination presence- resonate locally. However, heat may be conducted through the material, resulting in a weak thermographic signal. If this is the case, the SPATE technique can be of great importance. Vibrothermography and SPATE can be combined together to form a very powerful NDE package for composite materials and structures.

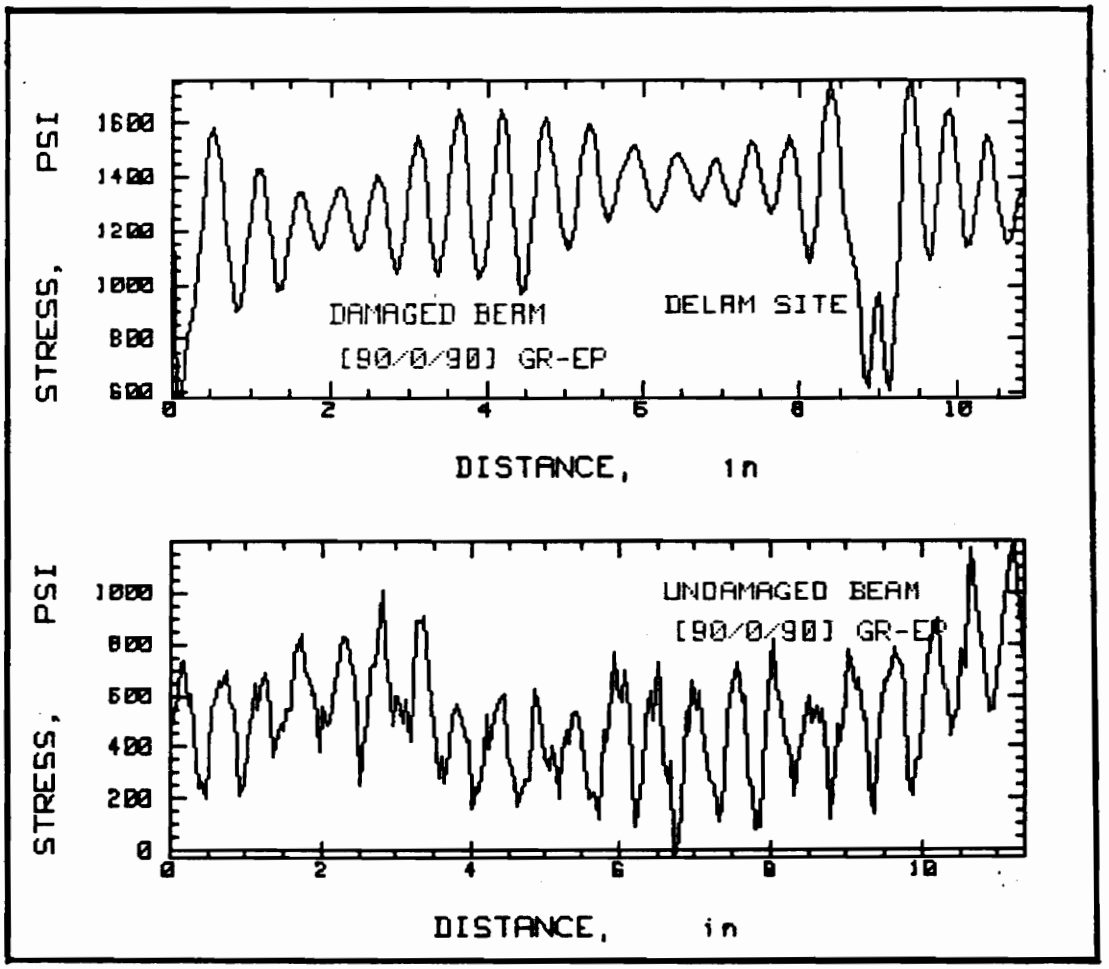

Figure 23. (a) SPATE signal of the delaminated (90/0/90) beam.

(b) Infrared signal of the damage-free beam at 13.5 kHz;

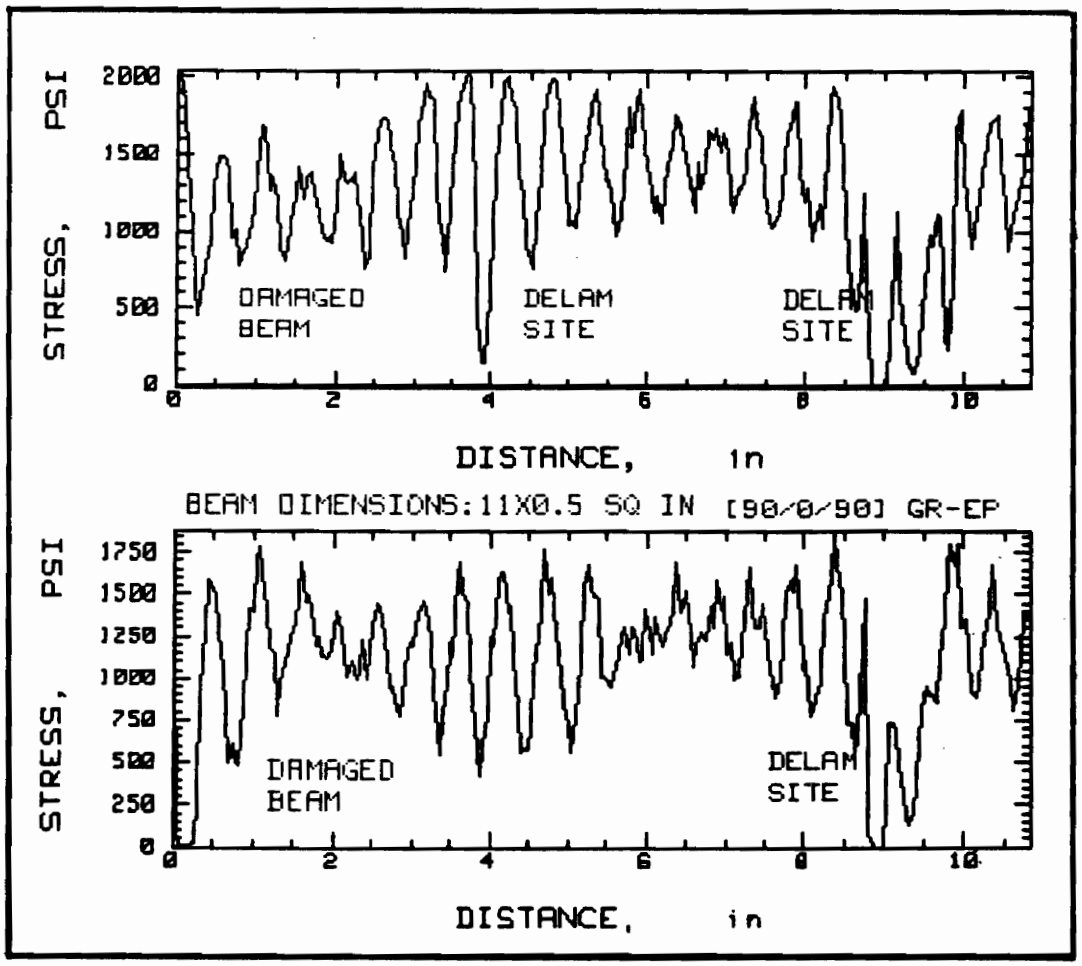

Figure 24. (a) Infrared Signal at 14.5 kHz. The smaller flaw is now getting excited; (b) SPATE signal at 13.5 kHZ. Only the biger flaw is resonating locally.

4.4 Mechanisms for Energy Dissipation into Heat

When an outside source of work is performed on a body deformation results. In an elastic body potential energy (strain energy) is created, which will be recovered upon unloading. In a viscoelastic material - such as a composite - part of this work may be lost in the sense that it is transformed into heat and therefore is not recoverable. Such energy is called *dissipated*[11].

Lazan[12] discussed a number of mechanisms responsible for energy dissipation into heat in metals and polymers. He also presented experimental data reporting the variation of *loss coefficient n* (*n* is expressed as the imaginary over the real part of the modulus) with temperature and frequency.

He categorized damping mechanisms as follows :

A. Linear Damping Associated with Anelastic Mechanisms in Metals

(1) Relaxation Peaks.

(2) Point Defect Relaxations.

(3) Grain Boundary Viscosity.

(4) Macro-thermoelasticity.

(5) Micro-thermoelasticity.

(6) Eddy-current effects.

(7) Other Linear Anelastic Mechanisms.

B. Linear Damping Associated with Dislocation Mechanisms in Metals

(1) Vibrating String Dislocation Model.

(2) Bordoni Peaks.

C Damping in Polymers and Other Nonmetallic Materials

(1) Mechanical Dispersion.

Many composites have polymeric matrices possesing viscoelastic characteristics. Lazan proposed that several types of dispersion phenomena are observed in polymers including optical, dielectric, and mechanical. **Mechanical dispersion** effects are directly related to the strong frequency and temperature effects observed in polymers. As a result of mechanical dispersion the following features are observed over a wide range of frequencies and temperatures.

(a) The elastic and complex modulus (for example, G', G^*) increase rather rapidly in a **critical region** as either the frequency is increased or the temperature is decreased.

(b) The loss modulus and loss coefficient (for example G'' and n pass through a **maximum** in this critical region.

At low frequencies (or at high temperature) a "**rubbery**" region is observed in which relaxation processes follow the slow change in stress **(in phase)** resulting in an equilibrium condition not conducive to energy dissipation. Thus, in the "rubbery" region, all moduli and damping values are relatively low. At intermediate frequencies and temperatures there is a "**transition**" region in which the loss modulus is largest. In

this region a variety of dispersion and mobility effects are observed. As a result of this spectrum of processes a broad resonance type curve not dissimilar to that observed in anelastic behavior is produced. **Flow and diffusion processes are generally out of phase with the cyclic stress, and the resulting phase lag in the cyclic strain provides a mechanism for dissipating damping energy.** At very high frequencies (or at low temperature) the storage modulus is large, the loss modulus is quite small, and the behavior resembles that of a stiff *"elastic"* material. This is known as the "*glassy*" region where various types of molecular mobilities are gradually "frozen in". In terms of anelasticity nomenclature the diffusion and relaxation processes cannot occur rapidly enough to follow the stress, and the material behaves essentially "elastically". Lazan also presented data for "typical elastomers", polymers and plastics, in order to indicate general levels of damping in the transition region.

As the transitional frequency is approached, the damping properties are governed by the chain segments in the amorphous (non crystalline) phase of the polymer. The chain segment, which is much smaller than the macromolecule but large compared to the chain length of the monomer group, undergoes coiling and uncoiling motions. In this region the glass-transition dispersion micromechanism produces high damping over a broad range of frequencies.

Lee[13] presented a general lecture in Thermo-Viscoelasticity, and also discussed vibration problems. He stated that for oscillatory loading, the continuous dissipation of mechanical work into heat due to

viscoelasticity can cause an appreciable temperature increase, with a corresponding effect on the stress distribution due to sensitivity of viscoelastic characteristics to temperature change. He also presented the heat balance equation in one-dimensional form and found that the work dissipation term dominates thermo-mechanical coupling effects since it is cumulative in time because of the **phase difference** between the stress and strain oscillations. He concluded that the other terms in the heat equation cause a small temperature oscillation during each vibration cycle. Due to viscoelastic attenuation, peak stresses and hence peak dissipation occur adjacent to the surfaces where the force is applied or near defect areas with high stress concentrations, and the resulting heating can soften the material there and hence further localize the dissipation and cause marked temperature rise.

Ting, Yuan and Wang[14] discussed thermal effects in viscoelasticity due to thermo-mechanical coupling. In their paper they gave numerical results for a vibrating viscoelastic rod. They indicated that at $\omega = 10^4$ rad/sec (1.592 kHz) the temperature rise in the rod can reach 130^0 F in only 4 hours. They also found that *for higher frequencies, the temperature increases more rapidly.*

By relating now all the above observations with the experimental results reported in the present study, some reasonable explanations about the frequency dependent heat generation mechanisms can be given. We recall that no thermal patterns appeared in the 0-10 kHz frequency range for all specimens tested. It is thus assumed that in this region there is no phase lag between the stresses and the strains. As we

proceed to higher frequencies, a transitional frequency occurs and a phase difference between the stresses and the strains takes place. **Mechanical dispersion** and **thermomechanical behavior** (interaction between the mechanical energy and heat), convert mechanical energy into heat. That was most likely the heat generation mechanism for the graphite and glass epoxy specimens. It appears that the viscoelastic nature of the epoxy matrix material produces significant damping in the frequency transitional region. At defect areas such as delaminations, stress concentrations cause peak dissipation and local temperature rise. For the ceramic specimens other mechanisms such as internal micromolecular friction may contribute to a slight heat increase observed in Figure 17. It is unlikely that mechanical dispersion primarily produces damping for this type of material.

4.5 A Three Dimensional Expression for Energy Dissipated per Unit Time.

Modeling a single bar in tension and assuming oscillatory motion, the amount of energy dissipated per unit time is given by the simple relation

$$W^d = \frac{1}{2} \sigma^2 \omega D''(\omega) \qquad (4.1)$$

where

σ : max normal stress

ω : frequency

$D''(\omega)$: Loss compliance

For a three dimensional state of stress the rate of work done is given by the relation

$$W = \sigma_x \dot{\varepsilon}_x + \sigma_y \dot{\varepsilon}_y + \sigma_z \dot{\varepsilon}_z + \tau_{yz} \dot{\gamma}_{yz} + \tau_{xz} \dot{\gamma}_{xz} + \tau_{xy} \dot{\gamma}_{xy}$$

Assuming oscillatory stresses, introducing the stress-strain law, integrating over one period and applying symbolic computational algebra we get a three dimensional expression for the dissipated energy

$$W^d = \frac{1}{2} \omega \left(D''_{11} \sigma_x^2 + D''_{22} \sigma_y^2 + D''_{33} \sigma_z^2 \right) + D''_{26} \omega \sigma_y \sigma_z + \quad (4.2)$$
$$D''_{16} \omega \sigma_x \sigma_z + D''_{12} \omega \sigma_x \sigma_y + D''_{44} \omega \tau_{yz}^2 +$$
$$D''_{55} \omega \tau_{xz}^2 + D''_{66} \omega \tau_{xy}^2$$

where

W^d : energy dissipated per unit time

ω : angular frequency

σ_{ii} : normal components of stress tensor, i=x,y,z

τ_{ij} : off-diagonal terms of stress tensor i,j=1,2,6

$D''_{ij}(\omega)$: Loss compliance tensor, i,j = 1,2,6

As expression (4.2) indicates, the energy dissipated per unit time is directly related to the prescribed stress state of the material at a particular frequency. It also depends on the loss compliance tensor. For composite materials D''_{11} is very close to zero, since the fibers

induce very little damping (if σ_x obtains a high value, the term is still important). Also for thin laminates (where Kirchoff's hypothesis is valid) the shearing stresses -especially those out of plane- are of very small magnitude and can be neglected. In Chapter 2 it was found than no significant shearing stress state exists when local resonance occurs. Thus equation (4.2) can be simplified as follows

$$W^d = \frac{1}{2}\,\omega\,\left(D''_{22}\,\sigma_y^2 + D''_{33}\,\sigma_z^2 \right) + D''_{26}\,\omega\,\sigma_y\,\sigma_z$$

Note that this relation is valid in the layer material coordinate system. For laminates composed of many layers all quantities must transformed to a global XYZ coordinate system.

When local resonance occurs, normal stresses peak. At the transitional frequency stresses and strains are out of phase, and the loss compliance entries become maximum. At this time a significant amount of mechanical energy is dissipated into heat. As a result, the easy, nondestructive detection of flaws follows.

5. CONCLUSIONS AND DISCUSSION

The analytical and experimental study presented in this work, numerically justified the **'Local Resonance Principle'** and developed it further. Local sub-plates formed as a result of delamination presence, resonate locally, independently, and out of phase with the rest of the structure at particular mechanical excitation frequencies. While this depends strictly on the delamination geometry, it was found that as a result of this dynamic behavior, heat will be generated only after the excitation exceeded approximately 10 kHz. **Mechanical dispersion** and **thermomechanical coupling** are major dissipative mechanisms contributing to mechanical energy dissipation into heat. Two NDE techniques, namely **Vibrothermography** and **SPATE**, were used to verify the analytical predictions. These methods are responsive to this type of dynamic flaw behavior, and nondestructively detect and evaluate damage. The best **frequency region** for infrared thermographic analysis was identified; the excitation frequency must reach approximately 10 kHz. Relating the excitation frequency with plate geometry, information about the damage area and depth can be obtained. The combination of Vibrothermography and SPATE can result in a very powerful NDE package. It's biggest advantage over other NDE methods are the significantly reduced time required for inspection, the 'in situ' application of the method, the capability of examining large structural areas, and the performance of the method under relatively adverse conditions. The performed analysis

also shows that based on the 'local resonance model', the SPATE technique will also be capable of revealing delaminations in relatively thick composite structural components.

REFERENCES

1. Reifsnider, K.L., Henneke, E.G. II, and Stinchcomb, W.W., "The Mechanics of Vibrothermography," *The Mechanics of Nondestructive Testing,* W.W. Stinchcomb, Ed., Plemum Press, New York, 1980, pp.149-276.

