ABSTRACT

INTERNAL DELAMINATION DETECTION PROBLEMS USING A COMBINED IMPROVED-COUNTERPROPAGATION NEURAL NETWORK AND GENETIC ALGORITHM TECHNIQUE

By

Tran N. Phuong

August 2013

Detection of delamination in laminated composites is first formulated as a simulation and second as an optimization problem and solved by the new approach utilizing improved-counterpropagation neural networks and genetic algorithms, respectively. A recently developed improved-layerwise composite laminate theory is extended to model composite laminates with delamination. This new layerwise finite element model is employed to calculate natural frequencies of cross-ply laminates with given delamination patterns placed at different locations. Improved-counterpropagation neural networks are trained to simulate dynamic responses from the finite element analysis. These artificial neural networks are chosen as function approximations which are developed on the available input-output data from a finite element model. Genetic algorithms with mixed type design variables are used to search the optimum delamination patterns associated with the given natural frequencies. Results with remarkable accuracy have been obtained in detecting the internal delamination using a combination of both techniques.

INTERNAL DELAMINATION DETECTION PROBLEMS USING A COMBINED IMPROVED-COUNTERPROPAGATION NEURAL NETWORK AND GENETIC ALGORITHM TECHNIQUE

A THESIS

Presented to the Department of Mechanical and Aerospace Engineering

California State University, Long Beach

In Partial Fulfillment

Of the Requirements for the Degree

Master of Science in Mechanical Engineering

Committee Members:

Hsin-Piao Chen, Ph.D. (Chair)
Min-Ten Jahn, Ph.D.
Jalal Torabzadeh, Ph.D.

College Designee:

Burkhard Englert, Ph.D.

By Tran N. Phuong

B.S., 2000, California State University, Long Beach

August 2013

ACKNOWLEDGEMENTS

I wish to thank my thesis advisor, Dr. Hsin-Piao Chen, for his guidance and teachings throughout the duration of my master courses and preparation of this thesis. I wish to thank my former graduate advisor, Dr. Hamid Rahai, and my undergraduate advisor, Dr. Jalal Torabzadeh, for all their support. I also wish to thank my sister for checking the grammar of this thesis.

TABLE OF CONTENTS

Page

CHAPTER

CHAPTER Page

CHAPTER Page

APPENDICES Page

LIST OF TABLES

TABLE Page

TABLE Page

LIST OF FIGURES

FIGURE Page

FIGURE Page

LIST OF ABBREVIATIONS

BPNN	Back-Propagation Neural Network
CPNN	Counterpropagation Neural Network
FEA	Finite Element Analysis
GA	Genetic Algorithm
K-NN	Kohonen Neural Network
PIN	Delamination patterns
TMSE	Mean Square Error of the Training data sets
VRME	Relative Mean Error of Validation data sets
VRRMSE	Relative Root Mean Square Error of Validation data sets

LIST OF SYMBOLS

a	Delamination size
a_x, a_y	Length and width of delamination
b_j	Bias
E	Young’s modulus
f	Objective function
G_{12}	In-plane shear modulus
L_x	Plate length
n_p	Number of input elements
n_m	Number of output frequencies
n_v	Number of validation data sets
P	Initial population of individuals
p_c	Probability of crossover
p_m	Probability of mutation
t	Generation counter
u_i, w	Displacement of the reference plane
U_i^k	In-plane displacement of the k-th layer of the laminate
U_3^k	Transverse deflection of the k-th layer of the laminate

W_j	Weight associated with j-th frequency
X_i	Each individual of population
x_d	Lengthwise coordinate of the leading edge of the delamination
x_d, y_d	In-plane x and y coordinates of the left-lower corner of the delamination
z_d	Thickness location index of the delamination (0, 1, 2, 3)
z_j	Delamination interface coordinate in z-axis
μ	Number of individuals in population
ν_{12}	Major Poisson's ratio
ρ	Density
$\wp$	Delamination in x or y direction
$\wp_{\%}$	Percentage of elamination in x or y direction
$\wp_{\text{start}}$	Delamination start in x or y direction
$\wp_{\text{end}}$	Delamination end in x or y direction
σ	Standard deviation
ω_m	Measured natural frequency (natural frequency simulated by CPNN)
ω_p	Predicted natural frequency (natural frequency searched by GA)
ϕ_i	Rotations of the normal to the reference plane about the x and y axes
θ_i^k, ψ_i^k	Layerwise structural unknowns defined at each lamina
$\bar{u}_i^j$, $\bar{w}^j$	Represent possible jumps in displacement field due to delamination allowing slipping and separation between sublaminate

