MULTILAYER BEAM ANALYSIS INCLUDING
SHEAR AND GEOMETRIC NONLINEAR EFFECTS

By

Herman Ho Ming Kwan

B.A.Sc., THE UNIVERSITY OF BRITISH COLUMBIA, 1985

A THESIS SUBMITTED IN PARTIAL FULFILLEMNT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE STUDIES

(CIVIL ENGINEERING DEPARTMENT)

We accept this thesis as conforming
to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

May 1987

2

32

Department of Civil Engineering

The University of British Columbia
1956 Main Mall
Vancouver, Canada
V6T 1Y3

Date April 30, 1987

Abstract

This thesis presents an analysis and experimental verification for a multilayer beam in bending.

The formulation of the theoretical analysis includes the combined effect of shear and geometric nonlinearity. From this formulation, a finite element program (CUBES) is developed.

The experimental tests were done on multilayer, corrugated paper beams. Failure deflections and loads are thus obtained. The experimental results are reasonably predicted by the numerical results. Based upon this comparison, a maximum compressive stress is determined for the tested beam.

Finally, design curves for the tested beam are drawn using the determined maximum compressive stress and the finite element program.

Contents

List of Tables

List of Figures

Acknowledgement

I would like to express my gratitude to my advisor, Dr. R. O. Foschi, for providing me with his invaluable assistance and guidance throughout the duration of the research period and the writing of this thesis.

The financial support from the Natural Sciences and Engineering Research Council of Canada is gratefully acknowledged. Also, the testing materials supplied by MacMillan Bloedel Limited is received with thanks.

Finally, I would like to thank the U.B.C. Civil Engineering Department staff, my friends and my parents for their assistance, advice, and encouragement in completing this thesis.

Chapter 1

Introduction

Beam bending analysis often neglects shear strains by assuming negligible shear contribution to deflection. However, this assumption is not always correct. For example, shear effects can be very important in bending of short, stocky members, glulam beams in high humidity environment, and multilayer beams with relatively weak layers in shear.

In addition, the geometric nonlinear effect (large deformation) in beam bending is important for the study of beam behaviour under the interaction of axial and lateral loads. Elastic deformation of the beam under such loading conditions requires that large displacement terms be added to the usual small deformation equations.

Beam analysis considering the effects of shear and large deformations is required for many applications. Combined action of axial and lateral forces on laminated beams, multilayer composite beams, or even corrugated cardboard beams are examples to which this analysis may be applied.

Many researchers have investigated the bending of multilayer beams with core layers weak in shear. Kao and Ross (1968) considered the variation of total energy in obtaining a system of equations which describes the bending behaviour of a multilayer beam. Shear strains from the core layers weak in shear, bending strains from the stiff face layers and in-plane displacements of the stiff layers were included in the strain energy calculation. Shear strain in each core is assumed to be constant over the thickness of the core. However, no shear stress continuity was imposed at the interfaces of the layers. The shear strains of the cores due to the in-plane displacements, were approximated as the rate of change of the in-plane displacements at the midplanes of the two stiff layers in contact ($\gamma_u = \frac{\Delta u}{\Delta t}$) as shown in Figure 1.1. Also, the normal stresses perpendicular to the span were ignored; thus, all points

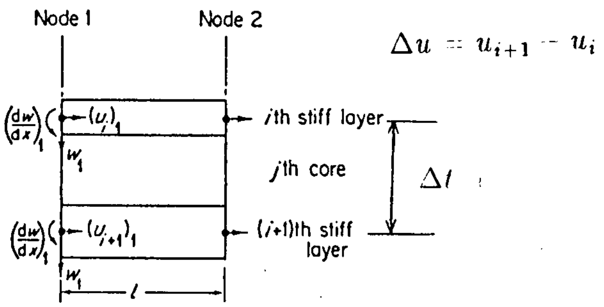

Figure 1.1: Kao's Shear Strain Approximation (Khatua and Cheung, 1973)

on the same cross-section were assumed to have the same lateral deflection.

Khatua and Cheung (1973) modified Kao's shear strain equation at each core to approximate the shear strains more accurately. The shear strains of the cores due to the in-plane displacements were then approximated as the rate of change of the in-plane displacements across the thickness of the weak cores. This different assumption was represented by a factor multiplied to Kao's shear strain equations. This factor was dependent on the thickness of the core and the adjacent stiff layers. In addition, a finite element approach was taken to formulate an approximate solution to the problem.

Further investigation by Foschi (1973) suggested that shear deformation in the stiff layers should be included in the analysis. Also, shear stress continuity should be imposed at the interfaces of the layers. Foschi proposed a piecewise shear strain function to model the shear strains over the thickness of the entire plate. Shear strains in the stiff layers were represented by a quadratic polynomial over the thickness of the layers; while, constant shear strains were assumed in the weak cores. However, in-plane displacements of the neutral plane ($z = 0$ plane) were neglected. Therefore, in the finite element formulation, the unknowns at each node were only the lateral deflection w, the slope w', and the shear strain γ at the midplane of each layer. In addition, symmetric cross-sections and small deformation were assumed in the analysis.

Putcha and Reddy (1986) proposed a higher order plate theory to account for the additional deformation contributed by shear strains. Thus, the normals to the midplane ($z = 0$ plane) before deformation were now neither straight nor normal to the midplanes. This higher order theory is advantageous over the Reissner-Mindlin first-order theory (Figure 1.2) because this theory satisfied the zero shear stresses conditions required on the top and the bottom faces of the plate. Also, the shear

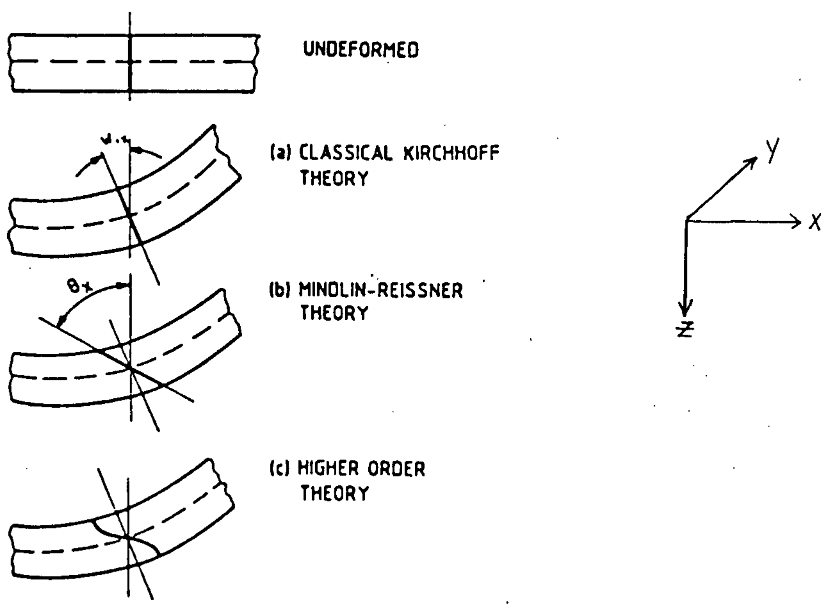

Figure 1.2: Cross-sectional Displacement for Three Bending Theories (Ren, 1986)

correction coefficient required in Reissner's theory was no longer necessary. In addition, geometric nonlinear effect was included in the strain equations. Finally, a mixed finite element model was formulated to approximate the theoretical solution. Each node of the element contained eleven degrees-of-freedom: 1. the usual u, v and w displacements, 2. rotations of the normals in the xz and yz planes (θ_x, θ_y) and 3. six moment resultants.

Ren and Hinton (1986) later modified Reddy's higher order theory to develop a new finite element to investigate the simpler problem of laminated plate without geometric nonlinearity. Ren replaced the the rotation of the normals' degrees-of-freedom with the shear strains at the $z = 0$ plane. The u and v displacement functions were modified to contain the unknown shear strains. However, the zero shear stress conditions were still maintained.

Di Sciuva (1986) proposed a'zig-zag displacement model' to account for shear effect on plate bending. This model incorporated a piecewise linear distribution of the in-plane, u and v, displacements. Shear stress continuity was imposed at the layers' interfaces. However, since the in-plane displacements were only linear, the shear strains in each layers were forced to be constant across the thickness of the layers. Thus, the model could not satisfy the zero shear stresses conditions on the top and the bottom faces of the plate. Therefore, shear stresses could not be directly obtained from the displacement model; instead, membrane stresses obtained from the model were substituted into the elasticity equilibrium equations to determine the shear stresses. The analysis also included the geometric nonlinearity terms in the strain equations. Finally, the model assumed that the plate's cross-sections were symmetric about the midplanes.

This thesis utilizes many of the ideas mentioned in the above review to study the behaviour of multilayer beams in bending. These ideas include:

1. Strain energy contribution from bending and shear from all layers,

2. Finite element method to formulate the approximate solution to the problem,

3. Negligible normal stresses perpendicular to the span,

4. Shear stress continuity between each layer,

5. Inclusion of geometric nonlinear terms for combined lateral and axial loads,

6. Piecewise linear shear strains across beam depth,

7. Zero shear stresses conditions on the top and bottom faces of the beam.

The finite element program CUBES is applied to the case of multilayer beams manufactured from corrugated paper. The theoretical solutions are compared to experimental results from bending tests of such beams. The program is also applied to the developement of strength interactions criterion when these corrugated beams are subjected to combined lateral and axial loads. Finally, a formulation to include material nonlinearity in shear is proposed.

Part I

General Theoretical Analysis

This section discusses the general formulation of the analysis. Chapters 2, 3, and 4 are included in this section. Chapter 2 presents the general formulation of the beam bending theory which includes shear effect, geometric nonlinear effect, and multilayer beams. Chapter 3 describes a finite element formulation to approximate a solution to the theoretical problem. Finally, chapter 4 presents several comparisons of results from the finite element program CUBES and other theoretical and numerical analyses.

Chapter 2

Formulation of Theory

A virtual work approach is taken to analyze the problem of bending of a multilayer beam. The general assumptions made in the analysis are first outlined. Kinematic relationships for strains and displacements are then developed. Finally, the governing equations are derived by applying the principle of virtual work.

2.1 General Assumptions

Several assumptions are made to simplify the analysis:

1. Small strains are assumed.

2. Normal stresses perpendicular to the beam span are ignored; hence, across the beam depth, the lateral deflection w is assumed to be the same for all layers.

3. Elastic material properties are assumed.

4. Homogeneous material properties are assumed for each layer.

5. Solid rectangular sections are assumed for all layers.

6. All layers in the beam are in constant contact with each other, thus no discontinuity exists between layers.

7. Out-of-plane warping of the beam is prevented.

8. Poisson effects are ignored.

A typical beam cross-section with the layers' number is shown in Figure 2.1.

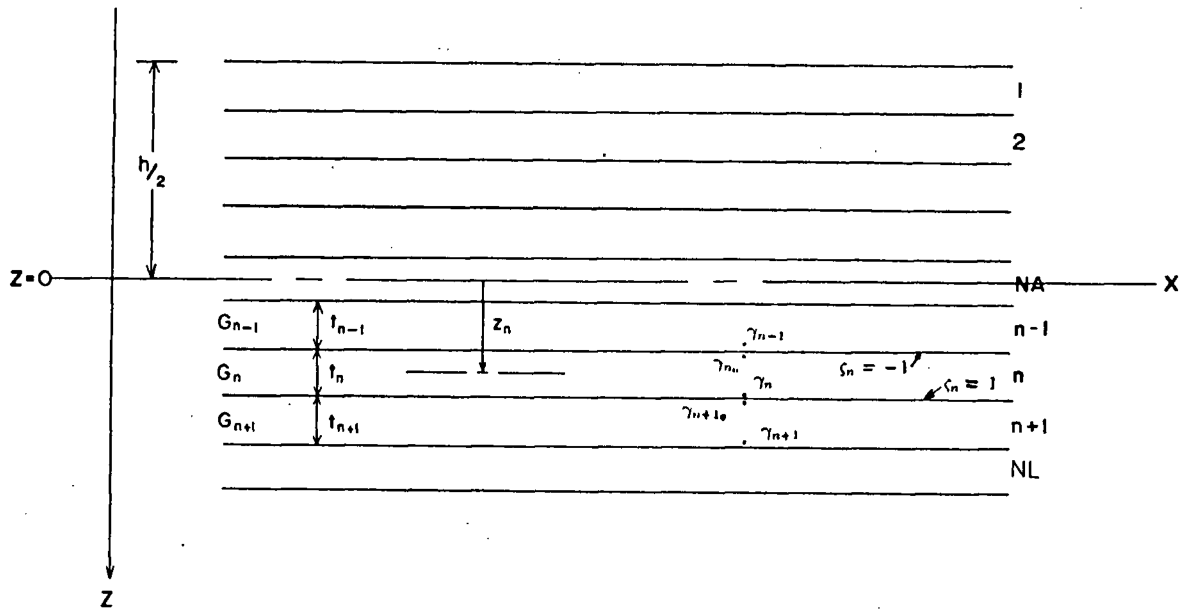

Figure 2.1: Side View of Typical Beam Layout

2.2 Kinematic Relationships

Beam bending analysis begins with the consideration of strain. The strain equations can be obtained by looking at a point P and its deflection in the beam (Figure 2.2).

Let us define a coordinate system (x, y, z) with origin at point O. The x-axis is parallel to the span and the plane $y - z$ is the plane of the beam cross-section at O. Consider now a point P on this cross-section. The vertical distance between points O and P is then equal to z. Also, the displacements u and w are defined to

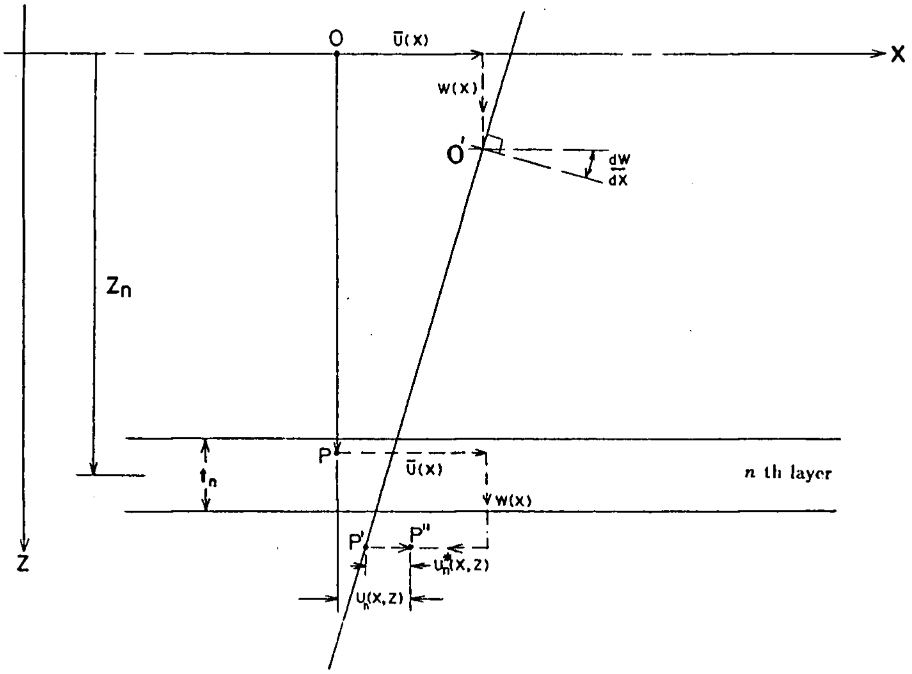

Figure 2.2: Deflection of Beam

correspond with the directions x and z. The lateral deflections $w(x)$ are assumed to be the same for all points on the same $y-z$ plane. The axial displacement on the neutral axis is defined as $\bar{u}(x)$. Referring to Figure 2.2, if shear deformation is neglected, point P will deflect to P$'$. However, if shear is included, a term $u^*(x,z)$ must be added to the small deformation theory's axial displacement equation. The final axial displacement $u(x,z)$ is thus expressed as

$$u(x,z) = \bar{u}(x) - z\frac{dw}{dx} + u^*(x,z) \tag{2.1}$$

The strains at any point are defined by the equations

$$\varepsilon_x = \frac{du}{dx} + \frac{1}{2}(\frac{dw}{dx})^2 \tag{2.2}$$

$$\gamma_{xz} = \frac{dw}{dx} + \frac{du}{dz}(1 + \frac{du}{dx}) \tag{2.3}$$

The last term in equation 2.2 is included because of the geometric nonlinearity (large deformation) consideration for the displacement w. The last term in equation 2.3 is approximated as $\frac{du}{dz}$ because $\frac{du}{dx}$ is approximated to be much smaller than 1. Substituting equation 2.1 into equations 2.2 and 2.3, the final strain equations become

$$\varepsilon_x = \frac{d\bar{u}}{dx} - z\frac{d^2w}{dx^2} + \frac{du^*}{dx} + \frac{1}{2}\left(\frac{dw}{dx}\right)^2 \tag{2.4}$$

$$\gamma_{xz} = \frac{dw}{dx} - \frac{dw}{dx} + \frac{du^*}{dz} = \frac{du^*}{dz} \tag{2.5}$$

2.3 Shear Strain Approximation

Given the layout of a NL-layers beam shown in Figure 2.1, it will be assumed that the shear strain γ_{xz} varies linearly over the thickness of each layer. Thus, two shear strain parameters per layer are needed to fully describe the shear strain distribution. The shear strain equation for the n-th layer can thus be expressed as

$$\gamma_n(\varsigma) = \gamma_n\frac{(1+\varsigma)}{2} + \gamma_{n_0}\frac{(1-\varsigma)}{2} \tag{2.6}$$

It should be noted that γ_n and γ_{n_0} are functions of the x-coordinate. The parameter ς is a non-dimensional local coordinate in the z direction. At the point of maximum z, ς equals to 1 in the layer while at minimum z, ς equals to -1.

In order to approximate the behaviour of continuous shear stress across the layers, the assumption of shear stress continuity is imposed. Thus, in addition to the virtual work requirement, the solution to the bending problem must satisfy the shear stress continuity constraint (see section2.6).

Now, with given shear moduli G_n and G_{n-1} for n-th and (n-1)-th layers, the assumption of shear stress continuity between layers gives

$$G_{n-1}\gamma_{n-1} = G_n\gamma_{n_0} \tag{2.7}$$

Substituing γ_{n_0} from equation 2.7 into equation 2.6, the shear strain in the n-th layer is finally expressed as

$$\gamma_n(\varsigma) = \gamma_n \frac{(1+\varsigma)}{2} + \frac{G_{n-1}}{G_n}\gamma_{n-1}\frac{(1-\varsigma)}{2} \tag{2.8}$$

Integrating the above equation with respect to z gives the shear distortion equation $u^*(x,z)$ at any point (x,z) in the beam. Thus, $u^*(x,z)$ is given by

$$u^*(x,z) = \int_0^z \gamma(x,s)ds = \int_{-h/2}^z \gamma(x,s)ds - \int_{-h/2}^0 \gamma(x,s)ds \tag{2.9}$$

Now, the global z coordinate in the n-th layer is transformed to the local coordinate ς. For the n-th layer, z_n is defined as the z value of the midplane ($\varsigma = 0$) and t_n is defined as the thickness. The following transformation equation is obtained.

$$z = z_n + \frac{t_n}{2}\varsigma \tag{2.10}$$

Equation 2.10 is then substituted into equation 2.9 which results in the following expression for the n-th layer's shear distortion.

$$\begin{aligned} u_n^*(x,\varsigma) &= \sum_{i=1}^{n-1}\left[\frac{t_i}{2}\int_{-1}^{1}\gamma_i(x,\varsigma)d\varsigma\right] + \frac{t_n}{2}\int_{-1}^{\varsigma}\gamma_n(x,\varsigma)d\varsigma \\ &\quad + \sum_{i=1}^{NA}\left[\frac{t_i}{2}\int_{-1}^{1}\gamma_i(x,\varsigma)d\varsigma\right] \\ &\quad - \frac{t_{NA}}{2}\int_{-1}^{LOCAL}\gamma_{NA}(x,\varsigma)d\varsigma \end{aligned} \tag{2.11}$$

The number LOCAL is defined as the local coordinate value in the NA-th layer where the plane $z = 0$ is located; thus, NA is defined as the layer number for the point O. The last two terms in equation 2.11 are constants in z and are determined by the location of the $z = 0$ plane. But these two terms are functions of the x-coordinate. Finally, using equation 2.8, equation 2.11 can be changed to give the shear distortion $u_n^*(x,z)$ at any point (x,z) in the n-th layer of the beam as

$$u_n^*(x,z) = \sum_{i-1}^{n} \gamma_i(x) F(n,i) - \sum_{i}^{NA} \gamma_i(x) F^*(n,i) \tag{2.12}$$

where

$$F(n,i) = \frac{t_i}{2} + \frac{G_i t_{i+1}}{2G_{i+1}}(1 - \Delta(n-i)) + \frac{G_i t_{i+1}}{2G_{i+1}}(\frac{\varsigma}{2} - \frac{\varsigma^2}{4} + \frac{3}{4})\Delta(n-i)$$

for $i = 1, 2, \ldots, n-1$

$$F(n,n) = \frac{t_n}{2}(\frac{\varsigma}{2} + \frac{\varsigma^2}{4} + \frac{1}{4})$$

for $i = n$

$$F^*(NA,i) = \frac{t_i}{2} + \frac{G_i t_{i+1}}{2G_{i+1}}(1 - \Delta(NA-i))$$

$$+\frac{G_i t_{i+1}}{2G_{i+1}}(\frac{LOCAL}{2} - \frac{LOCAL^2}{4} + \frac{3}{4})\Delta(NA-i)$$

for $i = 1, 2, \ldots, NA-1$

$$F^*(NA,NA) = \frac{t_{NA}}{2}(\frac{LOCAL}{2} + \frac{LOCAL^2}{4} + \frac{1}{4})$$

for $i = NA$

and $\Delta(l) = 0$ if $l > 1$ and $\Delta(l) = 1$ if $l = 1$

where l is the argument of the Δfunction

2.4 Stress-strain Relationship

The material properties expressed in terms of E and G are assumed to be elastic for each layer. Poisson effects are ignored. The bending and shear stress-strain relationships in the n-th layer can be expressed as

$$\sigma_{x_n} = E_{x_n} \varepsilon_{x_n} \tag{2.13}$$

$$\tau_{xz_n} = G_{xz_n}\gamma_{xz_n} \tag{2.14}$$

or

$$\begin{aligned}
\{\sigma(n)\} &= \left\{ \begin{array}{c} \sigma_{x_n} \\ \tau_{xz_n} \end{array} \right\} \\
&= \left[\begin{array}{cc} E_{x_n} & 0 \\ 0 & G_{xz_n} \end{array} \right] \left\{ \begin{array}{c} \varepsilon_{x_n} \\ \gamma_{xz_n} \end{array} \right\} \\
&= [D(n)]\{\varepsilon(n)\}
\end{aligned} \tag{2.15}$$

Another nonlinear effect not considered in the formulation of this analysis is the material nonlinearity. However, this effect can be important if the stress-strain curve deviates greatly from the linear approximation during deformation. This topic, specifically for the shear moduli, is discussed in Appendix B.

2.5 Virtual Work Equation

Knowing the stress-strain relationship, we must now determine a governing equation in order to find the unknown deflections. Applying a virtual displacement $\{\delta a\}$ to the system, the resulting external and internal work (δW and δU) done by the forces and the stresses in the system are respectively given as

$$\begin{aligned}
\delta U &= \int_V \{\delta\varepsilon\}^T\{\sigma\}dV \\
\delta W &= \int_V \{\delta a\}^T\{F\}dV
\end{aligned} \tag{2.16}$$

Equating the external and internal work done by the system gives the virtual work equation of

$$\int_V \{\delta\varepsilon\}^T\{\sigma\}dV = \int_V \{\delta a\}^T\{F\}dV \tag{2.17}$$

2.6 Effect of Shear Stress Continuity Constraint

In addition to the virtual work requirement of equating the external and internal work done, a constraint of shear stress continuity between layers was imposed. Thus, a 'constrained' virtual work equation would be derived which would satisfy the requirements of equating the work done and imposing the shear stress continuity. Such a constrained set of equations normally complicate the solution as the number of unknown parameters usually increases (Zienkiewicz, 1979). However, this shear stress continuity constraint actually simplifies the solution by reducing the number of unknowns in the problem.

Because the set of shear stress continuity equations are simple linear equations, the constraint can be directly substituted into the virtual work equation. The substitution of the shear stress equations would thus eliminate the shear stress continuity constraint. Assumed displacement interpolations can now be applied to solve the bending problem.

In our case, the unconstrained virtual work equation would contain two shear strain parameters from each layer in addition to the displacement parameters of u, w, and their derivatives. By applying the shear stress constraint as shown in equation 2.7, the displacement parameters of shear strain from each layer would be reduced from two to one parameter per layer in the 'constrained' virtual work equation. This equation can now be solved using finite element method with assumed interpolations for the unknown displacements.

Chapter 3

Finite Element Formulation

The finite element method is used to obtain approximate, numerical solutions to the set of governing equations derived from the concept of virtual work. A beam element with two end nodes is used in the formulation. Local coordinate ξ is used in each element (Figure 3.1) along the x-axis. Each element has two end nodes, n and n+1, separated by the length Δx. The x-coordinate at ξ equals to -1 of the n-th element is defined as x_n.

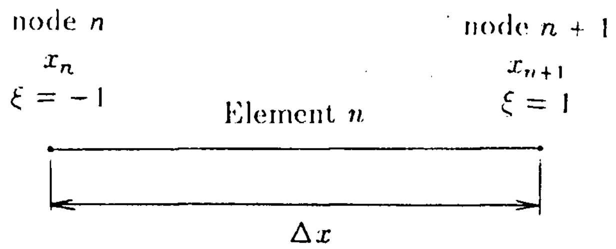

Figure 3.1: n-th Finite Element in x-coordinate

Thus, the x-coordinate of any point in the n-th element can be expressed as

$$x = x_n + \frac{\Delta x}{2}(\xi + 1) \tag{3.1}$$

and the differential, dx is

$$dx = \frac{\Delta x}{2} d\xi \tag{3.2}$$

therefore

$$\frac{d\xi}{dx} = \frac{2}{\Delta x} \tag{3.3}$$

The displacements u and w and their 1-st derivatives $u'(=\frac{du}{d\xi})$ and $w'(=\frac{dw}{d\xi})$ are specified as the nodal degrees of freedom (DOF) for the interpolations. In addition to the four DOFs mentioned above, the nodal displacement vector for each node contains every layer's shear strain parameter at the node. Therefore, for a NL-layers beam, each node will have (4+NL) degree of freedoms per node. The degree of freedom of each element can then be assembled into a column vector called the displacement vector a. This vector takes the form:

$$\{a\} = \left\{ \begin{array}{c} w_n \\ w'_n \\ \bar{u}_n \\ \bar{u}'_n \\ \gamma_{1_n} \\ \gamma_{2_n} \\ \vdots \\ \gamma_{NL_n} \\ w_{n+1} \\ w'_{n+1} \\ \bar{u}_{n+1} \\ \bar{u}'_{n+1} \\ \gamma_{1_{n+1}} \\ \gamma_{2_{n+1}} \\ \vdots \\ \gamma_{NL_{n+1}} \end{array} \right\} \tag{3.4}$$

3.1 Interpolations

Complete cubic interpolations are used to approximate the displacements u and w within any element. It should be noted that, according to the compatibility requirement, displacement u needs only be a linear interpolation. However, during program implementation, a cubic interpolation was found to give a much improved approximation of the axial stresses. A linear interpolation is used to approximate the shear strain along x. Also, as previously described, the shear strain in each layer is approximated with a linear interpolation along z.

For a complete cubic interpolation, four constant parameters are needed to define the function. The displacement and its 1-st derivative at the two nodes provide sufficient parameteres to fully describe a cubic interpolation. The displacements $\bar{u}_n$ and w_n can thus be expressed as

$$\begin{aligned}\bar{u}_n(\xi) &= \bar{u}_n\left[1-3\left(\frac{\xi+1}{2}\right)^2+2\left(\frac{\xi+1}{2}\right)^3\right]+\Delta x\,\bar{u}'_n\left[\left(\frac{\xi+1}{2}\right)^1\right.\\ &\left.-2\left(\frac{\xi+1}{2}\right)^2+\left(\frac{\xi+1}{2}\right)^3\right]+\bar{u}_{n+1}\left[\left(\frac{\xi+1}{2}\right)^2-3\left(\frac{\xi+1}{2}\right)^3\right]\\ &+\Delta x\,\bar{u}'_{n+1}\left[\left(\frac{\xi+1}{2}\right)^3-\left(\frac{\xi+1}{2}\right)^2\right]\end{aligned} \tag{3.5}$$

$$\begin{aligned}w_n(\xi) &= w_n\left[1-3\left(\frac{\xi+1}{2}\right)^2+2\left(\frac{\xi+1}{2}\right)^3\right]+\Delta x\,w'_n\left[\left(\frac{\xi+1}{2}\right)^1\right.\\ &\left.-2\left(\frac{\xi+1}{2}\right)^2+\left(\frac{\xi+1}{2}\right)^3\right]+w_{n+1}\left[\left(\frac{\xi+1}{2}\right)^2-3\left(\frac{\xi+1}{2}\right)^3\right]\\ &+\Delta x\,w'_{n+1}\left[\left(\frac{\xi+1}{2}\right)^3-\left(\frac{\xi+1}{2}\right)^2\right]\end{aligned} \tag{3.6}$$

Linear shear interpolations in both x and z direction are similar in form. For the x direction, the interpolation is

$$\gamma_{i_n}(\xi,\varsigma)=\frac{\gamma_{i_n}(\varsigma)}{2}(1-\xi)+\frac{\gamma_{i_{n+1}}(\varsigma)}{2}(1+\xi) \tag{3.7}$$

where γ_{i_n} is shear strain at the i-th layer of the n-th node and $\gamma_{i_{n+1}}$ is shear strain at the i-th layer of the (n+1)-th node. For the z direction, the interpolation has already taken form in the previous section of shear strain approximation (eq. 2.6).