2. Russell, S.S., "An Investigation of the Excitation Frequency Dependent Behavior of Fiber-Reinforced Epoxy Composites during Vibrothermographic Inspection," Ph.D Dissertation, Virginia Polytechnic Institute and State University, VA., U.S.A., Nov. 1982.

3. Lin, S.S., "Frequency Dependent Heat Generation During Vibrothermographic Testing of Composite Materials," Ph.D Dissertation, Virginia Polytechnic Institute and State University, VA., U.S.A., Aug. 1987.,

4. Duvaut, G., and Lions, L., *Inequalities in Mechanics and Physics,* Springer-Verlag, Berlin, 1976.

5. Zinkiewicz, O., and Hinton, E., "Reduced integration, functions smoothing and non-conformity in finite element analysis (with special reference to thick plates)", *J. Franklin Institute* 302, 443-461 (1976).

6. Kamoulakos, A., "Understanding and improving the reduced integration of Midlin shell elements", *Int. J. Numer. Meth. Engrg* 26, 2009-2029 (1988).

7. Bathe, K.J., and Wilson, E., *Numerical Methods in Finite Element Analysis*, Prentice-Hall, Eaglewood Cliffs, 1976.

8. Thangitham, S., and Laterman, G., "Experimental and Analytical Study of the Dynamic Behavior of Cantilever Composite Plates", Blacksburg, VA: Virginia Polytechnic Institute and State University Center for Composite Materials and Structures (1990).

9. Tenek, L.H., Henneke, E.G., and Gunzburger, M.D., "Vibration of Delaminated Anisotropic Composite Plates", IMAC 9, Florence, Italy, April 1991.

10. Tenek, L.H., Henneke, E.G., and Gunzburger, M.D., "Local, Independent Vibrations of Delaminations in Anisotropic Composite Plates", to appear.

11. Flügge, W., *Viscoelasticity*, Springer-Verlag, Berlin, 1975.

12. Lazan B.J., *Damping Materials and Members in Structural Mechanics*, Pergamon, New York, 38-55, (1968).

13. Lee E.H.,"Thermo-Viscoelasticity", in the *Mechanics of Visco - Elastic Media and Bodies* (ed.by J.Hult) Springer-Verlag, Gothenburg/Sweden, 339-356 (1974).

14. Ting, E.C., Yuan H.L., and Wang, H.C., "Thermal Effects in Viscoelasticity due to Thermo-Mechanical Coupling", in the *Mechanics of Visco - Elastic Media and Bodies* (ed.by J.Hult) Springer-Verlag, Gothenburg/Sweden, 358-374 (1974).

15. N.F.Enke,"Fundamentals of Thermographic Stress Analysis", in the *Theory and Applications of Thermographic Stress Analysis* (ed. by N.F.Enke), Workshop Notes for ASTM Committee Weak, Atlanta, 2.1-2.7 (1988).

16. D.Zhang,N.Enke and B.Sandor,"Thermographic Stress Analysis of Composite materials", in the *Theory and Applications of Thermographic Stress Analysis* (ed. by N.F.Enke),Workshop Notes for ASTM Committee Weak, Atlanta, 5.1-5.7 (1988).

17. Tenek, L.H., and Henneke, E.G., "Flaw Dynamics and Vibrothermographic-Thermoelastic NDE of Advanced Composite Meterials", Thermosense XIII, Orlando, Florida, April 1991.

APPENDIX

FINITE ELEMENT COMPUTER PROGRAM FOR CALCULATING THE NATURAL FREQUENCIES OF LAMINATED AND DELAMINATED COMPOSITE PLATES

```
@PROCESS DC(STMAS)
      PROGRAM FEMVS
*=====================================================================
*                                                                     =
*  F I N I T E   E L E M E N T                    V I B R A T I O N   =
*  A N A L Y S I S   O F  L A M I N A T E D  & DELAMINATED            =
*           C O M P O S I T E   P L A T E S                           =
*=====================================================================
*      L A Z A R U S   H   T E N E K                                  =
*      V.T. 1990                                                      =
*=====================================================================
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER(IVDELAM=0,ISDELAM=0,NDELL=0,NODEL= 0,
     .          NX=18,NY=18,NZ=11,XL= 8.0D0,YL= 4.0D0,MODN=4,
     .          ZL=0.00605D0,NNE=8,NQ=8,MATE=2,ALPHA=2.D0*2000.0D0,
     .          IFLAG=3,IBOUND=2,IVEC=0,IPRVEC=26,IZOLINE=0,
     .          NROOT=30,RTOL=1.D-6,NC=NROOT,IGRAPH=0,ISTRES=0,
     .          NNC=NC*(NC+1)/2,LYNUM=1,
     .          NITEM=16,IFSS=0,IFPR=0)
      DIMENSION A(1000000),B(1000000)
      COMMON / STMAS / A,B
      INTEGER    MAXA(3*(NX*NY*NZ+NODEL)+1),LB(3*(NX*NY*NZ+NODEL))
      DIMENSION  TT(3*(NX*NY*NZ+NODEL))
      DIMENSION  W(3*(NX*NY*NZ+NODEL))
      DIMENSION  D(NC),VEC(NC,NC),AR(NNC),BR(NNC)
      DIMENSION  RTOLV(NC),BUP(NC),BLO(NC),BUPC(NC)
      DIMENSION  XN(NX*NY*(NZ+NDELL)),YN(NX*NY*(NZ+NDELL)),
     .           ZN(NX*NY*(NZ+NDELL))
      DIMENSION  NODE((NX-1)*(NY-1)*(NZ-1),8),
     .           INDX(NX*NY*(NZ+NDELL),3)
      DIMENSION  XQ(NQ),YQ(NQ),ZQ(NQ)
      DIMENSION  EIGV(NC)
      DIMENSION  R(3*(NX*NY*NZ+NODEL),NC)
      DIMENSION  OMEGA(NC)
      DIMENSION  DEGR(NZ-1),THETA(NZ-1)
      INTEGER    KDC(NZ),XDST(NDELL+1),XDEN(NDELL+1)
      INTEGER    YDST(NDELL+1),YDEN(NDELL+1)
      DIMENSION  ESTR((NX-1)*(NY-1)*(NZ-1),6)
      DIMENSION  F(3*(NX*NY*NZ+NODEL)),V(3*(NX*NY*NZ+NODEL))
      DIMENSION  VNEW(3*(NX*NY*NZ+NODEL))

      DO 5 II = 1,NZ-1
      DEGR(II) = 0.D0
5     CONTINUE

      DEGR(1) =  0.0D0
      DEGR(2) = 90.0D0
      DEGR(3) = 90.0D0
      DEGR(4) =  0.0D0
      DATA XDST /  3,0 /
      DATA XDEN /  5,0 /
      DATA YDST /  3,0 /
      DATA YDEN /  5,0 /
      DO 3000 ID = 0,NZ-1
        KDC(ID) = 0
```

```
3000  CONTINUE
                KDC(3)=1
      IF(IFLAG.EQ.1) THEN
      DEN = 1.467D-4
      ELSEIF(IFLAG.EQ.2) THEN
      DEN = 1.4643D-4
      ELSEIF(IFLAG.EQ.3) THEN
      DEN = 1.4752D-4
      ELSEIF(IFLAG.EQ.4) THEN
      DEN = 1.D0
      ELSE
      DEN = 1.97D-4
      ENDIF
      PII = 3.141592654D0
      IF(IVDELAM.EQ.0)THEN
          CALL VPERFP(LYNUM,NX,NY,NZ,XL,YL,ZL,NNE,NQ,
     .                MATE,ALPHA,IFLAG,IBOUND,DEGR,DEN,IGRAPH,
     .     .          XN,YN,ZN,NODE,INDX,A,B,XQ,YQ,ZQ,
     .                EIGV,R,OMEGA,THETA,ISTRES,MODN,PII,
     .                IVEC,IPRVEC,
     .                NROOT,RTOL,NC,NNC,NITEM,IFSS,IFPR,MAXA,
     .                TT,W,D,VEC,AR,BR,RTOLV,BUP,BLO,BUPC,LB)
      ELSE
          CALL VDELCP(IVDELAM,LYNUM,NDELL,NODEL,NX,NY,NZ,DEGR,
     .                XL,YL,ZL,NNE,NQ,MATE,ALPHA,IFLAG,IBOUND,
     .                XDST,XDEN,YDST,YDEN,KDC,DEN,PII,IGRAPH,
     .                XN,YN,ZN,NODE,INDX,A,B,XQ,YQ,ZQ,
     .                EIGV,R,OMEGA,THETA,ISTRES,MODN,
     .                IVEC,IPRVEC,ESTR,
     .                NROOT,RTOL,NC,NNC,NITEM,IFSS,IFPR,MAXA,
     .                TT,W,D,VEC,AR,BR,RTOLV,BUP,BLO,BUPC,LB)
      ENDIF
      END

      SUBROUTINE VPERFP(LYNUM,NX,NY,NZ,XL,YL,ZL,NNE,NQ,MATE,
     .                  ALPHA,IFLAG,IBOUND,DEGR,DEN,IGRAPH,
     .                  XN,YN,ZN,NODE,INDX,A,B,XQ,YQ,ZQ,
     .                  EIGV,R,OMEGA,THETA,ISTRES,MODN,PII,
     .                  IVEC,IPRVEC,
     .                  NROOT,RTOL,NC,NNC,NITEM,IFSS,IFPR,MAXA,
     .                  TT,W,D,VEC,AR,BR,RTOLV,BUP,BLO,BUPC,LB)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A(1000000),B(1000000)
      DIMENSION XN(NX*NY*NZ),YN(NX*NY*NZ),
     .          ZN(NX*NY*NZ)
      DIMENSION NODE((NX-1)*(NY-1)*(NZ-1),8),
     .          INDX(NX*NY*NZ,3)
      DIMENSION XQ(NQ),YQ(NQ),ZQ(NQ)
      DIMENSION DEGR(NZ-1),THETA(NZ-1)
      DIMENSION EIGV(NC),CB(6,6)
      DIMENSION R(3*NX*NY*NZ,NC)
      DIMENSION OMEGA(NC)
      INTEGER    MAXA(3*(NX*NY*NZ)+1),LB(3*(NX*NY*NZ))
      DIMENSION  TT(3*(NX*NY*NZ))
      DIMENSION  W(3*(NX*NY*NZ))
```

```
3DFEM    FORTRAN  A1 11/19/90 22:43    F 80  2264 RECS 11/19/90 22:44 PAGE

      DIMENSION  D(NC),VEC(NC,NC),AR(NNC),BR(NNC)
      DIMENSION  RTOLV(NC),BUP(NC),BLO(NC),BUPC(NC)
      CALL GEOM(LYNUM,NODE,INDX,XN,YN,ZN,NE,NN,NU,ISDELAM,ISTATIC,
     +          NX,NY,NZ,XL,YL,ZL,NNE,LEPL,DEGR,MATE,IBOUND)
      CALL ASEMBL(A,B,XN,YN,ZN,NODE,INDX,NNE,NE,NN,
     .            NQ,LYNUM,NX,NY,NZ,DEGR,LEPL,XQ,YQ,ZQ,
     .            THETA,MATE,DEN,ANG,IFLAG,NU,IBOUND,
     .            NWK,NWM,MAXA,LB)
       CALL SSPACE(A,B,MAXA,R,EIGV,TT,W,AR,BR,VEC,D,RTOLV,
     .             BUP,BLO,BUPC,NU,NU+1,NWK,NWM,NRCOT,RTOL,
     .             NC,NNC,NITEM,IFSS,IFPR,9,IOUT)
      DO 85 I=1,NC
        OMEGA(I)=(DSQRT(EIGV(I))/(2.D0*PII))
85    CONTINUE
      CALL GVREST(OMEGA,NU,NC,EIGV,R,IVEC,IPRVEC,IZOLINE)
      IF(IGRAPH.EQ.0) GOTO 4444
*     IF USE THESE  ADD FILEDEF DISK 30 NODEXYZ DATA A IN EXEC  *
*4606  WRITE(30,*) NN
*       DO 4600 I=1,NN
*4600      WRITE(30,4001) I,XN(I),YN(I),ZN(I)*100.D0
4001  FORMAT(3X,I5,3F18.4)
4646  WRITE(31,*) NE
        DO 4650 I=1,NE
4650      WRITE(31,4002) I,(NODE(I,K),K=1,8)
4002  FORMAT(2X,I5,8I5)
      WRITE(42,*) NN
      DO 4700 I = 1,NN

      IF ( INDX(I,1).EQ.0.AND.INDX(I,2).EQ.0.AND.
     .     INDX(I,3).EQ.0 ) THEN
              XXX     = 0.D0 + XN(I)
              YYY     = 0.D0 + YN(I)
              ZZZ     = 0.D0 + ZN(I)
      ELSE
          XXX     = R(INDX(I,1),IPRVEC) + XN(I)
          YYY     = R(INDX(I,2),IPRVEC) + YN(I)
          ZZZ     = R(INDX(I,3),IPRVEC) + ZN(I)
      ENDIF

      WRITE(42,4001) I,XXX,YYY,ZZZ/10.D0

4700  CONTINUE

4444  RETURN
      END
      SUBROUTINE VDELCP(IVDELAM,LYNUM,NDELL,NODEL,NX,NY,NZ,DEGR,
     .                  XL,YL,ZL,NNE,NQ,MATE,ALPHA,IFLAG,IBOUND,
     .                  XDST,XDEN,YDST,YDEN,KDC,DEN,PII,IGRAPH,
     .                  XN,YN,ZN,NODE,INDX,A,B,XQ,YQ,ZQ,
     .                  EIGV,R,OMEGA,THETA,ISTRES,MODN,
     .                  IVEC,IPRVEC,ESTR,
     .                  NROOT,RTOL,NC,NNC,NITEM,IFSS,IFPR,MAXA,
     .                  TT,W,D,VEC,AR,BR,RTOLV,BUP,BLO,BUPC,LB)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A(1000000),B(1000000)
```

```
      DIMENSION XN(NX*NY*(NZ+NDELL)),YN(NX*NY*(NZ+NDELL)),
     .          ZN(NX*NY*(NZ+NDELL))
      DIMENSION NODE((NX-1)*(NY-1)*(NZ-1),8),
     .          INDX(NX*NY*(NZ+NDELL),5)
      DIMENSION XQ(NQ),YQ(NQ),ZQ(NQ)
      DIMENSION DEGR(NZ-1),THETA(NZ-1)
      DIMENSION UU(8),VV(8),WW(8),ESTR((NX-1)*(NY-1)*(NZ-1),6)
      INTEGER KDC(NZ),XDST(NDELL+1),XDEN(NDELL+1)
      INTEGER YDST(NDELL+1),YDEN(NDELL+1)
      DIMENSION EIGV(NC)
      DIMENSION R(3*(NX*NY*NZ+NODEL),NC)
      DIMENSION OMEGA(NC),CB(6,6)
      INTEGER    MAXA(3*(NX*NY*NZ+NODEL)+1),LB(3*(NX*NY*NZ+NODEL))
      DIMENSION  TT(3*(NX*NY*NZ+NODEL))
      DIMENSION  W(3*(NX*NY*NZ+NODEL))
      DIMENSION  D(NC),VEC(NC,NC),AR(NNC),BR(NNC)
      DIMENSION  RTOLV(NC),BUP(NC),BLO(NC),BUPC(NC)
      CALL*GGEOM(IVDELAM,NDELL,NODEL,LYNUM,NODE,INDX,XN,YN,ZN,NE,NN,NU,
     .           NX,NY,NZ,XL,YL,ZL,NNE,LEPL,DEGR,MATE,IBOUND,KDC,
     .           XDST,XDEN,YDST,YDEN,ISDELAM,ISTATIC)
      CALL AASEMB(A,B,XN,YN,ZN,NODE,INDX,NNE,NE,NN,NDELL,
     .            NQ,LYNUM,NX,NY,NZ,DEGR,LEPL,XQ,YQ,ZQ,
     .            THETA,MATE,DEN,ANG,IFLAG,NU,IBOUND,
     .            NNK,NWM,MAXA,LB)
       CALL SSPACE(A,B,MAXA,R,EIGV,TT,W,AR,BR,VEC,D,RTOLV,
     .             BUP,BLO,BUPC,NU,NU+1,NNK,NWM,NRCOT,RTOL,
     .             NC,NNC,NITEM,IFSS,IFPR,9,IOUT)
      DO 85 I=1,NC
        OMEGA(I)=(DSQRT(EIGV(I))/(2.D0*PII))
 85   CONTINUE
      CALL GVREST(OMEGA,NU,NC,EIGV,R,IVEC,IPRVEC,IZ0LINE)
      IF(ISTRES.EQ.0) GOTO 3606
      DO 5000 IT = 1,NE
                 IF (MATE.EQ.2) THEN
                    KK = 1 + INT((IT-1)/LEPL)
                    TH = THETA(KK)
                    CALL CBAR3D(IT,LYNUM,TH,CB,NE,NX,NY,NZ,MATE,IFLAG)
                 ELSE
                    CALL CBAR3D(IT,LYNUM,0,CB,NE,NX,NY,NZ,MATE,IFLAG)
                 ENDIF
        DO 5010 IN = 1,NNE
                                    IP = NODE(IT,IN)
                                    HX = (XL/(NX-1.D0))
                                    HY = (YL/(NY-1.D0))
                                    HZ = (ZL/(NZ-1.D0))
      IF (INDX(IP,1).EQ.0 .AND. INDX(IP,2).EQ.0 .AND.
     .    INDX(IP,3).EQ.0 ) THEN
               UU(IN) = 0.D0
               VV(IN) = 0.D0
               WW(IN) = 0.D0
      ELSE
           UU(IN) = R(INDX(IP,1),IPRVEC)
           VV(IN) = R(INDX(IP,2),IPRVEC)
           WW(IN) = R(INDX(IP,3),IPRVEC)
      ENDIF
```

