DTIC
S ELECTE D
JAN 0 4 1995
B

# FINITE ELEMENT METHODS FOR NONLINEAR STATIC ANALYSIS OF SANDWICH PLATES

THESIS

Damin J Siler, Second Lieutenant, USAF

AFIT/GAE/ENY/94D-18

19941228 087

DISTRIBUTION STATEMENT B
Approved for public release
Distribution Unlimited

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
**AIR FORCE INSTITUTE OF TECHNOLOGY**

Wright-Patterson Air Force Base, Ohio

AFIT/GAE/ENY/94D-18

# FINITE ELEMENT METHODS FOR NONLINEAR STATIC ANALYSIS OF SANDWICH PLATES

THESIS

Damin J Siler, Second Lieutenant, USAF

AFIT/GAE/ENY/94D-18

DTIC QUALITY INSPECTED 2

Approved for public release; distribution unlimited

# REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

| 1. AGENCY USE ONLY (Leave blank) | 2. REPORT DATE | 3. REPORT TYPE AND DATES COVERED |
|---|---|---|
| | December 1994 | Master's Thesis |

**4. TITLE AND SUBTITLE**

FINITE ELEMENT METHODS FOR NONLINEAR STATIC ANALYSIS OF SANDWICH PLATES

**5. FUNDING NUMBERS**

**6. AUTHOR(S)**

Damin J Siler, 2nd Lieutenant, USAF

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**

Air Force Institute of Technology
WPAFB OH 45433-6583

**8. PERFORMING ORGANIZATION REPORT NUMBER**

AFIT/GAE/ENY/94D-18

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**

Mr. William Baron WL/FIBA
WPAFB OH 45433

**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

**11. SUPPLEMENTARY NOTES**

**12a. DISTRIBUTION/AVAILABILITY STATEMENT**

Approved for public release; distribution unlimited

**12b. DISTRIBUTION CODE**

**13. ABSTRACT** (*Maximum 200 words*)

A finite element method, developed for static analysis of composite plates, was enhanced to be used with sandwich plates. The theory considers geometric nonlinearity and transverse shear. Furthermore, a new postprocessor was written to check for initial failure using the maximum stress criteria. It also includes a procedure for evaluating transverse normal stresses by enforcing equilibrium. The code modifications for sandwiches were verified by comparing finite element solutions to closed-form sandwich theories. Both methods showed good correlation. In addition, comparisons between one type of composite plate and a similar sandwich plate found that the sandwich had better specific stiffness for various geometries. Finally, quasi-static models of low-velocity impacts on sandwich plates succeeded in predicting incipient damage due to core crushing in the impact region, but they were inconsistent in estimating the actual load levels for initial failure, based on experiments from other research.

**14. SUBJECT TERMS**

Finite Elements, Composite Plates, Sandwich Plates
Impact Loads, Structures

**15. NUMBER OF PAGES** 127

**16. PRICE CODE**

| 17. SECURITY CLASSIFICATION OF REPORT | 18. SECURITY CLASSIFICATION OF THIS PAGE | 19. SECURITY CLASSIFICATION OF ABSTRACT | 20. LIMITATION OF ABSTRACT |
|---|---|---|---|
| Unclassified | Unclassified | Unclassified | UL |

## Disclaimer

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.


Accession For

NTIS GRA&I ☑
DTIC TAB ☐
Unannounced ☐
Justification

By
Distribution/

Availability Codes

| Dist | Avail and/or Special |
| --- | --- |
| A-1 | |

AFIT/GAE/ENY/94D-18

# FINITE ELEMENT METHODS FOR NONLINEAR STATIC ANALYSIS OF SANDWICH PLATES

THESIS

Presented to the Faculty of the Graduate School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Aeronautical Engineering

Damin J Siler, B.S.

Second Lieutenant, USAF

December 1994

Approved for public release; distribution unlimited

## Preface

This thesis is part of an overall research project dealing with sandwich constructions and their usage in structural panels. The research is sponsored by the Flight Dynamics Directorate of Wright Laboratories. Mr. William Baron is the point-of-contact for the sponsor, and Dr. Anthony Palazotto, AFIT/ENY, is the principal investigator.

The successful completion of this thesis was not an entirely individual effort. I would like to thank my advisor Dr. Palazotto for his invaluable guidance, patience and confidence. I would also like to acknowledge Capt. Timberlyn Harrington whose thesis work was directly related to mine and provided essential data for both modeling and comparison purposes. Finally, I would like to thank my parents Dave and Diane, my sister Deawn and the rest of my family for their long-distance support.

# Table of Contents

# List of Figures

# List of Tables

# List of Symbols

| Symbol | Definition |
|---|---|
| A through F | Coefficients of two parabolic curves |
| $A_i$ | Intermediate calculations in Whitney's plate solution |
| a , b | Plate dimensions in the X and Y directions |
| $D_{ij}$ | Bending stiffness matrix from CLPT |
| $E_{ii}$ | Elastic modulus |
| $G_{ij}$ | Shear modulus |
| $H_i$ | Hermitian shape functions |
| h | Total plate thickness |
| $h_a$ | Thickness of adhesive layer |
| $h_c$ | Thickness of sandwich core |
| $h_f$ | Thickness of each sandwich face |
| $k$ | Thickness factor for transverse shear |
| L-T-Z | Principal material direction coordinates for orthotropic material |
| $L_{ij}$ | Element labels within the load zone |
| $N_i$ | Lagrangian shape functions |
| $P_p$ | Peak total applied force |
| $\overline{Q}_{ij}$ | Constitutive stress-strain relations |
| q | Transverse pressure |
| $q_0$ | Transverse pressure intensity |
| $q_i$ | Nodal displacement vector |
| $\overline{q}$ | Nondimensional load |
| R | Plate indentation radius |
| S | Plate width-to-thickness aspect ratio |
| u | Translation in the X-direction |
| v | Translation in the Y-direction |
| w | Translation in the Z-direction |
| $w_{,1}$ , $w_{,2}$ | Physical slope in the X-Z and Y-Z planes |
| $w_c$ | Plate center deflection |
| $\overline{w}$ | Nondimensional plate center deflection |
| $\overline{w}_\rho$ | Nondimensional plate center deflection including density |
| $X_1$-$X_2$-$X_3$ | Cartesian coordinate system |
| X-Y-Z | Cartesian coordinate system (alternate form) |
| $x^*$ , $y^*$ | Element local in-plane coordinates |
| $\Delta x$ , $\Delta y$ | Element dimensions in the X and Y-direction |
| $\varepsilon_{ij}$ | Strain components |
| $\nu_{ij}$ | Poisson ratio |
| $\theta$ | Ply orientation angle |

| | |
|---|---|
| $\rho_c$ | Density of core material |
| $\rho_f$ | Density of face material |
| $\rho_s$ | Average density of sandwich construction |
| $\sigma_{ij}$ | Stress components |
| $\sigma_p$ | Isotropic principal stresses |
| $\xi$ , $\eta$ | Natural coordinates |
| $\xi_K$ , $\eta_K$ | Natural coordinates of element nodes |
| $\psi_1$ , $\psi_2$ | Bending rotations in X-Z and Y-Z planes |

## Abstract

In this research, a finite element method, originally developed for analyzing the static behavior of composite flat plates, was enhanced so it could also be used with sandwich plates. The governing theory considers geometric nonlinearity and transverse shear effects. Furthermore, a new external postprocessor was written in order to check plate models for initial failure using the maximum stress criteria. It also includes a procedure for evaluating transverse normal stresses by enforcing equilibrium through the thickness.

The programming modifications to allow modeling of sandwich plates were verified by comparing finite element solutions to those from closed-form linear theories for sandwich plates. Results showed good correlation between the numerical and theoretical solutions. In addition, displacement results (using the same program) from previous research for particular composite plates were compared to sandwich plates of similar composition and equal size. The sandwiches were more flexible in absolute terms, but displayed higher stiffness-to-weight ratios than the composites. Finally, low-velocity impact tests were modeled quasi-statically with the finite element code, and the new postprocessor was employed to predict incipient plate damage. Locations and modes of failure were correctly determined, but the predicted load levels for initial failure were inconsistent with experimental results from other research.

# I. Introduction

The need for strong, lightweight materials in aerospace structural components has led to a renewed interest in sandwich plates and shells using modern composite materials. These hybrid constructions consist of two dense outer faces that are bonded to a lightweight core. The core usually has little in-plane and flexural stiffness, compared to the faces, but it can have significant transverse strength and acts as a spacer to enhance the bending resistance of the faces. The result is a thicker plate or shell with a higher stiffness-to-weight ratio than the facesheets alone.

Predicting the static response of laminated composite plates is complicated due to effects such as: property variation through-the-thickness, geometric and physical nonlinearity, transverse shear and multiple failure modes. The additional complexity of sandwich constructions further complicates the analysis. Closed-form methods are limited to linear solutions (with many simplifying assumptions) for specific geometries and boundary conditions. Furthermore, experimental testing can give good results for a particular plate, but it can be impractical, in terms of time and money, for analyzing the effects of a wide range of variables. On the other hand, numerical techniques like finite elements (FE) can be applied to plates of different shapes, sizes, compositions, loadings and supports with greater flexibility. The accuracy and practicality of FE methods are dependent on the governing theories, model complexity and a given computer's speed and precision. Therefore, the application of advanced theories and finely-detailed models is conditioned by the availability of affordable technology that can handle them in a timely manner.

This research employs a finite element method developed for laminated plates and cylindrical shells and enhances it for use with sandwich plates. The governing theory considers geometric nonlinearity (through the von Karman strain-displacement relations) and transverse shear effects. In addition, plate materials are assumed linearly elastic to exclude such effects as plasticity and failure within the solution. A full three-dimensional plate model is reduced to a two-dimensional analysis by describing all displacement variations in the thickness direction relative to those at the mid-plate surface. This assumption ignores transverse normal stresses, but it greatly reduces the solution's complexity.

The main objective of this research was to test the effectiveness of the given finite element method in analyzing sandwich plates. Three cases were considered. First, linear and nonlinear displacement results were obtained for the same sandwich plate models used by Pagano [18] and Whitney [21] in developing closed-form solutions. This provided comparisons to established theories in order to verify the FE algorithms. Second, displacement results (using the same FE code) from previous research by Owens [16,17] for laminated plates were compared to sandwiches of equivalent overall geometry-- in which the core was half the total thickness and the face plies were constructed from the same material as the laminated plates. Finally, an attempt was made to predict initial failure using postprocessed stress calculations and maximum stress failure criteria. Sandwich plates used in experimental work by Harrington [7] were modeled, so that the FE results could be compared to actual, incipient plate damage from low velocity impacts. In this research, a quasi-static approach was used to simulate the dynamic loading employed in the experiments.

Previous Work

The textbook by Palazotto and Dennis [19] gives a detailed history of the use of finite elements for analyzing flat plates and cylindrical shells. It also describes the past research and theories which led to the development of the FE code (called SHELL) used in this thesis. Dennis [4] wrote the original version of SHELL for the study of large displacements and rotations of shells. Owens [16] used it to analyze composite plates and made comparisons between linear, geometrically nonlinear, classical and nonclassical solutions. He showed, for a given loading, that membrane stiffness due to nonlinear in-plane strain terms becomes significant in thin plates. In addition, the transverse shear flexibility present in nonclassical theories is important for thick plates. Linear and nonlinear solutions become alike for thicker plates as do classical and nonclassical solutions for thinner plates.

Early work in the study of sandwich plates was conducted by Pagano [18]. He developed a linear, three-dimensional elasticity solution for rectangular plates with simply-supported edges. Pagano's results for both composite and sandwich plates further emphasized the limitations of classical laminated plate theory (CLPT) for thick plates due to its neglect of transverse shear. Later work by Whitney [21] yielded an alternative closed-form solution that resembles CLPT with additional contributions for transverse shear. Displacement results from both methods showed good agreement over a wide range of plate width-to-thickness ratios; although, for thin plates, Pagano's solution converged closer to CLPT than Whitney's did.

Both Pagano and Whitney calculated in-plane stresses from strain values and the constitutive relations, but they derived transverse shear stresses from the equilibrium equations in elasticity theory (see the textbook by Sadda [20]). This method provides better accuracy than computing all stresses through the constitutive relations. Engblom and Ochoa [5,6] also used this procedure in developing linear finite element formulations for laminated composite plates. By assuming linear stress distributions through the thickness of each ply, the integral-differential equations of equilibrium can be converted into matrix operations for easy implementation.

On the other hand, SHELL does not satisfy localized equilibrium (although such errors tend to cancel out on a global scale) since it relates all stresses to strains through the constitutive relations. It does this because its consideration of nonlinear strains and higher-order stress distributions prevents it from obtaining transverse stresses within the solution through a linear system of equations that enforce equilibrium. Furthermore, compatibility is satisfied at the nodes but not through the thickness, in general. SHELL could be modified to employ an iterative process in which the postprocessor calculates transverse stresses by enforcing equilibrium and then uses them to alter the strain and displacement solutions until both equilibrium and compatibility are satisfied. However, this would have to be nested within the existing iteration scheme for load or displacement incrementing in SHELL's nonlinear solution control, thus greatly reducing the code's speed. Furthermore, such a modification is beyond the scope of this thesis.

Other research related to sandwich plates has been oriented towards studying and predicting failure due to damage brought about by low velocity impacts. McQuillan et al. [14] were among the first to observe that static and dynamic loading had similar failure mechanisms. Experimental work by Kelkar et al. [10] employed this concept to simulate low-velocity impact damage using equivalent quasi-static loads. Low-velocity impacts are representative of physical damage caused by such actions as a dropping a blunt object onto the plate from a short height. In addition, quasi-static means that loads are applied slowly enough to ignore inertial effects, and the plate is assumed to remain in static equilibrium as damage progression alters its equilibrium state. Finite element models using ANSYS (a commercial software package) were developed by Dandy et al. [3] for comparison with Kelkar's experiments. Both had agreeable results for thick plates but not for thin plates. The ANSYS solver can only consider linear strains, and the large displacements present in the thin plates invalidate this simplification.

Nemes and Simmonds [15] analyzed the impact response of sandwich plates with a foam core. The experimental plates were square, but the finite element models were circular disks of equivalent size. This allowed a complicated three-dimensional model to be reduced to an axisymmetric radial plane-section of the disk. Therefore, each sandwich's cross-section through the thickness was modeled as a continuum instead of distinct layers. The FE results overpredicted the experimental deflection but produced agreeable transverse shear stresses. In addition, the low stiffness of the foam cores allowed significant relative displacement between the faces. This effect could be reduced by introducing a honeycomb core with greater transverse normal stiffness, but the voids present in honeycomb cells may invalidate treating it as a continuum.

Sandwich plate failure modes due to impacts and static indentations were investigated by Lagace and Williamson [12]. As predicted, both types of loadings had similar damage responses. They tested sandwiches with graphite-epoxy laminate faces and Nomex honeycomb cores, which are similar to those used in this thesis. Initial damage occurred under the applied load from core crushing or buckling near the top face. As a consequence, this face experienced local fiber, matrix and delamination failure because the core could no longer support it. Thicker faces reduced the extent of damage by stiffening the indentation responses. On the other hand, thicker cores increased the chances of buckling more than they enhanced the faces' bending stiffness.

In summary, the majority of research dealing with sandwich plates has been limited to experimentation and linearized solutions. This thesis goes one step further by employing a finite element method that includes geometric nonlinearities. The goal is to validate it as an accurate and practical tool for static displacement and initial failure analysis.

# II. Theoretical Considerations

## Sandwich Plate

Figure 2.1 contains the geometry and coordinate systems used for modeling sandwich plates. Both X-Y-Z ($X_1$-$X_2$-$X_3$) and L-T-Z represent orthogonal systems. The longitudinal and lateral directions correspond to the principal material directions of an orthotropic ply. A ply's orientation angle $\theta$ is the angle from X to L (or from Y to T). All plates used in this research were symmetrical about their midsurfaces (z=0).

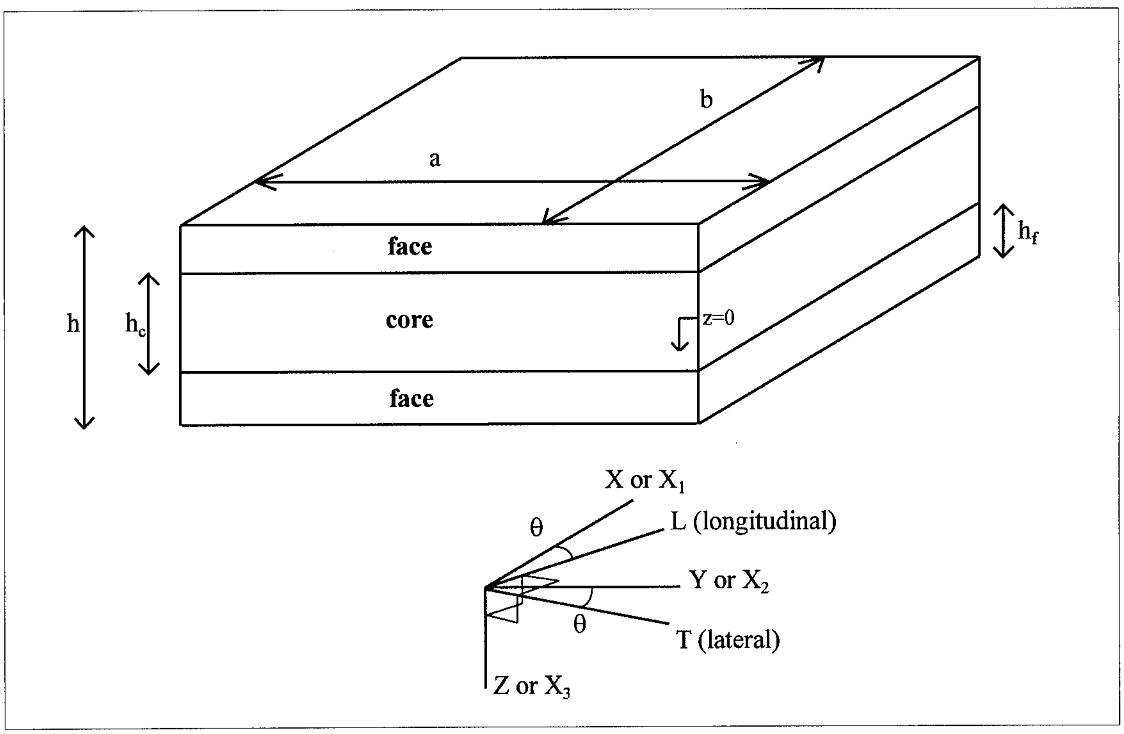


Figure 2.1: Sandwich Plate Geometry and Coordinate Systems

## Governing Equations

A plate is assumed to be in a state of plane stress. As a result, all transverse normal stresses $\sigma_{zz}$ are zero, and plate behavior can be described by displacements and rotations at and relative to the midsurface. Transverse normal strains $\varepsilon_{zz}$ are nonzero in general, but they are consequences (due to Poisson effects) of the other strains and do not affect the stress state. Transverse shear strains $\varepsilon_{xz}$ and $\varepsilon_{yz}$ are assumed to have parabolic distributions in the Z-direction. This can be done since the plane stress assumption decouples the in-plane and transverse shear constitutive relations. It also satisfies the boundary conditions of zero transverse shear on the top and bottom plate surfaces (none of the prescribed loading in this research imposed surface shears).

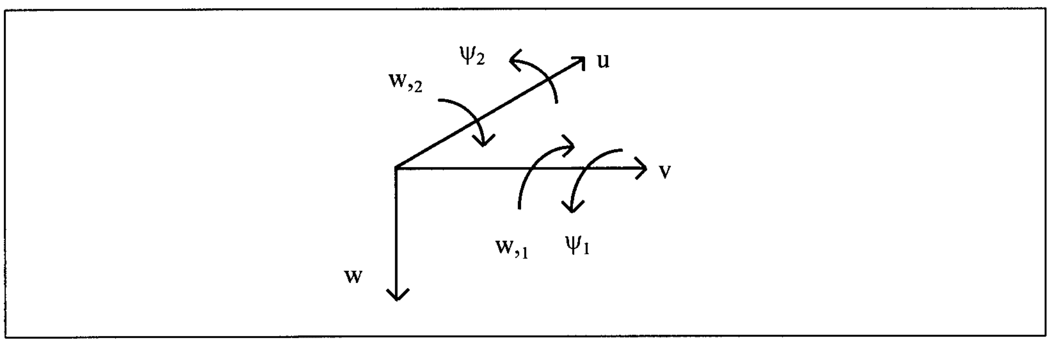


Figure 2.2: Plate Displacement Degrees-of-Freedom

Each point within the plate's midsurface has seven degrees-of-freedom as shown in Figure 2.2. Displacements u, v and w are translations in the X, Y and Z directions. The terms $w_{,1}$ and $w_{,2}$ are physical slopes of the midsurface in the X-Z and Y-Z planes, while $\psi_1$ and $\psi_2$ are rotations due to bending alone in those respective planes. Transverse shearing in a single plane is described by the algebraic sum of the two rotations.

Translational displacements away from the midsurface are evaluated through the following plate kinematics:

$$\begin{aligned} u_1(x,y,z) &= u + z\psi_1 + z^3 k(\psi_1 + w_{,1}) \\ u_2(x,y,z) &= v + z\psi_2 + z^3 k(\psi_2 + w_{,2}) \\ u_3(x,y,z) &= w \\ k &= -4/(3h^2) \end{aligned} \tag{2.1}$$

Furthermore, nonlinear strain and displacement are related through the von Karman plate equations [19] (linear plate solutions disregard all nonlinear terms):

$$\begin{aligned} \varepsilon_{xx} &= u_{1,1} + \tfrac{1}{2} w_{,1}{}^2 \\ \varepsilon_{yy} &= u_{2,2} + \tfrac{1}{2} w_{,2}{}^2 \\ \varepsilon_{xy} &= u_{1,2} + u_{2,1} + w_{,1} w_{,2} \\ \varepsilon_{yz} &= (1 + 3z^2 k)(w_{,2} + \psi_2) \\ \varepsilon_{xz} &= (1 + 3z^2 k)(w_{,1} + \psi_1) \end{aligned} \tag{2.2}$$

All ply materials are assumed linearly elastic and at least orthotropic. The assumption of plane stress allows the need for only six elastic constants: $E_{LL}$, $E_{TT}$, $G_{LT}$, $G_{TZ}$, $G_{LZ}$ and $\nu_{LT}$. The constitutive relations for stress and strain are:

$$\begin{aligned} \begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_6 \end{Bmatrix}^K &= \begin{bmatrix} \overline{Q}_{11} & \overline{Q}_{12} & \overline{Q}_{16} \\ \overline{Q}_{12} & \overline{Q}_{22} & \overline{Q}_{26} \\ \overline{Q}_{16} & \overline{Q}_{26} & \overline{Q}_{66} \end{bmatrix}^K \begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_6 \end{Bmatrix} \\ \begin{Bmatrix} \sigma_4 \\ \sigma_5 \end{Bmatrix}^K &= \begin{bmatrix} \overline{Q}_{44} & \overline{Q}_{45} \\ \overline{Q}_{45} & \overline{Q}_{55} \end{bmatrix}^K \begin{Bmatrix} \varepsilon_4 \\ \varepsilon_5 \end{Bmatrix} \end{aligned} \tag{2.3}$$

where K is the ply number and $\overline{Q}_{ij}^K$ are the components of that ply material's elastic stiffness matrix-- reduced for plane stress and transformed into the X and Y directions.

The numerical subscripts represent a simplified indexing of the stress and strain tensor components:

$$
\begin{aligned}
[\sigma] &= \begin{bmatrix} \sigma_1 & \sigma_6 & \sigma_5 \\ \sigma_6 & \sigma_2 & \sigma_4 \\ \sigma_5 & \sigma_4 & \sigma_3 \end{bmatrix} = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{xy} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{xz} & \sigma_{yz} & \sigma_{zz} \end{bmatrix} \\
[\varepsilon] &= \begin{bmatrix} \varepsilon_1 & \frac{1}{2}\varepsilon_6 & \frac{1}{2}\varepsilon_5 \\ \frac{1}{2}\varepsilon_6 & \varepsilon_2 & \frac{1}{2}\varepsilon_4 \\ \frac{1}{2}\varepsilon_5 & \frac{1}{2}\varepsilon_4 & \varepsilon_3 \end{bmatrix} = \begin{bmatrix} \varepsilon_{xx} & \varepsilon_{xy} & \varepsilon_{xz} \\ \varepsilon_{xy} & \varepsilon_{yy} & \varepsilon_{yz} \\ \varepsilon_{xz} & \varepsilon_{yz} & \varepsilon_{zz} \end{bmatrix}
\end{aligned}
\tag{2.4}
$$

Finite Element Solution

This research employed rectangular plate elements with four nodes and 28 degrees-of-freedom (seven per node as in Figure 2.2). The geometry of an individual element and the representation of its global, local and natural coordinates are shown in Figure 2.3. Displacements within the given element are interpolated from the nodal displacements through appropriate shape functions. The displacement field for w

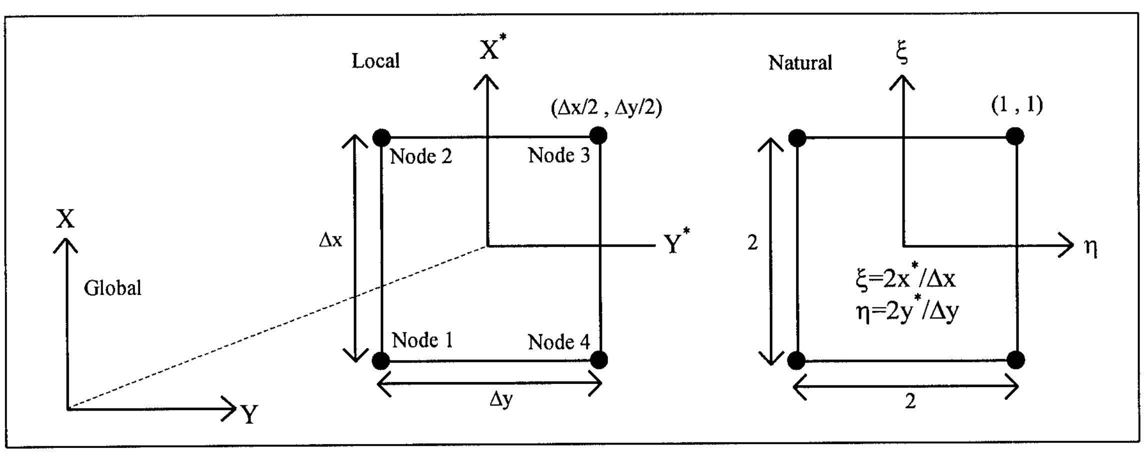


Figure 2.3: Four-Node Plate Element Geometry and Coordinate Systems

requires $C^1$ continuity (as defined in the textbook by Cook et al. [2]), therefore Hermitian shape functions are used for nodal displacements w , $w_{,1}$ and $w_{,2}$:

$$w(x,y) = \begin{bmatrix} H_1 & H_2 & H_3 & H_4 \end{bmatrix} \begin{Bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{Bmatrix}$$

$$H_K = \begin{Bmatrix} \frac{1}{8}(1+\xi_K\xi)(1+\eta_K\eta)(2+\xi_K\xi+\eta_K\eta-\xi^2-\eta^2) \\ \frac{1}{8}\Delta x\xi_K(1+\xi_K\xi)^2(\xi_K\xi-1)(1+\eta_K\eta) \\ \frac{1}{8}\Delta y\eta_K(1+\xi_K\xi)(\eta_K\eta-1)(1+\eta_K\eta)^2 \end{Bmatrix}^T \tag{2.5}$$

$$q_K = \begin{Bmatrix} w & w_{,1} & w_{,2} \end{Bmatrix}_K^T$$

where K=1 through 4 represent the local node numbers for an element found at global position (x,y). The other displacement fields only need $C^0$ continuity and employ Lagrangian shape functions:

$$\begin{Bmatrix} u(x,y) \\ v(x,y) \\ \psi_1(x,y) \\ \psi_2(x,y) \end{Bmatrix} = \begin{bmatrix} N_1 & 0 & 0 & 0 & \ldots & N_4 & 0 & 0 & 0 \\ 0 & N_1 & 0 & 0 & \ldots & 0 & N_4 & 0 & 0 \\ 0 & 0 & N_1 & 0 & \ldots & 0 & 0 & N_4 & 0 \\ 0 & 0 & 0 & N_1 & \ldots & 0 & 0 & 0 & N_4 \end{bmatrix} \begin{Bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{Bmatrix} \tag{2.6}$$

$$N_K = \tfrac{1}{4}(1+\xi_K\xi)(1+\eta_K\eta)$$
$$q_K = \begin{Bmatrix} u & v & \psi_1 & \psi_2 \end{Bmatrix}_K^T$$

The complex formulation of SHELL's finite element solution is fully described in the textbook by Palazotto and Dennis [19]. In a nonlinear model, the code allows either load or displacement incrementing for solution control. For each increment, it uses a Newton-Raphson iteration scheme to converge to a solution which minimizes potential energy. All plates modeled in this research used some form of distributed or multi-nodal loading. Therefore, load control was employed in each nonlinear case, since displacement control would require an assumed shape relation between the prescribed nonzero degrees-of-freedom. In this research, no iteration convergence problems arose using load-control.

Every case study in this research considered square plates with simply supported edges (u and v translations were free). Since all ply orientations were either 0 or 90 degrees, it was only necessary to generate FE meshes for a single quadrant of each plate by prescribing bi-axial symmetry. Figure 2.4 shows the displacement boundary conditions that were applied to each square quarter-plate.

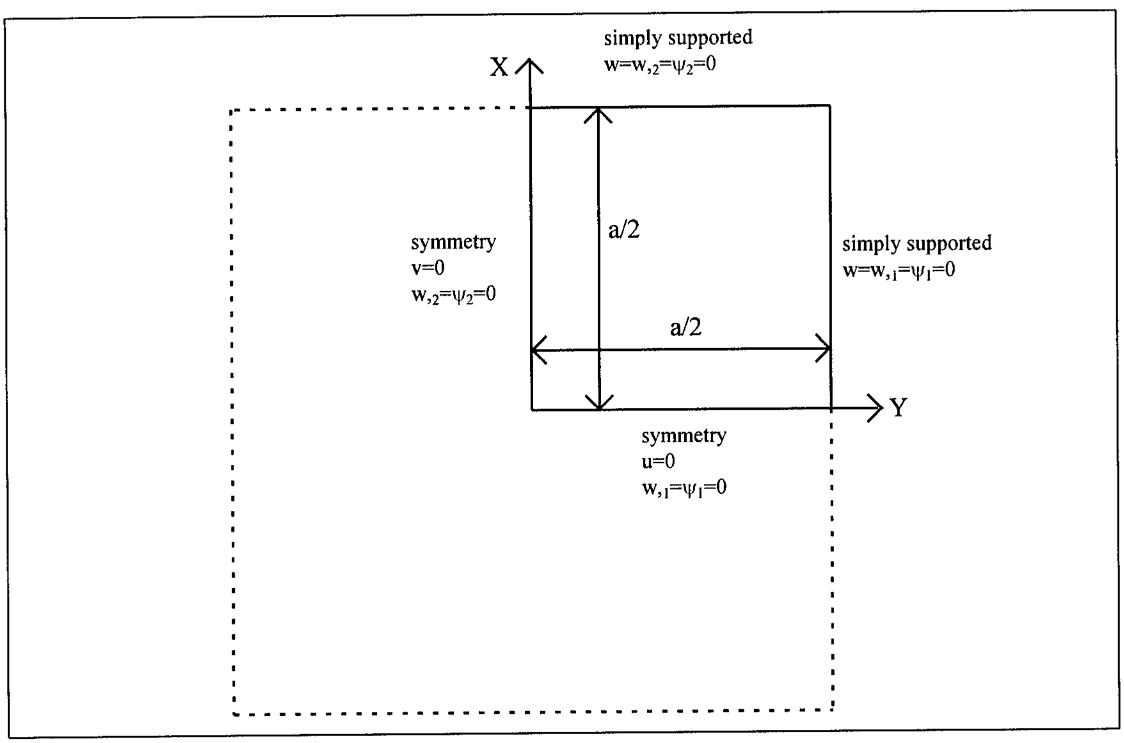


Figure 2.4: Boundary Conditions of Square Quarter Plate

Code Enhancements

SHELL required some modifications before being used with sandwich plates. The most critical change was to allow multiple sets of elastic properties. Without this, the faces and core could not be represented as different materials. Furthermore, the typically large variation between face and core thickness commanded the need to define a separate thickness for each ply. Otherwise, it would be necessary to divide all plies into a common uniform thickness, which could cause severe redundancies. Finally, SHELL now gives the user the opportunity to generate a secondary output file for use with a newly written (as part of this research) and separately executed postprocessor program, called FAILURE. Appendix A provides a more detailed explanation of these and other enhancements to SHELL and includes the new structure of its input deck.

FAILURE was written for the purpose of predicting the initial failure regions and modes of rectangular plates (with certain modeling restrictions) using the maximum stress criteria. SHELL's own postprocessor could have been altered for this task. In fact, FAILURE utilizes some of the same subroutines (with minor modifications). However, a separate program has several advantages. First, the same plate model can be rechecked for failure using different parameters and criteria values without having to re-execute SHELL to obtain the same displacement solution before postprocessing (a valuable feature for code debugging and validation of the methodology, since it has not been tried before with this FE theory). Second, the code's structure does not need to conform to that of SHELL. Therefore it gives the user more flexibility in adding or changing program features. On the other hand, the secondary output file generated by SHELL (which contains all preprocessor and solution data needed by FAILURE) is usually as large as its regular output file. Hence, using it with a set of complex FE models requires a computer with plentiful storage space. Appendix B includes the entire FAILURE code, the structure of its additional user-defined input deck and some of its other features.

Since postprocessor results are based on the assumption of perfect linearly elastic materials, FAILURE is invalid for predicting failure beyond the point of initiation (even with a more sophisticated failure theory than maximum stress). In addition, SHELL's basic design does not allow elastic property variation from element-to-element (a necessary feature for localizing the effects of failure within the solution). Hence, the implementation of progressive failure into the FE solution would require massive amounts of code alteration.

## Initial Failure Criteria

FAILURE is designed to report failure when averaged stresses within a given element exceed user-defined maximum magnitudes. For a laminated plate, element stresses are calculated at 12 discrete points per ply-- the four outermost Gauss points (see Figure 2.5) at the upper, middle and lower surfaces of each ply (the FE solution uses 5x5

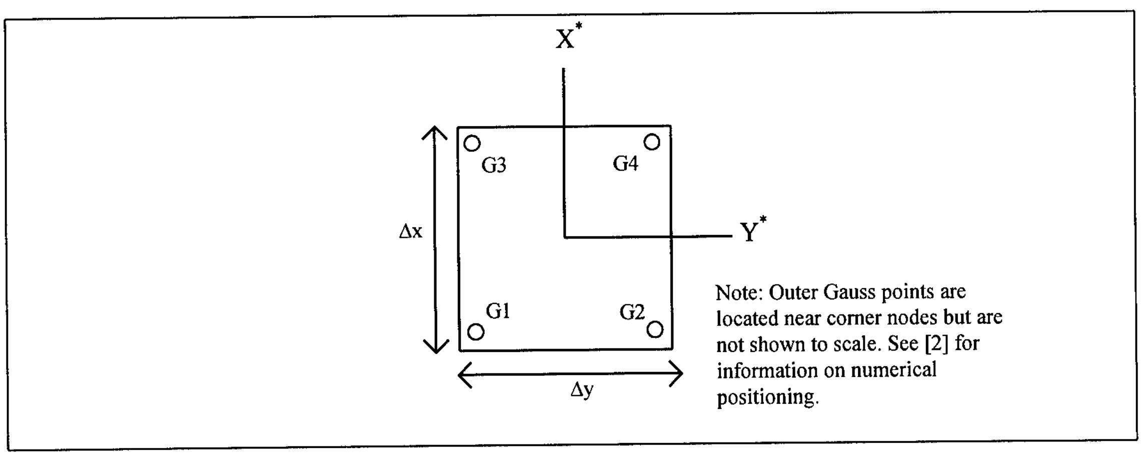


Figure 2.5: Single-Element Outer Gauss Points for Stress Calculation

Gauss quadrature for numerical integration [2,19]). Therefore, at a given Gauss point, each ply's stress distribution through the thickness is characterized by three values per component at known Z-coordinates. This provides sufficient data to obtain an average stress state by calculating parabolas that fit each component's discrete quantities and then finding a mean value for each continuous function (the area under the curve divided by the ply thickness). This is a better method because it is equivalent (for the assumed shape) to the arithmetic average of an infinite number stress values (per component) instead of just three. Figure 2.6 graphically demonstrates how ply stresses are averaged.

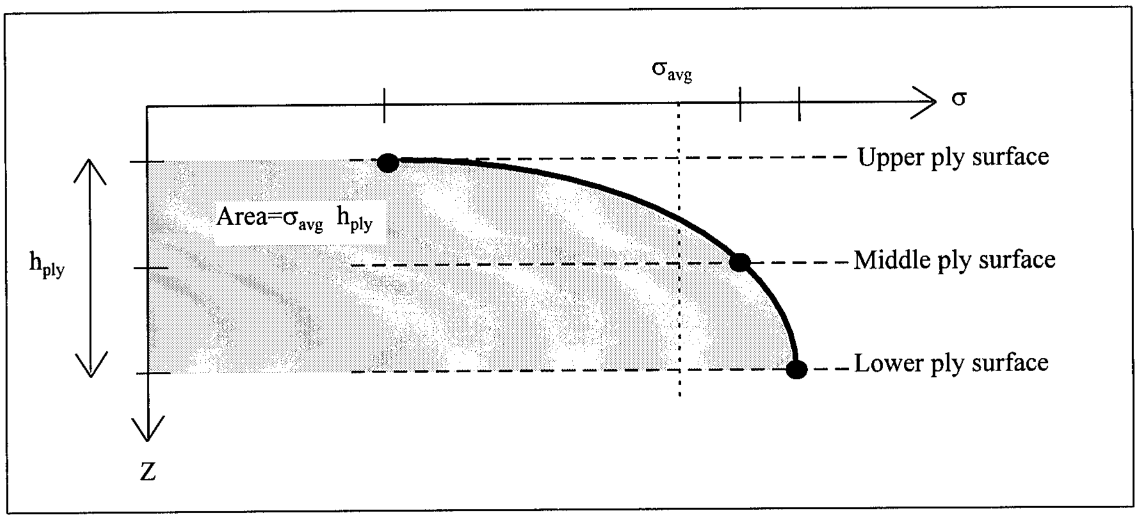


Figure 2.6: Ply Stress Averaging Through the Thickness

Average stress components are given in X-Y-Z coordinates, so they must be transformed to match the directions of the material failure criteria. For an orthotropic ply, the stresses are transformed into the L-T-Z system by the following matrix operation:

$$\begin{bmatrix} \sigma_{LL} & \sigma_{LT} & \sigma_{LZ} \\ & \sigma_{TT} & \sigma_{TZ} \\ \text{sym} & & 0 \end{bmatrix} = \begin{bmatrix} c & s & 0 \\ -s & c & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ & \sigma_{yy} & \sigma_{yz} \\ \text{sym} & & 0 \end{bmatrix} \begin{bmatrix} c & -s & 0 \\ s & c & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (2.7)$$

where c=cos θ and s=sin θ. On the other hand, stresses for an isotropic ply are converted to principal stresses by evaluating the eigenvalues of the stress tensor-- as shown in the textbook by Sadda [20]. The resulting characteristic polynomial is cubic and its roots are solved through a closed-form technique found in the mathematics handbook by Korn and Korn [11]. Seven possible modes of failure were given to orthotropic materials and three were given to isotropic materials. These modes and their maximum stress criteria are listed in Table 2.1.

Table 2.1: Material Failure Modes and Criteria

| Material | Mode | Criteria |
|---|---|---|
| Orthotropic (L-T-Z stresses) | Longitudinal Tension | $\sigma_{LL} \geq \sigma_{LL\,max}$ |
| | Longitudinal Compression | $\sigma_{LL} \leq \sigma_{LL\,min}$ |
| | Lateral Tension | $\sigma_{TT} \geq \sigma_{TT\,max}$ |
| | Lateral Compression | $\sigma_{TT} \leq \sigma_{TT\,min}$ |
| | Long.-Lat. Shear | $\lvert\sigma_{LT}\rvert \geq \sigma_{LT\,max}$ |
| | Long.-Z Shear | $\lvert\sigma_{LZ}\rvert \geq \sigma_{LZ\,max}$ |
| | Lat.-Z Shear | $\lvert\sigma_{TZ}\rvert \geq \sigma_{TZ\,max}$ |
| Isotropic (principal stresses) | Tension | $\sigma_{p\,max} \geq \sigma_{max\,uniaxial}$ |
| | Compression | $\sigma_{p\,min} \leq -\sigma_{max\,uniaxial}$ |
| | Shear | $\sigma_{p\,max} - \sigma_{p\,min} \geq \sigma_{max\,uniaxial}$ |

FAILURE also includes a procedure for checking delamination at the ply interfaces due to transverse shearing. Since the constitutive relations (Equation 2.3) cause stress discontinuities, in general, across the interface between unlike plies, the values of $\sigma_{xz}$ calculated on each side of the interface are usually different. This also holds true for $\sigma_{yz}$. The arithmetic mean stress at the interface for each component is used for checking delamination. FAILURE reports shear delamination for a given element and interface if (at any Gauss point) the magnitude of either mean value exceeds defined maximums.

Estimation of Transverse Normal Stresses

Although SHELL's theory assumes all transverse normal stresses are zero, FAILURE includes a routine which estimates $\sigma_{zz}$ by enforcing equilibrium. Neglecting body forces, the equation of equilibrium in the Z-direction is:

$$\sigma_{xz,x} + \sigma_{yz,y} + \sigma_{zz,z} = 0 \tag{2.8}$$

The shear stress gradients in Equation 2.8 are related to displacement gradients of $w$, $\psi_1$ and $\psi_2$ by taking partial derivatives of Equations 2.2 and 2.3:

$$\begin{Bmatrix} \sigma_{yz,y} \\ \sigma_{xz,x} \end{Bmatrix}^K = \begin{bmatrix} \overline{Q}_{44} & 0 & \overline{Q}_{45} & 0 \\ 0 & \overline{Q}_{45} & 0 & \overline{Q}_{55} \end{bmatrix}^K \begin{Bmatrix} \varepsilon_{yz,y} \\ \varepsilon_{yz,x} \\ \varepsilon_{xz,y} \\ \varepsilon_{xz,x} \end{Bmatrix} \tag{2.9}$$

for ply K

$$\begin{Bmatrix} \varepsilon_{yz,y} \\ \varepsilon_{yz,x} \\ \varepsilon_{xz,y} \\ \varepsilon_{xz,x} \end{Bmatrix} = \left(1 + 3z^2 k\right) \begin{Bmatrix} w_{,22} + \psi_{2,2} \\ w_{,21} + \psi_{2,1} \\ w_{,12} + \psi_{1,2} \\ w_{,11} + \psi_{1,1} \end{Bmatrix} \tag{2.10}$$

$$k = -4/(3h^2)$$

The displacement gradients in Equation 2.10 are related to nodal displacements through derivatives of the shape functions in Equations 2.5 and 2.6. Since these gradients are also used in other strain terms, SHELL's stress calculation subroutine (which is modified for use with FAILURE) already contains code for determining their values. Furthermore, the assumed parabolic distribution of transverse shear strains forces each ply's transverse shear stresses, and their in-plane gradients, to be parabolic functions of Z:

$$\begin{aligned}\sigma_{yz,y}(x,y,z) &= {}^{K}A(x,y)z^2 + {}^{K}B(x,y)z + {}^{K}C(x,y)\\ \sigma_{xz,x}(x,y,z) &= {}^{K}D(x,y)z^2 + {}^{K}E(x,y)z + {}^{K}F(x,y)\end{aligned} \tag{2.11}$$

where z is located within a given ply K. The parabola coefficients for a particular ply and Gauss point are obtained by curve fitting the discrete values of $\sigma_{yz,y}$ and $\sigma_{xz,x}$ at the ply's upper, middle and lower ply surfaces.

