Design and Analysis of Laminated Composite Materials

Thesis submitted in partial fulfillment of the requirements for the Degree of

Bachelor of Technology (B. Tech.)

In

Mechanical Engineering

By

BHAGYASHREE SUNA
Roll No. 110ME0335

Under the Guidance of

Prof. J. Srinivas

NATIONAL INSTITUTE OF TECHNOLOGY

ROURKELA 769008, INDIA

NATIONAL INSTITUTE OF TECHNOLOGY

ROURKELA 769008, INDIA

Certificate of Approval

This is to certify that the thesis entitled **Design and Analysis of Laminated Composite Materials** submitted by ***Miss Bhagyashree Suna*** has been carried out under my supervision in partial fulfillment of the requirements for the Degree of ***Bachelor of Technology*** in ***Mechanical Engineering*** at **National Institute of Technology**, **NIT Rourkela**, and this work has not been submitted elsewhere before for any other academic degree/diploma.

Dr. J. Srinivas

Associate Professor
Department of Mechanical Engineering
National Institute of Technology, Rourkela

Rourkela-769008

Date:

Acknowledgement

I would like to express my deep sense of gratitude and indebtedness to **Dr. J. Srinivas***, Associate Professor, Department of Mechanical Engineering, NIT Rourkela, my supervisor, whose invaluable encouragement, suggestions, and support leads to make successful completion of the thesis work. His meticulous guidance at each phase of this thesis has inspired and helped me innumerable ways.*

I would also like to show my sincere thanks to **Prof. K. P. Maity***, Professor and Head of the Department, Mechanical Engineering; I offer my thanks to all my friends and especially those involved in helping me to give basic ideas in writing graphic user interfaces and use of ANSYS software in laminated composite analysis. I am grateful to my parents and all my teachers who indirectly helped me through-out.*

Bhagyashree Suna

Abstract

Composite materials have interesting properties such as high strength to weight ratio, ease of fabrication, good electrical and thermal properties compared to metals. A laminated composite material consists of several layers of a composite mixture consisting of matrix and fibers. Each layer may have similar or dissimilar material properties with different fiber orientations under varying stacking sequence. There are many open issues relating to design of these laminated composites. Design engineer must consider several alternatives such as best stacking sequence, optimum fiber angles in each layer as well as number of layers itself based on criteria such as achieving highest natural frequency or largest buckling loads of such structure. Analysis of such composite materials starts with estimation of resultant material properties. Both classical theory and numerical methods such as finite element modeling may be employed in this line. Further, these estimated properties are to be used for computing the dynamic properties of the members made-up of these materials as equivalent isotropic members. At this level, a Graphic User Interface (GUI) device is developed with MATLAB programming to interactively create a user friendly environment for computing overall material properties using classical laminate theory. User can enter the number of layers and layer orthotropic properties and the back end program calculates the extension, bending and coupling stiffness matrices and further it estimates the overall elastic constants, Poisson ratios and density. The result will be displayed in the front end interface boxes. The obtained constants are validated with an ANSYS model, where the laminate stacking sequence is built and the member is subjected to a uniform strain at free end, while the reaction stress at the fixed end is predicted. The developed interface simplifies the design process to some extent. The dynamic analysis in terms of fundamental natural frequency and critical buckling load is illustrated by using these overall material constants as a later part of analysis.

Contents

Content *Page*

1.Introduction

2. Mathematical Modeling

3. Methodology

4. Results & Discussion

Chapter 1

Introduction

Laminated composite materials are extensively used in aerospace, defense, marine, automobile, and many other industries. They are generally lighter and stiffer than other structural materials. A laminated composite material consists of several layers of a composite mixture consisting of matrix and fibers. Each layer may have similar or dissimilar material properties with different fiber orientations under varying stacking sequence. Because, composite materials are produced in many combinations and forms, the design engineer must consider many design alternatives. It is essential to know the dynamic and buckling characteristics of such structures subjected to dynamic loads in complex environmental conditions. For example, when the frequency of the loads matches with one of the resonance frequencies of the structure, large translation/torsion deflections and internal stresses occur, which may lead to failure of structure components. The structural components made of composite materials such as aircraft wings, helicopter blades, vehicle axles and turbine blades can be approximated as laminated composite beams.

1.1 Laminated Composite Structures

A laminate is constructed by stacking a number of laminas in the thickness (z) direction. Each layer is thin and may have different fiber orientation. The fiber orientation, stacking arrangements and material properties influence the response from the laminate. The theory of lamination is same whether the composite structure may be a plate, a beam or a shell. Fig.1.1 shows a laminated plate or panel considered in most of the analysis. The following assumptions are made in formulations: (i) The middle plane of the plate is taken as the reference plane. (ii) The laminated plate consists of arbitrary number of homogeneous, linearly elastic orthotropic layers perfectly bonded to each other. (iii) The analysis follows linear constitutive relations i.e.

obeys generalized Hooke's law for the material. (iv)The lateral displacements are small compared to plate thickness. (v) Normal strain in z-direction is neglected.

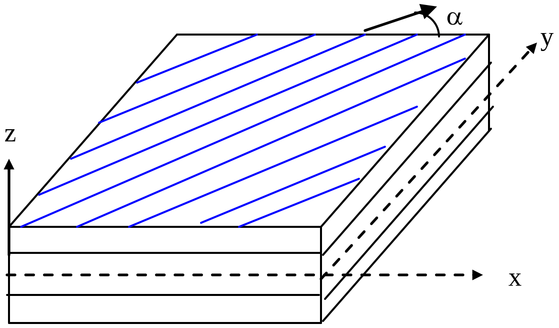


Fig.1.1 Plate

As shown in Fig.1.2, laminated beams are made-up of many plies of orthotropic materials and the principal material axes of a ply may be oriented at an arbitrary angle with respect to the x-axis. In the right-handed Cartesian coordinate system, the x-axis coincides with the beam axis and its origin is on the mid-plane of the beam. The length, breadth and thickness of the beam are represented by L, b and h, respectively.

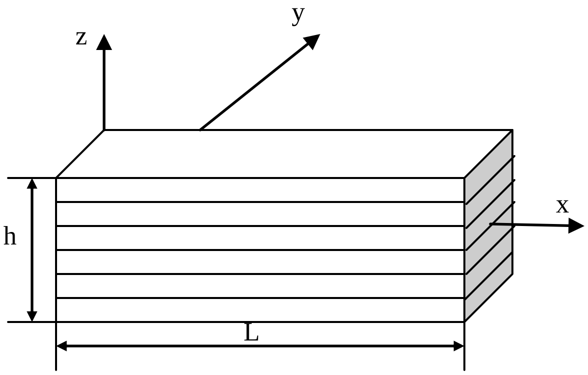


Fig.1.2 Beams

In practical engineering applications, laminated shells of revolution may have different geometries based mainly on their curvature characteristics such as cylindrical shells, spherical shells and conical shells. The composite shell of revolution is composed of orthotropic layers of uniform thickness as shown in Fig.1.3. A differential element of a laminated shell shown with orthogonal curvilinear coordinate system located on the middle surface of the shell. The total thickness of the shell is *h*.

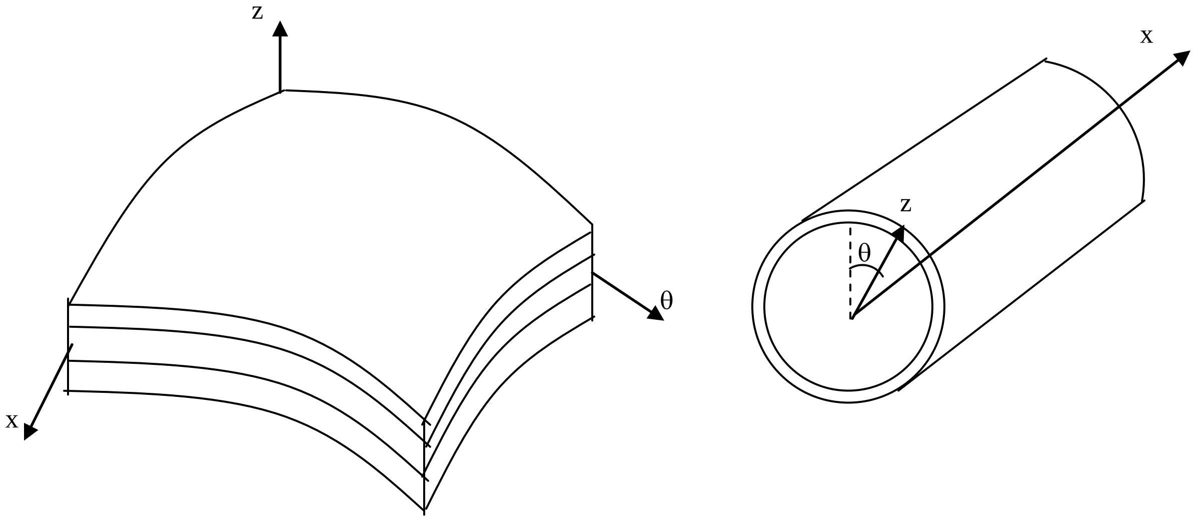


Fig.1.3 Shell (cylindrical)

1.2 Literature Review

This section brief-outs the various earlier works done in the area of laminated composite material. These are grouped under four broad headings. More recently, Hajianmaleki[1] presented a review of analysis of laminated composite structures used in recent decades.

Laminated Beams

Many authors analyzed the laminated beam structures.

Yildirim [2] used stiffness method for the solution of the purely in-plane free vibration problem of symmetric cross-ply laminated beams. The rotary inertia, axial and transverse shear deformation effects are considered in the mathematical model by the first-order shear deformation theory. A total of six degrees of freedom, four displacements and two rotations are defined for an element. The exact in-plane element stiffness matrix of 6×6 is obtained based on the transfer matrix method. The element inertia matrix consists of the concentrated masses. The sub-space iteration and Jacobi's methods are employed in the solution of the large-scale general eigenvalue problem.

Jun *et al*. [3] introduced a dynamic finite element method for free vibration analysis of generally laminated composite beams on the basis of first-order shear deformation theory. The influences of Poisson effect, couplings among extensional, bending and torsional deformations, shear deformation and rotary inertia are incorporated in the formulation. The dynamic stiffness matrix is formulated based on the exact solutions of the differential equations of motion governing the free vibration of generally laminated composite beam.

Gurban and Gupta [4] analyzed the natural frequencies of composite tubular shafts using equivalent modulus beam theory (EMBT) with shear deformation, rotary inertia and gyroscopic effects has been modified and used for the analysis. The modifications take into account effects of stacking sequence and different coupling mechanisms present in composite materials. Results obtained have been compared with that available in the literature using different modeling. The close agreement in the results obtained clearly show that, in spite of its simplicity, modified EMBT can be used effectively for rotor-dynamic analysis of tubular composite shafts.

Yegao *et al*.[5] presented a general formulation for free and transient vibration analysis of composite laminated beams with arbitrary lay ups and any boundary conditions. A modified variational principle combined with a multi-segment partitioning technique is employed to derive the formulation based on a general higher order shear defomation theory. The material coupling for bending-stretching, bending-twist, and stretching twist as well as the poison’s effect are taken into account.

Shell Structures

Qu et al. [6] introduced a variational formulation for predicting the free, steady-state and transient vibrations of composite laminated shells of revolution subjected to various combinations of classical and non-classical boundary conditions. A modified variational principle in conjunction with a multi-segment partitioning technique was employed to derive the formulation based on the first-order shear deformation theory.

Xiang *et al.*[7] studied a simple yet accurate solution procedure based on the Haar wavelet discretization method (HWDM) is applied to the free vibration analysis of composite laminated cylindrical shells subjected to various boundary conditions. The Reissner–Naghdi’s shell theory is adopted to formulate the theoretical model. The initial partial differential equations (PDE) are first converted into system of ordinal differential equations by the separation of variables. Then the discretizations of governing equations and corresponding boundary conditions are implemented by means of the HWDM, which leads to a standard linear eigenvalue problem.

Plates

Sahoo and Singh [8] proposed a new trigonometric zigzag theory for the static analysis of laminated composite and sandwich plates. This theory considers shear strain shape function assuming the non-linear distribution of in-plane displacement across the thickness. It satisfies the shear-stress-free boundary conditions at top and bottom surfaces of the plate as well as the continuity of transverse shear stress at the layer interfaces obviating the need of an artificial shear correction factor.