3DFEM FORTRAN A1 11/19/90 22:43 F 80 2264 RECS 11/19/90 22:44 PAGE

```
5010    CONTINUE
      AVU11 =(UU(2)-UU(1)+UU(3)-UU(4)+UU(6)-UU(5)+UU(7)-UU(8))/(HX*4.D0)
      AVU22 =(VV(5)-VV(1)+VV(6)-VV(2)+VV(8)-VV(4)+VV(7)-VV(3))/(HY*4.D0)
      AVU33 =(WW(4)-WW(1)+WW(3)-WW(2)+WW(8)-WW(5)+WW(7)-WW(6))/(HZ*4.D0)
      AVU12 =(UU(5)-UU(1)+UU(6)-UU(2)+UU(8)-UU(4)+UU(7)-UU(3))/(HY*4.D0)
      AVU13 =(UU(4)-UU(1)+UU(3)-UU(2)+UU(8)-UU(5)+UU(7)-UU(6))/(HZ*4.D0)
      AVU23 =(VV(4)-VV(1)+VV(3)-VV(2)+VV(8)-VV(5)+VV(7)-VV(6))/(HZ*4.D0)
      ESTR(IT,1) = CB(1,1)*AVU11+CB(1,2)*AVU22+CB(1,3)*AVU33+
     .             CB(1,4)*AVU23+CB(1,5)*AVU13+CB(1,6)*AVU12
      ESTR(IT,2) = CB(2,1)*AVU11+CB(2,2)*AVU22+CB(2,3)*AVU33+
     .             CB(2,4)*AVU23+CB(2,5)*AVU13+CB(2,6)*AVU12
      ESTR(IT,3) = CB(3,1)*AVU11+CB(3,2)*AVU22+CB(3,3)*AVU33+
     .             CB(3,4)*AVU23+CB(3,5)*AVU13+CB(3,6)*AVU12
      ESTR(IT,4) = CB(4,1)*AVU11+CB(4,2)*AVU22+CB(4,3)*AVU33+
     .             CB(4,4)*AVU23+CB(4,5)*AVU13+CB(4,6)*AVU12
      ESTR(IT,5) = CB(5,1)*AVU11+CB(5,2)*AVU22+CB(5,3)*AVU33+
     .             CB(5,4)*AVU23+CB(5,5)*AVU13+CB(5,6)*AVU12
      ESTR(IT,6) = CB(6,1)*AVU11+CB(6,2)*AVU22+CB(6,3)*AVU33+
     .             CB(6,4)*AVU23+CB(6,5)*AVU13+CB(6,6)*AVU12
5000  CONTINUE

1111  FORMAT(1X,'SXX', 9X,'SYY', 9X,'SZZ', 9X,'T23',10X,'T13', 9X,'T12')
1113  FORMAT(6(D10.2,2X))
      WRITE(*,*) ' '
          WRITE(39,*) NX-1,NY-1
         LAYER = 4
         ISTR  = 3
         LK = ((LAYER-1)*LEPL)+1
         DO 6001 I = LK,LK+NX-2
            DO 6003 J = 0,NY-2

            KK = I + (NX-1)*J
            XC = (XN(NODE(KK,7))+XN(NODE(KK,1)))/2.D0
            YC = (YN(NODE(KK,7))+YN(NODE(KK,1)))/2.D0
            WRITE(39,6007) XC,YC,ESTR(KK,ISTR)/1.D6
6003       CONTINUE
6001     CONTINUE
6007     FORMAT(3(F16.4,4X))
3606  IF(IGRAPH.EQ.0) GOTO 4444
*4606  WRITE(30,*) NN
*        DO 4600 I=1,NN
*4600      WRITE(30,4001) I,XN(I),YN(I),ZN(I)*100.D0
4001  FORMAT(3X,I5,3F18.4)
4646  WRITE(31,*) NE
        DO 4650 I=1,NE
4650      WRITE(31,4002) I,(NODE(I,K),K=1,8)
4002  FORMAT(2X,I5,8I5)
      WRITE(42,*) NN
      DO 4700 I = 1,NN
      IF ( INDX(I,1).EQ.0.AND.INDX(I,2).EQ.0.AND.
     .    INDX(I,3).EQ.0 ) THEN
            XXX     = 0.D0 + XN(I)
            YYY     = 0.D0 + YN(I)
            ZZZ     = 0.D0 + ZN(I)
      ELSE
```

```
            XXX     = R(INDX(I,1),IPRVEC) + XN(I)
            YYY     = R(INDX(I,2),IPRVEC) + YN(I)
            ZZZ     = R(INDX(I,3),IPRVEC) + ZN(I)
         ENDIF
         WRITE(42,4001) I,XXX,YYY,ZZZ/100.D0
 4700    CONTINUE
 4444    RETURN
         END
         SUBROUTINE GEOM(LYNUM,NODE,INDX,XN,YN,ZN,NE,NN,NU,ISDELAM,ISTATIC,
        +                NX,NY,NZ,XL,YL,ZL,NNE,LEPL,DEGR,MATE,IBOUND)
         IMPLICIT DOUBLE PRECISION (A-H,O-Z)
         DIMENSION XN(LYNUM*NX*NY*NZ),YN(LYNUM*NX*NY*NZ),
        +          ZN(LYNUM*NX*NY*NZ)
         DIMENSION NODE(LYNUM*(NX-1)*(NY-1)*(NZ-1),8),
        +          INDX(LYNUM*NX*NY*NZ,3)
         DIMENSION DEGR(NZ-1)
         IF(ISTATIC.EQ.0) THEN
         WRITE(*,*) '  '
         WRITE(*,*)  '=============================================='
         WRITE(*,*)  '=                                            ='
         WRITE(*,*)  '= F I N I T E   E L E M E N T  A N A L Y S I S ='
         WRITE(*,*)  '=          FREE  VIBRATION PROBLEM           ='
         WRITE(*,*)  '=             LAMINATED  PLATE               ='
         WRITE(*,*)  '=                                            ='
         WRITE(*,*)  '=============================================='
         WRITE(*,*) '  '
         ELSEIF(ISTATIC.EQ.1) THEN
         WRITE(*,*) '  '
         WRITE(*,*)  '=============================================='
         WRITE(*,*)  '=                                            ='
         WRITE(*,*)  '= F I N I T E   E L E M E N T  A N A L Y S I S ='
         WRITE(*,*)  '=              STATIC  ANALYSIS              ='
         WRITE(*,*)  '=            LAMINATED  PLATE                ='
         WRITE(*,*)  '=                                            ='
         WRITE(*,*)  '=============================================='
         WRITE(*,*) '  '
         ENDIF

          IF(IBOUND.EQ.1) THEN
             WRITE(*,*) ' FULLY FREE BOUNDARY CONDITIONS '
          ELSEIF(IBOUND.EQ.2) THEN
             WRITE(*,*) ' CLAMPED  FREE FREE FREE BOUNDARY CONDITIONS'
          ELSEIF(IBOUND.EQ.3) THEN
             WRITE(*,*) ' FULLY  CLAMPED  BOUNDARY CONDITIONS'
          ELSEIF(IBOUND.EQ.4) THEN
             WRITE(*,*) ' FULLY  SIMPLY SUPPORTED BOUNDARY CONDITIONS'
          ELSEIF(IBOUND.EQ.5) THEN
             WRITE(*,*) ' CLAMPED FREE CLAMPED CLAMPED BOUNDARY CONDITIONS'
          ELSE
             WRITE(*,*) ' S.SUP FREE S.SUP S.SUP BOUNDARY CONDITIONS'
          ENDIF

         WRITE(*,*) '  '
         WRITE(*,*) 'NUMBER OF ELEMENT NODES      ->',NNE
         WRITE(*,*) 'NUMBER OF LAYERS             ->',NZ-1
```

```
3DFEM    FORTRAN  A1 11/19/90 22:43    F 80  2264 RECS 11/19/90 22:44 PAGE

         WRITE(*,*) 'STRUCTURE LENGTH (IN)        ->',XL
         WRITE(*,*) 'STRUCTURE WIDTH              ->',YL
         WRITE(*,*) 'STRUCTURE HEIGHT             ->',ZL*(NZ-1)
         WRITE(*,*) 'ELEMENTS ACROSS THICKNESS    ->',1
         WRITE(*,*) 'NODES IN X DIRECTION         ->',NX
         WRITE(*,*) 'NODES IN Y DIRECTION         ->',NY
         WRITE(*,*) 'NODES IN Z DIRECTION         ->',NZ
         WRITE(*,*) 'ELEMENTS PER LAYER           ->',(NX-1)*(NY-1)
         WRITE(*,*) 'NODES PER LAYER              ->',NX*NY
         WRITE(*,*) 'MATERIAL TYPE <1>ISO<2>ANISO ->',MATE
         WRITE(*,*) '  '
         IF(MATE.NE.2) GOTO 777
         DO 65 I = 1,NZ-1
         WRITE(*,67) I,DEGR(I)
 67      FORMAT(1X,'LAYER->',I4,2X,'FIBER ORIENTATION',F12.4)
 65      CONTINUE
         WRITE(*,*) '  '
 777     LEPL= (NX-1)*(NY-1)
         IT = 0.D0
         IU = 0.D0
         IP = 0.D0
         ZINCR = 0.D0
         HX = (XL-0.D0)/(NX-1.D0)
         HY = (YL-0.D0)/(NY-1.D0)
         HZ = ZL
C        HZ = (ZL-0.D0)/(NZ-1.D0)
C
*        DO 1025 LY = 1, LYNUM
           DO 1040 JC = 0, NY-1
            DO 1050 IC = 0, NX-1
             DO 1030 KC = 0, NZ-1
*
*
*    CANTILEVER PLATE
*
*
                    IF(IBOUND.EQ.2.AND.IC.EQ.0) GOTO 3100
*
*
*    FULLY CLAMPED
*
*
                    IF(IBOUND.EQ.3.AND.IC.EQ.0) GOTO 3100
                    IF(IBOUND.EQ.3.AND.IC.EQ.(NX-1)) GOTO 3100
                    IF(IBOUND.EQ.3.AND.JC.EQ.0) GOTO 3100
                    IF(IBOUND.EQ.3.AND.JC.EQ.(NY-1)) GOTO 3100
*
*
*    SIMPLY SUPPORTED
*
*
          IF(IBOUND.EQ.4.AND.IC.EQ.0.AND.KC.EQ.0) GOTO 3100
          IF(IBOUND.EQ.4.AND.IC.EQ.(NX-1).AND.KC.EQ.0) GOTO 3100
          IF(IBOUND.EQ.4.AND.JC.EQ.0.AND.KC.EQ.0) GOTO 3100
          IF(IBOUND.EQ.4.AND.JC.EQ.(NY-1).AND.KC.EQ.0) GOTO 3100
```

```
C
C
C   CLAMPED FREE CLAMPED CLAMPED
C
C
                IF(IBOUND.EQ.5.AND.IC.EQ.0) GOTO 3100
                IF(IBOUND.EQ.5.AND.JC.EQ.0) GOTO 3100
                IF(IBOUND.EQ.5.AND.JC.EQ.(NY-1)) GOTO 3100
C
C
C   SIMPLY SUPPORTED FREE SIMPLY SUPPORTED SIMPLY SUPPORTED
C
C
      IF(IBOUND.EQ.6.AND.IC.EQ.0.AND.KC.EQ.0) GOTO 3100
      IF(IBOUND.EQ.6.AND.JC.EQ.0.AND.KC.EQ.0) GOTO 3100
      IF(IBOUND.EQ.6.AND.JC.EQ.(NY-1).AND.KC.EQ.0) GOTO 3100
C
C
C   SIMPLY SUPPORTED CONSTRAINED AT THE MID SURFACE
C
C
      IF(IBOUND.EQ.7.AND.IC.EQ.0.AND.KC.EQ.((NZ-1)/2)) GOTO 3100
      IF(IBOUND.EQ.7.AND.IC.EQ.(NX-1).AND.KC.EQ.((NZ-1)/2)) GOTO 3100
      IF(IBOUND.EQ.7.AND.JC.EQ.0.AND.KC.EQ.((NZ-1)/2)) GOTO 3100
      IF(IBOUND.EQ.7.AND.JC.EQ.(NY-1).AND.KC.EQ.((NZ-1)/2)) GOTO 3100
C
            IU = IU+3.D0
3100        IP = IP+1.D0
             XN(IP) =IC*HX
             YN(IP) =JC*HY
             ZN(IP) =KC*HZ+ZINCR
C
            IF(IBOUND.EQ.2.AND.IC.EQ.0) THEN
             INDX(IP,1) = 0
             INDX(IP,2) = 0
             INDX(IP,3) = 0
            ELSEIF(IBOUND.EQ.3.AND.IC.EQ.0) THEN
             INDX(IP,1) = 0
             INDX(IP,2) = 0
             INDX(IP,3) = 0
            ELSEIF(IBOUND.EQ.3.AND.IC.EQ.(NX-1)) THEN
             INDX(IP,1) = 0
             INDX(IP,2) = 0
             INDX(IP,3) = 0
            ELSEIF(IBOUND.EQ.3.AND.JC.EQ.0) THEN
             INDX(IP,1) = 0
             INDX(IP,2) = 0
             INDX(IP,3) = 0
            ELSEIF(IBOUND.EQ.3.AND.JC.EQ.(NY-1)) THEN
             INDX(IP,1) = 0
             INDX(IP,2) = 0
             INDX(IP,3) = 0
            ELSEIF(IBOUND.EQ.4.AND.IC.EQ.0.AND.KC.EQ.0) THEN
             INDX(IP,1) = 0
             INDX(IP,2) = 0
```

3DFEM FORTRAN A1 11/19/90 22:43 F 80 2264 RECS 11/19/90 22:44 PAGE

```
             INDX(IP,3) = 0
            ELSEIF(IBOUND.EQ.4.AND.IC.EQ.(NX-1).AND.KC.EQ.0) THEN
             INDX(IP,1) = 0
             INDX(IP,2) = 0
             INDX(IP,3) = 0
            ELSEIF(IBOUND.EQ.4.AND.JC.EQ.0.AND.KC.EQ.0) THEN
             INDX(IP,1) = 0
             INDX(IP,2) = 0
             INDX(IP,3) = 0
            ELSEIF(IBOUND.EQ.4.AND.JC.EQ.(NY-1).AND.KC.EQ.0) THEN
             INDX(IP,1) = 0
             INDX(IP,2) = 0
             INDX(IP,3) = 0
            ELSEIF(IBOUND.EQ.5.AND.IC.EQ.0) THEN
             INDX(IP,1) = 0
             INDX(IP,2) = 0
             INDX(IP,3) = 0
            ELSEIF(IBOUND.EQ.5.AND.JC.EQ.0) THEN
             INDX(IP,1) = 0
             INDX(IP,2) = 0
             INDX(IP,3) = 0
            ELSEIF(IBOUND.EQ.5.AND.JC.EQ.(NY-1)) THEN
             INDX(IP,1) = 0
             INDX(IP,2) = 0
             INDX(IP,3) = 0
            ELSEIF(IBOUND.EQ.6.AND.IC.EQ.0.AND.KC.EQ.0) THEN
             INDX(IP,1) = 0
             INDX(IP,2) = 0
             INDX(IP,3) = 0
            ELSEIF(IBOUND.EQ.6.AND.JC.EQ.0.AND.KC.EQ.0) THEN
             INDX(IP,1) = 0
             INDX(IP,2) = 0
             INDX(IP,3) = 0
            ELSEIF(IBOUND.EQ.6.AND.JC.EQ.(NY-1).AND.KC.EQ.0) THEN
             INDX(IP,1) = 0
             INDX(IP,2) = 0
             INDX(IP,3) = 0
      ELSEIF(IBOUND.EQ.7.AND.IC.EQ.0.AND.KC.EQ.((NZ-1)/2)) THEN
             INDX(IP,1) = 0
             INDX(IP,2) = 0
             INDX(IP,3) = 0
      ELSEIF(IBOUND.EQ.7.AND.IC.EQ.(NX-1).AND.KC.EQ.((NZ-1)/2)) THEN
             INDX(IP,1) = 0
             INDX(IP,2) = 0
             INDX(IP,3) = 0
      ELSEIF(IBOUND.EQ.7.AND.JC.EQ.0.AND.KC.EQ.((NZ-1)/2)) THEN
             INDX(IP,1) = 0
             INDX(IP,2) = 0
             INDX(IP,3) = 0
      ELSEIF(IBOUND.EQ.7.AND.JC.EQ.(NY-1).AND.KC.EQ.((NZ-1)/2)) THEN
             INDX(IP,1) = 0
             INDX(IP,2) = 0
             INDX(IP,3) = 0
            ELSE
             INDX(IP,1) = IU-2.D0
```

```
               INDX(IP,2) = IU-1.D0
               INDX(IP,3) = IU
              ENDIF
*
1030      CONTINUE
1050     CONTINUE
1040    CONTINUE
*                                ZINCR = ZL*LY
*1025    CONTINUE
                        NN = IP
       WRITE(*,*)   'TOTAL NODES OF STRUCTURE      ->',NN
                        NU = IU
       WRITE(*,*)   'TOTAL NUMBER OF UNKNOWNS      ->',NU
                               NODINCR = 0

          DO 1060 KC = 0,NZ-2
             DO 1070 JC = 0,NY-2
               DO 1080 IC = 0,NX-2

           IT = IT + 1.D0

                 NODE(IT,1) = (NX*NZ*JC)+(IC*NZ)+1.D0+NODINCR
                 NODE(IT,2) = NODE(IT,1)+NZ
                 NODE(IT,3) = NODE(IT,2)+1.D0
                 NODE(IT,4) = NODE(IT,1)+1.D0
                 NODE(IT,5) = NODE(IT,1)+(NX*NZ)
                 NODE(IT,6) = NODE(IT,5)+NZ
                 NODE(IT,7) = NODE(IT,6)+1.D0
                 NODE(IT,8) = NODE(IT,5)+1.D0

1080        CONTINUE
1070     CONTINUE
                    NODINCR = NODINCR+1.D0
1060     CONTINUE

                        NE = IT
       WRITE(*,*)   'TOTAL ELEMENTS IN STRUCTURE  ->',NE
404    FORMAT(23X,'X',15X,'Y',15X,'Z')
*              WRITE(*,*) '  '
*              DO 400 I = 1,IP
*              WRITE(*,402) I,XN(I),YN(I),ZN(I)
402            FORMAT('NODE:',I5,1X,D16.4,1X,D16.4,1X,D16.4)
*400            CONTINUE
*              WRITE(*,*) '  '
*              WRITE(*,220)
220    FORMAT(12X,'1',6X,'2',6X,'3',6X,'4',6X,'5',6X,'6',6X,'7',6X,'8')
               WRITE(*,*) '  '
            DO 111 I = 1,NE
               I1 = NODE(I,1)
               I2 = NODE(I,2)
               I3 = NODE(I,3)
               I4 = NODE(I,4)
               I5 = NODE(I,5)
               I6 = NODE(I,6)
               I7 = NODE(I,7)
```