Equation 2.8 forces $\sigma_{zz,z} = 0$ at a plate's top and bottom surfaces, $z = \pm h/2$, because $\sigma_{yz}(x, y, \pm h/2) = \sigma_{xz}(x, y, \pm h/2) = 0$ from Equations 2.2 and 2.3. In other words, since the transverse shear stresses are zero everywhere on those surfaces, their in-plane gradients must also be zero on those surfaces. In addition, the finite element solution assumes that all prescribed transverse loading occurs on the top surface. Therefore, the bottom of the plate is free of transverse normal stresses: $\sigma_{zz}(x, y, h/2) = 0$. By combining Equations 2.8 and 2.11 and integrating in the Z-direction, the calculation of $\sigma_{zz}$ becomes:

$$\begin{aligned}\sigma_{zz}(x,y,z_n) &= \sum_{K=1}^{n} \int_{z_{K-1}}^{z_K} [({}^{K}A + {}^{K}D)z^2 + ({}^{K}B + {}^{K}E)z + ({}^{K}C + {}^{K}F)]\,dz\\ &= \sum_{K=1}^{n} \left[\tfrac{1}{3}({}^{K}A + {}^{K}D)z^3 + \tfrac{1}{2}({}^{K}B + {}^{K}E)z^2 + ({}^{K}C + {}^{K}F)z\right]\Big|_{z_{K-1}}^{z_K}\end{aligned} \tag{2.12}$$

where K=1 denotes the top ply and K=n is the ply located at $z = -z_n$. The bounds $z_{K-1}$ and $z_K$ are the Z-coordinates of ply K's upper and lower surfaces, except when K=n. In that case $z_K = -z_n$. Note that the use of $-z_n$ arises because the Z-axis is positive downward through the plate. Integrating in the positive Z-direction imposes an initial value of $\sigma_{zz}=0$

at the top of the plate (z= -h/2). Equation 2.12 includes sign changes that alter the constant of integration (and thus translates the function $\sigma_{zz}(z)$ for a fixed x and y) in order to enforce the boundary condition of $\sigma_{zz}$=0 at the bottom of the plate (z= h/2). Code testing verified that Equation 2.12 calculates negative (compression) values of $\sigma_{zz}$(x, y,-h/2) in plate regions subjected to a compressive transverse pressure on the top surface.

Plate symmetry, with respect to the midsurface, forces all transverse shear stress profiles in the Z-direction (and their in-plane gradients) to be symmetric about z=0. Hence, integrating Equation 2.8 generates $\sigma_{zz}$ profiles, for a fixed x and y, that are the superposition of an antisymmetric function of z and a constant (a line normal to z=0). As a consequence, the three $\sigma_{zz}$ boundary conditions on the top and bottom plate surfaces ($\sigma_{zz,z}(x,y,\pm h/2)=\sigma_{zz}(x,y,h/2)=0$) cause each profile to resemble a cubic polynomial with a maximum magnitude at the top surface of $\sigma_{zz}$(x,y,-h/2). However, for a laminate, each ply is usually associated with a different cubic polynomial because the coefficients in Equation 2.11 change from ply to ply. Therefore, the actual profiles generated from Equation 2.12 are continuous but piecewise smooth at the ply interfaces. Figure 2.7 displays a typical $\sigma_{zz}$ profile for a sandwich plate region subjected to a transverse pressure on the top surface.

Testing of the FAILURE code revealed a numerical problem associated with directly calculating $\sigma_{yz,y}$ and $\sigma_{xz,x}$ from nodal displacements. The use of Lagrangian shape functions-- with only $C^0$ continuity-- for u , v , $\psi_1$ and $\psi_2$ (Equation 2.6) cause in-plane discontinuities in all stress fields. The effects are usually less significant for $\sigma_{xx}$ ,

$\sigma_{yy}$ and $\sigma_{xy}$ because their corresponding strains have more w , $w_{,1}$ and $w_{,2}$ terms (Equations 2.1 and 2.2) which tend to minimize the variations across adjacent elements.

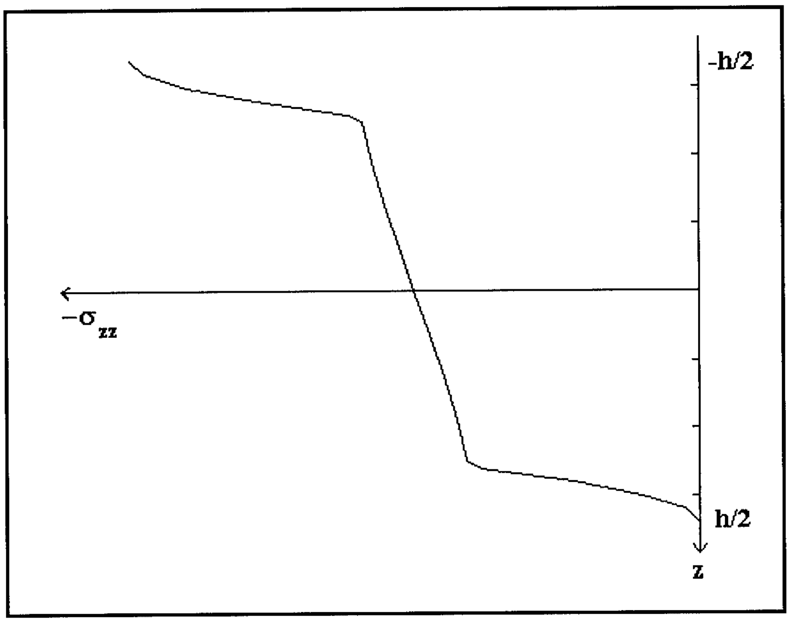


Figure 2.7: Shape of $\sigma_{zz}$ Distribution for Sandwich Plate under Transverse Pressure

However, transverse shear strains can fluctuate severely in the direction normal to the transverse plane-- the X-direction for $\varepsilon_{yz}$ and the Y-direction for $\varepsilon_{xz}$. Therefore, the fluctuations amplify when calculating $\sigma_{yz,y}$ and $\sigma_{xz,x}$ from Equation 2.9 because they require transverse strain gradients in both X and Y-directions. This resulted in highly varied and inaccurate estimations of $\sigma_{zz}$. Fortunately, a simple averaging of the $\sigma_{yz,y}$ and $\sigma_{xz,x}$ values obtained at an element's outer Gauss points greatly reduced these fluctuations. Using the Gauss point labeling as shown in Figure 2.5, the eight stress gradient values for a given element and ply surface (upper, middle or lower) are combined into four average values.

$$\begin{aligned}
{}^{1}\sigma^{av}_{xz,x} = {}^{3}\sigma^{av}_{xz,x} &= \tfrac{1}{2}\left({}^{1}\sigma_{xz,x} + {}^{3}\sigma_{xz,x}\right) \\
{}^{2}\sigma^{av}_{xz,x} = {}^{4}\sigma^{av}_{xz,x} &= \tfrac{1}{2}\left({}^{2}\sigma_{xz,x} + {}^{4}\sigma_{xz,x}\right) \\
{}^{1}\sigma^{av}_{yz,y} = {}^{2}\sigma^{av}_{yz,y} &= \tfrac{1}{2}\left({}^{1}\sigma_{yz,y} + {}^{2}\sigma_{yz,y}\right) \\
{}^{3}\sigma^{av}_{yz,y} = {}^{4}\sigma^{av}_{yz,y} &= \tfrac{1}{2}\left({}^{3}\sigma_{yz,y} + {}^{4}\sigma_{yz,y}\right)
\end{aligned} \tag{2.13}$$

When the calculated stress gradients were replaced by these averages, sample plates with a uniform transverse pressure obtained peak $\sigma_{zz}$ values (from Equation 2.12) that typically ranged between 65 and 85 percent of the applied pressure. These estimates were better than expected, since $\sigma_{zz}$ is related to the nodal displacement through third-hand calculations (strain-displacement equations, constitutive relations and equilibrium in the Z-direction).

One problem with Equation 2.12 is that it prevents satisfaction of a fourth boundary condition, $\sigma_{zz}$ (x, y, -h/2) = 0, when a region of the top surface is free of transverse loading. It will generally calculate a nonzero value because a cubic polynomial is fully constrained by four boundary conditions, and allowing $\sigma_{zz}$ (x, y, ±h/2) = $\sigma_{zz,z}$ (x, y, ±h/2) = 0 causes zero transverse normal stress throughout the thickness. The only way this can occur is if $\sigma_{yz,y}$ (x, y, z) = -$\sigma_{xz,x}$ (x, y, z) for a particular x and y.

However, the previously mentioned research by Engblom and Ochoa [5,6] may provide a means of masking the problem. They obtained $\sigma_{xz}$ and $\sigma_{yz}$ through the equations of equilibrium in the X and Y-directions and also utilized Equation 2.8 to calculate $\sigma_{zz}$. The resulting thickness profile of $\sigma_{zz}$ for a stress-free top surface resembled a sine wave. Hence, one possibility is to map the curve obtained from Equation 2.12 to a sine wave with an equivalent area under the curve, as shown in Figure 2.8. The area calculated from Figure 2.7 (and other curves based on Equation 2.12) is $(h/2)\bullet\sigma_{zz}(x, y, -h/2)$ because its antisymmetric shape has the same area as a triangle formed by the Z-axis, z= -h/2 and the line connecting $\sigma_{zz}(x, y, h/2)$ and $\sigma_{zz}(x, y, -h/2)$. Also note that Figure 2.8 fails to satisfy the boundary conditions of $\sigma_{zz,z} = 0$ at $z=\pm h/2$. A more complicated mapping function-- one that permits additional constraints-- could solve this problem, but for simplicity the regions of small stress gradients near the top and bottom plate surfaces are assumed negligible with respect to the entire thickness. FAILURE does not presently include any mapping technique since the structure of SHELL's secondary output file does not contain information on the location of nodal loads used to generate a displacement solution. Thus, FAILURE cannot tell where mapping should be used. The task of interpreting the meaning of the $\sigma_{zz}$ calculations is left to the user.

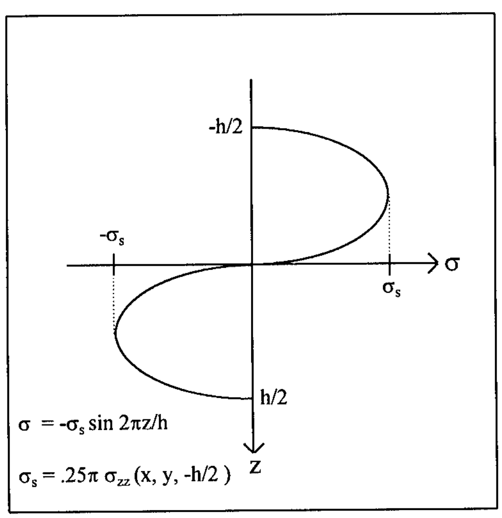


Figure 2.8: Sine Mapping Function

## III. Pagano/ Whitney Sandwich Plate Models

### Closed-Form Solutions

Both Pagano [18] and Whitney [21] obtained linear solutions for simply-supported square sandwich plates that are subjected to sinusoidal pressures. Each face sheet was a single [0°] ply of an unidentified composite material, and the core was some type of transversely isotropic material. The thickness of a single face was one-tenth that of the core ($h_f = h_c /10$ and $h = 2 h_f + h_c$). The face and core materials had the following relevant elastic properties (converted to SI units):

Face: $E_{LL}$=172.3 GPa $E_{TT}$=6.985 GPa

$G_{LT}$=$G_{LZ}$=3.447 GPa $G_{TZ}$=1.379 GPa (3.1)

$\nu_{LT}$=0.25

Core: $E_{XX}$=$E_{YY}$=275.8 MPa

$G_{XY}$=110.3 MPa $G_{XZ}$=$G_{YZ}$=413.7 MPa (3.2)

$\nu_{XY}$==0.25

Pagano [18] develops his elasticity solution for a generally loaded plate, but the results for these particular plates are given numerically at selected locations. However, he also includes a CLPT solution algorithm that can be conveniently applied to these cases. For the square quarter-plate notation shown in Figure 3.1, the pressure distribution q(x,y) on the top surface of the plate is:

$$q(x, y) = q_0 \cos(\pi x / \mathrm{a}) \cos(\pi y / \mathrm{a}) \tag{3.3}$$

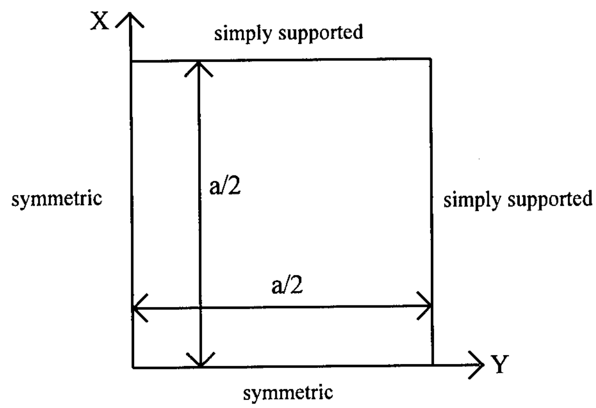

Figure 3.1: Boundary Conditions of Square Quarter Plate

where $q_0$ is the peak pressure at the plate's center (x=y=0). The transverse displacement field w(x,y) for a simply supported plate can be approximated with a similar sinusoidal distribution:

$$w(x,y) = w_c \cos(\pi x / \mathrm{a}) \cos(\pi y / \mathrm{a}) \tag{3.4}$$

where $w_c$ is the deflection at the plate's center. For a symmetric plate, Pagano's CLPT solution for $w_c$ becomes:

$$w_c = \frac{q_0}{(D_{11} + 2(D_{12} + 2D_{66}) + D_{22})(\pi / \mathrm{a})^4} \tag{3.5}$$

where $D_{11}$ , $D_{22}$ , $D_{12}$ and $D_{66}$ are components of the bending stiffness matrix and are integral functions of the constitutive relations in the Z-direction (as defined in the textbook by Agarwal and Broutman [1]).

Whitney's plate solution for the same load and displacement distributions as Equations 3.3 and 3.4 is the following expression [21]:

$$
\begin{aligned}
w_c &= \frac{q_0 a^2}{\pi^2 D}(A_1 A_4 - A_2^2) \\
D &= A_1 A_4 A_6 + 2A_2 A_3 A_5 - A_1 A_5^2 - A_4 A_3^2 - A_6 A_2^2 \\
A_1 &= D_{11} + D_{66} + G_{\text{xz core}} h a^2 / \pi^2 \\
A_2 &= D_{12} + D_{66} \\
A_3 &= G_{\text{xz core}} h a / \pi \\
A_4 &= D_{66} + D_{22} + G_{\text{yz core}} h a^2 / \pi^2 \\
A_5 &= G_{\text{yz core}} h a / \pi \\
A_6 &= (G_{\text{xz core}} + G_{\text{yz core}}) h
\end{aligned}
\tag{3.6}
$$

Finite Element Modeling

The enhanced version of SHELL was used to obtain linear and nonlinear displacement solutions for the aforementioned sandwich structure. Five plates with the same thickness and variable widths were considered in order to compare the static responses for thin and thick plates. A constant value of h=12.192 mm was assumed, and the aspect ratios (S=a/h) for the five plates were 10, 20, 30, 40 and 50. The meshes for each quarter-plate model consisted of N-by-N square elements of equal size ($\Delta x = \Delta y = a/2N$). The appropriate mesh resolution for each plate was obtained through convergence studies of meshes ranging from 12-by-12 to 24-by-24. Furthermore, the maximum pressure applied to each plate was $q_0$=6985 kPa.

SHELL cannot automatically consider the sinusoidal load distribution in Equation 3.3. Therefore it was necessary to calculate individual nodal loads. This was accomplished by integrating the product of Equation 3.3 and the Hermitian shape function $H_{K1}$ (associated with the nodal displacement w in Equation 2.5 and converted into global coordinates based on Figure 2.3) across each element's area for every local element node K. The nodal loads for an entire mesh were calculated by adding the contributions of each adjacent element for a given node. Note that this method ignores coupled loads obtained from the other shape functions in Equation 2.5, but it should be a good approximation of the distributed load for meshes with a relatively large number of elements. MATLAB™ [13] and Mathematica™ [22] (commercial mathematical software packages) were employed to perform the integrations and other computations. The accuracy of its nodal load calculations was verified by checking for load symmetry about each quarter-plate's plane of symmetry (x=y) and by comparing the sum of all nodal loads to the total force from a continuous distribution (integrating Equation 3.3 across the entire area of the quarter-plate yields a total load of $q_0a^2/\pi^2$).

Convergence Study

As shown in Table 3.1, 12x12 meshes converged well for S=10 and 20 while 20x20 meshes were satisfactory for S=30, 40 and 50. Deviations under 5% are considered good for the plate elements used by SHELL.

Table 3.1: Displacement Convergence for Plate Meshes (NL Solution)

| | $w_c$ [mm] for NxN mesh at $q_0$=6985 kPa | | | | |
|---|---|---|---|---|---|
| S | 12x12 | 16x16 | 20x20 | 24x24 | % Deviation |
| 10 | 2.5038 | 2.5058 | | | 0.080 |
| 20 | 20.853 | 20.933 | | | 0.384 |
| 30 | | | 61.534 | 61.783 | 0.405 |
| 40 | | | 111.86 | 112.82 | 0.858 |
| 50 | | | 167.29 | 169.53 | 1.34 |

## Results and Discussion

Figures 3.2 and 3.3 plot the nonlinear FE deflection of each plate at its center ($w_c$) due to varying peak load pressures. The two thickest plates, S=10 and 20, exhibit very linear behavior up to $q_0$=6985 kPa. On the other hand, geometric nonlinearities become evident in the curves for S=30, 40 and 50 when $q_0$ is greater than 500 kPa. At the highest load, the value of $w_c$ for the thinnest plate (S=50) is about 15 times greater than its thickness. By comparison, a linear FE solution (from Table 3.2) for S=50 would predict a maximum deflection that is 58 times greater than the plate thickness, which obviously violates the assumption of small displacements for linear behavior.

Table 3.2: Linear FE Results

| S | $q_0$ [kPa] | $w_c$ [mm] | slope= $w_c$ / $q_0$ [mm/kPa] | $w_c$ [mm] at $q_0$=6985 kPa |
|---|---|---|---|---|
| 10 | 698.5 | 0.251 | 3.593e-4 | 2.510 |
| 20 | 698.5 | 2.313 | 3.311e-3 | 23.13 |
| 30 | 698.5 | 10.04 | 1.437e-2 | 100.4 |
| 40 | 698.5 | 29.84 | 4.272e-2 | 298.4 |
| 50 | 698.5 | 70.70 | 1.012e-1 | 707.0 |

In order to obtain deflection results from CLPT and Whitney's solution (Equations 3.5 and 3.6), the bending stiffness matrix must be determined for each sandwich plate. SHELL's preprocessor calculates these numbers and can be commanded to display them in the output file. For these plates, the matrix does not change since face and core thickness are held constant. The relevant values (in N-m=$10^6$ kPa-mm$^3$) are: $D_{11}$=13215.7 , $D_{22}$=551.238 , $D_{12}$=137.810 and $D_{66}$=272.014. In addition, both Pagano and Whitney nondimensionalized their displacement results with the following expression:

$$\overline{w} = \frac{100 w_c E_{TT\,face}}{q_0 S^4 h} \tag{3.7}$$

Table 3.3 lists the nondimensional displacements from CLPT, Pagano's elasticity solution, Whitney's solution and both linear and nonlinear FE methods. Figures 3.4, 3.5 and 3.6 plot these values versus the corresponding aspect ratio. The linear FE solution produced deflections similar to both Pagano and Whitney (albeit slightly stiffer for thick plates). Furthermore, all linear cases converged to the CLPT solution as plates became thinner. The differences for thick plates can be attributed to variations in how each theory considers transverse shear effects (or neglects them in the case of CLPT). As bending effects become dominant for large aspect ratios, each linear theory produces nearly identical results. The divergence between linear and nonlinear FE results as S is increased is due to a coupling between bending and membrane stiffness that is not present in the linear case.

Table 3.3: Nondimensional Plate Deflection

| S | $\overline{w}$ | | | | | | |
|---|---|---|---|---|---|---|---|
| | CLPT | Pagano | Whitney | Linear FE | NL FE $q_0$=698.5 kPa | NL FE $q_0$=3492.5 kPa | NL FE $q_0$=6985 kPa |
| 10 | 0.9238 | 2.150 | 2.535 | 2.060 | 2.060 | 2.058 | 2.054 |
| 20 | 0.9238 | 1.300 | 1.317 | 1.186 | 1.184 | 1.148 | 1.069 |
| 30 | 0.9238 | 1.050 | 1.076 | 1.016 | 0.996 | 0.792 | 0.623 |
| 40 | 0.9238 | 0.950 | 0.990 | 0.956 | 0.846 | 0.501 | 0.358 |
| 50 | 0.9238 | 0.925 | 0.950 | 0.928 | 0.658 | 0.320 | 0.220 |

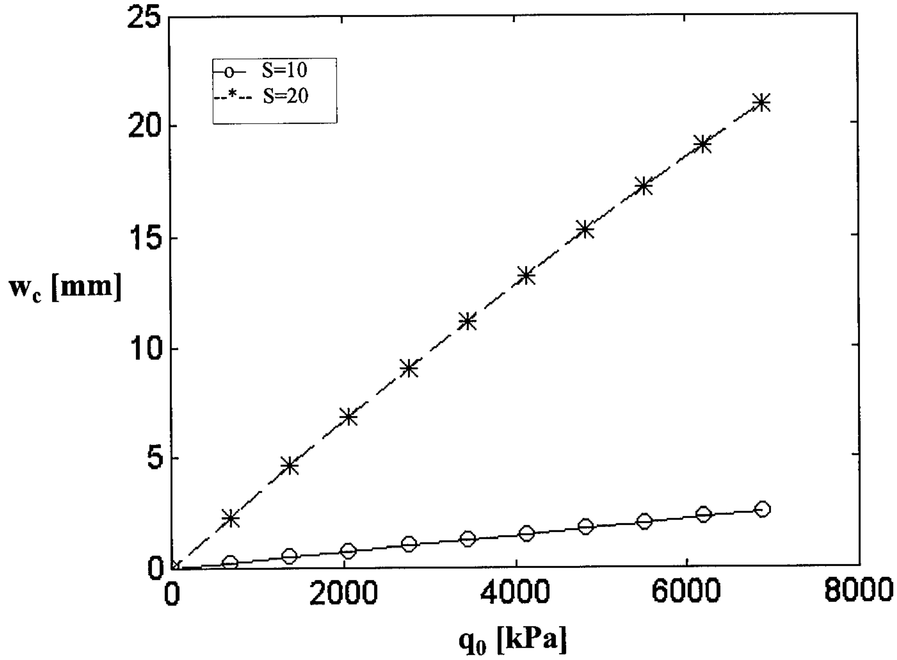


Figure 3.2: Plate Center Deflection (NL) vs. Peak Pressure

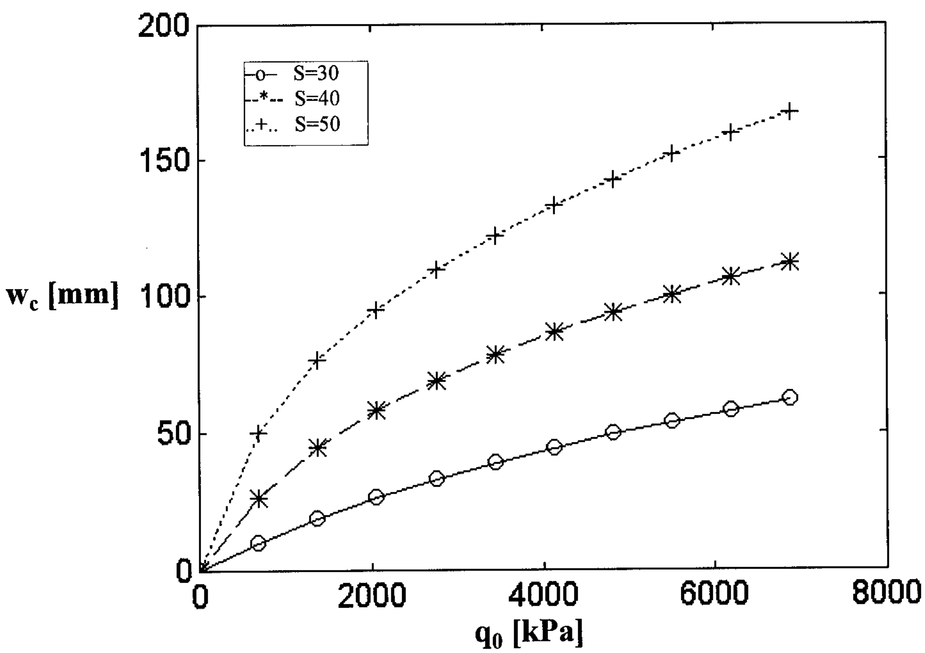


Figure 3.3: Plate Center Deflection (NL) vs. Peak Pressure

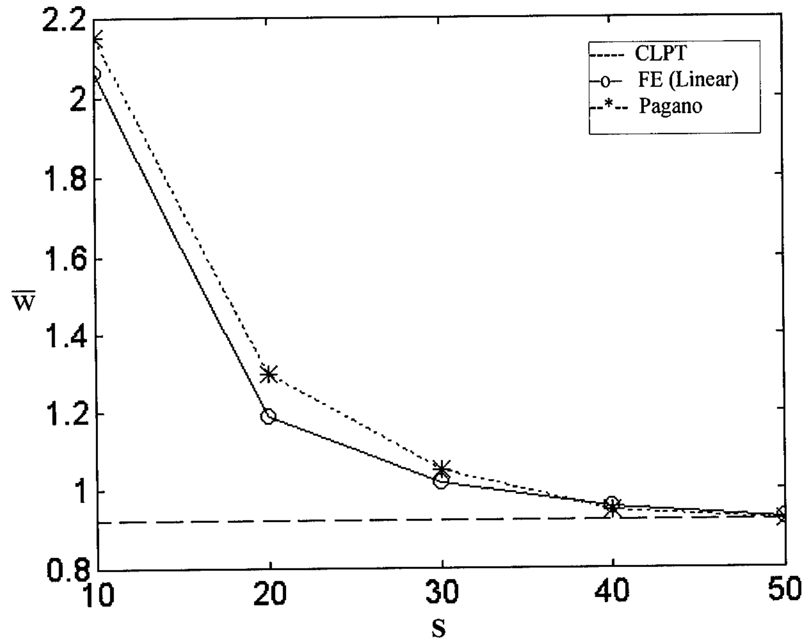


Figure 3.4: Nondimensional Plate Deflection (Linear) vs. Aspect Ratio

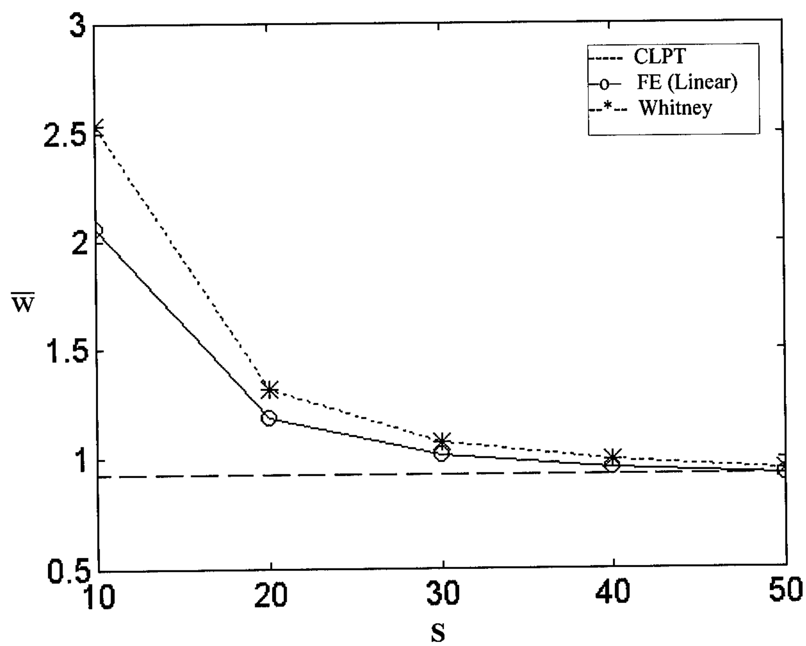


Figure 3.5: Nondimensional Plate Deflection (Linear) vs. Aspect Ratio

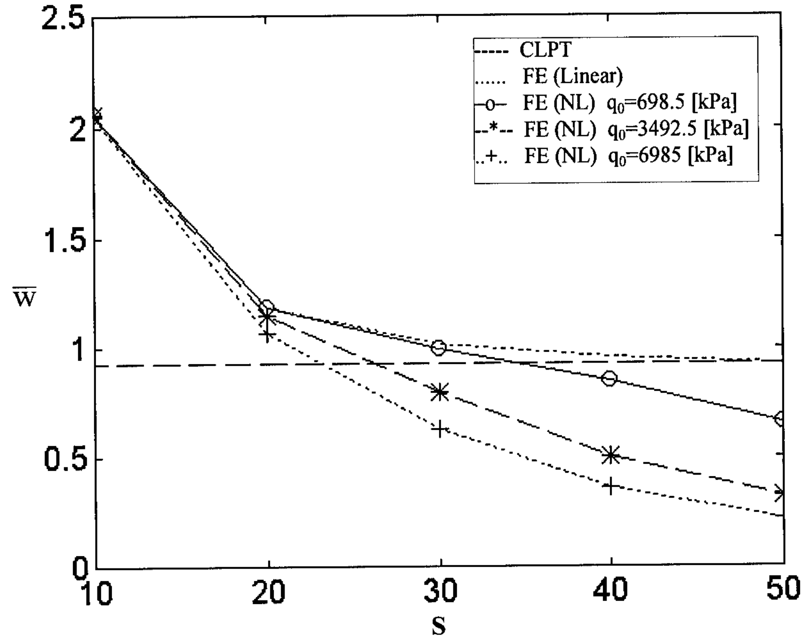


Figure 3.6: Nondimensional Plate Deflection (Linear and NL) vs. Aspect Ratio

# IV. Sandwich Plate Versus Composite Plate

## Finite Element Modeling

In order to demonstrate some of the advantages of sandwich plates in terms of stiffness-to-weight ratios, nonlinear displacement solutions for thin and thick sandwiches were obtained from SHELL and compared to solutions (also using SHELL) by Owens [16,17] for similar composite plates. His six plates were $[0_2/90]_s$ laminates of a transversely-isotropic graphite-epoxy composite. Each ply was 1.016 mm thick, thus h=6.096 mm. All plates were square and their widths were varied to produce aspect ratios (S=a/h) of 10, 20, 30, 40, 50 and 60. He computed nonlinear finite element solutions for simply-supported edges and uniform transverse pressures.

In this research, each sandwich plate had the same overall geometry as the plates modeled by Owens and were subjected to the same loads and boundary conditions. Each face sheet was made of the same graphite-epoxy material in a $[0/90_{1/2}]$ lay-up so that both faces comprised half of a plate's total thickness. The other (central) half was a honeycomb core made of Nomex™ (specifically classified as HRH-10-1/8-9.0) [8]. Figure 4.1 compares the geometry of both types of plates. The walls of the core's hexagon-shaped cells run parallel to the Z-axis, and the voids between the cells give it negligible in-plane stiffness compared to the facesheets. The relevant elastic and density properties of the face and core materials are:

$$\begin{array}{lll} \text{Face:} & E_{LL}=137.9\text{ GPa} & E_{TT}=3.447\text{ GPa} \\ & G_{LT}=G_{LZ}=1.724\text{ GPa} & G_{TZ}=0.6895\text{ GPa} \\ & \nu_{LT}=0.25 & \rho_f=1.6\text{ g/cm}^3\text{ (from [1])} \end{array} \qquad (4.1)$$

Core: $E_{LL}=E_{TT}=G_{LT}=0$

$$G_{LZ}=120.6 \text{ MPa} \qquad G_{TZ}=75.84 \text{ MPa} \qquad (4.2)$$

$$\nu_{LT}=0.5 \qquad \rho_c=0.14417 \text{ g/cm}^3$$

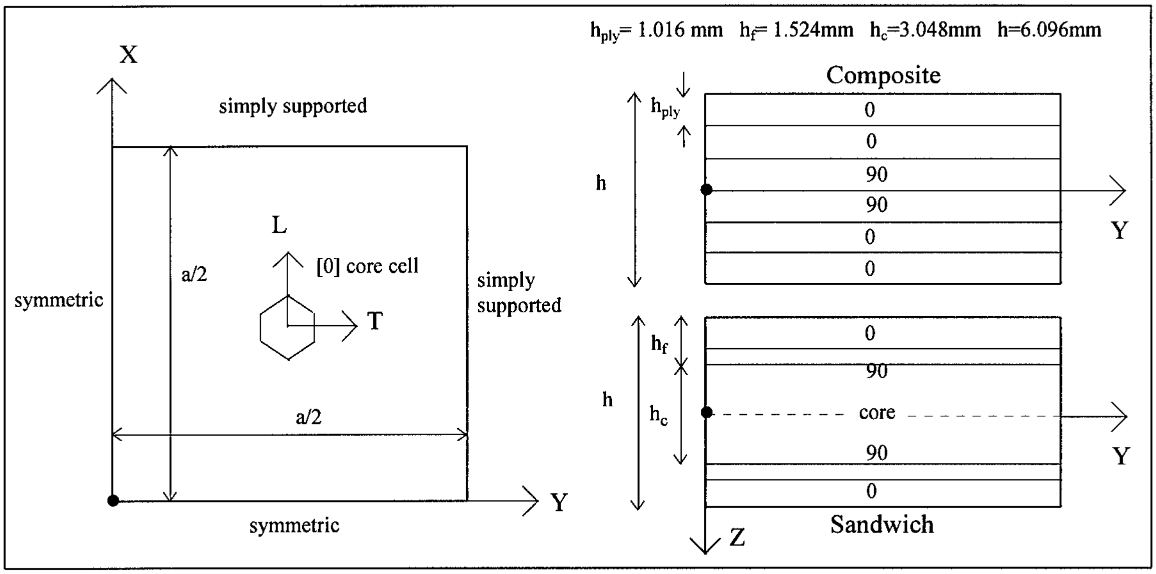


Figure 4.1: Composite and Sandwich Quarter-Plate Geometry

Just like the plates in the Pagano/ Whitney case study, the quarter-plate FE models for these sandwiches used an N by N mesh of square elements, and each mesh was refined to establish convergence. Furthermore, each plate geometry was solved twice to obtain results for core orientations of [0°] and [90°]. This was done to see if aligning the core's stiffest transverse plane with either the longitudinal or lateral directions of the outer face plies caused significant differences. Finally, the maximum uniform load applied to each plate was 6985 kPa, for which SHELL automatically generated the equivalent nodal forces.

Convergence Study

Displacement convergence at the plate center for the highest applied pressure was tested for three of the six plate geometries. The results are listed in Table 4.1. It was found that 12x12 meshes were acceptable for S=20 (and S=10 since it has a stiffer response). Similarly, 20x20 meshes converged reasonably well for S=30, 40, 50 and 60.

Table 4.1: Displacement Convergence for Sandwich Plate Meshes-[0°] Core

| S | $w_c$ [mm] for NxN mesh at maximum $q_0$ | | | | |
|---|---|---|---|---|---|
| | 12x12 | 16x16 | 20x20 | 24x24 | Min. % Deviation |
| 20 | 12.327 | 12.437 | | | 0.892 |
| 40 | 48.417 | 49.987 | 51.036 | 51.778 | 1.45 |
| 60 | 90.584 | 94.399 | 97.158 | 99.268 | 2.17 |

Results and Discussion

Figures 4.2 through 4.7 display the deflection of each sandwich plate's center as a function of applied pressure, and Figures 4.8 and 4.9 combine the first six graphs into a family of curves for easier comparison. The thickest plates, S=10, behaved very linearly over the entire pressure range, and the S=20 plates were linear up to about $q_0$=2000 kPa and then showed a very shallow nonlinear deviation. On the other hand, the thinner plates' deflections became highly nonlinear at pressures less than 1000 kPa. In addition, a [90°] orientation of the core caused the plate to be slightly more flexible for S=10 and 20, but for thinner plates, the curves were almost identical. In fact, Figures 4.5, 4.6 and 4.7 show only the results for a [0°] core to avoid redundancy. For a square plate with the same kind of support on every side, rotating the core through a right angle has the same effect as rotating both faces instead of the core. It is important to keep in mind that modifying the orientation of the same core or faces should have very different consequences if the whole plate was rectangular or had multiple types of edge supports.

Owens published deflection results [17] in a nondimensionalized form for selected pressures (which are also nondimensionalized) as a function of aspect ratio. The nondimensional forms of plate center deflection and applied pressure are:

$$\overline{w} = \frac{10 w_c \mathrm{E}_{\mathrm{LL\,face}}}{q_0 \mathrm{S}^4 h} \qquad \overline{q} = 10^4 q_0 / \mathrm{E}_{\mathrm{LL\,face}} \tag{4.3}$$

While $\overline{q}$ is directly proportional to $q_0$, the $S^4h$ term in $\overline{w}$ serves to cancel-out geometric effects on $w_c$ for CLPT solutions. Deflection results from the previous chapter (Figures 3.4 through 3.6) showed that $\overline{w}$ was a horizontal line-- unaffected by changes in S-- for the CLPT case. Furthermore, the weight of each plate can be considered by multiplying $\overline{w}$ by a ratio of the plate's overall density to the density of the face material $\rho_f$. The composite plates obviously had densities equal to $\rho_f$, so its nondimensional displacements were unchanged. On the other hand, the core accounts for half of each sandwich plate's volume. Therefore, the sandwich plates had an overall density equal to the average of the core and face densities. From the data in Equations 4.1 and 4.2:

$$\rho_s = \tfrac{1}{2}(\rho_f + \rho_c) = 0.872 \text{ g}/\text{cm}^3 = 0.545\rho_f \tag{4.4}$$

The resulting nondimensional variable $\overline{w}_\rho = \overline{w}\,(\rho_{plate} / \rho_f)$ can be used as an indicator for comparing two plates' stiffness-to-weight ratios.

Figures 4.10 through 4.17 compare the nondimensional nonlinear displacements of each composite plate and its corresponding sandwich plate (the same value of S) at four nondimensional load levels. Each consecutive pair of figures plots $\overline{w}$ and $\overline{w}_{\rho}$ versus S for a given $\overline{q}$. In Figures 4.10, 4.12, 4.14 and 4.16, the sandwich constructions were up to 85% more flexible than the composites . This is because the core material provides little bending stiffness (none for this FE modeling) and offers less resistance to transverse shear than the composite's additional face material in the central plies. Hence, the outer faces of each sandwich must compensate by bending more. Also note that thick sandwiches gain flexibility at a greater rate than the composite plates as S is reduced.

On the other hand, stiffness contributions due to bending and transverse shear from plies near the midsurface (either face or core material) have less significance in thinner plates. This caused the sandwich and composite curves to converge sharply at first and then become nearly parallel as S increased and bending in the outer faces became dominant. In addition, higher loads increased the rate at which the sandwich and composite curves converge and decreased the ultimate (large S) curve deviation.

When the nondimensional deflections of the sandwich plates were scaled-down to consider their lighter weights, the resulting plots (Figures 4.11, 4.13, 4.15 and 4.17) suggested equal or better stiffness-to-weight ratios than those of the composite plates for a wide range of aspect ratios. The $\overline{w}_{\rho}$ versus S curves also indicate that the range of S in which the particular sandwich outperforms the composite is bounded due to merging or crossing of the curves. Near S=10, the curves intersected due to the steeper increase in flexibility that is present in thick sandwiches. Furthermore, the sandwiches have lower axial stiffness for response to nonlinear membrane and flexural coupling, which may cause the curves to intersect again beyond S=60. The apparent advantages demonstrated by these sandwich plates, in terms of specific stiffness, may not hold true for all sandwich and composite plate constructions, but the specific case illustrates the potential benefits of using sandwich plates.