The generalized displacements ($u, w,$ and γ) can now be represented in vectorial form. The $\bar{u}$-displacement can be written as a function of $\{a\}$ in the following vectorial form

$$\bar{u}(\xi)=\{N1(\xi)\}^T\{a\} \tag{3.8}$$

where the vector $\{N1(\xi)\}$ is

$$\{N1(\xi)\} = \left\{ \begin{array}{c} 0 \\ 0 \\ \left[1 - 3\left(\frac{\xi+1}{2}\right)^2 + 2\left(\frac{\xi+1}{2}\right)^3\right] \\ \Delta x\left[\left(\frac{\xi+1}{2}\right) - 2\left(\frac{\xi+1}{2}\right)^2 + \left(\frac{\xi+1}{2}\right)^3\right] \\ 0 \\ \vdots \\ 0 \\ \left[3\left(\frac{\xi+1}{2}\right)^2 - 2\left(\frac{\xi+1}{2}\right)^3\right] \\ \Delta x\left[\left(\frac{\xi+1}{2}\right)^3 - \left(\frac{\xi+1}{2}\right)^2\right] \\ 0 \\ \vdots \\ 0 \end{array} \right\} \begin{array}{c} \text{term} \\ 1\ st \\ 2\ nd \\ 3\ rd \\ 4\ th \\ 5\ th \\ \vdots \\ NL+6\ th \\ NL+7\ th \\ NL+8\ th \\ NL+9\ th \\ \vdots \\ 2NL+8\ th \end{array} \tag{3.9}$$

Similarily, the displacement w can be expressed as

$$w(\xi) = \{M(\xi)\}^T\{a\} \tag{3.10}$$

where $\{M(\xi)\}$ is written as

$$\{M(\xi)\} = \left\{ \begin{array}{c} \left[1 - 3\left(\frac{\xi+1}{2}\right)^2 + 2\left(\frac{\xi+1}{2}\right)^3\right] \\ \Delta x\left[\left(\frac{\xi+1}{2}\right) - 2\left(\frac{\xi+1}{2}\right)^2 + \left(\frac{\xi+1}{2}\right)^3\right] \\ 0 \\ 0 \\ \vdots \\ 0 \\ \left[3\left(\frac{\xi+1}{2}\right)^2 - 2\left(\frac{\xi+1}{2}\right)^3\right] \\ \Delta x\left[\left(\frac{\xi+1}{2}\right)^3 - \left(\frac{\xi+1}{2}\right)^2\right] \\ 0 \\ 0 \\ \vdots \\ 0 \end{array} \right\} \begin{array}{c} \text{term} \\ 1\ st \\ 2\ nd \\ 3\ rd \\ 4\ th \\ \vdots \\ NL+4\ th \\ NL+5\ th \\ NL+6\ th \\ NL+7\ th \\ NL+8\ th \\ \vdots \\ 2NL+8\ th \end{array} \tag{3.11}$$

Finally, the linear shear interpolation in ξ can also be expressed as a product of vectors

$$\gamma_i(\xi) = \{N3_i(\xi)\}^T\{a\} \tag{3.12}$$

with the vector $\{N3_i(\xi)\}$ equals to

$$\{N3_i(\xi)\} = \begin{array}{cc} & \text{term} \\ \left\{\begin{array}{c} 0 \\ 0 \\ \vdots \\ 0 \\ \left(\frac{1-\xi}{2}\right) \\ 0 \\ 0 \\ \vdots \\ 0 \\ \left(\frac{\xi+1}{2}\right) \\ \vdots \\ 0 \end{array}\right\} & \begin{array}{c} 1\ st \\ 2\ nd \\ \vdots \\ i+3\ th \\ i+4\ th \\ i+5\ th \\ i+6\ th \\ \vdots \\ NL+i+7\ th \\ NL+i+8\ th \\ \vdots \\ 2NL+8\ th \end{array} \end{array} \tag{3.13}$$

In addition, derivatives of the displacements can be similarily expressed in the vector form. However, noting that the strain equations require derivatives with respect to the global coordinates x and z, the following equations are necessary to relate the derivatives in different coordinates.

$$\begin{aligned} \frac{du^*}{dz} &= \frac{du^*}{d\varsigma}\frac{2}{t_n} \\ \frac{d\bar{u}}{dx} &= \frac{d\bar{u}}{d\xi}\frac{2}{\Delta x} \\ \frac{dw}{dx} &= \frac{dw}{d\xi}\frac{2}{\Delta x} \end{aligned} \tag{3.14}$$

3.2 Virtual Work Equations

Given equations 3.4 to 3.14, the strain equations (eqs. 2.4 and 2.5) for the n-th layer can now be expressed as

$$\begin{aligned}
\varepsilon_{x_n}(\xi,\varsigma) &= \left\{\frac{2}{\Delta x}\frac{d\{N1\}}{d\xi} - \left(z_n + \frac{t_n}{2}\varsigma\right)\frac{4}{\Delta x^2}\frac{d^2\{M\}}{d\xi^2} + \right.\\
&\left.\sum_{i=1}^{n}\frac{2}{\Delta x}F(n,i)\frac{d\{N3_i\}}{d\xi} - \sum_{i=1}^{NA}\frac{2}{\Delta x}F^*(NA,i)\frac{d\{N3_i\}}{d\xi}\right\}^T\{a\}\\
&+\frac{1}{2}\frac{4}{\Delta x^2}\{a\}^T\frac{d\{M\}}{d\xi}\frac{d\{M\}}{d\xi}^T\{a\}\\
&= \{KX(\xi,\varsigma)\}^T\{a\} + \frac{1}{2}\{a\}^T[MX(\xi,\varsigma)]\{a\}
\end{aligned} \tag{3.15}$$

$$\begin{aligned}
\gamma_{xz_n}(\xi,\varsigma) &= \left\{\left(\frac{1+\varsigma}{2}\right)\{N3_n\} + \frac{G_{n-1}}{G_n}\left(\frac{1-\varsigma}{2}\right)\{N3_{n-1}\}\right\}^T\{a\}\\
&= \{KXZ(\xi,\varsigma)\}^T\{a\}
\end{aligned} \tag{3.16}$$

Therefore, the strain equations are now expressed as functions of the displacement vector $\{a\}$ and each strain contains two components: linear and nonlinear terms. Also, the strain vector $\{\varepsilon\}$ can be defined as

$$\{\varepsilon(\xi,\varsigma)\} = \left\{\begin{array}{c}\varepsilon_{x_n}(\xi,\varsigma)\\ \gamma_{xz_n}(\xi,\varsigma)\end{array}\right\} = [B_0(n)]\{a\} + [B_1(n)]\{a\} \tag{3.17}$$

where $[B_0(n)]$ is the linear component

$$[B_0(n)] = \left[\begin{array}{c}\{KX\}^T\\ \{KXZ\}^T\end{array}\right] \tag{3.18}$$

and $[B_1(n)]$ is the nonlinear component

$$[B_1(n)] = \left[\begin{array}{c}\frac{1}{2}\{a\}^T[MX]\\ \{0\}^T\end{array}\right] \tag{3.19}$$

Referring back to equation 2.17 of the virtual work equation section, (section 2.5) it can be seen that virtual strain equations are now needed in setting up the system of equations to be solved. Applying a virtual displacement to the strain equations, the virtual strain equations becomes

$$\{\delta\varepsilon(\xi,\varsigma)\} = \left\{\begin{array}{c}\delta\varepsilon_{x_n}(\xi,\varsigma)\\ \delta\gamma_{xz_n}(\xi,\varsigma)\end{array}\right\} = [B_0(n)]\{\delta a\} + [B_2(n)]\{\delta a\} \tag{3.20}$$

where $[B_0(n)]$ is the previously defined linear component and $[B_2(n)]$ is a nonlinear term which is written as

$$[B_2(n)] = \begin{bmatrix} \{a\}^T[MX] \\ \{0\}^T \end{bmatrix} = 2[B_1(n)] \tag{3.21}$$

Combining equations 2.15, 2.17, 3.17, and 3.20, the virtual energy equation can now be expressed as

$$\begin{aligned} \delta\Pi &= \{\delta a\}^T \int_V \Big(\big[[B_0(n)]^T + [B_2(n)]^T\big] \\ & [D(n)] \big[[B_0(n)] + [B_1(n)]\big] \{a\} - \{F\}\Big)\, dV \end{aligned} \tag{3.22}$$

Taking out the virtual displacement $\{\delta a\}$ and setting the above equation to zero ($\delta\Pi = 0$) , the equation takes the final form of

$$\begin{aligned} \{F\} &= \Big[\int_V \big[[B_0(n)]^T + [B_2(n)]^T\big] [D(n)] \big[[B_0(n)] \\ & + [B_1(n)]\big] dV\Big] \{a\} \\ & [K(n)]a \end{aligned} \tag{3.23}$$

where the right hand side of the equation excluding $\{a\}$ can be symbolized as

$$[K(n)] = \int_V \big[[B_0(n)]^T + [B_2(n)]^T\big] [D(n)] \big[[B_0(n)] + [B_1(n)]\big] dV \tag{3.24}$$

with $[K(n)]$ being defined as the elemental structural stiffness matrix for the n-th layer. Expressing in local coordinate ξ and ς, the same matrix becomes

$$\begin{aligned} [K(n)] &= \frac{t_n}{2}\frac{\Delta x}{2}\Delta y \int_{-1}^{1} d\varsigma \int_{-1}^{1} \big[[B_0(n)]^T + [B_2(n)]^T\big] [D(n)] \\ & \big[[B_0(n)] + [B_1(n)]\big] d\xi \end{aligned} \tag{3.25}$$

where Δy is the beam width. This matrix is determined for every layer in each element. The elemental structural stiffness matrices are then assembled into the global structural stiffness matrix by summing across the beam depth and span. Because second order effect of geometric nonlinearity is included in the analysis, an iteration scheme is needed to solve the system of equations. The standard Newton-Raphson method discussed in the next section is used to solve the problem.

3.3 Newton-Raphson Method

The Newton-Raphson method is a commonly used technique to solve nonlinear equations (Zienkiewicz, 1979). This method uses a first order approximation technique to solve the equations through iterations. The first order approximation is stated as

$$\{\Phi(a+\Delta a)\} = \{\Phi(a)\} + \left[\frac{d\Phi(a)}{d\{a\}}\right]\{\Delta a\} \tag{3.26}$$

where $\{\Phi\}$ is a function of the displacement vector $\{a\}$. Equation 3.26 can be re-arranged to become

$$\{\Delta a\} \quad = \quad \{\{\Phi(a+\Delta a)\} - \{\Phi(a)\}\}\left[\frac{d\{\Phi(a)\}}{d\{a\}}\right]^{-1} \tag{3.27}$$

The value $\{\Delta a\}$ can then be compared against an acceptable tolerance to determine whether further iteration is needed to obtain a sufficiently accurate solution $\{a\}$.

From equation 3.27, it is obvious that $\left[\frac{d\{\Phi(a)\}}{d\{a\}}\right]$ has to be determined before the Newton-Raphson method is used. In our case, let

$$\begin{aligned} \{\delta\Pi\} &= \{\Phi(a)\} \\ &= \left[\int_V \left[[B_0(n)]^T + [B_2(n)]^T\right][D(n)] \times \right. \\ &\quad \left. \left[\,[B_0(n)] + [B_1(n)]\,\right]dV\right]\{a\} - \{F\} \end{aligned} \tag{3.28}$$

where $\{\Phi(a)\}$ is now defined as the column vector $\{\delta\Pi\}$. The solution to the finite element problem is finding the displacement vector $\{a\}$ which results in $\{\Phi(a)\}$ equal to or near zero. However, since the equations are nonlinear, a direct solution to the equations was not possible.

Remembering that the matrix $[B_0]$ and forces $\{F\}$ are not functions of $\{a\}$, the 1-st derivative $\left[\frac{d\{\Phi(a)\}}{d\{a\}}\right]$ can be obtained by differentiating equation 3.28 with respect to $\{a\}$ and using equationd 3.20. The resulting equation is

$$\left[\frac{d\{\Phi(a)\}}{d\{a\}}\right] = \int_V \frac{d[B_2]^T}{d\{a\}}\{\sigma\} + \left[[B_0]^T + [B_2]^T\right] \frac{d\{\sigma\}}{d\{a\}} dV \tag{3.29}$$

Also, from the stress equation (eq. 2.15), the above term of $\frac{d\{\sigma\}}{d\{a\}}$ is determined as

$$\left[\frac{d\{\sigma\}}{d\{a\}}\right] = [D]\left[[B_0] + [B_2]\right] \tag{3.30}$$

In addition, after some special manipulation (see Appendix A), the term $\frac{d[B_2]^T}{d\{a\}}\{\sigma\}$ is found to equal

$$\left[\frac{d[B_2]^T}{d\{a\}}\{\sigma\}\right] = \frac{4}{\Delta x^2}\sigma_x[MX] \tag{3.31}$$

for our beam element. Substituting equations 3.30 and 3.31 into equation 3.29, $\frac{d\{\Phi(a)\}}{d\{a\}}$ then takes the final form of

$$\left[\frac{d\{\Phi(a)\}}{d\{a\}}\right] = \int_V \sigma_x[MX] + \left[[B_0]^T + [B_2]^T\right][D]\left[[B_0] + [B_2]\right] dV = [K_t] \tag{3.32}$$

Finally, this term $\left[\frac{d\{\Phi(a)\}}{d\{a\}}\right]$ is called the tangential stiffness matrix $[K_t]$. Noting that both terms on the right-hand-side of equation 3.32 are symmetric matrices, computer storage can be minimized by storing the matrices in vector form.

With the matrix $[K_t]$ known, the following solution scheme is used to determine the approximate solution vector $\{a_{m+1}\}$ in the (m+1)-th iteration.

$$\begin{aligned} \{\Phi_m(a)\} + [K_{t_m}](\{a_{m+1}\} - \{a_m\}) &= \{0\} \\ [K_{t_m}]\{a_{m+1}\} &= [K_{t_m}]\{a_m\} - \{\Phi_m(a)\} \\ \{a_{m+1}\} &= \{a_m\} - [K_{t_m}]^{-1}\{\Phi_m(a)\} \end{aligned} \tag{3.33}$$

The solution scheme is repeated until the displacement vector $\{\Delta a\} < \{tol\}$.

3.4 Method of Computation

The Cholesky decomposition method of solution is used to determine the vector $\{\Delta a\}$. Symmetry and bandwidth are considered in storing the matrices. Before applying the Cholesky method, boundary conditions are first applied to $[K_t]$ and $\{\Phi(a)\}$. For zero displacement boundary conditions, zeros are placed into the off-diagonal row and column of the specified degree of freedom in $[K_t]$ while a zero is placed into the same DOF of the $\{\Phi(a)\}$ vector. Also, the value 1 is placed into the diagonal term of the specified DOF in the $[K_t]$ matrix. Finally, the following procedures from the Cholesky method are used to solve the system of equations.

1. Decompose $[K_t]$
2. Solve $\{\Delta a\} = [K_t]^{-1} \{\Phi(a)\}$

3.5 Numerical Integration

Since the integrals in the previous sections are very complicated, closed form solution is difficult to obtain; therefore, numerical integration is used. Gaussian quadrature scheme is applied because the scheme is most suitable with the local coordinate variation of -1 to 1. Full integration is used to compute the integrals.

The maximum order of polynomial appearing in the integrals will determine the number of Gaussian points necessary to accurately integrate the function. The term $[B_2]$ contains a 4-th order polynomial in ξ and $[B_2]$ is then squared in the $[K_t]$ matrix. Thus, the highest order polynomial in the integrals is of 8-th order. Knowing that a n points Gaussian integration will integrate exactly a (2n-1)-th order polynomial, a 5-point Gaussian scheme is used for ξ in the numerical integration. Following the same procedure, a 3-points Gaussian scheme is determined to be necessary for the ς coordinate. Therefore, the integral can be represented as

	point	locations	weights
5-points	1	-0.9061798459	0.2369268850
integration	2	-0.5384693101	0.4786286705
	3	0.0000000000	0.5688888889
	4	0.5384693101	0.4786286705
	5	0.9061798459	0.2369268850
3-points	1	-0.7745966692	0.5555555556
	2	0.0000000000	0.8888888889
	3	0.7745966692	0.5555555556

Table 3.1: Locations and Weights of Integration Points

$$\int_{-1}^{1}\int_{-1}^{1} F(\xi,\varsigma)d\xi d\varsigma = \sum_{i=1}^{n}\sum_{j=1}^{m} F(\xi_i,\varsigma_j)W_iW_j \tag{3.34}$$

where $F(\xi,\varsigma)$ is any function in the coordinates (ξ,ς). The locations and weights for a 5-point and a 3-point Gaussian scheme are given in Table 3.1.

Since numerical integration is used, bending and shear stresses are computed only at the Gaussian points of each element. If stresses at any other point are desired, approximation such as stresses interpolation can be used.

3.6 Consistent Load Vector

Because finite element method is used in the analysis, consistent load vector must be used if the applied load is to be represented exactly. The consistent load representation for a concentrated load is simply the load value placed in the specific degree of freedom. However, the consistent load representation for a distributed load is somewhat more complicated. The consistent load vector is represented by the following equation

$$\{Q\} = \int_V q(\xi)\{M(\xi)\}d\xi \tag{3.35}$$

where $q(\xi)$ is the load function and $\{M(\xi)\}$ is the shape function within an element. In this program, only uniformly and linearly distributed loads within an element are accounted for. For example, assuming an element is subjected to a laterally distributed load which varies linearly from q_1 to q_2, the consistent load vector is determined by the following procedure.

The load function $q(\xi)$ within the element is

$$q(\xi) = q_1 \left(\frac{1-\xi}{2}\right) + q_2 \left(\frac{1+\xi}{2}\right) \tag{3.36}$$

Substituting this equation into equation 3.35 and performing the integration, the consistent load vector becomes

$$\{Q\} = \begin{Bmatrix} \frac{7}{20}\Delta x q_1 + \frac{3}{20}\Delta x q_2 \\ \frac{\Delta x^2}{20} q_1 + \frac{\Delta x^2}{30} q_2 \\ \frac{3}{20}\Delta x q_1 + \frac{7}{20}\Delta x q_2 \\ \frac{-\Delta x^2}{30} q_1 - \frac{\Delta x^2}{20} q_2 \end{Bmatrix} \tag{3.37}$$

A simple check can be done on this equation by determining the better known, uniformly distributed case. Equating $q_1 = q_2$, the consistent load vector of equation 3.37 is

$$\{Q\} = \begin{Bmatrix} \frac{1}{2}\Delta x q_1 \\ \frac{\Delta x^2}{12} q_1 \\ \frac{1}{2}\Delta x q_1 \\ \frac{-\Delta x^2}{12} q_1 \end{Bmatrix} \tag{3.38}$$

which corresponds with the uniformly distributed load case.

Chapter 4

Program Verification - Comparison with Previous Results

The finite element program CUBES was developed based upon the formulation discussed in chapters 2 and 3. The program's results were compared to referenced results; the two sets of results were in close agreement with each other. For the geometric nonlinearity effect, two results given by Timoshenko and Woinowsky-Krieger (1959) and one given by Popov (1968) were compared to the program's numerical solutions. For the shear effect, Popov's theoretical solutions and Foschi's numerical solutions were compared to CUBES' numerical results.

4.1 Geometric Nonlinearity

Two beam problems from Timoshenko were used to verify the program's result. The first problem considered was a simply supported beam with supports at a fixed distance apart subjected to an uniformly distributed load. Maximum deflections of the beam under different load value were calculated using Timoshenko's theory and the program CUBES. The analytical and numerical results plotted in Figure 4.1 were almost identical.

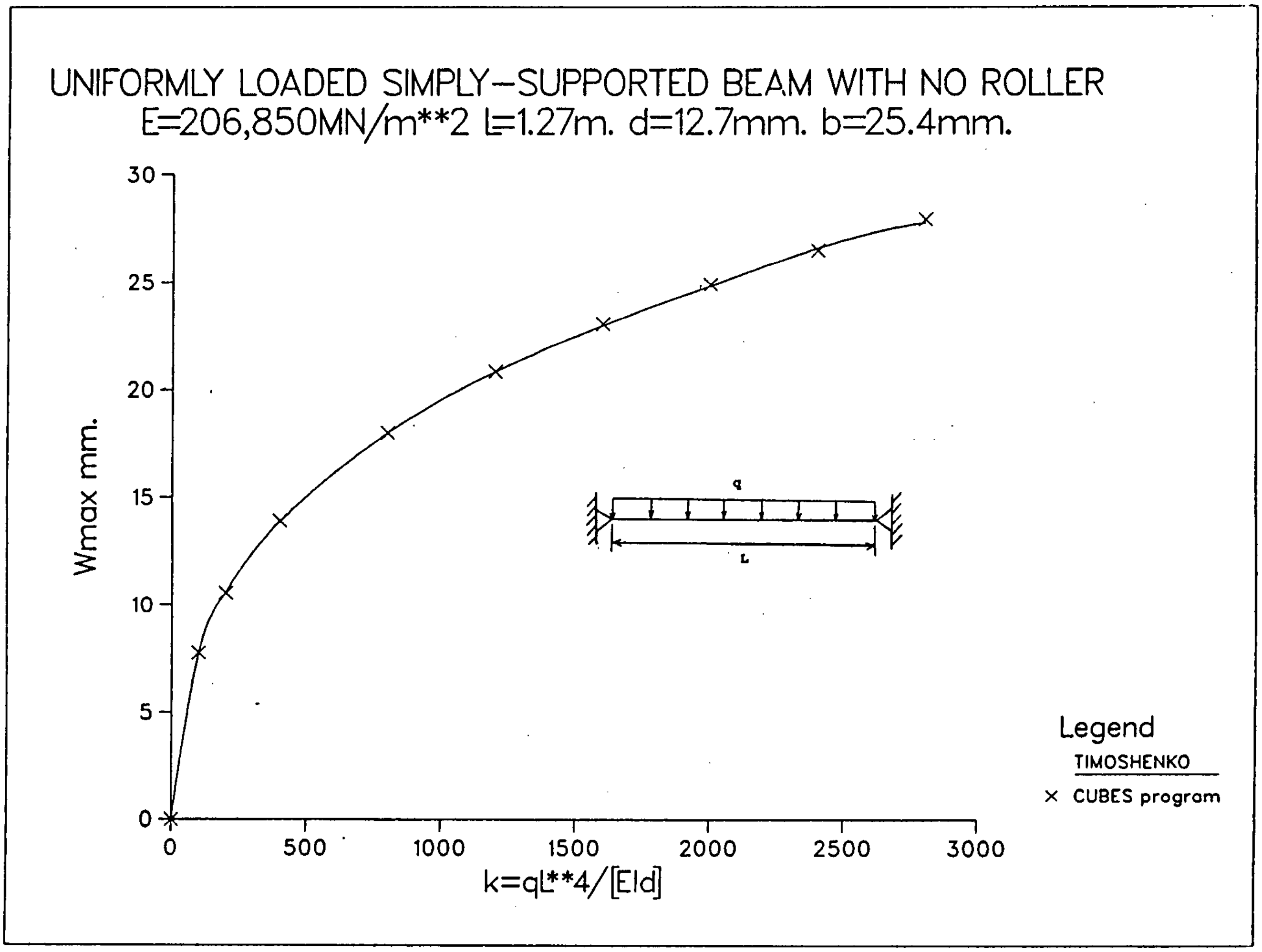

Figure 4.1: Comparison of Simply Supported Beam Problem

The second problem of deflection of a fixed ends beam under uniformly distributed load was also determined using Timoshenko's theory and the program CUBES. The results for this case are shown in Figure 4.2. The program's approximate solutions were also in close agreement with Timoshenko's theoretical solutions.

Finally, CUBES' results for the buckling of a simply supported beam under combined lateral and axial loads were compared to Popov's theoretical solutions. The results are shown in Figure 4.3.