Rarani *et al*. [9] used analytical and finite element methods for prediction of buckling behavior, including critical buckling load and modes of failure of thin laminated composites with different stacking sequences. A semi-analytical Rayleigh–Ritz approach is first developed to calculate the critical buckling loads of square composite laminates with SFSF (S: simply-support, F: free) boundary conditions. Then, these laminates are simulated under axially compression loading using the commercial finite element software, ABAQUS. Critical buckling loads and failure modes are predicted by both eigenvalue linear and nonlinear analysis.

Alnefaie [10] developed a 3D-FE model of delaminated fiber reinforced composite plates to analyse their dynamics. Natural frequencies and modal displacements are calculated for various case studies for different dimensions and delamination characteristics. Numerical results showed a good agreement with available experimental data. A new proposed model shows enhancement of the accuracy of the results.

Sino *et al*. [11] worked on the dynamic instability of an internally damped rotating composite shaft. A homogenized finite element beam model, which takes into account internal damping, is introduced and then used to evaluate natural frequencies and instability thresholds. The influence of laminate parameters: stacking sequences, fiber orientation, transversal shear effect on natural frequencies and instability thresholds of the shaft are studied. The results are compared to those obtained by using equivalent modulus beam theory (EMBT), modified EMBT and layerwise beam theory (LBT).

Optimization issues and dynamics

Topal[12] presented a multiobjective optimization of laminated cylindrical shells to maximize a weighted sum of the frequency and buckling load under external load. The layer fiber orientation is used as the design variable and the multi-objective optimization is formulated as the weighted combinations of the frequency and buckling under external load. The first order shear deformation theory is used for the finite element formulation of the laminated shells. Five shell configurations with eight layers are considered as candidate designs. The modified feasible direction method (MFD) is used as optimization routine. Finally, the effect of different weighting ratios, shell aspect ratio, shell thickness-to-radius ratios and boundary conditions on the optimal designs is investigated and the results are compared.

Topal and Uzman[13] proposed a multiobjective optimization of symmetrically angle-ply square laminated plates subjected to biaxial compressive and uniform thermal loads. The design objective is the maximization of the buckling load for weighted sum of the biaxial compressive and thermal loads. The design variable is the fiber orientations in the layers. The performance index is formulated as the weighted sum of individual objectives in order to obtain optimal solutions of the design problem. The first-order shear deformation theory (FSDT) is used in the mathematical formulation of buckling analysis of laminated plates.

Roos and Bakis [14] analysed the flexible matrix composites which consist of low modulus elastomers such as polyurethanes which are reinforced with high-stiffness continuous fibers such as carbon. This fiber–resin system is more compliant compared to typical rigid matrix composites and hence allows for higher design flexibility. Continuous, single-piece FMC driveshafts can be used for helicopter applications. Authors employed an optimization tool using a genetic algorithm approach to determine the best combination of stacking sequence, number of plies and number of in-span bearings for a minimum-weight, spinning, and misaligned FMC helicopter driveshaft. In order to gain more insight into designing driveshafts, various loading scenarios are analyzed and the effect of misalignment of the shaft is investigated. This is the first time that a self-heating analysis of a driveshaft with frequency- and temperature-dependent material properties is incorporated within a design optimization model. For two different helicopter drivelines, weight savings of about 20% are shown to be possible by replacing existing multi-segmented metallic drivelines with FMC drivelines.

Sadr and Bargh [15] studied the fundamental frequency optimization of symmetrically laminated composite plates using the combination of Elitist- Genetic algorithm(E-GA) and finite strip method(FSM). The design variables are the number of layers, the fiber orientation angles, edge conditions and plate length/width ratios.

Kayikci and Sonmez [16] studied and optimized the natural frequency response of symmetrically laminated composite plates. An analytical model accounting for bending–twisting effects was used to determine the laminate natural frequency. Two different problems, fundamental frequency maximization and frequency separation maximization, were considered. Fiber orientation angles were chosen as design variables. Because of the existence of numerous local optimums, a global search algorithm, a variant of simulated annealing, was utilized to find the optimal designs. Results were obtained for different plate aspect ratios. Effects of the number of design variables and the range of values they may take on the optimal frequency were investigated. Problems in which fiber angles showed uncertainty were considered. Optimal frequency response of laminates subjected to static loads was also investigated.

Khandan *et al.*[17] researched and added an extra term to the optimisation penalty function in order to consider the transverse shear effect. This modified penalty function leads to a new methodology whereby the thickness of laminated plate is minimised by optimizing the fiber orientations for different load cases. Therefore the effect of transverse shear forces is considered in this study.

Montagnier and Hochard [18] studied the optimisation of hybrid composite drive shafts operating at subcritical or supercritical speeds, using a genetic algorithm. A formulation for the flexural vibrations of a composite drive shaft mounted on viscoelastic supports including shear effects is developed. In particular, an analytic stability criterion is developed to ensure the integrity of the system in the supercritical regime. Then it is shown that the torsional strength can be computed with the maximum stress criterion. A shell method is developed for computing drive shaft torsional buckling. The optimisation of a helicopter tail rotor driveline is then performed. This study yielded some general rules for designing an optimum composite shaft without any need for optimisation algorithms.

Rocha *et al*. [19] presented a genetic algorithm combining two types of computational parallelization methods, resulting in a hybrid shared/distributed memory algorithm based on the island model using both Open MP and MPI libraries. In order to take further advantage of the island configuration, different genetic parameters are used in each one, allowing the consideration of multiple evolution environments concurrently. To specifically treat composite structures, a three-chromosome variable encoding and special laminate operators are used. The resulting gains in execution time due to the parallel implementation allow the use of high fidelity analysis procedures based on the Finite Element Method in the optimization of composite laminate plates and shells. Two numerical examples are presented in order to assess the performance and reliability of the proposed algorithm.

Abadi and Daneshmehr [20] developed the buckling analysis of composite laminated beams based on modified coupled stress theory. By applying principle of minimum potential energy and considering two different beam theories ,i.e, Euler-Bernouli and Timoshinko beam theories, governing equations, boundary and initial conditions are derived for micro composite laminated beam.

Apalak *et al*. [21] carried-out the layer optimization for achieving maximum fundamental frequency of laminated composite plates under any combination of the three classical edge conditions. The optimal stacking sequences of laminated composite plates were searched by means of Genetic Algorithm. The first natural frequencies of the laminated composite plates with various stacking sequences were calculated using the finite element method. Genetic Algorithm maximizes the first natural frequency of the laminated composite plate defined as a fitness function (objective function).

In addition to above, numerous conference articles and textbooks emphasize the analysis issues of composite laminated structures.

1.3 Objectives

Based on the above literature, it is observed that enormous works concentrated on the classical lamination theory along with micro-mechanics models for material analysis. There is a necessity to develop interactive software which generalizes the analysis procedure at least in computing the elastic properties of overall laminated composite material. The following are the objectives of the present work:

1. Develop computer program to predict the overall elastic properties of multilayer composite material of given number of layers, stacking sequence and elastic constants of each layer. 2.Validate the elastic data with finite element modeling. 3. Implement the program in a graphic user interface. 4. Using equivalent modulus beam theory, estimate the natural frequencies and critical buckling load of beams.

The organization of the thesis is as follows: Chapter-2 gives mathematical modeling including classical lamination theory, finite element modeling and concept of equivalent modulus beam theory to compute natural frequencies and buckling loads. Chapter-3 presents a brief methodology of the present work. Chapter-4 illustrates the approach with numerical examples. Brief conclusions and scope for future work is presented in chapter-5.

Chapter2

Mathematical Modeling

This chapter presents the mathematical analysis used in conventional classical lamination theory and other analysis issues.

2.1 Hooke's Law

Generalized Hooke's law for orthotropic material is given by:

$$\{\sigma\}=[Q]\{\varepsilon\} \tag{2.1}$$

[Q] is called material stiffness matrix. For plane stress conditions, we can write for each layer:

$$\begin{Bmatrix}\sigma_{11}\\ \sigma_{22}\\ \sigma_{23}\\ \sigma_{13}\\ \sigma_{12}\end{Bmatrix}=\begin{bmatrix}Q_{11} & Q_{12} & 0 & 0 & 0\\ Q_{12} & Q_{22} & 0 & 0 & 0\\ 0 & 0 & Q_{44} & 0 & 0\\ 0 & 0 & 0 & Q_{55} & 0\\ 0 & 0 & 0 & 0 & Q_{66}\end{bmatrix}\begin{Bmatrix}\varepsilon_{11}\\ \varepsilon_{22}\\ \varepsilon_{23}\\ \varepsilon_{13}\\ \varepsilon_{12}\end{Bmatrix} \tag{2.2}$$

The elastic constants in the principal material coordinate system are: $Q_{11}=\frac{E_1}{1-\nu_{12}\nu_{21}}$,

$Q_{12}=\frac{\nu_{12}E_2}{1-\nu_{12}\nu_{21}}$, $Q_{22}=\frac{E_2}{1-\nu_{12}\nu_{21}}$, $Q_{44}=G_{23}$, $Q_{55}=G_{13}$, $Q_{66}=G_{12}$.

Here, E_1, E_2, G_{12}, G_{23}, G_{13} and ν_{12} are engineering parameters of the nth layer (lamina) in the laminate obtained from rule of mixtures. The transformed stress-strain relations for each lamina can be written as:

$$\begin{Bmatrix}\sigma_{xx}\\ \sigma_{yy}\\ \tau_{yz}\\ \tau_{xz}\\ \tau_{xy}\end{Bmatrix}=\begin{bmatrix}\overline{Q}_{11} & \overline{Q}_{12} & 0 & 0 & \overline{Q}_{16}\\ \overline{Q}_{21} & \overline{Q}_{22} & 0 & 0 & \overline{Q}_{26}\\ 0 & 0 & \overline{Q}_{44} & \overline{Q}_{45} & 0\\ 0 & 0 & \overline{Q}_{45} & \overline{Q}_{55} & 0\\ \overline{Q}_{16} & \overline{Q}_{26} & 0 & 0 & \overline{Q}_{66}\end{bmatrix}\begin{Bmatrix}\varepsilon_{xx}\\ \varepsilon_{yy}\\ \gamma_{yz}\\ \gamma_{xz}\\ \gamma_{xy}\end{Bmatrix} \tag{2.3}$$

The transformed reduced stiffness terms are given as follows:

$[\overline{Q}] = [T]^{-1}[Q][T]^{-1}$, where $[T]=\begin{bmatrix} c^2 & s^2 & -2cs \\ s^2 & c^2 & 2cs \\ cs & -cs & (c^2-s^2) \end{bmatrix}$ with c=cos(ϕ) and s=sin(ϕ). Also ϕ is fiber orientation angle for kth lamina with respect to x-axis of the beam.

2.2 Force, moment relations

Fiber-reinforced composite consisting of multiple layers of material is called *laminate*. Each layer is thin and may have a different fiber orientation. Two laminates may have the same number of layers and the same fiber angles but the two laminates may be different because of the arrangement of the layers. Figure 2.1 shows a global Cartesian coordinate system and a general laminate consisting of *N* layers.

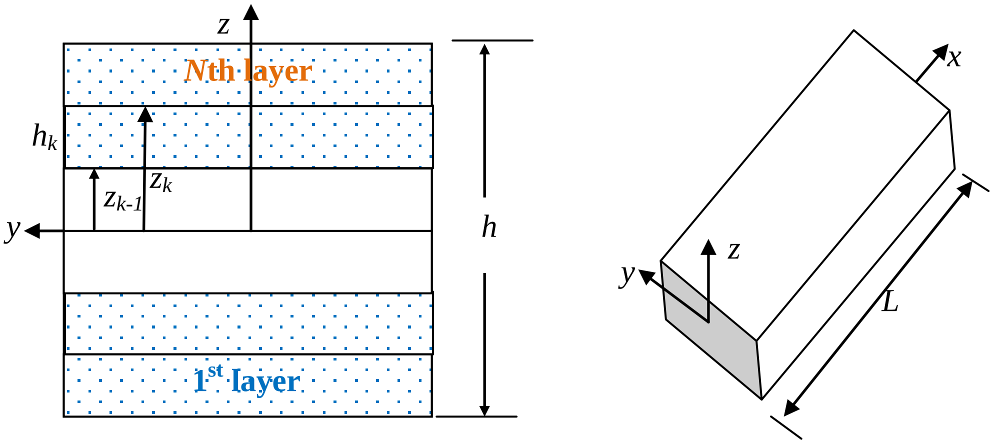


Fig.2.1 Geometry and coordinates of laminate

The laminate thickness is denoted by *h* and the thickness of a *k*th layer is h_k. The origin of the thickness coordinate, designated *z*, is located at the laminate geometric mid-plane. The geometric midplane may be within a particular layer or at an interface between layers. The laminate extends in the *z* direction from $-h/2$ to $+h/2$. The layer at the most negative location is as layer 1, the next layer in as layer 2, the layer at an arbitrary location is layer *k*, and the layer at the most positive *z* position is layer *N*. The locations of the layer interfaces are denoted by a subscripted *z*; The first layer is bounded by locations z_0 and z_1, the second layer by z_1 and z_2, the kth layer by z_{k-1} and z_k, and the Nth layer by z_{N-1} and z_N.