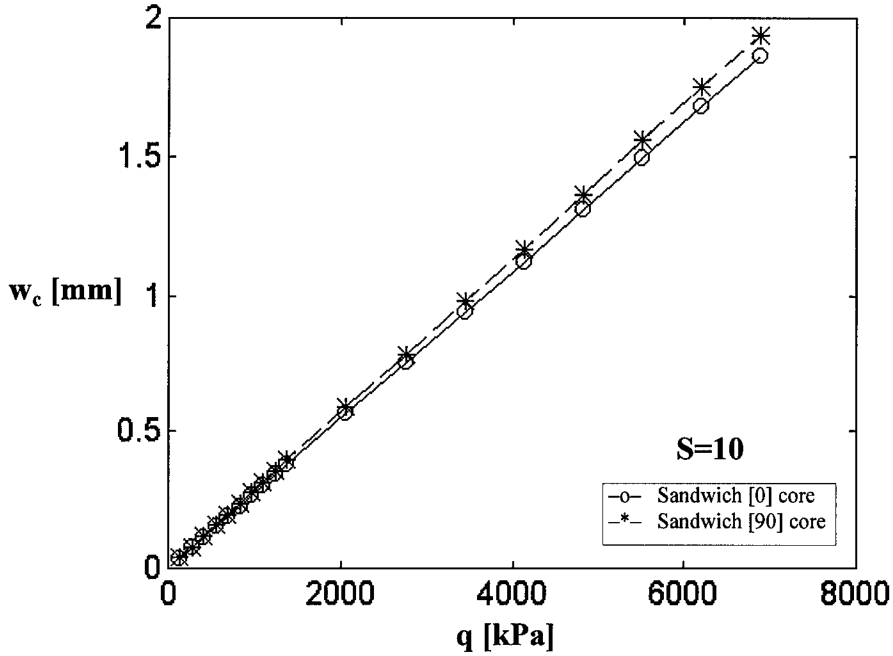


Figure 4.2: Plate Center Deflection vs. Uniform Pressure

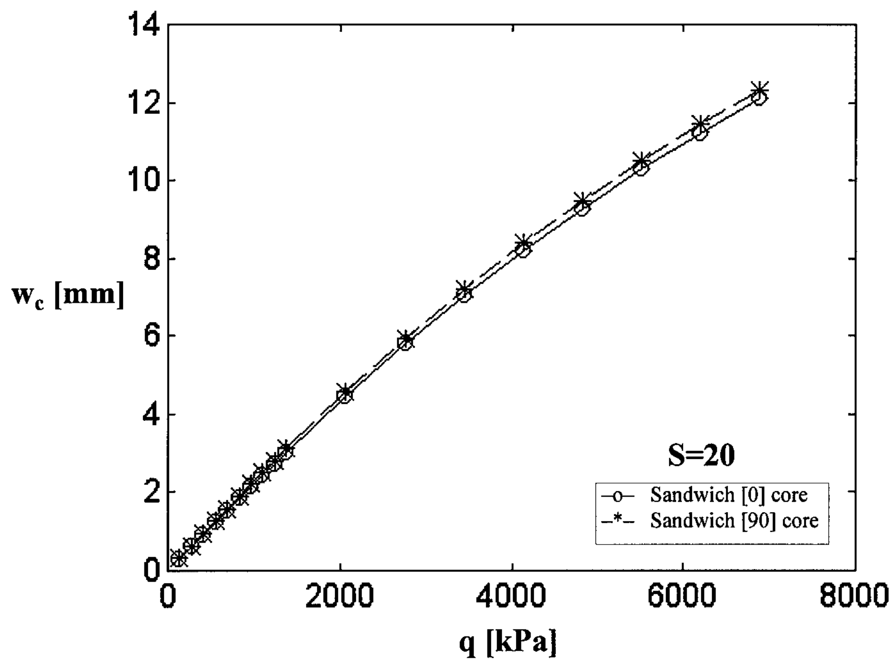


Figure 4.3: Plate Center Deflection vs. Uniform Pressure

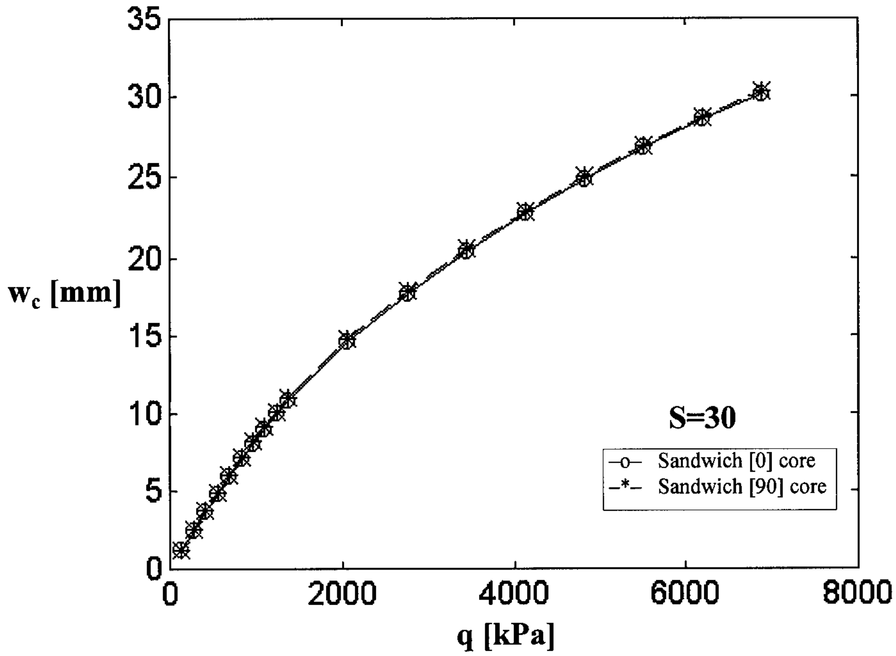


Figure 4.4: Plate Center Deflection vs. Uniform Pressure

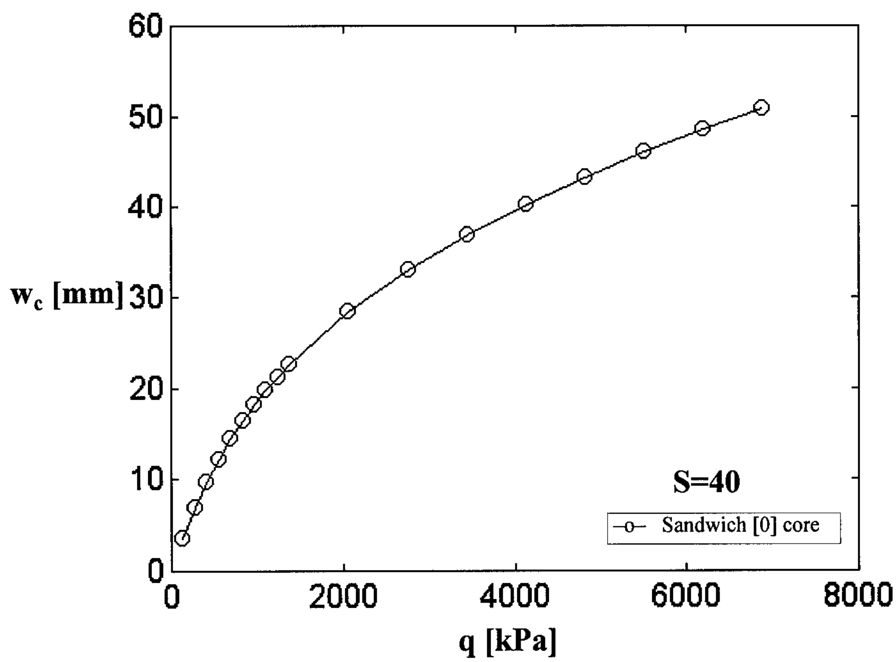


Figure 4.5: Plate Center Deflection vs. Uniform Pressure

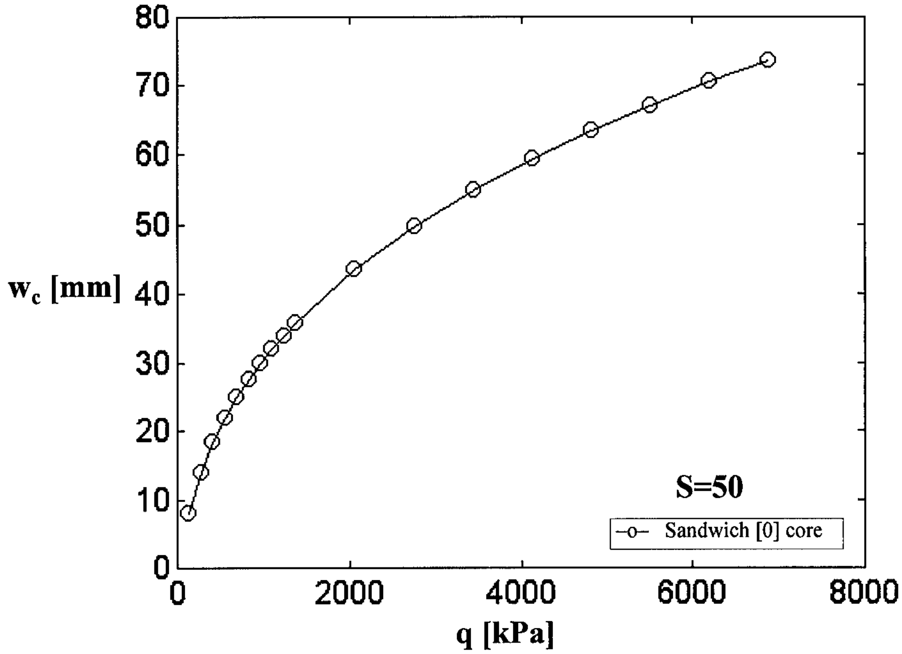


Figure 4.6: Plate Center Deflection vs. Uniform Pressure

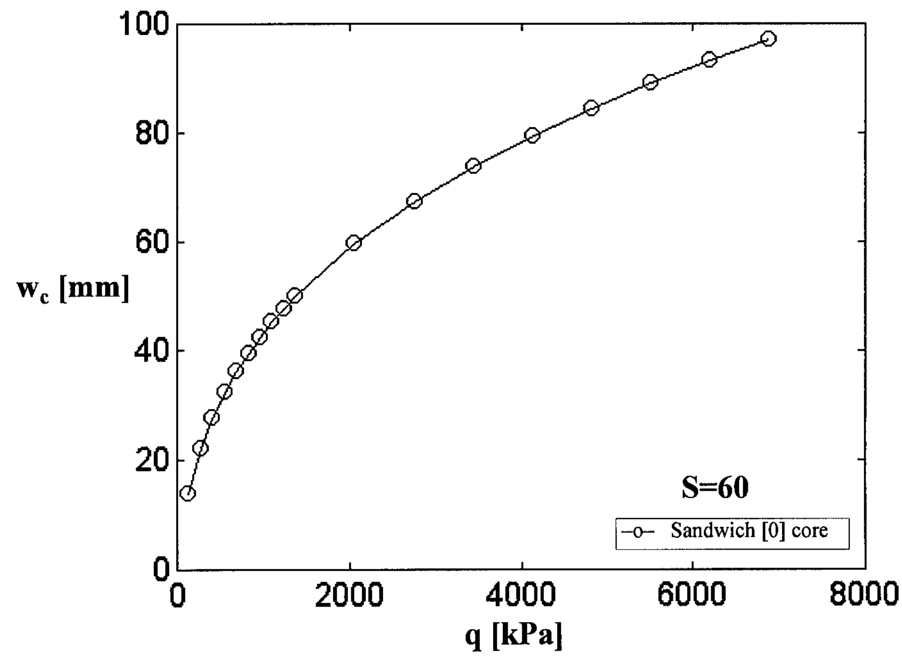


Figure 4.7: Plate Center Deflection vs. Uniform Pressure

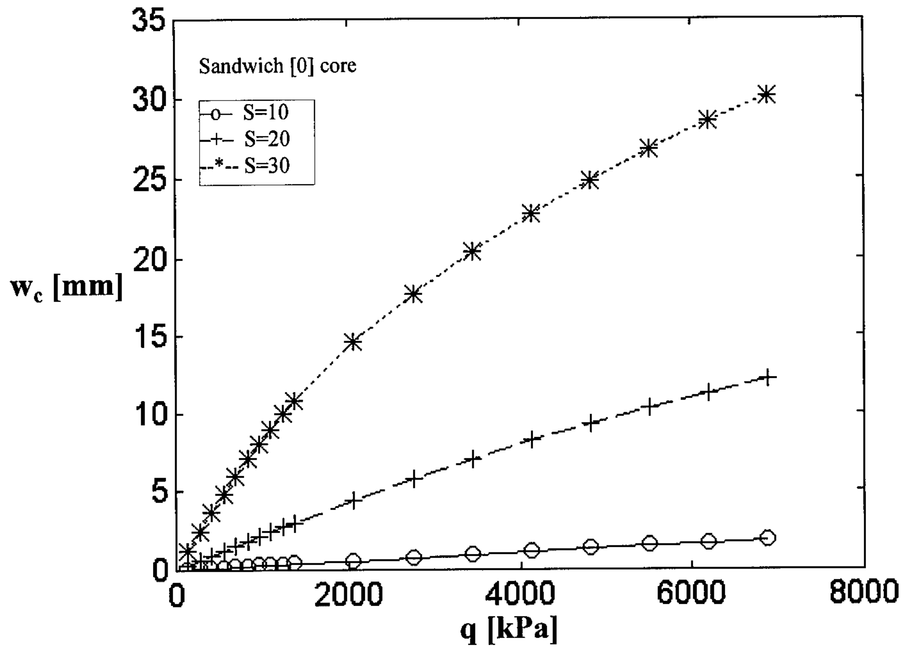


Figure 4.8: Plate Center Deflection vs. Uniform Pressure

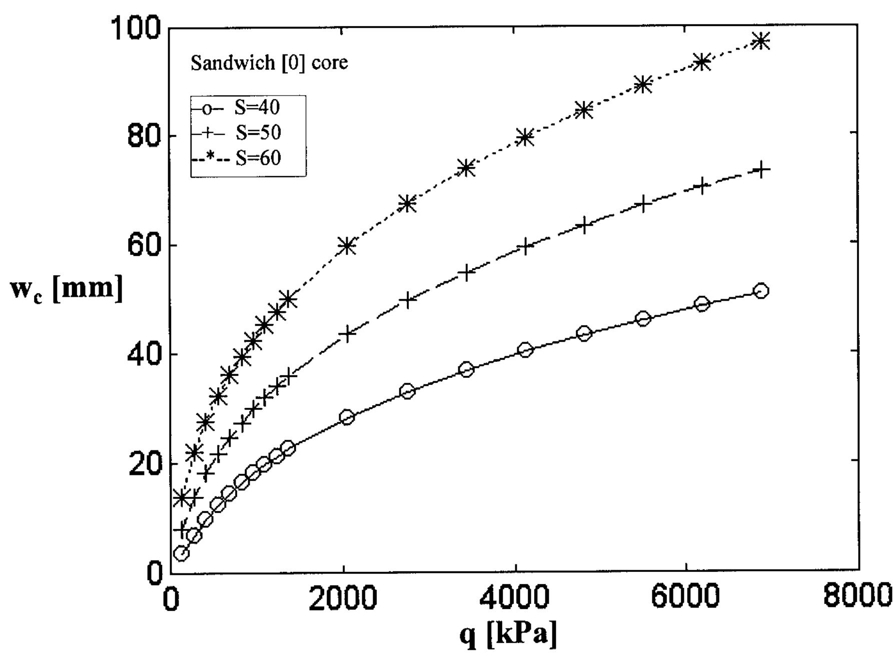


Figure 4.9: Plate Center Deflection vs. Uniform Pressure

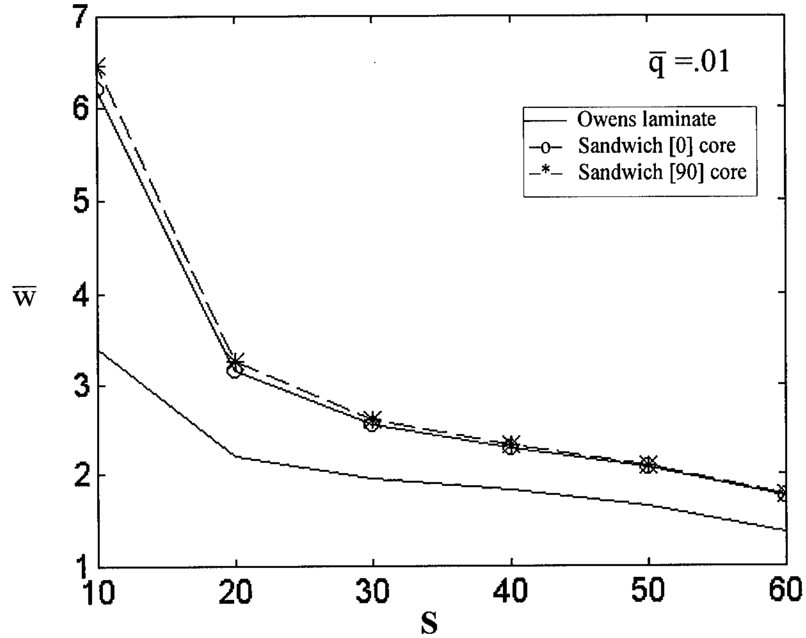


Figure 4.10: Nondimensional Plate Center Deflection vs. Aspect Ratio

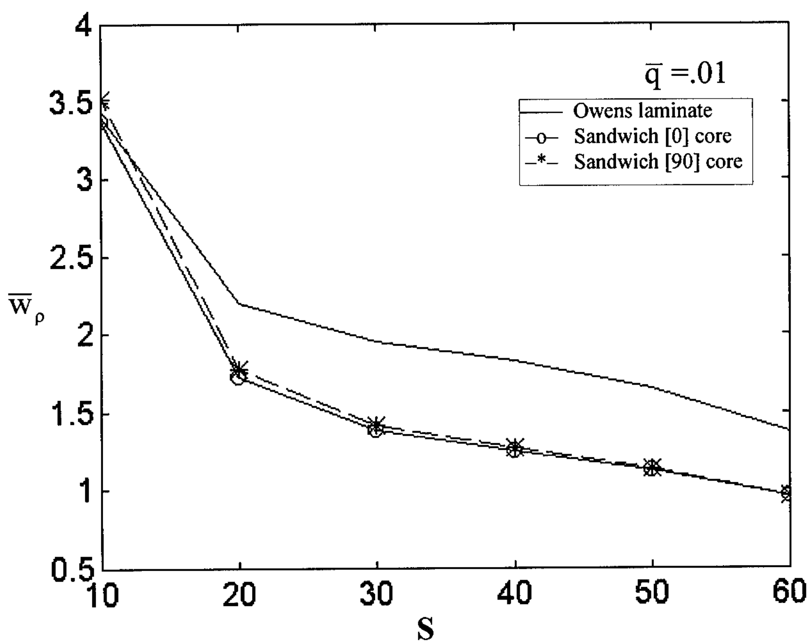


Figure 4.11: Nondimensional Plate Center Deflection (incl. weight) vs. Aspect Ratio

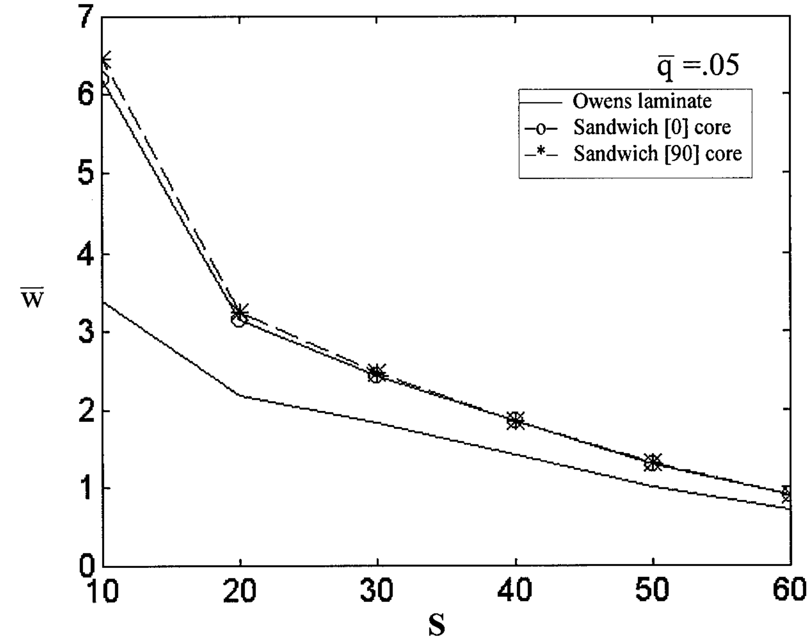


Figure 4.12: Nondimensional Plate Center Deflection vs. Aspect Ratio

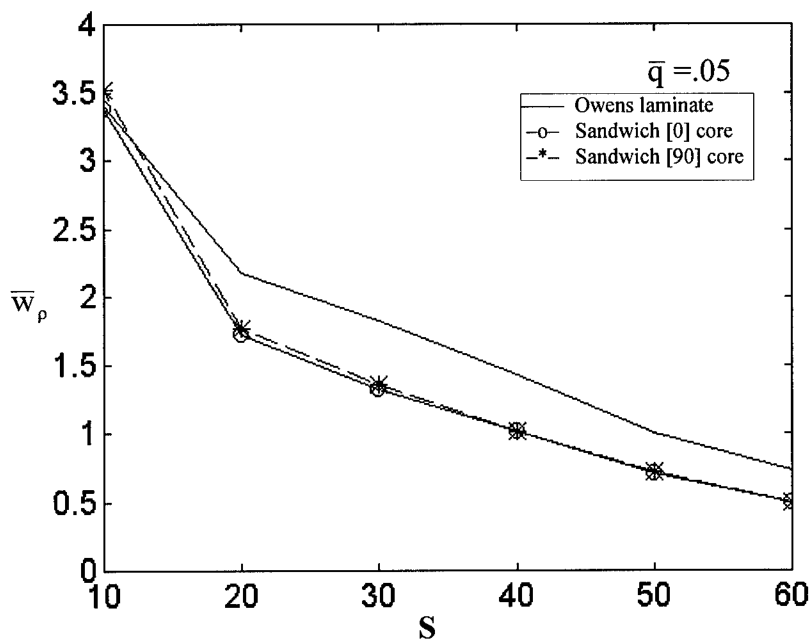


Figure 4.13: Nondimensional Plate Center Deflection (incl. weight) vs. Aspect Ratio

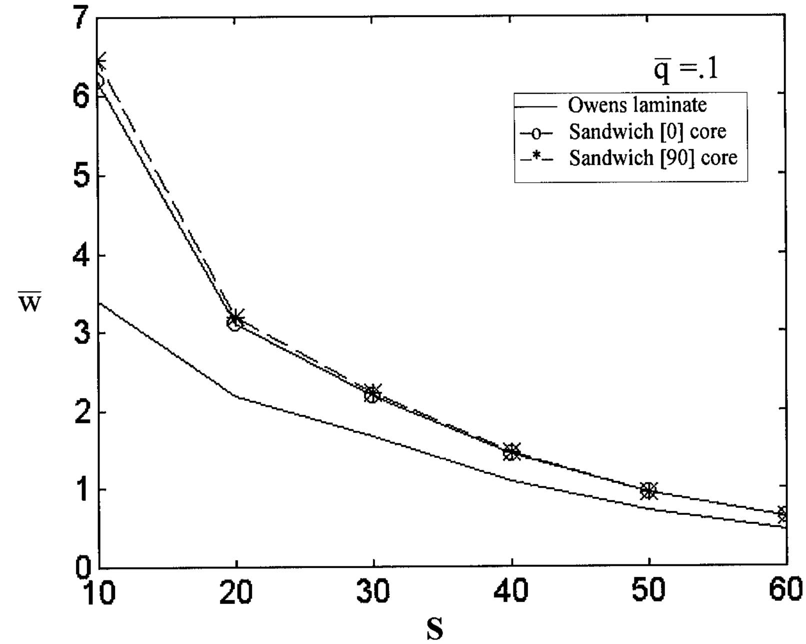


Figure 4.14: Nondimensional Plate Center Deflection vs. Aspect Ratio

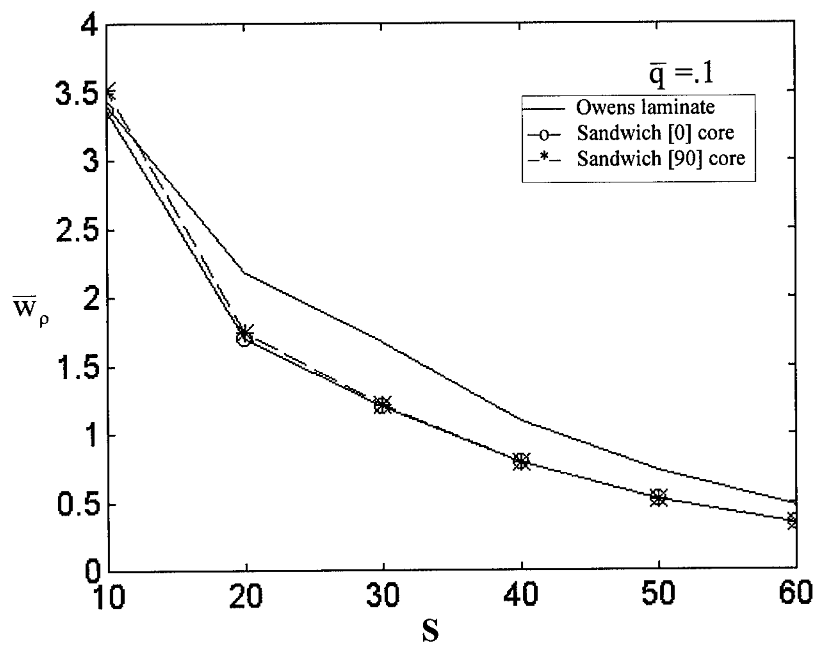


Figure 4.15: Nondimensional Plate Center Deflection (incl. weight) vs. Aspect Ratio

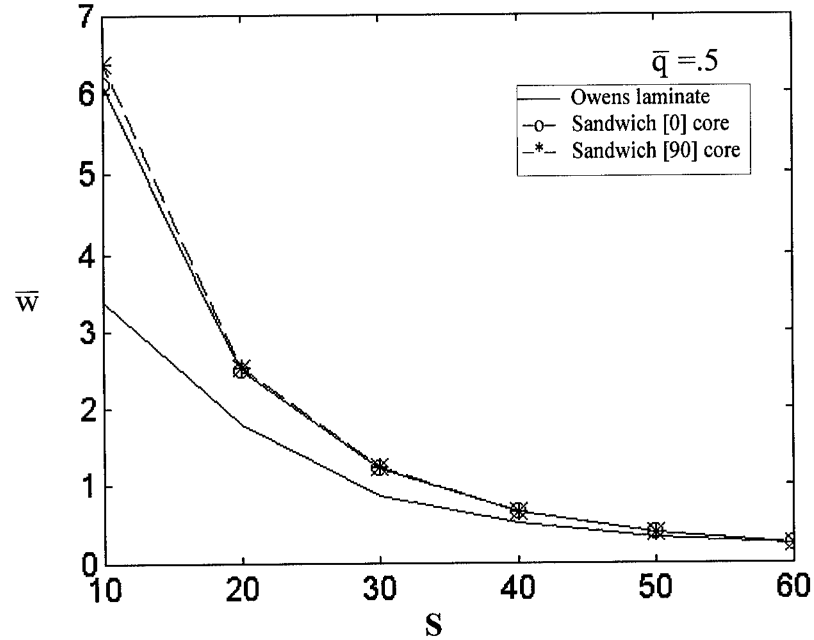


Figure 4.16: Nondimensional Plate Center Deflection vs. Aspect Ratio

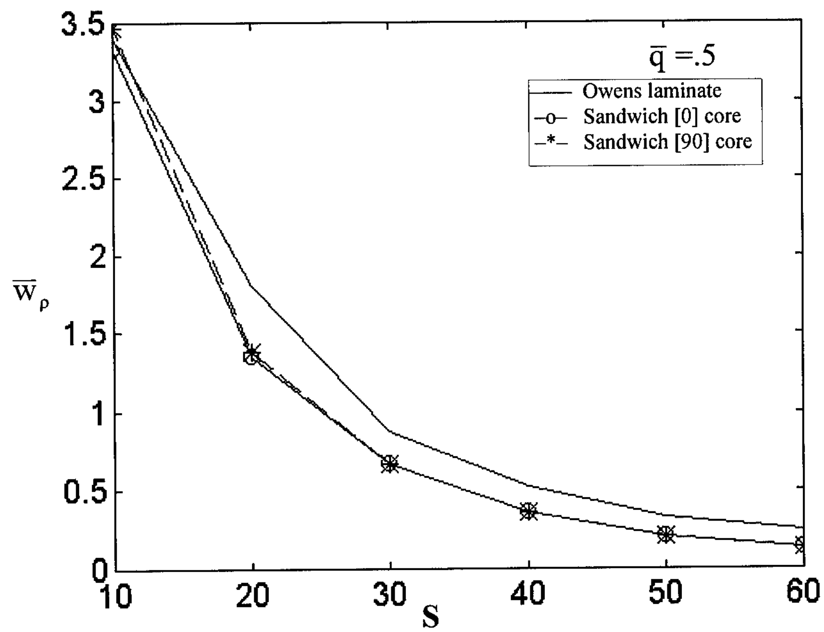


Figure 4.17: Nondimensional Plate Center Deflection (incl. weight) vs. Aspect Ratio

# V. Sandwich Plate Incipient Damage Predictions

## Finite Element Modeling

The preceding displacement solutions for various sandwich plates were obtained with SHELL under the assumption of perfect, linear elastic materials. However, an actual sandwich may experience some kind of initial failure well before any geometric nonlinearities take effect. Once this occurs, any solution SHELL generates beyond that point is invalid due to the presence of physical nonlinearities which the present code cannot consider. Therefore, FAILURE was written for the purpose of attempting to predict where, when and how a plate initially fails (using the maximum stress criteria presented in Chapter 2). Sample plate models were tested to verify the accuracy of its numerical and logical procedures (compared to manual calculations and expected output). Hence, the next step was to check the validity of its methodology by modeling actual plates and comparing FAILURE's results to what really happens. As previously mentioned in Chapter 1, a low-velocity impact for a composite plate has been found to have internal failure characteristics that can be predicted with a quasi-static response [10,12,14]. Thus, experimental impact studies on sandwich plates by Harrington [7] provided a means for applying the FAILURE program.

In Harrington's work, simply-supported square plates (a=127 mm) were subjected to impact loads at their center by the dropping of a spherical-nose punch from a series of heights. Four cases of plate and load combinations were considered for finite-element modeling, and these are listed in Table 5.1. The 4 and 16-ply facesheets were made of

AS4/3501-6 graphite-epoxy lamina with a thickness of 0.127 mm per ply. The 4-ply faces had $[0/90]_s$ lay-ups, while the 16-ply faces had $[0/90]_{4s}$ arrangements. Each plate's core was 12.7 mm thick and made of an HRX-10-1/8-9.0 Nomex™ honeycomb material [8]. Furthermore, two 0.254 mm thick adhesive layers of epoxy bonded each face to the core. For convenient referencing, the plates with 4 and 16-ply faces are denoted as Sandwich A and B, respectively. In addition, the numbering of the cases, one through four, in Table 5.1 corresponds to the radii of plate indentation (R) from smallest to largest. For each case, the peak load ($P_P$) equals the maximum instantaneous force acting on the plate within the experimental impact time history, and it is assumed quasi-static for FE analysis.

Table 5.1: Plate and Loading Cases

| Case | Sandwich Type (# Face Plies) | Plate Indentation Radius from Impact: R [mm] | Peak Load $P_p$ [N] |
|---|---|---|---|
| 1 | B (16) | 3.81 | 3304.6 |
| 2 | A (4) | 5.08 | 1620.9 |
| 3 | B (16) | 6.35 | 3914.0 |
| 4 | A (4) | 7.62 | 1969.7 |

The elastic properties of the face, core and adhesive layers have the following relevant values:

$$\begin{array}{lll} \text{Face:} & E_{LL}=119.3\text{ GPa} & E_{TT}=9.098\text{ GPa} \\ & G_{LT}=G_{LZ}=5.398\text{ GPa} & G_{TZ}=4.319\text{ GPa} \\ & \nu_{LT}=0.25 & \end{array} \tag{5.1}$$

$$\begin{array}{lll} \text{Core:} & E_{LL}=E_{TT}=G_{LT}=0 & \\ & G_{LZ}=120.6\text{ MPa} & G_{TZ}=75.84\text{ MPa} \\ & \nu_{LT}=0.5 & \end{array} \tag{5.2}$$

Adhesive (assumed isotropic): $E=3.447$ GPa

$$\nu=0.35 \tag{5.3}$$

In addition, the following maximum stress values were known or assumed:

$$\begin{aligned}
\text{Face:}\quad & \sigma_{LL\,max}=2.016\text{ GPa} && \sigma_{LL\,min}=-1.398\text{ GPa} \\
& \sigma_{TT\,max}=56.96\text{ MPa} && \sigma_{TT\,min}=-246.7\text{ MPa} \\
& \sigma_{LT\,max}=\sigma_{LZ\,max}=177.9\text{ MPa} && \\
& \sigma_{TZ\,max}=142.3\text{ MPa} &&
\end{aligned} \tag{5.4}$$

$$\begin{aligned}
\text{Core:}\quad & \sigma_{ZZ\,min}=-14.55\text{ MPa} && \\
& \sigma_{LZ\,max}=177.9\text{ MPa} && \sigma_{TZ\,max}=142.3\text{ MPa}
\end{aligned} \tag{5.5}$$

$$\text{Adhesive:}\quad \sigma_{max\ uniaxial}=108.8\text{ MPa} \tag{5.6}$$

$$\text{Face Ply Interface:}\quad \sigma_{XZ\,max}=\sigma_{YZ\,max}=142.3\text{ MPa} \tag{5.7}$$

FAILURE is not presently designed to directly use the core's transverse compressive strength ($\sigma_{ZZ\,min}$ from Equation 5.5) as a failure condition. This is because the user-defined Z-coordinate at which $\sigma_{ZZ}$ is calculated may or may not be the location of maximum compressive $\sigma_{ZZ}$ within the core. However, the reported $\sigma_{ZZ}$ values, if appropriate, can be manually used to check for initial failure due to crushing of the core.

The impact punch was assumed to create an ellipsoidal pressure distribution on each plate's top surface within its radius of indentation. This type of loading is similar to that obtained from Hertz contact stresses for isotropic materials [9]. Wu and Yen [23] also observed the presence of ellipsoidal distributions for composite plates in contact with rigid spheres. The ellipsoidal pressure (q) as a function of impact force (P) and radial distance from the center of the plate (r), has the following expression:

$$q(r) = q_0(1 - r^2 / \mathrm{R}^2)^{1/2}$$

$$r(x,y) = (x^2 + y^2)^{1/2} \tag{5.8}$$

$$\mathrm{P} = \int_0^{2\pi}\int_0^{\mathrm{R}} q(r)\, r\, dr\, d\theta = \tfrac{2}{3}\pi\, q_0\, \mathrm{R}^2$$

Note that the pressure is a peak value of $q_0$ at the center of the plate (x = y = r = 0) and goes to zero along the indentation radius (r = R). Also note that P refers to the impact force on the entire plate, while $q_0$ has the same pressure for both full and quarter-plate models. The load distribution parameters for each FE case are listed in Table 5.2 when $P=P_p$ .

Table 5.2: Ellipsoidal Impact Pressures

| Case | R [mm] | $P_p$ [N] | Maximum pressure at plate center: $q_0$ [MPa] |
|---|---|---|---|
| 1 | 3.81 | 3304.6 | 108.7 |
| 2 | 5.08 | 1620.9 | 30.00 |
| 3 | 6.35 | 3914.0 | 46.35 |
| 4 | 7.62 | 1969.7 | 16.20 |

These pressure distributions must be converted to discrete nodal loads in order to use SHELL. The method employed in Chapter 3 (pp. 3-3 and 3-4) was also used here by substituting Equation 5.8 for Equation 3.3 when integrating the product of the pressure and shape functions. Figure 5.1 shows three mesh arrangements that were used in the elliptical pressure zones. All plate centers were located at the node in the lower-left corner of each mesh. It was necessary to approximate the circular indentation zone with rectangles because FAILURE is limited to rectangular meshes. As a consequence, those elements which lie along the arc had to be integrated over a partial area when calculating their nodal loads. In order to minimize the number of elements requiring partial integration, the mesh lines in Figure 5.1 were designed so they intersected the arc at nodes. The 3x3 mesh was employed in each FE case, and the other two were only used in case 4 as part of the convergence study.

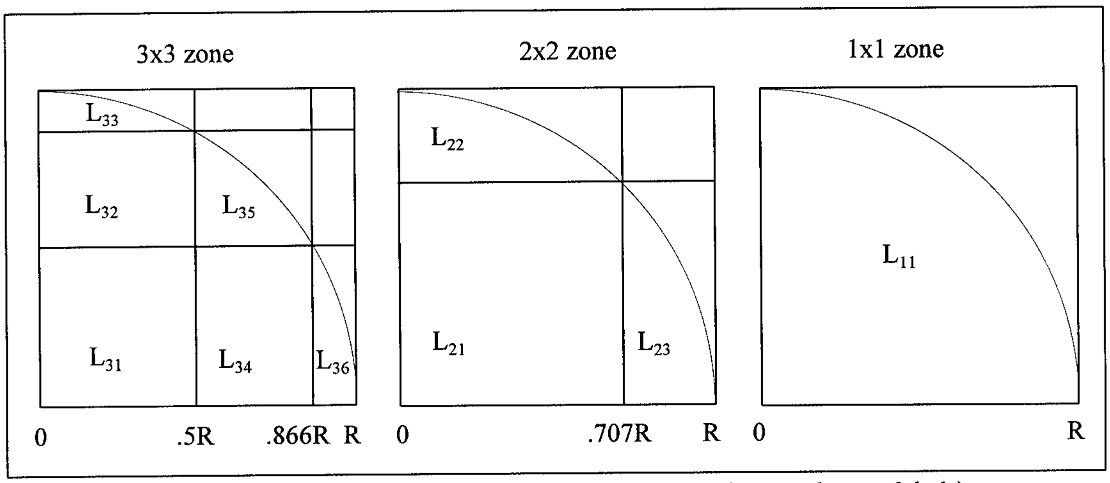


Figure 5.1: Quarter-Plate FE Meshes in Impact Zone ($L_{ij}$ are element labels)

Figures 5.2 through 5.9 display the entire mesh for each quarter plate model. Again, the center of each plate is the lower-left corner node. The size of each element can be determined from the element increment lengths (the distances between the nodes) listed in Table 5.3-- which are the same for both the X and Y-directions (defined upwards and to the right respectively) since all plates are square. The four 24-by-24 meshes were the primary models used for checking each case for failure, and the others applied to case four for convergence studies. The 23-by-23 and 22-by-22 meshes were coarser inside the impact zone, while the 18-by-18 and 12-by-12 meshes were coarser outside the impact zone.

Table 5.3: Element Sizing of Various Meshes

| Sandwich/ Loading Case | 1 | 2 | 3 | 4 | 4 | 4 | 4 | 4 |
|---|---|---|---|---|---|---|---|---|
| Mesh Resolution | 24x24 | 24x24 | 24x24 | 24x24 | 23x23 | 22x22 | 18x18 | 12x12 |
| Corresponding Figure | 5.2 | 5.3 | 5.4 | 5.5 | 5.6 | 5.7 | 5.8 | 5.9 |
| Element Increment | Increment Length (Node Spacing) [mm] | | | | | | | |
| 1st | 1.905 | 2.540 | 3.175 | 3.810 | 5.388 | 7.620 | 3.810 | 3.810 |
| 2nd | 1.395 | 1.859 | 2.324 | 2.789 | 2.232 | 0.707 | 2.789 | 2.789 |
| 3rd | 0.511 | 0.681 | 0.851 | 1.021 | 0.707 | 1.415 | 1.021 | 1.021 |
| 4th | 0.756 | 0.740 | 0.723 | 0.707 | 1.415 | 2.830 | 0.707 | 2.122 |
| 5th | 1.511 | 1.479 | 1.447 | 1.415 | 2.830 | 2.830 | 1.415 | 5.659 |
| 6th | 3.023 | 2.958 | 2.894 | 2.830 | 2.830 | 2.830 | 2.830 | 8.489 |
| 7th | 3.023 | 2.958 | 2.894 | 2.830 | 2.830 | 2.830 | 2.830 | 8.489 |
| 8th | 3.023 | 2.958 | 2.894 | 2.830 | 2.830 | 2.830 | 5.659 | 8.489 |
| 9th | 3.023 | 2.958 | 2.894 | 2.830 | 2.830 | 2.830 | 5.659 | 8.489 |
| 10th | 3.023 | 2.958 | 2.894 | 2.830 | 2.830 | 2.830 | 5.659 | 5.659 |
| 11th | 3.023 | 2.958 | 2.894 | 2.830 | 2.830 | 2.830 | 5.659 | 5.659 |
| 12th | 3.023 | 2.958 | 2.894 | 2.830 | 2.830 | 2.830 | 5.659 | 2.830 |
| 13th | 3.023 | 2.958 | 2.894 | 2.830 | 2.830 | 2.830 | 5.659 | |
| 14th | 3.023 | 2.958 | 2.894 | 2.830 | 2.830 | 2.830 | 2.830 | |
| 15th | 3.023 | 2.958 | 2.894 | 2.830 | 2.830 | 2.830 | 2.830 | |
| 16th | 3.023 | 2.958 | 2.894 | 2.830 | 2.830 | 2.830 | 2.830 | |
| 17th | 3.023 | 2.958 | 2.894 | 2.830 | 2.830 | 2.830 | 2.830 | |
| 18th | 3.023 | 2.958 | 2.894 | 2.830 | 2.830 | 2.830 | 2.830 | |
| 19th | 3.023 | 2.958 | 2.894 | 2.830 | 2.830 | 2.830 | | |
| 20th | 3.023 | 2.958 | 2.894 | 2.830 | 2.830 | 2.830 | | |
| 21st | 3.023 | 2.958 | 2.894 | 2.830 | 2.830 | 2.830 | | |
| 22nd | 3.023 | 2.958 | 2.894 | 2.830 | 2.830 | 2.830 | | |
| 23rd | 3.023 | 2.958 | 2.894 | 2.830 | 2.830 | | | |
| 24th | 3.023 | 2.958 | 2.894 | 2.830 | | | | |

Note: 1st Increment is at the center of the plate (lower left corner of each mesh)