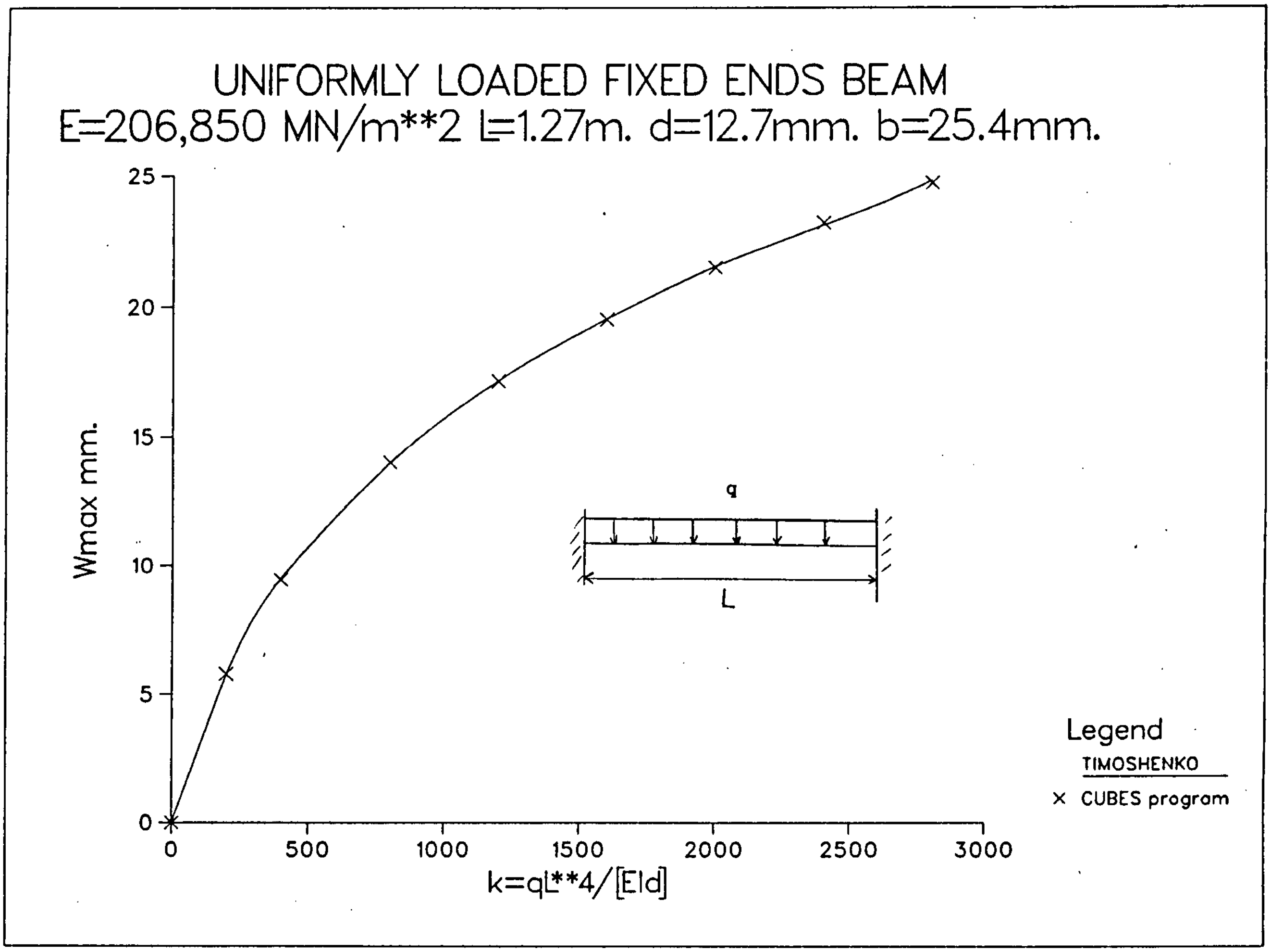

Figure 4.2: Comparison of Fixed Ends Beam Problem

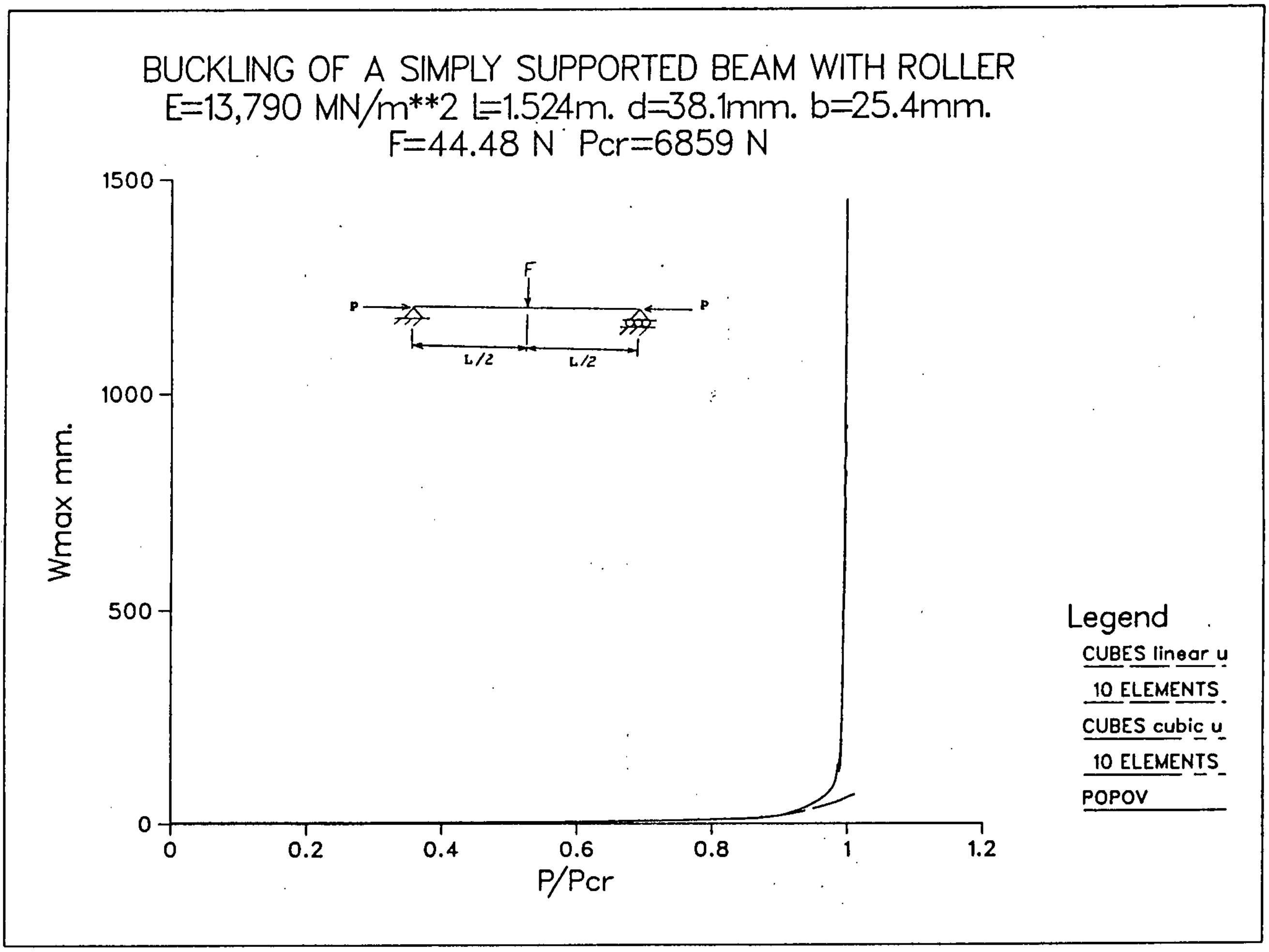

Figure 4.3: Comparison of Buckling Problem

As mentioned in chapter 3, a cubic u interpolation was chosen over a linear u interpolation in the program. For small number of elements used, the results of the linear u interpolation near critical load converge to the theoretical buckling solutions very slowly; thus, a cubic interpolation of u was used to improve the convergence. As shown in Figure 4.3, this cubic u interpolation approximated the buckling results quite accurately even when small number of elements (ten elements along span) were used. All three of the above comparisons indicated that the program can accurately approximate the geometric nonlinear effect.

4.2 Shear

For shear effect verification, a cantilever problem from Popov was used to compare with CUBES's results. Popov assumed a parabolic shear strain distribution across the beam depth. To include shear in CUBES's result, the beam was divided into equivalent fictitous layers thus approximating the shear strains with a piecewise linear distribution. The free end deflections were computed using two different approximating distribution: a 2-layers and a 4-layers approximation. It should be noted that, except for $\bar{u}_x$, all the degrees-of-freedom were specified as 0's at the fixed end of the cantilever. Figure 4.4 shows the results of the two analyses.

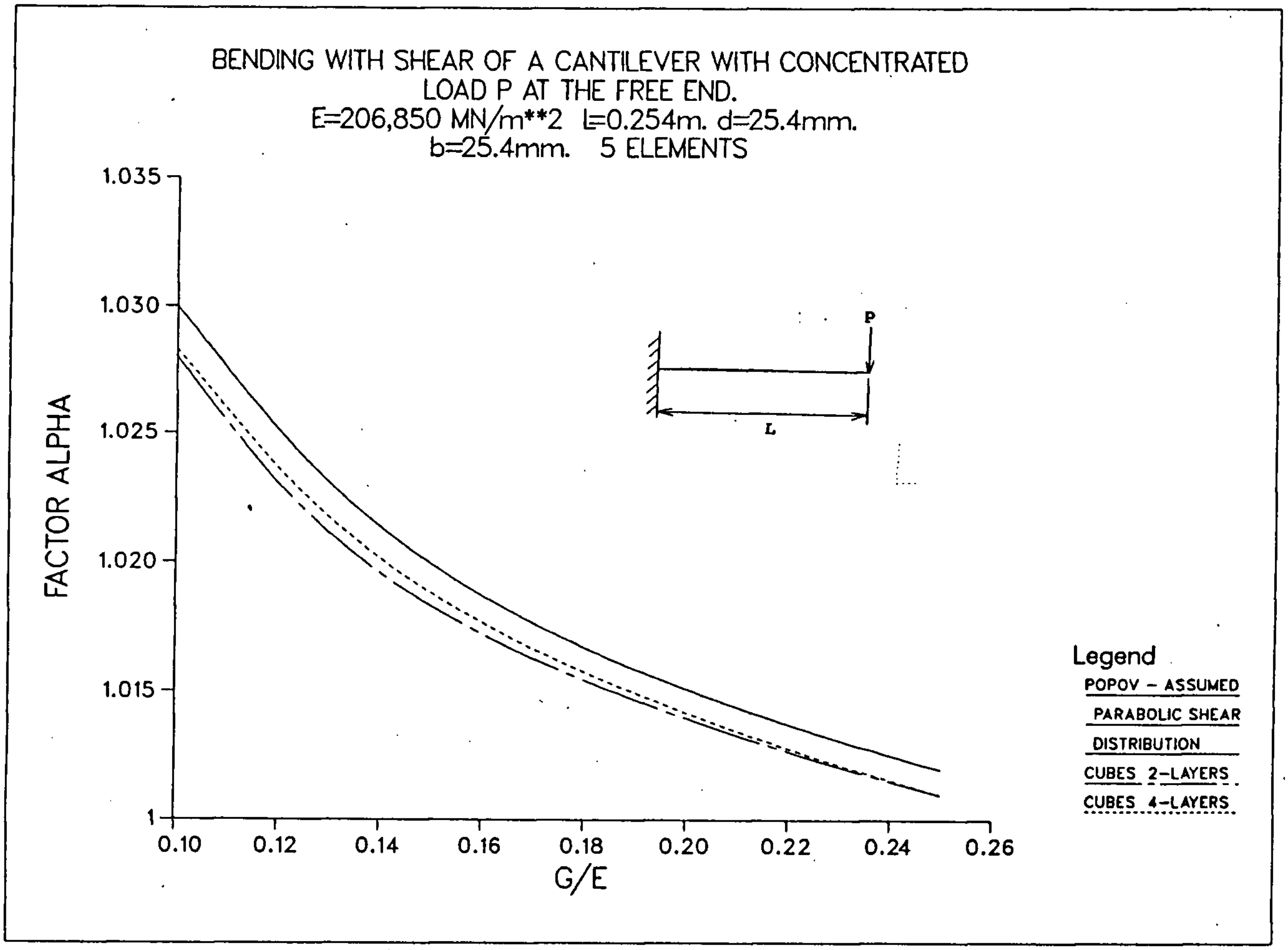

Figure 4.4: Comparison of Cantilever-Shear Problem

The parameter α was defined as the ratio of total deflection with shear over flexural deflection. CUBES' predicted results were quite close to the theoretical results. Also, the piecewise shear strains distribution approaches the parabolic distribution with greater number of fictitous layers. Figure 4.4 shows the comparison of the results.

Finally, deflections, including shear contribution, of a three-span simply-supported beam were determined using the program CUBES. These results were compared and found to be near Foschi's results (1973) as shown in Figure 4.5.

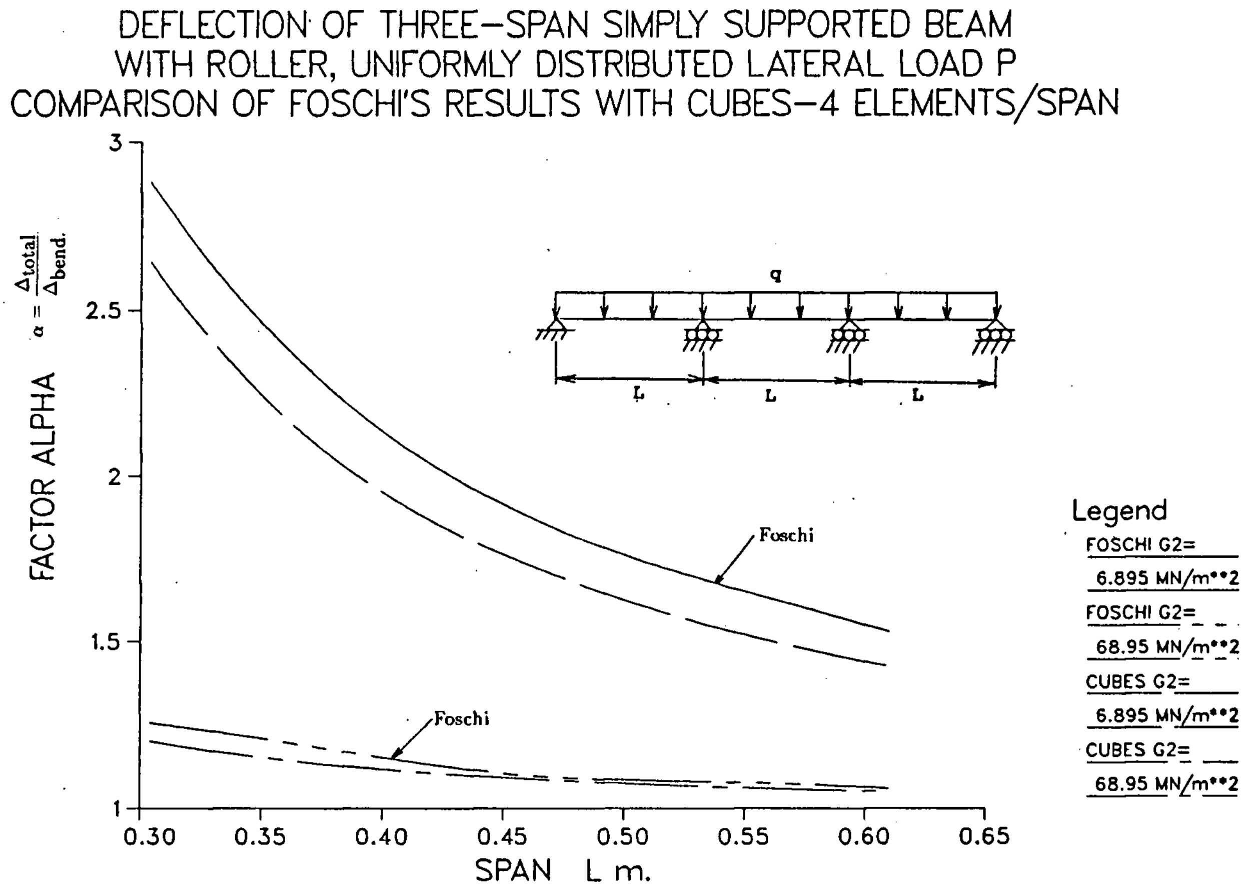

Figure 4.5: Comparison of Three-Span Beam Problem

Four elements per span were used in the program to obtain CUBES' results shown in Figure 4.5. Only the linear results from the program CUBES were used in the comparison since Foschi's formulation did not include geometric nonlinear effect. The results from CUBES were not identical to Foschi's results because Foschi assumed a constant shear strain in the weak cores and quadratic shear strain in the stiff layers. This assumption was different with our present assumption of linear shear strains in all layers. Also, Foschi's formulation ignored $\bar{u}$, the axial displacement of the $z = 0$ plane.

The above two comparison with Popov and Foschi indicated that the program's shear approximations were also accurate. Therefore, combined with the geometric nonlinear approximations, the program could readily approximate a solution to the bending of a multilayer beam with shear and geometric nonlinear effects included.

Part II

An Application: Multilayer Corrugated Paper Beam

This section focus on a specific application of the proposed theory: the bending of multilayer corrugated paper beams. Chapters 5 and 6 are included in this section. Chapter 5 describes the setup of a one-third point bending test of the paper beams and compares the experimental results to the program's numerical results. Chapter 6 shows a number of loads interaction design curves obtained from the program CUBES.

Chapter 5

Experiment

A bending test was performed on multilayer corrugated beams. Results from this test were compared to the numerical results from the program CUBES. A strength failure criterion for the specific outer liner was then determined based upon the above comparison.

5.1 Experimental Description and Results

A one-third point bending test was done to determine the midspan deflection of the tested beams. The beams were simply supported at both ends and were allowed to move axially (see Figure 5.1).

Each beam was made from nine-layers of paper: one layer of 90-lb paper liner, four layers of 42-lb paper liner, and four layers of 26C paper corrugation. Each layer had orthotropic properties. The corrugation was machined in one direction only; thus, two directions were defined for the corrugated layer: 1. machine direction and 2. cross-machine direction. Machine direction (Figure 5.2) was the direction of the approximate sinusoidal waves while cross-machine direction was the direction perpendicular to the machine direction in the $x - y$ plane.

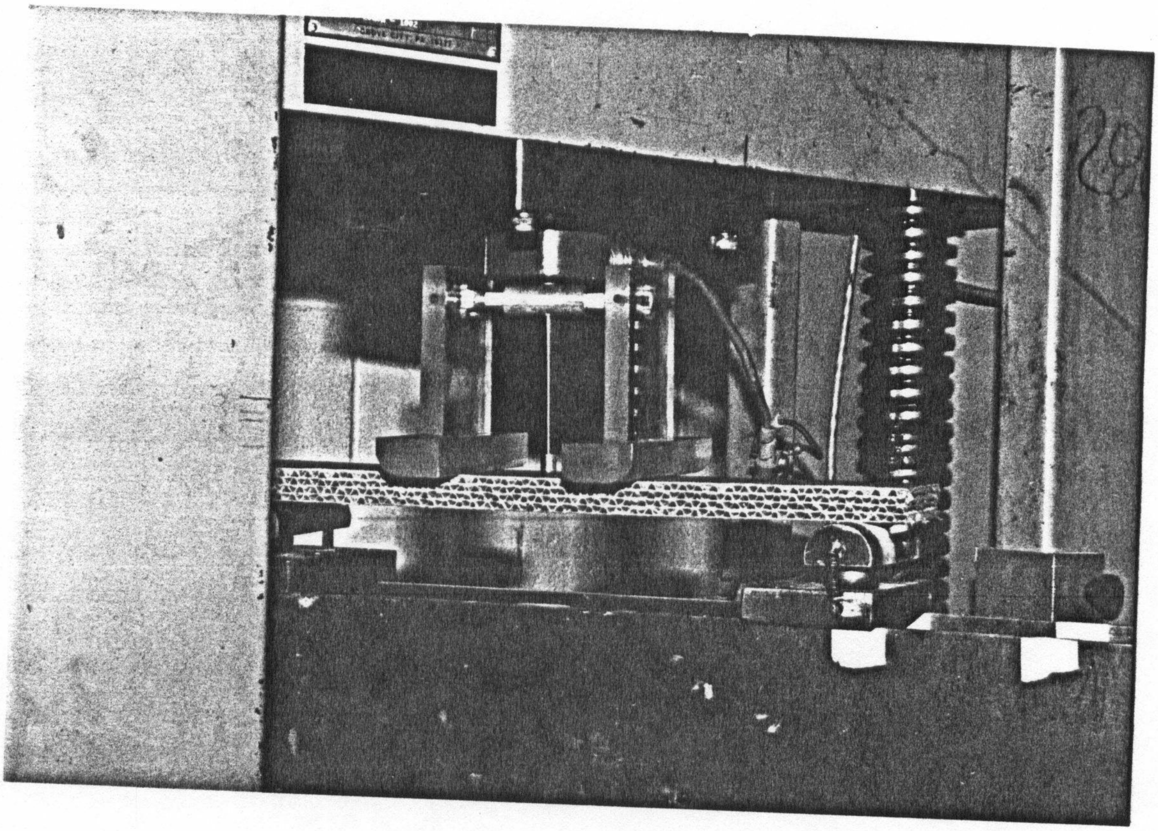

Figure 5.1: Experimental Setup

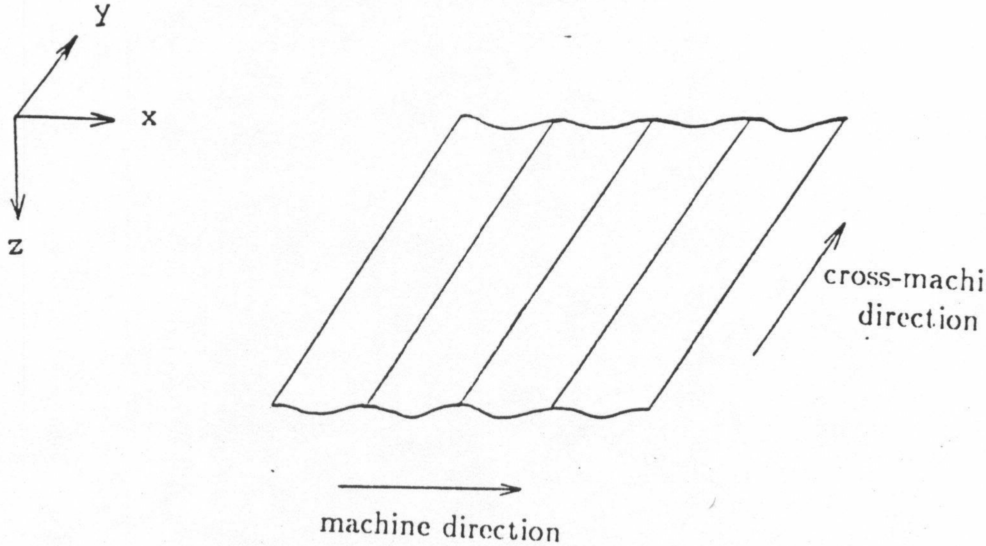

Figure 5.2: Direction of Corrugation

The thickness of the paper was measured and listed in Table 5.1. The amplitude f and period L_c of the corrugation were respectively measured as 3.58mm. and 7.80mm..

Material Type	Thickness (mm.)
90-lb	0.635
42-lb	0.305
26C	0.203

Table 5.1: Thickness of Paper

The test was done in the U.B.C. materials lab on the SATEC machine. The relative humidity and temperature of the testing environment was measured daily and were respectively found to be around fifty percent (50%) and sixty-eight degree Fahrenheit (68°F). Also, before testing, each beam was placed overnight in the testing room to reach atmospheric equilibrium. The load was measured with a 444.8 N (100 lb.) load cell and applied at a displacement controlled rate of approximately 7.62mm./min.. This load cell accurately measured the applied loads of the tests which were in the 88.96-355.84 N (20-80 lb.) range. The midspan displacement was measured using a LVDT.

Eighteen-layers beams were made from gluing two nine-layers boards together. After applying glue to both sides of the interface, the boards were pressed together and left to bond overnight. Constant pressure of approximately 13.79 kN/m^2 (2.0 lb/in^2) was applied to the beam during the bonding period. The glued board was then checked to ensure that no crushing had occured from the applied pressure.

Six different tests were done on the cardboard beams. They are listed in Table 5.2. The material properties of each of the tested beam are given in Table 5.3.

Test #	Test Direction	Layer Arrangement	Sample Size
1	machine	90-26C-42	17
2	machine	90-26C-42-26C-90	11
3	machine	90-26C-42-90-26C-42	2
4	machine	90-26C-42	10
5	machine	42-26C-90	4
6	cross-machine	90-26C-42	2

Table 5.2: List of Tests Performed

Test #	Beam Span (m.)	No. of Layers	Beam Depth d_b (mm.)
1	0.6	9	16.28
2	0.6	18	32.56
3	0.6	18	32.56
4	0.3	9	16.28
5	0.3	9	16.28
6	0.9	9	16.28

Table 5.3: Tests Properties

Figure 5.3 and 5.4 respectively show a front and a cross-section view of a 90-26C-42 beam tested in machine direction.

The load-deflection curve was recorded during each test (Figure 5.5). Each beam was considered to have failed when a compressive crease had fully developed in the outer liner at the compression side of the beam (Figure 5.6). The complete development of this crease could also be seen in the load-deflection curve (see Figure 5.5) as a significant sudden decrease of the applied load . This definition of failure was applied to bending in both machine and cross-machine directions.

The averages and standard deviations of the experimental failure loads and deflections for all the tests are given in Table 5.4.

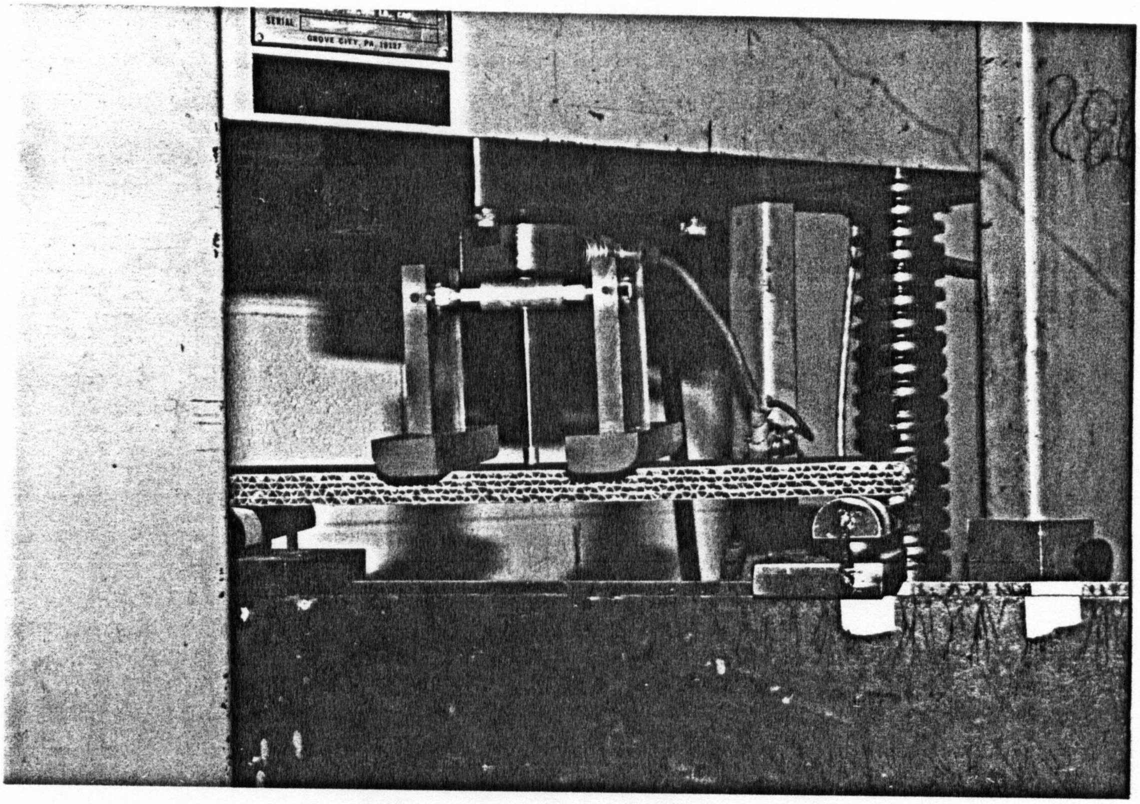

Figure 5.3: Front View of Beam Tested in Machine Direction

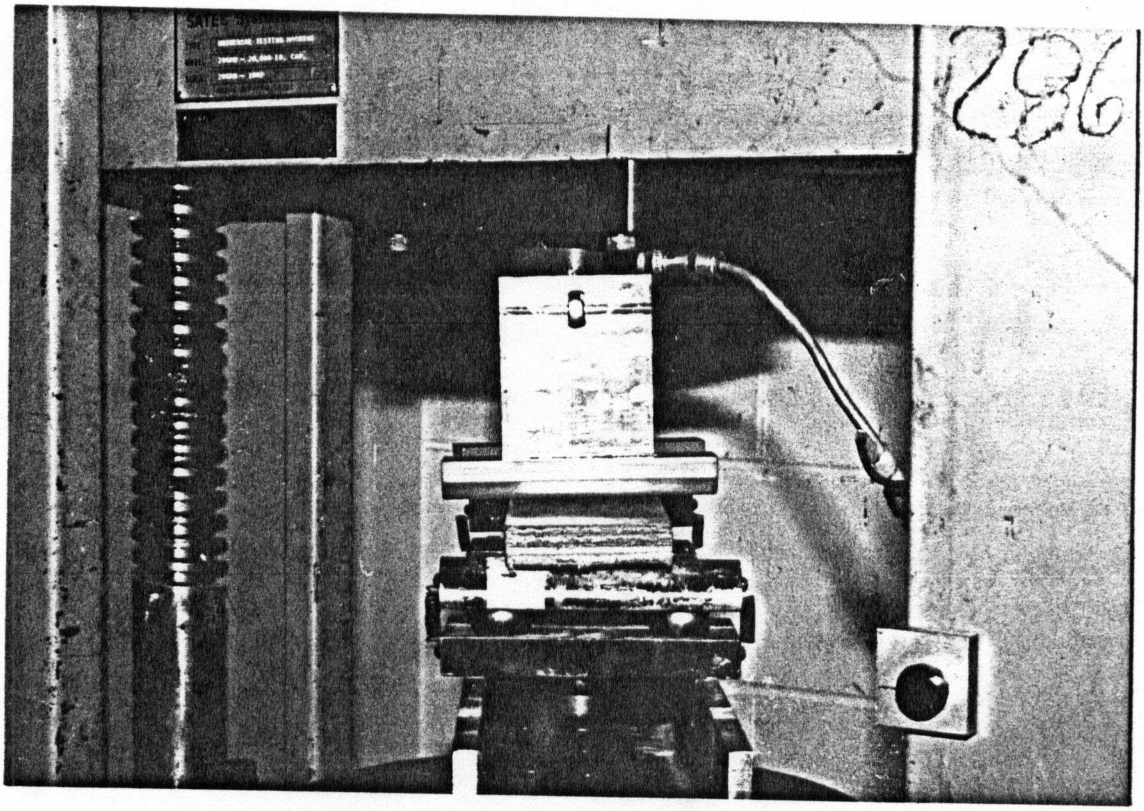

Figure 5.4: Cross-Section View of Beam Tested in Machine Direction

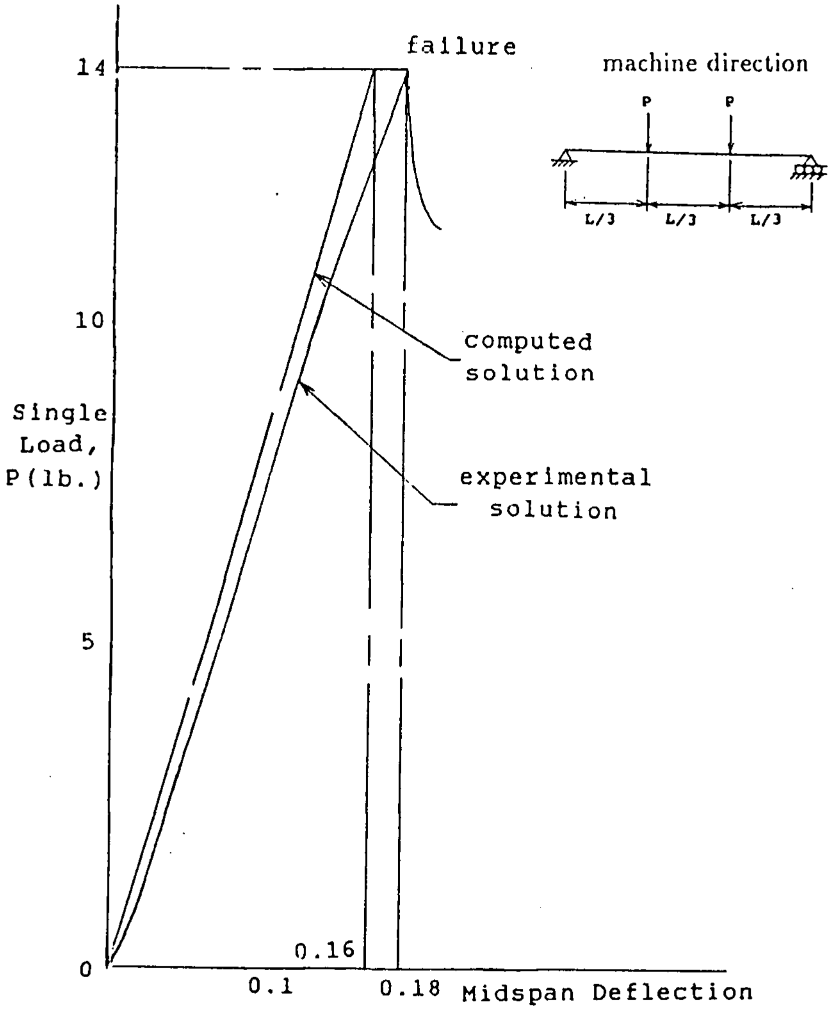

Figure 5.5: Typical Load-Deflection Curve of Beam Tested in Machine Direction

5.2 Comparison of Experimental and Numerical Results

The program's approximate solutions were compared to the experimental results. However, certain data required special consideration because of the cardboard's unique properties. The strength of the paper was highly dependent on the testing conditions of bending direction and relative humidity. Also, the program assumed a solid rectangular cross-section in each layer. This cross-section would provide a much higher bending stiffness than the actual cross-section in a corru-