The important assumption of classical lamination theory is that each point within the volume of a laminate is in a state of plane stress. Therefore, stresses can be computed if we know the strains and curvatures of the reference surface. Given the force and moment resultants, we want to calculate the stresses and strains through the thickness as well as the strains and curvatures on the reference surface. We also want to do this by computing the laminate stiffness matrix. The force resultants N_x, N_y, and N_{xy} can be shown to be related to the mid-plane strains ε^0 and curvatures κ^0 at the reference surface by the following equation:

$$\begin{Bmatrix} N_x \\ N_y \\ N_{xy} \end{Bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{16} \\ A_{12} & A_{22} & A_{26} \\ A_{16} & A_{26} & A_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_x^0 \\ \varepsilon_y^0 \\ \gamma_{xy}^0 \end{Bmatrix} + \begin{bmatrix} B_{11} & B_{12} & B_{16} \\ B_{12} & B_{22} & B_{26} \\ B_{16} & B_{26} & B_{66} \end{bmatrix} \begin{Bmatrix} \kappa_x^0 \\ \kappa_y^0 \\ \kappa_{xy}^0 \end{Bmatrix} \tag{2.4}$$

Similarly, the moment resultants M_x, M_y, and M_{xy} can also be shown to be related to the strains and curvatures at the reference surface by the following equation:

$$\begin{Bmatrix} M_x \\ M_y \\ M_{xy} \end{Bmatrix} = \begin{bmatrix} B_{11} & B_{12} & B_{16} \\ B_{12} & B_{22} & B_{26} \\ B_{16} & B_{26} & B_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_x^0 \\ \varepsilon_y^0 \\ \gamma_{xy}^0 \end{Bmatrix} + \begin{bmatrix} D_{11} & D_{12} & D_{16} \\ D_{12} & D_{22} & D_{26} \\ D_{16} & D_{26} & D_{66} \end{bmatrix} \begin{Bmatrix} \kappa_x^0 \\ \kappa_y^0 \\ \kappa_{xy}^0 \end{Bmatrix} \tag{2.5}$$

where matrix [A], [B] and [D] are given by

$$A_{ij} = \sum_{k=1}^{N} (\overline{Q}_{ij})_k \left(z_k - z_{k-1}\right) \tag{2.6}$$

$$B_{ij} = \frac{1}{2} \sum_{k=1}^{N} (\overline{Q}_{ij})_k \left(z_k^2 - z_{k-1}^2\right) \tag{2.7}$$

$$D_{ij} = \frac{1}{3} \sum_{k=1}^{N} (\overline{Q}_{ij})_k \left(z_k^3 - z_{k-1}^3\right) \tag{2.8}$$

The eqs. can be combined as

$$\begin{Bmatrix} \mathrm{N_x} \\ \mathrm{N_y} \\ \mathrm{N_{xy}} \\ \mathrm{M_x} \\ \mathrm{M_y} \\ \mathrm{M_{xy}} \end{Bmatrix} = \begin{bmatrix} [\mathrm{A}] & [\mathrm{B}] \\ [\mathrm{B}] & [\mathrm{D}] \end{bmatrix} \begin{Bmatrix} \varepsilon_{\mathrm{x}}^0 \\ \varepsilon_{\mathrm{y}}^0 \\ \gamma_{\mathrm{xy}}^0 \\ \kappa_{\mathrm{x}}^0 \\ \kappa_{\mathrm{y}}^0 \\ \kappa_{\mathrm{xy}}^0 \end{Bmatrix} \tag{2.9}$$

Note that the inverse of [A] matrix relates strains with forces according to $\{\varepsilon\}=[a]\{N\}$, where $[a]=[A]^{-1}$. A laminate is *symmetric* if for every layer to one side of the laminate reference surface (with a specific thickness, specific material properties and specific fiber orientation), there is another layer at the same distance on the opposite side of the reference surface (with the same thickness, material properties and fiber orientation). If the laminate is not symmetric, then it is referred to as an *unsymmetric* laminate. For a symmetric laminate, all the elements of the [B] matrix are identically zero. In engineering design, the stacking sequence of laminated plates is designed to be symmetric and balanced to avoid unpredictable warp deflections.

2.3 Effective elastic constants

We assume that the effective material properties of each ply can be expressed in terms of a micromechanical model using rule of mixture:

$$E_{11}=V_f E_{11}^f+V_m E_m \tag{2.10}$$

$$\frac{1}{E_{22}}=\frac{V_f}{E_{22}^f}+\frac{V_m}{E^m}-V_f V_m \frac{\upsilon_f^2 E_m / E_{22}^f+\upsilon_m^2 E_{22}^f / E_m-2\upsilon_f \upsilon_m}{V_f E_{22}^f+V_m E_m} \tag{2.11}$$

$$\frac{1}{G_{ij}}=\frac{V_f}{G_{ij}^f}+\frac{V_m}{G^m} \text{ (ij=12,13 and 23)} \tag{2.12}$$

$$\nu_{12}=V_f \nu_f+V_m \nu_m \tag{2.13}$$

$$\rho=V_f \rho_f+V_m \rho_m \tag{2.14}$$

where E_{11}^f, E_{22}^f, G_{12}^f, G_{13}^f, G_{23}^f, ν_f, and ρ_f are the Young′s moduli, shear moduli, Poisson′s ratio and mass density, respectively, of the fiber, while E_m, G_m, ν_m and ρ_m are the corresponding properties for the matrix, respectively. V_{f} and V_{m} are the fiber and matrix volume fractions and must satisfy the unity condition of $V_{\text{f}}+V_{\text{m}}=1$.

The concept of *effective elastic constants* for the laminate is useful to idealize the system as equivalent isotropic material. These constants are the effective extensional modulus in the *x* direction $\bar{E}_x$, the effective extensional modulus in the *y* direction $\bar{E}_y$, the effective Poisson's ratios $\bar{\nu}_{xy}$ and $\bar{\nu}_{\text{yx}}$ and the effective shear modulus in the *x*-*y* plane $\bar{G}_{xy}$. The effective elastic constants are usually defined when considering the in-plane loading of symmetric balanced laminates. The following three average laminate stresses are defined:

$$\bar{\sigma} = \frac{1}{h}\int_{-h/2}^{h/2} \sigma_x dz \tag{2.15}$$

$$\bar{\sigma}_y = \frac{1}{h}\int_{-h/2}^{h/2} \sigma_y dz \tag{2.16}$$

$$\bar{\tau}_{xy} = \frac{1}{h}\int_{-h/2}^{h/2} \tau_{xy} dz \tag{2.17}$$

Using the above relations we obtain the relation between the average stresses and resultant forces as follows:-

$$\bar{\sigma}_x = \frac{1}{h} N_x \tag{2.18}$$

$$\bar{\sigma}_y = \frac{1}{h} N_y \tag{2.19}$$

$$\bar{\tau}_{xy} = \frac{1}{h} N_{xy} \tag{2.20}$$

Hence, the laminate compliance matrix of 3×3 size can be defined as follows [20]:-

$$\begin{Bmatrix} \varepsilon_x^0 \\ \varepsilon_y^0 \\ \gamma_{xy}^0 \end{Bmatrix} = \begin{pmatrix} a_{11}h & a_{12}h & 0 \\ a_{21}h & a_{22}h & 0 \\ 0 & 0 & a_{66}h \end{pmatrix} \begin{Bmatrix} \bar{\sigma}_x \\ \bar{\sigma}_y \\ \bar{\tau}_{xy} \end{Bmatrix} \tag{2.21}$$

By comparing with stress-strain relations, we obtain the elastic constants for the laminate as follows:-

$$\bar{E}_x = \frac{1}{a_{11}h} \tag{2.22}$$

$$\bar{E}_y = \frac{1}{a_{22}h} \tag{2.23}$$

$$\bar{G}_{xy} = \frac{1}{a_{66}h} \tag{2.24}$$

$$\bar{\nu}_{xy} = -\frac{a_{12}}{a_{11}} \tag{2.25}$$

$$\bar{\nu}_{yx} = -\frac{a_{12}}{a_{22}} \tag{2.26}$$

We can observe that $\bar{\nu}_{yx}$ and $\bar{\nu}_{xy}$ are not independent of each other and their reciprocity relation can be shown as below:

$$\bar{\nu}_{xy} = \bar{E}_x \frac{\bar{\nu}_{yx}}{\bar{E}_Y} \tag{2.27}$$

2.4 Finite element modeling of laminated composite

Finite element model of composite laminated structure discretizes the entire thickness along the linear direction into number of elements. Often 2D-modeling is sufficient for getting accurate results. The shell elements are the famous 2D discretisation elements. A shell element has n-nodes with each node having 6 DOFs. The present work FEM is merely employed for the verification of elastic constants obtained from classical beam theory. In addition the stresses and strain at the each layer and the interfaces can be addressed. An eight-nodded quadrilateral C^0 continuous isoparametric shell element (SHELL 281) with six-degrees-of-freedom per node (u_x, uy, uz, θx, θy, θz) is employed. The generalized displacements included are expressed as:

$$\delta = \sum_{i=1}^{n} N_i \delta_i \tag{2.28}$$

where, N_i is the shape function associated with the node i and n is the number of nodes per element, which is eight in present study. The strain vector $\{\varepsilon\}$ can be expressed in terms of δ containing nodal degrees of freedom as,

$$\{\varepsilon_b^k\}=[B_b^k]\{\delta\}$$

$$\{\gamma_s^k\}=[B_s^k]\{\delta\}, \tag{2.29}$$

where $[B]$ is the strain–displacement matrix in the Cartesian coordinate system. The $[B]$ matrix can be divided in two parts, one which contains the bending terms and other containing the shear terms. ANSYS SHELL281 is suitable for analyzing thin to moderately-thick shell structures. SHELL281 may be used for layered applications for modeling composite shells or sandwich construction. The accuracy in modeling composite shells is governed by the first-order shear-deformation theory (usually referred to as Mindlin-Reissner shell theory). Fig.2.2 shows the element description.

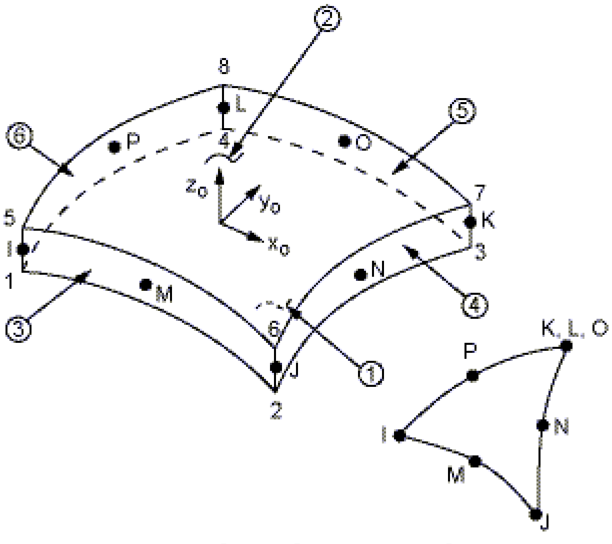


Fig.2.2 Eight node SHELL element

The element is defined by shell section information and by eight nodes (I, J, K, L, M, N, O and P). The shell section commands allow for layered shell definition. Options are available for specifying the thickness, material, orientation and number of integration points through the thickness of the layers. You can designate the number of integration points (1, 3, 5, 7, or 9) located through the thickness of each layer when using section input. SHELL281 includes the effects of transverse shear deformation.