3DFEM FORTRAN A1 11/19/90 22:43 F 80 2264 RECS 11/19/90 22:44 PAGE

```
               I8 = NODE(I,8)
*              WRITE(*,222) I,I1,I2,I3,I4,I5,I6,I7,I8
222    FORMAT(I4,'>',2X,I6,1X,I6,1X,I6,1X,I6,1X,I6,1X,I6,1X,I6,1X,I6)
111         CONTINUE
230    FORMAT(17X,'1',8X,'2',7X,'3')
*              WRITE(*,*) '  '
*           DO 115 I = 1,NN
*              J1 = INDX(I,1)
*              J2 = INDX(I,2)
*              J3 = INDX(I,3)
*              WRITE(*,232) I,J1,J2,J3
232    FORMAT('NODE',I5,4X,I5,4X,I5,4X,I5)
115          CONTINUE
C
       RETURN
       END
       SUBROUTINE ASEMBL(A,B,XN,YN,ZN,NODE,INDX,NNE,NE,NN,
      .                  NQ,LYNUM,NX,NY,NZ,DEGR,LEPL,XQ,YQ,ZQ,
      .                  THETA,MATE,DEN,ANG,IFLAG,NU,IBOUND,
      .                  NWK,NWM,MAXA,LB)
       IMPLICIT DOUBLE PRECISION (A-H,O-Z)
       DIMENSION A(1000000),B(1000000)
       INTEGER MAXA(NU+1),LB(NU)
       DIMENSION XN(NX*NY*NZ),YN(NX*NY*NZ),
      +          ZN(NX*NY*NZ)
       DIMENSION NODE((NX-1)*(NY-1)*(NZ-1),8),
      +          INDX(NX*NY*NZ,3)
       DIMENSION XQ(NQ),YQ(NQ),ZQ(NQ)
       DIMENSION DEGR(NZ-1),THETA(NZ-1)
       DIMENSION CB(6,6)
       DO 4500 J = 1,NU
              LB(J) = 0
4500   CONTINUE
               DO 4550 IT = 1,NE

                  DO 4560  IN = 1,NNE

                                       IP = NODE(IT,IN)

                       DO 4570 IU = 1,3

                                       I = INDX(IP,IU)
                          IF(I.EQ.0) GOTO 4570
                        DO 4580 JN = 1,NNE

                                       IPP = NODE(IT,JN)

                            DO 4590 IUK = 1,3

                                       J = INDX(IPP,IUK)
                            IF(J.EQ.0) GOTO 4590
*
                  IJ = J-I
                  IF(IJ.GT.LB(J)) LB(J)=IJ
*
```

```
4590                          CONTINUE
4580                       CONTINUE
4570                    CONTINUE
4560                 CONTINUE
4550              CONTINUE

                  MAXA(1) = 1

                     DO 4600 J = 2,NU

                        MAXA(J) = MAXA(J-1)+LB(J-1)+1
*                       WRITE(*,*) 'LB,MAXA ',LB(J),MAXA(J)
4600                 CONTINUE
                      NWK = MAXA(NU)+LB(NU)
                      NWM = NWK
                      MAXA(NU+1)=NWK+1
            *           WRITE(*,*) '      '
                        WRITE(*,*) 'NWK   ',NWK
                        WRITE(*,*) '      '
         DO 1700 IK = NWK
              A(IK) = 0.D0
              B(IK) = 0.D0
1700     CONTINUE
         CALL LOCCOR(NQ,XQ,YQ,ZQ)
         IF(MATE.EQ.2) THEN
         DO 1980 L = 1,NZ-1
               THETA(L) = (3.14159265358793239D0*DEGR(L))/180.D0
1980     CONTINUE
*           DO 1985 I = 1,LYNUM
*              DO 1990 J = 1,LEPL
*                    ANG(J) = THETA(I)
*1990          CONTINUE
*1985       CONTINUE
         ENDIF
         DO 2000 IT = 1,NE
                                    X1 = XN(NODE(IT,1))
                                    Y1 = YN(NODE(IT,1))
                                    Z1 = ZN(NODE(IT,1))
                                    X7 = XN(NODE(IT,7))
                                    Y7 = YN(NODE(IT,7))
                                    Z7 = ZN(NODE(IT,7))
            IF(MATE.EQ.2) THEN
             KK = 1 + INT((IT-1)/LEPL)
             TH = THETA(KK)
*        WRITE(*,*) 'IT ',IT
*        WRITE(*,*) 'ELE ID (KK) ',KK
*        WRITE(*,*) 'TH IN ASSEMBLY',TH
             CALL CBAR3D(IT,LYNUM,TH,CB,NE,NX,NY,NZ,MATE,IFLAG)
            ELSE
             CALL CBAR3D(IT,LYNUM,0,CB,NE,NX,NY,NZ,MATE,IFLAG)
            ENDIF
                               VOL = (X7-X1)*(Y7-Y1)*(Z7-Z1)
         DO 2020 IQ = 1,NQ
*                           XI = XQ(IQ)
```

3DFEM FORTRAN A1 11/19/90 22:43 F 80 2264 RECS 11/19/90 22:44 PAGE

```
                           EN = YQ(IQ)
                           ZI = ZQ(IQ)
         DO 2040 IN = 1,NNE
                                 IP = NODE(IT,IN)
                           CALL TBF(XI,EN,ZI,IN,BB,BX,BY,BZ,
     +                                X1,Y1,Z1,X7,Y7,Z7)
          DO 2060 IU = 1,3
                                 I = INDX(IP,IU)
                                  IF(I.EQ.0) GOTO 2060
            DO 2070 INN = 1,NNE
                                 IPP = NODE(IT,INN)
                           CALL TBF(XI,EN,ZI,INN,BBB,BBX,BBY,BBZ,
     +                                X1,Y1,Z1,X7,Y7,Z7)
             DO 2080 JU = 1,3
                                 J = INDX(IPP,JU)
                                  IF(I.GT.J) GOTO 2080
         IJ = MAXA(J) + (J-I)
                             BIJ = 0.D0

       IF(IU.EQ.1) THEN
          IF(JU.EQ.1) THEN
           AIJ = BX*(CB(1,1)*BBX+(CB(1,5))*BBZ+(CB(1,6))*BBY)+
     +           BY*((CB(6,1))*BBX+(CB(6,5))*BBZ+(CB(6,6))*BBY)
           BIJ = BB*BBB*DEN
          ELSEIF(JU.EQ.2) THEN
           AIJ = BX*(CB(1,2)*BBY+(CB(1,4))*BBZ+(CB(1,6))*BBX)+
     +            BY*((CB(6,2))*BBY+(CB(6,4))*BBZ+(CB(6,6))*BBX)
          ELSE
           AIJ = BX*(CB(1,3)*BBZ+(CB(1,4))*BBY+(CB(1,5))*BBX)+
     +           BY*((CB(6,3))*BBZ+(CB(6,4))*BBY+(CB(6,5))*BBX)
          ENDIF
       ELSEIF(IU.EQ.2) THEN
          IF(JU.EQ.1) THEN
           AIJ = BY*(CB(2,1)*BBX+(CB(2,5))*BBZ+(CB(2,6))*BBY)+
     +           BX*((CB(6,1))*BBX+(CB(6,5))*BBZ+(CB(6,6))*BBY)
          ELSEIF(JU.EQ.2) THEN
           AIJ = BY*(CB(2,2)*BBY+(CB(2,4))*BBZ+(CB(2,6))*BBX)+
     +           BX*((CB(6,2))*BBY+(CB(6,4))*BBZ+(CB(6,6))*BBX)
           BIJ = BB*BBB*DEN
          ELSE
           AIJ = BY*(CB(2,3)*BBZ+(CB(2,4))*BBY+(CB(2,5))*BBX)+
     +           BX*((CB(6,3))*BBZ+(CB(6,4))*BBY+(CB(6,5))*BBX)
          ENDIF
       ELSE
          IF(JU.EQ.1) THEN
           AIJ = BZ*(CB(3,1)*BBX+(CB(3,5))*BBZ+(CB(3,6))*BBY)
          ELSEIF(JU.EQ.2) THEN
           AIJ = BZ*(CB(3,2)*BBY+(CB(3,4))*BBZ+(CB(3,6))*BBX)
          ELSE
           AIJ = BZ*(CB(3,3)*BBZ+(CB(3,4))*BBY+(CB(3,5))*BBX)
           BIJ = BB*BBB*DEN
          ENDIF
       ENDIF
*
```

```
                        A(IJ) = A(IJ)+AIJ*VOL
                        B(IJ) = B(IJ)+BIJ*VOL

*
2080            CONTINUE
2070          CONTINUE
2060        CONTINUE
2040      CONTINUE
2020    CONTINUE
*2000   CONTINUE
        DO 3020 IQ = 1,1
                            XI = 0.D0
                            EN = 0.D0
                            ZI = 0.D0
          DO 3040 IN = 1,NNE
                                IP = NODE(IT,IN)
                             CALL TBF(XI,EN,ZI,IN,BB,BX,BY,BZ,
     +    *                            X1,Y1,Z1,X7,Y7,Z7)
            DO 3060 IU = 1,3
                               I = INDX(IP,IU)
                                IF(I.EQ.0) GOTO 3060
              DO 3070 INN = 1,NNE
                               IPP = NODE(IT,INN)
                             CALL TBF(XI,EN,ZI,INN,BBB,BBX,BBY,BBZ,
     +                            X1,Y1,Z1,X7,Y7,Z7)
                DO 3080 JU = 1,3
                               J = INDX(IPP,JU)
                                IF(I.GT.J) GOTO 3080
           IJ = MAXA(J) + (J-I)
         IF(IU.EQ.1) THEN
           IF(JU.EQ.1) THEN
         AIJ = BZ*((CB(5,1))*BBX+(CB(5,5))*BBZ+(CB(5,6))*BBY)
           ELSEIF(JU.EQ.2) THEN
         AIJ = BZ*((CB(5,2))*BBY+(CB(5,4))*BBZ+(CB(5,6))*BBX)
           ELSE
         AIJ = BZ*((CB(5,3))*BBZ+(CB(5,4))*BBY+(CB(5,5))*BBX)
           ENDIF
         ELSEIF(IU.EQ.2) THEN
           IF(JU.EQ.1) THEN
         AIJ = BZ*((CB(4,1))*BBX+(CB(4,5))*BBZ+(CB(4,6))*BBY)
           ELSEIF(JU.EQ.2) THEN
         AIJ = BZ*((CB(4,2))*BBY+(CB(4,4))*BBZ+(CB(4,6))*BBX)
           ELSE
         AIJ = BZ*((CB(4,3))*BBZ+(CB(4,4))*BBY+(CB(4,5))*BBX)
           ENDIF
         ELSE
-          IF(JU.EQ.1) THEN
         AIJ = BX*((CB(5,1))*BBX+(CB(5,5))*BBZ+(CB(5,6))*BBY)+
     +         BY*((CB(4,1))*BBX+(CB(4,5))*BBZ+(CB(4,6))*BBY)
           ELSEIF(JU.EQ.2) THEN
         AIJ = BX*((CB(5,2))*BBY+(CB(5,4))*BBZ+(CB(5,6))*BBX)+
     +         BY*((CB(4,2))*BBY+(CB(4,4))*BBZ+(CB(4,6))*BBX)
           ELSE
         AIJ = BX*((CB(5,3))*BBZ+(CB(5,4))*BBY+(CB(5,5))*BBX)+
-    +         BY*((CB(4,3))*BBZ+(CB(4,4))*BBY+(CB(4,5))*BBX)
```