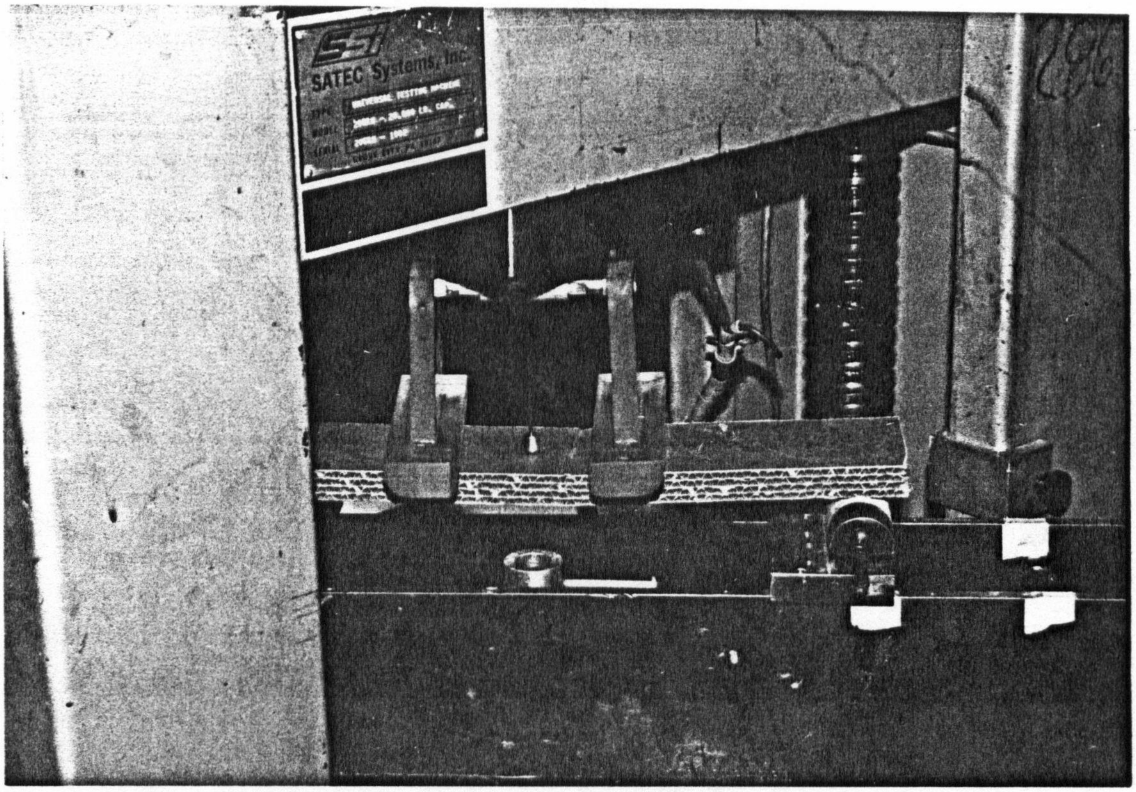

Figure 5.6: Compression Crease Failure

Test #	Load		Midspan Deflection	
	Average (N.)	Standard Deviation (N.)	Average (mm.)	Standard Deviation (mm.)
1	69.4	8.4	12.45	1.40
2	162.6	8.2	6.10	0.58
3	158.4	0.7	6.86	0.25
4	68.3	7.1	4.83	0.51
5	45.6	1.8	3.81	0.13
6	40.8	0.0	64.77	1.98

Table 5.4: Test Results of Loads and Midspan-Deflection at Failure

gated layer; thus, an equivalent elastic modulus was used to better approximate the beam's true bending stiffness. After the data were determined, results from the program were then compared to the experiemental results.

5.2.1 Data Consideration and Numerical Results

Figure 5.7 shows the properties of the corrugation which has an assumed sinusoidal shape. Special considerations were given to the following data:

1. Shear moduli of all the layers,

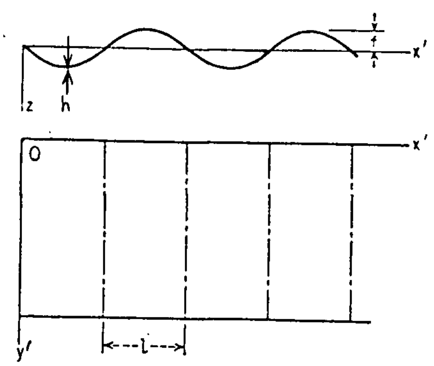

Figure 5.7: Sinusoidal Shape of Corrugation (Timoshenko,1959)

2. Elastic moduli of the liners,

3. Elastic moduli of the corrugated layers,

The elastic and shear moduli values were given by MacMillan Bloedel Research. The elastic moduli for each type of paper were determined from tension tests. The shear moduli for the corrugation were determined from direct shear tests of the complete corrugated sections.

Shear Moduli

The shear modulus G_{xz} was used in each layer for bending in the machine direction. G_{yz} was used for bending in the cross-machine direction. In addition, effect on the shear modulus due to relative humidity should be considered. Shear moduli at three different relative humidity values were given. The shear moduli corresponding to the measured relative humidity were then linearly interpolated from these three points.

An alternate method of determining the shear modulus G_{xz} of the corrugation was done using the program NISA. Three approximate shapes of the corrugation were modelled in NISA to obtain shear stress-strain relationships in the x-z direction. The three shapes are a straight-line, triangular shape, a sinusoidal shape, and a semi-circular shape. A single corrugation (see Figure 5.8) spanned over one wave

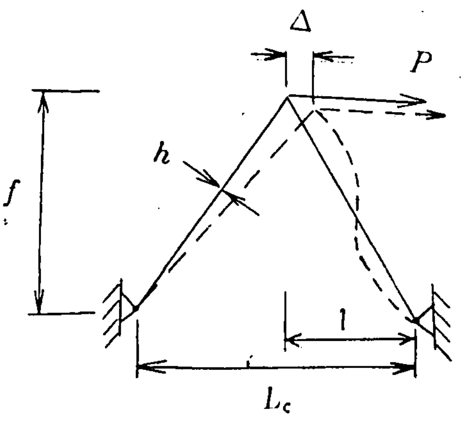

Figure 5.8: Geometric Properties of the Approximated Triangular Shape

length L_c was used and the ends are assumed to be pin-ended with no roller. A concentrated force P was applied at the apex of the corrugation to produce the shearing action.

The elastic modulus of the 26C paper was assumed to be constant. The shear strain was approximated as the displacement Δ over the amplitude f. The shear stress was approximated as the force per unit width P over the wave length L_c. The resulting shear stress-strain relationships were plotted in Figure 5.9. The curve for the triangular shape is shown to be linear. But the curves for the sinusoidal and semi-circular shapes are shown to be nonlinear. The constant value of G_{xz} given by MacMillan Bloedel Research falls within these two nonlinear curves.

It is interesting to note that, although the elastic modulus of the paper was kept constant, the corrugation's geometry produced a nonlinear shear stress-strain relationship.

Elastic Moduli of Liners

For bending in the machine direction, elastic modulus E_x was used in each layer; while E_y was used for bending in the cross-machine direction. Relative humidity effect on the liners' elastic moduli was also considered using the same interpolation

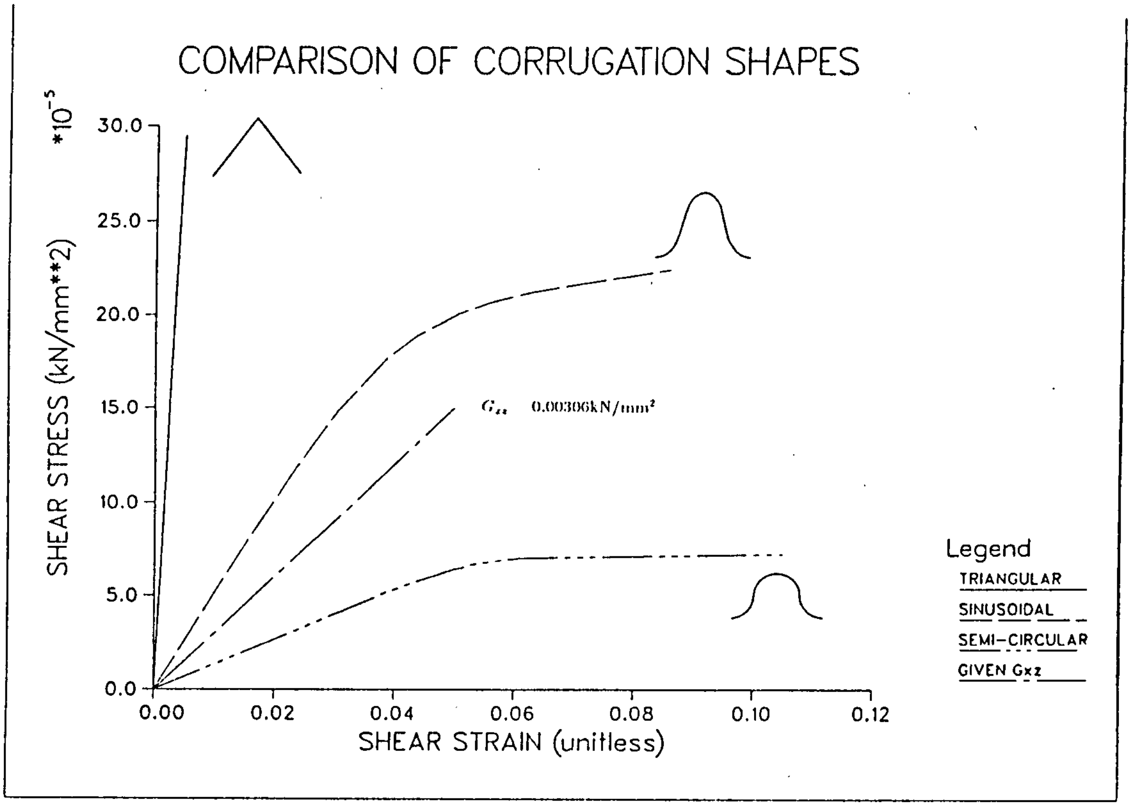

Figure 5.9: Comparison of the Approximate Shapes of Corrugation

method discussed in the shear moduli section.

Elastic Moduli of Corrugated Layers

In addition to the above considerations for bending direction and relative humidity, the elastic moduli of the corrugated layers should be modified because the program assumed a solid, rectangular cross-section in every layer. This assumption was not correct for corrugated layers. To correct for this assumption, equivalent elastic modulus for each corrugated layer should be computed and placed into the data file. To determine an equivalent elastic modulus, the layer's bending stiffness contribution in both $x-$ and $y-$directions were equated for the corrugated and the solid layer:

$$E_{x_{eq}} I_{x_{eq}} = E_x I_{x_{cor}} (= D_x) \quad (5.1)$$

$$E_{y_{eq}} I_{y_{eq}} = E_y I_{y_{cor}} (= D_y) \quad (5.2)$$

where

$E_{x_{eq}}, E_{yeq}$ = equivalent elastic modulus for bending in the $x-$ and $y-$directions,

$I_{x_{eq}}, I_{y_{eq}}$ = moment of inertia about the beam's global $x-$ and y-axes of the assumed solid rectangular cross-sections,

E_x, E_y = elastic moduli of the paper for bending in the x- and y-directions,

$I_{x_{cor}}, I_{y_{cor}}$ = moment of inertia about the beam's x- and y-axes of the corrugated layer.

D_x, D_y = bending stiffness in the x- and y-directions

If $l = s$(s is defined below), the corrugation would become a flat rectangular cross-section of depth h. Its contribution to the beam's bending stiffness EI would be

$$EI = E\frac{h^3}{12} + Ehd^2 \tag{5.3}$$

On the other hand, if $l = 0$ the corrugation would not contribute to the beam's EI. It may be assumed (Timoshenko, 1959) that for the intermediate values of l (Figure 5.7), the bending stiffnesses in the x and y directions are:

$$D_{x'} = \frac{l}{s}E_x\frac{h^3}{12} \tag{5.4}$$

$$D_{y'} = E_y I \tag{5.5}$$

where h is the thickness of the paper. The length of the arc for a half-wave is represented as s and is given as

$$s = l\left(1 + \frac{\pi^2 f^2}{4l^2}\right)$$

and

$$I = \frac{f^2 h}{2}\left[1 - \frac{0.81}{1 + 2.5\left(\frac{f}{2l}\right)^2}\right]$$

Because these bending stiffness were determined about the local axes x' and y' of the corrugation, the parallel axis theorem must be applied to determine the bending stiffness about the global axes x and y of the beam. Thus, if the distance in z direction between x' and x (also y' and y) was given as d, then the bending stiffness of the corrugation about the beam's axes (x and y) were obtained as

$$D_x = D_{x'} + \frac{l}{s}E_x hd^2 \tag{5.6}$$

$$D_y = D_{y'} + \frac{s}{l}E_y hd^2 \tag{5.7}$$

The second term in equation 5.6 was determined using the same linear approximation ($\frac{l}{s}$ factor) that Timoshenko had presented (see equation 5.4). The second term in equation 5.7 was determined from finding the area per unit length of the cross-section (equal to $\frac{sh}{l}$) and then applying the parallel axis theorem.

Applying the criterion in equation 5.2, the equivalent elastic moduli were then given as

$$E_{x_{eq}} = \frac{D_x}{\left[\frac{f^3}{12} + fd^3\right]} \tag{5.8}$$

$$E_{y_{eq}} = \frac{D_y}{\left[\frac{f^3}{12} + fd^3\right]} \tag{5.9}$$

Applying equations 5.8 and 5.9 to the tested beams' corrugated layers, the equivalent elastic moduli $E_{x_{eq}}$ and $E_{y_{eq}}$ were respectively found to be around 55.2 MPa and 262.0 MPa for all corrugated layers. The equivalent elastic moduli of all corrugations did not deviate greatly from the above two values (sample calculation in Appendix C).

Numerical Results

Numerical results of the tested beam at the experimental average failure loads were obtained from the program. Table 5.5 shows a comparison of the experimental average deflections and the computed deflections at the average failure loads.

Test #	Sample Size	Average Failure Load, P (N)	Exper. Average Failure Defl. (mm.)	Computed @ Failure Load Deflection (mm.)	Maximum Compressive Stress (MPa)
1	17	69.26	12.45	13.63	10.1
2	11	162.35	6.10	6.71	8.0
3	2	158.13	6.86	6.82	8.0
4	10	68.23	4.83	4.46	9.1
5	4	45.59	3.81	2.98	10.2
6	2	40.03	64.77	40.08	6.4

Table 5.5: Experimental and Numerical Results
*note: test # 6 was tested in the cross-machine direction

5.2.2 Discussion of Results

From Table 5.5 above, the computed and experimental deflections in tests #1-4 were observed to be in close agreement with each other.

Figure 5.10 shows a plot of the means and standard deviations of the experimental deflections vs. the computed failure deflections. The computed deflections of tests #1-4 are all located within the determined standard deviations. Also, Figure 5.5 shows that the load-deflection curves in tests #1-4 are almost linear.

Based on the above comparison, the program may be said to provide a reasonable approximation to the test problem. Maximum compressive stresses in the 90-lb liner were then determined from the program for bending in the machine direction and the specific relative humidity in tests #1-4. By averaging the four tests' values, a maximum compressive stress of 8.96 MPa was obtained as the compressive failure stress of the 90-lb liner in the machine direction. This failure criterion was then imposed to determine load-interaction design curves for machine direction bending of any corrugated beam with the compressive crease failure occuring in the 90-lb liner.

Result of test #5 was different because the compressive failure had occurred in the 42-lb liner. Thus, the maximum compressive failure stress was not included in

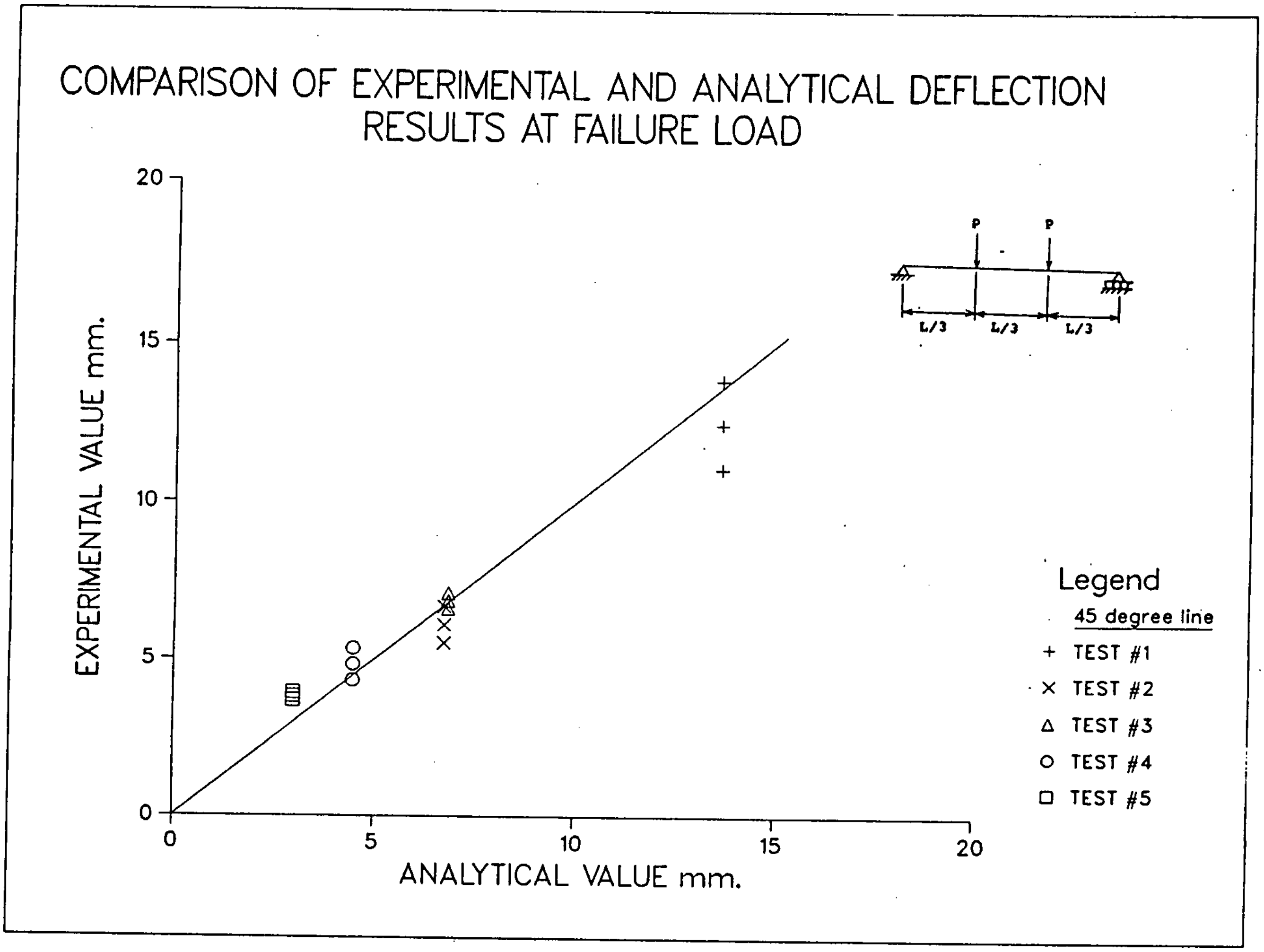

Figure 5.10: Comparison of Experimental and Numerical Results in Machine Direction

determining the 8.96 MPa compressive stress value. Also, the computed deflection was significantly lower than the experimental average deflection. This difference may be attributed to nonlinear elastic moduli of the 90-lb liner in tension.

In test #6, the beams were bent in the cross-machine direction. The experimental average deflection was much larger than the computed deflection. However, extrapolating a linear solution of 45.72 mm. from the experimental load-deflection curve (Figure 5.11), the computed deflection of 40.08 mm. would be a reasonable approximation to the linear solution.

The concave shape of the experimental curve in Figure 5.11 indicated that the beam was softening during the test. Nonlinear effect had contributed significantly to the beam's deflection. This additional nonlinear deflection may be caused by nonlinear shear moduli of the corrugated layers. The deflections of the beams tested in the machine direction were in the small deformation range of $\frac{\Delta}{d_b} < 1$. Previous discussion of the shear moduli modelled by NISA has shown that the shear moduli in machine direction is highly nonlinear but the experimental curves are shown to be almost linear. Therefore, the beam's deflection must be occurring in the small shear strain range, thus producing a linear load-deflection curve. However, for beams tested in the cross-machine direction, the deflections were in the large deformation range of $\frac{\Delta}{d_b} >> 1$. The above explanation of different deformation range can be observed in Figure 5.12. Typical shear stress-strain relationships for both test directions are assumed and the shear behaviour for the two directions are indicated. This difference may account for the linear and nonlinear experimental curves for the two different bending directions.

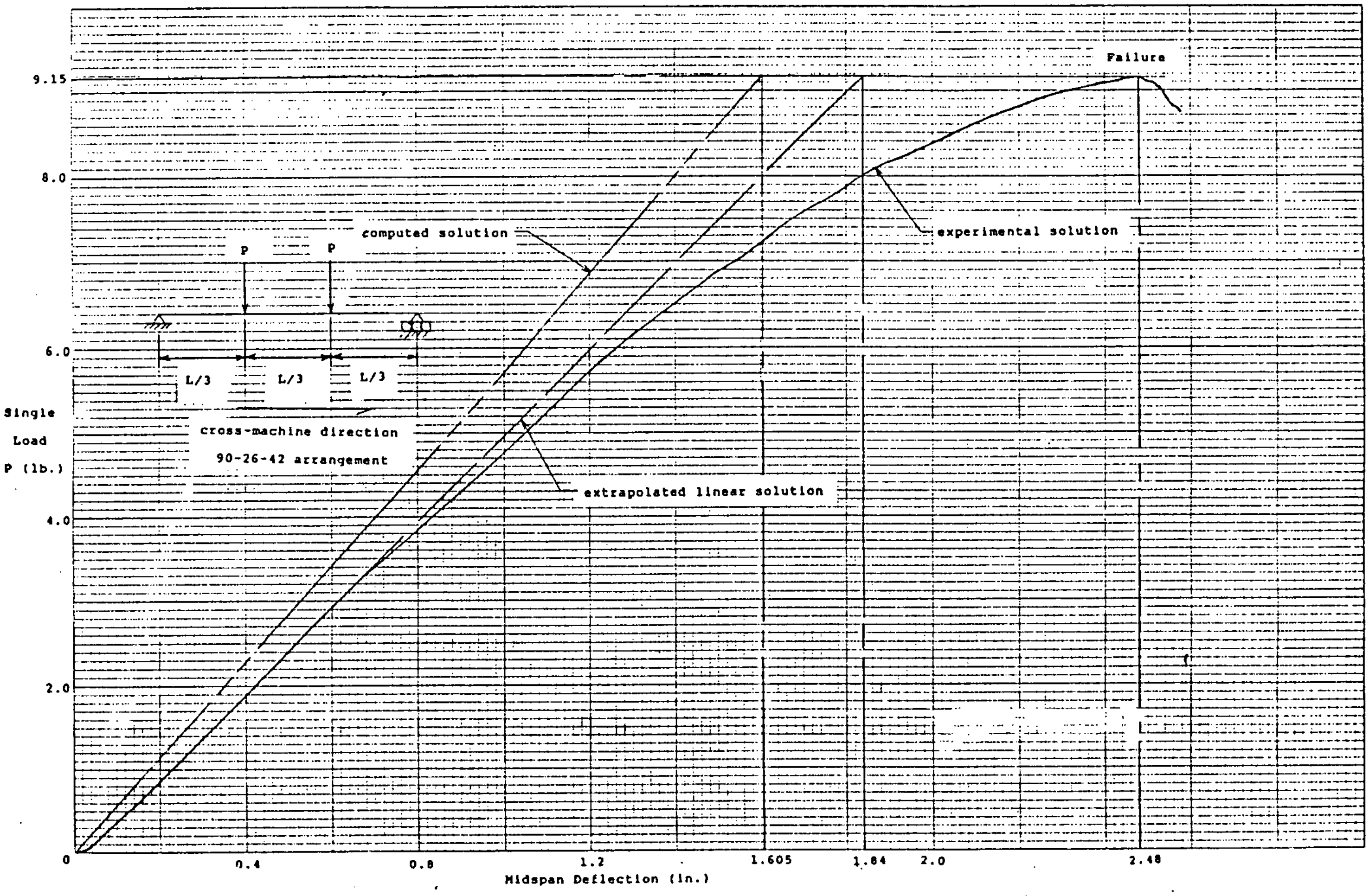

Figure 5.11: Comparison of Computed and Experimental Load-Deflection Curve in Cross-Machine Direction

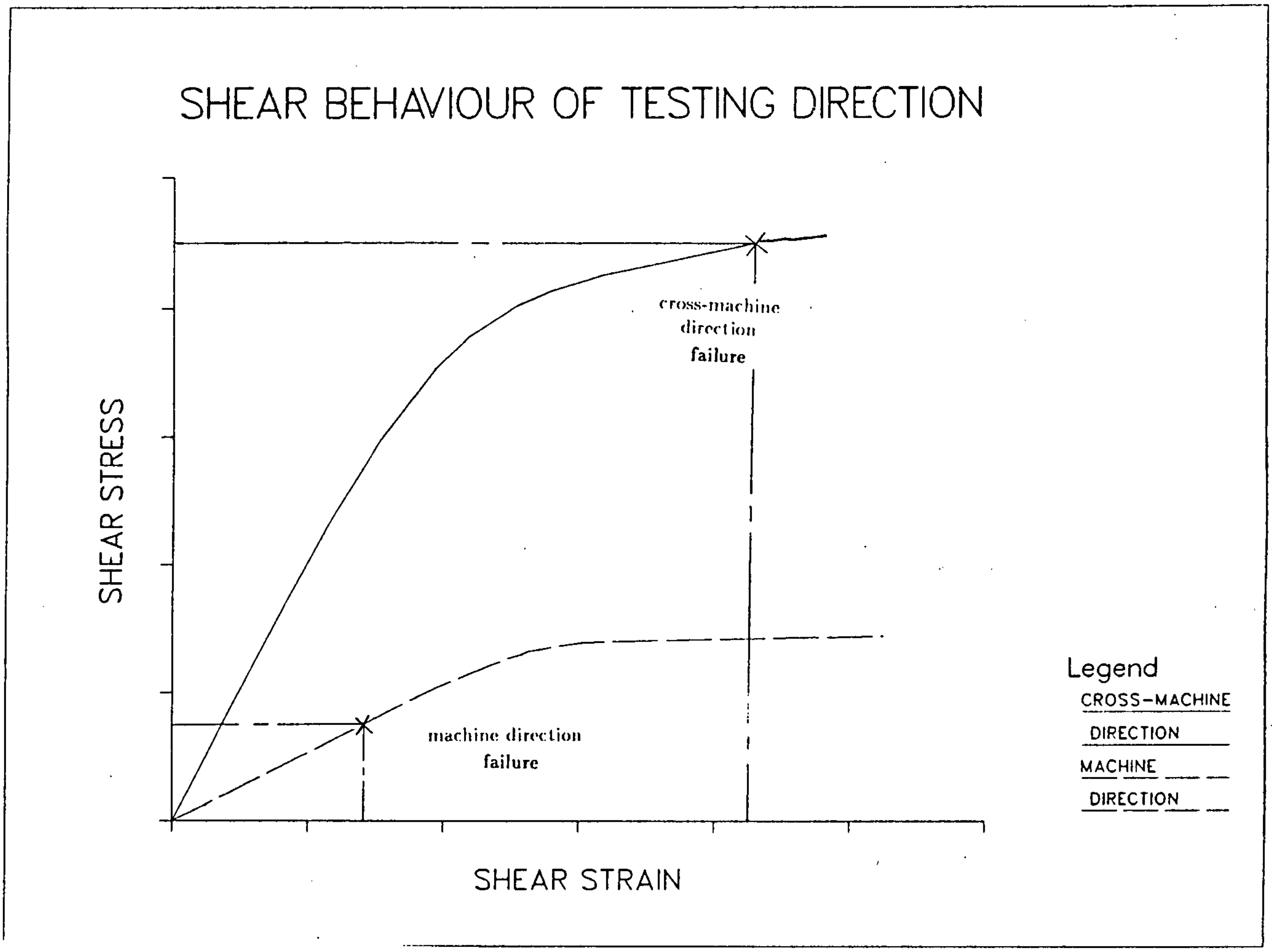

Figure 5.12: Typical Shear Behaviour of Testing Direction

Chapter 6

Design Curves

Design curves of load-interaction are produced by applying the maximum compressive failure stress criterion to the program's stress results. An example of load-interaction curve is presented for the bending of a simply supported beam subjected to an axial load and a linearly distributed lateral load

The failure stress criterion of 8.96 MPa (1300 psi.) is applied to the outer liner (90-lb liner) on the compressive side of the beam. Combination of axial and lateral load values for spans of 1.219 m. (48 in.) and 1.829 m. (72 in.) are determined from the program. The results are plotted in Figure 6.1.