2.5 Dynamic analysis

Modal analysis of composite structure will be carried out as well as the natural frequencies and mode shapes of homogenized equivalent isotropic characteristics with different boundary conditions are obtained. For a beam structure, we can write axial and transverse displacements as

$$u = u_0 + z\theta$$

$$v = z\Psi$$

$$w = w_0 \tag{2.30}$$

Here, θ and ψ are rotations of normal to mid-plane. As for a beam N_y=N_{xy}=M_y=0, the constitutive equations of laminate from classical laminate theory becomes

$$\begin{Bmatrix} N_x \\ M_y \\ M_{xy} \end{Bmatrix} = \begin{pmatrix} A_{11} & B_{11} & B_{16} \\ B_{11} & D_{11} & D_{11} \\ D_{16} & D_{16} & D_{66} \end{pmatrix} \begin{Bmatrix} \varepsilon_x^0 \\ \kappa_x^0 \\ \kappa_{xy}^0 \end{Bmatrix} \tag{2.31}$$

Transverse shear force / unit length

$$Q_{xz} = A_{55}\gamma_{xz} \tag{2.32}$$

where $A_{55} = k\int_{-h/2}^{h/2} \bar{Q}_{55} dz$ with k as shear correction factor and $\bar{Q}_{55} = G_{13}\cos\theta + G_{23}\sin\theta$.

Total strain energy of beam

$$V = \frac{1}{2}\int_0^L (N_x\varepsilon_x^0 + M_x\kappa_x + M_{xy}\kappa_{xy} + Q_{xz}\gamma_{xz})bdz$$

$$= \frac{1}{2}\left[\int_0^L \mathrm{EI}\left\{\left(\frac{\partial\theta_y}{\partial x}\right)^2 + \left(\frac{\partial\theta_z}{\partial x}\right)^2\right\}dx + \int_0^L \kappa \mathrm{GA}\left\{\left(\frac{\partial v}{\partial x} - \theta_z\right)^2 + \left(\frac{\partial w}{\partial x} + \theta_y\right)^2\right\}dx\right] \tag{2.33}$$

where,

$$EI=\sum_{i=1}^{N} E_x^i \pi\left(\frac{R_i^4-R_{i-1}^4}{4}\right) \tag{2.34}$$

$$GA=\sum_{i=1}^{N} G_{xy}^i \pi(R_i^2-R_{i-1}^2) \tag{2.35}$$

are the homogenized flexural and torsional rigidities and R_i and R_{i-1} are external and internal radius of layer *i*.

Total kinetic energy is

$$T=\frac{1}{2}\int_0^L\int_{-h/2}^{h/2}\rho[(\frac{\partial y}{\partial t})^2+(\frac{\partial v}{\partial t})^2+(\frac{\partial w}{\partial t})^2]bdzdx$$

$$=\frac{1}{2}\left[\int_0^L\left\{\rho A\left\{\dot{v}^2+\dot{w}^2\right\}+I_D\left(\dot{\theta}_y^2+\dot{\theta}_z^2\right)+I_P\left(\Omega^2+\Omega\left(\dot{\theta}_z\theta_y-\dot{\theta}_y\theta_z\right)\right)\right\}dx\right] \tag{2.36}$$

Here, I_D and I_P are transverse and polar mass moments of inertia of the shaft. A simply supported composite shaft is shown in Fig.2.3.

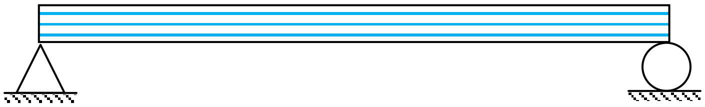


Fig.2.3 Laminated composite simply-supported beam

In present work, the fundamental natural frequency of simply supported beam is computed for only Euler-Bernouli beam model as follows:

$$\omega_n^2=\frac{\alpha^2 n^4\pi^4}{\ell^2}\text{- simply supported beam}$$

$$\omega_n^2=\frac{\alpha^2(\beta_n\ell)^4}{\ell^2}\text{-- cantilever beam} \tag{2.37}$$

where $\alpha^2=\frac{EI}{\rho A}$ with E= $\overline{E}_x$ obtained from effective elastic constant result. If first natural frequency is required n=1 and β*l*=1.8751. Also, I and A are moment of inertia and cross-section.

Chapter 3

Methodology

The following methodology is being adopted to carry out the above mentioned objectives:

1. An interactive interface is created using GUI in MATLAB to compute the overall laminate properties of the composite.
2. Using ANSYS the overall material properties are computed and tried to validate with classical theory.
3. Using these equivalent properties of the composite the natural frequency computations are done.

Fig.3.1 shows present methodology.

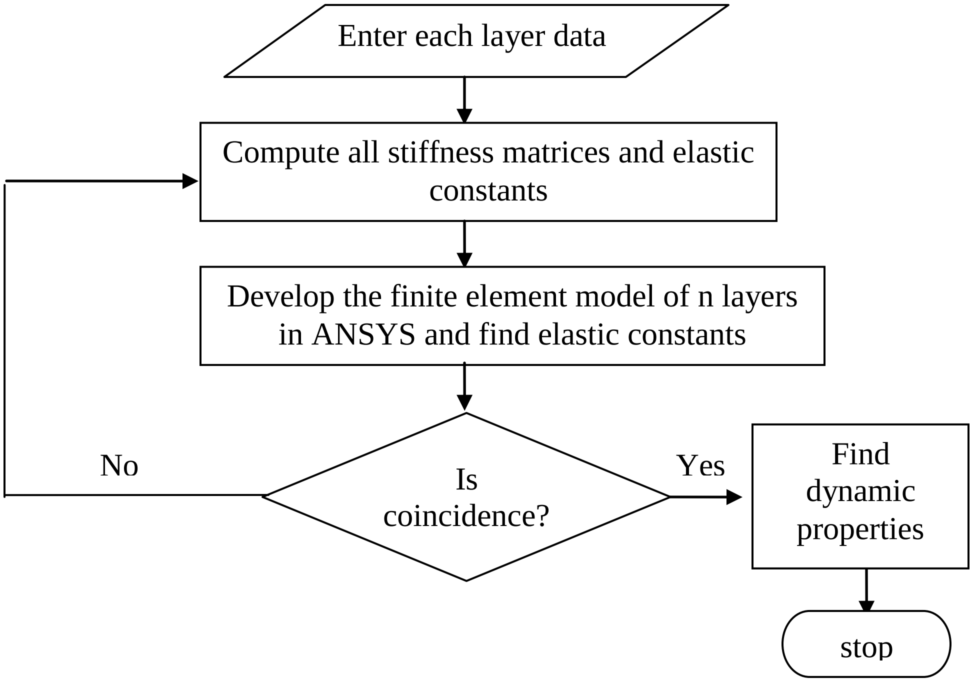

Enter each layer data
Compute all stiffness matrices and elastic constants
Develop the finite element model of n layers in ANSYS and find elastic constants
Is coincidence?
No
Yes
Find dynamic properties
stop

Fig.3.1 Flowchart of present methodology

3.1 Development of GUI

The steps employed for the calculation of overall laminate properties using MATLAB by creating the GUI are given as follows:

Program Description

The program makes use of the user input elastic properties (E1, E2, G12, NU12) of a single ply material in its given principal directions (1, 2, and 3) as well as the ply geometry and also tacking sequence to constructs the [A], [B] and [D] matrices of a laminated fiber-reinforced composite. Using these [A], [B] and [D] matrices, it finds the overall laminate elastic properties (Ex, Ey, Gxy, NUxy, etc). A fundamental understanding of composite laminates and its basic theories are expected in understanding the concepts and results presented.

System requirements

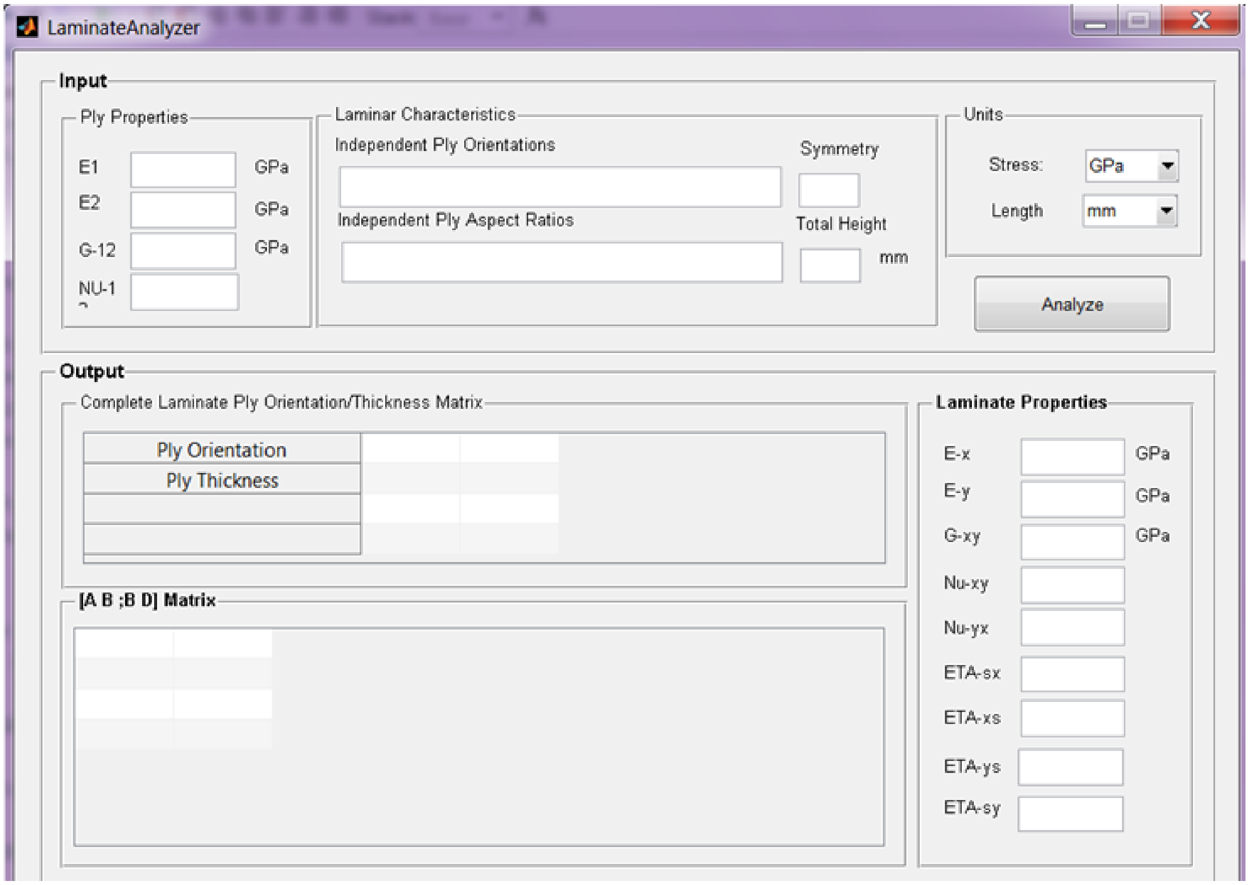
The program needs a computer that has MATLAB installed in it. The M-file *LaminateAnalyzer.m* can be opened and executed using the run command which will pop up the Graphical User Interface. The relevant Matlab code is presented at the appendix.

Program Functionality

Currently the program enclosed is incapable of analyzing symmetric-balanced laminates of varying thicknesses about the mid-plane. While the laminates with asymmetric stacking sequences with uneven number of plies can be analyzed with modification of the program.

Variables

Ex , Ey ,E1 , E2 - Young’s Modulus in the x- , y-, 1- , and 2- directions respectively NUxy , U12 – Poisson’s ratios for x-y and 1-2 directions Gxy – Shear modulus referred to the x- and y- axes ETAsx , ETAsy – Shear coupling coefficients. Fig.3.2 shows the GUI environment.



Stress: GPa

Length: mm

Fig. 3.2 Interface environment

3.2 Overall material properties using ANSYS

In order to predict elastic modulus in longitudinal direction, the laminated model is loaded in axial tension by applying a small normal displacement at one side and fully restraining the other side. For example, the x = 0 end is constrained in the axial direction (x direction) and free to move in the lateral directions as shown in Fig.3.3. The free edges are constrained to their respective normal directions in order to allow contraction of the model due to tension. An axial displacement, equivalent to the approximate less than 10% of total length of plate, is applied to all nodes on the end surface (x = L), where L is the length of the laminated structure. The displacement and reaction forces are calculated on the data collection surfaces (i.e. constraint area). Then, the values of these displacements and reaction forces are employed to evaluate the effective elastic properties of the composites.