CHAPTER 1

INTRODUCTION

Composite laminates are increasingly used in the construction of aerospace, mechanical, civil, marine, automotive, and other high performance structures due to their high specific stiffness and strength, excellent fatigue resistance, and longer durability as compared to metallic structures. Composite laminates have also played an important role in weight saving. Additionally, their design flexibility allows engineers to carve out a niche as the material of choice in a variety of specialized applications. Delamination, which is an interlayer debonding or separation between individual plies of the laminate, is a prevalent form of damage phenomenon in laminated composites [1]. Delamination can often be pre-existing or generated during service life. For example, delamination often occurs at stress free edges due to the mismatch of properties at ply interfaces and it can also be generated by external forces such as out of plane loading or impact during the service life. Delaminations may not be visible or barely visible on the surface, since they are embedded within the composite structures. The existence of delaminations not only alters the load carrying capacity of the structure, but can also affect its dynamic response. Delaminations reduce the natural frequency, as a direct result of the reduction in stiffness, which may cause resonance if the reduced frequency is close to the working frequency. In order to assure the successful implementation and improved reliability of such structures, the detection and quantification of delamination is an important technology that must be addressed.

All load-carrying members of structures continuously accumulate damage in their service environment. All types of damages in composite structures result in changes in stiffness, strength, and fatigue properties. Measurement of strength or fatigue properties during damage development is not feasible because destructive testing is required. However, stiffness reduction due to damage can be measured since damage directly affects structural responses. The most widely used method for damage assessment is to identify the occurrence, location, and extent of the damage from measured structural dynamic characteristics (e.g., natural frequencies and modal shapes). Since a change in stiffness will cause changes in the natural frequencies of a vibrating system (whether the damage is localized or distributed), the measurement of natural frequencies of a structure at two or more stages of its life can help locate the damage in a structure [2]. Therefore, the relationship between the physical parameters, such as mass and stiffness, and the dynamic characteristics, such as frequencies and mode shapes, must be determined by modal analysis.

Detection of the location and size of delamination in composite structures involves both an extensive computational effort and the mathematically challenging task of solving an inverse problem. There is no unique or exact solution to the inverse eigenvalue problem for any "real" mechanical system due to nonlinearities and strong couplings between the eigenstructure and the parameters of the system. Moreover, the existence of a feasible solution cannot be guaranteed for an arbitrary real system. Since it is essentially an inverse multi-modal problem, usually a large number of local optima exist and the traditional gradient-based technique may result in a trapped local optimum. Thus, in identifying the delamination pattern of composite structures, global damage detection techniques based on the natural frequencies and mode shapes have been employed. This detection of delamination can be formulated as an optimization problem with an appropriately chosen objective function, which represents the quantified difference in vibration responses (e.g., natural frequencies and modal shapes) between measurements and predictions using analytical methods. To further complicate this problem, both continuous and discrete design variables need to be adopted because in general, the size and in-plane location of delaminations are defined by continuous real variables, and the thickness location of delamination is defined by discrete variables, since delamination only appear at the interface of distinct layers.

In this research study, the delamination detection problem is formulated in MATLAB codes as a multi-modal optimization problem with mixed type design variables. The current method comprises of three distinct segments: finite element, counterpropagation neural network, and genetic algorithm. First, the natural frequencies are chosen as the structural response which will be measured. A finite element model of the damaged structure based on an improved-layerwise composite laminate theory developed by Kim, Chattopadhyay, and Ghoshal (2003) is employed to calculate the natural frequencies of laminates with given delamination patterns. Second, to resolve the computational problem, an improved-counterpropagation neural network has been developed to simulate dynamic responses based on the input-output data of the finite element model. This artificial neural network will be used as a function approximation with satisfactory accuracy to determine natural frequencies of the delaminated laminate with various delamination patterns. Third, a search and optimization technique based on the genetic algorithm (GA) is adopted to find the global optimum in a multi-modal search space with continuous, discrete, or mixed design variables. For each potential solution in the current population (a pool of potential solutions), natural frequencies of the corresponding delaminated laminate must be calculated in order to determine the objective (or fitness) function in genetic algorithm. The computational work will be very intensive if the finite element model is used to calculate these frequencies directly. For this reason, the counterpropagation neural network was chosen to be the interactive module used to simulate the dynamic responses of the delaminated laminates from the finite element model, and then to feed this saved data into genetic algorithm. Single internal delamination is the type of delamination configuration that will be considered in this research study. Excellent results have been obtained in detecting the internal delamination for delaminated plates using the combined technique of finite element, counterpropagation neural network, and genetic algorithm.