Similar plots can be done to study the effects of different layer arrangements, elastic moduli, shear moduli, maximum failure stress value, etc., on the allowable load combination.

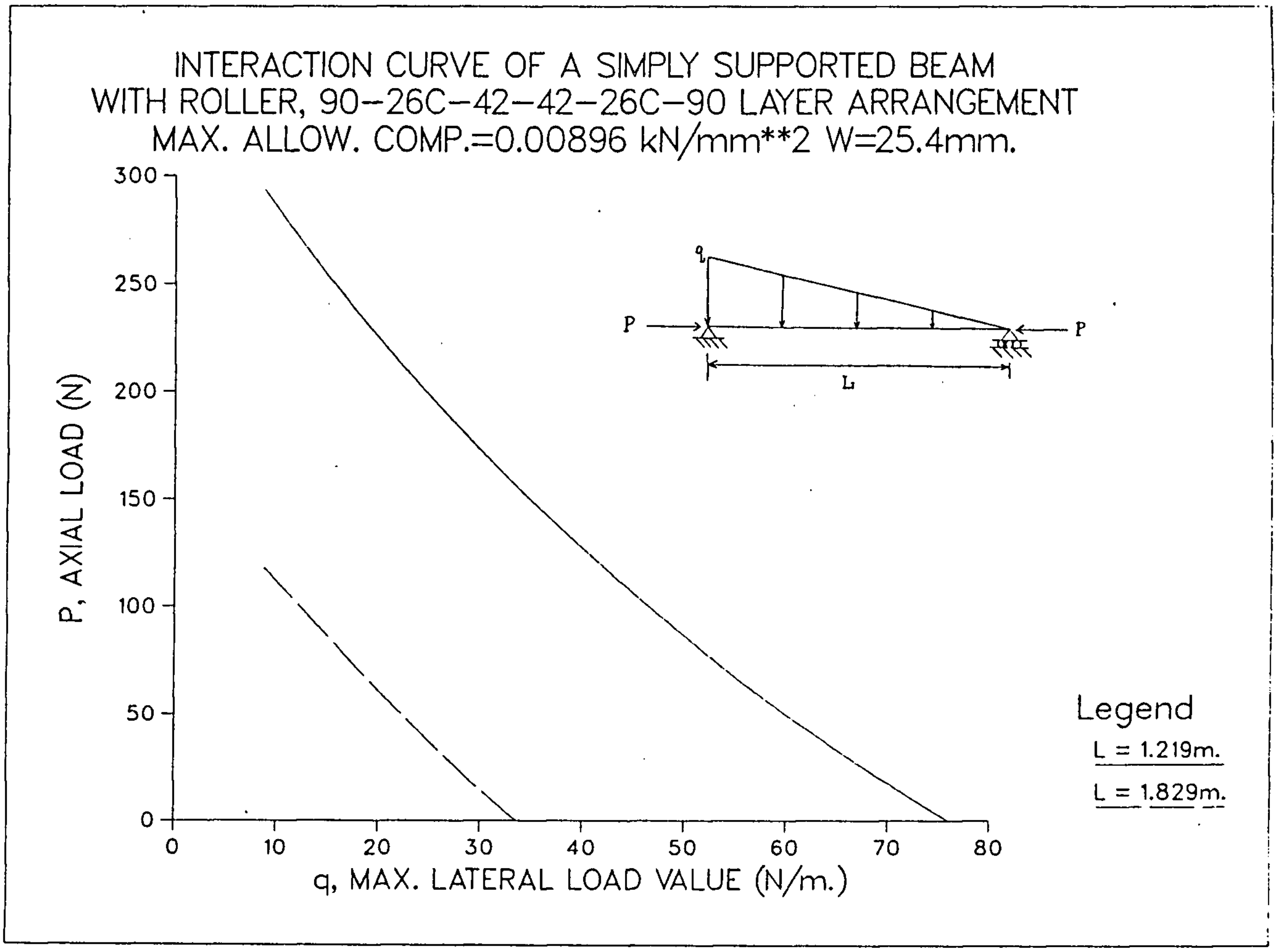

Figure 6.1: Loads Interaction Curves at Failure Stress of 8.96 MPa

Chapter 7

Conclusion

A finite element program has been developed using multilayer beam elements. Shear and large deformation effects are included in the analysis. The program is shown to accurately predict solutions which agree with several solutions found in the literature.

Also, an experiment was done to measure the deflection of multilayer, corrugated, cardboard beams which are weak in shear strength. The experimental results for bending in the machine direction were accurately predicted by the program's numerical results. A procedure to obtain combined axial and lateral loads interaction curves was presented using maximum compressive stress as a failure criterion.

Further research relating to this subject should include: bending in the cross-machine direction, shear modulus nonlinearity, extension to plate bending, a shear stress approximation across the beam depth and dynamic loading.

Bibliography

[1] DI SCIUVA, M. 1986. Bending, Vibration and Buckling of Simply Supported Thick Multilayered Orthotropic Plates: An Evaluation of a New Displacement Model. Journal of Sound and Vibrations 105(3), pp. 425-442.

[2] FOSCHI, R.O. 1973. Deflection of Multilayer-Sandwich Beams with Application to Plywood Panels. Wood and Fiber, Vol. 5, No. 3, pp. 182-191.

[3] KAO, J.S. and ROSS, R.J. 1968. Bending of Multilayer Sandwich Beams. AIAA Journal, Vol. 6, No. 8, pp. 1583-1585.

[4] KHATUA, T.P. and CHEUNG, Y.K. 1973. Bending and Vibration of Multilayer Sandwich Beams and Plates. International Journal for Numerical Methods in Engineering, Vol. 6, pp. 11-24.

[5] PUTCHA, N.S. and REDDY, J.N. 1986. A Refind Mixed Shear Flexible Finite Element for the Nonlinear Analysis of Laminated Plates. Computer & Structures, Vol. 22, No. 4, pp. 529-538.

[6] REN, J.G. and HINTON, E. 1986. The Finite Element Analysis of Homogeneous and Laminated Composite Plates Using a Simple Higher Order Theory. Communications in Applied Numerical Methods, Vol. 2, pp. 217-228.

[7] TIMOSHENKO, S. and WOINOWSKY-KRIEGER, S. 1959. Theory of Plates and Shells. McGraw-Hill Book Company, New York, U.S.A., pp. 4-15 and pp.367-368.

[8] ZIENKIEWICZ, O.C. 1979. The Finite Element Method, 3rd Edition. Tata McGraw-Hill Publishing Company Limited, New Delhi, India, pp. 500-512.

Appendix A

A Tangential Stiffness Matrix Term

The term $\frac{d[B_2]^T}{d\{a\}}\{\sigma\}$ is required in the derivation of the tangential stiffness matrix $[K_t]$. Zienkiewicz (1979) presented a procedure to formulate an equivalent expression for this term. Applying this procedure to the analysis, the derivative vector $\{\theta\}$ with respect to the natural coordinates ξ and ς is found to be

$$\begin{aligned} \{\theta\} &= \left\{ \begin{array}{c} \frac{dw}{d\xi}\left(\frac{2}{\Delta x}\right) \\ \frac{dw}{d\varsigma}\left(\frac{2}{\Delta y}\right) \end{array} \right\} \\ &= \left\{ \begin{array}{c} \frac{dw}{d\xi}\left(\frac{2}{\Delta x}\right) \\ 0 \end{array} \right\} \\ &= [G]\{a\} \end{aligned} \tag{A.1}$$

where

$$[G] = \left[\begin{array}{c} \frac{2}{\Delta x}\frac{d\{M\}^T}{d\xi} \\ \{0\}^T \end{array} \right]$$

and $\{a\}$ is the displacement vector.

The nonlinear term in the strain equation expressed in natural coordinates is

$$\begin{aligned} \{\varepsilon_{NL}\} &= \frac{4}{\Delta x^2}\left\{ \begin{array}{c} \frac{1}{2}\left(\frac{dw}{d\xi}\right)^2 \\ 0 \end{array} \right\} \\ &= \frac{1}{2}[A][G]\{a\} \end{aligned} \tag{A.2}$$

with $[A]$ being

$$[A] = \frac{2}{\Delta x}\left[\begin{array}{cc} \frac{dw}{d\xi} & 0 \\ 0 & \frac{dw}{d\varsigma} \end{array}\right] \tag{A.3}$$

Therefore, the $[B_1]$ term discussed in previous section is now equal to

$$[B_1] = \frac{1}{2}[A][G] \tag{A.4}$$

Applying a special property described by Zienkiewicz , the term $\frac{d[B_2]^T}{d\{a\}}\{\sigma\}$ in our beam analysis can now be stated as

$$\begin{aligned} \frac{d[B_2^T]}{d\{a\}}\{\sigma\} &= \int_V [G]^T \left[\begin{array}{cc} \sigma_x & \gamma_{xy} \\ \gamma_{xy} & \sigma_y \end{array}\right][G]dV \\ &= \frac{4}{\Delta x^2}\int_V \left[\begin{array}{cc} \frac{d\{M\}}{d\varsigma} & \{0\} \end{array}\right]\left[\begin{array}{cc} \sigma_x & \gamma_{xy} \\ \gamma_{xy} & \sigma_y \end{array}\right] \\ &\quad \left[\begin{array}{c} \frac{d\{M\}^T}{d\varsigma} \\ \{0\}^T \end{array}\right] dV \end{aligned} \tag{A.5}$$

Since γ_{xy} and σ_y are assumed to be 0's in the analysis, equation A.5 can be simplified to

$$\frac{d[B_2]^T}{d\{a\}}\{\sigma\} = \frac{4}{\Delta x^2}\int_V \sigma_x \frac{d\{M\}}{d\xi}\frac{d\{M\}^T}{d\xi}dV \tag{A.6}$$

This equation can now be substituted into the tangential stiffness matrix equation.

Appendix B

Material Nonlinearity - Shear Moduli

Material nonlinearity occurs because of a nonlinear stress-strain relationship. The discussion in this section will be focussed on the assumption of a nonlinear shear stress-strain relationship for a particular layer. Previous assumptions such as stress continuity and linear shear strain interpolation within each layer remains.

B.1 General Formulation

Consider a general shear stress-strain function

$$\tau = g(\gamma) \tag{B.1}$$

The elasticity matrix $[D]$ is now dependent upon the strain vector $\{\varepsilon\}$. For example, a commonly used nonlinear stress-strain relationship is given as

$$\tau = (P_0 + P_1\gamma)\left(1 - \exp\left(\frac{-k}{P_0}\gamma\right)\right) \tag{B.2}$$

B.2 Piecewise Linear Shear Strain Assumption

The shear stress is now a nonlinear function of shear strain for each layer. Recalling the shear strain approximation discussed in chapter 2, the shear strain at the n-th layer is

$$\gamma_n(\varsigma) = \gamma_{n_0}\left(\frac{1-\varsigma}{2}\right) + \gamma_n\left(\frac{1+\varsigma}{2}\right) \tag{B.3}$$

where γ_{n_0} is dependent upon the previous layer's shear strain γ_{n-1}. Shear stress continuity was applied at the interface of any two consecutive layeres. For material nonlinearity and a general stress-strain function, the value γ_{n_0} can now be expressed as

$$\gamma_{n_0} = f_n(\gamma_{n-1}) \tag{B.4}$$

where $f_n(\gamma_{n-1})$ is a nonlinear function of γ_{n-1}. In the previous discussion of linear material properties, $f_n(\gamma_{n-1})$ was simply

$$f_n(\gamma_{n-1}) = \frac{G_{n-1}}{G_n}\gamma_{n-1} \tag{B.5}$$

However, with a general nonlinear stress-strain relationship, equation B.3 can now be expressed as

$$\gamma_n(\varsigma) = f_n(\gamma_{n-1})\left(\frac{1-\varsigma}{2}\right) + \gamma_n\left(\frac{1+\varsigma}{2}\right) \tag{B.6}$$

The shear distortion $u^*(\xi,\varsigma)$ can now be obtained using the same procedure as described in chapter 2. The shear strain in each layer is assumed to be linear and shear stress continuity between layers is imposed. Integrating with respect to the coordinate ς, u^* at the n-th layer is expressed as

$$\begin{aligned} u_n^*(\xi,\varsigma) &= \left\{\sum_{k=1}^{n-1}\frac{t_k}{2}\left[\gamma_k + f_k(\gamma_{k-1})\right]\right\} + \frac{t_n}{2}\left[\gamma_n\left(\frac{\varsigma^2}{4}+\frac{\varsigma}{2}+\frac{1}{4}\right)\right. \\ &\left. f_n(\gamma_{n-1})\left(\frac{\varsigma}{2}-\frac{\varsigma^2}{4}+\frac{3}{4}\right)\right] - \left\{\sum_{k=1}^{NA}\frac{t_k}{2}\left[\gamma_k + f_k(\gamma_{k-1})\right]\right\} \end{aligned}$$

$$-\frac{t_{NA}}{2}\left[\gamma_{NA}\left(\frac{LOCAL^2}{4}+\frac{LOCAL}{4}+\frac{1}{4}\right)\right.$$

$$\left.+f_{NA}(\gamma_{NA-1})\left(\frac{LOCAL}{2}-\frac{LOCAL^2}{4}+\frac{3}{4}\right)\right] \qquad \text{(B.7)}$$

Expressing this equation in a more compact form, the shear distortion $u_n^*(\xi,\varsigma)$ can finally be expressed as

$$\begin{aligned} u_n^*(\xi,\varsigma) &= \sum_{k=1}^{n}\gamma_k(\xi)F(n,k,\varsigma)-\sum_{k=1}^{NA}\gamma_k(\xi)F^*(NA,k,LOCAL) \\ &+\sum_{k=1}^{n}T(n,k,\xi,\varsigma)-\sum_{k=1}^{NA}T^*(NA,k,\xi,LOCAL) \qquad \text{(B.8)} \end{aligned}$$

where

$$F(n,k,\varsigma) = \frac{t_k}{2}$$

for $k=1,2,\ldots,n-1$

$$F(n,n,\varsigma) = \frac{t_n}{2}\left(\frac{\varsigma^2}{4}+\frac{\varsigma}{4}+\frac{1}{4}\right)$$

for $k=n$

$$F^*(NA,k,LOCAL) = \frac{t_k}{2}$$

for $k=1,2,\ldots,n-1$

$$F(NA,NA,LOCAL) = \frac{t_{NA}}{2}\left(\frac{LOCAL^2}{4}+\frac{LOCAL}{4}+\frac{1}{4}\right)$$

for $k=NA$

and

$$T(n,k,\varsigma) = f_k(\gamma_{k-1})\frac{t_k}{2}$$

for $k = 1, 2, \ldots, n-1$

$$T(n,n,\varsigma) = f_n(\gamma_{n-1})\frac{t_n}{2}\left(\frac{\varsigma^2}{2} - \frac{\varsigma^2}{4} + \frac{3}{4}\right)$$

for $k = n$

$$T^*(NA,k,LOCAL) = f_k(\gamma_{k-1})\frac{t_k}{2}$$

for $k = 1, 2, \ldots, n-1$

$$T^*(NA,NA,LOCAL) = f_{NA}(\gamma_{NA-1})\frac{t_NA}{2}\left(\frac{LOCAL}{2} - \frac{LOCAL^2}{4} + \frac{3}{4}\right)$$

for $k = NA$

B.3 Finite Element Formulation

The shear distortion u^* and shear strain γ consists of two components: 1. linear component of F and F^* and 2. nonlinear component of T and T^*. Proceeding as before, the strain equations of

$$\varepsilon_x = \frac{d\bar{u}}{dx} - z\frac{d^2w}{dx^2} + \frac{du^*}{dx} + \frac{1}{2}\left(\frac{dw}{dx}\right)^2$$

$$\gamma_{xz} = \frac{du^*}{dz}$$

are applied to the displacements. Knowing the shape functions of

$$\bar{u} = \{N1\}^T\{a\}$$

$$w = \{M\}^T\{a\}$$

$$\gamma_k = \{N3_k\}^T\{a\} \tag{B.9}$$

and applying the chain rule, the strain ε_{x_n} is then expressed as

$$\begin{aligned}
\varepsilon_{x_n} = & \left\{ \frac{2}{\Delta x}\frac{d\{N1\}^T}{d\xi} - \left(z_n + \frac{t_n}{2}\varsigma\right)\frac{4}{\Delta x^2}\frac{d^2\{M\}^T}{d\xi^2} \right. \\
& \left. + \sum_{k=1}^{n} \frac{2}{\Delta x} F(n,k,\varsigma)\frac{d\{N3_k\}^T}{d\xi} - \sum_{k=1}^{NA}\frac{2}{\Delta x}F^*(NA,k,LOCAL)\frac{d\{N3_k\}^T}{d\xi} \right\}\{a\} \\
& + \sum_{k=1}^{n}\frac{2}{\Delta x}\frac{\partial T}{\partial\gamma_{k-1}}\frac{d\gamma_{k-1}}{d\xi} - \sum_{k=1}^{NA}\frac{2}{\Delta x}\frac{\partial T^*}{\partial\gamma_{k-1}}\frac{d\gamma_{k-1}}{d\xi} \\
& + \frac{1}{2}\frac{4}{\Delta x^2}\{a\}^T\frac{d\{M\}}{d\xi}\frac{d\{M\}^T}{d\xi}\{a\}
\end{aligned} \tag{B.10}$$

The derivative term of γ's respect to ξ and the partial derivative term of T and T^* are then obtained as

$$\frac{d\gamma_{k-1}}{d\xi} = \frac{d\left\{N3_{k-1}\right\}^T}{d\xi}\{a\} \tag{B.11}$$

$$\begin{aligned}
\left(\frac{\partial T}{\partial\gamma_{k-1}}\right) &= \frac{\partial f_k(\gamma_{k-1})}{\partial\gamma_{n-1}}\frac{t_k}{2} \\
& \quad \text{for } k = 1,2,\ldots,n-1 \\
&= \frac{\partial f_n(\gamma_{n-1})}{\partial\gamma_{n-1}}\frac{t_n}{2}\left(\frac{\varsigma}{2} - \frac{\varsigma^2}{4} + \frac{3}{4}\right) \\
& \quad \text{for } k = n
\end{aligned} \tag{B.12}$$

and

$$\left(\frac{\partial T^*}{\partial \gamma_{k-1}}\right) = \frac{\partial f_k(\gamma_{k-1})}{\partial \gamma_{k-1}}\frac{t_k}{2}$$

$$\text{for } \; k = 1, 2, \ldots, NA-1$$

$$= \frac{\partial f_{NA}(\gamma_{NA-1})}{\partial \gamma_{NA-1}}\frac{t_{NA}}{2}\left(\frac{LOCAL}{2} - \frac{LOCAL^2}{4} + \frac{3}{4}\right)$$

$$\text{for } \; k = NA \qquad \text{(B.13)}$$

Equations B.11 to B.13 are now substituted into equation B.10 to give the following bending strain equation

$$\varepsilon_{x_n} = \{KX\}^T\{a\} + \{KXS\}^T + \{a\} + \{KXG\}^T\{a\} \qquad \text{(B.14)}$$

where

$$\{KX\}^T = \left\{\frac{2}{\Delta x}\frac{d\{N1\}^T}{d\xi} - \left(z_n + \frac{t_n}{2}\varsigma\right)\frac{4}{\Delta x^2}\frac{d^2\{M\}^T}{d\xi^2}\right.$$

$$+\sum_{k=1}^{n}\frac{2}{\Delta x}F(n,k,\varsigma)\frac{d\{N3_k\}^T}{d\xi}$$

$$\left.-\sum_{k=1}^{NA}\frac{2}{\Delta x}F^*(NA,k,LOCAL)\frac{d\{N3_k\}^T}{d\xi}\right\}$$

$$\{KXS\}^T = \sum_{k=1}^{n}\frac{2}{\Delta x}\left(\frac{\partial T}{\partial \gamma_{k-1}}\right)\frac{d\{N3_{k-1}\}^T}{d\xi} - \sum_{k=1}^{NA}\frac{2}{\Delta x}\left(\frac{\partial T^*}{\partial \gamma_{k-1}}\right)\frac{d\{N3_{k-1}\}^T}{d\xi}$$

$$\{KXG\}^T = \frac{1}{2}\frac{4}{\Delta x^2}\{a\}^T\frac{d\{M\}}{d\xi}\frac{d\{M\}^T}{d\xi}$$

The shear strain has already been determined in the previous section. Further manipulation of equation B.14 gives the shear strain for the n-th layer as

$$\gamma_{xz_n}(\varsigma) = \{KXZ\}^T\{a\} + \{KXZS\}^T\{a\} \tag{B.15}$$

where

$$\{KXZ\}^T = \left(\frac{1+\varsigma}{2}\right)\{N3_n\}^T$$

and

$$\{KXZS\}^T = \left(\frac{1-\varsigma}{2}\right)\{N3_{n-1}\}^T\frac{f_n(\gamma_{n-1})}{\gamma_{n-1}}$$

Equations B.14 and B.15 are combined to produce the strain vector $\{\varepsilon\}$ which is given by

$$\{\varepsilon(n)\} = \left\{ \begin{array}{c} \varepsilon_{x_n} \\ \gamma_{xz_n} \end{array} \right\} = [B_0(n)]\{a\} + [BS_1(n)]\{a\} + [B_1(n)]\{a\} \tag{B.16}$$

where

$$\begin{aligned} [B_0(n)] &= \left[\begin{array}{c} \{KX(n)\}^T \\ \{KXZ(n)\}^T \end{array} \right] \\ [BS_1(n)] &= \left[\begin{array}{c} \{KXS(n)\}^T \\ \{KXZS(n)\}^T \end{array} \right] \\ [B_1(n)] &= \left[\begin{array}{c} \{KXG(n)\}^T \\ \{0\}^T \end{array} \right] \end{aligned}$$

The matrix $[B_0]$ is independent of the displacement vector $\{a\}$ while the other two matrices $[B_1]$ and $[BS_1]$ are dependent on $\{a\}$. The virtual strain equations are now determined in order to produce the virtual work equation.

The virtual strain terms from $[B_0]\{a\}$ and $[B_1]\{a\}$ have already been determined as

$$\begin{aligned} \delta\{[B_0(n)]\{a\}\} &= [B_0(n)]\{\delta a\} \\ \delta\{[B_1(n)]\{a\}\} &= 2[B_1(n)]\{\delta a\} = [B_2(n)]\{\delta a\} \end{aligned} \tag{B.17}$$

Therefore, virtual strain term from $[BS_1]\{a\}$ is now needed to complete the virtual strain equations. Using the chain rule, the term $\delta\Big\{[BS_1]\{a\}\Big\}$ is expressed as

$$\delta\Big\{[BS_1(n)]\{a\}\Big\} = \delta[BS_1(n)]\{a\} + [BS_1(n)]\{\delta a\} \tag{B.18}$$

$$= \left[\begin{array}{c} \delta\{KXS(n)\}^T \\ \delta\{KXZS(n)\}^T \end{array}\right]\{a\} + \left[\begin{array}{c} \{KXS(n)\}^T \\ \{KXZS(n)\}^T \end{array}\right]\{\delta a\}$$

The term $\delta\{KXS\}^T$ is determined as

$$\begin{aligned} \delta\{KXS(n)\}^T &= \sum_{k=1}^{n} \frac{2}{\Delta x}\delta\left(\frac{\partial T}{\partial \gamma_{k-1}}\right)\frac{d\left\{N3_{k-1}\right\}^T}{d\xi} \\ &\quad -\sum_{k=1}^{NA} \frac{2}{\Delta x}\delta\left(\frac{\partial T^*}{\partial \gamma_{k-1}}\right)\frac{d\left\{N3_{k-1}\right\}^T}{d\xi} \end{aligned} \tag{B.19}$$

with

$$\delta\left(\frac{\partial T}{\partial \gamma_{k-1}}\right) = \frac{\partial}{\partial a_k}\left(\frac{\partial T}{\partial \gamma_{k-1}}\right) = \left(\frac{\partial^2 T}{\partial {\gamma_{k-1}}^2}\right)\left\{N3_{k-1}\right\}^T\left\{\delta a\right\}$$

$$\delta\left(\frac{\partial T^*}{\partial \gamma_{k-1}}\right) = \frac{\partial}{\partial a_k}\left(\frac{\partial T^*}{\partial \gamma_{k-1}}\right) = \left(\frac{\partial^2 T^*}{\partial {\gamma_{k-1}}^2}\right)\left\{N3_{k-1}\right\}^T\left\{\delta a\right\}$$

which gives

$$\begin{aligned} \delta\{KXS(n)\}^T\{a\} &= \sum_{k=1}^{n} \frac{2}{\Delta x}\left(\frac{\partial^2 T}{\partial {\gamma_{k-1}}^2}\right)\left\{N3_{k-1}\right\}^T\left\{\delta a\right\}\frac{d\left\{N3_{k-1}\right\}^T}{d\xi}\{a\} \\ &\quad -\sum_{k=1}^{NA} \frac{2}{\Delta x}\left(\frac{\partial^2 T^*}{\partial {\gamma_{k-1}}^2}\right)\left\{N3_{k-1}\right\}^T\left\{\delta a\right\}\frac{d\left\{N3_{k-1}\right\}^T}{d\xi}\{a\} \end{aligned} \tag{B.20}$$

In order to remove the $\{\delta a\}$ vector in a later stage, the order of the vectors are rearranged while maintaining the same scalar product. The resulting equation is

$$\delta\{KXS(n)\}^T\{a\} = \left\{\sum_{k=1}^{n} \frac{2}{\Delta x}\left(\frac{\partial^2 T}{\partial {\gamma_{k-1}}^2}\right)\frac{d\left\{N3_{k-1}\right\}^T}{d\xi}\{a\}\left\{N3_{k-1}\right\}^T\right.$$

$$-\sum_{k=1}^{NA}\frac{2}{\Delta x}\left(\frac{\partial^2 T^*}{\partial {\gamma_{k-1}}^2}\right)\frac{d\left\{N3_{k-1}\right\}^T}{d\xi}\{a\}\left\{N3_{k-1}\right\}^T\Bigg\}\{\delta a\}$$

(B.21)

Defining

$$\delta\{KXS\}^T\{a\} = \{KXSD\}^T\{\delta a\} \tag{B.22}$$

and

$$\{KXS_2\}^T = \{KXS\}^T + \{KXSD\}^T \tag{B.23}$$

The virtual bending strain can finally be expressed in the finite element form of

$$\delta\varepsilon_x = \{KX\}^T\{\delta a\} + \{KXS_2\}^T\{\delta a\} + 2\{KXG\}^T\{\delta a\} \tag{B.24}$$

The term $\delta\{KXZS\}^T$ is now determined by the same procedure as that used for $\delta\{KXS\}^T$

$$\delta\{KXZS\}^T = \left(\frac{1-\varsigma}{2}\right)\left\{N3_{k-1}\right\}^T\delta\left(\frac{f_n(\gamma_{n-1})}{\gamma_{n-1}}\right) \tag{B.25}$$

where

$$\begin{aligned}\delta\left(\frac{f_n(\gamma_{n-1})}{\gamma_{n-1}}\right) &= \frac{\partial}{\partial a_k}\left(\frac{f_n(\gamma_{n-1})}{\gamma_{n-1}}\right)\\ &= \frac{1}{\gamma_{n-1}}\frac{\partial f_n(\gamma_{n-1})}{\partial\gamma_{n-1}}\left\{N3_{k-1}\right\}^T\{\delta a\}\\ &\quad +f_n(\gamma_{n-1})\left(\frac{-1}{\gamma_{n-1}^2}\right)\left\{N3_{k-1}\right\}^T\{\delta a\}\end{aligned}$$