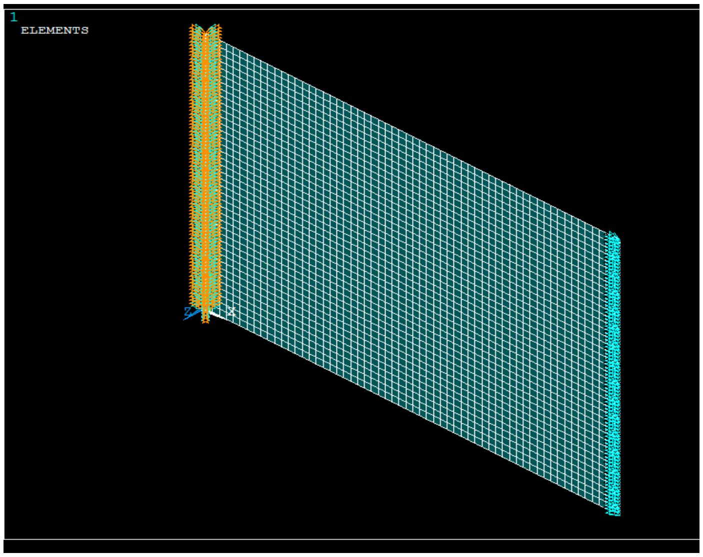


Fig.3.3 Methodology of predicting longitudinal modulus

The reaction forces calculated on constraint surface gives average stress on that surface. A far field uni-axial displacement is applied to the finite domain only in the longitudinal direction (the x-direction). The modulus of the laminated composite is estimated using the displacement and average stress result at data collection surfaces normal to the x-axis by the following formula:

$$E_X = \frac{\bar{\sigma}_x}{\varepsilon_x} = \frac{L}{(\Delta U)_{avg}}\bar{\sigma}$$

where U is the displacement applied and the average value of stress on a surface is given by

$$\bar{\sigma} = \frac{1}{A}\int_A \sigma_x((z=0), x, y)dxdy$$

where A is the area of constrained surface at L=0 with the help of FEM results of average stress can be evaluated for the laminated composite. The laminated structure is subjected to uniform extension within the linear region of the stress -strain curve.

We use

$$\varepsilon_y = -\frac{\nu_{xy}}{E_x}\sigma_x = -\nu_{xy}\frac{\Delta L}{L} = \frac{\Delta b}{b}$$

or

$$\nu_{xy} = \frac{\Delta b / b}{\Delta L / L}$$

where L and b are the length and width of laminated structure respectively.

3.3 Calculation of natural frequencies

Then the static and dynamic analysis of the composite is done as follows:

The dynamic analysis of laminated composite rotating shaft with the given fiber orientations and different boundary fixations, are investigated analytically. The equivalent modulus beam theory is employed for simply-supported and cantilever boundary conditions. The equivalent stiffness and mass of laminated composite laminate that have been derived by using the lamination theory is used here to compute the natural frequencies.

In equivalent modulus beam theory (EMBT), the equivalent longitudinal and in-plane shear modulii are determined using Classical Laminate Theory (CLT). These modulii are used to calculate shaft natural frequencies using beam theory in the same manner as that for isotropic shafts. Following are limitations of this theory: In multilayered composite shaft, different layers (plies) have different contributions to the overall stiffness of the shaft depending on their locations from the mid-plane. For unbalanced configuration, shear–normal and bending–twisting couplings are present. These couplings affect significantly shaft natural frequencies. However, these effects are not incorporated in EMBT formulation. In unsymmetric configurations, bending–stretching coupling is present which affects the shaft natural frequencies. This effect is also not included in the EMBT formulation.

Chapter 4

Results and Discussion

This chapter deals with the output results for some existing cases in literature.

4.1 Graphical User Interface (GUI) functionality

In order to illustrate the methodology, we considered two cases of laminates. In the first case, carbon-epoxy with 8 layers having a stacking sequence of $[0/45/-45/90]_s$ is considered.

Inputs:

Units can be selected based on user requirement whether in GPa or Psi for stress, and mm or inches for length.

Ply Elastic Properties: E-1, E-2, G-12, and NU-12 (in the relevant units)

Laminate Characteristics:

a. Orientations: enter the angles of orientations of each independent ply starting from the bottom to top; make sure to separate each angle by a space or a comma.

b. Symmetry: leave this as zero if the laminate is asymmetric or enter the value of symmetry.

Ex: (0/90) sym=0 , (0/90)s sym=1, (0/45)4s sym=4

c. Aspect ratio: give the aspect ratio of each independent ply.

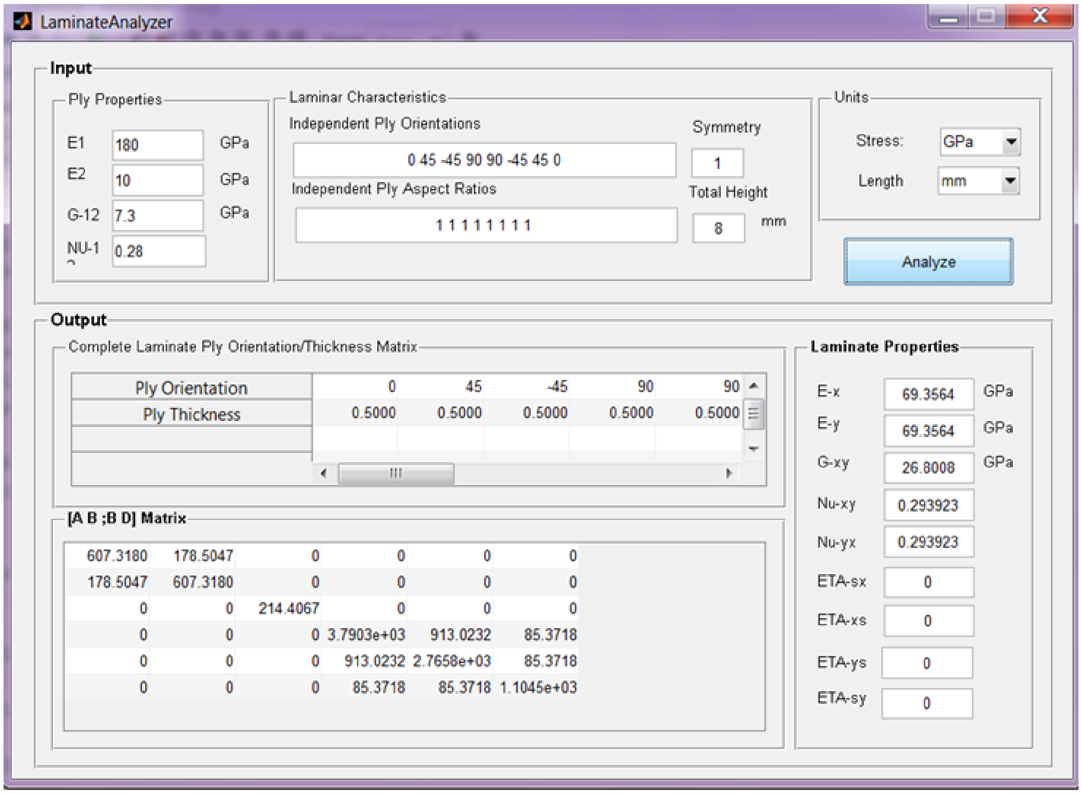
d. Total Height: Enter the total height of the laminate (not the ply).

The program computes the stiffness matrices and overall elastic properties based on the relations given in earlier chapter. Fig.4.1 shows the screen-shot of the GUI where the data is entered accordingly.

Outputs:

There are 3 different outputs that portray the overall characteristics of the laminate being analyzed as seen from Fig.4.1.

These are: ply data, [A], [B] and [D] matrices and overall elastic properties.



E1: 180 GPa
E2: 10 GPa
G-12: 7.3 GPa
NU-12: 0.28

Independent Ply Orientations: 0 45 -45 90 90 -45 45 0
Independent Ply Aspect Ratios: 1 1 1 1 1 1 1 1
Symmetry: 1
Total Height: 8 mm
Stress: GPa
Length: mm

Ply Orientation	0	45	-45	90	90
Ply Thickness	0.5000	0.5000	0.5000	0.5000	0.5000

607.3180	178.5047	0	0	0	0
178.5047	607.3180	0	0	0	0
0	0	214.4067	0	0	0
0	0	0	3.7903e+03	913.0232	85.3718
0	0	0	913.0232	2.7658e+03	85.3718
0	0	0	85.3718	85.3718	1.1045e+03

E-x: 69.3564 GPa
E-y: 69.3564 GPa
G-xy: 26.8008 GPa
Nu-xy: 0.293923
Nu-yx: 0.293923
ETA-sx: 0
ETA-xs: 0
ETA-ys: 0
ETA-sy: 0

Fig. 4.1 The Carbon/Epoxy example carried out in the GUI Interface

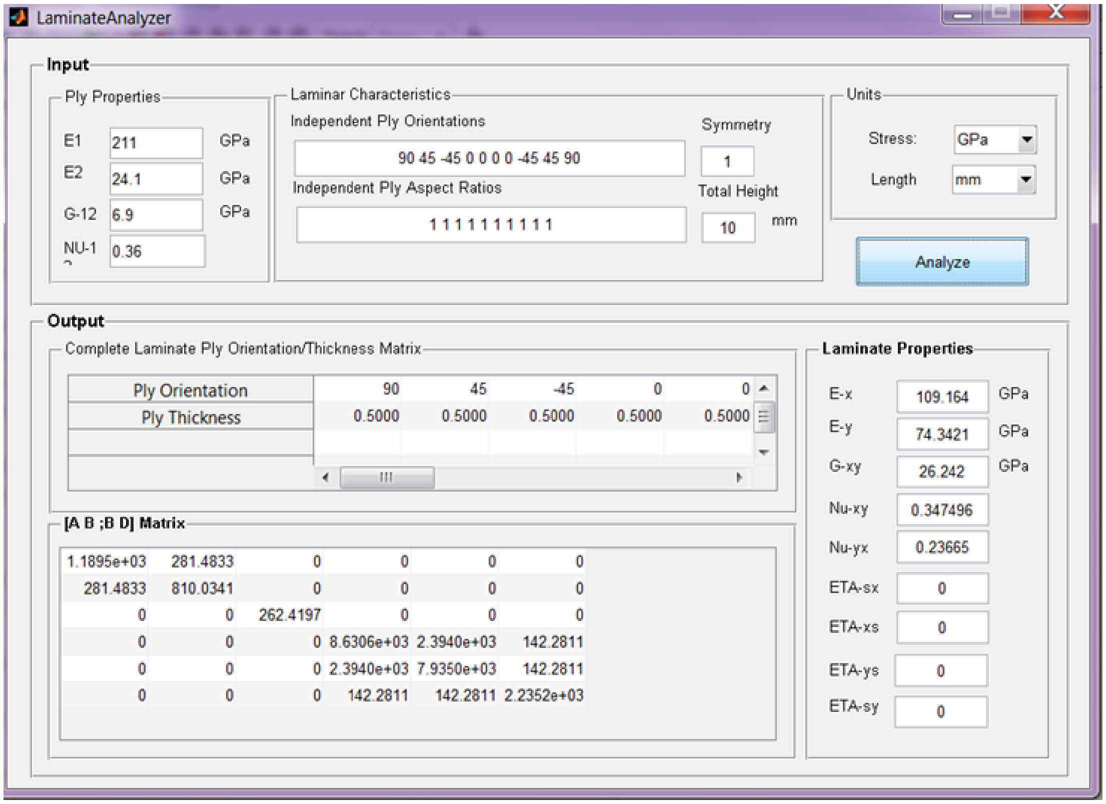
It is seen that last 4 constants are not accounted due to transverse isotropic approximation. Also, we can observe in the [A] matrix, $A_{16}=A_{26}=0$ with [B]=0, indicating balanced-symmetric laminate condition.

Case-2: Boron-epoxy composite

The following properties of the layer are considered:

Material	density(kg/m^3)	E_{11} (GPa)	E_{22}(GPa)	G_{12}(GPa)	ν_{12}	t_{ply}
Narmaco 5505	1965	211	24.1	6.9	0.36	1mm

Fig.4.2 shows the outputs computed from the GUI platform.