CHAPTER 2

MECHANICS OF COMPOSITE LAMINATES

Introduction to Composite Lamina and Composite Laminates

A structural composite is a material system consisting of two or more phases on a macroscopic scale, whose mechanical performance and properties are designed to be superior to those of the constituent materials acting independently. The main components of composite materials are fibers and matrix. The fibers provide most of the stiffness and strength, and the matrix binds the fibers together, thus providing load transfer between fibers and the composite and between the external loads and supports. The basis for the superior structural performance of composite materials lies in the high specific strength (strength to density ratio) and high specific stiffness (modulus to density ratio), and in the anisotropic and heterogeneous character of the material. The latter provides the composite with many degrees of freedom, enabling simultaneous material optimization with several given constraints, such as minimum weight, maximum dynamic stability, cost effectiveness, and so on.

Because of the inherently heterogeneous nature of composite materials, they are conveniently studied from two points of view: micromechanics and macromechanics. Micromechanics is the study of composite materials taking into account the interaction of the constituent materials in detail. Micromechanics allows the designer to represent a heterogeneous material as an equivalent homogeneous material, usually anisotropic. Micromechanics can be used to predict stiffness (with great success) and strength (with less success). One objective of micromechanics is to obtain functional relationships for the elastic constants of the composite in the form:

$$C_{ij} = C_{ij}\left(E_{f1},\, E_{f2},\, E_m,\, G_{f12},\, G_m,\, \nu_{f12},\, \nu_m,\, V_f,\, V_m,\, V_v,\, S,\, A\right) \qquad (1)$$

where $E, G, \nu, V, S,$ and A are modulus of elasticity, shear modulus, Poisson's ratio, volume, shape of reinforcement, and array (packing) of the reinforcement, respectively. Macromechanics is the study of a laminate's response to loading based on the properties of each lamina and the stacking sequence of the laminae. In macromechanical analysis, where the material is treated as quasi-homogeneous, the average material behavior can be controlled and predicted from the properties of the constituents.

The basic building block of a laminate is a lamina, which is a plane layer of unidirectional fibers or woven fibers in a supporting matrix. A laminate is a bonded stack of laminae (plies, or layers) with various orientations of principal material directions in the laminae. Thus, knowledge of the mechanical behavior of a lamina is essential to the understanding of laminated fiber-reinforced structures. Both the laminae and the laminate are assumed to behave as a linear elastic material, which can be described in terms of the stresses, corresponding strains, and deformation hypotheses.

Classical Lamination Theory consists of a collection of mechanics-of-materials stress and deformation hypotheses. By using this theory, one can consistently proceed directly from the basic building block, the lamina, to the end result, a structural laminate. The whole process is one of finding effective and reasonably accurate simplifying assumptions that enables the reduction of a complicated three-dimensional elasticity problem to a solvable two-dimensional mechanics of deformable bodies problem. Thus, the transition from the lamina to the laminate level entails several simplifying assumptions that are commonly practiced in the real world.

Throughout this section of the thesis, the focus on macromechanical behavior of composite lamina and composite laminates are discussed [3-6]. The intention is to provide a basic understanding of the following aspects of structural behavior: the stress and strain behavior of an individual lamina, the variations of stress and strain through the thickness of the laminate, the relation of the laminate forces and moments to the strains and curvatures, and the defining characteristics for the various types of composite laminates. This will serve as a precursor to the background forming the basis of this research.

Macromechanical Behavior of Composite Lamina

General Anisotropic Material

When there are no symmetry planes with respect to the alignment of the fibers, the material is referred to as generally anisotropic. The state of stress at a point in a material can be represented by nine stress components σ_{ij} (where i, j =1, 2, 3). Similarly, the state of deformation is represented by nine strain components, ε_{ij}. In the most general case, the stress and strain components are related by a generalization of Hooke's law as follows:

$$\{\sigma\}_{9x1} = [C]_{9x9} \{\varepsilon\}_{9x1}, \qquad \{\varepsilon\}_{9x1} = [S]_{9x9} \{\sigma\}_{9x1} \tag{2}$$

where $[C]$ is the stiffness matrix and $[S]$ is the compliance matrix.