||

3DFEM FORTRAN A1 11/19/90 22:43 F 80 2264 RECS 11/19/90 22:44 PAGE

```
           ENDIF
         ENDIF
                        A(IJ) = A(IJ)+AIJ*VOL
3080            CONTINUE
3070          CONTINUE
3060        CONTINUE
3040      CONTINUE
3020    CONTINUE
2000    CONTINUE
        RETURN
        END
        SUBROUTINE  LOCCOR(NQ,XQ,YQ,ZQ)
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
        DIMENSION XQ(NQ),YQ(NQ),ZQ(NQ)
        XQ(1) = (-1.D0/DSQRT(3.D0))
        YQ(1) = (-1.D0/DSQRT(3.D0))
        ZQ(1) = (-1.D0/DSQRT(3.D0))
        XQ(2) = (+1.D0/DSQRT(3.D0))
        YQ(2) = (-1.D0/DSQRT(3.D0))
        ZQ(2) = (-1.D0/DSQRT(3.D0))
        XQ(3) = (+1.D0/DSQRT(3.D0))
        YQ(3) = (+1.D0/DSQRT(3.D0))
        ZQ(3) = (-1.D0/DSQRT(3.D0))
        XQ(4) = (-1.D0/DSQRT(3.D0))
        YQ(4) = (+1.D0/DSQRT(3.D0))
        ZQ(4) = (-1.D0/DSQRT(3.D0))
        XQ(5) = (-1.D0/DSQRT(3.D0))
        YQ(5) = (-1.D0/DSQRT(3.D0))
        ZQ(5) = (+1.D0/DSQRT(3.D0))
        XQ(6) = (+1.D0/DSQRT(3.D0))
        YQ(6) = (-1.D0/DSQRT(3.D0))
        ZQ(6) = (+1.D0/DSQRT(3.D0))
        XQ(7) = (+1.D0/DSQRT(3.D0))
        YQ(7) = (+1.D0/DSQRT(3.D0))
        ZQ(7) = (+1.D0/DSQRT(3.D0))
        XQ(8) = (-1.D0/DSQRT(3.D0))
        YQ(8) = (+1.D0/DSQRT(3.D0))
        ZQ(8) = (+1.D0/DSQRT(3.D0))
        RETURN
        END
        SUBROUTINE  TBF(XI,EN,ZI,IN,BB,BX,BY,BZ,
       +                X1,Y1,Z1,X7,Y7,Z7)
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
        IF(IN.EQ.1) THEN
         BB = (1.D0/8.D0)*(1.D0-XI)*(1.D0-EN)*(1.D0-ZI)
         BX = -(1.D0/8.D0)*(1.D0-EN)*(1.D0-ZI)*(2.D0/(X7-X1))
         BY = -(1.D0/8.D0)*(1.D0-XI)*(1.D0-ZI)*(2.D0/(Y7-Y1))
         BZ = -(1.D0/8.D0)*(1.D0-XI)*(1.D0-EN)*(2.D0/(Z7-Z1))
        ELSEIF(IN.EQ.2) THEN
          BB = (1.D0/8.D0)*(1.D0+XI)*(1.D0-EN)*(1.D0-ZI)
          BX =  (1.D0/8.D0)*(1.D0-EN)*(1.D0-ZI)*(2.D0/(X7-X1))
          BY = -(1.D0/8.D0)*(1.D0+XI)*(1.D0-ZI)*(2.D0/(Y7-Y1))
          BZ = -(1.D0/8.D0)*(1.D0+XI)*(1.D0-EN)*(2.D0/(Z7-Z1))
        ELSEIF(IN.EQ.3) THEN
```

```
          BB = (1.D0/8.D0)*(1.D0+XI)*(1.D0-EN)*(1.D0+ZI)
          BX =  (1.D0/8.D0)*(1.D0-EN)*(1.D0+ZI)*(2.D0/(X7-X1))
          BY = -(1.D0/8.D0)*(1  D0+XI)*(1.D0+ZI)*(2.D0/(Y7-Y1))
          BZ =  (1.D0/8.D0)*(1.D0+XI)*(1.D0-EN)*(2.D0/(Z7-Z1))
        ELSEIF(IN.EQ.4) THEN
           BB = (1.D0/8.D0)*(1.D0-XI)*(1.D0-EN)*(1.D0+ZI)
           BX = -(1.D0/8.D0)*(1.D0-EN)*(1.D0+ZI)*(2.D0/(X7-X1))
           BY = -(1.D0/8.D0)*(1.D0-XI)*(1.D0+ZI)*(2.D0/(Y7-Y1))
           BZ =  (1.D0/8.D0)*(1.D0-XI)*(1.D0-EN)*(2.D0/(Z7-Z1))
        ELSEIF(IN.EQ.5) THEN
           BB = (1.D0/8.D0)*(1.D0-XI)*(1.D0+EN)*(1.D0-ZI)
           BX = -(1.D0/8.D0)*(1.D0+EN)*(1.D0-ZI)*(2.D0/(X7-X1))
           BY =  (1.D0/8.D0)*(1.D0-XI)*(1.D0-ZI)*(2.D0/(Y7-Y1))
           BZ = -(1.D0/8.D0)*(1.D0-XI)*(1.D0+EN)*(2.D0/(Z7-Z1))
        ELSEIF(IN.EQ.6) THEN
           BB = (1.D0/8.D0)*(1.D0+XI)*(1.D0+EN)*(1.D0-ZI)
           BX = (1.D0/8.D0)*(1.D0+EN)*(1.D0-ZI)*(2.D0/(X7-X1))
           BY = (1.D0/8.D0)*(1.D0+XI)*(1.D0-ZI)*(2.D0/(Y7-Y1))
           BZ =-(1.D0/8.D0)*(1.D0+XI)*(1.D0+EN)*(2.D0/(Z7-Z1))
        ELSEIF(IN.EQ.7) THEN
           BB = (1.D0/8.D0)*(1.D0+XI)*(1.D0+EN)*(1.D0+ZI)
           BX = (1.D0/8.D0)*(1.D0+EN)*(1.D0+ZI)*(2.D0/(X7-X1))
           BY = (1.D0/8.D0)*(1.D0+XI)*(1.D0+ZI)*(2.D0/(Y7-Y1))
           BZ = (1.D0/8.D0)*(1.D0+XI)*(1.D0+EN)*(2.D0/(Z7-Z1))
        ELSEIF(IN.EQ.8) THEN
           BB = (1.D0/8.D0)*(1.D0-XI)*(1.D0+EN)*(1.D0+ZI)
           BX = -(1.D0/8.D0)*(1.D0+EN)*(1.D0+ZI)*(2.D0/(X7-X1))
           BY =  (1.D0/8.D0)*(1.D0-XI)*(1.D0+ZI)*(2.D0/(Y7-Y1))
           BZ =  (1.D0/8.D0)*(1.D0-XI)*(1.D0+EN)*(2.D0/(Z7-Z1))
        ENDIF
C
        RETURN
        END
        SUBROUTINE CBAR3D(IT,LYNUM,TH,CB,NE,NX,NY,NZ,MATE,IFLAG)
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
        DIMENSION S(6,6),C(6,6)
        DIMENSION CB(6,6)
*       WRITE(*,*) 'THETA IN CBAR ',TH
                              IF(MATE.EQ.1) GOTO 666
        IF(IFLAG.EQ.1) THEN
        E1 = 20000000.0D0
        E2 = 1300000.0D0
        E3 = 1250000.0D0
        P23 = 0.49D0
        P13 = 0.3D0
-       P12 = 0.3D0
        G23 = 900000.0D0
        G13 = 1030000.0D0
        G12 = 1030000.0D0
        ELSEIF(IFLAG.EQ.2) THEN
         E1 = 25047135.60D0
         E2 = 1044234.95D0
         E3 = E2
         P23 = 0.35D0
-        P13 = 0.3D0
```

||

3DFEM FORTRAN A1 11/19/90 22:43 F 80 2264 RECS 11/19/90 22:44 PAGE

```
         P12 = 0.3D0
         G12 = 545322.67D0
         G23 = 0.9D0*G12
         G13 = G12
        ELSEIF(IFLAG.EQ.3) THEN
           E1 = 18700000.0D0
           E2 = 1070000.0D0
           E3 = 1070000.0D0
           P23 = 0.400D0
           P13 = 0.330D0
           P12 = 0.330D0
           G12 = 610000.0D0
           G23 = 695000.0D0
           G13 = 610000.0D0
        ELSEIF(IFLAG.EQ.4) THEN
           E1 = 25.D0
           E2 = 1.D0
           E3 = 1.D0
           P23 = 0.25D0
           P13 = 0.25D0
           P12 = 0.25D0
           G12 = 0.5D0
           G23 = 0.2D0
           G13 = 0.5D0
        ELSE
           E1 = 2.695D6
           E2 = 2.56D6
           E3 = E2
           P23 = 0.230D0
           P13 = 0.230D0
           P12 = 0.242D0
           G12 = 0.6D6
           G23 = 0.9D0*G12
           G13 = G12
        ENDIF
*
        S(1,1) = 1.D0/E1
        S(1,2) = -P12/E1
        S(1,3) = -P13/E1
        S(2,2) = 1.D0/E2
        S(2,3) = -P23/E2
        S(3,3) = 1.D0/E3
        S(4,4) = 1.D0/G23
        S(5,5) = 1.D0/G13
        S(6,6) = 1.D0/G12
-       TM =DCOS(TH)
        TN =DSIN(TH)
        SA= S(1,1)*S(2,2)*S(3,3)-S(1,1)*S(2,3)**2-
     1     S(2,2)*S(1,3)**2-S(3,3)*S(1,2)**2+
     2     2*S(1,2)*S(2,3)*S(1,3)
         C(1,1) = (S(2,2)*S(3,3)-S(2,3)**2)/SA
         C(1,2) = (S(1,3)*S(2,3)-S(1,2)*S(3,3))/SA
         C(2,2) = (S(3,3)*S(1,1)-S(1,3)**2)/SA
         C(1,3) = (S(1,2)*S(2,3)-S(1,3)*S(2,2))/SA
=        C(3,3) = (S(1,1)*S(2,2)-S(1,2)**2)/SA
```

```
      C(2,3) = (S(1,2)*S(1,3)-S(2,3)*S(1,1))/SA
      C(4,4) = 1.D0/S(4,4)
      C(5,5) = 1.D0/S(5,5)
      C(6,6) = 1.D0/S(6,6)
      CB(1,1) = TM**4*C(1,1)+2*TM**2*TN**2*(C(1,2)+2*C(6,6))+
     &          TN**4*C(2,2)
      CB(1,2) = TN**2*TM**2*(C(1,1)+C(2,2)-4*C(6,6))+
     &          (TN**4+TM**4)*C(1,2)
      CB(1,3) = TM**2*C(1,3)+TN**2*C(2,3)
      CB(1,4) = 0.D0
      CB(1,5) = 0.D0
      CB(1,6) = TM*TN*(TM**2*(C(1,1)-C(1,2)-2*C(6,6))+
     &          TN**2*(C(1,2)-C(2,2)+2*C(6,6)))
      CB(2,2) = TN**4*C(1,1)+2*TN**2*TM**2*(C(1,2)+
     &          2*C(6,6))+TM**4*C(2,2)
      CB(2,3) = TN**2*C(1,3)+TM**2*C(2,3)
      CB(2,4) = 0.D0
      CB(2,5) = 0.D0
      CB(2,6) = TM*TN*(TN**2*(C(1,1)-C(1,2)-2*C(6,6))+
     1          TM**2*(C(1,2)-C(2,2)+2*C(6,6)))
      CB(3,3) = C(3,3)
      CB(3,4) = 0.D0
      CB(3,5) = 0.D0
      CB(3,6) = TM*TN*(C(1,3)-C(2,3))
      CB(4,4) = TM**2*C(4,4)+TN**2*C(5,5)
      CB(4,5) = TM*TN*(C(5,5)-C(4,4))
      CB(4,6) = 0.D0
      CB(5,5) = TN**2*C(4,4)+TM**2*C(5,5)
      CB(5,6) = 0.D0
      CB(6,6) = TM**2*TN**2*(C(1,1)-2*C(1,2)+C(2,2))+
     &          C(6,6)*(TM**2-TN**2)**2
      DO 90 I = 1,6
           DO 95 J = 1,6
                CB(J,I) = CB(I,J)
*     WRITE(*,*) I,J,CB(I,J)
95         CONTINUE
90    CONTINUE
*     WRITE(*,*) ' '
      GOTO 177
C
666     CONTINUE
      WRITE(*,*) ' '
177   RETURN
      END
      SUBROUTINE GVREST(OMEGA,NU,NC,EIGV,R,IVEC,IPRVEC,IZOLINE)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION OMEGA(NC)
      DIMENSION EIGV(NC),R(NU,NC)
      WRITE(*,*) ' '
      WRITE(*,*) '-----------------------------------------'
      WRITE(*,*) ' N A T U R A L    F R E Q U E N C I E S '
      WRITE(*,*) '----------------------------------------- '
      WRITE(*,*) ' '
      DO 250 I=1,NC
      WRITE(*,255) I,OMEGA(I),EIGV(I)
```

||

```
3DFEM    FORTRAN  A1 11/19/90 22:43    F 80  2264 RECS 11/19/90 22:44 PAGE

255   FORMAT(1X,'NO:',I4,1X,'FREQ:',D12.6,1X,'HZ',2X,'E-VAL',1X,D12.6)
250   CONTINUE

       IF(IVEC.EQ.1) THEN

                          KK = 0
           DO 264 K=IPRVEC,IPRVEC

*          DO 264 K=1,IPRVEC

      WRITE(*,*) ' '
      WRITE(*,*) ' -------------------- '
      WRITE(*,*) '  '
      WRITE(*,*) ' MODE -----> ',K
      WRITE(*,*) ' '
      WRITE(*,*) ' -------------------- '
      WRITE(*,*) ' '
      PII = 3.141592654D0
      WRITE(*,*) ' NATURAL FREQUENCY :',OMEGA(K)
      WRITE(*,*) ' '

              DO 268 L = 1,NU,3
C             DO 268 L = 361,363,3

                                  KK=KK+1

              WRITE(*,277) KK,R(L,K),R(L+1,K),R(L+2,K)
277   FORMAT(I4,3X,'U :',D16.4,3X,'V :',D16.4,3X,'W :',D16.4)

268           CONTINUE

                                  KK = 0

*  NODES OF INTEREST

*                                 KK = 1236

          RMS = 0.D0

         DO 288  I = 3,NU,3

               RMS = RMS+R(I,K)*R(I,K)

288      CONTINUE

               RMS = DSQRT(RMS)
               WRITE(*,*) 'RMS ',RMS
               WRITE(*,*) 'W   ',R(179*3,K)
               WRITE(*,*) 'W   ',R(180*3,K)

      RATIO1 = R(179 *3,K)/RMS
      RATIO2 = R(180 *3,K)/RMS
C     RATIO3 = R(1307*3,K)/RMS
C     RATIO4 = R(1308*3,K)/RMS
```

```
          WRITE(*,*) 'NODE 179    ',RATIO1
          WRITE(*,*) 'NODE 180    ',RATIO2
C         WRITE(*,*) 'NODE 1307   ',RATIO3
C         WRITE(*,*) 'NODE 1308   ',RATIO4
264          CONTINUE

         ENDIF

*
*  WRITE THE Z COORDS TO FILE 10 TO PLOT USING SAS-GRAF
*
             IF(IZOLINE.EQ.1) THEN
                        L = 0
                      DO 3010 J = 0,5
                        DO 3020 I = 0,4
                         L=L+5
                    WRITE(10,1001) I*1.8D0,J*0.72D0,R(L*3,IPRVEC)
1001       FORMAT(3F14.4,3X)

3020       CONTINUE

3010       CONTINUE


             ENDIF

        RETURN
        END

        SUBROUTINE GGEOM(IVDELAM,NDELL,NODEL,LYNUM,NODE,INDX,XN,YN,ZN,NE,
     +           NN,NU,NX,NY,NZ,XL,YL,ZL,NNE,LEPL,DEGR,MATE,IBOUND,KDC,
     +           XDST,XDEN,YDST,YDEN,ISDELAM,ISTATIC)
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
        DIMENSION XN(NX*NY*(NZ+NDELL)),YN(NX*NY*(NZ+NDELL)),
                  ZN(NX*NY*(NZ+NDELL))
        DIMENSION NODE((NX-1)*(NY-1)*(NZ-1),8),
                  INDX(NX*NY*(NZ+NDELL),3)
        DIMENSION DEGR(NZ-1)
        INTEGER KDC(NZ),XDST(NDELL+1),XDEN(NDELL+1)
        INTEGER YDST(NDELL+1),YDEN(NDELL+1)

        IF(ISTATIC.EQ.0) THEN
        WRITE(*,*) '  '
        WRITE(*,*)  '================================================='
        WRITE(*,*)  '=                                               ='
-       WRITE(*,*)  '= F I N I T E   E L E M E N T   A N A L Y S I S='
        WRITE(*,*)  '=           FREE  VIBRATION PROBLEM             ='
        WRITE(*,*)  '=             DELAMINATED  PLATE                ='
        WRITE(*,*)  '=                                               ='
        WRITE(*,*)  '================================================='
        WRITE(*,*) '  '
        ELSEIF(ISTATIC.EQ.1) THEN
        WRITE(*,*) '  '
        WRITE(*,*)  '================================================='
=       WRITE(*,*)  '=                                               ='
```

```
3DFEM    FORTRAN  A1 11/19/90 22:43    F 80  2264 RECS 11/19/90 22:44 PAGE      2

       WRITE(*,*)  '= F I N I T E   E L E M E N T   A N A L Y S I S='
       WRITE(*,*)  '=                                               ='
       WRITE(*,*)  '=            DELAMINATED  PLATE                 ='
       WRITE(*,*)  '=                                               ='
       WRITE(*,*)  '================================================='
       WRITE(*,*) '  '
       ENDIF

        IF(IBOUND.EQ.1) THEN
           WRITE(*,*) ' FULLY FREE BOUNDARY CONDITIONS '
        ELSEIF(IBOUND.EQ.2) THEN
           WRITE(*,*) ' CLAMPED  FREE FREE FREE BOUNDARY CONDITIONS'
        ELSEIF(IBOUND.EQ.3) THEN
           WRITE(*,*) ' FULLY  CLAMPED  BOUNDARY CONDITIONS'
        ELSEIF(IBOUND.EQ.4) THEN
           WRITE(*,*) ' FULLY  SIMPLY SUPPORTED BOUNDARY CONDITIONS'
        ELSEIF(IBOUND.EQ.5) THEN
           WRITE(*,*) ' CLAMPED FREE CLAMPED CLAMPED BOUNDARY CONDITIONS'
        ELSE
           WRITE(*,*) ' S.SUP FREE S.SUP S.SUP BOUNDARY CONDITIONS'
        ENDIF
*
      WRITE(*,*) '  '
      IF(IVDELAM.EQ.0) THEN
      WRITE(*,*) 'UNDAMAGED LAMINATE '
      ELSE
       WRITE(*,*) 'DELAMINATED LAMINATE '
       WRITE(*,*) 'NUMBER OF DELAMINATED LAYERS ->',NDELL
       WRITE(*,*) 'TOTAL NODES IN DAMAGED REGIONS>',NODEL
      ENDIF
      WRITE(*,*) '  '

      IF(IVDELAM.EQ.1) THEN
      WRITE(*,*) '-------------------------'
      WRITE(*,*) 'DELAMINATION INFORMATION'
      WRITE(*,*) '1 DELAM PER LAYER       '
      WRITE(*,*) '-------------------------'
      WRITE(*,*) '  '
      WRITE(*,*) 'FROM BOTTOM TO TOP '
      WRITE(*,*) '  '

       DO 3012 LM=0,NZ-1

          IF(KDC(LM).NE.0) THEN
            WRITE(*,*)'DELAMINATION AT INTERFACE :',LM
-         ENDIF
3012   CONTINUE

      WRITE(*,*) '  '
      DO 3010 II = 1,NDELL
      WRITE(*,*) 'DELAM :',II
      DELX1 = XDST(II)*(XL/(NX-1.D0))
      DELX2 = XDEN(II)*(XL/(NX-1.D0))
=     DELY1 = YDST(II)*(YL/(NY-1.D0))
```

```
      DELY2 = YDEN(II)*(YL/(NY-1.D0))
      WRITE(*,*) 'X DEL SIZE (IN)',DELX2-DELX1
      WRITE(*,*) 'Y DEL SIZE (IN)',DELY2-DELY1
      WRITE(*,*) 'X BOX ST',XDST(II)
      WRITE(*,*) 'X BOX ED',XDEN(II)
      WRITE(*,*) 'Y BOX ST',YDST(II)
      WRITE(*,*) 'Y BOX ED',YDEN(II)
      WRITE(*,*) '  '
 3010 CONTINUE
      ENDIF

      WRITE(*,*) '  '
      WRITE(*,*) 'NUMBER OF ELEMENT NODES       ->',NNE
      WRITE(*,*) 'NUMBER OF LAYERS              ->',NZ-1
      WRITE(*,*) 'STRUCTURE LENGTH (IN)         ->',XL
      WRITE(*,*) 'STRUCTURE WIDTH               ->',YL
      WRITE(*,*) 'STRUCTURE HEIGHT              ->',ZL*(NZ-1)
      WRITE(*,*) 'ELEMENTS ACROSS THICKNESS     ->',1
      WRITE(*,*) 'NODES IN X DIRECTION          ->',NX
      WRITE(*,*) 'NODES IN Y DIRECTION          ->',NY
      WRITE(*,*) 'NODES IN Z DIRECTION          ->',NZ
      WRITE(*,*) 'ELEMENTS PER LAYER            ->',(NX-1)*(NY-1)
      WRITE(*,*) 'NODES PER LAYER               ->',NX*NY
      WRITE(*,*) 'MATERIAL TYPE <1>ISO<2>ANISO ->',MATE
      WRITE(*,*) '  '
      IF(MATE.NE.2) GOTO 777
      DO 65 I = 1,NZ-1
      WRITE(*,67) I,DEGR(I)
 67   FORMAT(1X,'LAYER->',I4,2X,'FIBER ORIENTATION',F12.4)
 65   CONTINUE
      WRITE(*,*) '  '
 777  LEPL= (NX-1)*(NY-1)
      IT = 0.D0
      IU = 0.D0
      IP = 0.D0
      ZINCR = 0.D0
      HX = (XL-0.D0)/(NX-1.D0)
      HY = (YL-0.D0)/(NY-1.D0)
      HZ = ZL
C     HZ = (ZL-0.D0)/(NZ-1.D0)
C
*     DO 1025 LY = 1, LYNUM
        DO 1040 JC = 0, NY-1
          DO 1050 IC = 0, NX-1
            DO 1030 KC = 0, NZ-1
*
*
*   CANTILEVER PLATE
*
*
                  IF(IBOUND.EQ.2.AND.IC.EQ.0) GOTO 3100
*
*
*   FULLY CLAMPED
```