Therefore

$$\begin{aligned}\delta\{KXZS\}^T\{a\} &= \left\{\left(\frac{1-\varsigma}{2}\right)\left\{N3_{k-1}\right\}^T\left(\frac{1}{\gamma_{n-1}}\right)\left[\frac{\partial f_n(\gamma_{n-1})}{\partial\gamma_{n-1}} - \frac{f_n(\gamma_{n-1})}{\gamma_{n-1}}\right]\right.\\ &\quad \left.\{a\}\left\{N3_{k-1}\right\}^T\right\}\{\delta a\}\\ &= \{KXZSD\}^T\{\delta a\}\end{aligned} \tag{B.26}$$

and defining

$$\{KXZS_2\}^T = \{KXZS\}^T + \{KXZSD\}^T \tag{B.27}$$

The virtual shear strain is expressed as

$$\delta\gamma_{xz_n} = \{KXZ\}^T\{\delta a\} + \{KXZS_2\}^T\{\delta a\} \tag{B.28}$$

Now, the virtual strain vector $\{\delta\varepsilon\}$ can be defined as

$$\{\delta\varepsilon\} = \left\{ \begin{array}{c} \delta\varepsilon_x \\ \delta\gamma_{xz} \end{array} \right\} = \Big[[B_0] + [BS_2] + [B_2]\Big]\{\delta a\} \tag{B.29}$$

where

$$\begin{aligned} [B_0] &= \left[\begin{array}{c} \{KX\}^T \\ \{KXZ\}^T \end{array} \right] \\ [BS_2] &= \left[\begin{array}{c} \{KXS_2\}^T \\ \{KXZS_2\}^T \end{array} \right] \\ [B_2] &= \left[\begin{array}{c} 2\{KXG\}^T \\ \{0\}^T \end{array} \right] \end{aligned}$$

Therefore, the column vectors of strain, virtual strain and stress are defined as follow

$$\begin{aligned} \{\varepsilon\} &= [B_0 + BS_1 + B_1]\{a\} \\ \{\delta\varepsilon\} &= [B_0 + BS_2 + B_2]\{\delta a\} \\ \{\sigma\} &= [D]\{\varepsilon\} \end{aligned} \tag{B.30}$$

and the virtual work equation gives

$$\begin{aligned} \Phi(a) &= \left[\int_V \left[B_0^T + BS_2^T + B_2^T\right][D]\,[B_0 + BS_1 + B_1]\,dV\right]\{a\} - \{F\} \\ &= [K]\{a\} - \{F\} \end{aligned} \tag{B.31}$$

B.4 Newton-Raphson Method

If Standard Newton-Raphson method is again used to solve the system of non-linear equations, the tangential stiffness matrix $[K_t]$ must also be determined. From discussion of the Newton-Raphson method in chapter 3, the matrix $[K_t]$ is defined as

$$[K_t] = \frac{d\{\Phi(a)\}}{d\{a\}} = \tag{B.32}$$
$$\left[\int_V \frac{d}{da}\left[B_0^T + BS_2^T + B_2^T\right]\{\sigma\} + \left[B_0^T + BS_2^T + B_2^T\right]\frac{d\{\sigma\}}{d\{a\}}dV\right]$$

Applying chain rule again, the term $\frac{d\{\sigma\}}{d\{a\}}$ becomes

$$\frac{d\{\sigma\}}{d\{a\}} = \frac{d\{\sigma\}}{d\{\varepsilon\}}\frac{d\{\varepsilon\}}{d\{a\}} \tag{B.33}$$

Substituting this equation into equation B.30 gives

$$\begin{aligned}[K_t] = \int_V &\left[\frac{d\left[B_0^T + BS_2^T + B_2^T\right]}{d\{a\}}\{\sigma\}\right. \\ &\left. + \left[B_0^T + BS_2^T + B_2^T\right]\frac{d\{\sigma\}}{d\{\varepsilon\}}\frac{d\{\varepsilon\}}{d\{a\}}\right] dV\end{aligned} \tag{B.34}$$

From previous sections, the $\frac{d\{\varepsilon\}}{d\{a\}}$ term has already been defined as

$$\frac{d\{\varepsilon\}}{d\{a\}} = [B_0 + BS_2 + B_2] \tag{B.35}$$

The term $\frac{d\{\sigma\}}{d\{\varepsilon\}}$ is the slope of the stress-strain curve at any given strain. This term is then defined as the tangential elasticity matrix $[D_t]$ where

$$[D_t] = \begin{bmatrix} \frac{d\sigma_x}{d\varepsilon_x} & 0 \\ 0 & \frac{d\tau_{xz}}{d\gamma_{xz}} \end{bmatrix} \tag{B.36}$$

The matrix $[B_0]$ is independent of the vector $\{a\}$ thus $\frac{d[B_0]^T}{d\{a\}}$ is equal to zero. The term $\frac{d[B_2]^T}{d\{a\}}\{\sigma\}$ was already found in chapter 3 as

$$\frac{d[B_2]^T}{d\{a\}}\{\sigma\} = \sigma_x[MX] \tag{B.37}$$

Finally, $\frac{d[BS_2]^T}{d\{a\}}\{\sigma\}$ has to be determined to complete the $[K_t]$ matrix. From equation B.29, the matrix $[BS_2]^T$ is given as

$$[BS_2]^T = \begin{bmatrix} \{KXS_2\}^T \\ \{KXZS_2\}^T \end{bmatrix}^T \tag{B.38}$$

therefore, the terms $\frac{d\{KXS_2\}}{d\{a\}}$ and $\frac{d\{KXZS_2\}}{d\{a\}}$ have to be found.

Equation B.23 separated $\frac{d\{KXS_2\}}{d\{a\}}$ into two components: $\frac{d\{KXS\}}{d\{a\}}$ and $\frac{d\{KXSD\}}{d\{a\}}$. This leads to

$$\frac{d}{d\{a\}}\{KXS_2\} = \frac{d}{d\{a\}}\{KXS\} + \frac{d}{d\{a\}}\{KXSD\} \tag{B.39}$$

From equation B.20, it was found that

$$\begin{aligned} \frac{d}{d\{a\}}\{KXS\} &= \sum_{k=1}^{n} \frac{2}{\Delta x}\left(\frac{\partial^2 T}{\partial \gamma_{k-1}{}^2}\right)\frac{d\{N3_{k-1}\}}{d\xi}\{N3_{k-1}\}^T \\ & \sum_{k=1}^{NA} \frac{2}{\Delta x}\left(\frac{\partial^2 T^*}{\partial \gamma_{k-1}{}^2}\right)\frac{d\{N3_{k-1}\}}{d\xi}\{N3_{k-1}\}^T \end{aligned} \tag{B.40}$$

However, the term $\frac{d\{KXSD\}^T}{d\{a\}}$ still remains to be found. Differentiating equation B.21 with respect to the vector $\{a\}$, the following equation

$$\begin{aligned} \frac{d}{d\{a\}}\{KXSD\} &= \sum_{k=1}^{n} \frac{2}{\Delta x}\frac{\partial^3 T}{\partial \gamma_{k-1}{}^3}\frac{d\{N3_{k-1}\}^T}{d\xi}\{a\}\{N3_{k-1}\}\{N3_{k-1}\}^T \\ &+ \sum_{k=1}^{n} \frac{2}{\Delta x}\left(\frac{\partial^2 T}{\partial \gamma_{k-1}{}^2}\right)\{N3_{k-1}\}\frac{d\{N3_{k-1}\}^T}{d\xi} \\ &- \sum_{k=1}^{n} \frac{2}{\Delta x}\frac{\partial^3 T^*}{\partial \gamma_{k-1}{}^3}\frac{d\{N3_{k-1}\}^T}{d\xi}\{a\}\{N3_{k-1}\}\{N3_{k-1}\}^T \\ &- \sum_{k=1}^{n} \frac{2}{\Delta x}\left(\frac{\partial^2 T^*}{\partial \gamma_{k-1}{}^2}\right)\{N3_{k-1}\}^T\frac{d\{N3_{k-1}\}^T}{d\xi} \end{aligned} \tag{B.41}$$

is found. Finally, combining equation B.40 and B.41, the term $\frac{d\{KXS_2\}}{d\{a\}}$ is expressed as a matrix

$$\frac{d}{d\{a\}}\{KXS_2\} = \frac{d}{d\{a\}}\{KXS\} + \frac{d}{d\{a\}}\{KXSD\} = [MXS] \tag{B.42}$$

where

$$[MXS] = \sum_{k=1}^{n} \frac{2}{\Delta x}\frac{\partial^3 T}{\partial {\gamma_{k-1}}^3}\frac{d\{N3_{k-1}\}^T}{d\xi}\{a\}\{N3_{k-1}\}\{N3_{k-1}\}^T$$

$$\sum_{k=1}^{NA} \frac{2}{\Delta x}\frac{\partial^3 T^*}{\partial \gamma_{k-1}^3}\frac{d\{N3_{k-1}\}^T}{d\xi}\{a\}\{N3_{k-1}\}\{N3_{k-1}\}^T$$

$$+\sum_{k=1}^{n} \frac{2}{\Delta x}\left(\frac{\partial^2 T}{\partial {\gamma_{k-1}}^2}\right)\left[\{N3_{k-1}\}\frac{d\{N3_{k-1}\}^T}{d\xi} + \frac{d\{N3_{k-1}\}}{d\xi}\{N3_{k-1}\}^T\right]$$

$$+\sum_{k=1}^{n} \frac{2}{\Delta x}\left(\frac{\partial^2 T^*}{\partial {\gamma_{k-1}}^2}\right)\left[\{N3_{k-1}\}\frac{d\{N3_{k-1}\}^T}{d\xi} + \frac{d\{N3_{k-1}\}}{d\xi}\{N3_{k-1}\}^T\right]$$

Following the same procedure to find $\frac{d\{KXZS_2\}}{d\{a\}}$, the following equations are obtained. First, the term $\frac{d\{KXZS\}}{d\{a\}}$ was expressed as

$$\frac{d}{da}\{KXZS\} = \left(\frac{1-\varsigma}{2}\right)\frac{1}{\gamma_{k-1}}\left[\frac{\partial f_n(\gamma_{n-1})}{\partial \gamma_{n-1}} - \frac{f_n(\gamma_{n-1})}{\gamma_{n-1}}\right]$$

$$\{N3_{n-1}\}\{N3_{n-1}\}^T \tag{B.43}$$

and then the term $\frac{d\{KXZSD\}}{d\{a\}}$ was found to equal

$$\frac{d}{d\{a\}}\{KXZSD\} = \left(\frac{1-\varsigma}{2}\right)\frac{1}{\gamma_{k-1}}\left[\frac{\partial f_n(\gamma_{n-1})}{\partial \gamma_{n-1}} - \frac{f_n(\gamma_{n-1})}{\gamma_{n-1}}\right]$$

$$\{N3_{n-1}\}\{N3_{n-1}\}^T + \left(\frac{1-\varsigma}{2}\right)\left[\frac{1}{\gamma_{n-1}}\right.$$

$$\frac{\partial^2 f_n(\gamma_{n-1})}{\partial \gamma_{n-1}^2} - \frac{2}{\gamma_{n-1}^2}\frac{\partial f_n(\gamma_{n-1})}{\partial \gamma_{n-1}}$$

$$\left.+\frac{2 f_n(\gamma_{n-1})}{\gamma_{n-1}^3}\right]\{a\}^T\{N3_{n-1}\}$$

$$\{N3_{n-1}\}\{N3_{n-1}\}^T \tag{B.44}$$

Adding equations B.43 and B.44 and defining the matrix $[MXZS]$ as

$$[MXZS] = \frac{d}{d\{a\}}\{KXZS\}^T + \frac{d}{d\{a\}}\{KXZSD\}^T \tag{B.45}$$

where

$$[MXZS] = 2\left(\frac{1-\varsigma}{2}\right)\frac{1}{\gamma_{n-1}}\left[\frac{\partial f_n(\gamma_{n-1})}{\partial\gamma_{n-1}} - \frac{f_n(\gamma_{n-1})}{\gamma_{n-1}}\right]$$

$$\{N3_{n-1}\}\{N3_{n-1}\}^T + \left(\frac{1-\varsigma}{2}\right)\left[\frac{1}{\gamma_{n-1}}\frac{\partial^2 f_n(\gamma_{n-1})}{\partial\gamma_{n-1}^2}\right.$$

$$\left. - \frac{2}{\gamma_{n-1}^2}\frac{\partial f_n(\gamma_{n-1})}{\partial\gamma_{n-1}} + \frac{2f_n(\gamma_{n-1})}{\gamma_{n-1}^3}\right]$$

$$\{a\}^T\{N3_{n-1}\}\{N3_{n-1}\}\{N3_{n-1}\}^T$$

Therefore, $\frac{d[BS_2]^T}{d\{a\}}\sigma$ is equal to

$$\frac{d[BS_2]^T}{d\{a\}}\sigma = \sigma_x[MXS] + \tau_{xz}[MXZS] \tag{B.46}$$

and the matrix $[K_t]$ is given by the expression

$$\begin{aligned}[K_t] = & \int_V \left[\sigma_x[MXS] + \sigma_x[MX] + \tau_{xz}[MXZS]\right] \\ & \left[[B_0^T + BS_2^T + B_2^T][D_t][B_0 + BS_2 + B_2]\right] dV \end{aligned} \tag{B.47}$$

It should be noted that the matrices $[MX]$, $[MXS]$ and $[MXZS]$ are all symmetric.

B.5 Determining the Unknowns

From above formulation of $[K]$ and $[K_t]$ matrices, the value of the unknowns $f_n(\gamma_{n-1})$ and its derivatives with respect to γ_{n-1}'s are required in each iteration. Previous solution of γ's are substituted into the system of nonlinear equations to obtain an improved solution. However, for complicated functions of $g(\gamma)$, $f_n(\gamma_{n-1})$ and other unknowns are implicit functions. Thus, the value of the unknowns cannot be directly obtained. The Newton-Raphson method can also be applied to determine these unknowns. Using the stress-strain relationship stated in eqaution B.2 of

$$\tau = (P_0 + P_1\gamma)\left(1 - \exp\left(\frac{-k}{P_0}\gamma\right)\right)$$

where P_0, P_1 and k are known parameters of a layer's material. Applying shear stress continuity between (n-1)-th and n-th layers, the continuity equation becomes

$$(P_{0_{n-1}} + P_{1_{n-1}}\gamma_{n-1})\left(1 - \exp\left(\frac{-k_{n-1}}{P_{0_{n-1}}}\gamma_{n-1}\right)\right)$$

$$= (P_{0_n} + P_{1_n}\gamma_{n_0})\left(1 - \exp\left(\frac{-k_n}{P_{0_n}}\gamma_{n_0}\right)\right) \quad \text{(B.48)}$$

From this equation, it can be seen that γ_{n_0} (equivalent to $f_n(\gamma_{n-1})$) cannot be directly obtained. Substituting the material parameters and the previous solution of γ_{n-1} into equation B.48, the left-hand-side of the equation becomes a constant C_{n-1}. In order to express the continuity equation in a Newton-Raphson format, the constant is move to the right-hand-side of the equation which gives

$$\Phi(\gamma_{n_0}) = (P_{0_n} + P1_n\gamma_{n_0})\left(1 - \exp\left(\frac{-k_n}{P_{0_n}}\gamma_{n_0}\right)\right) - C_{n-1} \quad \text{(B.49)}$$

Using Newton-Raphson method, the value γ_{n_0} is then determined. Same procedure can be applied to determine the other unknowns of $\frac{\partial(f_n(\gamma_{n-1}))}{\partial\gamma_{n-1}}$, $\frac{\partial^2(f_n(\gamma_{n-1}))}{\partial\gamma_{n-1}^2}$ and $\frac{\partial^3(f_n(\gamma_{n-1}))}{\partial\gamma_{n-1}^3}$.

Appendix C

Sample Calculation of Equivalent Elastic Moduli

The equivalent elastic moduli ($E_{x_{eq}}$ and $E_{y_{eq}}$) for the various corrugated layers in a 18 layers (90-26C-42-42-26C-90) beam were calculated. The location of the corrugated layers in the beam was found to have little effect on the computed moduli of the layers.

The properties of the 26C corrugation are given as:

$$h = 0.2032 \text{ mm.} \qquad l = 3.8989 \text{ mm.}$$

$$f = 3.6068 \text{ mm.}$$

$$E_x = 3.1717 \times 10^3 MPa \quad E_y = 1.5169 \times 10^3 MPa$$

For a 18 layers beam with 8 layers of corrugation, the distances d's from the midplanes of the corrugated layers to the z=0 plane are shown in Figure C.1 and in Table C.1.

Substituting the corrugation's properties into the half wave-length equation (Section 5.2.1) gives

$$\begin{aligned} s &= 3.8989\left[1 + \frac{\pi^2}{4}\left(\frac{3.6068}{3.8989}\right)^2\right] \\ &= 12.1310 \text{ mm.} \end{aligned}$$

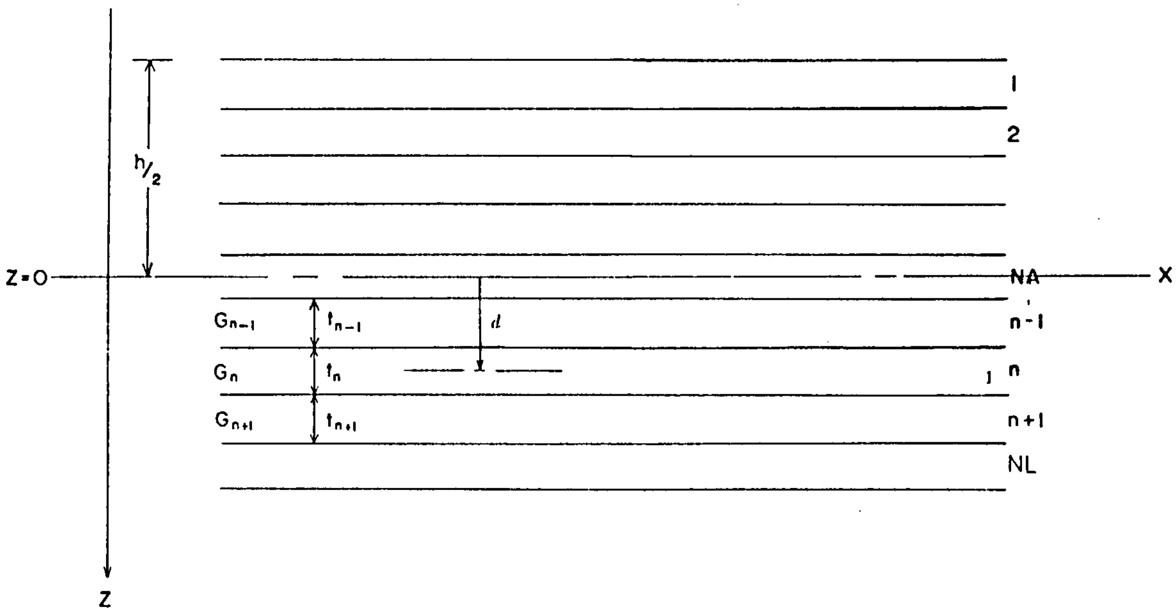

Figure C.1: Distances d's of the 18-layers beam

The moment of inertia I in the cross-machine direction is equal to

$$I = \frac{3.6068^2(0.2032)}{2}\left[1 - \frac{0.81}{1+2.5\left(\frac{3.6068}{3.8989\times 2}\right)^2}\right]$$

$$= 0.6242 \text{ mm.}^3$$

For layer #3 the bending rigidties D_x and D_y are respectively found to be:

$$D_x = \frac{3.8989}{12.1310}(3.1717)\frac{0.2302^3}{12}$$

$$+3.1717(0.2032)(5.8674)^2\frac{3.8989}{12.131}$$

$$= 7.131 \text{ kN-mm}$$

$$D_y = 1.5169(0.6242) + 1.5169\frac{12.131}{3.8989}(0.2032)(5.8674)^2$$

$$= 33.963 \text{ kN-mm}$$

Substituting these two values into the equivalent elastic moduli equations, the moduli are found to equal

$$E_{x_{eq}} = \frac{7.131}{\left[\frac{3.6068^3}{12} + (3.6068)(5.8674)^2\right]}$$

$$= 0.05568 \text{ kN/mm}^2$$

$$= 55.68 MPa$$

$$E_{y_{eq}} = \frac{33.963}{\left[\frac{3.6068^3}{12} + (3.6068)(5.8674)^2\right]}$$

$$= 0.2652 \text{ kN/mm}^2$$

$$= 265.2 MPa$$

Similar calculation for other corrugated layers in the above 18-layers (90-26C-42-42-26C-90) beam gives the following results:

Layer # n	d_n (mm.)	$E_{x_{eq}}$ (MPa)	$E_{y_{eq}}$ (MPa)
1	13.6906	57.1	265.8
2	9.7790	56.8	265.6
3	5.8674	55.7	265.2
4	1.9558	44.8	260.7

Table C.1: Computed Equivalent Elastic Moduli

From table C.1, the distance d_n is shown to have little effect on the computed equivalent elastic moduli. Therefore, $E_{x_{eq}}$ and $E_{y_{eq}}$ are approximated respectively as 55.2 MPa and 262.0 MPa for all corrugated layers.