Ply Properties

E1: 211 GPa
E2: 24.1 GPa
G-12: 6.9 GPa
NU-12: 0.36

Laminar Characteristics

Independent Ply Orientations: 90 45 -45 0 0 0 0 -45 45 90
Independent Ply Aspect Ratios: 1 1 1 1 1 1 1 1 1 1
Symmetry: 1
Total Height: 10 mm

Units

Stress: GPa
Length: mm

Complete Laminate Ply Orientation/Thickness Matrix

Ply Orientation	90	45	-45	0	0
Ply Thickness	0.5000	0.5000	0.5000	0.5000	0.5000

[A B ;B D] Matrix

1.1895e+03	281.4833	0	0	0	0
281.4833	810.0341	0	0	0	0
0	0	262.4197	0	0	0
0	0	0	8.6306e+03	2.3940e+03	142.2811
0	0	0	2.3940e+03	7.9350e+03	142.2811
0	0	0	142.2811	142.2811	2.2352e+03

Laminate Properties

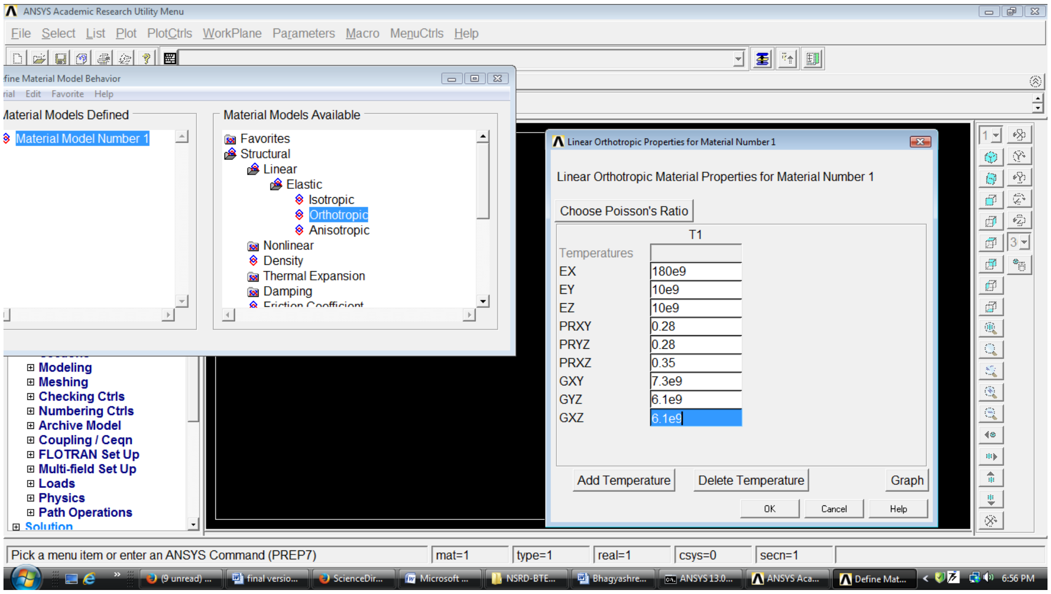
E-x: 109.164 GPa
E-y: 74.3421 GPa
G-xy: 26.242 GPa
Nu-xy: 0.347496
Nu-yx: 0.23665
ETA-sx: 0
ETA-xs: 0
ETA-ys: 0
ETA-sy: 0

Fig.4.2 Second example of 10-layered graphite epoxy composite

Obviously longitudinal modulus is 109.164 GPa.

4.2 Development of finite element model in ANSYS

In order to verify the longitudinal modulus obtained from classical lamination theory, a finite element model is developed in ANSYS software. Fig.4.3 shows the screen shot of material data entered for case-1 (Carbon-epoxy) and Fig.4.4 shows the corresponding ply data (thickness and orientation). A plate is created and is meshed with these elements (SHELL 281) and further applied with a constant displacement at one end and other end is fixed. The reaction forces (stresses) are computed at fixed end and finally the longitudinal modulus and Poisson ratio are estimated. Similarly other elastic constants can be computed. As in the present work, we employed only EMBT, other moduli are not required to validate.



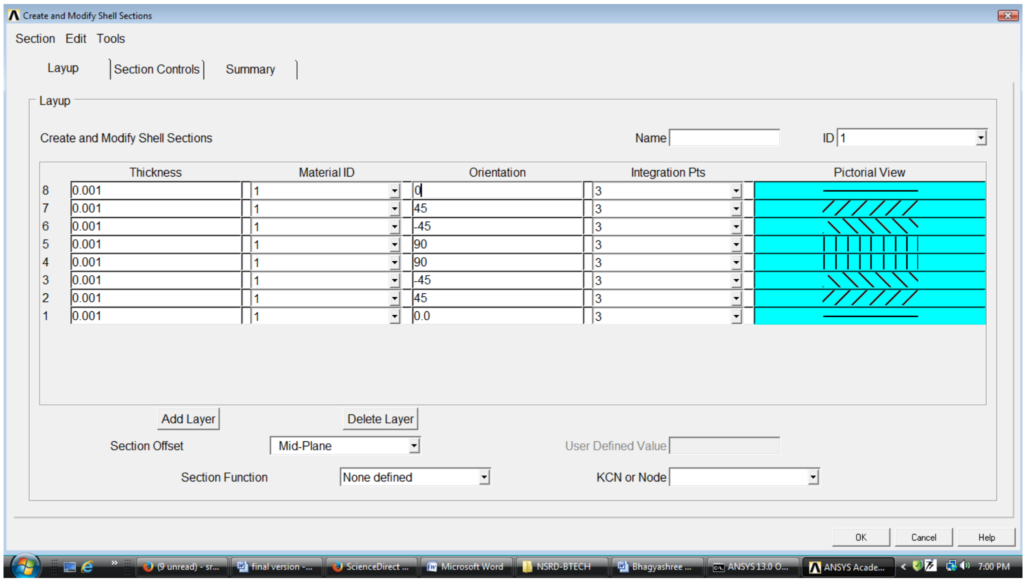
Linear Orthotropic Material Properties for Material Number 1

	T1
Temperatures	
EX	180e9
EY	10e9
EZ	10e9
PRXY	0.28
PRYZ	0.28
PRXZ	0.35
GXY	7.3e9
GYZ	6.1e9
GXZ	6.1e9

Pick a menu item or enter an ANSYS Command (PREP7)

mat=1 type=1 real=1 csys=0 secn=1

Fig.4.3 Screen shot of material data



Create and Modify Shell Sections

Name

ID 1

	Thickness	Material ID	Orientation	Integration Pts
8	0.001	1	0	3
7	0.001	1	45	3
6	0.001	1	-45	3
5	0.001	1	90	3
4	0.001	1	90	3
3	0.001	1	-45	3
2	0.001	1	45	3
1	0.001	1	0.0	3

Section Offset: Mid-Plane

User Defined Value

Section Function: None defined

KCN or Node

Fig.4.4 Screen shot showing ply data

Table-1 shows the comparison of the longitudinal modulus obtained from classical theory and using finite element modeling with ANSYS.

Table-1 Longitudinal Elastic modulus obtained by GUI and through ANSYS

Constant computed	Classical lamination theory (GUI)	Finite element model (ANSYS)
E_x (GPa)	69.69	72.43
ν_{xy}	0.293	0.356

4.3 Natural frequencies and Critical Buckling loads

To utilize the longitudinal modulus data for frequency computations, first case (carbon-epoxy) is only illustrated. Substituting the following values in eqs.(2.37) as: E=E_x=69.69 GPa, I=$bh^3/12$=0.00000032 kgm^4,with b=0.004m, h=8 mm, l=1 m, ρ=1680 kgm^{-3} and A=area=0.00024m^2, we get the natural frequencies and crucial buckling load as follows:

ω_n =600 kHz for simply supported beam

ω_n =19.446 kHz for cantilever beam

The critical buckling load can also be calculated as:

$$P_{cr} = EI\frac{\pi^2}{l^2} = 249 \text{ kN}$$

The fiber orientation in one of the layer of balanced-symmetrical laminate is varied and corresponding changes in elastic modulus are noted and the fundamental natural frequency variation of cantilever beam is observed as shown in Fig.4.5.

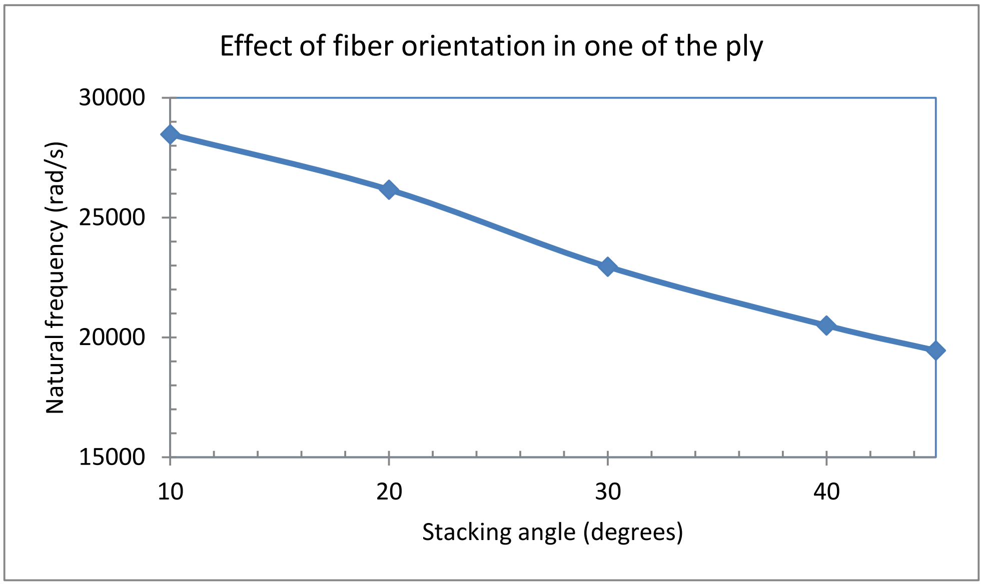


Fig.4.5 Variation of first frequency with fiber orietation

It has drastic effect on natural frequencies. The same thing can be illustrated for case-2 also.

Chapter5

Conclusions

In the present work, general classical lamination theory has been employed to predict the stiffness matrices connecting the forces and strains as well as moments and curvatures. The methodology was generalized by using a graphic user interface (GUI). At the back of this, the code employs all the classical relations and finally displays the stiffness matrices as well as the overall elastic constants of the laminate. As a next step, these values were validated using finite element modeling in commercial ANSYS software.

The concept of equivalent modulus beam theory introduced early 1990s for Euler-Bernoulli beams has been employed to obtain the fundamental natural frequencies for two end conditions. This method involves calculating the eigenvalues of the isotropic Bernoulli beam, using the longitudinal modulus of the composite material computed with the classical laminate theory. This theory is applicable for symmetric balanced laminates, but the EMBT approach has proved to have some limitations in the case of unbalanced and unsymmetrical laminates. The EMBT does not take into account the ply location relative to the axis when dealing with multilayered unsymmetrical laminates. Also, EMBT does not take shear–normal coupling in to account in the case of unbalanced laminates, or bending– stretching and bending–twisting coupling in that of unsymmetrical laminates. So, as a future scope, a more generalized solution approach based on finite element modeling can be also planned to display as a contour plot in GUI.

References

1. M. Hajianmaleki and M. S. Qatu**, '**Vibrations of straight and curved composite beams: A review', Composite Structures, Vol. 100, pp 218-232,2013**.**
2. V.Yildirim, 'Effect of the longitudinal to transverse moduli ratio on the in-plane natural frequencies of symmetric cross-ply laminated beams by the stiffness method', Composite structures, vol.50, pp.319-326, 2000.
3. L. Jun, H.Hongxing and S.Rongying, 'Dynamic finite element method for generally laminated composite beams', Int. J.Mechanical Sciences, vol.50, pp.466-480, 2008.
4. H.B.H.Gubran and K.Gupta, 'The effect of stacking sequence and coupling mechanisms on the natural frequencies of composite shafts', J.Sound and Vibration, vol.282, pp.231-248, 2005.
5. Yegao Qu, Xinhua Long, Hongguang Li, Guang Meng, 'A variational formulation for dynamic analysis of composite laminated beams on a general higher-order shear deformation theory', Composite Structures, Vol. 102 ,175–192, 2013.
6. Y.Qu, X. Long, S.Wu and G.Meng, 'A unified formulation for vibration analysis of composite laminated shells of revolution including shear deformation and rotary inertia', Composite structures, vol.98, pp.169-191, 2013.
7. Xiang Xie, Guoyong Jin, Yuquan Yan, S.X. Shi, Zhigang Liu, 'Free vibration analysis of composite laminated cylindrical shells using the Haar wavelet method', Composite Structures, vol. 109, pp 167-199,2014.
8. R.Sahoo and B.N. Singh, 'A new trigonometric zigzag theory for static analysis of laminated composite and sandwich plates', Aerospace science and technology, vol.35, pp.15-28, 2014.