3DFEM FORTRAN A1 11/19/90 22:43 F 80 2264 RECS 11/19/90 22:44 PAGE

```
*
                  IF(IBOUND.EQ.3.AND.IC.EQ.0) GOTO 3100
                  IF(IBOUND.EQ.3.AND.IC.EQ.(NX-1)) GOTO 3100
                  IF(IBOUND.EQ.3.AND.JC.EQ.0) GOTO 3100
                  IF(IBOUND.EQ.3.AND.JC.EQ.(NY-1)) GOTO 3100
*
*
*   SIMPLY SUPPORTED
*
*
       IF(IBOUND.EQ.4.AND.IC.EQ.0.AND.KC.EQ.0) GOTO 3100
       IF(IBOUND.EQ.4.AND.IC.EQ.(NX-1).AND.KC.EQ.0) GOTO 3100
       IF(IBOUND.EQ.4.AND.JC.EQ.0.AND.KC.EQ.0) GOTO 3100
       IF(IBOUND.EQ.4.AND.JC.EQ.(NY-1).AND.KC.EQ.0) GOTO 3100
*
*
*   CLAMPED FREE CLAMPED CLAMPED
*
*
                  IF(IBOUND.EQ.5.AND.IC.EQ.0) GOTO 3100
                  IF(IBOUND.EQ.5.AND.JC.EQ.0) GOTO 3100
                  IF(IBOUND.EQ.5.AND.JC.EQ.(NY-1)) GOTO 3100
*
*
*   SIMPLY SUPPORTED FREE SIMPLY SUPPORTED SIMPLY SUPPORTED
*
*
       IF(IBOUND.EQ.6.AND.IC.EQ.0.AND.KC.EQ.0) GOTO 3100
       IF(IBOUND.EQ.6.AND.JC.EQ.0.AND.KC.EQ.0) GOTO 3100
       IF(IBOUND.EQ.6.AND.JC.EQ.(NY-1).AND.KC.EQ.0) GOTO 3100
*
*
*   SIMPLY SUPPORTED CONSTRAINED AT THE MID SURFACE
*
*
       IF(IBOUND.EQ.7.AND.IC.EQ.0.AND.KC.EQ.((NZ-1)/2)) GOTO 3100
       IF(IBOUND.EQ.7.AND.IC.EQ.(NX-1).AND.KC.EQ.((NZ-1)/2)) GOTO 3100
       IF(IBOUND.EQ.7.AND.JC.EQ.0.AND.KC.EQ.((NZ-1)/2)) GOTO 3100
       IF(IBOUND.EQ.7.AND.JC.EQ.(NY-1).AND.KC.EQ.((NZ-1)/2)) GOTO 3100
*
            IU = IU+3.D0
3100        IP = IP+1.D0
             XN(IP) =IC*HX
             YN(IP) =JC*HY
             ZN(IP) =KC*HZ+ZINCR
*
            IF(IBOUND.EQ.2.AND.IC.EQ.0) THEN
             INDX(IP,1) = 0
             INDX(IP,2) = 0
             INDX(IP,3) = 0
            ELSEIF(IBOUND.EQ.3.AND.IC.EQ.0) THEN
             INDX(IP,1) = 0
             INDX(IP,2) = 0
             INDX(IP,3) = 0
            ELSEIF(IBOUND.EQ.3.AND.IC.EQ.(NX-1)) THEN
```

```
                INDX(IP,1) = 0
                INDX(IP,2) = 0
                INDX(IP,3) = 0
               ELSEIF(IBOUND.EQ.3.AND.JC.EQ.0) THEN
                INDX(IP,1) = 0
                INDX(IP,2) = 0
                INDX(IP,3) = 0
               ELSEIF(IBOUND.EQ.3.AND.JC.EQ.(NY-1)) THEN
                INDX(IP,1) = 0
                INDX(IP,2) = 0
                INDX(IP,3) = 0
               ELSEIF(IBOUND.EQ.4.AND.IC.EQ.0.AND.KC.EQ.0) THEN
                INDX(IP,1) = 0
                INDX(IP,2) = 0
                INDX(IP,3) = 0
               ELSEIF(IBOUND.EQ.4.AND.IC.EQ.(NX-1).AND.KC.EQ.0) THEN
                INDX(IP,1) = 0
                INDX(IP,2) = 0
                INDX(IP,3) = 0
               ELSEIF(IBOUND.EQ.4.AND.JC.EQ.0.AND.KC.EQ.0) THEN
                INDX(IP,1) = 0
                INDX(IP,2) = 0
                INDX(IP,3) = 0
               ELSEIF(IBOUND.EQ.4.AND.JC.EQ.(NY-1).AND.KC.EQ.0) THEN
                INDX(IP,1) = 0
                INDX(IP,2) = 0
                INDX(IP,3) = 0
               ELSEIF(IBOUND.EQ.5.AND.IC.EQ.0) THEN
                INDX(IP,1) = 0
                INDX(IP,2) = 0
                INDX(IP,3) = 0
               ELSEIF(IBOUND.EQ.5.AND.JC.EQ.0) THEN
                INDX(IP,1) = 0
                INDX(IP,2) = 0
                INDX(IP,3) = 0
               ELSEIF(IBOUND.EQ.5.AND.JC.EQ.(NY-1)) THEN
                INDX(IP,1) = 0
                INDX(IP,2) = 0
                INDX(IP,3) = 0
               ELSEIF(IBOUND.EQ.6.AND.IC.EQ.0.AND.KC.EQ.0) THEN
                INDX(IP,1) = 0
                INDX(IP,2) = 0
                INDX(IP,3) = 0
               ELSEIF(IBOUND.EQ.6.AND.JC.EQ.0.AND.KC.EQ.0) THEN
                INDX(IP,1) = 0
                INDX(IP,2) = 0
                INDX(IP,3) = 0
               ELSEIF(IBOUND.EQ.6.AND.JC.EQ.(NY-1).AND.KC.EQ.0) THEN
                INDX(IP,1) = 0
                INDX(IP,2) = 0
                INDX(IP,3) = 0
      ELSEIF(IBOUND.EQ.7.AND.IC.EQ.0.AND.KC.EQ.((NZ-1)/2)) THEN
                INDX(IP,1) = 0
                INDX(IP,2) = 0
                INDX(IP,3) = 0
```

```
3DFEM    FORTRAN  A1 11/19/90 22:43    F 80  2264 RECS 11/19/90 22:44 PAGE

      ELSEIF(IBOUND.EQ.7.AND.IC.EQ.(NX-1).AND.KC.EQ.((NZ-1)/2)) THEN
                INDX(IP,1) = 0
                INDX(IP,2) = 0
                INDX(IP,3) = 0
      ELSEIF(IBOUND.EQ.7.AND.JC.EQ.0.AND.KC.EQ.((NZ-1)/2)) THEN
                INDX(IP,1) = 0
                INDX(IP,2) = 0
                INDX(IP,3) = 0
      ELSEIF(IBOUND.EQ.7.AND.JC.EQ.(NY-1).AND.KC.EQ.((NZ-1)/2)) THEN
                INDX(IP,1) = 0
                INDX(IP,2) = 0
                INDX(IP,3) = 0
               ELSE
                INDX(IP,1) = IU-2.D0
                INDX(IP,2) = IU-1.D0
                INDX(IP,3) = IU
               ENDIF
C
C
C  BY MANIPULATING GT=GE .. IN THE FOLLOWING IF STATEMENTS
C  THE DELAMINATIONS CAN BE EXTENDED TO THE FREE ENDS
C  PUT DOUBLE NODES ACROSS THE DELAMINATED LAYER
C  ONLY NODES IN THE DELAMINATED REGION GENERATE 3 MORE UKNOWNS
C  U,V,W SAME FOR OTHER NODES
      IF(IVDELAM.EQ.1.AND.KDC(KC).NE.0) THEN
      IF(XN(IP).GT.(XDST(KDC(KC))*HX).AND.XN(IP).LT.(XDEN(KDC(KC))*HX).
     + AND.YN(IP).GT.(YDST(KDC(KC))*HY).AND.YN(IP).LT.
     +  (YDEN(KDC(KC))*HY)) THEN
            IP = IP+1
            IU = IU+3
            XN(IP) = IC*HX
            YN(IP) = JC*HY
            ZN(IP) = KC*HZ
               INDX(IP,1) = IU-2.D0
               INDX(IP,2) = IU-1.D0
               INDX(IP,3) = IU
         ELSE
              IP = IP+1
              XN(IP) = IC*HX
              YN(IP) = JC*HY
              ZN(IP) = KC*HZ
              INDX(IP,1) = INDX(IP-1,1)
              INDX(IP,2) = INDX(IP-1,2)
              INDX(IP,3) = INDX(IP-1,3)
         ENDIF
      ENDIF
C*
 1030     CONTINUE
 1050    CONTINUE
 1040   CONTINUE

                    NN = IP
      WRITE(*,*)  'TOTAL NODES OF STRUCTURE     ->',NN
                    NU = IU
      WRITE(*,*)  'TOTAL NUMBER OF UNKNOWNS     ->',NU
```

```
                              NODINCR = 0

          DO 1060 KC = 0,NZ-2
              IF(IVDELAM.EQ.1.AND.KDC(KC).NE.0) THEN
                  NODINCR = NODINCR + 1
              ENDIF
             DO 1070 JC = 0,NY-2
               DO 1080 IC = 0,NX-2
           IT = IT + 1.D0
*
          NODE(IT,1) = (NX*(NZ+NDELL)*JC)+(IC*(NZ+NDELL))+1.D0+NODINCR
                NODE(IT,2) = NODE(IT,1)+(NZ+NDELL)
                NODE(IT,3) = NODE(IT,2)+1.D0
                NODE(IT,4) = NODE(IT,1)+1.D0
                NODE(IT,5) = NODE(IT,1)+(NX*(NZ+NDELL))
                NODE(IT,6) = NODE(IT,5)+(NZ+NDELL)
           .    NODE(IT,7) = NODE(IT,6)+1.D0
                NODE(IT,8) = NODE(IT,5)+1.D0

1080        CONTINUE
1070     CONTINUE
                    NODINCR = NODINCR+1.D0
1060     CONTINUE

                       NE = IT
      WRITE(*,*)  'TOTAL ELEMENTS IN STRUCTURE  ->',NE
404   FORMAT(23X,'X',15X,'Y',15X,'Z')
*             WRITE(*,*) '  '
*             DO 400 I = 1,IP
*             WRITE(*,402) I,XN(I),YN(I),ZN(I)
402           FORMAT('NODE:',I5,1X,D16.4,1X,D16.4,1X,D16.4)
*400           CONTINUE
220   FORMAT(12X,'1',6X,'2',6X,'3',6X,'4',6X,'5',6X,'6',6X,'7',6X,'8')
*             WRITE(*,*) '  '
           DO 111 I = 1,NE
              I1 = NODE(I,1)
              I2 = NODE(I,2)
              I3 = NODE(I,3)
              I4 = NODE(I,4)
              I5 = NODE(I,5)
              I6 = NODE(I,6)
              I7 = NODE(I,7)
              I8 = NODE(I,8)
*             WRITE(*,222) I,I1,I2,I3,I4,I5,I6,I7,I8
222   FORMAT(I4,'>',2X,I6,1X,I6,1X,I6,1X,I6,1X,I6,1X,I6,1X,I6,1X,I6,1X,I6)
111        CONTINUE
230   FORMAT(17X,'1',8X,'2',7X,'3')
*             WRITE(*,*) '  '
*          DO 115 I = 1,NN
*             J1 = INDX(I,1)
*             J2 = INDX(I,2)
*             J3 = INDX(I,3)
*             WRITE(*,232) I,J1,J2,J3
232   FORMAT('NODE',I5,4X,I5,4X,I5,4X,I5)
```

```
3DFEM    FORTRAN  A1 11/19/90 22:43    F 80  2264 RECS 11/19/90 22:44 PAGE

*115         CONTINUE
C
      RETURN
      END
      SUBROUTINE AASEMB(A,B,XN,YN,ZN,NODE,INDX,NNE,NE,NN,NDELL,
     .                  NQ,LYNUM,NX,NY,NZ,DEGR,LEPL,XQ,YQ,ZQ,
     .                  THETA,MATE,DEN,ANG,IFLAG,NU,IBOUND,
     .                  NWK,NWM,MAXA,LB)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A(1000000),B(1000000)
      INTEGER   MAXA(NU+1),LB(NU)
      DIMENSION XN(NX*NY*(NZ+NDELL)),YN(NX*NY*(NZ+NDELL)),
     +          ZN(NX*NY*(NZ+NDELL))
      DIMENSION NODE((NX-1)*(NY-1)*(NZ-1),8),
     +          INDX(NX*NY*(NZ+NDELL),3)
      DIMENSION XQ(NQ),YQ(NQ),ZQ(NQ)
      DIMENSION DEGR(NZ-1),THETA(NZ-1)
      DIMENSION CB(6,6)
      DO 4500 J = 1,NU
              LB(J) = 0
4500  CONTINUE
*
               DO 4550 IT = 1,NE
                  DO 4560  IN = 1,NNE

                                          IP = NODE(IT,IN)

                        DO 4570 IU = 1,3

                                          I = INDX(IP,IU)
                              IF(I.EQ.0) GOTO 4570

                           DO 4580 JN = 1,NNE

                                          IPP = NODE(IT,JN)

                              DO 4590 IUK = 1,3

                                          J = INDX(IPP,IUK)
                              IF(J.EQ.0) GOTO 4590
*
                     IJ = J-I
                     IF(IJ.GT.LB(J)) LB(J)=IJ
*
*4590                          CONTINUE
4580                       CONTINUE
4570                    CONTINUE
4560                CONTINUE
4550             CONTINUE
                 MAXA(1) = 1

                    DO 4600 J = 2,NU
```

```
                    MAXA(J) = MAXA(J-1)+LB(J-1)+1
 4600             CONTINUE

                  NWK = MAXA(NU)+LB(NU)
                  NWM = NWK
                  MAXA(NU+1)=NWK+1
                    WRITE(*,*) '     '
                    WRITE(*,*) 'NWK   ',NWK
                    WRITE(*,*) '     '
       DO 1700 IK = NWK
            A(IK) = 0.D0
            B(IK) = 0.D0
 1700  CONTINUE
       CALL LOCCOR(NQ,XQ,YQ,ZQ)
       IF(MATE.EQ.2) THEN
       DO 1980 L = 1,NZ-1
          THETA(L) = (3.14159265359879323900*DEGR(L))/180.D0
 1980  CONTINUE
*           DO 1985 I = 1,LYNUM
*              DO 1990 J = 1,LEPL
*                 ANG(J) = THETA(I)
*1990          CONTINUE
*1985       CONTINUE
       ENDIF
       DO 2000 IT = 1,NE
                                  X1 = XN(NODE(IT,1))
                                  Y1 = YN(NODE(IT,1))
                                  Z1 = ZN(NODE(IT,1))
                                  X7 = XN(NODE(IT,7))
                                  Y7 = YN(NODE(IT,7))
                                  Z7 = ZN(NODE(IT,7))
          IF(MATE.EQ.2) THEN
           KK = 1 + INT((IT-1)/LEPL)
           TH = THETA(KK)
*       WRITE(*,*) 'IT ',IT
*       WRITE(*,*) 'ELE ID (KK) ',KK
*       WRITE(*,*) 'TH IN ASSEMBLY',TH
          CALL CBAR3D(IT,LYNUM,TH,CB,NE,NX,NY,NZ,MATE,IFLAG)
          ELSE
           CALL CBAR3D(IT,LYNUM,0,CB,NE,NX,NY,NZ,MATE,IFLAG)
          ENDIF
                               VOL = (X7-X1)*(Y7-Y1)*(Z7-Z1)
       DO 2020 IQ = 1,NQ
                          XI = XQ(IQ)
                          EN = YQ(IQ)
                          ZI = ZQ(IQ)
         DO 2040 IN = 1,NNE
                               IP = NODE(IT,IN)
                          CALL TBF(XI,EN,ZI,IN,BB,BX,BY,BZ,
                               X1,Y1,Z1,X7,Y7,Z7)
          DO 2060 IU = 1,3
                              I = INDX(IP,IU)
                               IF(I.EQ.0) GOTO 2060
           DO 2070 INN = 1,NNE
```