Appendix D

Program Listing

A listing of the program CUBES is presented in this appendix. The computer language FORTRAN is used to develop the program. Also, it should be mentioned that a user's manual describing the data file necessary to run the program is available.

```
      IMPLICIT REAL*8(A - H,O - Z)
C
C     THIS PROGRAM PERFORMS A FINITE ELEMENT ANALYSIS
C     OF A STRUCTURE USING BEAM BENDING ELEMENT WITH SHEAR
C     INCLUDED.  MAXIMUM OF 10 ELEMENTS AND 20 LAMINAE IS ALLOWED.
C     CUBIC INTERPOLATIONS ARE USED FOR BOTH LONGITUDINAL AND
C     LATERAL DEFLECTIONS (U AND W).
C
C     DEFINES ALL VARIABLES BEGINNING WITH THE LETTERS A TO H AND
C     AND O TO Z AS DOUBLE PRECISION
C
C     DEFINE VARIABLES
C         EL     = CALCULATED ELEMENT LOAD VECTOR FROM PREVIOUS
C                  SOLUTION
C         SN     = ELEMENT STIFFNESS MATRIX
C         ALOAD  = STRUCTURE LOAD VECTOR
C         ICO    = ELEMENT CORNER NODE NUMBERS
C         X      = GLOBAL COORDINATES OF NODES IN FINITE ELEMENT GRID
C         IX     = BOUNDARY CONDITION INTEGER VECTOR, 1 FOR VARIABLE
C                      TO BE RETAINED, 0 FOR ITS ELIMINATION
C         NO,ALCOPY = MISC. STORAGE
C         NNODEL= NO. OF NODES PER ELEMENT
C         NE     = TOTAL NO. OF ELEMENTS IN STRUCTURE
C         NNODES= TOTAL NO. OF NODES IN STRUCTURE
C         NMAT   = GROSS NO. OF VARIABLES IN STRUCTURE
C         NVAR   = NO. OF VARIABLES PER NODE
C         D      = ELASTIC MODULUS MATRIX
C         E      = ELASTIC MODULUS
C         G      = SHEAR MODULUS
C         ESOLTN= ELEMENT SOLUTION VECTOR
C         PSOLTN= PREVIOUS SOLUTION VECTOR
C         SOLTN  = PRESENT SOLUTION VECTOR
C         TSOL1  = INCREMENTAL SOLUTION VECTOR
C         TK     = STRUCTURE TANGENT STIFFNESS MATRIX
C         TN     = ELEMENT TANGENT STIFFNESS MATRIX
C         NDIMD  = DIMENSION OF THE ELASTIC MODULUS MATRIX, D
C         NL     = NUMBER OF LAMINA IN BEAM
C         NDIMB  = DIMENSION OF THE ELEMENTAL MATRICES
C         TH     = THICKNESS OF THE LAMINA
C         Z      = Z COORDINATE OF THE MIDPLANE OF THE LAMINA
C         NPRST= STRESS PRINTOUT CONTROL PARAMETERS
C                  IF NPRST=1, STRESSES ARE PRINTED
C                  IF NPRST=0, NO STRESSES ARE PRINTED
C         NA = LAMINA WHERE Z=0 AXIS IS LOCATED
C         LOCAL   = LOCAL COORDINATE OF NA-TH LAMINA WHERE
C                  Z=0 AXIS IS LOCATED
C         DERN1 = VECTOR OF THE 1-ST DERIVATIVE OF U W.R.T.
C                  X-COORDINATE AT EACH INTEGRATION POINT
C         EDERN3= VECOTR OF THE 1-ST DERIVATIVE OF SHEAR STRAIN
C                  W.R.T. X-COORDINATE AT EACH INTEGRATION POINT
C
C         ALL SYMMETRIC MATRICES ARE STORED AS COLUMN VECTORS
C         CONTAINING ONLY THE LOWER HALF OF THE MATRICES.
C
      DIMENSION EL(48), ALOAD(264), ALCOPY(264), ALOADI(264)
      DIMENSION IX(264), NO(2), E(20), ICO(10,2), XX(2)
      DIMENSION SOLTN(264), PSOLTN(264), ESOLTN(48)
      DIMENSION TN(1176), TK(12672)
```

```
      DIMENSION TSOL1(264), STRESS(2)
      COMMON /CONST1/ D(2,2), G(20), TH(20), DELX, DELY, NDIMD
      COMMON /CONST2/ Z(20)
      COMMON /CONST3/ DERN1(48), EDERN3(48)
      COMMON /CONST4/ NL, NDIMB
      COMMON /CONST5/ LOCAL, NA
      COMMON /CONST7/ X(11)
C
C     DEFINE AS A 1-DIMENSIONAL PROBLEM WITH ONLY ONE BENDING
C     AND ONE SHEAR STRAIN AS UNKNOWNS.
C
      NDIMD = 2
C
C     SET ITERATION COUNT TO ZERO
C
      ITER = 0
C
C     CALL SUBROUTINE TO READ IN ALL FINITE ELEMENT GRID DATA
C     FOR THE PROBLEM
C
      CALL LAYOUT(X, ICO, NNODEL, IX, NE, NNODES, NMAT, NVAR)
C
C     FIND DIMENSION OF ELEMENT ARRAYS FOR THE PROBLEM BEING ANALYSED
C
      NDIMB = NNODEL * NVAR
C
C     INITIALIZE ARRAY TO HAVE ZERO ENTRIES
C
      DO 10 JJ = 1, NDIMD
        DO 10 KK = 1, NDIMD
C
C     ALOADI = INCREMENTAL LOADING VECTOR WHICH STORES THE
C              THE VALUE OF THE LOAD BEING INCREMENTED.
C
   10 D(JJ,KK) = 0.D0
      DO 20 I = 1, NMAT
        TSOL1(I) = 0.D0
        ALOAD(I) = 0.D0
        ALOADI(I) = 0.D0
   20 SOLTN(I) = 0.D0
C
C     CALL SUBROUTINE TO READ IN THE NUMBER OF LAMINAE, THICKNESS,
C     ELASTIC MODULUS, SHEAR MODULUS, AND Z-COORDINATE OF MID-PLANE
C     OF EACH LAMINA. OTHER PARAMETERS READ IN ARE: THE LOAD,
C     ULTIMATE TENSILE AND COMPRESSIVE CAPACITIES, AND THE STRESS
C     OUTPUT CONTROL VALUE.
C
      CALL PROP(E, ALOAD, NVAR, NPRST, NMAT, ALOADI, INCRE)
C
C     CHECK THE NUMBER OF DEGREES OF FREEDOM PER NODE
C
      NDOF = 4 + NL
      IF (NDOF .NE. NVAR) GO TO 570
C
C     PLACE ZEROES AND CONSTANTS INTO VECTOR OF THE 1-ST DERIVATIVE
C     OF SHEAR (EDERN3) W.R.T. X-COORDINATE RESULTED FROM THE
C     NUMBERING SEQUENCE OF THE LAMINAE.
C
```

```
      DO 30 N2 = 1, NDIMB
   30 EDERN3(N2) = 0.D0
C
      DO 70 K = 1, NA
        IF (K .NE. NA) GO TO 40
        EC = 0.5D0 * TH(K) * (LOCAL*0.5D0+0.25D0*LOCAL**2 + 0.25D0)
        GO TO 60
   40   K1 = K + 1
        IF (K1 .NE. NA) GO TO 50
        EC = 0.5D0 * TH(K) + 0.5D0 * G(K) * TH(K1) * (0.5D0*LOCAL - 0.
     1  25D0*LOCAL**2 + 0.75D0) / G(K1)
        GO TO 60
   50   EC = 0.5D0 * TH(K) + 0.5D0 * G(K) * TH(K1) / G(K1)
   60   K4 = K + 4
        NEXT = NL + K4 + 4
        EDERN3(K4) = -EC * 0.5D0
        EDERN3(NEXT) = -EDERN3(K4)
   70 CONTINUE
C
C     MAIN PROGRAM
C
C     LHB  = LOWER HALF BANDWIDTH OF TANGENTIAL STIFFNESS MATRIX
C     NA   = NUMBER OF ELEMENT IN THE VECTOR REPRESENTING THE
C             TANGENTIAL STIFFNESS MATRIX
      LHB = NDIMB
      NA = NMAT * LHB
CC
CC    CHECK FOR INCREMENTAL REQUIREMENTS
CC
CC    IF (INCRE .LE. 1) GO TO 120
C
C     REMOVE FROM THE LOAD VECTOR THE LOAD BEING INCREMENTED
C
      DO 80 K1 = 1, NMAT
   80 ALOAD(K1) = ALOAD(K1) - ALOADI(K1)
      DO 560 KK = 1, INCRE
        ITER = 0
C
C     INCREMENT LOAD
C
        DO 90 MM = 1, NMAT
   90   ALOAD(MM) = ALOAD(MM) + ALOADI(MM) / INCRE
        WRITE (6,100)
        WRITE (1,100)
        WRITE (2,100)
  100   FORMAT (/, '---------------------------------------------------')
        WRITE (6,110) KK
        WRITE (1,110) KK
        WRITE (2,110) KK
  110   FORMAT (/, 1X, 'SOLUTION AT LOAD INCREMENT:', I2)
  120   CONTINUE
C
C     INITIALIZE STIFFNESS MATRIX AND A COPY OF THE LOAD VECTOR
C     TO HAVE ZERO ENTRIES.  ALSO ASSIGN EXISTING SOLUTION AS
C     PREVIOUS SOLUTION AT THE START OF A NEW ITERATION.
C
        DO 130 I = 1, NA
  130   TK(I) = 0.D0
```

```
        DO 140 I = 1, NMAT
          ALCOPY(I) = 0.DO
  140   PSOLTN(I) = SOLTN(I)
        DO 190 IEL = 1, NE
          DO 180 ILAM = 1, NL
            DO 160 I = 1, NNODEL
              NO(I) = ICO(IEL,I)
              DO 150 II = 1, NVAR
                K = (NO(I) - 1) * NVAR + II
                L = (I - 1) * NVAR + II
C
C      IDENTIFY THE PARTICULAR ELEMENT SOLUTION VECTO FROM
C      THE ENTIRE SOLUTION VECTOR AND IDENTIFY THE X-COORDINATE
C      OF THE ELEMENT'S NODE.
C
  150         ESOLTN(L) = PSOLTN(K)
  160       XX(I) = X(NO(I))
C
C     FORM ELASTICITY MATRIX FOR THE ILAM-TH LAYER OF THE
C     BEAM ELEMENT, D(2 x 2) MATRIX
C
            D(1,1) = E(ILAM)
            D(2,2) = G(ILAM)
C
C     CALCULATE THE ELEMENT'S LENGTH
C
            DELX = XX(2) - XX(1)
C
C     CALL SUBROUTINE TO CALCULATE ELEMENT STIFFNESS MATRIX,SN
C     AND ELEMENT TANGENT STIFFNESS MATRIX,TN USING NUMERICAL
C     INTEGRATION OF GAUSSIAN QUADRATURE
C
            CALL NSTIFF(ILAM, ESOLTN, EL, TN)
C
C     CALL SUBROUTINE TO INSERT ELEMENT STIFFNESS MATRIX AND
C     LOAD VECTOR INTO STRUCTURE STIFFNESS MATRIX AND LOAD VECTOR
C
            DO 170 J = 1, NNODEL
              DO 170 JJ = 1, NVAR
                K = (NO(J) - 1) * NVAR + JJ
                L = (J - 1) * NVAR + JJ
C
C     ADD ALL THE ELEMENTAL CONTRIBUTION TO FORM THE
C     VECTOR OF LOAD ON THE STRUCTURE CURVE. ALSO PLACE THE
C     ELEMENT'S TANGENTIAL STIFFNESS MATRIX INTO THE OVERALL
C     STRUCTURE STIFFNESS MATRIX.
C
  170       ALCOPY(K) = ALCOPY(K) + EL(L)
            CALL SETUP(NO, NVAR, TK, TN, NDIMB)
  180     CONTINUE
  190   CONTINUE
C
C     IMPOSE HOMONGENOUS BOUNDARY CONDITIONS
C
        CALL DISCRD(TK, NMAT, ALCOPY, ALOAD, IX, NDIMB)
        DO 200 I = 1, NMAT
C
C     FIND THE DIFFERENCE BETWEEN THE PRESENT LOAD VALUE AT THE
```

```
C     STRUCTURE CURVE AND THE APPLIED LOAD VALUE.  THE CURVE'S
C     LOAD VALUE IS FOUND BY SUBSTITUTING THE PREVIOUS SOLUTION
C     VECTOR.  THE SYSTEM OF EQUATIONS IS THEN SOLVED USING
C     CHOLESKY METHOD AND A NEW SOLUTION IS CALCULATED.
C
  200   TSOL1(I) = ALCOPY(I) - ALOAD(I)
        CALL DECOMP(NMAT, NDIMB, TK)
        CALL SOLV(NMAT, NDIMB, TK, TSOL1)
        DO 210 I = 1, NMAT
  210   SOLTN(I) = PSOLTN(I) - TSOL1(I)
C
C     CALL SUBROUTINE TO CHECK THE ERROR BETWEEN EXISTING
C     SOLUTION AND PREVIOUS SOLUTION
C
        CALL ERR(PSOLTN, SOLTN, IERR, NMAT)
  220   ITER = 1 + ITER
C
C     WRITE THE DISPLACEMENT VECTOR
C
        IF (ITER .EQ. 1) GO TO 240
        IF (IERR .NE. 0) GO TO 120
        WRITE (6,230)
  230   FORMAT (/, '-----------NONLINEAR SOLUTION-------')
        GO TO 260
  240   WRITE (6,250)
  250   FORMAT (/, '-----------LINEAR SOLUTION-----------')
  260   WRITE (6,270) ITER
  270   FORMAT (/, 'NUMBER OF ITERATION :', 2X, I2)
        WRITE (6,280)
  280   FORMAT (/, 1X, 'SOLUTION VECTOR')
C
C     PRINT OUT THE SOLUTION VECTOR FOR THE LINEAR AND THE
C     FINAL SOLUTIONS.
C
        NDS = NMAT / NVAR
        NP = NMAT
        NDSP = NDS
        IF (NDS .LE. 6) GO TO 290
        NP = 6 * NVAR
        NDSP = 6
  290   WRITE (6,300) (I,I=1,NDSP)
  300   FORMAT (/, 2X, 'NODE #', 2X, 6(I2,12X))
        WRITE (6,310) (SOLTN(J),J=1,NP,NVAR)
  310   FORMAT (/, 3X, 'W', 3X, 6(2X,G12.4))
        WRITE (6,320) (SOLTN(J),J=2,NP,NVAR)
  320   FORMAT (/, 3X, 'Wx', 2X, 6(2X,G12.4))
        WRITE (6,330) (SOLTN(J),J=3,NP,NVAR)
  330   FORMAT (/, 3X, 'U', 3X, 6(2X,G12.4))
        WRITE (6,340) (SOLTN(J),J=4,NP,NVAR)
  340   FORMAT (/, 3X, 'Ux', 2X, 6(2X,G12.4))
        WRITE (6,350)
  350   FORMAT (/, 1X, 'SHEAR STRAIN')
        WRITE (6,360)
  360   FORMAT (/, 1X, 'LAMINA #')
        WRITE (6,370) (SOLTN(J),J=5,NP,NVAR)
  370   FORMAT (/, 4X, '1', 2X, 6(2X,G12.4))
        IF (NVAR .LT. 6) GO TO 400
        DO 380 I = 6, NVAR
```

```
         II = I - 4
  380    WRITE (6,390) II, (SOLTN(J),J=I,NP,NVAR)
  390    FORMAT (/, 2X, I3, 2X, 6(2X,G12.4))
C
C     PRINT OUT STATEMENTS FOR STRUCTURE WITH MORE THAN
C     SIX NODES.
C
  400    IF (NMAT .LE. NP) GO TO 430
         WRITE (6,410)
  410    FORMAT (/, ' ')
         WRITE (6,300) (I,I=7,NDS)
         NP1 = NP + 1
         WRITE (6,310) (SOLTN(J),J=NP1,NMAT,NVAR)
         NP2 = NP1 + 1
         WRITE (6,320) (SOLTN(J),J=NP2,NMAT,NVAR)
         NP3 = NP2 + 1
         WRITE (6,330) (SOLTN(J),J=NP3,NMAT,NVAR)
         NP4 = NP3 + 1
         WRITE (6,340) (SOLTN(J),J=NP4,NMAT,NVAR)
         NP5 = NP4 + 1
         WRITE (6,350)
         WRITE (6,360)
         WRITE (6,370) (SOLTN(J),J=NP5,NMAT,NVAR)
         IF (NVAR .LT. 6) GO TO 430
         NP6 = NP5 + 1
         NPVAR = NP + NVAR
         II = 1
         DO 420 I = NP6, NPVAR
           II = 1 + II
  420    WRITE (6,390) II, (SOLTN(J),J=I,NMAT,NVAR)
  430    CONTINUE
         IF (ITER .EQ. 1) GO TO 120
C
C     CHECK FOR STRESS PRINTOUT CONTROL
C
         IF (NPRST .EQ. 0) GO TO 560
         WRITE (6,440)
         WRITE (1,440)
         WRITE (2,440)
  440    FORMAT (/, 1X, ' ELEMENT STRESSES')
         WRITE (6,450)
  450    FORMAT (/, 1X, 'ELEMENT STRESSES AT INTEGRATION POINTS ALONG X')
         WRITE (6,460)
  460    FORMAT (/, 1X, 'AT CENTER OF EACH LAMINA')
         WRITE (6,470)
  470    FORMAT (/, 1X, '(BENDING STRESS, SHEAR STRESS)')
         WRITE (1,480)
  480    FORMAT (/, 1X, ' BENDING STRESSES ONLY AT ALL INTEGRATION POINTS
      1')
         WRITE (2,490)
  490    FORMAT (/, 1X, ' SHEAR STRESSES ONLY AT ALL INTEGRATION POINTS')
         DO 550 IEL = 1, NE
           WRITE (6,500) IEL
           WRITE (1,500) IEL
           WRITE (2,500) IEL
  500      FORMAT (/, 1X, 'ELEMENT #', 2X, I3)
           WRITE (6,510)
  510      FORMAT (/, 1X, 'LAMINA #', 2X, '1 ST INTEGRATION POINT', 2X,
```

```
     1            '5 TH INTEGRATION POINT')
            DO 540 ILAM = 1, NL
              DO 530 I = 1, NNODEL
                NO(I) = ICO(IEL,I)
                DO 520 II = 1, NVAR
                  K = (NO(I) - 1) * NVAR + II
                  L = (I - 1) * NVAR + II
  520           ESOLTN(L) = SOLTN(K)
  530         XX(I) = X(NO(I))
              D(1,1) = E(ILAM)
              D(2,2) = G(ILAM)
              DELX = XX(2) - XX(1)
C
C     CALL STRESS SUBROUTINE.
C
              CALL FSTRES(ILAM, ESOLTN, IEL, NE, IFAIL)
  540       CONTINUE
  550     CONTINUE
          IF (IFAIL .EQ. 1) GO TO 570
  560 CONTINUE
  570 CONTINUE
      STOP
      END
C
C     END OF MAIN PROGRAM
C
      SUBROUTINE LAYOUT(X, ICO, NNODEL, IX, NE, NNODES, NMAT, NVAR)
C
C     THIS SUBROUTINE READS IN ALL FINITE ELEMENT DATA FOR A PROBLEM
C
C
C     X(I)          = X COORDINATES OF I-TH NODE (RETURNED)
C     ICO(I,J)      = J NODE NUMBERS FOR I-TH ELEMENT (RETURNED)
C     NNODEL        = NUMBER OF NODES PER ELEMENT (RETURNED)
C     IX            = BOUNDARY CONDITION CODE VECTOR, =1 IF VARIABLE
C                         IS TO BE RETAINED, =0 IF VARIABLE IS TO BE
C                         ZEROED(RETURNED)
C     NE            = NUMBER OF ELEMENTS IN TOTAL PROBLEM (RETURNED)
C     NNODES        = TOTAL NUMBER OF NODES IN PROBLEM (RETURNED)
C     NMAT          = GROSS NUMBER OF VARIABLES IN PROBLEM (RETURNED)
C     NVAR          = NO. OF VARIABLES PER NODE (RETURNED)
C
      IMPLICIT REAL*8(A - H,O - Z)
      DIMENSION X(11), ICO(10,2), IX(264)
      READ (5,10) NE, NNODES, NVAR, NNODEL
   10 FORMAT (4I5)
      WRITE (6,20) NE, NNODES, NVAR, NNODEL
   20 FORMAT (//, '   NO. OF ELEMENTS ', I5, 5X, 'NO. OF NODES', I5, 5X,
     1        'VARIABLES PER NODE', I5, '   NODES PER ELEMENT', I5, /)
      WRITE (6,30)
   30 FORMAT (/, 3X, ' NODE', 7X, ' X-CORD', 6X, '   VARIABLES')
      DO 60 I = 1, NNODES
        I2 = NVAR * I
        I1 = I2 - NVAR + 1
        READ (5,40) X(I), (IX(J),J=I1,I2)
   40   FORMAT (F12.4, 24I2)
        WRITE (6,50) I, X(I), (IX(J),J=I1,I2)
   50   FORMAT (1X, I5, 5X, F12.4, 5X, 24I4)
```

```
   60 CONTINUE
      WRITE (6,70)
   70 FORMAT (///, 5X, 'ELEMENT', 5X, 'NODE NUMBERS ')
      DO 90 I = 1, NE
C
C     READ IN I-TH ELEMENT'S NODE NUMBER
C
        READ (5,80) (ICO(I,J),J=1,NNODEL)
   80   FORMAT (16I4)
   90 WRITE (6,100) I, (ICO(I,J),J=1,NNODEL)
  100 FORMAT (5X, I5, 5X, 4I4, 4I7)
C
C     FIND THE GROSS NUMBER OF VARIABLES IN STRUCTURE
C
      NMAT = NVAR * NNODES
      RETURN
      END
C
      SUBROUTINE PROP(E, ALOAD, NVAR, NPRST, NMAT, ALOADI, INCRE)
C
C     SUBROUTINE TO READ IN THE MATERIAL PROPERTIES,
C     THICKNESS, AND Z-COORDINATE OF MID-PLANE OF LAMINA
C
C     NL = NUMBER OF LAMINA(RETURNED)
C     TH = THICKNESS OF LAMINA(RETURNED)
C     E  = ELASTIC MODULUS OF LAMINA(RETURNED)
C     G  = SHEAR MODULUS OF LAMINA(RETURNED)
C     Z  = Z-COORDINATE OF MID-PLANE OF LAMINA(RETURNED)
C     ALOAD   =STRUCTURE LOAD VECTOR (RETURNED)
C     DELX     =ELEMENT LENGTH IN X DIRECTION (RETURNED)
C     DELY     =ELEMENT WIDTH IN Y DIRECTION (RETURNED)
C     NA    =LAMINA NUMBER WHERE Z=0 IS LOCATED (RETURNED)
C     LOCAL     =VALUE IN NATURAL COORDINATE WHERE Z=0 IS
C                 LOCATED(RETURNED)
C     NPRST  =STRESS OUTPUT PARAMETER
C
      IMPLICIT REAL*8(A - H,O - Z)
      COMMON /CONST1/ D(2,2), G(20), TH(20), DELX, DELY, NDIMD
      COMMON /CONST2/ Z(20)
      COMMON /CONST4/ NL, NDIMB
      COMMON /CONST5/ LOCAL, NA
      COMMON /CONST6/ ULTCOM, ULTEN
      COMMON /CONST7/ X(11)
      DIMENSION E(20), ALOAD(264), TLOAD(264), ALOADI(264)
      READ (5,10) NL, DELY
   10 FORMAT (I3, E10.4)
      WRITE (6,20) NL, DELY
   20 FORMAT (/, 1X, 'NUMBER OF LAMINA:', 2X, I3, 6X, 'BEAM WIDTH:', 2X,
     1       E10.4)
      WRITE (6,30)
   30 FORMAT (/, 1X, 'LAMINA NUMBER', 2X, 'THICKNESS', 2X,
     1        'ELASTIC AND SHEAR MODULI', 2X, 'MID-PLANE Z-COORD')
      DO 50 I = 1, NL
        READ (5,40) TH(I), E(I), G(I), Z(I)
   40   FORMAT (4F12.3)
   50 WRITE (6,60) I, TH(I), E(I), G(I), Z(I)
   60 FORMAT (5X, I3, 8X, F10.4, 3X, F10.1, 2X, F10.1, 7X, F10.4)
C
```

```
C     NPRST = 1 STRESS OUTPUT IS PRINTED
C           = 0 STRESS OUTPUT IS SUPPRESSED
C
      READ (5,70) ULTCOM, ULTEN, NPRST
   70 FORMAT (2G12.3, I1)
      WRITE (6,80) ULTCOM
   80 FORMAT (/, ' ULTIMATE COMPRESSIVE STRENGTH = ', G12.4)
      WRITE (6,90) ULTEN
   90 FORMAT (/, ' ULTIMATE TENSILE STRENGTH = ', G12.4)
      IF (NPRST .EQ. 1) GO TO 110
      WRITE (6,100)
  100 FORMAT (/, '-------- NO STRESS OUTPUT --------')
      GO TO 130
  110 WRITE (6,120)
  120 FORMAT (/, '-----STRESS IS PRINTED FOR THE FINAL SOLUTION-----')
  130 READ (5,140) NA, LOCAL
  140 FORMAT (I3, E10.4)
      WRITE (6,150) NA, LOCAL
  150 FORMAT (/, 'Z=0 AXIS IS LOCATED IN LAMINA ', I3,
     1        ' NATURAL COORDINATES ARE:', 2X, E10.4)
C
C     READ IN CONCENTRATED LOADS ONLY
C     NPLOAD = NO. OF CONCENTRATED LOADS
C
      READ (5,160) NPLOAD
      IF (NPLOAD .EQ. 0) GO TO 230
  160 FORMAT (I3)
      WRITE (6,170)
  170 FORMAT (/, '---------- CONCENTRATED LOADS ----------')
      WRITE (6,180)
  180 FORMAT (/, 1X, 'LOAD #', 2X, 'NODE #', 2X, 'DEGREE OF FREEDOM',
     1        3X, 'LOAD VALUE', 4X, 'NO. OF INCREMENTS')
      INCRE = 1
      DO 210 II = 1, NPLOAD
        READ (5,190) NI, NV, PLOAD, INC
  190   FORMAT (2I3, E10.4, I2)
C
C     NI    = NODE NUMBER WHERE THE CONCENTRATED LOAD IS APPLIED
C     NV    = VARIABLE NUMBER OF THE NI-TH NODE WHERE THE CONCENTRATED
C             LOAD IS APPLIED
C     INC   = NO. OF LOAD INCREMENTS
C             IF INC = 1 NO INCREMENT
C             IF INC > 1 SAY =3 THEN THE CONCENTRATED LOAD IS DIVIDED
C                        INTO 3 EQUAL SEGMENTS AND THE VALUE IS
C                        STORED INTO THE PARAMETER INCRP
C
C     **NOTE** ONLY ONE PARTICULAR LOAD CAN BE INCREMENTED DURING
C              ANY SINGLE RUN OF THE PROGRAM
C
        M = NVAR * (NI - 1) + NV
C
C     STORE DEGREE OF FREEDOM, NUMBER OF INCREMENTS OF
C     THE CONCENTRATED LOAD THAT IS INCREMENTED AND THE
C     VALUE OF THE LOAD INTO SEPARATE VECTOR
C
        IF (INC .LE. 1) GO TO 200
        INCRE = INC
        ALOADI(M) = PLOAD
```

```
C
  200   ALOAD(M) = PLOAD
  210 WRITE (6,220) II, NI, M, ALOAD(M), INC
  220 FORMAT (/, 4X, I2, 4X, I2, 10X, I3, 12X, G12.4, 10X, I2)
  230 CONTINUE
C
C     READ IN LATERAL DISTRIBUTED LOAD IN Z DIRECTION WITH
C     STARTING VALUE QST AT NODEST-TH NODE TO ENDING VALUE QEN
C     AT NODEN-TH NODE
C
C     NQLOAD = NO. OF DISTRIBUTED LOADS
C     NODEST = NODE NUMBER WHERE THE DISTRIBUTED LOAD STARTS
C     NODEN  = NODE NUMBER WHERE THE DISTRIBUTED LOAD ENDS
C     QST    = STARTING LOAD VALUE
C     QEN    = ENDING LOAD VALUE
C
C     DISTRIBUTED LOADS ARE REPLACED BY CONSISTENT LOAD IN
C     THE PROGRAM
C
      READ (5,240) NQLOAD
      IF (NQLOAD .EQ. 0) GO TO 350
  240 FORMAT (I2)
      WRITE (6,250)
  250 FORMAT (/, 1X, '---------- DISTRIBUTED LOADS ----------')
      DO 340 II = 1, NQLOAD
        DO 260 L = 1, NMAT
  260   TLOAD(L) = 0.D0
        READ (5,270) NODEST, NODEN, QST, QEN, INC
  270   FORMAT (2I3, 2G12.4, I2)
        WRITE (6,290)
        WRITE (6,280) II, NODEST, NODEN, QST, QEN, INC
  280   FORMAT (/, 3X, I2, 6X, 2(I2,2X), 3X, 2(G12.4,2X), 10X, I2)
  290   FORMAT (/, 1X, 'LOAD #', 2X, 'NODE:FROM--TO', 2X,
     1         'LOAD VALUE:FROM--TO', 6X, 'NO. OF INCREMENTS')
        M2 = (NODEN - 1) * NVAR + 1
        M1 = (NODEST - 1) * NVAR + 1
  300   LOADEL = NODEN - NODEST
        QDIFF = (QEN - QST) / LOADEL
        DO 310 JJ = 1, LOADEL
          QLOAD1 = QST + QDIFF * (JJ - 1)
          QLOAD2 = QLOAD1 + QDIFF
          NODE = JJ + NODEST - 1
C
C     TLOAD = TEMPORARY LOAD VECTOR
C     CALL SUBROUTINE CONSLO TO FIND THE CONSISTENT LOAD
C     FOR ANY SINGLE ELEMENT WITH LOAD QLOAD1 AT THE 1-ST
C     NODE AND LOAD QLOAD2 AT THE 2-ND NODE.
C
          CALL CONSLO(QLOAD1, QLOAD2, TLOAD, NODE, NVAR, X)
  310   CONTINUE
        DO 320 I = 1, NMAT
  320   ALOAD(I) = TLOAD(I) + ALOAD(I)
        IF (INCRE .GT. 1) GO TO 340
        IF (INC .LE. 1) GO TO 340
        INCRE = INC
C
C     STORES THE DISTRIBUTED LOAD THAT IS BEING INCREMENTED INTO
C     A SEPARATE VECTOR
```

```
C
          DO 330 J = 1, NMAT
  330     ALOADI(J) = TLOAD(J)
  340 CONTINUE
  350 CONTINUE
      IF (NPRST .EQ. 0) GO TO 380
C
C     STORES THE FINAL SOLUTION'S BENDING STRESSES AND SHEAR
C     STRESSES AT ALL INTEGRATION POINTS INTO TWO SEPARATE
C     FILES.
C
      WRITE (1,360)
  360 FORMAT (/, ' STRESS OUTPUT FORMAT # 2  BENDING STRESSES', ' ONLY')
      WRITE (2,370)
  370 FORMAT (/, ' STRESS OUTPUT FORMAT # 3  SHEAR STRESSES', ' ONLY')
  380 CONTINUE
      RETURN
      END
C
      SUBROUTINE CONSLO(QLOAD1, QLOAD2, TLOAD, NODE, NVAR, X)
C
C     SUBROUTINE TO FIND THE CONSISTENT LOAD TO REPRESENT
C     THE DISTRIBUTED LOAD
C
      IMPLICIT REAL*8(A - H,O - Z)
      DIMENSION TLOAD(264), X(11)
      M1 = (NODE - 1) * NVAR + 1
      M2 = M1 + 1
      M3 = M1 + NVAR
      M4 = M3 + 1
      NODE1 = NODE + 1
      DELX = X(NODE1) - X(NODE)
      TLOAD(M1) = 0.35D0 * DELX * QLOAD1 + 0.15D0 * DELX * QLOAD2 +