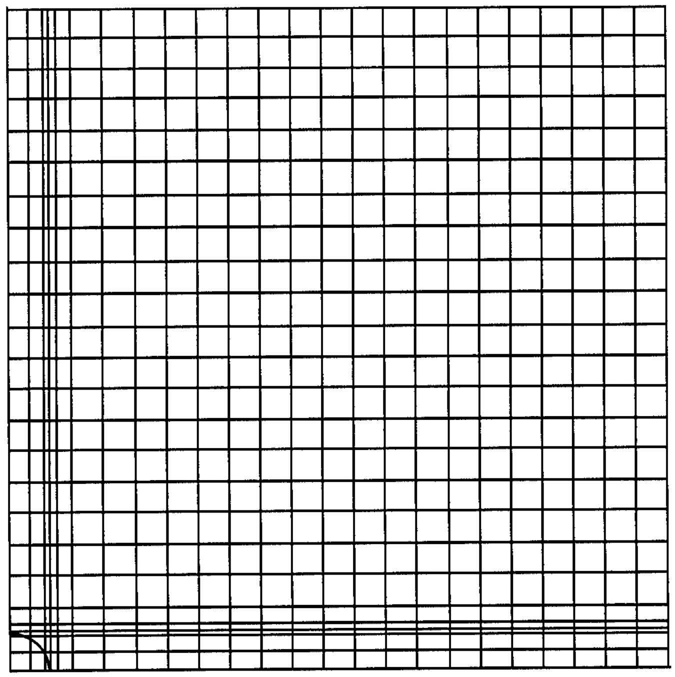

Figure 5.2: Case 1, 24x24 Mesh

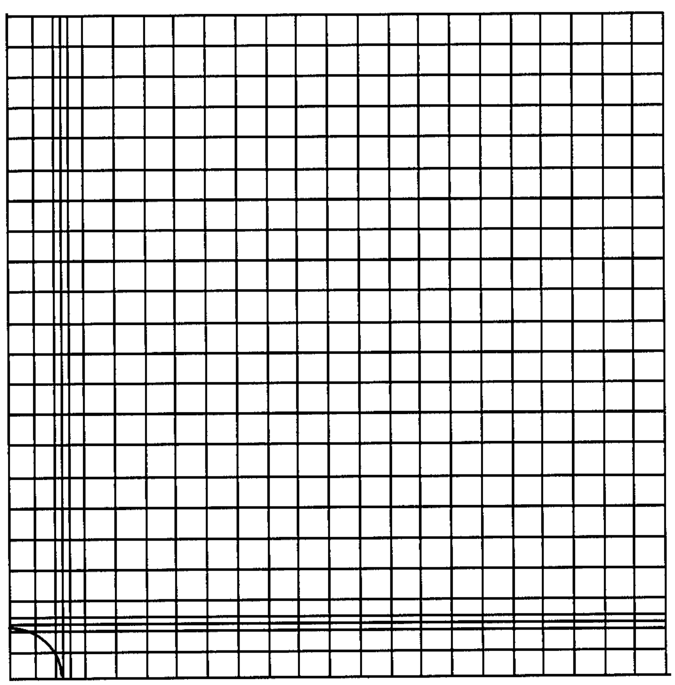

Figure 5.3: Case 2, 24x24 Mesh

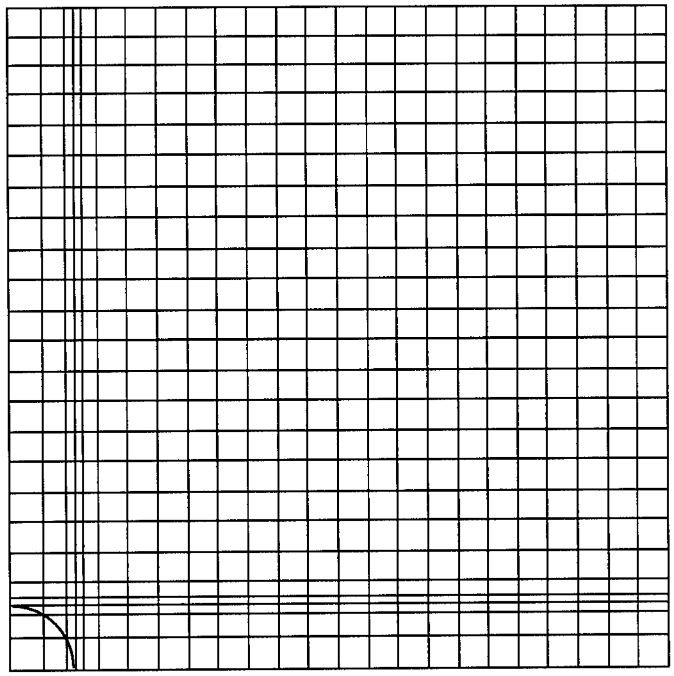

Figure 5.4: Case 3, 24x24 Mesh

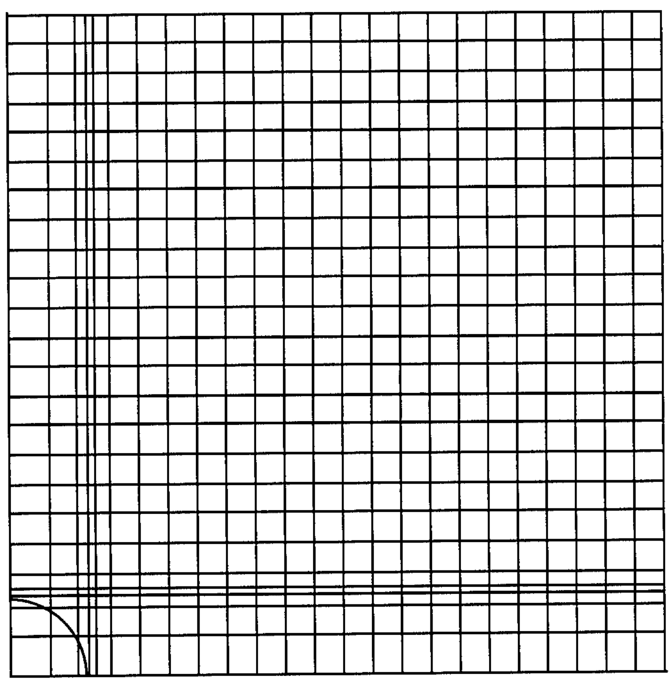

Figure 5.5: Case 4, 24x24 Mesh

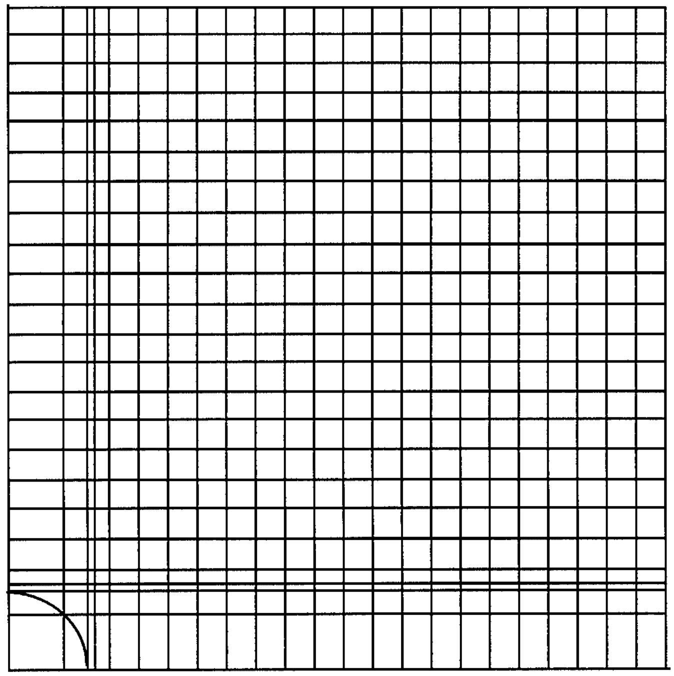

Figure 5.6: Case 4, 23x23 Mesh

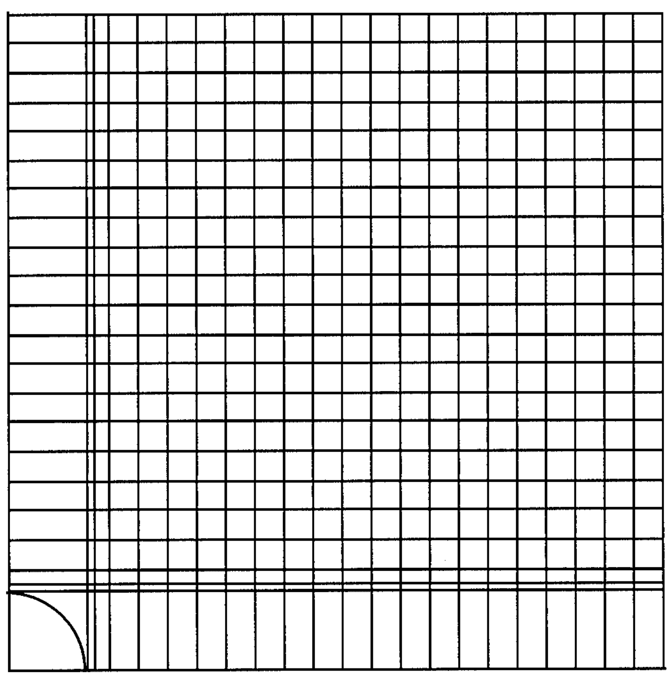

Figure 5.7: Case 4, 22x22 Mesh

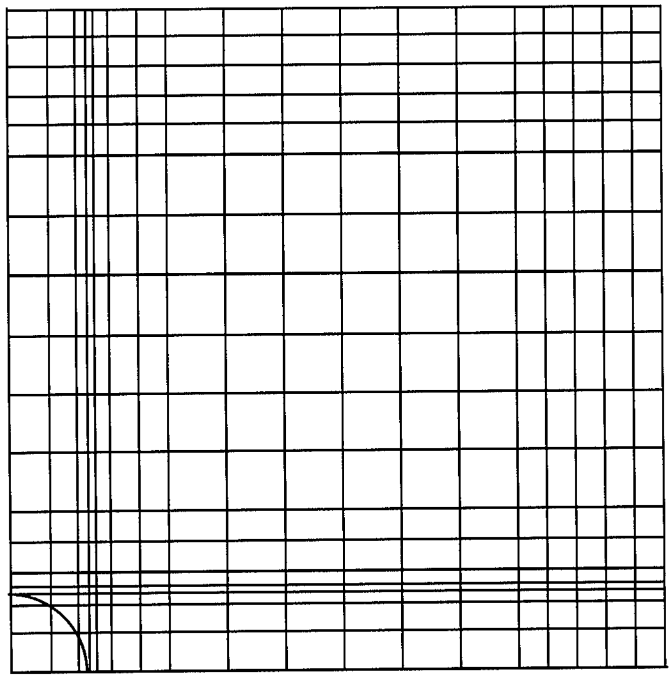

Figure 5.8: Case 4, 18x18 Mesh

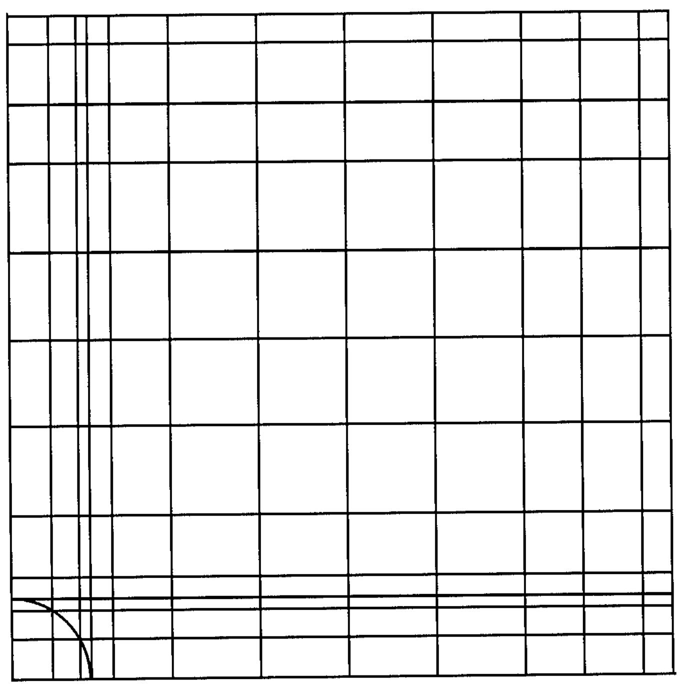

Figure 5.9: Case 4, 12x12 Mesh

Convergence Study

The 24-by-24 mesh for case 4 (Figure 5.5) underwent two types of refinement to check for convergence of the displacement results (within 5% relative deviation). First, the element spacing in regions outside the load zone was widened to form 18-by-18 and 12-by-12 meshes (Figures 5.8 and 5.9). The profile of plate deflection along the line x=y was compared for each mesh at the peak load. The profiles turned out to be identical (very small numerical deviations could not be identified graphically). This suggests that these plate models were practically insensitive to changes away from the applied load. The profile is not included here, since it does not show anything extraordinary in comparison to a typical profile for a simply-supported plate with a central transverse load.

Second, the mesh arrangement within the load zone for case 4 was modified according to Figure 5.1. This produced the 23-by-23 and 22-by-22 plate meshes shown in Figures 5.6 and 5.7. Calculations for $\sigma_{zz}$ were obtained for each mesh at the top of the plate near the center (where the pressure is highest). The distance from the center of the plate to the closest Gauss points in local elements $L_{11}$ , $L_{21}$ and $L_{31}$ (Figure 5.1) varies slightly for each mesh (Gauss point positions are fixed in the natural coordinate system and scaled for each element's dimensions). However, they are all near enough to the center that the variation in the elliptical pressure distribution is negligible with respect to the peak value. For $q_0$=16.20 MPa, the calculated values, in decreasing order of mesh resolution, were: 16.14 , 13.09 and 10.46 MPa. Hence, the 24-by-24 mesh converged to within 0.4% of the actual value.

Results and Discussion

For every finite element case, the plate's maximum deflection at peak load was very small relative to its geometry. Sandwich A (4 ply face) is the more flexible plate, and for it's highest load (case 4) the midsurface deflection at the plate's center was only 0.71mm-- about 5% of its total thickness ($h_A$=14.224 mm). Therefore, although each case was solved nonlinearly, its static results were practically linear.

Table 5.4 lists the incipient failure results for each case, in which all maximum stresses except transverse core compression were considered (Equations 5.4 through 5.7).

Table 5.4: Initial FE Sandwich Failure Ignoring Core Compression

| Case (Sandwich Type / # Face Plies) | $P_p$ [N] | % of $P_p$ Range for First Failure | Mode of Failure | Location (material) |
|---|---|---|---|---|
| 1 (B / 16) | 3304.6 | 110-120 | Lateral Tension | Bottom of plate near center (face) |
| 2 (A / 4) | 1620.9 | 80-90 | Transverse Shear | Midsurface near center (core) |
| 3 (B / 16) | 3914.0 | 150-160 | Lateral Tension | Bottom of plate near center (face) |
| 4 (A / 4) | 1969.7 | 100-110 | Transverse Shear | Midsurface near center (core) |

In each case, the initial failure occurred near or beyond the peak impact force. Tension fracture of the face's matrix material on the bottom of the plate is a possible failure mode for a standard laminated composite, but it is not realistic for these kinds of sandwiches. The lower surface of the top face is more likely to fail that way as a progressive mode when core damage from indentation removes the top face's localized support in the Z-direction. It occurs on the bottom face for sandwich B because SHELL's plate kinematics (Equation 2.1) prevents describing indentation. The top and bottom faces (for a fixed x and y) must have the same w-translation as the midsurface. Transverse shear failure of sandwich A's core is a possible mode in later stages, but core crushing is more likely to occur first.

Before examining the core for transverse compression failure, it was necessary to see if Equation 2.12 was successful in obtaining values of $\sigma_{zz}$ in the impact zone that were close to the applied pressure (Equation 5.8) on the top surface (or at least close to $q_0$ near the plate's center). Table 5.5 compares the known pressures to the calculated stresses.

Table 5.5: Maximum Compressive Stress Results at Top and Center of Plate

| Case | $q_0$ [MPa] | $\sigma_{zz}(0,0, -h/2)$ [MPa] | % error |
|---|---|---|---|
| 1 | 108.7 | 70.44 | 35 |
| 2 | 30.00 | 28.37 | 5.4 |
| 3 | 46.35 | 45.72 | 1.3 |
| 4 | 16.20 | 16.14 | 0.4 |

For cases 2 through 4, the stress results were very good, but case 1 underpredicted the applied pressure by a large margin. There is an inverse correlation between the error and radius of the pressure zone (the indentation radius). Since pressure must drop from a peak value to zero within this zone, its gradient may be too high in case 1 for the employed mesh . However, finer meshes were not generated because the predominantly manual process used to calculate the nodal loads limited the practicality of finer meshes. Mathematica™ [22] was used for the complex integrations, but due to difficulties in automating the entire process, each element's nodal loads had to be individually evaluated and manually added to the nodal loads in adjacent elements. Besides, the other three cases provided sufficient data for comparing core crushing predictions.

Table 5.6 lists each case's calculated value of $\sigma_{zz}(0,0, -h_c/2)$ at the top of the core and the center of the plate, using Equation 2.12, for the highest load levels. The linear behavior exhibited by all the plates permitted the use of linear interpolation to determine the failure loads at which the FE models predicted core crushing.

Table 5.6: FE Failure Load Predictions for Core Compression

| Case | $q_0$ [MPa] | $\sigma_{zz}(0,0, -h_c/2)$ [MPa] at $q_0$ | % of $q_0$ for core failure at 14.55 [MPa] pct=14.55 / $\sigma_{zz}(0,0, -h_c/2)$ x 100% | $P_p$ [N] | FE Failure Load [N] $P_p$ x pct |
|---|---|---|---|---|---|
| 1 | 108.7 | 42.94 | 34 % | 3304.6 | 1123.6 |
| 2 | 30.00 | 23.56 | 62 % | 1620.9 | 1005.4 |
| 3 | 46.35 | 26.26 | 55 % | 3914.0 | 2152.7 |
| 4 | 16.20 | 13.40 | 109 % | 1969.7 | 2146.9 |

Compared to the other initial failure results in Table 5.4, crushing of the core within the impact region occurred at substantially lower fractions of the maximum load for each case, except 4. Furthermore, although case 4 did not show failure from any mode until the peak load was exceeded, both tables predicted some kind of core failure (either crushing or shearing) at nearly the same load level. Hence, through the use of an analytical method that assumed zero transverse normal stresses in determining the static response, it was still possible to detect transverse compression of the core as the primary mode (or one of several modes) in which these types of sandwich plates initially fail. Regardless of the projected load levels, the mode agrees with Harrington's experimental findings which showed incipient indentation damage, at least partially due to core crushing, within the impact zone for sandwiches with 4 and 16-ply faces.

In Figures 5.10 through 5.13, the FE load levels at predicted core failure have been superimposed, for their respective cases, onto the actual time histories of impact loading from Harrington's experiments. Actual initial failure is usually represented by the first sharp drop in load (in excess of noise on the curve), which signifies a sudden shift in a plate's equilibrium state due to a reduction in stiffness. For the 16-ply face sandwiches (type B) in cases 1 and 3, such drops were clearly evident at loads of about 2500 N for both Figure 2.10 and 2.12. The FE failure load in case 1 underestimated the onset of failure by about 50%, but case 3 provided a very close estimate of the actual failure load.

For the 4-ply face sandwiches (type A) in cases 2 and 4, the actual failure loads were not clearly distinguishable from the noise present in Figures 5.11 and 5.13, although moderate spikes between 500 and 1000 N may or may not be due to failure. The sandwich A plates experienced more data noise than the sandwich B plates because they were more flexible and thus subjected to greater momentum transfer during the impact. In addition, sandwich A displayed the same phenomenon as sandwich B in the finite element results-- a near doubling of the projected first-core-crushing load for the case with the higher applied load. One would expect the loads that cause initial internal failure to remain constant for different applied loads on the same plate (as they clearly do for sandwich B). This inconsistency between the finite element and experimental results suggests that maximum stress may not be the best choice of failure criteria. Although the primary mode was core crushing, other stress components may contribute to the initial failure. The maximum stress criteria isolate the stresses and do not permit coupling to affect the failure predictions.

Another source of inconsistency in the failure results could be the FE modeling of the pressure distribution. Cases 2 and 4 had larger impact forces than cases 1 and 3, respectively, but their larger indentation radii (the assumed constant radii of the impact zones) spread out the loading so that the peak pressures actually dropped for higher total loads, as shown in Table 5.2. In the former cases, the values of $q_0$ were roughly cut in half, which explains the doubling of the projected load for initial internal failure. In an actual impact, both total load and contact area vary with time, and the right combination produces a peak pressure high enough to initiate failure in the core. This type of nonlinear behavior cannot be modeled with SHELL; therefore, the good results that occurred in case 3 were most likely a coincidence brought about by nearly having the right mesh size to produce the actual failure load.

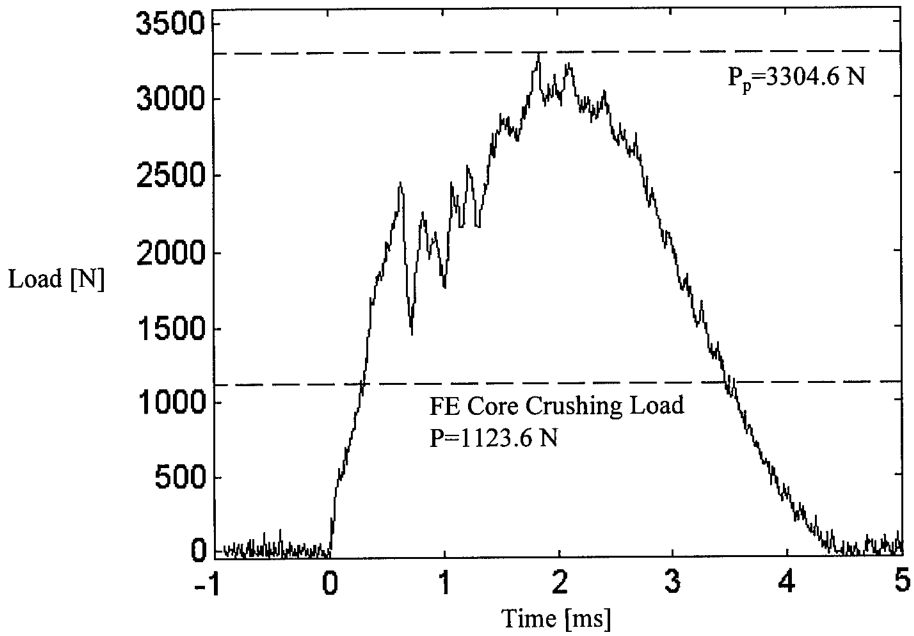


Figure 5.10: Impact Load vs. Time [Ref 7]

Case 1- 16 ply face (Sandwich B)

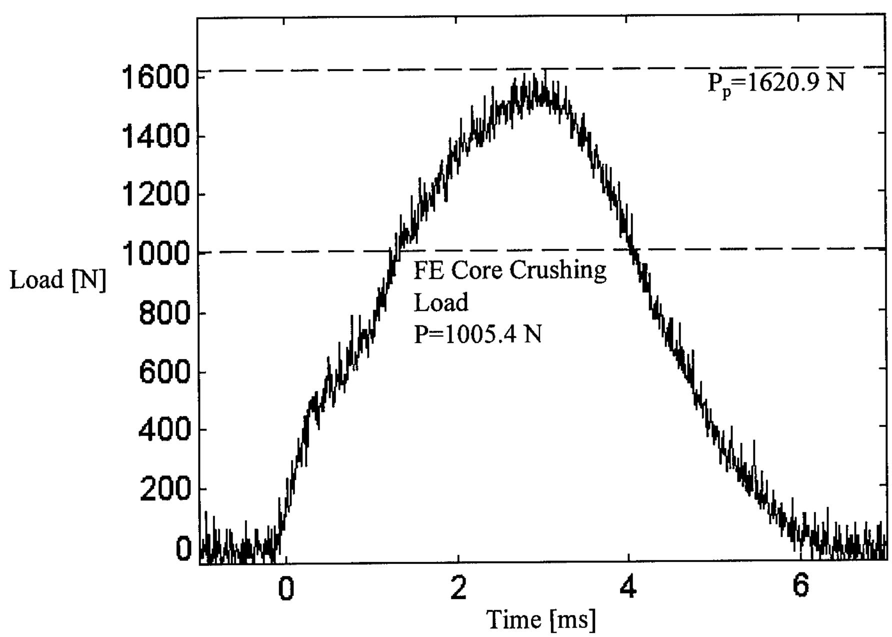


Figure 5.11: Impact Load vs. Time [Ref 7]

Case 2- 4 ply face (Sandwich A)

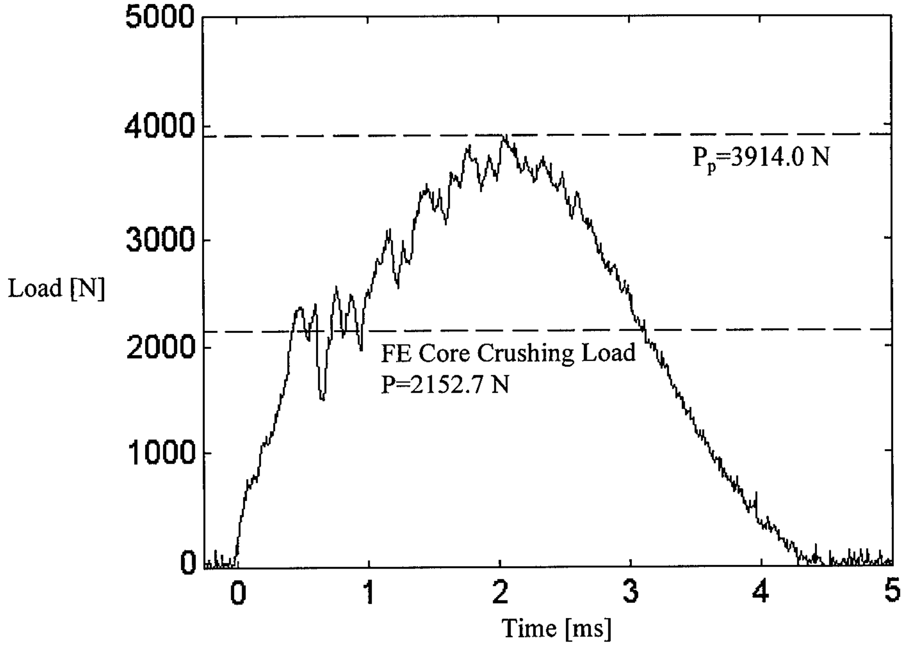


Figure 5.12: Impact Load vs. Time [Ref 7]

Case 3- 16 ply face (Sandwich B)

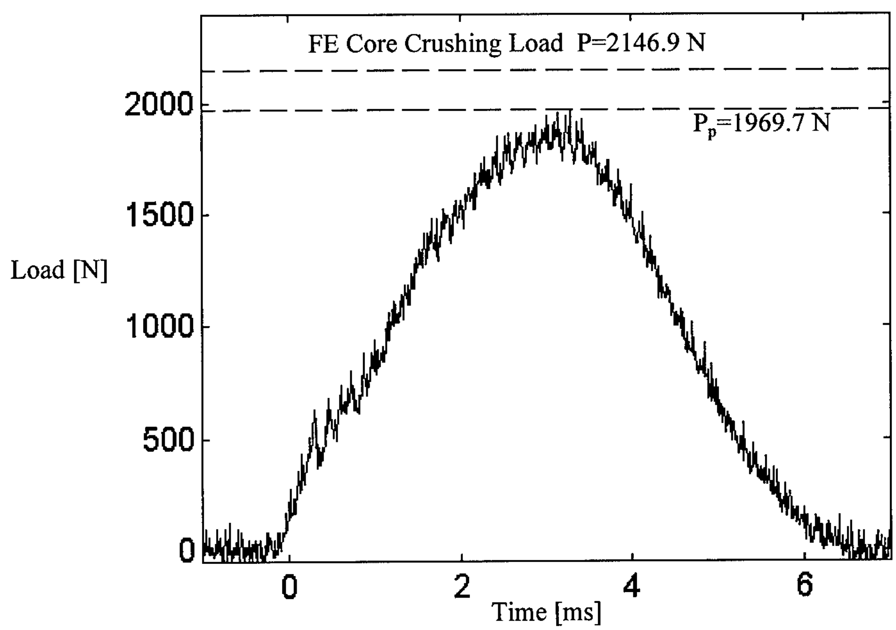


Figure 5.13: Impact Load vs. Time [Ref 7]

Case 4- 4 ply face (Sandwich A)

# VI. Conclusions

In this thesis a geometrically nonlinear finite element program, created for static analysis of composite plates and shells, was enhanced so that it could be used to study sandwich plates. Furthermore, a new, separate postprocessing unit was created in order to detect initial failure in a plate using the maximum stress criteria. The program was also given the capability of estimating the transverse normal stresses within a plate by enforcing equilibrium through its thickness. Three case studies were investigated for the following purposes:

1. To validate the sandwich plate enhancements to the FE code by comparing its displacement results to those of established linear solutions for a particular sandwich plate problem.

2. To compare the stiffness and stiffness-to-weight characteristics of regular composite plates to those of sandwich plates for different load intensities and aspect ratios.

3. To simulate low-velocity impact tests on sandwich plates with a quasi-static FE solution and attempt to predict incipient plate damage using the maximum stress criteria.

Linear solutions for simply-supported sandwich plates, under a sinusoidal transverse pressure, showed good agreement with Pagano's exact elasticity solution and Whitney's laminated plate solution for both thick and thin plates. For thin plates, all three methods converged to the CLPT solution. The code enhancements related to sandwich plates only affected the formulation of the constitutive relations in the preprocessor. Since they do not change for either a linear or nonlinear solution of the same plate, the modified code can be considered valid for sandwich plates using either solution method.

Comparisons between a graphite-epoxy composite and a sandwich with similar facesheets, a honeycomb core and the same overall geometry, show that the sandwich is more flexible (especially for thick plate). The differences in stiffness become smaller for thin plates as bending in the outer faces dominate the response. When the lighter weight of the core material is taken into account, the sandwich plate demonstrates a significantly higher stiffness-to-weight ratio than the composite for both thick and thin plate within certain bounds. If specific stiffness is the primary criterion in selecting a material, a sandwich construction may be a better choice than a laminated composite of similar construction.

The procedure for calculating $\sigma_{zz}$ was shown to be capable of highly accurate estimates of transverse pressures on the top surface of a plate, provided the FE mesh in these areas were properly refined. Therefore, the method was partially successful in extracting a three-dimensional stress state from a two-dimensional solution. In addition, for the case of sandwich plates subjected to low-velocity impact loads, the use of quasi-static FE modeling and the maximum stress criteria was successful in detecting core crushing in the impact zone as one of the primary modes of incipient damage. However, inconsistencies in the predicted load levels for initial failure suggest the need for a more complex criteria that considers stress coupling. Although the presence of time-dependent nonlinearities, like momentum transfer and variable contact areas, also contributed to preventing the quasi-static models from making good predictions of when initial failure occurred, this quasi-static approach was at least able to identify where and how it occurred.

## References

1. Agarwal, B.D. and L.J. Broutman Analysis and Performance of Fiber Composites, 2nd Ed.: 190-197, 437. New York: John Wiley and Sons, 1990.

2. Cook, R.D., D.S. Malkus and M.E. Plesha Concepts and Applications of Finite Element Analysis, 3rd Ed.: 95-112,170-173. New York: John Wiley and Sons, 1989.

3. Dandy, L.T., A.D. Kelkar and R.S Sandhu "Analysis of Progressive Damage in Thin and Thick Composite Laminates Subjected to Low-Velocity Loadings," Ninth International Conference on Mathematical and Computer Modelling. University of California-Berkeley, 1993.

4. Dennis, Capt Scott T. Large Displacement and Rotational Formulation for Laminated Cylindrical Shells Including Parabolic Transverse Shear. PhD Disertation. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, May 1988. (AD-194871).

5. Engblom, J.J. and O.O. Ochoa "Finite Element Formulation Including Interlaminar Stress Calculations," Computers and Structures, 23-2: 241-249. Great Britain: Pergamon Press Ltd., 1986

6. -----. "Through the Thickness Stress Predictions for Laminated Plates of Advanced Composite Materials," International Journal for Numerical Methods in Engineering, 21: 1759-1776. New York: John Wiley and Sons, 1985.

7. Harrington, Capt Timberlyn M. An Investigation of Sandwich Flat Panels Under Low Velocity Impact. M.S. Thesis. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1994.

8. Hexcel Corporation, TSB 120: Mechanical Properties of Hexcel Honeycomb Materials: 2, 19. Dublin CA (No year).

9. Juvinall, R.C. and K.M. Marshek Fundamentals of Machine Component Design , 2nd Ed.: 322-327. New York: John Wiley and Sons, 1991.

10. Kelkar, A.D., W.J. Craft and R.S. Sandhu "Study of Progressive Damage in Thin and Thick Composite Laminates Subjected to Low-Velocity Impact Loading," Recent Advances in Structural Mechanics, PVP-259/NE-13: 67-73. ASME, 1993.

11. Korn, G.A. and T.M. Korn Mathematical Handbook for Scientists and Engineers, 2nd Ed.: 22-23. New York: McGraw Hill Book Co. (No year).

12. Lagace, P.A. and J.E. Williamson "Contributions of the Core and Facesheet to Damage Resistance of Composite Sandwich Panels," Proceedings of the Tenth DOD/NASA/FAA Conference on Fibrous Composites in Structural Design, Volume 1: 53-73. Warminster PA: NAWCADWAR, April 1994.

13. Math Works, The PRO-MATLAB Users Guide. South Natick PA: The Math Works, Inc., 1990.

14. McQuillan, E.J., L.W. Gause and R.E. Lorens "Low Velocity Transverse Normal Impact of Graphite Epoxy Composite Laminates," Journal of Composite Materials, 10. Lancaster PA: Technomic Publishing Co., 1976.

15. Nemes, J.A. and K.E. Simmonds "Low Velocity Impact Response of Foam-Core Sandwich Composites," Journal of Composite Materials, 26-4: 500-519. Lancaster PA: Technomic Publishing Co., 1992.

16. Owens, Capt Mark E. Finite Element Analysis of Composite Plates Including Shear Deformation. M.S. Thesis. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1988. (AAJ8857).

17. Owens, M.E., A.N. Palazotto and S.T. Dennis "Transverse Shear Flexibility in Laminated Composites," Computers and Structures, 45-1: 69-78. Great Britain: Pergamon Press Ltd., 1992.

18. Pagano, N.J. "Exact Solutions for Rectangular Bidirectional Composites and Sandwich Plates," Journal of Composite Materials, 4: 20-34. Lancaster PA: Technomic Publishing Co., 1970.

19. Palazotto, A.N. and S.T. Dennis Nonlinear Analysis of Shell Structures. Washington DC: AIAA, 1992.

20. Sadda, Adel S. Elasticity: Theory and Applications: 153-159. Malabar FL: R.E. Krieger Publishing Co., 1989.

21. Whitney, James M. Structural Analysis Of Laminated Anisotropic Plates: 296-311. Lancaster PA: Technomic Publishing Co., 1987.

22. Wolfram, Stephen Mathematica: A System for Doing Mathematics by Computer, 2nd Ed. Addison-Wesley Publishing Co., 1991.

23. Wu, E. and C.S. Yen "The Contact Behavior Between Laminated Composite Plates and Rigid Spheres," Journal of Applied Mechanics, 61-1: 60-66. Fairfield NJ: ASME, 1994.

## Appendix A:

## Modifications to Finite Element Code SHELL

The original version of SHELL was written by Dennis [4]. It is a FORTRAN code with the following components:

Table A.1: Components of SHELL code

| Component Name | Description |
|---|---|
| Main program SHELL | Preprocessor, solver and postprocessor |
| Subroutines: | |
| MESH | Automatically generates rectangular mesh |
| ELAST | Calculates constitutive relations and elasticity matrices |
| STIFF | Calculates element stiffness matrix and force vector |
| SHAPE | Evaluates shape functions at Gauss points |
| DIS | Evaluates displacement gradients at Gauss points |
| BNDY | Imposes prescribed boundary conditions |
| SOLVE | Solves linear equation systems |
| STRESS | Calculates stresses at outer Gauss points from displacement results |
| INCREMENT | Increments loads and/or displacements for nonlinear analysis |
| CONVERGE | Tests convergence of nonlinear analysis |
| PK | Calculates independent element stiffness matrix K for a plate |
| PN1 | N1 for a plate |
| PN2 | N2 for a plate |
| SK | K for a shell |
| SN1 | N1 for a shell |
| SN2 | N2 for a shell |
| STOP | Stops program if unable to converge itterations using Riks method |
| CHSIGN | Used in Riks method to allow backwards incrementing |

The code required three major modifications for use in this research:

1. Enhance the preprocessor to allow sandwich constructions
2. Restore load and displacement control options to the nonlinear solver
3. Generate a secondary output file for use with a separate initial-failure check program

Since the entire SHELL code is very large, only those components requiring extensive changes are listed at the end of this appendix. The other components remain unaltered or needed minor corrections in its nonexecutable statements (i.e. altering common variable blocks to make them consistent with the rest of the program). The new structure for SHELL's input deck is also included just before the code listings.

Sandwich Construction

The previous version of SHELL could model an isotropic material or a symmetric laminate consisting of plies of the same orthotropic material at different orientations. It also required each ply to have the same uniform thickness. The input deck for a laminate contained: the number of plies, the ply thickness, the orthotropic elastic properties and each ply's orientation. In order to model sandwich constructions, the code was altered to allow multiple sets of elastic properties and variable ply-to-ply thicknesses (although still uniform across a single ply). The new input deck for a laminate includes: the number of plies, the number of materials, an indicator for uniform or variable ply thickness, each material's orthotropic elastic properties, and each ply's orientation, material reference number, and thickness (or a single thickness value if uniform). Note that a sandwich containing isotropic materials can be modeled as a laminate by treating their elastic properties as orthotropic but numerically consistent with isotropic.

The use of multiple elastic property sets was implemented into the code by converting the single-value variables for $E_{LL}$, $E_{TT}$, $G_{LT}$, $G_{LZ}$, $G_{TZ}$ and $\nu_{LT}$ into one-dimensional arrays . Since plies can be at different orientations, the preprocessor was already designed to calculate a separate set of constitutive relations for each one. Therefore, all that was needed was a way to index the correct element in each elastic property array (corresponding to the ply's material reference number). This was done by creating a material-stacking-sequence (MSS) array similar to the preexisting orientation angle array. As the code cycles through each ply, it reads a new material number and angle and uses the former to load the proper elastic constants.

Enhancing the code to allow variable ply thicknesses was not essential, but it could greatly reduce redundancies in the input deck and calculations throughout the program. For example, a 3-ply laminate with ply thicknesses of 5, 36 and 5 units would otherwise require a 46-ply model with unit thickness per ply. The large variation between face and core thicknesses in typical sandwiches would amplify this redundancy. The old preprocessor used the uniform ply thickness to calculate the through-the-thickness (Z) coordinates of each ply at its upper, middle and lower surfaces. A modified method using a ply thickness array (similar in form to the MSS and orientation arrays) is now employed when variable thickness is indicated. The method involves simple step-by-step addition and is too elementary to warrant an in-depth explanation.

Some core materials have negligible stiffness in the in-plane directions. However, setting $E_{LL}=E_{TT}=0$ in the input deck will cause the program to crash. The error is due to a line that calculates $\nu_{TL}$ from the relation:

$$\upsilon_{TL} = \upsilon_{LT} \frac{E_{TT}}{E_{LL}} \tag{A.1}$$

To prevent division by zero, a precautionary step was added to the code which sets $\nu_{TL}=\nu_{LT}$ and skips Equation A.1 whenever $E_{LL}=E_{TT}$.

Load and Displacement Control

The original version of SHELL included both load and displacement control methods for the nonlinear solver. A later version included the Riks-Wemper method which allows a better description of a cylindrical shell's behavior when it undergoes snapping instability [18 ]. However, this research uncovered the fact that in adding the Riks method the other methods had been removed. Load or displacement control is usually more practical when modeling flat plates because plates do not tend to snap, and the Riks method normally does not increment the loads and/or displacements at regular intervals.

In converting the program to the Riks method, certain parts of the original code were deleted or turned into comments. Fortunately, another finite element program called ISHELL contains a processor unit nearly identical in format to SHELL but features the original load and displacement control instead of the Riks method. The process of enhancing SHELL to include all three methods, involved a line-by-line comparison and merging of both processor codes. An indicator was added to the input deck to trigger the use or disuse of the Riks method, and many if-then statements were added to the code to skip unwanted operations in either case.

Secondary Output File

In order to execute the separate initial-failure check program FAILURE, certain model-dependent information is needed from SHELL. The required data includes:

1. Model parameters from the input deck (an isotropic or laminate model, a linear or nonlinear solution, the number of elements, plies and materials and the elastic material properties)
2. Preprocessor calculations (the nodal coordinates, nodal connectivity of elements, constitutive relations, z-coordinates and thickness factor for transverse shear)
3. Nodal displacement results (for each increment if solution is nonlinear)

All of the above is written to a separate output file in a format easily read by FAILURE. A new indicator in the SHELL input deck allows the user to decide whether or not to generate the file.

Other Modifications

In addition to the aforementioned code changes, several other nonessential modifications were implemented to enhance the user-friendliness of the software. First, direct keyboard input was added to the beginning of the program to allow user-defined names for the input and output files. This alleviates the task of renaming or relocating old files before running a new model with default filenames. It also allows simultaneous execution of multiple models on a computer network without the need for extra copies of the program in separate file directories.

Second, the double-precision variables that contain nodal coordinate and displacement values were reformatted. Each value's scientific notation is now printed to the output files with an "E" (instead of a "D") to indicate the exponent. This data can be cut-and-pasted into separate files for use with commercial math or graphing software. However, it was discovered that some software packages do not recognize "D" as an acceptable substitute for "E" and will misread the data or generate a syntax error. The modified output format alleviated this problem.