3DFEM FORTRAN A1 11/19/90 22:43 F 80 2264 RECS 11/19/90 22:44 PAGE

```
                              IPP = NODE(IT,INN)
                          CALL TBF(XI,EN,ZI,INN,BBB,BBX,BBY,BBZ,
                               X1,Y1,Z1,X7,Y7,Z7)
            DO 2080 JU = 1,3
                              J = INDX(IPP,JU)
                               IF(I.GT.J) GOTO 2080
      IJ = MAXA(J) + (J-I)
                    BIJ = 0.D0

     IF(IU.EQ.1) THEN
        IF(JU.EQ.1) THEN
         AIJ = BX*(CB(1,1)*BBX+(CB(1,5))*BBZ+(CB(1,6))*BBY)+
     +         BY*((CB(6,1))*BBX+(CB(6,5))*BBZ+(CB(6,6))*BBY)
         BIJ = BB*BBB*DEN
        ELSEIF(JU.EQ.2) THEN
         AIJ = BX*(CB(1,2)*BBY+(CB(1,4))*BBZ+(CB(1,6))*BBX)+
     +        BY*((CB(6,2))*BBY+(CB(6,4))*BBZ+(CB(6,6))*BBX)
        ELSE
         AIJ = BX*(CB(1,3)*BBZ+(CB(1,4))*BBY+(CB(1,5))*BBX)+
     +         BY*((CB(6,3))*BBZ+(CB(6,4))*BBY+(CB(6,5))*BBX)
        ENDIF
     ELSEIF(IU.EQ.2) THEN
        IF(JU.EQ.1) THEN
         AIJ = BY*(CB(2,1)*BBX+(CB(2,5))*BBZ+(CB(2,6))*BBY)+
     +         BX*((CB(6,1))*BBX+(CB(6,5))*BBZ+(CB(6,6))*BBY)
        ELSEIF(JU.EQ.2) THEN
         AIJ = BY*(CB(2,2)*BBY+(CB(2,4))*BBZ+(CB(2,6))*BBX)+
     +         BX*((CB(6,2))*BBY+(CB(6,4))*BBZ+(CB(6,6))*BBX)
         BIJ = BB*BBB*DEN
        ELSE
         AIJ = BY*(CB(2,3)*BBZ+(CB(2,4))*BBY+(CB(2,5))*BBX)+
     +         BX*((CB(6,3))*BBZ+(CB(6,4))*BBY+(CB(6,5))*BBX)
        ENDIF
     ELSE
        IF(JU.EQ.1) THEN
         AIJ = BZ*(CB(3,1)*BBX+(CB(3,5))*BBZ+(CB(3,6))*BBY)
        ELSEIF(JU.EQ.2) THEN
         AIJ = BZ*(CB(3,2)*BBY+(CB(3,4))*BBZ+(CB(3,6))*BBX)
        ELSE
         AIJ = BZ*(CB(3,3)*BBZ+(CB(3,4))*BBY+(CB(3,5))*BBX)
         BIJ = BB*BBB*DEN
        ENDIF
     ENDIF
                    A(IJ) = A(IJ)+AIJ*VOL
                    B(IJ) = B(IJ)+BIJ*VOL
*2080          CONTINUE
*2070        CONTINUE
 2060       CONTINUE
 2040     CONTINUE
 2020   CONTINUE
*2000   CONTINUE
        DO 3020 IQ = 1,1
                          XI = 0.D0
                          EN = 0.D0
                          ZI = 0.D0
```

```
            DO 3040 IN = 1,NNE
                                        IP = NODE(IT,IN)
                                  CALL TBF(XI,EN,ZI,IN,BB,BX,BY,BZ,
                                           X1,Y1,Z1,X7,Y7,Z7)
              DO 3060 IU = 1,3
                                        I = INDX(IP,IU)
                                         IF(I.EQ.0) GOTO 3060
                DO 3070 INN = 1,NNE
                                        IPP = NODE(IT,INN)
                                  CALL TBF(XI,EN,ZI,INN,BBB,BBX,BBY,BBZ,
                                           X1,Y1,Z1,X7,Y7,Z7)
                  DO 3080 JU = 1,3
                                        J = INDX(IPP,JU)
                                         IF(I.GT.J) GOTO 3080
           IJ = MAXA(J) + (J-I)
         IF(IU.EQ.1) THEN
           IF(JU.EQ.1) THEN
         AIJ =*BZ*((CB(5,1))*BBX+(CB(5,5))*BBZ+(CB(5,6))*BBY)
           ELSEIF(JU.EQ.2) THEN
         AIJ = BZ*((CB(5,2))*BBY+(CB(5,4))*BBZ+(CB(5,6))*BBX)
           ELSE
         AIJ = BZ*((CB(5,3))*BBZ+(CB(5,4))*BBY+(CB(5,5))*BBX)
           ENDIF
         ELSEIF(IU.EQ.2) THEN
           IF(JU.EQ.1) THEN
         AIJ = BZ*((CB(4,1))*BBX+(CB(4,5))*BBZ+(CB(4,6))*BBY)
           ELSEIF(JU.EQ.2) THEN
         AIJ = BZ*((CB(4,2))*BBY+(CB(4,4))*BBZ+(CB(4,6))*BBX)
           ELSE
         AIJ = BZ*((CB(4,3))*BBZ+(CB(4,4))*BBY+(CB(4,5))*BBX)
           ENDIF
         ELSE
           IF(JU.EQ.1) THEN
         AIJ = BX*((CB(5,1))*BBX+(CB(5,5))*BBZ+(CB(5,6))*BBY)+
               BY*((CB(4,1))*BBX+(CB(4,5))*BBZ+(CB(4,6))*BBY)
           ELSEIF(JU.EQ.2) THEN
         AIJ = BX*((CB(5,2))*BBY+(CB(5,4))*BBZ+(CB(5,6))*BBX)+
               BY*((CB(4,2))*BBY+(CB(4,4))*BBZ+(CB(4,6))*BBX)
           ELSE
         AIJ = BX*((CB(5,3))*BBZ+(CB(5,4))*BBY+(CB(5,5))*BBX)+
               BY*((CB(4,3))*BBZ+(CB(4,4))*BBY+(CB(4,5))*BBX)
           ENDIF
         ENDIF
                         A(IJ) = A(IJ)+AIJ*VOL
 3080          CONTINUE
 3070        CONTINUE
 3060      CONTINUE
 3040    CONTINUE
 3020   CONTINUE
 2000  CONTINUE

       RETURN
       END

       SUBROUTINE MULTT(C,A,B,N,M,L)
```

```
 3DFEM    FORTRAN  A1 11/19/90 22:43    F 80  2264 RECS 11/19/90 22:44 PAGE

       IMPLICIT DOUBLE PRECISION (A-H,O-Z)
       DIMENSION A(N,M),B(M,L),C(N,L)
       DO 10 I=1,N
       DO 10 J=1,L
       C(I,J)=0.
       DO 10 K=1,M
 10    C(I,J)=C(I,J)+A(I,K)*B(K,J)
       RETURN
       END

       SUBROUTINE SSPACE(A,B,MAXA,R,EIGV,TT,W,AR,BR,VEC,D,RTOLV,BUP,BLO,
                         BUPC,NN,NNM,NWK,NWM,NROOT,RTOL,NC,NNC,NITEM,
                         IFSS,IFPR,NSTIF,IOUT)
       IMPLICIT DOUBLE PRECISION (A-H,O-Z)
       DIMENSION A(NWK),B(NWM),R(NN,NC),TT(NN),W(NN),EIGV(NC),
                 D(NC),VEC(NC,NC),AR(NNC),BR(NNC),RTOLV(NC),BUP(NC),
                 BLO(NC),BUPC(NC)
       INTEGER MAXA(NNM)
 C     SET TOLERANCE FOR JACOBI ITERATION
       TOLJ = 0.000000000001D0
       ICONV = 0
       NSCH  = 0
       NSMAX = 12
       N1 = NC+1
       NC1 = NC-1
       DO 60 I=1,NC
 60    D(I) = 0.D0
       ND = NN/NC
       IF (NWM.GT.NN) GOTO 4
       J = 0
       DO 2 I=1,NN
       II = MAXA(I)
       R(I,1) = B(I)
       IF(B(I).GT.0) J=J+1
 2     W(I) = B(I)/A(II)
       IF (NC.LE.J) GOTO 16
       WRITE(*,1007)
       STOP
 4     DO 10 I=1,NN
       II = MAXA(I)
       R(I,1) = B(II)
 10    W(I) = B(II)/A(II)
 16    DO 20 J = 2,NC
       DO 20 I = 1,NN
 20    R(I,J)=0.D0
 C
       L = NN-ND
       DO 30 J = 2,NC
       RT = 0.D0
       DO 40 I = 1,L
       IF (W(I).LT.RT) GOTO 40
       RT = W(I)
       IJ = I
 40    CONTINUE
       DO 50 I = L,NN
```

```
      IF (W(I).LE.RT) GOTO 50
      RT = W(I)
      IJ = I
50    CONTINUE
      TT(J) = FLOAT(IJ)
      W(IJ) = 0.D0
      L = L-ND
30    R(IJ,J) = 1.D0
C     FACTORIZE MATRIX A INTO (L)*(D)*(L(T))
C
      ISH = 0
      CALL DECOMP (A,MAXA,NN,ISH,IOUT)
      NITE = 0
100   NITE = NITE + 1
      IF(IFPR.EQ.0) GOTO 90
      WRITE(*,1010) NITE
90    IJ = 0
      DO 140 J = 1,NC
      DO 120 K = 1,NN
120   TT(K) = R(K,J)
      CALL REDBAK (A,TT,MAXA,NN)
      DO 130 I = J,NC
      ART = 0.D0
      DO 140 K = 1,NN
140   ART = ART+R(K,I)*TT(K)
      IJ = IJ+1
130   AR(IJ) = ART
      DO 150 K = 1,NN
150   R(K,J) = TT(K)
110   CONTINUE
      IJ = 0
      DO 160 J = 1,NC
      CALL MULT (TT,B,R(1,J),MAXA,NN,NWM)
      DO 180 I = J,NC
      BRT = 0.D0
      DO 190 K = 1,NN
190   BRT = BRT+R(K,I)*TT(K)
      IJ = IJ+1
180   BR(IJ) = BRT
      IF(ICONV.GT.0) GOTO 160
      DO 200 K = 1,NN
200   R(K,J) = TT(K)
160   CONTINUE
      IF(IFPR.EQ.0) GOTO 320
      IND = 1
210   WRITE(*,1020)
      II = 1
      DO 300 I = 1,NC
      ITEMP = II+NC-I
      WRITE(*,1005) (AR(J),J=II,ITEMP)
300   II = II+N1-I
      WRITE(*,1030)
      II = 1
*
      DO 310 I = 1,NC
```

3DFEM FORTRAN A1 11/19/90 22:43 F 80 2264 RECS 11/19/90 22:44 PAGE

```
      ITEMP = II+NC-I
      WRITE(*,1005) (BR(J),J=II,ITEMP)
310   II = II+N1-I
      IF(IND.EQ.2) GOTO 350
C
320   CALL JACOBI (AR,BR,VEC,EIGV,W,NC,NNC,TOLJ,NSMAX,IFPR,IOUT)
C
      IF(IFPR.EQ.0) GOTO 350
      WRITE(*,1040)
      IND = 2
      GOTO 210
350   IS = 0
      II = 1
      DO 360 I = 1,NC1
      ITEMP = II+N1-I
      IF (EIGV(I+1).GE.EIGV(I)) GOTO 360
      IS = IS+1
      EIGVT = EIGV(I+1)
      EIGV(I+1) = EIGV(I)
      EIGV(I) = EIGVT
      BT = BR(ITEMP)
      BR(ITEMP) = BR(II)
      BR(II) = BT
      DO 370 K = 1,NC
      RT = VEC(K,I+1)
      VEC(K,I+1) = VEC(K,I)
370   VEC(K,I) = RT
360   II = ITEMP
      IF(IS.GT.0) GOTO 350
      IF(IFPR.EQ.0) GOTO 375
      WRITE(*,1035)
      WRITE(*,1006) (EIGV(I),I=1,NC)
375   DO 420 I = 1,NN
      DO 422 J = 1,NC
422   TT(J) = R(I,J)
      DO 424 K = 1,NC
      RT = 0.D0
      DO 430 L = 1,NC
430   RT = RT+TT(L)*VEC(L,K)
424   R(I,K) = RT
420   CONTINUE
      IF(ICONV.GT.0) GOTO 500
C
C     CHECK FOR CONVERGENCE OF EIGENVALUES
C
      DO 380 I = 1,NC
      DIF = DABS(EIGV(I)-D(I))
380   RTOLV(I) = DIF/EIGV(I)
      IF (IFPR.EQ.0) GOTO 385
      WRITE(*,1050)
      WRITE(*,1005) (RTOLV(I),I=1,NC)
C
385   DO 390 I = 1,NROOT
      IF(RTOLV(I).GT.RTOL) GOTO 400
390   CONTINUE
```

```
      WRITE (*,1060) RTOL
      ICONV = 1
      GOTO 100
400   IF (NITE.LT.NITEM) GOTO 410
      WRITE(*,1070)
      ICONV = 2
      IFSS = 0
      GOTO 100
C
410   DO 440 I = 1,NC
440   D(I) = EIGV(I)
      GOTO 100
500   WRITE(*,*) ' '
*     WRITE(*,1100)  ' '
      WRITE(*,1006) (EIGV(I),I=1,NROOT)
*     WRITE(*,1110)
      DO 530 J = 1,NROOT
530   WRITE(*,1005)  (R(K,J),K=1,NN)
      DO 580 L = 1,NROOT
      RT = EIGV(L)
      CALL MULT(TT,A,R(1,L),MAXA,NN,NWK)
      VNORM = 0.D0
*
      DO 590 I = 1,NN
590   VNORM = VNORM+TT(I)*TT(I)
      CALL MULT(W,B,R(1,L),MAXA,NN,NWM)
      WNORM = 0.D0
      DO 600 I = 1,NN
      TT(I) = TT(I) - RT*W(I)
600   WNORM = WNORM+TT(I)*TT(I)
      VNORM = DSQRT(VNORM)
      WNORM = DSQRT(WNORM)
      D(L) = WNORM/VNORM
580   CONTINUE
*     WRITE(*,1115)
      WRITE(*,1006) (D(I),I=1,NROOT)
      IF(IFSS.EQ.0) GOTO 700
      CALL SCHECK (EIGV,RTOLV,BUP,BLO,BUPC,D,NC,NEI,RTOL,SHIFT)
      IF(NWM.GT.NN) GOTO 645
      DO 640 I = 1,NN
      II = MAXA(I)
640   A(II) = A(II)-B(I)*SHIFT
      GOTO 660
645   DO 650 I = 1,NWK
650   A(I) = A(I)-B(I)*SHIFT
660   ISH = 1
      CALL DECOMP (A,MAXA,NN,ISH,IOUT)
      NSCH = 0
      DO 664  I = 1,NN
      II = MAXA(I)
      IF ( A(II).LT.0.D0) NSCH = NSCH+1
664   CONTINUE
      IF (NSCH.EQ.NEI) GOTO 670
      NMIS = NSCH-NEI
*     WRITE(*,1130) NMIS
```