9. M. H.Rarani,S .S. K.Sharifi and M.M. Shokrieh, ‘Effect of ply stacking sequence on buckling behavior of E-glass/epoxy laminated composites’, Computational Material Science, vol.89, pp.89-96, 2014.
10. K Alnefaie, ‘Finite Element Modeling of composite plates with internal delamination’, Composite Structures, Vol. 90, pp 21-27, 2009.
11. R. Sino a, T.N. Baranger, ‘Dynamic analysis of a rotating composite shaft’, Composites Science and Technology, Vol. 68 337–345, 2008.
12. U.Topal, ‘Multiobjective optimization of laminated composite cylindrical shells for maximum frequency and buckling load’, Materials & Design**,** vol.30, pp.2584–2594, 2009.
13. U.Topal and Ü.Uzman, ‘Multiobjective optimisation of angle-ply laminated plates for maximum buckling load’, Finite Elements in Analysis and Design**,** Pp 273–279, 2010.
14. C.Roos and C.E.Bakis, ‘Multi-physics design and optimization of flexible matrix composite driveshafts’, Composite Structures, Vol. 93, Issue 9, Pp 2231–2240,2011.
15. M. H. Sadr and H. G.Bargh, ‘Optimization of laminated composite plates for maximum fundamental frequency using Elitist-Genetic algorithm and finite strip method’, Springer Science+ Business Media,2011
16. R.Kayikci and F.O.Sonmez, ‘Design of composite laminates for optimum frequency response’, Journal of Sound and Vibration, Vol. 331, Pages 1759–1776, 2012.
17. R. Khandan, S.Noroozi, P.Sewell, J.Vinney and M.Koohgilani, ‘Optimum design of fibre orientation in composite laminate plates for out-plane stresses’, Advances in Material Science and Engineering, Article ID 232847, 2012.

18. O.Montagnier and Ch. Hochard, ‘Optimization of hybrid high-modulus/high-strength carbon fibre reinforced plastic composite drive shafts’, Materials & Design**,** Vol. 46, Pages 88–100,2013
19. I.B.C.M. Rocha , E. Parente and A.M.C. Melo, ‘A hybrid shared/distributed memory parallel genetic algorithm for optimization of laminate composites’, Composite Structures**,** Vol. 107, Pages 288–297, 2014.
20. M. M.Abadi and A.R Daneshmehr, ‘An investigation of modified couple stress theory in buckling analysis of micro composite laminated Euler-Bernouli and Timoshinko beams’, International Journal of Engineering Science’, vol. 75,pp 40-53, 2014.
21. M.K.Apalak, M.Yildirim and R.Ekici, ‘Layer optimization for maximum fundamental frequency of laminated composite plates for different edge conditions’, Composite science and Technology, vol.68, pp.537-550, 2008.
22. E.J.Barbero, ‘Finite element analysis of Composite Materials using ANSYS’, CRC Press, 2008.