Input Deck to SHELL

| Card | Variable | Type | Variable Description & Allowable Contents/Array Size | Notes |
|---|---|---|---|---|
| 1 | TITLE | String | Title of problem | |
| 2 | IEL | Integer | Element type: 1 for plate, 2 for cylindrical shell | |
| | NPE | Integer | Nodes per element: 4 or 8 | |
| | NANAL (*) | Integer | Analysis parameters: array (1 to 3) | |
| | | | NANAL(1): 0 for nonlinear, 1 for linear, 2 for eigenvalue | |
| | | | NANAL(2): 1 for isotropic, 2 for symmetric laminate | NANAL(2)=0 for arbitrary laminate (currently unavailable) |
| | | | NANAL(3): 0 for SLR, 1 for von Karman plate/ Donnell shell | |
| | IMESH | Integer | Mesh generation type: 0 for manual, 1 for automatic (rectangular) | |
| | NPRNT | Integer | Print elasticity matricies, element stiffness matricies and vectors? 0 for no, 1 for yes | |
| | NCUT | Integer | Number of elements to cut-out (if none enter 0) | |
| 2a | INTYP | Integer | Increment type: 0 for load control, 1 for displacement control | **INCLUDE card if NANAL(1)=0** |
| | NINC | Integer | Number of increments (or max. number of increments for Riks method) | |
| | IMAX | Integer | Maximum number of itterations per increment (21 typical) | |
| | IRES | Integer | Stiffness updates every itteration =0 | Stiffness updates every increment =1 (currently unavailable) |
| | TOL | Real | Percent tolerance for convergence (0.01 typical) | |
| | iriks | Integer | Indicator for using Riks method: 0 for no, 1 for yes | |
| 2b | TABLE(*) | Real | Multiplicative factors for non-Riks displacement control: array (1 to NINC) | **INCLUDE card if NANAL(1)=0 and INTYP=1 and iriks=0** |
| 2c | pincr | Real | (Riks) Initial load increment parameter (typical 0.1, 0.2, 0.02) | **INCLUDE card if NANAL(1)=0 and iriks=1** |
| | ttpi | Real | (Riks) Parameter for stopping load incr. (typical 1.0) | |
| | icontt | Integer | (Riks) Number of itterations for each load step for decreasing load step | |
| | nlcut | Integer | (Riks) Max number of times load increment is cut in half if no real roots are obtained: enter 0 for no increment cutting | |
| | nrestr | Integer | (Riks) Restart parameter: 0 for no restart/ no output, 1 for no restart/ output, N for restart from Nth load step | |
| | nstore | Integer | (Riks) Last step stored in file for restart | |
| 2d | RSTEP | Real | Step for eigenvalues | **INCLUDE card if NANAL(1)= 2** |
| 2e | NX | Integer | Number of element sudivisions in X-direction | **INCLUDE card if IMESH=1** |
| | NY | Integer | Number of element subdivisions in Y-direction | |
| | DX (*) | Real | Node spacing in X-direction: array (1 to NX*NPE/4) | |
| | DY (*) | Real | Node spacing in Y-direction: array (1 to NY*NPE/4) | |

| Card | Variable | Type | Variable Description & Allowable Contents/Array Size | Notes |
|---|---|---|---|---|
| 2f | NEM | Integer | Number of elements | **INCLUDE card if IMESH=0** |
| | NNM | Integer | Total number of nodes | |
| | NX | Integer | Number of element sudivisions in X-direction | |
| | NY | Integer | Number of element subdivisions in Y-direction | |
| | NOD (*,*) | Integer | Nodal connectivity matrix: array (1 to NEM, 1 to NPE) | |
| | X (*) | Real | Global node coordinates in X-direction: array (1 to NNM) | |
| | Y (*) | Real | Global node coordinates in Y-direction: array (1 to NNM) | |
| 2g | ICUT (*) | Integer | Element numbers cut-out: array (1 to NCUT) | **SKIP card if NCUT=0** |
| 3 | LD | Integer | Distributed load parameter: 0 for none, 1 for transverse normal, 2 for dead weight, 3 for axial, 4 for in-plane shear | LD=2 available for cylindrical shell only |
| | PO | Real | Distr. load intensity (or incremental intensity), (if LD=0 enter 0.0) | |
| 3a | NEDGE | Integer | Number of nodes with in-plane edge distr. loading | **INCLUDE card if LD=3 or 4** |
| | IEDGE (*) | Integer | Nodes with in-plane edge distr. loading: array (1 to NEDGE) Enter in ascending order | |
| 4 | NBDY1 | Integer | Number of nodes with specified geometric BCs | |
| 4a | NBOUND (*,*) | Integer | Nodal DOFs with specified geom. values: array (1 to NBDY1, 1 to 8) NBOUND (*,1)=node number NBOUND (*,2 to 8): 0 for free, 1 for specified | **SKIP card if NBDY=0** |
| | VBDY (*) | Real | Specified geom. values: array (1 to # of 1's in NBOUND (*,2 to 8)) List in order left to right then down | |
| 5 | NBSF | Integer | Number of nodal DOFs with specified natural BCs | |
| 5a | IBSF (*) | Integer | DOF numbers with specified nat. values: array (1 to NBSF) | **SKIP card if NBSF=0** |
| | VBSF (*) | Real | Specified nat. values (or incremental values) cooresponding to IBSF (*): ar | Incremental for non-Riks load control |
| 6a | EY | Real | Isotropic Young's modulus | **INCLUDE card if NANAL(2)=1** |
| | NU | Real | Isotropic Poisson's ratio | |
| | HT | Real | Isotropic plate/shell thickness | |
| 6b | NMAT | Integer | Number of laminate materials | **SKIP card if NANAL(2)=1** |