     1TLOAD(M1)
      TLOAD(M2) = (DELX**2) * QLOAD1 / 20.D0+(DELX**2) * QLOAD2 / 30.D0+
     1TLOAD(M2)
      TLOAD(M3) = 0.15D0 * DELX * QLOAD1 + 0.35D0 * DELX * QLOAD2 +
     1TLOAD(M3)
      TLOAD(M4) = -(DELX**2) * QLOAD1 / 30.D0-(DELX**2) * QLOAD2 / 20.
     1D0+TLOAD(M4)
      RETURN
      END
C
      SUBROUTINE SETUP(NODES, NVAR, A, C, LHB)
C
C     PROGRAM TO SETUP MASTER STIFFNESS MATRIX FROM INDIVIDUAL
C     FINITE ELEMENT STIFFNESS MATRICES.
C
C     NODES      = INTEGER VECTOR CONTAINING ELEMENT NODE NUMBERS
C                  IN ORDER
C     NVAR       = NUMBER OF VARIABLES PER NODE
C     A          = MASTER STIFFNESS MATRIX WHICH MUST BE ZEROED
C                  BEFORE FIRST ENTRY TO SETUP
C     C          = ELEMENT STIFFNESS MATRIX
C     LHB        = LOWER HALF BANDWIDTH INCLUDING DIAGONAL
C
C
      IMPLICIT REAL*8(A - H,O - Z)
```

```
      COMMON /CONST4/ NL, NDIMB
      DIMENSION A(12672), C(1176), NODES(2)
      DO 50 I = 1, NDIMB
        IM = MOD(I,NVAR)
        IF (IM .NE. 0) GO TO 10
        NOI = I / NVAR
        IM = NVAR
        GO TO 20
   10   NOI = (I - IM) / NVAR + 1
   20   DO 50 J = 1, I
          IJ = I * (I - 1) / 2 + J
          JM = MOD(J,NVAR)
          IF (JM .NE. 0) GO TO 30
          NOJ = J / NVAR
          JM = NVAR
          GO TO 40
   30     NOJ = (J - JM) / NVAR + 1
C
C     FIND POSITIONS (M,N) ROW AND COLUMN OF MATRIX A, THE
C     STRUCTURE'S TANGENTIAL STIFFNESS MATRIX, AND PLACES
C     INTO THESE POSITIONS THE CORRESPONDING VALUE FROM C, THE
C     ELEMENT'S TANGENTIAL STIFFNESS MATRIX.
   40     M = (NODES(NOI) - 1) * NVAR + IM
          N = (NODES(NOJ) - 1) * NVAR + JM
          MN = (LHB - 1) * (N - 1) + M
   50 A(MN) = A(MN) + C(IJ)
      RETURN
      END
C
      SUBROUTINE DISCRD(TK, NMAT, ALCOPY, ALOAD, IX, LHB)
C
C     THIS SUBROUTINE APPLIES HOMOGENEOUS BOUNDARY CONDITIONS BY
C     PLACING ZEROS ON OFF DIAGONAL TERMS AND ONE ON THE DIAGONAL
C     TERM OF THE STIFFNESS MATRICES. ALSO THE LOAD VECTOR TERM
C     IS REPLACED BY THE HOMOGENEOUR BOUNDARY VALUE OF ZERO.
C
C     TK     =TANGENT STIFFNESS MATRIX (RETURNED)
C     NMAT   =GROSS SIZE OF PROBLEM (GIVEN)
C     ALOAD  =GIVEN LOAD VECTOR (RETURNED)
C     ALCOPY =CALCULATED LOAD VECTOR (RETURNED)
C     IX     =BOUNDARY CONDITION CODE VECTOR (GIVEN)
C     LHB    =HALF BANDWIDTH INCLUDING THE DIAGONAL TERM (GIVEN)
C
      IMPLICIT REAL*8(A - H,O - Z)
      DIMENSION IX(264), ALOAD(264), ALCOPY(264), TK(12672)
      MULTI = LHB - 1
      DO 50 L = 1, NMAT
        IF (IX(L) .NE. 0) GO TO 50
        IJ = MULTI * L + L - MULTI
        TK(IJ) = 1.D0
        ALCOPY(L) = 0.D0
        ALOAD(L) = 0.D0
        DO 10 N = 1, MULTI
          IJC = N + IJ
   10   TK(IJC) = 0.D0
        IF (L .LT. LHB) GO TO 30
        DO 20 N1 = 1, MULTI
          IJR = IJ - MULTI * N1
```

```
   20   TK(IJR) = 0.D0
        GO TO 50
   30   LL = L - 1
        DO 40 N2 = 1, LL
          IJR = IJ - MULTI * N2
   40   TK(IJR) = 0.D0
   50 CONTINUE
      RETURN
      END
C
      SUBROUTINE ERR(PSOLTN, SOLTN, IERR, NMAT)
C
C     SUBROUTINE TO CHECK THE NONLINEAR SOLUTION HAS
C     CONVERGED TO DESIRED ACCURACY IN PERCENT
C
C     PSOLTN   =PREVIOUS SOLUTION (GIVEN)
C     SOLTN    =PRESENT SOLUTION (GIVEN)
C     IERR     =ERROR INDICATOR : 0 ERROR COMPARISON OK (RETURNED)
C                                 1 ERROR COMPARISON FAILED
C     NMAT     =GROSS NUMBER OF EQUATIONS WITH BOUNDARY CONDITIONS
C               ELIMINATED
C     ALPER    =ALLOWABLE ERROR PERCENTAGE
C     ABPER    =ABSOLUTE VALUE OF CALCULATED ERROR PERCENTAGE
C
      IMPLICIT REAL*8(A - H,O - Z)
      DIMENSION PSOLTN(264), SOLTN(264)
      I = 0
      IERR = 0
C
C     SET THE ALLOWABLE ERROR PERCENTAGE, ALPER
C
      ALPER = 0.01D0
   10 I = I + 1
C
C     CHECK THAT SOLUTION IS NOT EQUAL TO ZERO OR LESS THAN
C     10E-6 IN ABSOLUTE VALUE.  IF THE ABOVE IS TRUE THEN THE
C     ERROR CALCULATION IS IGNORED.
C
      IF (DABS(SOLTN(I)) .LE. 0.000001D0) GO TO 20
      DIFF = PSOLTN(I) - SOLTN(I)
      PER = DABS(DIFF) / DABS(SOLTN(I))
      IF (PER .GT. ALPER) GO TO 30
   20 CONTINUE
      IF (I .EQ. NMAT) GO TO 40
      GO TO 10
   30 IERR = 1
   40 CONTINUE
      RETURN
      END
C
      SUBROUTINE STRKX(II, JJ, ILAM, VALUE, AKX, VALUE3)
C
C     SUBROUTINE TO GENERATE THE VECTOR {AKX}
C
C     II   =LOCATION OF INTEGRATION POINTS ALONG Z DIRECTION
C     JJ   =LOCATION OF INTEGRATION POINTS ALONG X DIRECTION (GIVEN)
C     ILAM =LAMINA NUMBER (GIVEN)
C     VALUE=GAUSSIAN INTEGRATION POINTS (GIVEN)
```

```
C      AKX  =LINEAR STRAIN VECTOR IN X DIRECTION (RETURNED)
C
       IMPLICIT REAL*8(A - H,O - Z)
       COMMON /CONST1/ D(2,2), G(20), TH(20), DELX, DELY, NDIMD
       COMMON /CONST2/ Z(20)
       COMMON /CONST3/ DERN1(48), EDERN3(48)
       COMMON /CONST4/ NL, NDIMB
       DIMENSION AKX(48), VALUE(5), DDERM(48), FDERN3(48), TEMP(1)
       DIMENSION VALUE3(3)
       CALL BSTRAM(JJ, VALUE, DELX, DDERM)
       CALL BSTRAF(II, VALUE3, ILAM, FDERN3)
       CALL BSTRAU(JJ, VALUE, DELX, DERN1)
C
C      LINEAR PORTION OF BENDING STRAIN OF THE
C      STIFFNESS MATRICES
C
C      DERN1 = VECTOR OF THE VALUE dU/dX
C      DDERM = VECTOR OF THE VALUE d2W/dX2
C      FDERN3= VECTOR OF THE X-DEPENDENT VALUE OF dU*/dX
C      EDERN3= VECTOR OF THE CONSTANT VALUE OF dU*/dX
C
       DO 10 K = 1, NDIMB
         TEMP(1) = (Z(ILAM) + 0.5DO*TH(ILAM)*VALUE3(II)) * DDERM(K) * 2.
      1  DO / DELX
   10 AKX(K) = 2.DO * (DERN1(K) - TEMP(1) + FDERN3(K) - EDERN3(K)) /
      1DELX
       RETURN
       END
C
       SUBROUTINE BSTRAM(JJ, VALUE, DELX, DDERM)
C
C      SUBROUTINE TO GENERATE THE 2ND DERIVATIVE
C      OF THE VECTOR {M}
C
C      JJ,VALUE,DELX =SEE PREVIOUS EXPLANATIONS (GIVEN)
C      DDERM         =VECTOR CONTAINING THE 2 ND DERIVATIVE
C                     OF THE FUNCTION RELATING THE TRANSVERSE
C                     DISPLACEMENT, W, AND THE NODAL DISPLACEMENTS
C                     {M} (RETURNED)
C
       IMPLICIT REAL*8(A - H,O - Z)
       COMMON /CONST4/ NL, NDIMB
       DIMENSION VALUE(5), DDERM(48)
       DO 10 L = 1, NDIMB
   10 DDERM(L) = 0.DO
       DDERM(1) = VALUE(JJ) * 1.5DO
       DDERM(2) = (0.75DO*VALUE(JJ) - 0.25DO) * DELX
       L1 = 5 + NL
       L2 = 6 + NL
       DDERM(L1) = -DDERM(1)
       DDERM(L2) = DDERM(2) + 0.5DO * DELX
       RETURN
       END
C
       SUBROUTINE BSTRAF(II, VALUE3, ILAM, FDERN3)
C
C      SUBROUTINE TO GENERATE THE DERIVATIVE OF THE
C      VECTOR {N3}XF
```

```
C
C       II,VALUE3,ILAM =SEE PREVIOUS EXPLANATIONS (GIVEN)
C       FDERN3          =VECTOR CONTAINING THE 1 ST DERIVATIVE OF THE
C                        FUNCTION RELATING THE X-DISPLACEMENT DUE TO
C                        SHEAR DISTORTION OF CROSS-SECTION AND THE
C                        NODAL DISPLACEMENT {N3} (GIVEN)
C
      IMPLICIT REAL*8(A - H,O - Z)
      COMMON /CONST1/ D(2,2), G(20), TH(20), DELX, DELY, NDIMD
      COMMON /CONST4/ NL, NDIMB
      DIMENSION FDERN3(48), VALUE3(3)
      DO 10 KK = 1, NDIMB
   10 FDERN3(KK) = 0.D0
      DO 50 K = 1, ILAM
        IF (K .NE. ILAM) GO TO 20
        F = 0.5D0 * TH(K) * (VALUE3(II)*0.5D0+0.25D0*VALUE3(II)**2 + 0.
     1  25D0)
        GO TO 40
   20   K1 = K + 1
        IF (K1 .NE. ILAM) GO TO 30
        F = 0.5D0 * TH(K) + 0.5D0 * G(K) * TH(K1) * (0.5D0*VALUE3(II) -
     1  0.25D0*VALUE3(II)**2 + 0.75D0) / G(K1)
        GO TO 40
   30   F = 0.5D0 * TH(K) + 0.5D0 * G(K) * TH(K1) / G(K1)
   40   K4 = K + 4
        NEXT = NL + K4 + 4
        FDERN3(K4) = -F * 0.5D0
        FDERN3(NEXT) = -FDERN3(K4)
   50 CONTINUE
   60 CONTINUE
      RETURN
      END
C
      SUBROUTINE STRKXZ(II, JJ, ILAM, VALUE3, AKXZ, VALUE)
C
C     SUBROUTINE TO GENERATE VECTOR {AKXZ}
C
C     LINEAR PORTION OF SHEAR STRAIN OF THE
C     STIFFNESS MATRICES
C
C     II,JJ,ILAM,VALUE3 =SEE PREVIOUS EXPLANATIONS (GIVEN)
C     AKXZ              =LINEAR SHEAR STRAIN VECTOR IN
C                        X-Z PLANE(RETURNED)
C
      IMPLICIT REAL*8(A - H,O - Z)
      COMMON /CONST1/ D(2,2), G(20), TH(20), DELX, DELY, NDIMD
      COMMON /CONST4/ NL, NDIMB
      DIMENSION AKXZ(48), S1N3(48), S2N3(48), VALUE3(3), VALUE(5)
      CALL SSTRA(ILAM, VALUE, G, JJ, S1N3)
      IF (ILAM .GT. 1) GO TO 20
      DO 10 M = 1, NDIMB
   10 S2N3(M) = 0.D0
      GO TO 30
   20 CONTINUE
      LOWER = ILAM - 1
      CALL SSTRA(LOWER, VALUE, G, JJ, S2N3)
   30 CONTINUE
      DO 40 MM = 1, NDIMB
```

```
   40 AKXZ(MM) = ((1 + VALUE3(II))*S1N3(MM) + (1 - VALUE3(II))*S2N3(MM))
     1 / (2*G(ILAM))
      RETURN
      END
C
      SUBROUTINE SSTRA(NLAM, VALUE, G, JJ, SN3)
C
C     SUBROUTINE OF STRKXZ
C     GENERATE THE VECTOR {N3}XG(NLAM)
C
C     NLAM    =LAMINA NUMBER (GIVEN)
C     JJ,VALUE=SEE PREVIOUS EXPLANATIONS (GIVEN)
C     G       =SHEAR MODULUS (GIVEN)
C     SN3     =VECTOR OF {N3} TIMES SHEAR MODULUS (RETURNED)
C
      IMPLICIT REAL*8(A - H,O - Z)
      COMMON /CONST4/ NL, NDIMB
      DIMENSION SN3(48), VALUE(5), G(20)
      NL1 = NLAM + 4
      NL2 = NL1 + NL + 4
      DO 10 K = 1, NDIMB
   10 SN3(K) = 0.D0
      SN3(NL1) = (1.D0-VALUE(JJ)) * G(NLAM) * 0.5D0
      SN3(NL2) = (1.D0+VALUE(JJ)) * G(NLAM) * 0.5D0
      RETURN
      END
      SUBROUTINE NSTIFF(ILAM, ESOLTN, EL, TN)
C
C     SUBROUTINE TO CALCULATE THE LINEAR COMPONENT OF
C     THE STIFFNESS MATRIX, [BO]T[D][BO], USING A
C     5 POINTS GAUSSIAN INTEGRATION FOR AN ELEMENT
C
C     ILAM  =LAMINA NUMBER (GIVEN)
C     ESOLTN=PREVIOUS SOLUTION (GIVEN)
C     TN    =NONLINEAR ELEMENT STIFFNESS MATRIX (RETURNED)
C     AKX,AKXZ,BO,B1,B2=TEMPORARY STORAGE(GENERATED)
C     TEMP2,TEMP3,TEMP4=TEMPORARY STORAGE(GENERATED)
C
      IMPLICIT REAL*8(A - H,O - Z)
      COMMON /CONST1/ D(2,2), G(20), TH(20), DELX, DELY, NDIMD
      COMMON /CONST2/ Z(20)
      COMMON /CONST4/ NL, NDIMB
      DIMENSION TN(1176), TE1(2,48), TE2(2,48), WEIGHT(5), ESOLTN(48)
      DIMENSION VALUE(5), AKX(48), AKXZ(48), AMX(48,48), B2R(48)
      DIMENSION TEMP1(48,48), TEMP2(48,48), EL(48), SN(48,48)
      DIMENSION BO(2,48), B1(2,48), B2(2,48), BO1(2,48), BO2(2,48)
      DIMENSION STRAIN(2), STRESS(2), WEIGH3(3), VALUE3(3)
      DATA VALUE /-0.9061798459, -0.5384693101, 0.000000000000000,
     1     0.5384693101, 0.9061798459/
      DATA WEIGHT /0.2369268850, 0.4786286705, 0.5688888889,
     1     0.4786286705, 0.2369268850/
      DATA VALUE3 /-0.7745966692, 0.000000000, 0.7745966692/
      DATA WEIGH3 /0.5555555555, 0.888888889, 0.5555555555/
      NT = NDIMB * (NDIMB + 1) / 2
C
C     TN     = ELEMENT TANGENTIAL STIFFNESS MATRIX
C              (SYMMETRIC)
C     SN     = ELEMENT STRUCTURAL STIFFNESS MATRIX
```

```
C              (NONSYMMETRIC)
C
      DO 10 I3 = 1, NT
   10 TN(I3) = 0.D0
      DO 20 I1 = 1, NDIMB
        DO 20 I2 = 1, NDIMB
   20 SN(I1,I2) = 0.D0
      DO 80 I = 1, 3
        DO 80 J = 1, 5
C
C     I IS FOR PHI(IN Z) VALUES, AND J IS FOR PSI(IN X) VALUES
C
          CALL STRKX(I, J, ILAM, VALUE, AKX, VALUE3)
          CALL STRKXZ(I, J, ILAM, VALUE3, AKXZ, VALUE)
          CALL NBSTMX(J, VALUE, DELX, AMX)
C
C     FORM THE [B]'S MATRICES FROM THE VECTORS AKX, AKXZ, AMX,
C     AND THE SOLUTION VECTOR
C
          DO 40 JJ = 1, NDIMB
            TEM1 = 0.D0
            DO 30 II = 1, NDIMB
              TEM1 = ESOLTN(II) * AMX(II,JJ) + TEM1
   30       CONTINUE
            B2R(JJ) = TEM1
   40     CONTINUE
          DO 50 L = 1, NDIMB
            B1(1,L) = 0.5D0 * B2R(L)
            B1(2,L) = 0.D0
            B2(1,L) = B2R(L)
            B2(2,L) = 0.D0
            B0(1,L) = AKX(L)
            B0(2,L) = AKXZ(L)
            B01(1,L) = B0(1,L) + B1(1,L)
            B01(2,L) = B0(2,L) + B1(2,L)
            B02(1,L) = B0(1,L) + B2(1,L)
   50     B02(2,L) = B0(2,L) + B2(2,L)
C
C     FORM THE ELEMENT'S TANGENTIAL STIFFNESS MATRIX, TN
C
          CALL DGMULT(D, B02, TE1, NDIMD, NDIMD, NDIMB, NDIMD, NDIMD,
     1          NDIMD)
          CALL DGPROD(B02, TE1, TEMP1, NDIMB, NDIMD, NDIMB, 2, 2, 48, 1,
     1          0, 1)
          CALL DGMATV(B01, ESOLTN, STRAIN, NDIMD, NDIMB, NDIMD)
          CALL DGMATV(D, STRAIN, STRESS, NDIMD, NDIMD, NDIMD)
          DO 60 M = 1, NDIMB
            DO 60 N = 1, M
              MN = M * (M - 1) / 2 + N
   60     TN(MN) = WEIGH3(I) * WEIGHT(J) * DELX * 0.25D0 * TH(ILAM) *
     1    DELY * (TEMP1(M,N) + STRESS(1)*AMX(M,N)) + TN(MN)
C
C     FORM THE ELEMENT'S STRUCTURE STIFFNESS MATRIX, SN
C
          CALL DGMULT(D, B01, TE2, NDIMD, NDIMD, NDIMB, NDIMD, NDIMD,
     1          NDIMD)
          CALL DGPROD(B02, TE2, TEMP2, NDIMB, NDIMD, NDIMB, 2, 2, 48, 1,
     1          0, 2)
```

```
          DO 70 M = 1, NDIMB
            DO 70 N = 1, NDIMB
   70     SN(M,N) = WEIGH3(I) * WEIGHT(J) * DELX * 0.25DO * TH(ILAM) *
     1    DELY * TEMP2(M,N) + SN(M,N)
   80 CONTINUE
C
C     FORM THE LOAD CURVE VALUE VECTOR, EL WHERE EL IS THE
C     VECTOR OF LOAD VALUES ON THE CURVE WITH THE PREVIOUS
C     SOLUTION SUBSTITUTED INTO THE SYSTEM OF EQUATIONS.
C
      DO 100 II = 1, NDIMB
        TEM = 0.DO
        DO 90 JJ = 1, NDIMB
   90   TEM = SN(II,JJ) * ESOLTN(JJ) + TEM
        EL(II) = TEM
  100 CONTINUE
      RETURN
      END
C
      SUBROUTINE NBSTMX(JJ, VALUE, DELX, AMX)
C
C     SUBROUTINE TO GENERATE THE NON-LINEAR PART
C     OF THE STIFFNESS MATRIX. [MX]
C
C     JJ,VALUE,DELX =SEE PREVIOUS EXPLANATIONS (GIVEN)
C     AMX           =NONLINEAR STRAIN MATRIX OF THE 1 ST DERIVATIVE
C                    OF {M} TIMES ITS TRANSPOSE (RETURNED)
C
      IMPLICIT REAL*8(A - H,O - Z)
      COMMON /CONST4/ NL, NDIMB
      DIMENSION VALUE(5), AMX(48,48), DM(48), VALUE3(3)
      DO 10 K = 1, NDIMB
   10 DM(K) = 0.DO
      DM(1) = -1.5DO * (VALUE(JJ) + 1.DO) + 0.75DO * (VALUE(JJ) + 1.DO)
     1** 2
      DM(2) = (0.5DO-(VALUE(JJ) + 1.DO) + 0.375DO*(VALUE(JJ) + 1.DO)**2)
     1 * DELX
      K1 = NL + 5
      K2 = NL + 6
      DM(K1) = -DM(1)
      DM(K2) = (-0.5DO*(VALUE(JJ) + 1.DO) + 0.375DO*(VALUE(JJ) + 1.DO)**
     12) * DELX
      DO 20 I = 1, NDIMB
        DO 20 J = 1, NDIMB
   20 AMX(I,J) = DM(I) * DM(J) * 4.DO / (DELX**2)
      RETURN
      END
C
      SUBROUTINE DECOMP(N, LHB, A)
C
C     SUBROUTINE TO DECOMPOSE A MATRIX IN A SYSTEM OF EQUATIONS
C     USING CHOLESKY METHOD FOR BANDED, SYMMETRIC, POSITIVE
C     DEFINITE MATRIX TO SOLVE THE SYSTEM.
C
      IMPLICIT REAL*8(A - H,O - Z)
      DIMENSION A(12672)
C
C     A IS STORED COLUMNWISE
```

```
C
      KB = LHB - 1
C
C     DECOMPOSITION
C
      A(1) = DSQRT(A(1))
      IF (N .EQ. 1) RETURN
      DO 10 I = 2, LHB
   10 A(I) = A(I) / A(1)
      DO 60 J = 2, N
        J1 = J - 1
        IJD = LHB * J - KB
        SUM = A(IJD)
        KO = 1
        IF (J .GT. LHB) KO = J - KB
        DO 20 K = KO, J1
          JK = KB * K + J - KB
   20   SUM = SUM - A(JK) * A(JK)
        A(IJD) = DSQRT(SUM)
        DO 50 I = 1, KB
          II = J + I
          KO = 1
          IF (II .GT. LHB) KO = II - KB
          SUM = A(IJD + I)
          IF (I .EQ. KB) GO TO 40
          DO 30 K = KO, J1
            JK = KB * K + J - KB
            IK = KB * K + II - KB
   30     SUM = SUM - A(IK) * A(JK)
   40     A(IJD + I) = SUM / A(IJD)
   50   CONTINUE
   60 CONTINUE
      RETURN
      END
C
      SUBROUTINE SOLV(N, LHB, A, B)
C
C     SUBROUTINE TO SOLVE THE SYSTEM OF EQUATIONS USING CHOLESKY
C     METHOD AFTER THE MATRIX HAS BEEN DECOMPOSED BY DECOMP
C     SUBROUTINE CAN SOLVE FOR DIFFERENT VALUES OF B WITHOUT
C     DECOMPOSING THE MATRIX A REPEATEDLY.
C
      IMPLICIT REAL*8(A - H,O - Z)
      DIMENSION A(12672), B(264)
C
C     FORWARD SUBSTITUTION
C
      KB = LHB - 1
      B(1) = B(1) / A(1)
      IF (N .EQ. 1) GO TO 30
      DO 20 I = 2, N
        I1 = I - 1
        KO = 1
        IF (I .GT. LHB) KO = I - KB
        SUM = B(I)
        II = LHB * I - KB
        DO 10 K = KO, I1
          IK = KB * K + I - KB
```

```
10    SUM = SUM - A(IK) * B(K)
      B(I) = SUM / A(II)
   20 CONTINUE
C
C     BACKWARD SUBSTITUTION
C
   30 N1 = N - 1
      LB = LHB * N - KB
      B(N) = B(N) / A(LB)
      IF (N .EQ. 1) RETURN
      DO 50 I = 1, N1
        I1 = N - I + 1
        NI = N - I
        KO = N
        IF (I .GT. KB) KO = NI + KB
        SUM = B(NI)
        II = LHB * NI - KB
        DO 40 K = I1, KO
          IK = KB * NI + K - KB
   40   SUM = SUM - A(IK) * B(K)
        B(NI) = SUM / A(II)
   50 CONTINUE
      RETURN
      END
C
      SUBROUTINE BSTRAU(JJ, VALUE, DELX, DERN1)
C
C     SUBROUTINE TO GENERATE THE 1ST DERIVATIVE OF THE VECTOR {U}
C
C     JJ,VALUE,DELX =SEE PREVIOUS EXPLANATION
C     DERN1         =VECTOR CONTAINING THE 1ST DERIVATIVE OF
C                    THE AXIAL DISPLACEMENT, U, AND THE NODAL
C                    DISPLACEMENT VECTOR. (RETURNED)
C
      IMPLICIT REAL*8(A - H,O - Z)
      COMMON /CONST4/ NL, NDIMB
      DIMENSION VALUE(5), DERN1(48)
      DO 10 L = 1, NDIMB
   10 DERN1(L) = 0.DO
      DERN1(3) = -1.5DO * (VALUE(JJ) + 1.DO) + 0.75DO * (VALUE(JJ) + 1.
     1DO) ** 2
      DERN1(4) = (0.5DO-(VALUE(JJ) + 1.DO) + 0.375DO*(VALUE(JJ) + 1.DO)*
     1*2) * DELX
      L1 = NL + 7
      L2 = NL + 8
      DERN1(L1) = -DERN1(3)
      DERN1(L2) = (-0.5DO*(VALUE(JJ) + 1.DO) + 0.375DO*(VALUE(JJ) + 1.
     1DO)**2) * DELX
      RETURN
      END
C
      SUBROUTINE FSTRES(ILAM, ESOLTN, IEL, NE, IFAIL)
C
C     SUBROUTINE TO CALCULATE THE STRESS OF AN ELEMENT
C     USING A 5 POINTS GAUSSIAN INTEGRATION  IN X
C     AND A 3 POINTS INTEGRATION IN Z
C
C     ILAM  =LAMINA NUMBER (GIVEN)
```

```
C     ESOLTN=PREVIOUS SOLUTION (GIVEN)
C     AKX,AKXZ,BO,B1,B2=TEMPORARY STORAGE(GENERATED)
C     TEMP2,TEMP3,TEMP4=TEMPORARY STORAGE(GENERATED)
C
      IMPLICIT REAL*8(A - H,O - Z)
      COMMON /CONST1/ D(2,2), G(20), TH(20), DELX, DELY, NDIMD
      COMMON /CONST2/ Z(20)
      COMMON /CONST4/ NL, NDIMB
      COMMON /CONST6/ ULTCOM, ULTEN
      DIMENSION ESOLTN(48)
      DIMENSION VALUE(5), AKX(48), AKXZ(48), AMX(48,48), B2R(48)
      DIMENSION BO(2,48), B1(2,48), BO1(2,48)
      DIMENSION STRAIN(2), STRESS(2), BENDS(5), SHEARS(5), VALUE3(3)
      DATA VALUE /-0.9061798459, -0.5384693101, 0.000000000000000,
     1     0.5384693101, 0.9061798459/
      DATA VALUE3 /-0.7745966692, 0.0000000000, 0.7745966692/
      IF (IEL .NE. 1) GO TO 10
      TENMAX = O.DO
      COMMAX = O.DO
   10 WRITE (2,20) ILAM
      WRITE (1,20) ILAM
   20 FORMAT (/, ' LAMINA # ', I2)
      DO 140 I = 1, 3
        DO 90 J = 1, 5
C
C     I IS FOR PHI(IN Z) VALUES, AND J IS FOR PSI(IN X) VALUES
C
          CALL STRKX(I, J, ILAM, VALUE, AKX, VALUE3)
          CALL STRKXZ(I, J, ILAM, VALUE3, AKXZ, VALUE)
          CALL NBSTMX(J, VALUE, DELX, AMX)
          DO 40 JJ = 1, NDIMB
            TEM1 = O.DO
            DO 30 II = 1, NDIMB
              TEM1 = ESOLTN(II) * AMX(II,JJ) + TEM1
   30       CONTINUE
            B2R(JJ) = TEM1
   40     CONTINUE
          DO 50 L = 1, NDIMB
            B1(1,L) = 0.5DO * B2R(L)
            B1(2,L) = O.DO
            BO(1,L) = AKX(L)
            BO(2,L) = AKXZ(L)
            BO1(1,L) = BO(1,L) + B1(1,L)
   50     BO1(2,L) = BO(2,L) + B1(2,L)
          CALL DGMATV(BO1, ESOLTN, STRAIN, NDIMD, NDIMB, NDIMD)
          CALL DGMATV(D, STRAIN, STRESS, NDIMD, NDIMD, NDIMD)
C
C     CALCULATE THE BENDING AND SHEAR STRESSES AT THE PARTICULAR
C     INTEGRATION POINT
C
          BENDS(J) = STRESS(1)
          SHEARS(J) = STRESS(2)
C
C     FIND THE MAXIMUM COMPRESSIVE AND TENSILE STRESSES FOR
C     THE BEAM AND RECORD THEIR LOCATIONS
C
          IF (BENDS(J) .LE. O.DO) GO TO 70
          IF (BENDS(J) .LE. TENMAX) GO TO 60
```

```
          TENMAX = BENDS(J)
          MAXTEL = IEL
          MAXTLA = ILAM
          MAXTI = I
          MAXTJ = J
   60     IF (BENDS(J) .LT. ULTEN) GO TO 90
C
C      SET ANY TENSILE STRESS THAT EXCEEDS THE ALLOWABLE TENSILE
C      STRESS AS EQUAL TO 1.00E+30
C
          BENDS(J) = 1.00E+30
          IFAIL = 1
          GO TO 90
   70     IF (BENDS(J) .GE. COMMAX) GO TO 80
          COMMAX = BENDS(J)
          MAXCEL = IEL
          MAXCLA = ILAM
          MAXCI = I
          MAXCJ = J
   80     IF (DABS(BENDS(J)) .LT. ULTCOM) GO TO 90
C
C      SET ANY COMPRESSIVE STRESS THAT EXCEEDS THE ALLOWABLE
C      COMPRESSIVE STRESS AS EQUAL TO -1.00E+30
C
          BENDS(J) = -1.00E+30
          IFAIL = 1
   90   CONTINUE
C
C      PRINT OUT THE STRESS IF REQUESTED
C
        IF (NL .EQ. 1) GO TO 100
        IF (I .NE. 2) GO TO 120
  100   WRITE (6,110) ILAM, BENDS(1), SHEARS(1), BENDS(5), SHEARS(5)
  110   FORMAT (/, I3, 6X, 2('(',E9.3,',',E9.3,')',3X))
  120   WRITE (1,130) (BENDS(II),II=1,5)
  130   FORMAT (/, 2X, 5(G12.4,1X))
        WRITE (2,130) (SHEARS(II),II=1,5)
  140 CONTINUE
      IF ((IEL .NE. NE) .OR. (ILAM .NE. NL)) GO TO 220
      WRITE (6,150) MAXCEL
  150 FORMAT (/, ' MAXIMUM COMPRESSION OCCURS AT ELEMENT # ', I2)
      WRITE (6,160) MAXCLA
  160 FORMAT (1X, 'LAMINA # ', I2)
      WRITE (6,170) MAXCI, MAXCJ
  170 FORMAT (1X, 'INTEGRATION POINTS I AND J OF ', I2, 2X, 'AND  ', I2)
      WRITE (6,180) COMMAX
  180 FORMAT (1X, 'COMPRESSION = ', G12.4)
      WRITE (6,190) MAXTEL
  190 FORMAT (/, ' MAXIMUM TENSION OCCURS AT ELEMENT # ', I2)
      WRITE (6,160) MAXTLA
      WRITE (6,170) MAXTI, MAXTJ
      WRITE (6,200) TENMAX
  200 FORMAT (1X, 'TENSION = ', G12.4)
      IF (IFAIL .NE. 1) GO TO 220
      WRITE (6,210)
  210 FORMAT (/, '********* THE BEAM HAS FAILED. *********')
  220 CONTINUE
      RETURN
      END
```