APPENDIX-A

MATLAB program developed

Now-a-days, MATLAB graphic user interface (called GUIDE) is one of the popular means of projecting the front-end data. Here, is the program developed in this work:

```
function varargout = LaminateAnalyzer(varargin)
%LAMINATEANALYZER M-file for LaminateAnalyzer.fig
%      LAMINATEANALYZER, by itself, creates a new LAMINATEANALYZER or raises the existing
%      singleton*.
%
%      H = LAMINATEANALYZER returns the handle to a new LAMINATEANALYZER or the handle to
%      the existing singleton*.
%
%      LAMINATEANALYZER('Property','Value',...) creates a new LAMINATEANALYZER using the
%      given property value pairs. Unrecognized properties are passed via
%      varargin to LaminateAnalyzer_OpeningFcn.  This calling syntax produces a
%      warning when there is an existing singleton*.
%
%      LAMINATEANALYZER('CALLBACK') and LAMINATEANALYZER('CALLBACK',hObject,...) call the
%      local function named CALLBACK in LAMINATEANALYZER.M with the given input
%      arguments.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help LaminateAnalyzer


% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @LaminateAnalyzer_OpeningFcn, ...
                   'gui_OutputFcn',  @LaminateAnalyzer_OutputFcn, ...
                   'gui_LayoutFcn',  [], ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
   gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before LaminateAnalyzer is made visible.
function LaminateAnalyzer_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   unrecognized PropertyName/PropertyValue pairs from the
%            command line (see VARARGIN)

% Choose default command line output for LaminateAnalyzer
handles.output = hObject;

% Update handles structure
```

```
guidata(hObject, handles);

% UIWAIT makes LaminateAnalyzer wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = LaminateAnalyzer_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


% --- Executes on button press in commandSubmit.
function commandSubmit_Callback(hObject, eventdata, handles)
% hObject    handle to commandSubmit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

set(handles.tableABD,'Data',eye(6))
% ---Define Inputs
% Ply Properties
E1 = str2num(get(handles.inputE1,'String'));
E2 = str2num(get(handles.inputE2,'String'));
G12 = str2num(get(handles.inputG12,'String'));
NU12 = str2num(get(handles.inputNU12,'String'));
% Symmetry
sym = str2num(get(handles.inputSym,'String'));
% Total Height
htotal= str2num(get(handles.inputTotHeight,'String'));
% Independent ply orientations
O = str2num(get(handles.inputPlyOri,'String'));
% Independent ply Aspect ratios
AR = str2num(get(handles.inputPlyAR,'String'));
% --- Error statements
if (isempty(E1))
errordlg('Please enter a numerical value for E1.','Missing Parameter')
end
if (isempty(E2))
errordlg('Please enter a numerical value for E2.','Missing Parameter')
end
if (isempty(G12))
errordlg('Please enter a numerical value for G12.','Missing Parameter')
end
if (isempty(NU12))
errordlg('Please enter a numerical value for NU12.','Missing Parameter')
end
if (isempty(AR))
errordlg('Please enter values separated by spaces or comas for the Independent Ply Aspect Ratios.','Missing Parameter')
end
if (isempty(O))
errordlg('Please enter values separated by spaces or comas for the Independent Ply Orientations.','Missing Parameter')
end
if (isempty(htotal))
errordlg('Please enter a numerical value for Total Height.','Missing Parameter')
end
% Counting number of rows and columns in AR and O inputs for comparison
%display(AR);
[ARrows,ARcols] = size(AR(1,:));
[orows,ocols] = size(O(1,:));
%Composite Laminate Analyzer Users Manual
% AR and O number count needs to be the same
if ocols~=ARcols
errordlg('Number of entries in the ply orientations field should equal the number of entries in aspect ratios field.','Check Independent
Ply Orientation/Aspect Ratio fields')
end
```

```
% =====END OF INPUT DATA CHECK=====
% =====START OF PREPROCESSING STATE=====
% --- Developing the complete laminate geometry
% Getting NU21
NU21 = (NU12*E2)/E1 ;
% Considering symmetry input counting the total number plies in laminate
if sym==0
plycount = ocols;
else
plycount= ocols*2*(sym);
end
% Considering symmetry input developing the the bottom half of the laminate
% orientations(O) and aspect ratios(AR)
if sym >= 2
Oini = O;
ARini = AR;
for i= 2:sym
O = [O Oini];
AR = [AR ARini];
end
end
% Applying symmetry(if needed) and developing the complete O and AR array
if sym>0
k=0;
for i = (plycount/2)+1 : plycount
AR(1,i) = AR(1, i-(1+2*k));
O(1,i) = O(1, i-(1+2*k));
k=k+1;
end
end
% Finding the Height(H) of each ply using the AR and total height (htotal)
sumAR = sum(AR);
H = htotal*(AR/sumAR);
% Developing the Z matrix
Z=zeros(1,plycount+1);
hindex = -(htotal)/2 ;
for i=1:plycount+1
if i==1
Z(:,i) = hindex;
else
Z(:,i) = Z(:,i-1)+ H(i-1);
end
% Rounding small values of Z to zero
if abs(Z(:,i)) <1.0e-4
Z(:,i)=0;
end
end
%Composite Laminate Analyzer Users Manual
% =====END OF PREPROCESSING STATE=====
% =====START OF CALCULATIONS=====
% Finding the components of Q matrix in the principal directions
% reference: Engineering mechanics of composites, second edition
% Daniel and Ishai P77, eq. 4.56
Q11 = E1/(1- NU12*NU21) ;
Q22 = E2/(1- NU12*NU21) ;
Q12 = (NU21*E1)/(1- NU12*NU21) ;
Q66 = G12;
% note that Q21 = Q12
Qp = [ Q11, Q12 , 0 ;
Q12, Q22 , 0 ;
0 , 0 , Q66 ];
% Intializing ABD as a 3x3 zero matrix
A = zeros(3,3) ;
B = zeros(3,3) ;
D = zeros(3,3) ;
% Calculating the A, B, D Matrices for each ply
Qp(3,3) = Qp(3,3)*2 ;
for l = 1 : plycount
thetar = (O(l)/180)*pi;
% define "m" and "n" as
```

```
m = cos(thetar);
n = sin(thetar);
% 2D Transformation matrix T(Daniel and Ishai P76) :
T = [m^2 , n^2 , 2*m*n ;
n^2 , m^2 , -2*m*n ;
-m*n , m*n , m^2 - n^2 ];
% Correcting for engineering strain-true strain(Daniel and Ishai P79)
Q = T\Qp*T ;
Q(:,3) = Q(:,3)*.5 ;
for i=1:3
for j=1:3
A(i,j)= A(i,j) + (Q(i,j))*( Z(l+1) - Z(l) );
B(i,j)= B(i,j) + 0.5* ((Q(i,j))*( (Z(l+1))^2 - (Z(l))^2 ));
D(i,j)= D(i,j) + (1/3)*((Q(i,j))*( (Z(l+1))^3 - (Z(l))^3 ));
end
end
end
% Filter to round close to zero numbers to zero
for i=1:3
for j=1:3
if abs(A(i,j))< 1.0e-4
A(i,j) = 0;
end
if abs(B(i,j))< 1.0e-4
B(i,j) = 0;
end
if abs(D(i,j))< 1.0e-4
D(i,j) = 0;
end
end
%Composite Laminate Analyzer Users Manual
end
%Finding AB-BD matrix and its inverse
ABDmatrix = [A B ; B D];
abcdmatrix = inv(ABDmatrix);
a = abcdmatrix(1:3, 1:3);
% Define matrix with ply orientations as row one and corresponding height
% as row two for display purposes
OHmatrix = [O;H];
% Find Overall laminate elastic properties
Ex = 1/((htotal)*a(1,1));
Ey = 1/((htotal)*a(2,2));
Gxy = 1/((htotal)*a(3,3));
NUxy = -a(2,1)/a(1,1);
NUyx = -a(1,2)/a(2,2);
ETAsx = a(1,3)/a(3,3);
ETAxs = a(3,1)/a(1,1);
ETAys = a(3,2)/a(2,2);
ETAsy = a(2,3)/a(3,3);

% =====END OF CALCULATIONS=====
% =====START OUTPUT RESULTS TO GUI=====
% Display the OHmatrix in GUI table

set(handles.tableOH,'Data',OHmatrix)
% Display the [A B ; B D] matrix in GUI table
set(handles.tableABD,'Data',ABDmatrix)
% Display the overall Laminate properties in Gui
set(handles.outputEx,'String',Ex);
set(handles.outputEy,'String',Ey);
set(handles.outputGxy,'String',Gxy);
set(handles.outputNUxy,'String',NUxy);
set(handles.outputNUyx,'String',NUyx);
set(handles.outputETAxs,'String',ETAxs);
set(handles.outputETAsx,'String',ETAsx);
set(handles.outputETAys,'String',ETAys);
set(handles.outputETAsy,'String',ETAsy);
% =====END OUTPUT RESULTS TO GUI=====
guidata(hObject, handles);
% =====END OF SCRIPT=====
function inputPlyOri_Callback(hObject, eventdata, handles)
% hObject    handle to inputPlyOri (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of inputPlyOri as text
%        str2double(get(hObject,'String')) returns contents of inputPlyOri as a double


% --- Executes during object creation, after setting all properties.
function inputPlyOri_CreateFcn(hObject, eventdata, handles)
% hObject    handle to inputPlyOri (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function inputSym_Callback(hObject, eventdata, handles)
% hObject    handle to inputSym (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of inputSym as text
%        str2double(get(hObject,'String')) returns contents of inputSym as a double


% --- Executes during object creation, after setting all properties.
function inputSym_CreateFcn(hObject, eventdata, handles)
% hObject    handle to inputSym (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

%% =====START OF INPUT ARGUMENTS=====
% --- Input Ply orientations for each ply

function inputPlyAR_Callback(hObject, eventdata, handles)
% hObject    handle to inputPlyAR (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of inputPlyAR as text
%        str2double(get(hObject,'String')) returns contents of inputPlyAR as a double


% --- Executes during object creation, after setting all properties.
function inputPlyAR_CreateFcn(hObject, eventdata, handles)
% hObject    handle to inputPlyAR (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function inputTotHeight_Callback(hObject, eventdata, handles)
% hObject    handle to inputTotHeight (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of inputTotHeight as text
%        str2double(get(hObject,'String')) returns contents of inputTotHeight as a double


% --- Executes during object creation, after setting all properties.
function inputTotHeight_CreateFcn(hObject, eventdata, handles)
% hObject    handle to inputTotHeight (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function inputE1_Callback(hObject, eventdata, handles)
% hObject    handle to inputE1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of inputE1 as text
%        str2double(get(hObject,'String')) returns contents of inputE1 as a double


% --- Executes during object creation, after setting all properties.
function inputE1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to inputE1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function inputE2_Callback(hObject, eventdata, handles)
% hObject    handle to inputE2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of inputE2 as text
%        str2double(get(hObject,'String')) returns contents of inputE2 as a double


% --- Executes during object creation, after setting all properties.
function inputE2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to inputE2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function inputG12_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to inputG12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of inputG12 as text
%        str2double(get(hObject,'String')) returns contents of inputG12 as a double


% --- Executes during object creation, after setting all properties.
function inputG12_CreateFcn(hObject, eventdata, handles)
% hObject    handle to inputG12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function inputNU12_Callback(hObject, eventdata, handles)
% hObject    handle to inputNU12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of inputNU12 as text
%        str2double(get(hObject,'String')) returns contents of inputNU12 as a double


% --- Executes during object creation, after setting all properties.
function inputNU12_CreateFcn(hObject, eventdata, handles)
% hObject    handle to inputNU12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes on selection change in popupmenuStress.
function popupmenuStress_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenuStress (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
handles.outputUnits1='';
guidata(hObject, handles);
switch get(handles.popupmenuStress,'Value')
case 1
set(handles.outputUnits1,'String','GPa');

case 2
set(handles.outputUnits1,'String','psi');

end
% Hints: contents = cellstr(get(hObject,'String')) returns popupmenuStress contents as cell array
%        contents{get(hObject,'Value')} returns selected item from popupmenuStress


% --- Executes during object creation, after setting all properties.
function popupmenuStress_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenuStress (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes on selection change in popupmenuLength.
function popupmenuLength_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenuLength (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
handles.outputUnits2='';
guidata(hObject, handles);
switch get(handles.popupmenuLength,'Value')
case 1
set(handles.outputUnits2,'String','mm');

case 2
set(handles.outputUnits2,'String','inches');

end
% Hints: contents = cellstr(get(hObject,'String')) returns popupmenuLength contents as cell array
%        contents{get(hObject,'Value')} returns selected item from popupmenuLength


% --- Executes during object creation, after setting all properties.
function popupmenuLength_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenuLength (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function outputEx_Callback(hObject, eventdata, handles)
% hObject    handle to outputEx (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of outputEx as text
%        str2double(get(hObject,'String')) returns contents of outputEx as a double


% --- Executes during object creation, after setting all properties.
function outputEx_CreateFcn(hObject, eventdata, handles)
% hObject    handle to outputEx (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function outputEy_Callback(hObject, eventdata, handles)
% hObject    handle to outputEy (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of outputEy as text
%        str2double(get(hObject,'String')) returns contents of outputEy as a double
% --- Executes during object creation, after setting all properties.
function outputEy_CreateFcn(hObject, eventdata, handles)
% hObject    handle to outputEy (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function outputGxy_Callback(hObject, eventdata, handles)
% hObject    handle to outputGxy (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of outputGxy as text
%        str2double(get(hObject,'String')) returns contents of outputGxy as a double


% --- Executes during object creation, after setting all properties.
function outputGxy_CreateFcn(hObject, eventdata, handles)
% hObject    handle to outputGxy (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function outputNUxy_Callback(hObject, eventdata, handles)
% hObject    handle to outputNUxy (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of outputNUxy as text
%        str2double(get(hObject,'String')) returns contents of outputNUxy as a double


% --- Executes during object creation, after setting all properties.
function outputNUxy_CreateFcn(hObject, eventdata, handles)
% hObject    handle to outputNUxy (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function outputNUyx_Callback(hObject, eventdata, handles)
% hObject    handle to outputNUyx (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of outputNUyx as text
%        str2double(get(hObject,'String')) returns contents of outputNUyx as a double


% --- Executes during object creation, after setting all properties.
```

```
function outputNUyx_CreateFcn(hObject, eventdata, handles)
% hObject    handle to outputNUyx (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function outputETAxs_Callback(hObject, eventdata, handles)
% hObject    handle to outputETAxs (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of outputETAxs as text
%        str2double(get(hObject,'String')) returns contents of outputETAxs as a double


% --- Executes during object creation, after setting all properties.
function outputETAxs_CreateFcn(hObject, eventdata, handles)
% hObject    handle to outputETAxs (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function outputETAsx_Callback(hObject, eventdata, handles)
% hObject    handle to outputETAsx (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of outputETAsx as text
%        str2double(get(hObject,'String')) returns contents of outputETAsx as a double


% --- Executes during object creation, after setting all properties.
function outputETAsx_CreateFcn(hObject, eventdata, handles)
% hObject    handle to outputETAsx (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function outputETAys_Callback(hObject, eventdata, handles)
% hObject    handle to outputETAys (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of outputETAys as text
%        str2double(get(hObject,'String')) returns contents of outputETAys as a double


% --- Executes during object creation, after setting all properties.
function outputETAys_CreateFcn(hObject, eventdata, handles)
% hObject    handle to outputETAys (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function outputETAsy_Callback(hObject, eventdata, handles)
% hObject    handle to outputETAsy (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of outputETAsy as text
%        str2double(get(hObject,'String')) returns contents of outputETAsy as a double


% --- Executes during object creation, after setting all properties.
function outputETAsy_CreateFcn(hObject, eventdata, handles)
% hObject    handle to outputETAsy (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes during object deletion, before destroying properties.
function outputEy_DeleteFcn(hObject, eventdata, handles)
% hObject    handle to outputEy (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

APPENDIX-B

USE OF SHELL ELEMENTS IN ANSYS

The FEM modeling is done in ANSYS and the steps for the same are given below:-

First start ANSYS: choose “Mechanical APDL Product Launcher” in the ANSYS menu and define job name and work directory. Make sure the “ANSYS” simulation environment and “ANSYS Academic Teaching Advanced” license are chosen.

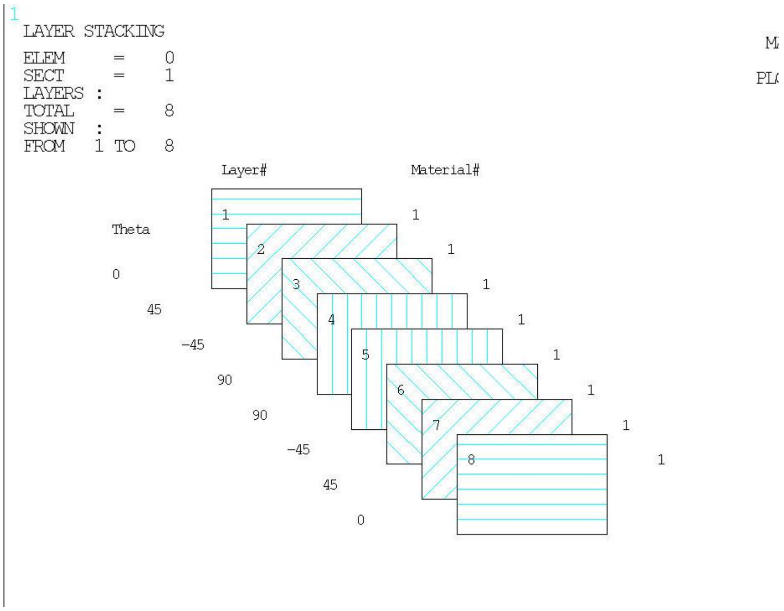
By “Preprocessor > Element Type > Add” we select the element to use as the 8-node shell element SHELL281. Alternatively, we could use SHELL181, but for this plane example with constant stress state, the element choice is of less importance.

The following options for the element are changed: Key option 3 (Integration option) is set to “Full w/ incompatible modes” instead of the default "Reduced integration”, and Key-option 8 (Storage of layer data) is changed to “All layers”.

Next material properties are defined: “Main Menu > Preprocessor > Material Props > Material Models > Structural > Linear> Elastic > Orthotropic” and you enter the material data.

As default you enter the major Poisson's ratio (PRXY, etc.). It is a good idea also to define the density (1470 kg/m3) for the laminate.

Using the menu “Main Menu > Preprocessor > Sections > Shell > Lay-up > Plot Section” we get the following plot (Fig.B1):



```
1
LAYER STACKING
ELEM    =    0
SECT    =    1
LAYERS :
TOTAL   =    8
SHOWN  :
FROM   1 TO  8
```

Fig.B1 Stacking layout

Using “Main Menu > Preprocessor > Meshing > MeshTool” we can select a given area and change the definition of area attributes (including the element coordinate system) and with this section laminated area has been meshed for FE analysis as shown in Fig.B2.

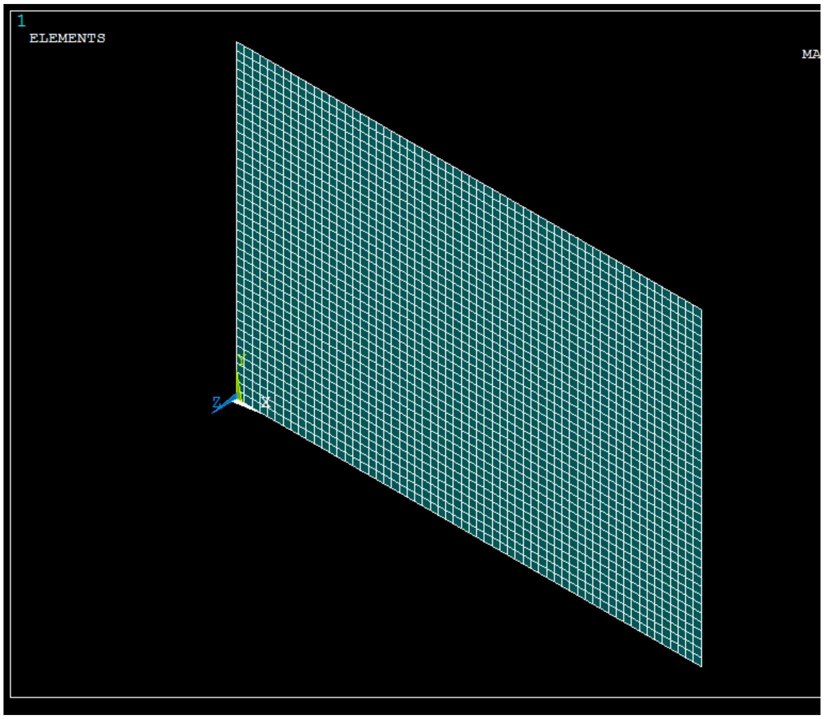


Fig.B2 Meshed-plate