Input Deck to SHELL

| Card | Variable | Type | Variable Description & Allowable Contents/Array Size | Notes |
|---|---|---|---|---|
| 6c | E1 (*) | Real | Longitudinal Young's modulus | **SKIP card if NANAL(2)=1** |
| | E2 (*) | Real | Lateral Young's modulus | **REPEAT card for all material properties: each array (1 to NMAT)** |
| | G12 (*) | Real | Long-Lat shear modulus | |
| | NU12 (*) | Real | Long-Lat Poisson's ratio | If a laminate material is isotropic, make properties numerically consistent: E=E1=E2, G=G13=G23= (E/2)/(1+NU12). Make accurate to 5 significant figures |
| | G13 (*) | Real | Long-Transverse shear modulus | |
| | G23 (*) | Real | Lat-Trans shear modulus | |
| 6d | NP | Integer | Number of plies | **SKIP card if NANAL(2)=1** |
| | IUT | Integer | Uniform ply thickness? 0 for no, 1 for yes | |
| | THE (*) | Real | Ply orientation sequence in degrees: array (1 to NP) | If a ply is isotropic, enter arbitrary orientation value |
| 6e | MSS (*) | Integer | Material sequence: array (1 to NP), values between 1 and NMAT | **SKIP card if NANAL(2)=1 or NMAT=1** |
| 6f | THI (*) | Real | Thickness sequence: array (1 to NP) | **SKIP card if NANAL(2)=1 or IUT=1** |
| 6g | PT | Real | Uniform ply thickness | **SKIP card if NANAL(2)=1 or IUT=0** |
| 7a | RAD | Real | Cylindrical shell radius | **SKIP card if IEL=1** |
| 8 | NFOR | Integer | Number of DOFs to calculate equivalent nodal loads for | |
| 8a | IFOR (*) | Integer | DOF numbers for equiv. nodal load calculation: array (1 to NFOR) | **SKIP card if NFOR=0** |
| 9 | NSTRESS | Integer | Number of elements to calculate stresses for | |
| 9a | ISTRESS (*) | Integer | Element numbers for stress calculation: array (1 to NSTRESS) | **SKIP card if NSTRESS=0** |
| 10 | IFAIL | Integer | Generate primary input file for initial failure check: 0 for no, 1 for yes | **INCLUDE card if IEL=1 and NPE=4 and IMESH=1** |

```
C
C          C O M P U T E R   P R O G R A M    S H E L L
C   (STATIC GEOMETRICALLY NONLINEAR ANALYSIS OF
C    ORTHOTROPIC PLATES AND SHELLS)
C
C MODIFIED VERSION: MULTIPLE MATERIAL PROPERTY SETS, VARIABLE PLY-
C   BY-PLY THICKNESS, SECONDARY OUTFUT FILE FOR EXTERNAL
C   POSTPROCESSOR (FAILURE), USER-DEFINED FILENAMES, RESTORED LOAD/
C   DISPLACEMENT CONTROL OPTIONS
C MODIFICATIONS BY: 2LT DAMIN SILER (DECEMBER 1994)
C NOTE: CODE ALTERATIONS ARE SCATTERED THROUGHOUT THE MAIN PROGRAM
C   AND ITS SUBROUTINES.
C  ..................................................................
C  .                                                                .
C  .      D E S C R I P T I O N   O F   T H E   V A R I A B L E S   .
C  .                                                                .
C  .    CON(M,N)..CONSTITUTIVE MATRIX USED IN STRESS                .
C  .    CONS(M,N).   "          "     "   "   "                      .
C  .    ELD(M)....ELEMENT DISPLACEMENT VECTOR                       .
C  .    ELP(M)....ELEMENT FORCE VECTOR                              .
C  .    ELXY(M,N) J-TH COORDINATE OF ELEMENT NODE I (J=1,2)         .
C  .    GD(M).....GLOBAL DISPLACEMENT VECTOR                        .
C  .    GF(M).....GLOBAL FORCE VECTOR; SOLUTION VECTOR FROM 'SOLVE' .
C  .    GSTIF.....GLOBAL STIFNESS MATRIX (IN BANDED FORM)           .
C  .    GN........GLOBAL N1 MATRIX FOR BIFURCATION ANALYSIS         .
C  .              OR EQUILIBRIUM STIFFNESS, K+N1/2+N2/3 IN NL ANAL  .
C  .    HT........THICKNESS OF THE PLATE OR SHELL                   .
C  .    IBDY(M)...ARRAY OF SPECIFIED GLOBAL DISPLACEMENTS           .
C  .    IBSF(M)...ARRAY OF SPECIFIED NONZERO GLOBAL FORCES          .
C  .    IEDGE(M)..ARRAY OF NODES THAT HAVE EDGE LOADING             .
C  .    IFOR(M)...ARRAY OF DOF FOR EQUIVALENT FORCE CALCULATION     .
C  .    ISTRES(M).ARRAY OF ELEMENT NO. FOR STRESS OUTPUT            .
C  .    IEL.......INDICATOR FOR THE ELEMENT TYPE:                   .
C  .                   IEL=1, PLATE ELEMENT                          .
C  .                   IEL=2, SHELL ELEMENT                          .
C  .    IMAX......MAXIMUM NUMBER OF ITERATIONS FOR AN INCREMENT     .
C  .    INODE.....USED IN MESH, =1 FOR NPE=4, =2 FOR NPE=8          .
C  .    INTYP.....PARAMETERS FOR NONLINEAR INCREMENTATION           .
C  .              INTYP=0 LOAD INCREMENTATION                       .
C  .              INTYP=1 DISPLACEMENT INCREMENTATION               .
C  .    IRES......=0 OR 1 FOR UPDATE OR NO UPDATE OF STIFFNESS      .
C  .    K1........CONSTANT FROM KINEMATICS                          .
C  .    LD........INDICATOR FOR THE DISTRIBUTED APPLIED LOADING     .
C  .                   LD=0 ZERO DISTRIBUTED LOADING                 .
C  .                   LD=1 TRANSVERSE OR NORMAL PRESSURE            .
C  .                   LD=2 DEAD WEIGHT (SHELL)                      .
C  .                   LD=3 AXIAL LOADING                            .
C  .                   LD=4 IN-PLANE SHEAR LOADING                   .
C  .    MSID......ARRAY OF SORTED MIDSIDE NODE NUMBERS              .
C  .    NANAL(M)..ANALYSIS PARAMETERS                               .
C  .                   NANAL(1)=0,1,2 FOR NL,LIN,EIGEN               .
C  .                   NANAL(2)=0,1,2 FOR ARB,ISO,SYM                .
C  .                   NANAL(3)=1 FOR VON KARMAN PLATE ANALYSIS      .
C  .                           =1 FOR DONNELL SHELL ANALYSIS         .
C  .    NCOUNT....INCREMENT COUNT                                   .
C  .    NCMAX.....VALUE OF THE COLUMN-DIMENSION OF GSTIF            .
C  .    NEDGE.....NUMBER OF NODES THAT HAVE EDGE LOADING            .
C  .    NFOR......NUMBER OF DOF FOR EQUIVALENT FORCE CALCULATION    .
C  .    NINC......NUMBER OF INCREMENTS                              .
C  .    NMID......NUMBER OF MIDSIDE NODES                           .
C  .    NRMAX.....VALUE OF THE ROW-DIMENSION OF GSTIF               .
C  .    NOD(M,N)..CONNECTIVITY MATRIX                               .
C  .    NBDY......TOTAL NUMBER OF SPECIFIED DEGREES OF FREEDOM      .
C  .    NBSF......TOTAL NUMBER OF SPECIFIED NONZERO FORCES          .
C  .    NPE.......NODES PER ELEMENT, 4 OR 8                         .
C  .    NEM.......# OF ELEMENTS TOTAL                               .
C  .    NNM.......# OF NODES TOTAL                                  .
C  .    NX........# OF X ELEMENT SUBDIV FOR MESH                    .
C  .    NY........# OF Y ELEMENT SUBDIV FOR MESH                    .
C  .    NN........DOF PER ELEMENT, 28 OR 36                         .
C  .    NDF.......DOF PER CORNER NODES = 7                          .
C  .    NEQ.......NDF * 7                                           .
C  .    NSTRES....NUMBER OF ELEMENTS TO CALCULATE STRESS            .
C  .    PO........INTENSITY OF THE DISTRIBUTED LOAD                 .
C  .    PKT.......ELEMENT INDEP STIFFNESS MATRIX                    .
C  .    RAD,P1....CYL SHELL RADIUS, 1/RAD                           .
C  .    RSTEP.....USED IN BIFURCATION ALGORITHM                     .
C  .    STIF......ELEMENT STIFFNESS MATRIX                          .
C  .    VBDY......VALUES OF THE DISPLACEMENTS IN THE ARRAY IBDY     .
C  .    VBSF......VALUES OF THE SPECIFIED FORCES IN THE ARRAY IBSF  .
C  .    VFOR......VALUES OF THE CALCULATED EQUIVALENT FORCES        .
C  .    VPRES.....ARRAY OF ZEROES USED IN DISP INCREMENT NL ANALYSIS.
C  .    X,Y.......ARRAYS OF X AND Y-COORDINATES OF GLOBAL NODES     .
C  .    ZZ(M,I)...LOCATES Z VALUES, USED IN STRESS                  .
C  .*******************SUBROUTINE LIST****************************.
C  .    MESH......AUTOMATICALLY FORMS MESH                          .
C  .    ELAST.....CALCULATES ELASTICITY MATRICES, A-T AND AS-FS     .
C  .    STIFF.....CALCULATES ELEMENT STIFFNESS MATRIX AND FORCE VEC .
C  .    SHAPE.....EVALUATES SHAPE FCNS AT GAUSS PT (XI,ETA)         .
C  .    DIS.......EVALUATES DISP GRADIENT AT GAUSS PT (XI,ETA)      .
C  .    PK........ELEMENT INDEP STIFFNESS, K FOR PLATE              .
C  .    PN1.......ELEMENT INDEP STIFFNESS, N1 FOR PLATE             .
C  .    PN2.......ELEMENT INDEP STIFFNESS, N2 FOR PLATE             .
C  .    SK........ELEMENT INDEP STIFFNESS, K FOR SHELL              .
C  .    SN1.......ELEMENT INDEP STIFFNESS, N1 FOR SHELL             .
```

```
C  .   SN2.......ELEMENT INDEP STIFFNESS, N2 FOR SHELL           .
C  .   BNDY......IMPOSE SPECIFIED DOF                             .
C  .   SOLVE.....SOLVE LINEAR EQNS                                .
C  .   STRESS....SOLVES FOR STRESSES                              .
C  .   INCREMENT.INCREMENTS LOAD OR DISPLACEMENT OR BOTH          .
C  .   CONVERGE..TESTS FOR CONVERGENCE IN NL ANALYSIS             .
C  .   nrestr.... 0...no restart, no output for restart           .
C  .              1...no restart, have output for restart         .
C  .              N...start from Nth load step                    .
C  ...............................................................
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      character*64 inname,outname,fname1,plname
      CHARACTER*4 TITLE
C     DIMENSION BIF(),UL()
      DIMENSION GSTIF(5000,310),GF(5000),TITLE(20),ISTRES(200),
     X          GN(5000,310),BIF(1,1),gld(5000),gld0(5000),gld1(5000),
     #       gdis(5000),gsti00(5000,310),gf0(5000),gf1(5000),gd00(5000),
     $       npdof(1300),ntdof(1300)
      DIMENSION MSID(1300),NBOUND(350,8),IEDGE(50),D(3,3),IFOR(50),
     X          VFOR(50),VPRES(2300),ICUT(50),VPERM(2300),TABLE(100)
      COMMON/STF/ELXY(8,2),STIF(56,56),ELP(56),RAD,ELN(56,56)
      COMMON/MSH/NOD(1300,8),X(1300),Y(1300),DX(150),DY(150)
      COMMON/ELAS/NANAL(3),E1(5),E2(5),G12(5),NU12(5),PT,NP,
     X          A(3,3),B(3,3),DD(3,3),E(3,3),F(3,3),G(3,3),H(3,3),
     X          I(3,3),J(3,3),K(3,3),L(3,3),P(3,3),R(3,3),
     X          S(3,3),T(3,3),AS(2,2),DS(2,2),FS(2,2),G13(5),
     X          G23(5),EY,NU,HT,GS
      COMMON/ELAS2/RTHE(100),MSS(100),NMAT
      COMMON/DISP/ELD(56),Q(18)
      COMMON/STR/CON(6,100),CONS(3,100),ZZ(5,100)
      COMMON/INCREM/NBSF,IBSF(1000),VBSF(1000),NBDY,IBDY(2300),
     X             VBDY(2300),NINC,IMAX,INTYP,IRES,PO,NCOUNT
      COMMON/CONV/TOL,NCON,ICOUNT,GD(5000)
        common/tsai/RINIT, PVALUE
        dimension pstk(18,18), stk(18,18)
      common/riks/iriks


      DOUBLE PRECISION K1,I,J,K,L,NU,NU12(5),NU21(5),KS1,KS2
      EQUIVALENCE(D(1,1),DD(1,1))
C      EQUIVALENCE(VPRES(1),BIF(1,1))
C
      DATA NDF,NRMAX,NCMAX/7,5000,310/
      IALL=5000
C
C     ...............................................................
C     .                 P R E P R O C E S S O R   U N I T           .
C     .  INPUT DECK, ECHO, CALL MESH AND ELAST, SET UP FOR PROCESSOR  .
C     ...............................................................
C
      write(*,923)
      read(*,925)inname
      write(*,924)
      read(*,925)outname
      write(*,927)
      read(*,925)plname
      write(*,926)
      read(*,925)fname1
  926 FORMAT('WHAT IS YOUR FAILURE CHECK FILE #1 NAME(if appl)?')
  927 FORMAT('WHAT IS YOUR PLOT FILE NAME?')

      OPEN(unit=5,FILE=inname)
      OPEN(unit=6,FILE=outname)
      open(unit=7,file=plname)

      READ (5,260) TITLE
c
c..... nlcut = 0: no load increment cut if no real roots
c..... nlcut = n: load increment cut half at most n time
c.....               until real roots are obtained
c......nstore:  the last nstore step stored in data file for restart
c
      READ (5,*) IEL,NPE,NANAL(1),NANAL(2),NANAL(3),IMESH,NPRNT,NCUT
      iriks=0
      IF(NANAL(1).EQ.0)READ(5,*)INTYP,NINC,IMAX,IRES,TOL,iriks
      IF(NANAL(1).EQ.0 .AND. INTYP.EQ.1 .and. iriks.ne.1)
     X  READ(5,*)(TABLE(MM),MM=1,NINC)
      if(nanal(1).eq.0 .and. iriks.eq.1)then
        read(5,*)pincr,ttpi,icontt,nlcut,nrestr,nstore
        open(unit=10,file='restart')
      endif
      IF(NANAL(1).EQ.2)READ(5,*)RSTEP
      INODE=NPE/4
      IF(IMESH.EQ.1)GOTO 20
      READ (5,*) NEM,NNM,NX,NY
      DO 10 M=1,NEM
   10 READ (5,*) (NOD(M,N),N=1,NPE)
      READ (5,*) (X(M),Y(M),M=1,NNM)
      GOTO 30
   20 READ (5,*) NX,NY
      NX1=INODE*NX
      NY1=INODE*NY
      READ (5,*) (DX(M),M=1,NX1)
      READ (5,*) (DY(M),M=1,NY1)
C
      CALL MESH(NX,NY,NPE,NNM,NEM)
```

```
C
   30 IF(NCUT.EQ.0) GOTO 25
      READ(5,*)(ICUT(M),M=1,NCUT)
   25 READ (5,*) LD,PO
      IF(LD.LE.2)GOTO 31
      READ(5,*)NEDGE
      READ(5,*)(IEDGE(M),M=1,NEDGE)
   31 READ (5,*) NBDY1
      DO 32 II=1,NBDY1
   32 READ (5,*) (NBOUND(II,JJ),JJ=1,8)
      do 863 ii=1,nem
      do 861 jj=1,4
 861  npdof(nod(ii,jj))=7
      do 862 jj=5,npe
 862  npdof(nod(ii,jj))=2
 863  continue
      ntdof(1)=1
      do 864 ii=2,nnm
 864  ntdof(ii)=ntdof(ii-1)+npdof(ii-1)
      neq=ntdof(nnm)-ntdof(1)+npdof(nnm)
 434  NBDY=0
      DO 34 II=1,NBDY1
      kk=ntdof(nbound(ii,1))-1
      ii1=npdof(nbound(ii,1))+1
      DO 34 JJ=2,ii1
      IF(NBOUND(II,JJ).EQ.0)GOTO 34
      NBDY=NBDY+1
      IBDY(NBDY)=kk + (JJ-1)
   34 CONTINUE
      READ (5,*) (VBDY(M),M=1,NBDY)
      READ (5,*) NBSF
      IF(NBSF.EQ.0)GOTO 35
      READ (5,*) (IBSF(M),M=1,NBSF)
      DO 37 M=1,50
   37 VBSF(M)=0.
      READ (5,*) (VBSF(M),M=1,NBSF)
C
C OUTPUT THE DATA INPUT AND THE MESH INFORMATION AND CALL ELAST
C
   35 WRITE (6,260) TITLE
      if(iriks.eq.1)then
        WRITE (6,310) IEL,NPE,pincr,icontt
      else
        write(6,315)IEL,NPE
      endif
  315 format(/,1x,'ELEMENT TYPE(1=PLATE, 2=CYL SHELL)=',i2,5x,
     X  'NODES PER ELEMENT=',i2)
      WRITE (6,320) NEM,NNM,NDF
      WRITE(6,338)
      WRITE(6,337)
      DO 36 II=1,NBDY1
   36 WRITE(6,339)(NBOUND(II,JJ),JJ=1,8)
      WRITE (6,340) NBDY
      WRITE (6,270) (IBDY(M),M=1,NBDY)
      WRITE (6,300) (VBDY(M),M=1,NBDY)
      WRITE (6,350) NBSF
      WRITE (6,270) (IBSF(M),M=1,NBSF)
      WRITE (6,300) (VBSF(M),M=1,NBSF)
      WRITE (6,360)
      DO 60 II=1,NEM
   60 WRITE (6,270) II,(NOD(II,N),N=1,NPE)
      WRITE (6,375)
      WRITE (6,339)(ICUT(M),M=1,NCUT)
      WRITE (6,370)
      do 302 m = 1, nnm
  302 WRITE (6,301) m, X(M),Y(M)
      WRITE (6,459)
      IF(NANAL(1).NE.0)GOTO 53
      WRITE(6,463)
      WRITE(6,464)INTYP,NINC,IMAX,TOL
      write(6,466)iriks
  466 format(/,5x,'USE RIKS METHOD(1=YES, 0=NO)=',i2)
      IF(INTYP.EQ.0)WRITE(6,465)LD,PO
      IF(INTYP.EQ.1)WRITE(6,295)
      IF(INTYP.EQ.1 .and. iriks.ne.1)WRITE(6,300)
     X  (TABLE(MM),MM=1,NINC)
      GOTO 54
   53 WRITE (6,460)LD,PO
   54 WRITE(6,462)NEDGE
      WRITE(6,270)(IEDGE(M),M=1,NEDGE)
      IF(NANAL(2).EQ.1)NP=1
C
      CALL ELAST(NPRNT)
C
      IF(IEL.EQ.1)GOTO 65
      READ(5,*)RAD
      WRITE(6,470)RAD
   65 READ(5,*)NFOR
      IF(NFOR.EQ.0)GOTO 51
      READ(5,*)(IFOR(M),M=1,NFOR)
   51 READ(5,*)NSTRES
      IF(NSTRES.EQ.0)GOTO 55
      READ(5,*)(ISTRES(M),M=1,NSTRES)
C
C  CALCULATE PARAMETERS, MATRICES DEPENDING IF NPE=4 OR 8
C
```

```
 55  NN=20+2*NPE
     NCOR=(NX+1)*(NY+1)
     NMID=NNM-NCOR
     if(npe.eq.4) NEQ=nnm*ndf
     IF(NPE.EQ.4)GOTO 66
C
C FOR 8 NODED ELEMENTS, IE, NPE=8
C
C FORM MATRIX MSID=# OF UNIQUE MIDSIDE NODES FROM NOD(I,J)
C
     II=1
     DO 500 JJ=1,NEM
     DO 500 KK=1,4
     DO 502 I1=1,II-1
 502 IF(NOD(JJ,KK+4).EQ.MSID(I1))GOTO 500
     MSID(II)=NOD(JJ,KK+4)
     II=II+1
 500 CONTINUE
C ADD TO ARRAYS IBDY AND VBDY FOR NODES WITH ONLY U AND V DOF
c      DO 520 KK=1,NMID
c      KN=MSID(KK)*7
c      DO 520 JJ=0,4
c       IBDY(NBDY+KK*5-JJ)=KN-JJ
c  520 VBDY(NBDY+KK*5-JJ)=0.
c      NBDY=NBDY+NMID*5
  66 IF(LD.LT.3)GOTO 69
C
C APPLY UNIFORM X-DIR EDGE LOADING ALONG X=0 OR X=L TO VBSF, IBSF ARRAYS
C FOR LD=3 OR 4
C
     DO 68 JJ=1,NEDGE
  68 IBSF(JJ+NBSF)=ntdof(IEDGE(JJ))
     IF(NPE.EQ.8)GOTO 62
     DO 63 II=1,NY
     CP=PO*HT*DY(II)/2.
     VBSF(NBSF+II)=CP+VBSF(NBSF+II)
  63 VBSF(NBSF+II+1)=CP
     GOTO 61
  62 DO 67 II=1,NY
     CQ=PO*HT*DY(2*II-1)/3.
     VBSF(NBSF+2*II-1)=VBSF(NBSF+2*II-1)+CQ
     VBSF(NBSF+2*II)=4.*CQ
  67 VBSF(NBSF+2*II+1)=VBSF(NBSF+2*II+1)+CQ
  61 NBSF=NBSF+NEDGE
     WRITE(6,350)NBSF
     WRITE(6,270)(IBSF(M),M=1,NBSF)
     WRITE(6,300)(VBSF(M),M=1,NBSF)
C
C FOR NL DISP INCREMENT, STORE PRES DISP IN PERMANENT ARRAY VPERM
C
  69 IF(NANAL(1).NE.0) GOTO 71
     IF(INTYP.EQ.0)GOTO 71
     DO 72 II=1,NBDY
  72 VPERM(II)=VBDY(II)
C
C COMPUTE THE HALF BAND WIDTH
C
  71 NHBW=0
     DO 70 N=1,NEM
     kk1=nod(n,1)
     kk2=kk1
     DO 771 II=2,NPE
     kk1=min0(kk1,nod(n,ii))
 771 kk2=max0(kk2,nod(n,ii))
     nw=ntdof(kk2)-ntdof(kk1)+npdof(kk2)
  70 IF (NHBW.LT.NW) NHBW=NW
     WRITE (6,400) NHBW

C
C  CREATE PREPROCESSOR PART OF FAILURE CHECK INPUT FILE 'infail1'
C  UPGRADE AUGUST 1994
C  CURRENTLY LIMITED TO 28-DOF PLATE ELEMENTS IN RECT. MESH
C
     IF((IEL.NE.1).OR.(NPE.NE.4).OR.(IMESH.NE.1))THEN
       IFAIL=0
       GOTO 799
     ENDIF
     READ(5,*)IFAIL
     IF(IFAIL.EQ.0) GOTO 799
     open(UNIT=8,FILE=fname1)
     WRITE(8,260)TITLE
     WRITE(8,700)(NANAL(II),II=1,3)
 700 FORMAT(3(I4,1X))
     IF(NANAL(2).NE.1)WRITE(8,705)NP,NMAT
 705 FORMAT(2(I4,1X))
     K1=-4./(HT**2*3.)
     WRITE(8,710)K1
 710 FORMAT(D20.13)
     DO 715 II=1,NP
 715   WRITE(8,720)(CON(JJ,II),JJ=1,6)
 720 FORMAT(6(D20.13,1X))
     DO 725 II=1,NP
 725   WRITE(8,730)(CONS(JJ,II),JJ=1,3)
 730 FORMAT(3(D20.13,1X))
     DO 735 II=1,NP
       IF(NP.EQ.1)THEN
```

```
         WRITE(8,740)(ZZ(JJ,II),JJ=1,5)
        ELSE
         WRITE(8,745)(ZZ(JJ,II),JJ=1,3)
        ENDIF
  735 CONTINUE
  740 FORMAT(5(D12.6,1X))
  745 FORMAT(3(D12.6,1X))
      IF(NANAL(2).NE.1)THEN
        DO 750 II=1,NMAT
         WRITE(8,755)E1(II),E2(II),G12(II)
         WRITE(8,755)NU12(II),G13(II),G23(II)
  750   CONTINUE
  755   FORMAT(3(D20.13,1X))
        DO 760 II=1,NP
  760    WRITE(8,710)RTHE(II)
        DO 765 II=1,NP
  765    WRITE(8,770)MSS(II)
  770   FORMAT(I4)
      ENDIF
      WRITE(8,705)NEM,NNM
      DO 775 II=1,NNM
  775   WRITE(8,780)X(II),Y(II)
  780 FORMAT(2(D12.6,1X))
      DO 785 II=1,NEM
        IF(NPE.EQ.4)WRITE(8,790)(NOD(II,JJ),JJ=1,NPE)
        IF(NPE.EQ.8)WRITE(8,792)(NOD(II,JJ),JJ=1,NPE)
  785 CONTINUE
  790 FORMAT(4(I4,1X))
  792 FORMAT(8(I4,1X))
  799 CONTINUE
C
C     E N D  O F  D A T A  I N P U T  T O  T H E  P R O G R A M
C     ..................................................................
C     .                  P R O C E S S O R  U N I T                     .
C     ..................................................................
C
      NCON=0
      do 1132 ii=1,56
 1132 eld(ii)=0.0
      DO 1131 II=1,1000
 1131 VPRES(II)=0.0
      if(iriks.eq.1)then
        if(nrestr.gt.1) go to 1134
      endif
      DO 1130 II=1,NEQ
 1130 GD(II)=0.
      NCOUNT=1
      if(iriks.ne.1)goto 1211
      senerg=0
      difor0=0.0
      tpincr=0.0
      if(ncount.eq.1) go to 1211
 1134 continue

c...Read in restart data
      read(10,*) tpincr,pincr1,dss,detm2,ncount,icount
      read(10,*) (gd(ii),ii=1,neq)
c...

      if(nrestr.gt.1) go to 1209
 2990 do 2991 ii=1,neq
 2991 gd(ii)=gd00(ii)
      iicut=1
      dss=dss/2.0
 1209 if(iriks.ne.1)goto 1211
      if(iicut.eq.0) dss=dss*icontt/icount
      difor2=gd(ifor(1))
      difor1=difor0
      difor0=difor2
 1212 ncount=ncount+1
      do 1208 ii=1,neq
      gd00(ii)=gd(ii)
 1208 gld0(ii)=0.0
 1211 icount=1
 1210 continue
      if(ncon.eq.1 .and. iriks.ne.1)call increment

C
C FOR NL DISP INCREMENT, PRESRIBE DISP ACCORDING TO ARRAY TABLE
C
      IF(NANAL(1).NE.0)GOTO 1200
      IF(INTYP.EQ.0)GOTO 1200
      if(iriks.ne.1 .and. icount.ne.1)goto 1200
      DO 83 NTAB=1,NBDY
      if(iriks.eq.1)then
        VBDY(NTAB)=VPERM(NTAB)
      else
        if(ncount.gt.1)vbdy(ntab)=vperm(ntab)*
     X       (table(ncount)-table(ncount-1))
        if(ncount.eq.1)vbdy(ntab)=vperm(ntab)*table(1)
      endif
   83 continue
 1200 continue
C
C     INITIALIZE THE GLOBAL STIFFNESS MATRIX AND FORCE VECTOR
```

```
C
      DO 81 II=1,NEQ
      GFO(II)=0.0
      DO 81 JJ=1,NHBW
      GN(II,JJ)=0.0
   81 GSTIF(II,JJ)=0.0
C
C LOOP OVER ALL ELEMENTS, CALCULATE ELEMENTAL STIFFNESS MATRICES
C
      KCALL=0
 1220 DO 135 N=1,NEM
      IF(NCUT.EQ.0) GOTO 132
      DO 131 KINGKUTS=1,NCUT
  131 IF(N.EQ.ICUT(KINGKUTS))GOTO 135
  132 DO 90 II=1,NPE
      NI=NOD(N,II)
      ELXY(II,1)=X(NI)
      ELXY(II,2)=Y(NI)
      kk1=npdof(ni)
      kk=ntdof(ni)-1
      nni=7*(ii-1)
      DO 91 JJ=1,kk1
   91 ELD(nni+JJ)=GD(kk+JJ)
   90 CONTINUE
      KCALL=KCALL +1
      call STIFF(IEL,NPE,NN,PO,NCOUNT,N,K1,LD,ICOUNT,KCALL,nrestr,
     >  pstk,stk)
      IF (NPRNT.EQ.0) GO TO 110
      IF (KCALL.NE.1) GO TO 110
      IF (NCOUNT.NE.1) GOTO 110
      IF (ICOUNT.NE.1) GOTO 110
      WRITE (6,380)
      NNN=NN
      IF(NPE.EQ.8)NNN=56
      DO 100 II=1,NNN
  100 WRITE (6,300) (STIF(II,JJ),JJ=1,NNN)
      WRITE (6,410)
      WRITE (6,300) (ELP(II),II=1,NNN)
      WRITE (6,410)
  110 CONTINUE
C
C     ASSEMBLE ELEMENT STIFFNESS MATRICES TO GET GLOBAL STIFFNESS MATRIX
C
      DO 130 M=1,NPE
      kkm=nod(n,m)
      NR=ntdof(kkm)-1
      ndf1=npdof(kkm)
      m1=(m-1)*ndf
      DO 1301 II=1,NDF1
      NR=NR+1
      LL=m1+II
      if(iriks.eq.1)then
        GFO(NR) = GFO(NR) + ELP(LL)
      else
        gf(nr)=gf(nr)+elp(ll)
      endif
      DO 1302 KK=1,NPE
      kkm1=nod(n,kk)
      NCL=ntdof(kkm1)-1
      ndf2=npdof(kkm1)
      m2=(kk-1)*ndf
      DO 1303 JJ=1,NDF2
      MM=m2+JJ
      NC=NCL+JJ-NR+1
      IF (NC) 1303,1303,120
  120 GSTIF(NR,NC)=GSTIF(NR,NC)+STIF(LL,MM)
      IF(NANAL(1).NE.0)GOTO 1303
      IF(NCOUNT.EQ.1 .AND. ICOUNT.EQ.1) GOTO 1303
      GN(NR,NC)=GN(NR,NC)+ELN(LL,MM)
 1303 continue
 1302 continue
 1301 continue
  130 CONTINUE
  135 CONTINUE
C
C IMPOSE FORCE BOUNDARY CONDITIONS, IE, PRESCRIBED NONZERO FORCES
C
  138 IF(NBSF.EQ.0)GOTO 145
      DO 140 II=1,NBSF
      NB=IBSF(II)
      if(iriks.ne.1)then
        gf(nb)=gf(nb)+vbsf(ii)
        goto 140
      endif
      GFO(NB)=GFO(NB)+VBSF(II)
  140 continue
  145 if(iriks.ne.1)then
        if(nanal(1).ne.0)goto 1201
        if(icount.eq.1 .and. ncount.eq.1)goto 1201
        if(icount.eq.1 .and. intyp.eq.1)goto 1201
        goto 1104
      endif
      do 1996 ii=1,neq
 1996 gf(ii)=gf0(ii)
      do 1999 ii=1,neq
      do 1999 jj=1,nhbw
```

```
1999 gsti00(ii,jj)=gstif(ii,jj)
      CALL BNDY(NRMAX,NCMAX,
     X             NEQ,NHBW,GSTIF,GF,NBDY,IBDY,VBDY)
      write(6,*) ncount,icount
      call solve(nrmax,ncmax,neq,nhbw,gstif,gf,0,detm,detml)
      dss0=0.0
      do 141 ii=1,neq
      gdis(ii)=gf(ii)
 141  dss0=dss0+gf(ii)*gf(ii)
      IF(NANAL(1).NE.0)GOTO 146
      if(icount.ne.1) go to 144
      detm1=detm2
      detm2=detm
      write(6,2002) detm,detml
 2002 format(/2x,'detm =',f3.0,2x,'detml =',e20.8,
     #        2x,'tangent stiffness matrix'/)
c      DO 2003 II=1,NEQ
c      ADD=0.
c      DO 2004 KK=1,II-1
c      IF(II-KK+1 .GT. NHBW)GOTO 2004
c      ADD=ADD+GN(KK,II-KK+1)*GD(KK)
c 2004 CONTINUE
c      RES=0.
c      DO 2005 JJ=1,NHBW
c      IF(JJ+II-1 .GT. NEQ)GOTO 2005
c      RES=RES + GN(II,JJ)*GD(JJ+II-1)
c 2005 CONTINUE
c 2003 GF(II)=RES+ADD
      write(6,*) detm1,detm,gf(3)
      if(ncount.eq.1.and.detm.lt.0.0) pincr=-pincr
      if(ncount.eq.1) dss=pincr*dsqrt(dss0)
      if(ncount.ne.1) pincr= dss/dsqrt(dss0)*detm*detm1
     #                      *pincr1/dabs(pincr1)
      pincr1=pincr
c      dfor12=difor2-difor1
      prs=0.0
      do 142 ii=1,neq
 142  prs=prs+gf0(ii)*gld(ii)
      stifpa=pincr*prs
c      if(ncount.ne.1) pincr= dss/dsqrt(dss0)*detm
c      if(ncount.ne.1.and.detm.gt.0.0.and.dfor12.lt.0.0)
c     #       pincr=dss/dsqrt(dss0)
      write(6,*) pincr,dss,dss0,stifpa
C
C CALCULATE THE RESIDUAL FORCE VECTOR FOR NL ANALYSIS
C                 [GN]*{GD}
C
      do 1105 ii=1,neq
      gld(ii)=pincr*gdis(ii)
 1105 continue
 144  continue
      if(icount.eq.1) go to 1139
c      do 1997 ii=1,neq
c 1997 gf(ii)=gf0(ii)
 1104 DO 1100 II=1,NEQ
      ADD=0.
      DO 1125 KK=1,II-1
      IF(II-KK+1 .GT. NHBW)GOTO 1125
      ADD=ADD+GN(KK,II-KK+1)*GD(KK)
 1125 CONTINUE
      RES=0.
      DO 1110 JJ=1,NHBW
      IF(JJ+II-1 .GT. NEQ)GOTO 1110
      RES=RES + GN(II,JJ)*GD(JJ+II-1)
 1110 CONTINUE
      if(iriks.eq.1)then
        GF(II)=gf0(ii)*(pincr+tpincr)-RES-ADD
      else
        gf(ii)=gf(ii)-res-add
      endif
 1100 continue
      if(iriks.ne.1)goto 1201
C
C IMPOSE DISPLACEMENT BOUNDARY CONDITIONS[D
C
      IF(NANAL(1).NE.0)GOTO 1201
      IF(INTYP.EQ.0)GOTO 1201
      DO 85 NTAB=1,NBDY
      VBDY(NTAB)=VPERM(NTAB)*(tpincr+pincr)
 85   continue
 1201 continue
      if(iriks.ne.1)then
        if(icount.eq.1 .or. intyp.eq.0)call bndy
     X      (NRMAX,NCMAX,NEQ,NHBW,GSTIF,GF,NBDY,IBDY,VBDY)
        if(icount.ne.1 .and. intyp.eq.1)call bndy
     X      (NRMAX,NCMAX,NEQ,NHBW,GSTIF,GF,NBDY,IBDY,VPRES)
        call solve(NRMAX,NCMAX,NEQ,NHBW,GSTIF,GF,0,detm,detm1)
        goto 146
      endif
      CALL BNDY(NRMAX,NCMAX,
     X             NEQ,NHBW,GSTI00,GF,NBDY,IBDY,VBDY)
c     IF(ICOUNT.NE.1 .AND. INTYP.EQ.1)CALL BNDY(NRMAX,NCMAX,NEQ,
c    X                          NHBW,GSTIF,GF,NBDY,IBDY,VPRES)
C
C CALL SUBROUTINE 'SOLVE' TO SOLVE THE SYSTEM OF EQUATIONS
C          THE SOLUTION IS RETURNED IN GF
```

```
C
      write(6,*) icount,dss0,dss,pincr,tpincr
      CALL SOLVE (NRMAX,NCMAX,NEQ,NHBW,GSTIF,GF,1,detm,detml)
      a1=dss0
      a2=0.0
      a3=0.0
      do 147 ii=1,neq
      a2=a2+(gld(ii)+gf(ii))*gdis(ii)
  147 a3=a3+gf(ii)*(2.0*gld(ii)+gf(ii))
      d12=a2*a2-a1*a3
      write(6,*) d12,a1,a2,a3
      if(d12.lt.0.0.and.nlcut.eq.0) call stop(icount,ncount)
      if(d12.lt.0.0) go to 2990
      iicut=0
      dpinc1=(-a2+dsqrt(d12))/a1
      dpinc2=(-a2-dsqrt(d12))/a1
      thit1=0.0
      thit2=0.0
      do 148 ii=1,neq
      gld0(ii)=gld(ii)
      gld(ii)=gld(ii)+gf(ii)+dpinc1*gdis(ii)
      gld1(ii)=gld0(ii)+gf(ii)+dpinc2*gdis(ii)
      thit1=thit1+gld0(ii)*gld(ii)
      thit2=thit2+gld0(ii)*gld1(ii)
  148 continue
      write(6,*) thit1,thit2
      thi12=thit1*thit2
      if(thi12.gt.0.0) go to 149
      dpincr=dpinc1
      if(thit2.gt.0.0) call chsign(gld,gld1,dpincr,dpinc2,neq)
      go to 150
  149 dpib=-a3/(a2*2.0)
      dpin1=dabs(dpib-dpinc1)
      dpin2=dabs(dpib-dpinc2)
      dpincr=dpinc1
      if(dpin2.lt.dpin1) call chsign(gld,gld1,dpincr,dpinc2,neq)
  150 pincr=pincr+dpincr
      write(6,666) pincr,dpincr,dpinc1,dpinc2
  666 format(/1x,'pincr =',e15.6,2x,'dpincr =',e15.6,2x,2(e15.6))
C
C UPDATE THE TOTAL DISPLACEMENT VECTOR GD
C
 1139 continue
      do 1140 ii=1,neq
 1140 gd(ii)=gd(ii)+gld(ii) -gld0(ii)
      go to 1135
  146 DO 1141 II=1,neq
 1141 GD(II)=GD(II) + GF(ii)
C
C IF LINEAR OR BIFURCATION ANALYSIS, BRANCH
C
C FOR NONLINEAR ANALYSIS, CHECK FOR CONVERGENCE.  IF CONVERGED,
C OUTPUT DISPLACEMENTS AND STRESSES, IF NOT, RETURN FOR ANOTHER
C ITERATION
C
 1135 IF(NANAL(1).NE.0)GOTO 1120
      CALL CONVERGE(NEQ)
      IF(NCON.EQ.0)GOTO 1210
      GOTO 195
 1120 IF (NANAL(1).EQ.1)GOTO 195
C*********************************************************************
C                                                                    *
C               BIFURCATION ANALYSIS  (FOR NANAL(1) = 2)             *
C                                                                    *
C*********************************************************************
      NCOUNT=2
C REINITIALIZE GSTIF AND GF, AND INITIALIZE GN1
  610 DO 522 II=1,NEQ
      GF(II)=0.
      DO 522 JJ=1,NHBW
      GSTIF(II,JJ)=0.
  522 GN(II,JJ)=0.
      DO 555 N=1,NEM
      IF(NCUT.EQ.0)GOTO 552
      DO 551 KINGKUTS = 1,NCUT
  551 IF(N.EQ.ICUT(KINGKUTS))GOTO 555
  552 DO 524 II=1,NPE
      NI=NOD(N,II)
      ELXY(II,1)=X(NI)
      ELXY(II,2)=Y(NI)
      kk1=npdof(ni)
      kk=ntdof(ni)-1
      nni=7*(ii-1)
      DO 530 JJ=1,kk1
 530  ELD(nni+JJ)=GD(kk+JJ)
 524  CONTINUE
      call STIFF(IEL,NPE,NN,PO,NCOUNT,N,K1,LD,ICOUNT,KCALL,nrestr,
     >  pstk,stk)
C ASSEMBLE GSTIF AND GN
      DO 550 M=1,NPE
      kkm=nod(n,m)
      NR=ntdof(kkm)-1
      ndf1=npdof(kkm)
      m1=(m-1)*ndf
      DO 550 II=1,NDF1
      NR=NR+1
```

```
      LL=M1+II
      GF(NR) = GF(NR) + ELP(LL)
      DO 550 KK=1,NPE
      kkm1=nod(n,kk)
      NCL=ntdof(kkm1)-1
      ndf2=npdof(kkm1)
      m2=(kk-1)*ndf
      DO 550 JJ=1,NDF2
      MM=m2+JJ
      NC=NCL+JJ-NR+1
      IF (NC) 550,550,560
  560 GSTIF(NR,NC)=GSTIF(NR,NC)+STIF(LL,MM)
      GN(NR,NC)=GN(NR,NC) + ELN(LL,MM)
  550 CONTINUE
  555 CONTINUE
C COPY GF INTO GD FOR USE IN EIGENVECTOR CALCULATION
      DO 576 II=1,NEQ
  576 GD(II)=GF(II)
C ITERATION FOR BIFURCATION LOAD
      DO 570 II=1,100
      EIG=RSTEP*(II-1) +1.
      DO 580 JJ=1,NEQ
      DO 580 KK=1,NHBW
  580 BIF(JJ,KK)=GSTIF(JJ,KK)+EIG*GN(JJ,KK)
C SCREEN OUTPUT
      WRITE(*,1002)BIF(1,1)
 1002 FORMAT(1X,D20.13)
      FAC=BIF(1,1)
      DO 588 MM=1,NEQ
      DO 588 NN=1,NHBW
 588  BIF(MM,NN)=BIF(MM,NN)/FAC
      CALL BNDY(NRMAX,NCMAX,NEQ,NHBW,BIF,GF,NBDY,IBDY,VBDY)
C SWITCH FROM UPPER BANDED TO LOWER BANDED
      DO 582 LL=NEQ,1,-1
      K4=0
      K2=NEQ-LL
      DO 582 JJ=NHBW,1,-1
      K3=NEQ-K2-K4
      IF(K3.GT.0 .AND. K3 .LE. NEQ) GOTO 585
      BIF(LL,JJ)=0.
      GOTO 582
 585  BIF(LL,JJ)=BIF(K3,K4+1)
 582  K4=K4+1
      IA=IALL
      IU=IALL
      NCODI=NHBW-1
C
C      CALL LUDAPB(BIF,NEQ,NCODI,IA,UL,IU,D1,D2,IER)
C
      IF(LD.EQ.0)PO=VBSF(1)
      PCR=PO*EIG
C SCREEN OUTPUT
      WRITE(*,1000)II,EIG,PCR,D1,D2
 1000 FORMAT(1X,I3,'  ',F20.13,' ',D20.13,' ',D20.13,' ',D12.5)
      IF(IER.EQ.129)GOTO 575
  570 CONTINUE
  575 WRITE(6,945)
      IF(NANAL(3).EQ.1 .AND. IEL.EQ.1)WRITE(6,946)
      IF(NANAL(3).EQ.1 .AND. IEL.EQ.2)WRITE(6,947)
      WRITE(6,935)IER
      WRITE(6,950)II,RSTEP,EIG,PO
      WRITE(6,299)
      EIG1=RSTEP*(II-2)+1.
      IF(II.EQ.1)EIG1=0.
      PCR1=PO*EIG1
      WRITE(6,940)PCR1,PCR
C
C CALCULATE THE EIGENVECTOR
C
      DO 590 JJ=1,NEQ
      GF(JJ)=EIG*GD(JJ)
      DO 590 KK=1,NHBW
      BIF(JJ,KK)=GSTIF(JJ,KK)+EIG*GN(JJ,KK)
  590 CONTINUE
C FIND A NONPRESCRIBED DOF, SET IT EQUAL TO 1.0
      DO 592 II=1,NEQ
      DO 593 MM=1,NBDY
      IF(II.EQ.IBDY(MM))GOTO 592
  593 CONTINUE
      GOTO 595
  592 CONTINUE
  595 NBDY=NBDY+1
      IBDY(NBDY)=II
      VBDY(NBDY)=1.0D0
      WRITE(*,1020)II,NBDY,IBDY(NBDY),VBDY(NBDY)
 1020  FORMAT(1X,I4,2X,I4,2X,I4,2X,D12.5)
      CALL BNDY(NRMAX,NCMAX,NEQ,NHBW,BIF,GF,NBDY,IBDY,VBDY)
      CALL SOLVE(NRMAX,NCMAX,NEQ,NHBW,BIF,GF,0,detm,detml)
      DO 598 II=1,NEQ
  598 GD(II)=GF(II)
C
C END OF BIFURCATION ANALYSIS
C***********************************************************************
C                                                                      *
C             OUTPUT DISPLACEMENTS OR EIGENVECTOR                      *
C                                                                      *
```

```
C*************************************************************************
 195  WRITE(6,299)
      if(iriks.eq.1)then
        tpincr=tpincr+pincr
        write(6,599) tpincr
 599    format(/1x,'tpincr =',2x,e12.5)
      endif
      IF(NANAL(1).EQ.1) WRITE(6,962)
      IF(NANAL(1).EQ.0 .AND. NCOUNT.EQ.1)WRITE(6,964)
      IF(NANAL(1).EQ.2) WRITE(6,955)
      IF(NANAL(1).EQ.0) WRITE(6,963)NCOUNT,ICOUNT
      WRITE(6,957)

C  -----------
C  FOR FAILURE CHECK UPGRADE AUG 1994
      IF(IFAIL.NE.0)THEN
        IF(NANAL(1).NE.0)THEN
          NFCOUNT=1
        ELSE
          NFCOUNT=NCOUNT
        ENDIF
        WRITE(8,770)NFCOUNT
      ENDIF
C  ------------

      DO 600 KK=1,NNM
      jj=ntdof(kk)
      jj1=npdof(kk)-1

C  -------------
C  FOR FAILURE CHECK UPGRADE AUG 1994
      IF(IFAIL.EQ.0)GOTO 600
      WRITE(8,795)(GD(JJ+JJ2),JJ2=0,JJ1)
 795  FORMAT(7(D20.13,1X))
C  -------------

 600  WRITE(6,960)KK,(GD(JJ+jj2),JJ2=0,jj1)
C       PRINT*,'W = ',GD(3)
      IF(NANAL(1).EQ.2) STOP
C
C     ...............................................................
C     .                 P O S T P R O C E S S O R   U N I T         .
C     ...............................................................
C
C  COMPUTE EQUIVALENT NODAL FORCES AT NFOR NODES
C
      IF(NFOR.EQ.0)GOTO 198
C REINITIALIZE GSTIF AND GF, AND GN
      DO 1322 II=1,NEQ
      GF(II)=0.
      DO 1322 JJ=1,NHBW
      GSTIF(II,JJ)=0.
1322  GN(II,JJ)=0.
      DO 1335 N=1,NEM
      IF(NCUT.EQ.0) GOTO 1332
      DO 1331 KINGKUTS=1,NCUT
1331  IF(N.EQ.ICUT(KINGKUTS))GOTO 1335
1332  DO 1390 II=1,NPE
      NI=NOD(N,II)
      ELXY(II,1)=X(NI)
      ELXY(II,2)=Y(NI)
      kk1=npdof(ni)
      kk=ntdof(ni)-1
      nni=7*(ii-1)
      DO 1391 JJ=1,kk1
1391  ELD(nni+JJ)=GD(kk+JJ)
1390  CONTINUE
      call STIFF(IEL,NPE,NN,PO,NCOUNT,N,K1,LD,ICOUNT,KCALL,nrestr,
     >  pstk,stk)
C
C     ASSEMBLE ELEMENT STIFFNESS MATRICES TO GET GLOBAL STIFFNESS MATRIX
C
      DO 1330 M=1,NPE
      kkm=nod(n,m)
      NR=ntdof(kkm)-1
      ndf1=npdof(kkm)
      m1=(m-1)*ndf
      DO 1330 II=1,NDF1
      NR=NR+1
      LL=M1+II
      GF(NR) = GF(NR) + ELP(LL)
      DO 1330 KK=1,NPE
      kkm1=nod(n,kk)
      NCL=ntdof(kkm1)-1
      ndf2=npdof(kkm1)
      m2=(kk-1)*ndf
      DO 1330 JJ=1,NDF2
      MM=m2+JJ
      NC=NCL+JJ-NR+1
      IF (NC) 1330,1330,1320
1320  GSTIF(NR,NC)=GSTIF(NR,NC)+STIF(LL,MM)
      GN(NR,NC)=GN(NR,NC)+ELN(LL,MM)
1330  CONTINUE
1335  CONTINUE
C SET GN = GSTIF IF LINEAR ANALYSIS
      IF(NANAL(1).EQ.0)GOTO 1345
```

```
      DO 1340 II=1,NEQ
      DO 1340 JJ=1,NHBW
 1340 GN(II,JJ)=GSTIF(II,JJ)
C
C MULTIPLY GN*GD FOR DOF IFOR
C
 1345 DO 175 JJ=1,NFOR
      II=IFOR(JJ)
      ADD=0.
      DO 180 KK=1,II-1
      IF(II-KK+1.GT.NHBW)GOTO 180
      ADD=ADD+GN(KK,II-KK+1)*GD(KK)
  180 CONTINUE
      RES=0.
      DO 185 LL=1,NHBW
      IF(LL+II-1.GT.NEQ)GOTO 185
      RES=RES+GN(II,LL)*GD(LL+II-1)
  185 CONTINUE
  175 VFOR(JJ)=RES+ADD
      if(iriks.ne.1)goto 2008
      do 2006 ii=1,nfor
        jj = ifor(ii)
 2006 senerg=senerg+vfor(jj)*(gd(jj)-gd00(jj))
      write(6,2007) senerg
 2007 format(/2x,'strain energy =',e20.8/)
C
C OUTPUT NODAL FORCES
C
 2008 WRITE(6,352)NFOR
      DO 187 JJ=1,NFOR
  187 WRITE(6,354)IFOR(JJ),VFOR(JJ)
      VTOTAL=0.
      DO 189 JJ=1,NFOR
  189 VTOTAL=VTOTAL + VFOR(JJ)
      WRITE(6,355)VTOTAL
C      PRINT*,'P = ',VTOTAL
      write(7,2009) ncount,gd(ifor(1)),vfor(1),vtotal
 2009 format(2x,i8,3(e15.7))
C
C  COMPUTE STRESSES (AT THE GAUSS POINTS)
C
  198 IF(NSTRES.EQ.0)GOTO 1150
      DO 200 KK=1,NSTRES
      N=ISTRES(KK)
      DO 190 II=1,NPE
      NI=NOD(N,II)
      ELXY(II,1)=X(NI)
      ELXY(II,2)=Y(NI)
      kk1=npdof(ni)
      kk2=ntdof(ni)-1
      nni=7*(ii-1)
      DO 190 JJ=1,kk1
  190 ELD(nni+JJ)=GD(kk2+JJ)
      WRITE(6,455)N
      WRITE(6,450)
  200 CALL STRESS (NPE,NDF,IEL,ELXY,RAD,NP,K1,NANAL)
C
C START A NEW INCREMENT IF NONLINEAR ANALYSIS

c...Set up restart data
C
 1150 if(iriks.ne.1)goto 1152
      if(nrestr.eq.0) go to 1151
      ncoun1=ncount-nrestr
      ncoun2=(ncoun1-1)/nstore
      ncoun3=ncoun2*nstore+1
      if(ncoun1.eq.ncoun3) rewind 10
      write(10,*) tpincr,pincr,dss,detm,ncount,icount
      write(10,*) (gd(ii),ii=1,neq)
        write(10,99)
99      format('End of restart data')
c...

 1151 if(dabs(tpincr).ge.dabs(ttpi)) go to 1207
 1152 IF(NANAL(1).EQ.0 .AND. NCOUNT.LT.NINC)GOTO 1209
 1207 STOP
C
C                   F   O   R   M   A   T   S
C
  260 FORMAT (20A4)
  270 FORMAT (16I5)
  280 FORMAT (8D10.4)
  290 FORMAT (8D10.3)
  295 FORMAT (/,1X,'DISPLACEMENT INCREMENT TABLE')
  299 FORMAT(/)
  300 FORMAT (8(2X,D12.5))
  301 format (2x,i5,2(2x,e12.5))
  310 FORMAT (/,1X,'ELEMENT TYPE(1=PLATE, 2=CYL SHELL) =',I2,5X,' NODES
     1 PER ELEMENT=',I2,/,5x,
     >  'First load increment parameter in Riks, pincr =',f9.3,/,5x,
     >  'No. of ite. for a load step to decrease next'
     >  ' step, icontt =',i6)
  320 FORMAT (5X,'NUMBER OF ELEMENTS IN THE MESH =',I3,/,5X,
     1'NUMBER OF NODES IN THE MESH =',I4,/,5X,'DOF PER NODE =',I2,/)
  337 FORMAT(/,4X,'NODE   U    V    W   W-X  W-S PSI-X PSI-S')
  338 FORMAT(/,1X,'DISPLACEMENT BOUNDARY CONDITIONS, 1=PRESCRIBED,
```

```
      XO=FREE')
  339 FORMAT(4X,I4,1X,7(I3,2X))
  340 FORMAT (/,1X,'NUMBER OF PRESCRIBED DISPLACEMENTS=',I5,/,1X,'SPECIFI
     1ED DISPLACEMENT DOF AND THEIR VALUES FOLLOW:')
  350 FORMAT (/,1X,'NUMBER OF SPECIFIED FORCES=',I4,/,1X,'SPECIFIED FORC
     1E DEGREES OF FREEDOM AND THEIR SPECIFIED VALUES FOLLOW:')
  352 FORMAT (/,5X,'NUMBER OF EQUIVALENT NODAL FORCES OUTPUT= ',I4,
     1/,5X,'DOF',10X,'FORCE')
  354 FORMAT (5X,I4,4X,D12.5)
  355 FORMAT (/,5X,'P TOTAL = ',D12.5)
  360 FORMAT (/,1X,'BOOLEAN (CONNECTIVITY) MATRIX-NOD(I,J) ',/)
  370 FORMAT (/,1X,'COORDINATES OF THE GLOBAL NODES:',/)
  375 FORMAT (/,1X,'CUTOUTS',/,5X,'THE FOLLOWING ELEMENT NUMBERS ARE
     1 CUTOUT')
  380 FORMAT (/,1X,'ELEMENT STIFFNESS AND FORCE MATRICES:',/)
  390 FORMAT (120(';'),//)
  400 FORMAT (/,1X,'HALF BAND WIDTH OF GLOBAL STIFFNESS MATRIX - ',I5,/)
  410 FORMAT (//)
  420 FORMAT (/,5X,'TRANSVERSE DEFLECTION, W:',/)
  430 FORMAT (/,5X,'ELASTIC SLOPE, W-X:',/)
  440 FORMAT (/,5X,'ELASTIC SLOPE, W-S:',/)
  442 FORMAT (/,5X,'BENDING SLOPE, PSI-X:',/)
  444 FORMAT (/,5X,'BENDING SLOPE, PSI-S:',/)
  446 FORMAT (/,5X,'X DISPLACEMENT, U:',/)
  448 FORMAT (/,5X,'S DISPLACEMENT, V:',/)
  450 FORMAT(/,8X,'Z-COORD',5X,'X-COORD',5X,'S-COORD',5X,'SIGMA11 ',4X,
     1'SIGMA22 ',5X,'SIGMA12',5X,'SIGMA23',5X,'SIGMA13',/)
  455 FORMAT(1X,'ELEMENT ',I3)
  459 FORMAT(/,1X,'LOAD PARAMETER=1,2,3,4; NORMAL,DEADWT,AXIAL,SHEAR')
  460 FORMAT(/,1X,'LOAD PARAMETER =  ',I1,'   INTENSITY = ',D12.5)
  462 FORMAT(/,5X,'NUMBER OF NODES WITH IN-PLANE LOADING=',I5,/,5X,
     1'NODE NUMBERS:')
  463 FORMAT(/,1X,'NONLINEAR ANALYSIS PARAMETERS')
  464 FORMAT(/,5X,'INCREMENT LOAD (=0) OR DISP (=1) = ',I2,/,
     15X,'NUMBER OF INCREMENTS = ',I4,/,5X,'MAXIMUM ITERATIONS = ',
     2I4,/,5X,'CONVERGENCE TOLERANCE = ',D12.5)
  465 FORMAT(/,1X,'LOAD PARAMETER =  ',I1,'   INTENSITY STEP = ',D12.5)
  470 FORMAT(/,1X,'CIRCULAR CYL RADIUS = ',D12.5,/)
  923 FORMAT (' WHAT IS YOUR INPUT FILE NAME?')
  924 FORMAT (' WHAT IS YOUR OUTPUT FILE NAME?')
  925 FORMAT (A)
  935 FORMAT(1X,'IER=',I3)
  940 FORMAT(1X,D20.13,' < PCR < ',D20.13)
  945 FORMAT(1X,'RESULTS OF BIFURCATION ANALYSIS')
  946 FORMAT(1X,'VON KARMAN NONLINEARITY IN PLATE N1')
  947 FORMAT(1X,'DONNELL EQNS USED IN K AND N1')
  950 FORMAT(1X,'CNT=',I2,'  RSTEP= ',D12.7,'  EIG= ',D12.7,'
     XPO= ',D12.5)
  955 FORMAT(1X,'EIGENVECTOR')
  957 FORMAT(1X,'NODE',7X,'U',13X,'V',13X,'W',13X,'W-X',11X,'W-S',
     X       10X,'PSI-X',10X,'PSI-S')
  960 FORMAT(1X,I4,7(2X,E12.5))
  962 FORMAT(1X,'RESULTS OF LINEAR ANALYSIS')
  963 FORMAT(1X,'INCREMENT= ',I3,'  ITERATION= ',I3)
  964 FORMAT(1X,'RESULTS OF NONLINEAR ANALYSIS')
      END


C ------------------------------------------------------------------
      SUBROUTINE ELAST(NPRNT)
C.................................................................
C     THIS SUBROUTINE CALCULATES THE ELASTICITY MATRICES, A,B,DD,E,F
C     G,H,I,J,K,L,P,R,S,T,AS,DS,FS
C.................................................................
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION K1,I,J,K,L,NU,NU12(5),NU21(5),KS1,KS2
      COMMON/ELAS/NANAL(3),E1(5),E2(5),G12(5),NU12(5),PT,NP,
     X            A(3,3),B(3,3),DD(3,3),E(3,3),F(3,3),G(3,3),H(3,3),
     X            I(3,3),J(3,3),K(3,3),L(3,3),P(3,3),R(3,3),
     X            S(3,3),T(3,3),AS(2,2),DS(2,2),FS(2,2),G13(5),
     X            G23(5),EY,NU,HT,GS
      COMMON/ELAS2/RTHE(100),MSS(100),NMAT
      COMMON/STR/CON(6,100),CONS(3,100),ZZ(5,100)
      DIMENSION QBAR(3,3),QSBAR(2,2),D(3,3),DENOM(5)
      DIMENSION Q11(5),Q12(5),Q22(5),U1(5),U2(5),U3(5),U4(5),U5(5)
      DIMENSION THE(100),THI(100)
      EQUIVALENCE(D(1,1),DD(1,1))
C
C INITIALIZE THE ELASTICITY MATRICES
C
      DO 10 M=1,3
      DO 11 N=1,3
      A(M,N)=0.
      B(M,N)=0.
      D(M,N)=0.
      E(M,N)=0.
      F(M,N)=0.
      G(M,N)=0.
      H(M,N)=0.
      I(M,N)=0.
      J(M,N)=0.
      K(M,N)=0.
      L(M,N)=0.
      P(M,N)=0.
      R(M,N)=0.
      S(M,N)=0.
```

```
 11  T(M,N)=0.
 10  CONTINUE
     DO 15 M=1,2
     DO 16 N=1,2
     AS(M,N)=0.
     DS(M,N)=0.
 16  FS(M,N)=0.
 15  CONTINUE
     WRITE(6,896)
896  FORMAT(//)
     WRITE(6,897)
897  FORMAT(1X,'NANAL(1)=0,1,2 FOR NL,LIN,EIGEN')
     WRITE(6,895)
895  FORMAT(1X,'NANAL(2)=0,1,2 FOR ARB,ISO,SYM')
     WRITE(6,890)
890  FORMAT(1X,'NANAL(3)=1 FOR VON KARMAN PLATE OR DONNELL SHELL EQNS')
     WRITE(6,899)NANAL(1),NANAL(2),NANAL(3)
899  FORMAT(1X,'NANAL(1)= ',I1,'  NANAL(2)= ',I2,'  NANAL(3)= ',I2)
     WRITE(6,896)
     IF(NANAL(2).NE.1)GOTO 30
C
C  ISOTROPIC CASE
C
     READ(5,*)EY,NU,HT
     WRITE(6,901)
 901 FORMAT(1X,'THE FOLLOWING PROPERTIES WERE INPUT (E,NU,THICK)')
     WRITE(6,906)EY
     WRITE(6,906)NU
     WRITE(6,906)HT
     GS=EY/(2*(1+NU))
     DENO=1.-NU**2
     QBAR(1,1)=EY/DENO
     QBAR(1,2)=NU*EY/DENO
     QBAR(2,2)=QBAR(1,1)
     QBAR(2,1)=QBAR(1,2)
     QBAR(3,3)=GS
     QBAR(1,3)=0.
     QBAR(3,1)=0.
     QBAR(2,3)=0.
     QBAR(3,2)=0.
     QSBAR(1,1)=GS
     QSBAR(2,2)=GS
     QSBAR(1,2)=0.
     QSBAR(2,1)=0.
C FORM MATRICES CON,CONS,ZZ FOR STRESS SUBROUTINE
     CON(1,1)=QBAR(1,1)
     CON(2,1)=QBAR(1,2)
     CON(3,1)=QBAR(1,3)
     CON(4,1)=QBAR(2,2)
     CON(5,1)=QBAR(2,3)
     CON(6,1)=QBAR(3,3)
     CONS(1,1)=QSBAR(1,1)
     CONS(2,1)=QSBAR(1,2)
     CONS(3,1)=QSBAR(2,2)
     DO 18 II=1,5
 18  ZZ(II,1)=HT*(II-3)/4.
     DO 20 M=1,3
     DO 21 N=1,3
     A(M,N)=QBAR(M,N)*HT
     D(M,N)=QBAR(M,N)*HT**3/(3*2.**2)
     F(M,N)=QBAR(M,N)*HT**5/(5*2.**4)
     H(M,N)=QBAR(M,N)*HT**7/(7*2.**6)
     J(M,N)=QBAR(M,N)*HT**9/(9*2.**8)
     L(M,N)=QBAR(M,N)*HT**11/(11*2.**10)
     R(M,N)=QBAR(M,N)*HT**13/(13*2.**12)
 21  T(M,N)=QBAR(M,N)*HT**15/(15*2.**14)
 20  CONTINUE
     DO 25 M=1,2
     DO 26 N=1,2
     AS(M,N)=QSBAR(M,N)*HT
     DS(M,N)=QSBAR(M,N)*HT**3/(3*2.**2)
 26  FS(M,N)=QSBAR(M,N)*HT**5/(5*2.**4)
 25  CONTINUE
     GOTO 29
C
C  INPUT MATERIAL PROPERTIES, E1,E2,G12,NU12
C
 30  READ(5,*) NMAT
     DO 100 II=1,NMAT
 100 READ(5,*) E1(II),E2(II),G12(II),NU12(II),G13(II),G23(II)
     WRITE(6,500)NMAT
 500 FORMAT(/,1X,'NUMBER OF MATERIALS =',I3,/)
     WRITE(6,904)
 904 FORMAT(1X,'THE FOLLOWING PROPERTIES WERE INPUT (E1,E2,G12,NU12,
    XG13,G23)')
     DO 110 II=1,NMAT
     WRITE(6,505)II
     WRITE(6,906)E1(II)
     WRITE(6,906)E2(II)
     WRITE(6,906)G12(II)
     WRITE(6,906)NU12(II)
     WRITE(6,906)G13(II)
     WRITE(6,906)G23(II)
 110 CONTINUE
 505 FORMAT(/,2X,'MATERIAL #',I3)
 906 FORMAT(4X,D20.13)
```

```
      READ(5,*)NP,IUT
      READ(5,*) (THE(II),II=1,NP)
      WRITE(6,916)
  916 FORMAT(1X)
      WRITE(6,918)
  918 FORMAT(1X,'PLY ORIENTATION SEQUENCE')
      DO 35 II=1,NP
      WRITE(6,920)THE(II)
  920 FORMAT(2X,D20.13)
   35 CONTINUE
      IF(NMAT.NE.1) GOTO 120
      DO 115 II=1,NP
  115 MSS(II)=1
      GOTO 130
  120 READ(5,*)(MSS(II),II=1,NP)
      WRITE(6,510)
  510 FORMAT(/,1X,'MATERIAL NUMBER STACKING SEQUENCE')
  515 FORMAT(2X,I3)
      DO 125 II=1,NP
  125 WRITE(6,515) MSS(II)
  130 WRITE(6,916)
      IF(IUT.NE.1)GOTO 140
      READ (5,*)PT
      WRITE(6,922) PT
  922 FORMAT(1X,'UNIFORM PLY THICKNESS = ',D20.13)
      HT=PT*NP
      GOTO 180
  140 READ(5,*)(THI(II),II=1,NP)
      WRITE(6,520)
      WRITE(6,525)(THI(II),II=1,NP)
  520 FORMAT(1X,'VARIABLE PLY THICKNESS SEQUENCE')
  525 FORMAT(2X,D20.13)
      HT=0.
      DO 160 II=1,NP
  160 HT=HT+THI(II)
  180 WRITE(6,530)HT
  530 FORMAT(/,1X,'TOTAL LAMINATE THICKNESS = ',D20.13)

C
C CALCULATE REDUCED STIFFNESSES
C FOR LAMINATED ANISOTROPIC STRUCTURES
C
      DO 200 II=1,NMAT

      if (e1(ii).eq.e2(ii))then
        nu21(ii)=nu12(ii)
      else
        NU21(II)=E2(II)*NU12(II)/E1(II)
      endif
c above prevents program crashing if E1=E2=0

      DENOM(II)=1.-NU12(II)*NU21(II)
      Q11(II)=E1(II)/DENOM(II)
      Q12(II)=NU12(II)*E2(II)/DENOM(II)
      Q22(II)=E2(II)/DENOM(II)

C
C  CALCULATE INVARIANTS
C
      U1(II)=(3.*Q11(II)+3.*Q22(II)+2.*Q12(II)+4.*G12(II))/8.
      U2(II)=(Q11(II)-Q22(II))/2.
      U3(II)=(Q11(II)+Q22(II)-2.*Q12(II)-4.*G12(II))/8.
      U4(II)=(Q11(II)+Q22(II)+6.*Q12(II)-4.*G12(II))/8.
      U5(II)=(Q11(II)+Q22(II)-2.*Q12(II)+4.*G12(II))/8.
  200 CONTINUE

C**************************************************
C  CALCULATE THE ELASTICITY MATRICES               *
C                                                  *
C  REMEM THAT THE Z AXIS POINTS DOWN AS IN JONES   *
C  HOWEVER, THE FIRST PLY IS THE TOP PLY, IE,      *
C  THE PLY WITH THE MOST NEGATIVE Z !!!            *
C**************************************************
      DO 45 II=1,NP
   45 RTHE(II)=THE(II)*3.14159265/180.
      DO 50 KK=1,NP
      MN=MSS(KK)
      QBAR(1,1)=U1(MN)+U2(MN)*DCOS(2.*RTHE(KK))+U3(MN)*DCOS(4.*RTHE(KK))
      QBAR(1,2)=U4(MN)-U3(MN)*DCOS(4.*RTHE(KK))
      QBAR(2,2)=U1(MN)-U2(MN)*DCOS(2.*RTHE(KK))+U3(MN)*DCOS(4.*RTHE(KK))
      QBAR(1,3)=.5*U2(MN)*DSIN(2.*RTHE(KK))+U3(MN)*DSIN(4.*RTHE(KK))
      QBAR(2,3)=.5*U2(MN)*DSIN(2.*RTHE(KK))-U3(MN)*DSIN(4.*RTHE(KK))
      QBAR(3,3)=U5(MN)-U3(MN)*DCOS(4.*RTHE(KK))
      QBAR(2,1)=QBAR(1,2)
      QBAR(3,1)=QBAR(1,3)
      QBAR(3,2)=QBAR(2,3)
      QSBAR(1,1)=G23(MN)*DCOS(RTHE(KK))**2+G13(MN)*DSIN(RTHE(KK))**2
      QSBAR(2,2)=G13(MN)*DCOS(RTHE(KK))**2+G23(MN)*DSIN(RTHE(KK))**2
      QSBAR(1,2)=-(G23(MN)-G13(MN))*DCOS(RTHE(KK))*DSIN(RTHE(KK))
      QSBAR(2,1)=QSBAR(1,2)
C FORM MATRICES CON,CONS,ZZ FOR STRESS SUBROUTINE
      CON(1,KK)=QBAR(1,1)
      CON(2,KK)=QBAR(1,2)
      CON(3,KK)=QBAR(1,3)
      CON(4,KK)=QBAR(2,2)
      CON(5,KK)=QBAR(2,3)
```

```
      CON(6,KK)=QBAR(3,3)
      CONS(1,KK)=QSBAR(1,1)
      CONS(2,KK)=QSBAR(1,2)
      CONS(3,KK)=QSBAR(2,2)
      IF(IUT.NE.1) GOTO 310
      ZL=(KK*1. - NP*.5)*PT
      ZU=ZL-PT
      GOTO 315
  310 IF(KK.EQ.1) THEN
      ZU=-HT/2.
      ZL=ZU+THI(1)
      ELSE
      ZU=ZL
      ZL=ZL+THI(KK)
      ENDIF
  315 ZZ(1,KK)=ZU
      ZZ(2,KK)=(ZL+ZU)*.5
      ZZ(3,KK)=ZL
      IF(NP.NE.1)GOTO 57
      DO 55 II=1,5
   55 ZZ(II,1)=HT*(II-3)/4.
   57 DO 51 M=1,3
      DO 52 N=1,3
      A(M,N)=A(M,N) + QBAR(M,N)*(ZL-ZU)
      D(M,N)=D(M,N) + QBAR(M,N)*(ZL**3-ZU**3)/3.
      F(M,N)=F(M,N) + QBAR(M,N)*(ZL**5-ZU**5)/5.
      H(M,N)=H(M,N) + QBAR(M,N)*(ZL**7-ZU**7)/7.
      J(M,N)=J(M,N) + QBAR(M,N)*(ZL**9-ZU**9)/9.
      IF (NANAL(1).EQ.1) GOTO 40
      L(M,N)=L(M,N) + QBAR(M,N)*(ZL**11-ZU**11)/11.
      IF (NANAL(1).EQ.2) GOTO 40
      R(M,N)=R(M,N) + QBAR(M,N)*(ZL**13-ZU**13)/13.
      T(M,N)=T(M,N) + QBAR(M,N)*(ZL**15-ZU**15)/15.
   40 IF (NANAL(2).EQ.2) GOTO 52
      B(M,N)=B(M,N) + QBAR(M,N)*(ZL**2-ZU**2)/2.
      E(M,N)=E(M,N) + QBAR(M,N)*(ZL**4-ZU**4)/4.
      G(M,N)=G(M,N) + QBAR(M,N)*(ZL**6-ZU**6)/6.
      I(M,N)=I(M,N) + QBAR(M,N)*(ZL**8-ZU**8)/8.
      IF (NANAL(1).EQ.1) GOTO 52
      K(M,N)=K(M,N) + QBAR(M,N)*(ZL**10-ZU**10)/10.
      P(M,N)=P(M,N) + QBAR(M,N)*(ZL**12-ZU**12)/12.
      IF (NANAL(1).EQ.2) GOTO 52
      S(M,N)=S(M,N) + QBAR(M,N)*(ZL**14-ZU**14)/14.
   52 CONTINUE
   51 CONTINUE
      DO 60 M=1,2
      DO 61 N=1,2
      AS(M,N)=AS(M,N)+QSBAR(M,N)*(ZL-ZU)
      DS(M,N)=DS(M,N)+QSBAR(M,N)*(ZL**3-ZU**3)/3.
   61 FS(M,N)=FS(M,N)+QSBAR(M,N)*(ZL**5-ZU**5)/5.
   60 CONTINUE
   50 CONTINUE
C
C SET TO ZERO THOSE ENTRIES DUE TO ROUNDOFF ERROR
C
   29 DO 85 M=1,3
      DO 86 N=1,3
      IF(DABS(A(1,1)).GT.DABS(A(M,N)*1.D08))A(M,N)=0.
      IF(DABS(B(1,1)).GT.DABS(B(M,N)*1.D08))B(M,N)=0.
      IF(DABS(D(1,1)).GT.DABS(D(M,N)*1.D08))D(M,N)=0.
      IF(DABS(E(1,1)).GT.DABS(E(M,N)*1.D08))E(M,N)=0.
      IF(DABS(F(1,1)).GT.DABS(F(M,N)*1.D08))F(M,N)=0.
      IF(DABS(G(1,1)).GT.DABS(G(M,N)*1.D08))G(M,N)=0.
      IF(DABS(H(1,1)).GT.DABS(H(M,N)*1.D08))H(M,N)=0.
      IF(DABS(I(1,1)).GT.DABS(I(M,N)*1.D08))I(M,N)=0.
      IF(DABS(J(1,1)).GT.DABS(J(M,N)*1.D08))J(M,N)=0.
      IF(DABS(K(1,1)).GT.DABS(K(M,N)*1.D08))K(M,N)=0.
      IF(DABS(L(1,1)).GT.DABS(L(M,N)*1.D08))L(M,N)=0.
      IF(DABS(P(1,1)).GT.DABS(P(M,N)*1.D08))P(M,N)=0.
      IF(DABS(R(1,1)).GT.DABS(R(M,N)*1.D08))R(M,N)=0.
      IF(DABS(S(1,1)).GT.DABS(S(M,N)*1.D08))S(M,N)=0.
      IF(DABS(T(1,1)).GT.DABS(T(M,N)*1.D08))T(M,N)=0.
   86 CONTINUE
   85 CONTINUE
      DO 90 M=1,2
      DO 91 N=1,2
      IF(DABS(AS(1,1)).GT.DABS(AS(M,N)*1.D08))AS(M,N)=0.
      IF(DABS(DS(1,1)).GT.DABS(DS(M,N)*1.D08))DS(M,N)=0.
      IF(DABS(FS(1,1)).GT.DABS(FS(M,N)*1.D08))FS(M,N)=0.
   91 CONTINUE
   90 CONTINUE
      IF(NPRNT.EQ.0)RETURN
C
C  OUTPUT THE MATRICES
C
      WRITE(6,916)
      WRITE(6,950)
  950 FORMAT(1X,'A(I,J)=')
      DO 65 II=1,3
   65 WRITE(6,952)A(II,1),A(II,2),A(II,3)
      WRITE(6,916)
      WRITE(6,954)
  954 FORMAT(1X,'B(I,J)=')
      DO 66 II=1,3
   66 WRITE(6,952)B(II,1),B(II,2),B(II,3)
      WRITE(6,916)
```

```
      WRITE(6,956)
  956 FORMAT(1X,'D(I,J)=')
      DO 67 II=1,3
   67 WRITE(6,952)D(II,1),D(II,2),D(II,3)
      WRITE(6,916)
      WRITE(6,958)
  958 FORMAT(1X,'E(I,J)=')
      DO 68 II=1,3
   68 WRITE(6,952)E(II,1),E(II,2),E(II,3)
      WRITE(6,916)
      WRITE(6,960)
  960 FORMAT(1X,'F(I,J)=')
      DO 69 II=1,3
   69 WRITE(6,952)F(II,1),F(II,2),F(II,3)
      WRITE(6,916)
      WRITE(6,962)
962   FORMAT(1X,'G(I,J)= ')
      DO 70 II=1,3
   70 WRITE(6,952)G(II,1),G(II,2),G(II,3)
      WRITE(6,916)
      WRITE(6,964)
  964 FORMAT(1X,'H(I,J)= ')
      DO 71 II=1,3
   71 WRITE(6,952)H(II,1),H(II,2),H(II,3)
      WRITE(6,916)
      WRITE(6,966)
  966 FORMAT(1X,'I(I,J)= ')
      DO 72 II=1,3
   72 WRITE(6,952)I(II,1),I(II,2),I(II,3)
      WRITE(6,916)
      WRITE(6,968)
  968 FORMAT(1X,'J(I,J)= ')
      DO 73 II=1,3
   73 WRITE(6,952)J(II,1),J(II,2),J(II,3)
      WRITE(6,916)
      WRITE(6,970)
  970 FORMAT(1X,'K(I,J)= ')
      DO 74 II=1,3
   74 WRITE(6,952)K(II,1),K(II,2),K(II,3)
      WRITE(6,916)
      WRITE(6,972)
  972 FORMAT(1X,'L(I,J)= ')
      DO 75 II=1,3
   75 WRITE(6,952)L(II,1),L(II,2),L(II,3)
      WRITE(6,916)
      WRITE(6,974)
  974 FORMAT(1X,'P(I,J)= ')
      DO 76 II=1,3
   76 WRITE(6,952)P(II,1),P(II,2),P(II,3)
      WRITE(6,916)
      WRITE(6,976)
  976 FORMAT(1X,'R(I,J)= ')
      DO 77 II=1,3
   77 WRITE(6,952)R(II,1),R(II,2),R(II,3)
      WRITE(6,916)
      WRITE(6,978)
  978 FORMAT(1X,'S(I,J)= ')
      DO 78 II=1,3
   78 WRITE(6,952)S(II,1),S(II,2),S(II,3)
      WRITE(6,916)
      WRITE(6,980)
  980 FORMAT(1X,'T(I,J)= ')
      DO 79 II=1,3
   79 WRITE(6,952)T(II,1),T(II,2),T(II,3)
      WRITE(6,916)
      WRITE(6,982)
  982 FORMAT(1X,'AS(I,J)= ')
      DO 80 II=1,2
   80 WRITE(6,953)AS(II,1),AS(II,2)
      WRITE(6,916)
      WRITE(6,984)
  984 FORMAT(1X,'DS(I,J)= ')
      DO 81 II=1,2
   81 WRITE(6,953)DS(II,1),DS(II,2)
      WRITE(6,916)
      WRITE(6,986)
  986 FORMAT(1X,'FS(I,J)= ')
      DO 82 II=1,2
   82 WRITE(6,953)FS(II,1),FS(II,2)
  952 FORMAT(1X,3(D20.13,2X))
  953 FORMAT(1X,2(D20.13,2X))
      RETURN
      END

c-----------------------------------------------------------------------
      SUBROUTINE STIFF(IEL,NPE,NN,PO,NCOUNT,N,K1,LD,ICOUNT,KCALL,nrestr,
     >  pstk,stk)
C  8 JAN---- VERSION
C     ...................................................................
C     CALLED BY SHELL FOR EACH ELEMENT IN MESH. THE PROGRAM IS WRITTEN
C     FOR ORTHOTROPIC PLATES AND CYLINDRICAL SHELLS.  THE ELEMENT IS
C     BASED ON A HIGHER ORDER SHEAR-DEFORMABLE THEORY. THE FOUR NODE
C     ELEMENT HAS SEVEN DOF PER NODE (U,V,W,W1,W2,PSI1,PSI2).  THE
C     EIGHT NODE ELEMENT HAS 2 DOF AT EACH MIDSIDE NODE (U,V).
C
C     PSTK..........PLATE ELEMENT INDEP STIFFNESS, K
```

```
C     PSTN1.........PLATE ELEMENT INDEP STIFFNESS, N1
C     PSTN2.........PLATE ELEMENT INDEP STIFFNESS, N2
C     STK...........SHELL ELEMENT INDEP STIFFNESS, K
C     SN1...........SHELL ELEMENT INDEP STIFFNESS, N1
C     SN2...........SHELL ELEMENT INDEP STIFFNESS, N2
C     PKT...........ELEMENT INDEP INCREMENTAL STIFFNESS
C     STIF..........ELEMENT INCREMENTAL STIFFNESS
C     ELN...........ELEMENTAL INCREMENTAL STIFFNESS N1 FOR BIFURCATION
C                   ELEMENTAL EQUILIBRIUM STIFFNESS FOR NL ANALYSIS
C     ..............................................................
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION K1,I,J,K,L,NU,NU12(5),NU21(5),KS1,KS2
      COMMON/STF/ELXY(8,2),STIF(56,56),ELP(56),RAD,ELN(56,56)
      COMMON/SHP/SF(4),DSF(2,4),HRM(4,3),D1HRM(4,3),D2HRM(4,3),
     X          DD1HRM(4,3),DD2HRM(4,3),D12HRM(4,3),QSF(8),DQSF(2,8)
      COMMON/DISP/ELD(56),Q(18)
      COMMON/ELAS/NANAL(3),E1(5),E2(5),G12(5),NU12(5),PT,NP,
     X          A(3,3),B(3,3),DD(3,3),E(3,3),F(3,3),G(3,3),H(3,3),
     X          I(3,3),J(3,3),K(3,3),L(3,3),P(3,3),R(3,3),
     X          S(3,3),T(3,3),AS(2,2),DS(2,2),FS(2,2),G13(5),
     X          G23(5),EY,NU,HT,GS
      DIMENSION GAUSS(7,7),WT(7,7),PKT(36,36),PSTK(18,18),STK(18,18),
     X         PSTN1(18,18),PSTN2(18,18),STN1(18,18),STN2(18,18),
     X         DTK(36,18),MATRIX(6),PKN(18,18)
      common/riks/iriks

C
      DATA GAUSS/7*0.0D0,-.57735027D0,.57735027D0,5*0.0D0,-.77459667D0,0
     1.0D0,.77459667D0,4*0.0D0,-.86113631D0,-.33998104D0,.33998104D0,.86
     2113631D0,3*0.0D0,-.90617985D0,-.53846931D0,0.0D0,.53846931D0,.9061
     37985D0,2*0.0D0,-.93246951D0,-.66120939D0,-.23861919D0,.23861919D0,
     4.66120939D0,.93246951D0,0.0D0,-.94910791D0,-.74153119D0,-.40584515
     5D0,0.0D0,.40584515D0,.74153119D0,.94910791D0/
C
      DATA WT/2.0D0,6*0.0D0,2*1.0D0,5*0.0D0,.55555555D0,.88888888D0,.555
     155555D0,4*0.0D0,.34785485D0,2*.65214515D0,.34785485D0,3*0.0D0,
     2.23692689D0,.47862867D0,.56888888D0,.47862867D0,.23692689D0,2*0.0
     3D0,.17132449D0,.36076157D0,2*.46791393D0,.36076157D0,.17132449D0,
     40.0D0,.12948497D0,.27970539D0,.38183005D0,.41795918D0,.38183005D0,
     5.27970539D0,.12948497D0/
      DATA MATRIX/5,5,10,10,15,15/
C
C QUADRATURE ORDER = 4,5,7 FOR LINEAR, EIGEN, OR NL ANALYSIS
C
      NGP=2*NANAL(1)**2-5*NANAL(1)+7
      NGP=5
      IF(NCOUNT.EQ.1 .AND. ICOUNT.EQ.1)NGP=4
      NDF=7
C
C     INITIALIZE THE ELEMENT MATRICES AND FORCE VECTOR
C
      DO 10 II=1,56
      ELP(II)=0.0
      DO 10 JJ=1,56
      ELN(II,JJ)=0.0
   10 STIF(II,JJ)=0.0
      DO 12 II=1,36
      DO 12 JJ=1,18
   12 DTK(II,JJ)=0.
      K1=-4./(HT**2*3.)
       if(iriks.eq.1)then
        if(nrestr.le.1) go to 13
        ncoun1=ncount-nrestr
        if(ncoun1.gt.1) go to 134
        IF(ICOUNT.EQ.1 .AND. IEL.EQ.1 .AND. KCALL.EQ.1)
     X     CALL PK(PSTK,K1)
        IF(ICOUNT.EQ.1 .AND. IEL.EQ.2 .AND. KCALL.EQ.1)
     X     CALL SK(STK,K1,RAD)
  134   go to 133
   13   IF(NCOUNT.EQ.1 .AND. ICOUNT.EQ.1 .AND. IEL.EQ.1
     X    .AND. KCALL.EQ.1)CALL PK(PSTK,K1)
        IF(NCOUNT.EQ.1 .AND. ICOUNT.EQ.1 .AND. IEL.EQ.2
     X    .AND. KCALL.EQ.1)CALL SK(STK,K1,RAD)
       else
        if(iel.eq.1 .and. ncount.eq.1 .and. icount.eq.1
     X     .and. kcall.eq.1)call pk(pstk,k1)
        if(iel.eq.2 .and. ncount.eq.1 .and. icount.eq.1
     X     .and. kcall.eq.1)call sk(stk,k1,rad)
       endif
C
C      GAUSS QUADRATURE BEGINS HERE
C
  133 DO 800 NI=1,NGP
       DO 800 NJ=1,NGP
       XI=GAUSS(NI,NGP)
       ETA=GAUSS(NJ,NGP)
       CALL SHAPE (NPE,XI,ETA,ELXY,DET)
       CNST=WT(NI,NGP)*WT(NJ,NGP)*DET
C
C CALCULATE EQUIVALENT NODAL LOADING FOR DISTRIBUTED LOADS
C
C       RA=8.
C       RB=8.
       IF(LD.EQ.0 .OR. LD.GE.3)GOTO 35
       PI=3.14159265
       XX=0.
```

```
      YY=0.
      DO 24 II=1,NPE
      PIN=SF(II)
      IF(NPE.EQ.8)PIN=QSF(II)
      XX=XX+PIN*ELXY(II,1)
   24 YY=YY+PIN*ELXY(II,2)
      DO 20 II=1,NPE
      LL=(II-1)*NDF+1
C      RNUM1=PI*(XX+RA/2.)/RA
C      RNUM2=PI*(YY+RB/2.)/RB
C      RNUM1=PI*XX/RA
C      RNUM2=PI*YY/RB
C FOR CYL BEND STRIP = RA LENGTH
C      RLINT=PO*DSIN(RNUM1)
C FOR RECT PLATE RA X RB TOTAL
C      RLINT=PO*DSIN(RNUM1)*DSIN(RNUM2)
      IF(LD-2)26,27,20
   26 RLINT=PO
      GOTO 32
   27 QY=PO*DSIN(YY/RAD)
      PIN=SF(II)
      IF(NPE.EQ.8)PIN=QSF(II)
      ELP(LL+1)=ELP(LL+1)+CNST*QY*PIN
      RLINT=PO*DCOS(YY/RAD)
   32 IF(II.GT.4)GOTO 20
      DO 22 JJ=2,4
   22 ELP(LL+JJ)=ELP(LL+JJ)+CNST*RLINT*HRM(II,JJ-1)
   20 CONTINUE
C
   35 IF(IEL.EQ.2)GOTO 30
C***************************************
C          FOR FLAT PLATE              *
C***************************************
      IF(NANAL(1).EQ.1) GOTO 40
      IF(NCOUNT.EQ.1 .AND. ICOUNT.EQ.1)GOTO 42
      CALL DIS(NPE)
      CALL PN1(Q,PSTN1,K1)
C
      IF(NANAL(1).EQ.2)GOTO 40
      CALL PN2(Q,PSTN2,K1)
   40 IF(NANAL(1)-1) 44,42,46
   42 DO 90 II=1,18
      DO 90 JJ=1,18
   90 PKT(II,JJ)=PSTK(II,JJ)
      GOTO 48
   44 DO 100 II=1,18
      DO 100 JJ=1,18
      PKN(II,JJ)=PSTK(II,JJ)+PSTN1(II,JJ)/2.+PSTN2(II,JJ)/3.
  100 PKT(II,JJ)=PSTK(II,JJ)+PSTN1(II,JJ)+PSTN2(II,JJ)
      GOTO 48
   46 DO 110 II=1,18
      DO 110 JJ=1,18
      PKT(II,JJ)=PSTK(II,JJ)
      PKN(II,JJ)=PSTN1(II,JJ)
  110 CONTINUE
      GOTO 48
C***************************************
C          FOR CYL SHELL               *
C***********+***************************
   30 IF(NANAL(1).EQ.1)GOTO 52
      IF(NCOUNT.EQ.1 .AND. ICOUNT .EQ.1)GOTO 52
      CALL DIS(NPE)
      CALL SN1(Q,STN1,K1,RAD)
      IF(NANAL(1).EQ.2)GOTO 50
      CALL SN2(Q,STN2,K1,RAD)
   50 IF(NANAL(1)-1) 54,52,56
   52 DO 190 II=1,18
      DO 190 JJ=1,18
  190 PKT(II,JJ)=STK(II,JJ)
      GOTO 48
   54 DO 200 II=1,18
      DO 200 JJ=1,18
      PKN(II,JJ)=STK(II,JJ)+STN1(II,JJ)/2.+STN2(II,JJ)/3.
  200 PKT(II,JJ)=STK(II,JJ)+STN1(II,JJ)+STN2(II,JJ)
      GOTO 48
   56 DO 210 II=1,18
      DO 210 JJ=1,18
      PKT(II,JJ)=STK(II,JJ)
      PKN(II,JJ)=STN1(II,JJ)
  210 CONTINUE
C
C MULTIPLY DTKD=STIF
C
   48 DO 120 M=1,4
      II=M+4
      MMM=2*(M-1)
      DO 120 JJ=1,18
      DTK(7*M-1,JJ)=SF(M)*PKT(13,JJ)+DSF(1,M)*PKT(14,JJ)+DSF(2,M)*
     X              PKT(15,JJ)
      DTK(7*M,JJ)=SF(M)*PKT(16,JJ)+DSF(1,M)*PKT(17,JJ)+DSF(2,M)*
     X              PKT(18,JJ)
      DO 124 MM=2,4
      DTK(7*M-MM,JJ)=HRM(M,5-MM)*PKT(7,JJ)+D1HRM(M,5-MM)*PKT(8,JJ)+
     X              D2HRM(M,5-MM)*PKT(9,JJ)+DD1HRM(M,5-MM)*PKT(10,JJ)+
     X              DD2HRM(M,5-MM)*PKT(11,JJ)+D12HRM(M,5-MM)*PKT(12,JJ)
  124 CONTINUE
```

```
      IF(NPE.EQ.8)GOTO 122
      DTK(7*M-6,JJ)=SF(M)*PKT(1,JJ)+DSF(1,M)*PKT(2,JJ)+DSF(2,M)*
     X              PKT(3,JJ)
      DTK(7*M-5,JJ)=SF(M)*PKT(4,JJ)+DSF(1,M)*PKT(5,JJ)+DSF(2,M)*
     X              PKT(6,JJ)
      GOTO 120
  122 DTK(7*M-6,JJ)=QSF(M)*PKT(1,JJ)+DQSF(1,M)*PKT(2,JJ)+DQSF(2,M)*
     X              PKT(3,JJ)
      DTK(7*M-5,JJ)=QSF(M)*PKT(4,JJ)+DQSF(1,M)*PKT(5,JJ)+DQSF(2,M)*
     X              PKT(6,JJ)
      DTK(29+MMM,JJ)=QSF(II)*PKT(1,JJ)+DQSF(1,II)*PKT(2,JJ)+DQSF(2,II)*
     X              PKT(3,JJ)
      DTK(30+MMM,JJ)=QSF(II)*PKT(4,JJ)+DQSF(1,II)*PKT(5,JJ)+DQSF(2,II)*
     X              PKT(6,JJ)
  120 CONTINUE
      DO 88 M=1,4
      II=M+4
      MMM=7*(M-1)
      NUM1=1
      DO 88 JJ=1,NN
      JJJ=JJ
      IF(JJ.LE.30)GOTO 86
      JJJ=JJ+MATRIX(NUM1)
      NUM1=NUM1+1
   86 STIF(7*M-1,JJJ)=STIF(7*M-1,JJJ)+(SF(M)*DTK(JJ,13)+DSF(1,M)*
     X                DTK(JJ,14)+DSF(2,M)*DTK(JJ,15))*CNST
      STIF(7*M,JJJ)=STIF(7*M,JJJ)+(SF(M)*DTK(JJ,16)+DSF(1,M)*
     X              DTK(JJ,17)+DSF(2,M)*DTK(JJ,18))*CNST
      DO 84 MM=2,4
      STIF(7*M-MM,JJJ)=STIF(7*M-MM,JJJ)+(HRM(M,5-MM)*DTK(JJ,7)+
     X                D1HRM(M,5-MM)*DTK(JJ,8)+D2HRM(M,5-MM)*DTK(JJ,9)+
     X                DD1HRM(M,5-MM)*DTK(JJ,10)+DD2HRM(M,5-MM)*
     X                DTK(JJ,11)+D12HRM(M,5-MM)*DTK(JJ,12))*CNST
   84 CONTINUE
      IF(NPE.EQ.8)GOTO 82
      STIF(7*M-6,JJ)=STIF(7*M-6,JJ)+(SF(M)*DTK(JJ,1)+DSF(1,M)*
     X                DTK(JJ,2)+DSF(2,M)*DTK(JJ,3))*CNST
      STIF(7*M-5,JJ)=STIF(7*M-5,JJ)+(SF(M)*DTK(JJ,4)+DSF(1,M)*
     X                DTK(JJ,5)+DSF(2,M)*DTK(JJ,6))*CNST
      GOTO 88
   82 STIF(7*M-6,JJJ)=STIF(7*M-6,JJJ)+(QSF(M)*DTK(JJ,1)+DQSF(1,M)*
     X                DTK(JJ,2)+DQSF(2,M)*DTK(JJ,3))*CNST
      STIF(7*M-5,JJJ)=STIF(7*M-5,JJJ)+(QSF(M)*DTK(JJ,4)+DQSF(1,M)*
     X                DTK(JJ,5)+DQSF(2,M)*DTK(JJ,6))*CNST
      STIF(29+MMM,JJJ)=STIF(29+MMM,JJJ)+(QSF(II)*DTK(JJ,1)+DQSF(1,II)*
     X                DTK(JJ,2)+DQSF(2,II)*DTK(JJ,3))*CNST
      STIF(30+MMM,JJJ)=STIF(30+MMM,JJJ)+(QSF(II)*DTK(JJ,4)+DQSF(1,II)*
     X                DTK(JJ,5)+DQSF(2,II)*DTK(JJ,6))*CNST
   88 CONTINUE
      IF(NANAL(1).EQ.1)GOTO 800
      IF(NCOUNT.EQ.1 .AND. ICOUNT.EQ.1) GOTO 800
C
C MATRIX ELN USED IN BIFURCATION AND NONLINEAR ANALYSIS
C DO NOT CALCULATE IT ONLY IF NCOUNT AND ICOUNT = 1
C
      DO 250 NUT1=1,36
      DO 250 NUT2=1,18
  250 DTK(NUT1,NUT2)=0.
C
C MULTIPLY DTKD=ELN
C
      DO 320 M=1,4
      II=M+4
      MMM=2*(M-1)
      DO 320 JJ=1,18
      DTK(7*M-1,JJ)=SF(M)*PKN(13,JJ)+DSF(1,M)*PKN(14,JJ)+DSF(2,M)*
     X              PKN(15,JJ)
      DTK(7*M,JJ)=SF(M)*PKN(16,JJ)+DSF(1,M)*PKN(17,JJ)+DSF(2,M)*
     X              PKN(18,JJ)
      DO 324 MM=2,4
      DTK(7*M-MM,JJ)=HRM(M,5-MM)*PKN(7,JJ)+D1HRM(M,5-MM)*PKN(8,JJ)+
     X       D2HRM(M,5-MM)*PKN(9,JJ)+DD1HRM(M,5-MM)*PKN(10,JJ)+
     X       DD2HRM(M,5-MM)*PKN(11,JJ)+D12HRM(M,5-MM)*PKN(12,JJ)
  324 CONTINUE
      IF(NPE.EQ.8)GOTO 322
      DTK(7*M-6,JJ)=SF(M)*PKN(1,JJ)+DSF(1,M)*PKN(2,JJ)+DSF(2,M)*
     X              PKN(3,JJ)
      DTK(7*M-5,JJ)=SF(M)*PKN(4,JJ)+DSF(1,M)*PKN(5,JJ)+DSF(2,M)*
     X              PKN(6,JJ)
      GOTO 320
  322 DTK(7*M-6,JJ)=QSF(M)*PKN(1,JJ)+DQSF(1,M)*PKN(2,JJ)+DQSF(2,M)*
     X              PKN(3,JJ)
      DTK(7*M-5,JJ)=QSF(M)*PKN(4,JJ)+DQSF(1,M)*PKN(5,JJ)+DQSF(2,M)*
     X              PKN(6,JJ)
      DTK(29+MMM,JJ)=QSF(II)*PKN(1,JJ)+DQSF(1,II)*PKN(2,JJ)+
     X              DQSF(2,II)*PKN(3,JJ)
      DTK(30+MMM,JJ)=QSF(II)*PKN(4,JJ)+DQSF(1,II)*PKN(5,JJ)+
     X              DQSF(2,II)*PKN(6,JJ)
  320 CONTINUE
      DO 80 M=1,4
      II=M+4
      MMM=7*(M-1)
      NUM1=1
      DO 80 JJ=1,NN
      JJJ=JJ
      IF(JJ.LE.30)GOTO 386
```

```
      JJJ=JJ+MATRIX(NUM1)
      NUM1=NUM1+1
  386 ELN(7*M-1,JJJ)=ELN(7*M-1,JJJ)+(SF(M)*DTK(JJ,13)+DSF(1,M)*
     X                DTK(JJ,14)+DSF(2,M)*DTK(JJ,15))*CNST
      ELN(7*M,JJJ)=ELN(7*M,JJJ)+(SF(M)*DTK(JJ,16)+DSF(1,M)*
     X              DTK(JJ,17)+DSF(2,M)*DTK(JJ,18))*CNST
      DO 384 MM=2,4
      ELN(7*M-MM,JJJ)=ELN(7*M-MM,JJJ)+(HRM(M,5-MM)*DTK(JJ,7)+
     X                D1HRM(M,5-MM)*DTK(JJ,8)+D2HRM(M,5-MM)*DTK(JJ,9)+
     X                DD1HRM(M,5-MM)*DTK(JJ,10)+DD2HRM(M,5-MM)*
     X                DTK(JJ,11)+D12HRM(M,5-MM)*DTK(JJ,12))*CNST
  384 CONTINUE
      IF(NPE.EQ.8)GOTO 382
      ELN(7*M-6,JJ)=ELN(7*M-6,JJ)+(SF(M)*DTK(JJ,1)+DSF(1,M)*
     X                DTK(JJ,2)+DSF(2,M)*DTK(JJ,3))*CNST
      ELN(7*M-5,JJ)=ELN(7*M-5,JJ)+(SF(M)*DTK(JJ,4)+DSF(1,M)*
     X                DTK(JJ,5)+DSF(2,M)*DTK(JJ,6))*CNST
      GOTO 80
  382 ELN(7*M-6,JJJ)=ELN(7*M-6,JJJ)+(QSF(M)*DTK(JJ,1)+DQSF(1,M)*
     X                DTK(JJ,2)+DQSF(2,M)*DTK(JJ,3))*CNST
      ELN(7*M-5,JJJ)=ELN(7*M-5,JJJ)+(QSF(M)*DTK(JJ,4)+DQSF(1,M)*
     X                DTK(JJ,5)+DQSF(2,M)*DTK(JJ,6))*CNST
      ELN(29+MMM,JJJ)=ELN(29+MMM,JJJ)+(QSF(II)*DTK(JJ,1)+DQSF(1,II)*
     X                DTK(JJ,2)+DQSF(2,II)*DTK(JJ,3))*CNST
      ELN(30+MMM,JJJ)=ELN(30+MMM,JJJ)+(QSF(II)*DTK(JJ,4)+DQSF(1,II)*
     X                DTK(JJ,5)+DQSF(2,II)*DTK(JJ,6))*CNST
   80 CONTINUE
  800 CONTINUE
      RETURN
      END


c ----------------------------------------------------------------
      SUBROUTINE SOLVE(NRM,NCM,NEQNS,NBW,BAND,RHS,IRES,detm,detml)
C     ..........................................................
C     SOLVE A BANDED SYMMETRIC SYSTEM OF EQUATIONS
C     IN RESOLVING, IRES .GT. 0, LHS ELIMINATION IS SKIPPED
C     ..........................................................
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION BAND(NRM,NCM),RHS(NRM)
      common/riks/iriks

      MEQNS=NEQNS-1
      IF (IRES .GT. 0) PRINT*, ' IRES = ',IRES
      IF (IRES .GT. 0)  GO TO 90
      DO 500 NPIV=1,MEQNS
C      PRINT*,'NPIV= ',NPIV
      NPIVOT=NPIV+1
      LSTSUB=NPIV+NBW-1
      IF(LSTSUB.GT.NEQNS) LSTSUB=NEQNS
      DO 400 NROW=NPIVOT,LSTSUB
C     INVERT ROWS AND COLUMNS FOR ROW FACTOR
      NCOL=NROW-NPIV+1
      FACTOR=BAND(NPIV,NCOL)/BAND(NPIV,1)
      DO 200 NCOL=NROW,LSTSUB
      ICOL=NCOL-NROW+1
      JCOL=NCOL-NPIV+1
  200 BAND(NROW,ICOL)=BAND(NROW,ICOL)-FACTOR*BAND(NPIV,JCOL)
  400 RHS(NROW)=RHS(NROW)-FACTOR*RHS(NPIV)
  500 CONTINUE
      if(iriks.ne.1)goto 101
      detm=1.0
      detml=0.0
      do 600 ii=1,neqns
      detml=detml+dlog10(dabs(band(ii,1)))
  600 detm=detm*band(ii,1)/dabs(band(ii,1))
      GO TO 101
   90 DO 100 NPIV=1,MEQNS
      NPIVOT=NPIV+1
      LSTSUB=NPIV+NBW-1
      IF(LSTSUB.GT.NEQNS) LSTSUB=NEQNS
      DO 110 NROW=NPIVOT,LSTSUB
      NCOL=NROW-NPIV+1
      FACTOR=BAND(NPIV,NCOL)/BAND(NPIV,1)
  110 RHS(NROW)=RHS(NROW)-FACTOR*RHS(NPIV)
  100  CONTINUE
C      BACK SUBSTITUTION
  101 DO 800 IJK=2,NEQNS
      NPIV=NEQNS-IJK+2
      RHS(NPIV)=RHS(NPIV)/BAND(NPIV,1)
      LSTSUB=NPIV-NBW+1
      IF(LSTSUB.LT.1) LSTSUB=1
      NPIVOT=NPIV-1
      DO 700 JKI=LSTSUB,NPIVOT
      NROW=NPIVOT-JKI+LSTSUB
      NCOL=NPIV-NROW+1
      FACTOR=BAND(NROW,NCOL)
  700 RHS(NROW)=RHS(NROW)-FACTOR*RHS(NPIV)
  800 CONTINUE
      RHS(1)=RHS(1)/BAND(1,1)
      RETURN
      END
```