3DFEM FORTRAN A1 11/19/90 22:43 F 80 2264 RECS 11/19/90 22:44 PAGE

```
      GOTO 700
670   WRITE(*,1140) NSCH
700   RETURN
C
1002  FORMAT (10F10.0)
1005  FORMAT (12E11.4)
1006  FORMAT (6E22.14)
1007  FORMAT (' STOP, NC IS LARGER THAN THE NUMBER OF MASS
     .          DEGREES OF FREEDOM')
1008  FORMAT (' DEGREES OF FREEDOM EXCITED BY UNIT STARTING ITERATION
     .          VECTORS')
1010  FORMAT (' I T E R A T I O N   N U M B E R ',I4)
1020  FORMAT (' PROJECTION OF A (MATRIX AR')
1030  FORMAT (' PROJECTION OF B (MATRIX BR')
1035  FORMAT (' EIGENVALUES OF AR-LAMBDA*BR ')
1040  FORMAT (' AR AND BR AFTER JACOBI DIAGONALIZATION' )
1050  FORMAT (' RELATIVE TOLERANCE REACHED ON EIGENVALUES ' )
1060  FORMAT (' CONVERGENCE REACHED FOR RTOL ',D10.4)
1070  FORMAT ('***** NO CONVERGENCE IN MAXIMUM NUMBER OF ITERATIONS
     .          - WE ACCEPT CURRENT ITERATION VALUES - THE STURM
     .            SEQUENCE CHECK IS NOT PERFORMED ')
1100  FORMAT (' THE CALCULATED EIGENVALUES ARE ')
1115  FORMAT (' PRINT ERROR NORMS ON THE EIGENVALUES ' )
1110  FORMAT (' THE CALCULATED EIGENVECTORS ARE ' )
1120  FORMAT (' CHECK APPLIED AT SHIFT ',D22.14)
1130  FORMAT (' THERE ARE',I4,2X,'EIGENVALUES MISSING')
1140  FORMAT (' WE FOUND THE LOWEST ',I4,2X,'EIGENVALUES')
C
444   END

      SUBROUTINE DECOMP (A,MAXA,NN,ISH,IOUT)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A(1),MAXA(1)
      IF(NN.EQ.1) RETURN
C
      DO 200 N = 1,NN
      KN = MAXA(N)
      KL = KN+1
      KU = MAXA(N+1)-1
      KH = KU-KL
      IF(KH) 304,240,210
210   K = N-KH
      IC = 0
      KLT = KU
      DO 260 J = 1,KH
      IC = IC+1
      KLT = KLT-1
      KI = MAXA(K)
      ND = MAXA(K+1)-KI-1
      IF(ND) 260,260,270
270   KK = MIN0(IC,ND)
      C = 0.D0
      DO 280 L = 1,KK
280   C = C+A(KI+L)*A(KLT+L)
      A(KLT) = A(KLT)-C
```

```
 260    K = K+1
 240    K = N
        B = 0.D0
        DO 300 KK =KL,KU
        K = K-1
        KI = MAXA(K)
        C = A(KK)/A(KI)
        IF (DABS(C).LT.1.D07) GOTO 290
        WRITE(*,2010) N,C
        STOP
 290    B = B+C*A(KK)
 300    A(KK) = C
        A(KN) = A(KN)-B
 304    IF(A(KN)) 310,310,200
 310    IF(ISH.EQ.0) GOTO 320
        IF(A(KN).EQ.0) A(KN)=-1.D-16
        GOTO 200
 320    WRITE(*,2000) N,A(KN)
        STOP
 200    CONTINUE
 C
        RETURN
 2000   FORMAT('STOP-STIFNESS MATRIX NOT POSITIVE DEFINITE,
       .         NONPOSITIVE PIVOT FOR EQUATION ',I4,3X,'PIVOT',
       .         D20.12)
 2010   FORMAT('STOP-STURM SEQUENCE CHECK FAILED BECAUSE OF
       .         MULTIPLIER GROWTH FOR COLUMN NO ',I4,3X,
       .        'MULTIPLIER  ',D20.8)
        END


        SUBROUTINE  REDBAK (A,V,MAXA,NN)
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
        DIMENSION A(1),V(1),MAXA(1)
 C
        DO 400 N=1,NN
        KL=MAXA(N)+1
        KU=MAXA(N+1)-1
        IF(KU-KL) 400,410,410
 410    K=N
        C=0.D0
        DO 420 KK=KL,KU
        K=K-1
 420    C=C+A(KK)*V(K)
        V(N)=V(N)-C
 400    CONTINUE
 C
        DO 480 N=1,NN
        K=MAXA(N)
 480    V(N)=V(N)/A(K)
        IF (NN.EQ.1) RETURN
        N=NN
        DO 500 L=2,NN
        KL=MAXA(N)+1
        KU=MAXA(N+1)-1
```

3DFEM FORTRAN A1 11/19/90 22:43 F 80 2264 RECS 11/19/90 22:44 PAGE

```
        IF(KU-KL) 500,510,510
 510    K=N
        DO 520 KK=KL,KU
        K=K-1
 520    V(K)=V(K)-A(KK)*V(N)
 500    N=N-1
 C
        RETURN
        END


        SUBROUTINE MULT(TT,B,RR,MAXA,NN,NWM)
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
        DIMENSION TT(1),B(10),RR(1),MAXA(1)
 C
        IF (NWM.GT.NN) GOTO 20
        DO 10 I=1,NN
 10     TT(I)=B(I)*RR(I)
        RETURN
 C
 20     DO 40 I=1,NN
 40     TT(I)=0.D0
        DO 100 I=1,NN
        KL=MAXA(I)
        KU=MAXA(I+1)-1
        II=I+1
        CC=RR(I)
        DO 100 KK=KL,KU
        II=II-1
 100    TT(II)=TT(II)+B(KK)*CC
        IF (NN.EQ.1) RETURN
        DO 200 I=2,NN
        KL=MAXA(I)+1
        KU=MAXA(I+1)-1
        IF (KU-KL) 200,210,210
 210    II=I
        AA=0.D0
        DO 220 KK=KL,KU
        II=II-1
 220    AA=AA+B(KK)*RR(II)
        TT(I)=TT(I)+AA
 200    CONTINUE
 C
        RETURN
        END

        SUBROUTINE SCHECK(EIGV,RTOLV,BUP,BLO,BUPC,NEIV,NC,NEI,
       .                  RTOL,SHIFT)
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
        DIMENSION EIGV(NC),RTOLV(NC),BUP(NC),BLO(NC),
       .          BUPC(NC),NEIV(NC)
 C
        FTOL=0.01D0
 C
```

```
      DO 100 I=1,NC
      BUP(I)=EIGV(I)*(1.D0+FTOL)
100   BLO(I)=EIGV(I)*(1.D0-FTOL)
      NROOT=0
      DO 120 I=1,NC
120   IF (RTOLV(I).LT.RTOL) NROOT=NROOT+1
      IF (NROOT.GE.1) GOTO 200
      WRITE(6,1010)
      STOP
200   DO 240 I=1,NROOT
240   NEIV(I)=1
      IF (NROOT.NE.1) GOTO 260
      BUPC(1)=BUP(1)
      LM=1
      L=1
      I=2
      GOTO 295
260   L=1
      I=2
270   IF (BUP(I-1).LE.BLO(I)) GOTO 280
      NEIV(L)=NEIV(L)+1
      I=I+1
      IF (I.LE.NROOT) GOTO 270
280   BUPC(L)=BUP(I-1)
      IF (I.GT.NROOT) GOTO 290
      L=L+1
      I=I+1
      IF (I.LE.NROOT) GOTO 270
      BUPC(L)=BUP(I-1)
290   LM=L
      IF(NROOT.EQ.NC) GOTO 300
*
295   IF (BUP(I-1).LE.BLO(I)) GOTO 300
      IF (RTOLV(I).GT.RTOL) GOTO 300
      BUPC(L)=BUP(I)
      NEIV(L)=NEIV(L)+1
      NROOT=NROOT+1
      IF (NROOT.EQ.NC) GOTO 300
      I=I+1
      GOTO 295
300   WRITE(6,1020)
      WRITE(6,1005) (BUPC(I),I=1,LM)
      WRITE(6,1030)
      WRITE(6,1006) (NEIV(I),I=1,LM)
      LL=LM-1
      IF (LM.EQ.1) GOTO 310
330   DO 320 I=1,LL
320   NEIV(L)=NEIV(L)+NEIV(I)
      L=L-1
      LL=LL-1
      IF (L.NE.1) GOTO 330
310   WRITE(6,1040)
      WRITE(6,1006) (NEIV(I),I=1,LM)
      L=0
      DO 340 I=1,LM
```



```
      L=L+1
      IF (NEIV(I).GE.NROOT) GOTO 350
340   CONTINUE
350   SHIFT=BUPC(L)
      NEI=NEIV(L)
1005  FORMAT(6D22.14)
1006  FORMAT(6I22)
1010  FORMAT('******* ERROR STOP IN *SCHECK*  NO EIG-VALUES FOUND')
1020  FORMAT('UPPER BOUNDS ON EIGENVALUE CLUSTERS')
1030  FORMAT('NO OF EIG-VALUES IN EACH CLUSTER')
1040  FORMAT('NO OF EIG-VALUES LESS THAN UPPER BOUNDS')
      END

      SUBROUTINE JACOBI(A,B,X,EIGV,D,N,NWA,RTOL,NSMAX,IFPR,IOUT)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A(NWA),B(NWA),X(N,N),EIGV(N),D(N)
      N1=N+1

      II=1
      DO 10 I=1,N
      IF(A(II).GT.0.D0.AND.B(II).GT.0.D0) GOTO 4
*     WRITE(*,*) 'HERE'
      WRITE(*,2020) II,A(II),B(II)
      STOP
4     D(I)=A(II)/B(II)
      EIGV(I)=D(I)
10    II=II+N1-I
      DO 30 I=1,N
      DO 20 J=1,N
20    X(I,J)=0.D0
30    X(I,I)=1.D0
      IF(N.EQ.1) RETURN
      NSWEEP=0
      NR=N-1
40    NSWEEP=NSWEEP+1
      IF(IFPR.EQ.1) WRITE(*,2000)NSWEEP
      EPS=(0.01D0**NSWEEP)**2.D0
      DO 210 J=1,NR
      JP1=J+1
      JM1=J-1
      LJK=JM1*N-JM1*J/2
      JJ=LJK+J
      DO 210 K=JP1,N
      KP1=K+1
*
      KM1=K-1
      JK=LJK+K
      KK=KM1*N-KM1*K/2+K
      EPTOLA=(A(JK)*A(JK))/(A(JJ)*A(KK))
      EPTOLB=(B(JK)*B(JK))/(B(JJ)*B(KK))
      IF ((EPTOLA.LT.EPS).AND.(EPTOLB.LT.EPS))GOTO 210
      AKK=A(KK)*B(JK)-B(KK)*A(JK)
      AJJ=A(JJ)*B(JK)-B(JJ)*A(JK)
      AB=A(JJ)*B(KK)-A(KK)*B(JJ)
      CHECK=(AB*AB+4.D0*AKK*AJJ)/4.D0
```

```
      IF(CHECK)50,60,60
*     WRITE(*,*) 'HERE'
 50   WRITE(*,2020)
      STOP
 60   SQCH=DSQRT(CHECK)
      D1=AB/2.D0+SQCH
      D2=AB/2.D0-SQCH
      DEN=D1
      IF(DABS(D2).GT.DABS(D1))DEN=D2
      IF(DEN)80,70,80
 70   CA=0.D0
      CG=-A(JK)/A(KK)
      GOTO 90
 80   CA=AKK/DEN
      CG=-AJJ/DEN
 90   IF(N-2)100,190,100
100   IF(JM1-1)130,110,110
110   DO 120 I=1,JM1
      IM1=I-1
      IJ=IM1*N-IM1*I/2+J
      IK=IM1*N-IM1*I/2+K
      AJ=A(IJ)
      BJ=B(IJ)
      AK=A(IK)
      BK=B(IK)
      A(IJ)=AJ+CG*AK
      B(IJ)=BJ+CG*BK
      A(IK)=AK+CA*AJ
120   B(IK)=BK+CA*BJ
130   IF(KP1-N)140,140,160
140   LJI=JM1*N-JM1*J/2
      LKI=KM1*N-KM1*K/2
      DO 150 I=KP1,N
      JI=LJI+I
      KI=LKI+I
      AJ=A(JI)
      BJ=B(JI)
      AK=A(KI)
      BK=B(KI)
      A(JI)=AJ+CG*AK
      B(JI)=BJ+CG*BK
      A(KI)=AK+CA*AJ
150   B(KI)=BK+CA*BJ
160   IF(JP1-KM1)170,170,190
170   LJI=JM1*N-JM1*J/2
      DO 180 I=JP1,KM1
      JI=LJI+I
      IM1=I-1
      IK=IM1*N-IM1*I/2+K
      AJ=A(JI)
      BJ=B(JI)
      AK=A(IK)
      BK=B(IK)
      A(JI)=AJ+CG*AK
      B(JI)=BJ+CG*BK
```

```
SDFEM    FORTRAN  A1 11/19/90 22:43    F 80  2264 RECS 11/19/90 22:44 PAGE

      A(IK)=AK+CA*AJ
180   B(IK)=BK+CA*BJ
190   AK=A(KK)
      BK=B(KK)
      A(KK)=AK+2.D0*CA*A(JK)+CA*CA*A(JJ)
      B(KK)=BK+2.D0*CA*B(JK)+CA*CA*B(JJ)
      A(JJ)=A(JJ)+2.D0*CG*A(JK)+CG*CG*AK
      B(JJ)=B(JJ)+2.D0*CG*B(JK)+CG*CG*BK
      A(JK)=0.D0
      B(JK)=0.D0
      DO 200 I=1,N
      XJ=X(I,J)
      XK=X(I,K)
      X(I,J)=XJ+CG*XK
200   X(I,K)=XK+CA*XJ
*
210   CONTINUE
      II=1
      DO 220 I=1,N
      IF (A(II).GT.0.D0.AND.B(II).GT.0.D0) GOTO 215
      WRITE(*,2020) II,A(II),B(II)
      STOP
215   EIGV(I)=A(II)/B(II)
220   II=II+N1-I
      IF(IFPR.EQ.0) GOTO 230
      WRITE(*,2030)
      WRITE(*,2010) (EIGV(I),I=1,N)
230   DO 240 I=1,N
      TOL=RTOL*D(I)
      DIF=DABS(EIGV(I)-D(I))
      IF(DIF.GT.TOL) GOTO 280
240   CONTINUE
      EPS=RTOL**2.D0
      DO 250 J=1,NR
      JM1=J-1
      JP1=J+1
      LJK=JM1*N-JM1*J/2
      JJ=LJK+J
      DO 250 K=JP1,N
      KM1=K-1
      JK=LJK+K
      KK=KM1*N-KM1*K/2+K
      EPSA=(A(JK)*A(JK))/(A(JJ)*A(KK))
      EPSB=(B(JK)*B(JK))/(B(JJ)*B(KK))
      IF((EPSA.LT.EPS).AND.(EPSB.LT.EPS)) GOTO 250
      GOTO 280
250   CONTINUE
255   II=1
      DO 275 I=1,N
      BB=DSQRT(B(II))
      DO 270 K=1,N
270   X(K,I)=X(K,I)/BB
275   II=II+N1-I
      RETURN
280   DO 290 I=1,N
```

```
290   D(I)=EIGV(I)
      IF(NSWEEP.LT.NSMAX) GOTO 40
      GOTO 255
2000  FORMAT('SWEEP NUMBER IN *JACOBI* = ',I4)
2010  FORMAT(6D20.12)
2020  FORMAT('*** ERROR SOLUTION STOP - MATRICES NOT POSITIVE
     .          DEFINITE','II=',I4,'A(II) ',D20.12,'B(II) ',D20.12)
2030  FORMAT('CURRENT EIGENVALUES IN *JACOBI* ARE ')
      END
```

VITA

Lazarus H. Tenek was born November 4, 1962 in Thessaloniki, Greece, to Mr. Harry and Mrs. Alexandra Teneketzis. After completing a few grades in Queens, New York, he went back to Greece to complete his high school and undergraduate education. He received a Bachelor of Science degree in Civil Engineering from Aristotle University in 1987. The same year, he came to the United States, and enrolled in the Mechanical and Aerospace department at Polytechnic University, Brooklyn, New York. He graduated with a Master of Science in 1989. He transferred to Virginia Polytechnic Institute and State University to pursue a Master of Science degree in the field of composite materials in Engineering Science and Mechanics department in January 1990. Upon completion of his M.S. degree at Virginia Tech, he will join the UVA-NASA Center for Computational Structures Technology at NASA Langley Research Center in Hampton, Virginia.

ABSTRACT

During vibrothermographic experimental testing of damaged composite plates, frequency dependent heat generation phenomena were observed. Local hot spots were formed around imperfection areas especially delaminations. Heat generation was also found to relate to the crack size. In order to explain the above observed phenomena, the dynamic behavior of undamaged and damaged composite plates was studied over a broad frequency range. The analysis was carried out using the finite element method based on the concepts of the three dimensional theory of anisotropic elasticity. Delaminations were modeled, and the 'local crack resonance' was justified. Two NDE methods namely, Vibrothermography and SPATE were used to verify the numerical predictions. Experiments performed for both undamaged and damaged specimens, and good correlation between theory and testing was achieved.