## Appendix B:

## Initial Failure Postprocessor Code FAILURE

FAILURE is a FORTRAN code for use with the updated version of SHELL. It is an enhanced postprocessor designed to predict initial failure of flat plates using maximum stress criteria. It also features a procedure for estimating maximum transverse normal stress magnitudes through a plate's thickness. It is limited to finite element plate models constructed from 28 degree-of-freedom elements in a regular rectangular mesh. Execution of FAILURE requires two input decks. The first is generated by SHELL for a given model, and the second is user-defined. The structure of the second deck and a sample of program output is contained in this appendix (just before the listing of the FAILURE code).

FAILURE consists of the main program and four subroutines: FSTRESS, PARFIT, SHAPE and DIS. The latter two are identical to those used by SHELL (see Appendix A, Table A.1). In addition, FSTRESS is a modified version of SHELL's own STRESS subroutine. The main difference between the two is that FSTRESS does not place stress calculations in temporary variables for immediate reporting to the output file. Instead, it stores the values for a single element and solution increment in arrays, so they can be mathematically manipulated. Finally, PARFIT is a set of linear algebra computations used to curve fit discrete stress and Z-coordinate data to parabolic distributions. This is needed for stress averaging through a ply's thickness when checking material failure and for evaluating $\sigma_{zz}$ by enforcing equilibrium.

The program gives the user the option of checking for failure in a single element or all of them. The same option applies to load or displacement control increments (one or all) in a nonlinear solution. All criteria stresses should be entered as absolute values since negative stresses are accounted for when the program checks for failure. Furthermore, any of the material or shear delamination failure modes can be turned off by entering a value 0.0 for its maximum stress value.

When the algorithm for evaluating $\sigma_{zz}$ (denoted $\sigma_{33}$ in some parts of the code) is activated, the user must define a single Z-coordinate to identify the plate surface (on or between -h/2 and h/2) at which values are to be calculated. In addition, a nonzero shear-to-moment ratio may be entered in order to limit $\sigma_{zz}$ calculations to elements in which transverse shear stresses are significant compared to in-plane bending stresses. For this particular research, shear-moment ratios were not employed (although they were initially proposed which led to their inclusion in the code).

User-defined Input Deck to FAILURE

| Card | Variable | Type | Variable Description & Allowable Contents/Array Size | Notes |
|---|---|---|---|---|
| 1 | TITLE2 | String | Title of user-defined input deck | |
| 2 | NFEL | Integer | Element number to check (enter 0 for all) | |
| | NFINC | Integer | Increment number to check (enter 0 for all) | Linear solution only one increment |
| 2a | NINC | Integer | Total number of increments with solutions | **INCLUDE card if NFINC=0** |
| 3 | IRST | Integer | Report averaged and transformed stresses when failure detected (1 for yes, 0 for no) | |
| 4 | FST(*,*) | Real | Material failure max stress magnitudes (positive values)<br>isotropic plate: single value uniaxial ST (tension)<br>orthotropic laminate: array (1 to #materials ,1 to 7)<br>S1T, S1C, S2T, S2C, S12MX, S23MX, S13MX | To ignore any failure mode, enter a magnitude of 0.0<br><br>1-2-3 are Long-Lat-Trans directions. If a laminate material is isotropic, enter uniaxial ST for S1T and set others =0.0 |
| 5 | SLAMX | Real | Shear delamination failure max stress in X-direction | To ignore any failure mode, enter a magnitude of 0.0 |
| | SLAMY | Real | Same in Y-direction | |
| 6 | I33 | Integer | Estimate maximum magnitude of transverse normal stress (1=yes, 0=no) | |
| 6a | SHMT | Real | Shear/Moment threshold ratio below which program skips calculating of sigmaZZ. SHMT compared to maximum abs value of (peak sigmaXZ / peak sigma XX) or (peak sigmaYZ / sigmaYY) for a given element. | **INCLUDE card if I33=1**<br>Enter 0.0 to ignore threshold |
| | Z33 | Real | Postition through the thickness to evaluate sizmaZZ at | Must be between -h/2 and h/2 (where h is the total plate thickness) |

## Sample Program Output

```
Qtr sandwich  case2  24x24mesh  4 ply face  r=.2  lo load
Sandwich model  case2  4ply lo load  sig33 at top of plate

INITIAL FAILURE CHECK

  ELEMENT TO CHECK (0 FOR ALL):    1
  INCREMENT TO CHECK(0 FOR ALL):  15
  REPORT AVERAGED AND TRANSFORMED STRESSES
      FOR FAILED REGIONS (1 FOR YES,0 FOR NO): 1

  NUMBER OF MATERIALS=   3
  NUMBER OF PLIES=  11

  MATERIAL FAILURE MODES/CRITERIA
    ORTHOTROPIC (MAX STRESS)
      1-LONGITUDINAL (FIBER) TENSION
      2-LONGITUDINAL (FIBER) COMPRESSION
      3-LATERAL (MATRIX) TENSION
      4-LATERAL (MATRIX) COMPRESSION
      5-LONG/LAT IN-PLANE SHEAR
      6-LAT/Z TRANSVERSE SHEAR
      7-LONG/Z TRANSVERSE SHEAR
    ISOTROPIC (MAX SHEAR STRESS)
      1-UNIAXIAL TENSION
      2-UNIAXIAL COMPRESSION
      3-SHEAR

  CRITERIA VALUES (MAGNITUDES)
    MATERIAL # 1  (ORTHOTROPIC)
      MODE 1:0.292393E+06
      MODE 2:0.202778E+06
      MODE 3:0.826100E+04
      MODE 4:0.357770E+05
      MODE 5:0.258000E+05
      MODE 6:0.206400E+05
      MODE 7:0.258000E+05
    MATERIAL # 2  (ORTHOTROPIC)
      MODE 1:0.000000E+00
      MODE 2:0.000000E+00
      MODE 3:0.000000E+00
      MODE 4:0.000000E+00
      MODE 5:0.000000E+00
      MODE 6:0.300000E+03
      MODE 7:0.515000E+03
    MATERIAL # 3  (ISOTROPIC)
      MODE 1:0.157800E+05
      MODE 2:0.157800E+05
      MODE 3:0.789000E+04

  INTER-PLY SHEAR DELAMINATION
    X-DIRECTION (13):0.206400E+05
    Y-DIRECTION (23):0.206400E+05

  ESTIMATE MAX (MAGNITUDE) TRANSVERSE NORMAL
      STRESS (1 FOR YES,0 FOR NO): 1
    MINIMUM (ABS VALUE) SIGMA23/SIGMA22 OR
        SIGMA13/SIGMA11 RATIO FOR TRANSVERSE
        NORMAL ESTIMATE= 0.000000E+00
```

```
INITIAL FAILURE RESULTS

INCREMENT #  15

 ELEMENT #    1

   PLY #   6  MATERIAL # 2
     MATERIAL FAILURE MODES
       6

------AVERAGED PLY STRESSES (X,Y,S11,S22,S12,S23,S13)
       .25446E-02  .25446E-02  0.000000E+00  0.000000E+00  0.000000E+00  -.950280E+01  -.119206E+02
       .25446E-02  .97455E-01  0.000000E+00  0.000000E+00  0.000000E+00  -.352938E+03  -.107059E+02
       .97455E-01  .25446E-02  0.000000E+00  0.000000E+00  0.000000E+00  -.873531E+01  -.438665E+03
       .97455E-01  .97455E-01  0.000000E+00  0.000000E+00  0.000000E+00  -.323595E+03  -.392255E+03
------TRANSFORMED PLY STRESSES (X,Y,SLL,STT,SLT,STZ,SLZ)
       .25446E-02  .25446E-02  0.000000E+00  0.000000E+00  0.000000E+00  -.950280E+01  -.119206E+02
       .25446E-02  .97455E-01  0.000000E+00  0.000000E+00  0.000000E+00  -.352938E+03  -.107059E+02
       .97455E-01  .25446E-02  0.000000E+00  0.000000E+00  0.000000E+00  -.873531E+01  -.438665E+03
       .97455E-01  .97455E-01  0.000000E+00  0.000000E+00  0.000000E+00  -.323595E+03  -.392255E+03

   SHEAR/MOMENT RATIO MAGNITUDES (X,Y,S23MAX/S22MAX,S13MAX/S11MAX)
     .25446E-02  .25446E-02  0.167694E-02  0.113155E-02
     .25446E-02  .97455E-01  0.658002E-01  0.110972E-02
     .97455E-01  .25446E-02  0.166019E-02  0.438142E-01
     .97455E-01  .97455E-01  0.651023E-01  0.428504E-01
   MAX (MAGNITUDE) TRANS NORMAL STRESS (X,Y,ESTIMATE)
     .25446E-02  .25446E-02  -.612563E+04
     .25446E-02  .97455E-01  -.579547E+04
     .97455E-01  .25446E-02  -.588080E+04
     .97455E-01  .97455E-01  -.555064E+04
```

```
C  FAILURE -INITIAL FAILURE POSTPROCESSOR SUPPLEMENT TO 'SHELL'
C  AUTHOR: LT. DAMIN SILER, AFIT, GAE-94D
C  THESIS ADVISOR: DR. ANTHONY PALAZOTTO
C
C  THIS PROGRAM RUNS SEPARATELY FROM 'SHELL' BUT REQUIRES
C    AN INPUT FILE ('INFAIL1') CREATED BY AN UPDATED VERSION
C    OF 'SHELL'.  A SECOND INPUT FILE ('INFAIL2') IS USER DEFINED
C    AND CONTAINS PROGRAM PARAMETERS AND FAILURE CRITERIA VALUES
C  LIMITED TO 28-DOF FLAT PLATE ELEMENTS IN A RECTANGULAR
C    MESH ('SHELL' VARIABLE CONSTRAINTS: IEL=1,NPE=4,NDF=7,
C    IMESH=1,P1=0.)
C  'SHELL' SUBROUTINES STRESS(MODIFIED),DIS, AND SHAPE ARE USED


      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION K1,NU12(5)
      CHARACTER*80 TITLE1,TITLE2
      CHARACTER*64 INFILE1,INFILE2,OUTFILE
      COMMON/SHP/SF(4),DSF(2,4),HRM(4,3),D1HRM(4,3),D2HRM(4,3),
     X           DD1HRM(4,3),DD2HRM(4,3),D12HRM(4,3),QSF(8),DQSF(2,8)
      COMMON/DISP/ELD(56),Q(18)
      COMMON/FAIL/IEL,NPE,NDF,P1,NANAL(3),NP,K1,CON(6,100),
     X           CONS(3,100),ZZ(5,100),SG11(4,300),SG22(4,300),
     X           SG12(4,300),SG23(4,300),SG13(4,300),SG232(4,300),
     X           SG131(4,300),I33
      DIMENSION ELXY(8,2),GPXY(4,2),GD(5000),X(1300),Y(1300),
     X           NOD(1300,8),FST(5,7),E1(5),E2(5),G12(5),
     X           G13(5),G23(5),RTHE(100),MSS(100),ISOPLY(100),
     X           NMODE(100),MFAIL(100,7),ASG11(4,100),ASG22(4,100),
     X           ASG12(4,100),ASG23(4,100),ASG13(4,100),DSG23(4,99),
     X           DSG13(4,99),TSG1(4,100),TSG2(4,100),TSG3(4,100),
     X           TSG4(4,100),TSG5(4,100),ISDFAIL(99,2),SG33(4),
     X           XXX(5),Y232(5),Y131(5),A232(4,100),B232(4,100),
     X           C232(4,100),A131(4,100),B131(4,100),C131(4,100),
     X           Y11(5),Y22(5),Y12(5),Y23(5),Y13(5),ISOMAT(5),
     X           SR2322(4),SR1311(4),NPDOF(1300),NTDOF(1300),
     X           XUM(3),XML(3),YUM11(3),YUM22(3),YUM12(3),YML11(3),
     X           YML22(3),YML12(3)


      DATA IEL,NPE,NDF,P1/1,4,7,0./

C  CHECK COMPLETENESS OF NON-EXEC STATEMENTS BEFORE COMPILING
C    ESPECIALLY DIMENSIONS


C

C  LOAD INPUT FILES
C

      WRITE(*,901)
      READ(*,904)INFILE1
      WRITE(*,902)
      READ(*,904)INFILE2
      WRITE(*,903)
      READ(*,904)OUTFILE
  901 FORMAT('NAME OF PRIMARY INPUT FILE (FROM SHELL):')
  902 FORMAT('NAME OF SECONDARY INPUT FILE (USER DEF):')
  903 FORMAT('NAME OF OUTPUT FILE:')
  904 FORMAT(A)
      OPEN(UNIT=8,FILE=INFILE1)
      OPEN(UNIT=9,FILE=INFILE2)
      OPEN(UNIT=11,FILE=OUTFILE)

c     code addition to write sigma33 results to separate
c     output file for graphical usage
      open(unit=14,file='s33est')
c

      READ(8,900)TITLE1
  900 FORMAT(A)
      READ(8,*)(NANAL(I),I=1,3)
      NP=1
      NMAT=1
      IF(NANAL(2).NE.1)READ(8,*)NP,NMAT
      READ(8,*)K1
      DO 10 I=1,NP
   10   READ(8,*)(CON(J,I),J=1,6)
      DO 20 I=1,NP
   20   READ(8,*)(CONS(J,I),J=1,3)
      NUM=3
      IF(NP.EQ.1)NUM=5
      DO 30 I=1,NP
   30   READ(8,*)(ZZ(J,I),J=1,NUM)

C             NOTE: CON, CONS & ZZ STORED TRANSPOSED

      IF(NANAL(2).EQ.1)GOTO 38
      DO 35 I=1,NMAT
        READ(8,*)E1(I),E2(I),G12(I)
   35   READ(8,*)NU12(I),G13(I),G23(I)
      READ(8,*)(RTHE(J),J=1,NP)
      READ(8,*)(MSS(J),J=1,NP)

   38 READ(8,*)NEM,NNM
```

```
      DO 40 I=1,NNM
   40   READ(8,*)X(I),Y(I)
      DO 50 I=1,NEM
   50   READ(8,*)(NOD(I,J),J=1,NPE)

      READ(9,900)TITLE2
      READ(9,*)NFEL,NFINC
      IF(NANAL(1).EQ.1)NINC=1
      IF(NFINC.EQ.0)READ(9,*)NINC
      READ(9,*)IRST
      NCRIT=7
      IF(NANAL(2).EQ.1)NCRIT=1
      DO 60 I=1,NMAT
   60   READ(9,*)(FST(I,J),J=1,NCRIT)
      I33=0
      IF(NP.GT.1)THEN
        READ(9,*)SLAMX,SLAMY
        READ(9,*)I33
        IF(I33.EQ.1)READ(9,*)SHMT,Z33
      ENDIF

C  FIND START AND FINISH INDEXES FOR LOOPS
C  FOR SINGLE INCREMENT, SKIP DISP. DATA PRIOR TO DESIRED SET

      NDFT=NDF*NNM

C  (INCLUDE CALCULATION HERE FOR NDFT IF NPE=8
C   WHEN FEATURE AVAILABLE)


      INCST=NFINC
      INCFN=NFINC
      IF(NFINC.EQ.0)THEN
        INCST=1
        INCFN=NINC
      ENDIF
      NELST=NFEL
      NELFN=NFEL
      IF(NFEL.EQ.0)THEN
        NELST=1
        NELFN=NEM
      ENDIF
      IF(NFINC.GT.1)THEN
        DO 70 I=1,NFINC-1
          READ(8,*)NDUMMY
   70     READ(8,*)(GD(J),J=1,NDFT)
      ENDIF

C  SET UP PLY TYPE AND NUMBER OF FAILURE MODE INDICATORS
C  CHECK LAMINATE PLIES FOR ELASTIC PROPERTIES NUMERICALLY
C    CONSISTENT WITH AN ISOTROPIC MATERIAL
C  ACCOUNT FOR G12=0 TO PREVENT CRASHING

      DO 100 I=1,NMAT
        ISOMAT(I)=0
        IF(NANAL(2).EQ.1)THEN
          ISOMAT(I)=1
        ELSE
          TOLISO=.0001
          IF(E1(I).EQ.E2(I) .AND.
     X       G12(I).EQ.G13(I) .AND.
     X       G13(I).EQ.G23(I) .AND.
     X       G12(I).NE.0.)THEN
            GG=E1(I)/(2.*(1.+NU12(I)))
            GDEV=DABS((G12(I)-GG)/G12(I))
            IF(GDEV.LE.TOLISO)ISOMAT(I)=1
          ENDIF
        ENDIF
  100 CONTINUE
      DO 110 I=1,NP
        ISOPLY(I)=0
        NMODE(I)=7
        IF(NANAL(2).EQ.1)THEN
          MAT=1
        ELSE
          MAT=MSS(I)
        ENDIF
        IF(ISOMAT(MAT).EQ.1)THEN
          ISOPLY(I)=1
          NMODE(I)=3
        ENDIF
  110 CONTINUE

C  INITIALIZE OUTPUT FILE ('OUTFAIL') TEXT

      WRITE(11,900)TITLE1
      WRITE(11,900)TITLE2
      WRITE(11,1010)
      WRITE(11,1015)NFEL
      WRITE(11,1020)NFINC
      IF(NFEL.EQ.0)WRITE(11,1025)NEM
      IF(NFINC.EQ.0)WRITE(11,1030)NINC
      WRITE(11,1035)IRST
      WRITE(11,1040)NMAT
      WRITE(11,1050)NP
```

```
      WRITE(11,1055)
      WRITE(11,1060)
      WRITE(11,1065)
      WRITE(11,1070)
      WRITE(11,1075)
      WRITE(11,1080)
      WRITE(11,1085)
      WRITE(11,1090)
      WRITE(11,1100)
      WRITE(11,1105)
      WRITE(11,1110)
      WRITE(11,1115)
      WRITE(11,1120)
      WRITE(11,1125)
      DO 120 I=1,NMAT
        IF(ISOMAT(I).EQ.0)THEN
          WRITE(11,1130)I
        ELSE
          WRITE(11,1132)I
          FST(I,2)=FST(I,1)
          FST(I,3)=FST(I,1)/2.
        ENDIF
        DO 125 J=1,NMODE(I)
  125     WRITE(11,1135)J,FST(I,J)
  120 CONTINUE
      WRITE(11,1005)
      IF(NP.GT.1)THEN
        WRITE(11,1140)
        WRITE(11,1145)SLAMX
        WRITE(11,1150)SLAMY
      ENDIF
      WRITE(11,1155)I33
      WRITE(11,1160)SHMT
      WRITE(11,1165)

 1005 FORMAT(/)
 1010 FORMAT(/,'INITIAL FAILURE CHECK',/)
 1015 FORMAT('  ELEMENT TO CHECK (0 FOR ALL):',I4)
 1020 FORMAT('  INCREMENT TO CHECK(0 FOR ALL):',I4)
 1025 FORMAT('    NUMBER OF ELEMENTS=',I4)
 1030 FORMAT('    NUMBER OF INCREMENTS=',I4)
 1035 FORMAT('  REPORT AVERAGED AND TRANSFORMED STRESSES',/,
     X     '      FOR FAILED REGIONS (1 FOR YES,0 FOR NO):',I2,/)
 1040 FORMAT('  NUMBER OF MATERIALS=',I4)
 1050 FORMAT('  NUMBER OF PLIES=',I4,/)
 1055 FORMAT('  MATERIAL FAILURE MODES/CRITERIA')
 1060 FORMAT('    ORTHOTROPIC (MAX STRESS)')
 1065 FORMAT('      1-LONGITUDINAL (FIBER) TENSION')
 1070 FORMAT('      2-LONGITUDINAL (FIBER) COMPRESSION')
 1075 FORMAT('      3-LATERAL (MATRIX) TENSION')
 1080 FORMAT('      4-LATERAL (MATRIX) COMPRESSION')
 1085 FORMAT('      5-LONG/LAT IN-PLANE SHEAR')
 1090 FORMAT('      6-LAT/Z TRANSVERSE SHEAR')
 1100 FORMAT('      7-LONG/Z TRANSVERSE SHEAR')
 1105 FORMAT('    ISOTROPIC (MAX SHEAR STRESS)')
 1110 FORMAT('      1-UNIAXIAL TENSION')
 1115 FORMAT('      2-UNIAXIAL COMPRESSION')
 1120 FORMAT('      3-SHEAR',/)
 1125 FORMAT('  CRITERIA VALUES (MAGNITUDES)')
 1130 FORMAT('    MATERIAL #',I2,'  (ORTHOTROPIC)')
 1132 FORMAT('    MATERIAL #',I2,'  (ISOTROPIC)')
 1135 FORMAT('      MODE ',I1,':',E12.6)
 1140 FORMAT('  INTER-PLY SHEAR DELAMINATION')
 1145 FORMAT('    X-DIRECTION (13):',E12.6)
 1150 FORMAT('    Y-DIRECTION (23):',E12.6,/)
 1155 FORMAT('  ESTIMATE MAX (MAGNITUDE) TRANSVERSE NORMAL',/,
     X     '      STRESS (1 FOR YES,0 FOR NO):',I2)
 1160 FORMAT('    MINIMUM (ABS VALUE) SIGMA23/SIGMA22 OR',/,
     X     '        SIGMA13/SIGMA11 RATIO FOR TRANSVERSE',/,
     X     '        NORMAL ESTIMATE= ',E12.6)
 1165 FORMAT(/,/,'INITIAL FAILURE RESULTS',/)


C  ------------------
C  STRESS CALCULATIONS
C  ------------------

      DO 200 II=1,NEM
        DO 210 JJ=1,4
  210     NPDOF(NOD(II,JJ))=7
        IF(NPE.EQ.8)THEN
          DO 220 JJ=5,8
  220       NPDOF(NOD(II,JJ))=2
        ENDIF
  200 CONTINUE
      NTDOF(1)=1
      DO 230 II=2,NNM
  230 NTDOF(II)=NTDOF(II-1)+NPDOF(II-1)


      DO 700 N=INCST,INCFN

C  READ IN PRESENT INCREMENT DISPLACEMENTS
```

```
      READ(8,*)INCPR
      READ(8,*)(GD(J),J=1,NDFT)

C  OUTPUT INCREMENT DATA TEXT

      WRITE(11,1500)N
 1500 FORMAT(/,'INCREMENT #',I4)

      DO 600 M=NELST,NELFN

C  OUTPUT ELEMENT DATA TEXT

      WRITE(11,1505)M
 1505 FORMAT(/,'  ELEMENT #',I4)

C  DETERMINE PRESENT ELEMENT'S NODAL (X,Y) COORDINATES
C    AND DISPLACEMENTS. THEN CALCULATE ITS STRESSES AT
C    EACH GAUSS POINT AND Z-COORDINATE

      DO 240 II=1,NPE
        NI=NOD(M,II)
        ELXY(II,1)=X(NI)
        ELXY(II,2)=Y(NI)
        KK1=NPDOF(NI)
        KK2=NTDOF(NI)-1
        NNI=7*(II-1)
        DO 240 JJ=1,KK1
  240     ELD(NNI+JJ)=GD(KK2+JJ)

      CALL FSTRESS(ELXY,GPXY)

C  AVERAGE MATERIAL STRESSES THROUGH THICKNESS FOR EACH PLY
C    AND GAUSS POINT BY TAKING AVERAGE OF PARABOLIC CURVEFIT.
C    FOR SINGLE PLY MODEL (STRESSES SAMPLED AT 5 THRU-THICKNESS
C    LOCATIONS) BREAK CURVEFIT INTO TWO PARABOLAS (SAMPLES 1-3
C    AND 3-5) FOR IN-PLANE STRESSES (S11,S22,S12). NO BREAKING
C    NEEDED FOR TRANS SHEAR STRESSES SINCE STRAIN-DISPLACEMENT
C    AND CONSTITUTIVE RELATIONS ASSUME PARABOLIC DISTRIBUTION
C  AVERAGE TRANSVERSE SHEAR STRESSES FOR EACH PLY INTERFACE
C    AND GAUSS POINT (AVERAGE OF 2 SAMPLES FOR BOTH S23 AND S13)

      DO 250 JJ=1,4
      DO 250 II=1,NP

        DO 260 LL=1,NUM
          XXX(LL)=ZZ(LL,II)
          Y11(LL)=SG11(JJ,(II-1)*NUM+LL)
          Y22(LL)=SG22(JJ,(II-1)*NUM+LL)
          Y12(LL)=SG12(JJ,(II-1)*NUM+LL)
          Y23(LL)=SG23(JJ,(II-1)*NUM+LL)
          Y13(LL)=SG13(JJ,(II-1)*NUM+LL)
  260   CONTINUE
        IF(NUM.EQ.3)THEN
          CALL PARFIT(NUM,XXX,Y11,A11,B11,C11)
          CALL PARFIT(NUM,XXX,Y22,A22,B22,C22)
          CALL PARFIT(NUM,XXX,Y12,A12,B12,C12)
        ENDIF
        CALL PARFIT(NUM,XXX,Y23,A23,B23,C23)
        CALL PARFIT(NUM,XXX,Y13,A13,B13,C13)
        ZKU=ZZ(1,II)
        ZKL=ZZ(NUM,II)
        ZD1=ZKL-ZKU
        ZD2=ZKL**2-ZKU**2
        ZD3=ZKL**3-ZKU**3
        IF(NUM.EQ.3)THEN
          ASG11(JJ,II)=(A11*ZD3/3.+B11*ZD2/2.+C11*ZD1)/ZD1
          ASG22(JJ,II)=(A22*ZD3/3.+B22*ZD2/2.+C22*ZD1)/ZD1
          ASG12(JJ,II)=(A12*ZD3/3.+B12*ZD2/2.+C12*ZD1)/ZD1
        ENDIF
        ASG23(JJ,II)=(A23*ZD3/3.+B23*ZD2/2.+C23*ZD1)/ZD1
        ASG13(JJ,II)=(A13*ZD3/3.+B13*ZD2/2.+C13*ZD1)/ZD1
        IF(NUM.EQ.3)GOTO 250
C
        ZKM=ZZ(3,II)
        ZDUM1=ZKM-ZKU
        ZDUM2=ZKM**2-ZKU**2
        ZDUM3=ZKM**3-ZKU**3
        ZDML1=ZKL-ZKM
        ZDML2=ZKL**2-ZKM**2
        ZDML3=ZKL**3-ZKM**3
        DO 255 LL=1,3
          XUM(LL)=XXX(LL)
          XML(LL)=XXX(LL+2)
          YUM11(LL)=Y11(LL)
          YUM22(LL)=Y22(LL)
          YUM12(LL)=Y12(LL)
          YML11(LL)=Y11(LL+2)
          YML22(LL)=Y22(LL+2)
          YML12(LL)=Y12(LL+2)
  255   CONTINUE
        CALL PARFIT(3,XUM,YUM11,AUM11,BUM11,CUM11)
        CALL PARFIT(3,XUM,YUM22,AUM22,BUM22,CUM22)
        CALL PARFIT(3,XUM,YUM12,AUM12,BUM12,CUM12)
        CALL PARFIT(3,XML,YML11,AML11,BML11,CML11)
        CALL PARFIT(3,XML,YML22,AML22,BML22,CML22)
```

```
      CALL PARFIT(3,XML,YML12,AML12,BML12,CML12)
      ASG11(JJ,II)=(AUM11*ZDUM3/3.+BUM11*ZDUM2/2.+
     X              CUM11*ZDUM1+AML11*ZDML3/3.+
     X              BML11*ZDML2/2.+CML11*ZDML1)/ZD1
      ASG22(JJ,II)=(AUM22*ZDUM3/3.+BUM22*ZDUM2/2.+
     X              CUM22*ZDUM1+AML22*ZDML3/3.+
     X              BML22*ZDML2/2.+CML22*ZDML1)/ZD1
      ASG12(JJ,II)=(AUM12*ZDUM3/3.+BUM12*ZDUM2/2.+
     X              CUM12*ZDUM1+AML12*ZDML3/3.+
     X              BML12*ZDML2/2.+CML12*ZDML1)/ZD1

  250 CONTINUE
      IF(NP.EQ.1)GOTO 270
      DO 270 JJ=1,4
      DO 270 II=1,NP-1
        IZUP=II*NUM
        DSG23(JJ,II)=.5*(SG23(JJ,IZUP)+SG23(JJ,IZUP+1))
        DSG13(JJ,II)=.5*(SG13(JJ,IZUP)+SG13(JJ,IZUP+1))
  270 CONTINUE

C  STRESS TRANSFORMATIONS
C    ISOTROPIC: 3-D PRINCIPAL STRESSES
C      CUBIC CHAR POLY ROOTS SOLVED VIA CLOSED-FORM SOLUTION FROM KORN
C      CHAR POLY SIMPLIFIED BY ASSUMPTION THAT SIG33=0
C    ORTHOTROPIC: 2-D ROTATION TO ORIENTATION DIRECTION
C    INTERLAMINAR SHEAR:2-D MAX SHEAR STRESS (TRANSFORM EQUIVALENT
C      TO PYTHAGOREAN THEOREM FOR THIS SPECIFIC CASE) CONSIDERING
C      ONLY SG23 AND SG13
C  TRANSFORMED STRESS VARIABLES
C    ORTHOTROPIC:TSG1() TO TSG5() = LONG, LAT, LONG-LAT, LAT-TRANS,
C                                   LONG-TRANS
C    ISOTROPIC:TSG1() TO TSG3() = MAX PRIN, MIN PRIN, MAX SHEAR

      DO 280 JJ=1,4
      DO 280 II=1,NP

        TSG1(JJ,II)=0.
        TSG2(JJ,II)=0.
        TSG3(JJ,II)=0.
        TSG4(JJ,II)=0.
        TSG5(JJ,II)=0.
        SM1=ASG11(JJ,II)
        SM2=ASG22(JJ,II)
        SM3=ASG12(JJ,II)
        SM4=ASG23(JJ,II)
        SM5=ASG13(JJ,II)
        IF(ISOPLY(II).EQ.0)GOTO 300

C --------------CUBIC POLY. ROOT SOLVER

C  CHAR POLY: X^3+AAA*X^2+BBB*X+CCC=0
C  COEFFICIENTS DERIVED FROM INVARIANTS OF STRESS TENSOR

        AAA=-(SM1+SM2)
        BBB=SM1*SM2-SM3**2-SM4**2-SM5**2
        CCC=-(2.*SM3*SM4*SM5-SM5**2*SM2-SM4**2*SM1)

        PPP=-AAA**2/3.+BBB
        QQQ=2.*(AAA/3.)**3-AAA*BBB/3.+CCC
        QPQP=(PPP/3.)**3+(QQQ/2.)**2
        IF(QPQP.LE.0.)THEN
          ALPHA=DACOS(-QQQ/2./(-PPP/3.)**1.5)
          PSR1=2.*(-PPP/3.)**.5*DCOS(ALPHA/3.)-AAA/3.
          PSR2=-2.*(-PPP/3.)**.5*DCOS((ALPHA+3.14159265)/3.)-AAA/3.
          PSR3=-2.*(-PPP/3.)**.5*DCOS((ALPHA-3.14159265)/3.)-AAA/3.
        ELSE
          PSR1=2*(-QQQ/2.)**(1./3.)-AAA/3.
          PSR2=-PSR1/2.
          PSR3=-PSR1/2.
        ENDIF

C  ---------------

          TSG1(JJ,II)=DMAX1(PSR1,PSR2,PSR3)
          TSG2(JJ,II)=DMIN1(PSR1,PSR2,PSR3)
          TSG3(JJ,II)=.5*(TSG1(JJ,II)-TSG2(JJ,II))
          GOTO 320

  300     CS=DCOS(RTHE(II))
          SN=DSIN(RTHE(II))
          TSG1(JJ,II)=CS**2*SM1+SN**2*SM2+2.*CS*SN*SM3
          TSG2(JJ,II)=CS**2*SM2+SN**2*SM1-2.*CS*SN*SM3
          TSG3(JJ,II)=(CS**2-SN**2)*SM3-CS*SN*(SM1-SM2)
          TSG4(JJ,II)=CS*SM4-SN*SM5
          TSG5(JJ,II)=CS*SM5+SN*SM4

  320 CONTINUE
  280 CONTINUE

C  INITIALIZE FAILURE MODE MATRICIES

        DO 400 II=1,NP
          DO 405 JJ=1,7
  405     MFAIL(II,JJ)=0
          IF(II.EQ.NP)GOTO 400
          ISDFAIL(II,1)=0
```

```
      ISDFAIL(II,2)=0
  400 CONTINUE

C  COMPARE TRANSFORMED STRESSES TO MATERIAL AND SHEAR DELAM
C   FAILURE CRITERIA

      DO 500 JJ=1,4
      DO 500 II=1,NP
        IF (ISOPLY(II).EQ.1) GOTO 550

C   ORTHOTROPIC: MAX STRESS CRITERIA
C   MODE:1-LONGITUDINAL TENSION, 2-LONG COMPRESSION
C        3-LATERAL TEN,          4-LAT COMP
C        5-IN PLANE(LONG-LAT) SHEAR
C        6-TRANSVERSE(LAT-Z) SHEAR
C        7-TRANS(LONG-Z) SHEAR
C   A VALUE OF ZERO IN FST(*,*) IGNORES THAT MODE FOR
C     THAT MATERIAL

        MNUM=MSS(II)
        IF(FST(MNUM,1).NE.0. .AND.
     X     TSG1(JJ,II).GE.FST(MNUM,1))MFAIL(II,1)=1
        IF(FST(MNUM,2).NE.0. .AND.
     X     TSG1(JJ,II).LE.-FST(MNUM,2))MFAIL(II,2)=1
        IF(FST(MNUM,3).NE.0. .AND.
     X     TSG2(JJ,II).GE.FST(MNUM,3))MFAIL(II,3)=1
        IF(FST(MNUM,4).NE.0. .AND.
     X     TSG2(JJ,II).LE.-FST(MNUM,4))MFAIL(II,4)=1
        IF(FST(MNUM,5).NE.0. .AND.
     X     DABS(TSG3(JJ,II)).GE.FST(MNUM,5))
     X     MFAIL(II,5)=1
        IF(FST(MNUM,6).NE.0. .AND.
     X     DABS(TSG4(JJ,II)).GE.FST(MNUM,6))
     X     MFAIL(II,6)=1
        IF(FST(MNUM,7).NE.0. .AND.
     X     DABS(TSG5(JJ,II)).GE.FST(MNUM,7))
     X     MFAIL(II,7)=1
        GOTO 575

C   ISOTROPIC: MAX UNIAXIAL AND SHEAR STRESS CRITERIA
C   MODE:1-UNIAXIAL TENSION, 2-UNI COMPRESSION
C        3-MAX SHEAR     (PRINCIPAL STRESSES)

  550   IF(NANAL(2).EQ.1)THEN
          MNUM=1
        ELSE
          MNUM=MSS(II)
        ENDIF
        IF(FST(MNUM,1).NE.0. .AND.
     X     TSG1(JJ,II).GE.FST(MNUM,1))MFAIL(II,1)=1
        IF(FST(MNUM,1).NE.0. .AND.
     X     TSG2(JJ,II).LE.-FST(MNUM,1))MFAIL(II,2)=1
        IF(FST(MNUM,1).NE.0. .AND.
     X     DABS(TSG3(JJ,II)).GE.FST(MNUM,1))
     X     MFAIL(II,3)=1

C   SHEAR DELAMINATION AT PLY INTERFACES

  575   IF(NP.EQ.1 .OR. II.EQ.1)GOTO 500
        IF(SLAMX.NE.0. .AND.
     X     DABS(DSG13(JJ,II-1)).GE.SLAMX)
     X     ISDFAIL(II-1,1)=1
        IF(SLAMY.NE.0. .AND.
     X     DABS(DSG23(JJ,II-1)).GE.SLAMY)
     X     ISDFAIL(II-1,2)=1

  500 CONTINUE

C  DETERMINE GAUSS POINTS REQUIRING TRANS NORMAL ESTIMATE
C  PERFORM IF I33=1 AND THE SIG23/SIG2 OR SIG13/SIG1 RATIO
C   AT ANY GAUSS POINT IS GREATER THAN THE SHEAR/MOMENT
C   RATIO PARAMETER (SHMT)

C  INITIALIZE ELEMENT INDICATOR

      IZZ=0
      IF(I33.NE.1)GOTO 810

C  FIND MAX SG11,SG22,SG23,SG13 AT EACH GP
C  CALCULATE SHEAR/MOMENT RATIOS

      DO 810 JJ=1,4
        S11X=DABS(SG11(JJ,1))
        S22X=DABS(SG22(JJ,1))
        S23X=DABS(SG23(JJ,1))
        S13X=DABS(SG13(JJ,1))

        DO 820 II=2,NP*NUM
          IF(DABS(SG11(JJ,II)).GT.S11X)
     X        S11X=DABS(SG11(JJ,II))
          IF(DABS(SG22(JJ,II)).GT.S22X)
     X        S22X=DABS(SG22(JJ,II))
          IF(DABS(SG23(JJ,II)).GT.S23X)
     X        S23X=DABS(SG23(JJ,II))
          IF(DABS(SG13(JJ,II)).GT.S13X)
     X        S13X=DABS(SG13(JJ,II))
```

```
 820   CONTINUE

       IF(S22X.EQ.0.)SR2322(JJ)=999.
       IF(S11X.EQ.0.)SR1311(JJ)=999.
       IF(S22X.EQ.0. .OR. S11X.EQ.0.)GOTO 815
       SR2322(JJ)=S23X/S22X
       SR1311(JJ)=S13X/S11X
 815   IF(SR2322(JJ).GE.SHMT .OR.
     X    SR1311(JJ).GE.SHMT)IZZ=1
 810 CONTINUE

C  CALCULATE TRANS NORMAL ESTIMATE

      IF(I33.NE.1 .OR. IZZ.NE.1)GOTO 830
      DO 830 JJ=1,4
        SG33(JJ)=0.
        DO 840 II=1,NP
          DO 850 LL=1,NUM
            XXX(LL)=ZZ(LL,II)
            Y232(LL)=SG232(JJ,(II-1)*NUM+LL)
            Y131(LL)=SG131(JJ,(II-1)*NUM+LL)
 850      CONTINUE
          CALL PARFIT(NUM,XXX,Y232,A,B,C)
            A232(JJ,II)=A
            B232(JJ,II)=B
            C232(JJ,II)=C
          CALL PARFIT(NUM,XXX,Y131,A,B,C)
            A131(JJ,II)=A
            B131(JJ,II)=B
            C131(JJ,II)=C
          ZKU=ZZ(1,II)
          ZKL=ZZ(NUM,II)
          IF((ZKL.GE.(-Z33)).AND.(ZKU.GT.(-Z33)))GOTO 840
          IF(ZKL.GE.(-Z33))ZKL=(-Z33)

C  ENFORCE EQUILIBRIUM SIG33,3=-(SIG23,2+SIG13,1)
C    TO ESTIMATE MAX MAGNITUDE OF SIG33 THROUGH THICKNESS
C  INTEGRATION INCLUDES SIGN CHANGE TO IMPOSE SIG33=0
C    ON BOTTOM OF PLATE INSTEAD OF TOP

          SG33(JJ)=SG33(JJ)+(A232(JJ,II)+A131(JJ,II))
     X             *(ZKL**3-ZKU**3)/3.
          SG33(JJ)=SG33(JJ)+(B232(JJ,II)+B131(JJ,II))
     X             *(ZKL**2-ZKU**2)/2.
          SG33(JJ)=SG33(JJ)+(C232(JJ,II)+C131(JJ,II))
     X             *(ZKL-ZKU)

 840    CONTINUE
 830 CONTINUE

C  ------------------------------
C  OUTPUT INITIAL FAILURE RESULTS
C  ------------------------------
C  --REPORT MATERIAL FAILURE
      DO 625 II=1,NP
        IRPT=0
        DO 630 JJ=1,NMODE(II)
          IF(MFAIL(II,JJ).EQ.1)IRPT=1
 630    CONTINUE
        IF(IRPT.EQ.0)GOTO 625
        IF(NANAL(2).EQ.1)THEN
          MAT=1
        ELSE
          MAT=MSS(II)
        ENDIF
        WRITE(11,1510)II,MAT
        WRITE(11,1515)
        DO 633 JJ=1,NMODE(II)
          IF(MFAIL(II,JJ).EQ.1)WRITE(11,1520)JJ
 633    CONTINUE
        IF(IRST.NE.1)GOTO 625
        WRITE(11,1525)
        DO 636 JJ=1,4
 636      WRITE(11,1530)GPXY(JJ,1),GPXY(JJ,2),ASG11(JJ,II),
     X      ASG22(JJ,II),ASG12(JJ,II),ASG23(JJ,II),ASG13(JJ,II)
        IF(ISOPLY(II).EQ.1)THEN
          WRITE(11,1540)
        ELSE
          WRITE(11,1535)
        ENDIF
        DO 640 JJ=1,4
          IF(ISOPLY(II).EQ.1)THEN
            WRITE(11,1545)GPXY(JJ,1),GPXY(JJ,2),TSG1(JJ,II),
     X        TSG2(JJ,II),TSG3(JJ,II)
          ELSE
            WRITE(11,1530)GPXY(JJ,1),GPXY(JJ,2),TSG1(JJ,II),
     X        TSG2(JJ,II),TSG3(JJ,II),TSG4(JJ,II),TSG5(JJ,II)
          ENDIF
 640    CONTINUE
 625  CONTINUE

C  --REPORT SHEAR DELAMINATION FAILURE
      IF(NP.EQ.1)GOTO 650
      DO 650 II=1,NP-1
        IRPT=0
        IF(ISDFAIL(II,1).EQ.1 .OR. ISDFAIL(II,2).EQ.1)IRPT=1
```

```
      IF(IRPT.NE.1)GOTO 650
      WRITE(11,1550)II,II+1
      WRITE(11,1555)
      WRITE(11,1560)ISDFAIL(II,1)
      WRITE(11,1565)ISDFAIL(II,2)
      IF(IRST.NE.1)GOTO 650
      WRITE(11,1567)
      DO 643 JJ=1,4
 643    WRITE(11,1568)GPXY(JJ,1),GPXY(JJ,2),DSG13(JJ,II),
     X                 DSG23(JJ,II)
 650 CONTINUE

C  --REPORT TRANSVERSE NORMAL STRESS ESTIMATES
     IF(I33.NE.1 .OR. IZZ.NE.1)GOTO 675
     IF(IRST.NE.1)GOTO 657
     WRITE(11,1570)
     DO 655 JJ=1,4
 655   WRITE(11,1580)GPXY(JJ,1),GPXY(JJ,2),SR2322(JJ),
     X                 SR1311(JJ)
     WRITE(11,1585)
 657 DO 660 JJ=1,4
       WRITE(11,1590)GPXY(JJ,1),GPXY(JJ,2),SG33(JJ)

c    code addition to write sigma33 results to separate
c    output file for graphical usage
c    if removed, assign number 660 to previous WRITE line
 660   write(14,1595)n,m,gpxy(jj,1),gpxy(jj,2),sg33(jj)
1595   format(2(I3,2X),2(E10.5,2X),E12.6)
c
 675 CONTINUE


1510 FORMAT(/,'     PLY #',I4,'  MATERIAL #',I2)
1515 FORMAT('       MATERIAL FAILURE MODES')
1520 FORMAT(8X,I2)
1525 FORMAT(/,'------AVERAGED PLY STRESSES',1X,
     X       '(X,Y,S11,S22,S12,S23,S13)')
1530 FORMAT(8X,2(E10.5,2X),5(E12.6,2X))
1535 FORMAT(,'------TRANSFORMED PLY STRESSES',1X,
     X       '(X,Y,SLL,STT,SLT,STZ,SLZ)')
1540 FORMAT('------PRINCIPAL PLY STRESSES',1X,
     X       '(X,Y,SMAX,SMIN,SMAXSHR)')
1545 FORMAT(8X,2(E10.5,2X),3(E12.6,2X))
1550 FORMAT(/,'     INTERFACE BETWEEN PLIES',I4,' AND',I4)
1555 FORMAT('       SHEAR DELAMINATION MODES',1X,
     X       '(1=YES,0=NO)')
1560 FORMAT('          X-DIR(13):',I2)
1565 FORMAT('          Y-DIR(23):',I2)
1567 FORMAT('------AVERAGE INTERFACE STRESSES',1X,
     X       '(X,Y,S13,S23)')
1568 FORMAT(8X,2(E10.5,2X),2(E12.6,2X))
1570 FORMAT(/,'     SHEAR/MOMENT RATIO MAGNITUDES',1X,
     X       '(X,Y,S23MAX/S22MAX,S13MAX/S11MAX)')
1580 FORMAT(6X,2(E10.5,2X),2(E12.6,2X))
1585 FORMAT('     MAX (MAGNITUDE) TRANS NORMAL',1X,
     X        'STRESS (X,Y,ESTIMATE)')
1590 FORMAT(6X,2(E10.5,2X),E12.6)

C  END ELEMENT LOOP
 600 CONTINUE
C  END INRCREMENT LOOP
 700 CONTINUE
     STOP
     END

------------------------------------------------------------------

     SUBROUTINE PARFIT(N,X,Y,A,B,C)

C  CALCULATES LEAST-SQUARES CURVEFIT OF N PAIRS OF DATA
C  (X,Y) TO A PARABOLA Y=A*X^2+B*X+C

     IMPLICIT DOUBLE PRECISION(A-H,O-Z)
     DIMENSION X(N),Y(N)

C  INITIALIZE DATA SUMS

     SX=0.
     SX2=0.
     SX3=0.
     SX4=0.
     SY=0.
     SXY=0.
     SX2Y=0.

C  CALCULATE SUMS

     DO 10 I=1,N
       SX=SX+X(I)
       SX2=SX2+X(I)**2
       SX3=SX3+X(I)**3
       SX4=SX4+X(I)**4
       SY=SY+Y(I)
       SXY=SXY+X(I)*Y(I)
```

```
      SX2Y=SX2Y+(X(I)**2)*Y(I)
   10 CONTINUE

C  CALCULATE CRAMER'S RULE DETERMINANTS

      D= N*SX4*SX2 + 2.*SX3*SX2*SX
     X   -SX2**3 - SX4*(SX**2) - N*(SX3**2)

      D1=N*SX2Y*SX2 + SY*SX3*SX + SXY*SX2*SX
     X   -SY*(SX2**2) - SX2Y*(SX**2) - N*SXY*SX3

      D2=N*SXY*SX4 + SX2Y*SX2*SX + SY*SX3*SX2
     X   -SXY*(SX2**2) - N*SX2Y*SX3 - SY*SX4*SX

      D3=SY*SX4*SX2 + SXY*SX3*SX2 + SX2Y*SX3*SX
     X   -SX2Y*(SX2**2) - SXY*SX4*SX - SY*(SX3**2)

C  CALCULATE COEFICIENTS VIA CRAMER'S RULE

      A=D1/D
      B=D2/D
      C=D3/D

      RETURN
      END
```

---

```
      SUBROUTINE FSTRESS (ELXY,GPXY)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION K1,I,J,K,L,NU,NU12(5),KS1,KS2
C     ................................................................
C     THIS SUBROUTINE EVALUATES THE STRESSES AT THE OUTER GAUSS POINTS
C     ................................................................
C
      COMMON/SHP/SF(4),DSF(2,4),HRM(4,3),D1HRM(4,3),D2HRM(4,3),
     X          DD1HRM(4,3),DD2HRM(4,3),D12HRM(4,3),QSF(8),DQSF(2,8)
      COMMON/DISP/ELD(56),Q(18)
      COMMON/FAIL/IEL,NPE,NDF,P1,NANAL(3),NP,K1,CON(6,100),
     X          CONS(3,100),ZZ(5,100),SG11(4,300),SG22(4,300),
     X          SG12(4,300),SG23(4,300),SG13(4,300),SG232(4,300),
     X          SG131(4,300),I33
      DIMENSION ELXY(8,2),GPXY(4,2),GAUSS(7,7)
C
      DATA GAUSS/7*0.0D0,-.57735027D0,.57735027D0,5*0.0D0,-.77459667D0,0
     1.0D0,.77459667D0,4*0.0D0,-.86113631D0,-.33998104D0,.33998104D0,.86
     2113631D0,3*0.0D0,-.90617985D0,-.53846931D0,0.0D0,.53846931D0,.9061
     37985D0,2*0.0D0,-.93246951D0,-.66120939D0,-.23861919D0,.23861919D0,
     4.66120939D0,.93246951D0,0.0D0,-.94910791D0,-.74153119D0,-.40584515
     5D0,0.0D0,.40584515D0,.74153119D0,.94910791D0/
C
      NGP=2*NANAL(1)**2-5*NANAL(1)+7
      DO 40 NI=1,NGP,NGP-1
      DO 40 NJ=1,NGP,NGP-1
      XI=GAUSS(NI,NGP)
      ETA=GAUSS(NJ,NGP)
      CALL SHAPE (NPE,XI,ETA,ELXY,DET)
      CALL DIS(NPE)
      X=0.
      Y=0.
      DO 20 II=1,NPE
      PIN=SF(II)
      IF(NPE.EQ.8)PIN=QSF(II)
      X=X+PIN*ELXY(II,1)
   20 Y=Y+PIN*ELXY(II,2)

      NNII=1
      NNJJ=1
      IF(NI.EQ.NGP)NNII=2
      IF(NJ.EQ.NGP)NNJJ=2
      GPXY((NNII-1)*2+NNJJ,1)=X
      GPXY((NNII-1)*2+NNJJ,2)=Y

      E11=0.
      E12=0.
      E22=0.
      E23=0.
      E13=0.
      NL11=0.
      NL22A=0.
      NL22B=0.
      NL22=0.
      NL12A=0.
      NL12B=0.
      NL12=0.
      NUM=3
      IF(NP.EQ.1)NUM=5
      DO 22 JJ=1,NP
      DO 24 M=1,NUM
      E11=Q(2)+ZZ(M,JJ)*Q(14)+ZZ(M,JJ)**3*K1*(Q(10)+Q(14))
      E22=Q(6)-P1*Q(7)+ZZ(M,JJ)*(Q(18)-P1**2*Q(7))+ZZ(M,JJ)**2*P1*Q(18)
```

```
     1    +ZZ(M,JJ)**3*K1*(Q(11)+Q(18))+ZZ(M,JJ)**4*P1*K1*(Q(11)+Q(18))
      E12=Q(3)+Q(5)+ZZ(M,JJ)*(Q(3)*P1-Q(5)*P1+Q(15)+Q(17))+ZZ(M,JJ)**2
     1    *P1*Q(15)+ZZ(M,JJ)**3*K1*(2.*Q(12)+Q(15)+Q(17))+ZZ(M,JJ)**4
     1    *P1*K1*(Q(12)+Q(15))
      E23=Q(9)+Q(16)+ZZ(M,JJ)**2*3.*K1*(Q(9)+Q(16))
      E13=Q(8)+Q(13)+ZZ(M,JJ)**2*3.*K1*(Q(8)+Q(13))
C
C  E13,i AND E23,i INCLUDED FOR ESTIMATING SIGMA33 BY ENFORCING
EQUILIBRIUM
C
      IF(I33.EQ.1)THEN
        E231=Q(12)+Q(17)+ZZ(M,JJ)**2*3.*K1*(Q(12)+Q(17))
        E232=Q(11)+Q(18)+ZZ(M,JJ)**2*3.*K1*(Q(11)+Q(18))
        E131=Q(10)+Q(14)+ZZ(M,JJ)**2*3.*K1*(Q(10)+Q(14))
        E132=Q(12)+Q(15)+ZZ(M,JJ)**2*3.*K1*(Q(12)+Q(15))
      ENDIF
C
      IF(NANAL(1).NE.0) GOTO 70
C**********************************************************************
C FOR NONLINEAR ANALYSIS CALCULATE THE NONLINEAR STRAINS
C**********************************************************************
      NL11=.5*(Q(2)**2+Q(8)**2+Q(5)**2)+ZZ(M,JJ)*(-Q(5)**2*P1+Q(14)*
     1 Q(2)+Q(5)*Q(17))+ZZ(M,JJ)**2*(Q(5)**2*P1**2/2.-Q(17)*Q(5)*P1+
     2 .5*(Q(14)**2+Q(17)**2))+ZZ(M,JJ)**3*(Q(2)*K1*(Q(10)+Q(14))+
     3 Q(5)*K1*(Q(12)+Q(17)))+ZZ(M,JJ)**4*(-Q(5)*K1*P1*(Q(12)+Q(17))+
     4 Q(14)*K1*(Q(10)+Q(14))+Q(17)*K1*(Q(12)+Q(17)))+ZZ(M,JJ)**6*
     5 K1**2*.5*(Q(10)**2+2*Q(10)*Q(14)+Q(14)**2+Q(12)**2+2*Q(12)*
     6 Q(17)+Q(17)**2)
      NL22A=.5*(Q(6)**2+Q(9)**2+Q(3)**2+Q(4)**2*P1**2)-Q(6)*Q(7)*P1+
     1 Q(4)*Q(9)*P1+Q(7)**2*.5*P1**2+ZZ(M,JJ)*(Q(3)**2*P1+Q(9)**2*P1+
     2 Q(7)**2*P1**3-P1**2*(Q(6)*Q(7)-Q(4)*Q(9))+Q(4)*Q(16)*P1**2+Q(6)*
     3 Q(18)+Q(3)*Q(15)-P1*(Q(18)*Q(7)-Q(16)*Q(9)))+ZZ(M,JJ)**2*(2*P1*
     4 Q(15)*Q(3)+P1*Q(18)*Q(6)+Q(4)*Q(16)*P1**3-2*P1**2*(Q(18)*Q(7)-
     5 Q(16)*Q(9))+.5*P1**2*Q(16)**2+.5*(Q(18)**2+Q(15)**2))+ZZ(M,JJ)**4
     6 *(2*K1*P1*Q(3)*(Q(12)+Q(15))+K1*P1*Q(6)*(Q(11)+Q(18))+K1*P1**3
     7 *Q(4)*(Q(9)+Q(16))-2*K1*P1**2*Q(7)*(Q(11)+Q(18))+2*K1*P1**2*Q(9)
     8 *(Q(9)+Q(16))+K1*Q(18)*(Q(11)+Q(18))+K1*Q(15)*(Q(12)+Q(15))+
     9 K1*P1**2*Q(16)*(Q(9)+Q(16)))
      NL22B=ZZ(M,JJ)**3*(P1*(Q(18)**2+Q(15)**2)+Q(16)
     1 **2*P1**3+K1*Q(6)*(Q(11)+Q(18))+K1*Q(3)*(Q(12)+Q(15))+Q(4)*K1*
     2 P1**2*(Q(9)+Q(16))-K1*P1*(Q(7)*(Q(11)+Q(18))-Q(9)*(Q(9)+Q(16))))+
     3 ZZ(M,JJ)**5*2*K1*(P1*(Q(18)*(Q(11)+Q(18))+P1*Q(15)*(Q(12)+Q(15))+
     4 P1**3*Q(16)*(Q(9)+Q(16))))+ZZ(M,JJ)**6*K1**2*.5*(Q(11)**2+2*Q(11)
     5 *Q(18)+Q(18)**2+Q(12)**2+2*Q(12)*Q(15)+Q(15)**2+P1**2*(Q(9)**2+
     6 2*Q(9)*Q(16)+Q(16)**2))+ZZ(M,JJ)**7*K1**2*(P1*(Q(11)+Q(18))**2+
     7 P1*(Q(12)+Q(15))**2+P1**3*(Q(9)+Q(16))**2)
      NL22=NL22A+NL22B
      NL12A=Q(2)*Q(3)+Q(5)*Q(6)+Q(8)*Q(9)+P1*(-Q(5)*Q(7)+Q(4)*Q(8))+
     1 ZZ(M,JJ)*(P1*(Q(2)*Q(3)-Q(5)*Q(6)+Q(8)*Q(9))+Q(2)*Q(15)+Q(3)*
     2 Q(14)+Q(5)*Q(18)+Q(6)*Q(17)+P1*(-Q(7)*Q(17)+Q(8)*Q(16)))+
     3 ZZ(M,JJ)**2*(Q(14)*Q(15)+Q(17)*Q(18)+P1*(Q(15)*Q(2)+Q(14)*Q(3))+
     4 P1**2*(-Q(7)*Q(17)+Q(8)*Q(16)))+ZZ(M,JJ)**3*(K1*Q(3)*
     5 (Q(10)+Q(14))+K1*Q(2)*(Q(12)+Q(15))+K1*Q(5)*(Q(18)+Q(11))+
     6 K1*Q(6)*(Q(12)+Q(17))-K1*P1*Q(7)*(Q(12)+Q(17))+K1*P1*Q(8)*(Q(9)+
     7 Q(16))+P1*(Q(14)*Q(15)+Q(17)*Q(18)))
      NL12B=ZZ(M,JJ)**4*(K1*P1*Q(3)*(Q(10)+Q(14))-K1*P1**2*
     8 (Q(7)*(Q(12)+Q(17))-Q(8)*(Q(9)+Q(16)))+K1*(Q(14)*(Q(12)+Q(15))+
     9 Q(15)*(Q(10)+Q(14))+Q(17)*(Q(11)+Q(18))+Q(18)*(Q(12)+Q(17)))+
     1 K1*P1*Q(2)*(Q(12)+Q(15)))+ZZ(M,JJ)**5*K1*P1*(Q(14)*(Q(12)+Q(15))+
     2 Q(15)*(Q(10)+Q(14))+Q(17)*(Q(11)+Q(18))+
     3 Q(18)*(Q(12)+Q(17)))+ZZ(M,JJ)**6*K1**2*((Q(10)+Q(14))*(Q(12)
     4 +Q(15))+(Q(12)+Q(17))*(Q(11)+Q(18)))+ZZ(M,JJ)**7*P1*K1**2*
     5 ((Q(10)+Q(14))*(Q(12)+Q(15))*(Q(12)+Q(17))*(Q(11)+Q(18)))
      NL12=NL12A+NL12B
      E11=E11+NL11
      E22=E22+NL22
      E12=E12+NL12
C
C NROW IS THE GAUSS POINT LABEL= 1 TO 4 (INCREMENT ETA THEN XI)
C NCOL IS THE THRU THICKNESS LABEL=1 TO NP*NUM (INCREMENT Z-COORD
C          THEN PLY NUMBER)
C
   70 NROW=(NNII-1)*2+NNJJ
      NCOL=NUM*(JJ-1)+M
      SG11(NROW,NCOL)=CON(1,JJ)*E11+CON(2,JJ)*E22+CON(3,JJ)*E12
      SG22(NROW,NCOL)=CON(2,JJ)*E11+CON(4,JJ)*E22+CON(5,JJ)*E12
      SG12(NROW,NCOL)=CON(3,JJ)*E11+CON(5,JJ)*E22+CON(6,JJ)*E12
      SG23(NROW,NCOL)=CONS(1,JJ)*E23+CONS(2,JJ)*E13
      SG13(NROW,NCOL)=CONS(2,JJ)*E23+CONS(3,JJ)*E13
      IF(I33.EQ.1)THEN
        SG232(NROW,NCOL)=CONS(1,JJ)*E232+CONS(2,JJ)*E132
        SG131(NROW,NCOL)=CONS(2,JJ)*E231+CONS(3,JJ)*E131
      ENDIF
   24 CONTINUE
   22 CONTINUE
   40 CONTINUE
c
c   average stress gradients across element gauss points
c   in order to minimize sharp fluctuations due to in-plane
c   discontinuites in stress fields
c
      if(i33.ne.1)goto 50
      do 50 jj=1,np
      do 55 m=1,num
        ncol=num*(jj-1)+m
        sg232(1,ncol)=.5*(sg232(1,ncol)+sg232(2,ncol))
```

```
      sg232(2,ncol)=sg232(1,ncol)
      sg232(3,ncol)=.5*(sg232(3,ncol)+sg232(4,ncol))
      sg232(4,ncol)=sg232(3,ncol)
      sg131(1,ncol)=.5*(sg131(1,ncol)+sg131(3,ncol))
      sg131(3,ncol)=sg131(1,ncol)
      sg131(2,ncol)=.5*(sg131(2,ncol)+sg131(4,ncol))
      sg131(4,ncol)=sg131(2,ncol)
   55 continue
   50 continue
c
c
      RETURN
      END
```

---

```
      SUBROUTINE SHAPE(NPE,XI,ETA,ELXY,DET)
C
C     ..................................................................
C     THE SUBROUTINE EVALUATES THE INTERPOLATION FUNCTIONS (SF(I)) AND
C     ITS DERIVATIVES WITH RESPECT TO NATURAL COORDINATES (DSF(I,J)) FOR
C     THE LINEAR DISPLACEMENTS, U,V, PSI1,AND PSI2.  ALSO EVALUATES THE
C     INTERPOLATION FUNCTIONS (HRM(I,J)) AND DERIVATIVES WRT NATURAL
C     COORDINATES (DHRM(I,J) AND DDHRM(I,J), ALL AT GAUSS PT (XI,ETA)
C
C     SF(I)........INTERPOLATION FUNCTION FOR NODE I OF THE ELEMENT
C     DSF(I,J).....DERIVATIVE OF SF(J) WITH RESPECT TO XI IF I=1 AND
C                  WITH RESPECT TO ETA IF I=2.
C     QSF(I).......QUADRATIC INTERPOLATION FUNCTION FOR NODE I
C     DQSF(I,J)....DERIVATIVE OF QSF(I)
C     HRM(I,J).....HERMITIAN INTERPOLATION FUNCTION FOR NODE I
C     D1HRM(I,J)...FIRST DERIVATIVES OF HRM(I,J) WRT TO XI
C     D2HRM(I,J)...FIRST DERIVATIVES OF HRM(I,J) WRT TO ETA
C     DD1HRM(I,J)..SECOND DERIVATIVES OF HRM(I,J) WRT TO XI
C     DD2HRM(I,J)..SECOND DERIVATIVES OF HRM(I,J) WRT TO ETA
C     D12HRM(I,J)..SECOND MIXED DERIVATIVE OF HRM(I,J)
C     XNODE(I,J)...J-TH (J=1,2) COORDINATE OF NODE I OF THE ELEMENT
C     ..................................................................
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/SHP/ SF(4),DSF(2,4),HRM(4,3),D1HRM(4,3),
     X           D2HRM(4,3),DD1HRM(4,3),DD2HRM(4,3),D12HRM(4,3),
     X           QSF(8),DQSF(2,8)
      DIMENSION ELXY(8,2),XNODE(8,2)
      DATA XNODE/-1.0D0,1.0D0,1.0D0,-1.0D0,0.0D0,1.0D0,0.0D0,-1.0D0,
     X          -1.0D0,-1.0D0,1.0D0,1.0D0,-1.0D0,0.0D0,1.0D0,0.0D0/
C
C     LINEAR LAGRANGIAN INTERPOLATION  FUNCTIONS U,V,PSI1,PSI2
C
      DO 70 I = 1,4
      XP=XNODE(I,1)
      YP=XNODE(I,2)
      XI0=1.0+XI*XP
      ETA0=1.0+ETA*YP
      SF(I)=0.25*(XI0*ETA0)
      DSF(1,I)=0.25*(XP*ETA0)
   70 DSF(2,I)=0.25*(YP*XI0)
C
C  NONCONFORMING HERMITIAN INTERPOLATION FUNCTIONS FOR W
C
      AA=(ELXY(3,1)-ELXY(1,1))/2.
      BB=(ELXY(3,2)-ELXY(1,2))/2.
      DET=AA*BB
      DO 80 I=1,4
      XP=XNODE(I,1)
      YP=XNODE(I,2)
      XI0=1.0+XI*XP
      ETA0=1.0+ETA*YP
      HRM(I,1)=.125*(XI0*ETA0)*(XI0+ETA0-XI**2-ETA**2)
      HRM(I,2)=.125*XP*XI0**2*ETA0*(XP*XI-1.)*AA
      HRM(I,3)=.125*YP*ETA0**2*XI0*(YP*ETA-1.)*BB
      D1HRM(I,1)=.125*ETA0*XP*(2.+ETA0-ETA**2-3.*XI**2)
      D1HRM(I,2)=.125*ETA0*(-1.+2.*XP*XI+3.*XI**2)*AA
      D1HRM(I,3)=.125*XP*YP*ETA0**2*(-1.+YP*ETA)*BB
      D2HRM(I,1)=.125*XI0*YP*(2.+XI0-XI**2-3.*ETA**2)
      D2HRM(I,2)=.125*XP*YP*XI0**2*(-1.+XP*XI)*AA
      D2HRM(I,3)=.125*XI0*(-1.+3.*ETA**2+2.*YP*ETA)*BB
      DD1HRM(I,1)=-.75*ETA0*XP*XI
      DD1HRM(I,2)=.125*ETA0*2.*(XP+3.*XI)*AA
      DD1HRM(I,3)=0.
      DD2HRM(I,1)=-.75*YP*ETA*XI0
      DD2HRM(I,2)=0.
      DD2HRM(I,3)=.125*XI0*2.*(YP+3.*ETA)*BB
      D12HRM(I,1)=.125*YP*XP*(4.-3.*ETA**2-3.*XI**2)
      D12HRM(I,2)=.125*YP*(-1.+2.*XP*XI+3.*XI**2)*AA
   80 D12HRM(I,3)=.125*XP*(-1.+2.*YP*ETA+3.*ETA**2)*BB
C
C CONVERT TO GLOBAL SHAPE FUNCTIONS BY MULTIPLYING BY ELEMENTS
C OF THE JACOBIAN.  THIS APPLIES ONLY TO RECTANGULAR ELEMENTS,
C SEE 5JUN87 MEMO.
C
      DO 140 I=1,4
      DSF(1,I)=DSF(1,I)/AA
      DSF(2,I)=DSF(2,I)/BB
```

```
      DO 142 J=1,3
      D1HRM(I,J)=D1HRM(I,J)/AA
      D2HRM(I,J)=D2HRM(I,J)/BB
      DD1HRM(I,J)=DD1HRM(I,J)/AA**2
      DD2HRM(I,J)=DD2HRM(I,J)/BB**2
  142 D12HRM(I,J)=D12HRM(I,J)/(AA*BB)
  140 CONTINUE
      IF(NPE.EQ.4)RETURN
C
C QUADRATIC LAGRANGIAN INTERPOLATION FUNCTIONS FOR U AND V
C
      DO 10 I=1,8
      XP=XNODE(I,1)
      YP=XNODE(I,2)
      XIO=1.+XI*XP
      ETAO=1.+ETA*YP
      IF(I.GT.4)GOTO 11
      QSF(I)=.25*ETAO*XIO*(ETA*YP+XI*XP-1.)
      DQSF(1,I)=.25*XP*ETAO*(2.*XI*XP+ETA*YP)
      DQSF(2,I)=.25*YP*XIO*(2.*ETA*YP+XI*XP)
      GOTO 10
   11 IF(I.EQ.6 .OR. I.EQ.8)GOTO 12
      QSF(I)=.5*(1.-XI*XI)*ETAO
      DQSF(1,I)=-XI*ETAO
      DQSF(2,I)=.5*YP*(1.-XI*XI)
      GOTO 10
   12 QSF(I)=.5*(1.-ETA*ETA)*XIO
      DQSF(1,I)=.5*XP*(1.-ETA*ETA)
      DQSF(2,I)=-ETA*XIO
   10 CONTINUE
      DO 150 I=1,NPE
      DQSF(1,I)=DQSF(1,I)/AA
  150 DQSF(2,I)=DQSF(2,I)/BB
      RETURN
      END
```

---

```
      SUBROUTINE DIS(NPE)
C.................................................................
C  THIS SUBROUTINE GENERATES THE DISPLACEMENT GRADIENT VECTOR, Q(I),
C  FOR A ELEMENT AT A GAUSS POINT.  Q(I)=D(I,J)*ELD(J)
C.................................................................
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/DISP/ELD(56),Q(18)
      COMMON/SHP/SF(4),DSF(2,4),HRM(4,3),D1HRM(4,3),D2HRM(4,3),
     X          DD1HRM(4,3),DD2HRM(4,3),D12HRM(4,3),QSF(8),DQSF(2,8)
      DO 5 I=1,18
    5 Q(I)=0.
      DO 10 I=1,4
      DO 20 J=1,3
      TT=ELD(J+7*I-5)
      Q(7)=Q(7)+HRM(I,J)*TT
      Q(8)=Q(8)+D1HRM(I,J)*TT
      Q(9)=Q(9)+D2HRM(I,J)*TT
      Q(10)=Q(10)+DD1HRM(I,J)*TT
      Q(11)=Q(11)+DD2HRM(I,J)*TT
   20 Q(12)=Q(12)+D12HRM(I,J)*TT
      Q(13)=Q(13)+SF(I)*ELD(7*I-1)
      Q(14)=Q(14)+DSF(1,I)*ELD(7*I-1)
      Q(15)=Q(15)+DSF(2,I)*ELD(7*I-1)
      Q(16)=Q(16)+SF(I)*ELD(7*I)
      Q(17)=Q(17)+DSF(1,I)*ELD(7*I)
   10 Q(18)=Q(18)+DSF(2,I)*ELD(7*I)
      IF(NPE.EQ.8)GOTO 40
      DO 30 I=1,NPE
      Q(1)=Q(1)+SF(I)*ELD(7*I-6)
      Q(2)=Q(2)+DSF(1,I)*ELD(7*I-6)
      Q(3)=Q(3)+DSF(2,I)*ELD(7*I-6)
      Q(4)=Q(4)+SF(I)*ELD(7*I-5)
      Q(5)=Q(5)+DSF(1,I)*ELD(7*I-5)
   30 Q(6)=Q(6)+DSF(2,I)*ELD(7*I-5)
      RETURN
   40 DO 50 I=1,8
      Q(1)=Q(1)+QSF(I)*ELD(7*I-6)
      Q(2)=Q(2)+DQSF(1,I)*ELD(7*I-6)
      Q(3)=Q(3)+DQSF(2,I)*ELD(7*I-6)
      Q(4)=Q(4)+QSF(I)*ELD(7*I-5)
      Q(5)=Q(5)+DQSF(1,I)*ELD(7*I-5)
   50 Q(6)=Q(6)+DQSF(2,I)*ELD(7*I-5)
      RETURN
      END
```

# Vita

Lieutenant Damin J Siler was born on 2 October 1970 in Saginaw, Michigan and grew up in the neighboring town of Merrill. In 1989 he graduated from Nouvel Catholic Central High School in Saginaw and began study at Michigan Technological University. He graduated from there in May of 1993 with a Bachelor of Science in Mechanical Engineering and Summa Cum Laude honors. Upon graduation he also received a reserve commission in the USAF through AFROTC. His first active duty assignment was to attend the Air Force Institute of Technology and pursue a Master of Science in Aeronautical Engineering.