In general, it would require 81 elastic constants to characterize a material fully. However, the symmetry of the stress and strain tensors ($\sigma_{ij} = \sigma_{ji}$, $\varepsilon_{ij} = \varepsilon_{ji}$) reduces the number of independent elastic constants to 36. Thus the state of stress (or strain) at a point can be described by six components of stress (or strain).

$$\{\sigma\}_{6x1} = [C]_{6x6}\{\varepsilon\}_{6x1}, \qquad \{\varepsilon\}_{6x1} = [S]_{6x6}\{\sigma\}_{6x1} \tag{3}$$

Elastic strain energy considerations require additional symmetries of the stiffness and compliance matrices ($C_{ij} = C_{ji}$, $S_{ij} = S_{ji}$). Because of this symmetry, only 21 of the 36 elastic constants are independent in both $[C]$ and the $[S]$ matrices.

It is customary in mechanics of composites to express the order of stress and strain in tensor, contracted, or engineering notation as follows:

$$\underbrace{\left\{\begin{array}{c}\sigma_{11}\\ \sigma_{22}\\ \sigma_{33}\\ \hdashline \sigma_{23}\\ \sigma_{13}\\ \sigma_{12}\end{array}\right\}}_{\text{Tensor}} = \underbrace{\left\{\begin{array}{c}\sigma_{1}\\ \sigma_{2}\\ \sigma_{3}\\ \hdashline \sigma_{4}\\ \sigma_{5}\\ \sigma_{6}\end{array}\right\}}_{\text{Contracted}} = \underbrace{\left\{\begin{array}{c}\sigma_{1}\\ \sigma_{2}\\ \sigma_{3}\\ \hdashline \tau_{23}\\ \tau_{13}\\ \tau_{12}\end{array}\right\}}_{\text{Engineering}} \quad \begin{array}{c}\text{normal}\\ \text{- - - - - - - - -}\\ \text{shear}\end{array} \quad \underbrace{\left\{\begin{array}{c}\varepsilon_{11}\\ \varepsilon_{22}\\ \varepsilon_{33}\\ \hdashline 2\varepsilon_{23}\\ 2\varepsilon_{13}\\ 2\varepsilon_{12}\end{array}\right\}}_{\text{Tensor}} = \underbrace{\left\{\begin{array}{c}\varepsilon_{1}\\ \varepsilon_{2}\\ \varepsilon_{3}\\ \hdashline \varepsilon_{4}\\ \varepsilon_{5}\\ \varepsilon_{6}\end{array}\right\}}_{\text{Contracted}} = \underbrace{\left\{\begin{array}{c}\varepsilon_{1}\\ \varepsilon_{2}\\ \varepsilon_{3}\\ \hdashline \gamma_{23}\\ \gamma_{13}\\ \gamma_{12}\end{array}\right\}}_{\text{Engineering}} \tag{4}$$

With the foregoing reduction from 36 to 21 independent elastic constants, the stress and strain relations for an anisotropic body can be written in engineering notation as

$$\left\{\begin{array}{c}\sigma_{1}\\ \sigma_{2}\\ \sigma_{3}\\ \hdashline \tau_{23}\\ \tau_{13}\\ \tau_{12}\end{array}\right\} = \left[\begin{array}{ccc:ccc} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16}\\ C_{21} & C_{22} & C_{23} & C_{24} & C_{25} & C_{26}\\ C_{31} & C_{32} & C_{33} & C_{34} & C_{35} & C_{36}\\ \hdashline C_{41} & C_{42} & C_{43} & C_{44} & C_{45} & C_{46}\\ C_{51} & C_{52} & C_{53} & C_{54} & C_{55} & C_{56}\\ C_{61} & C_{62} & C_{63} & C_{64} & C_{65} & C_{66}\end{array}\right] \left\{\begin{array}{c}\varepsilon_{1}\\ \varepsilon_{2}\\ \varepsilon_{3}\\ \hdashline \gamma_{23}\\ \gamma_{13}\\ \gamma_{12}\end{array}\right\} \tag{5}$$

$$\left\{\begin{array}{c}\varepsilon_{1}\\ \varepsilon_{2}\\ \varepsilon_{3}\\ \hdashline \gamma_{23}\\ \gamma_{13}\\ \gamma_{12}\end{array}\right\} = \left[\begin{array}{ccc:ccc} S_{11} & S_{12} & S_{13} & S_{14} & S_{15} & S_{16}\\ S_{21} & S_{22} & S_{23} & S_{24} & S_{25} & S_{26}\\ S_{31} & S_{32} & S_{33} & S_{34} & S_{35} & S_{36}\\ \hdashline S_{41} & S_{42} & S_{43} & S_{44} & S_{45} & S_{46}\\ S_{51} & S_{52} & S_{53} & S_{54} & S_{55} & S_{56}\\ S_{61} & S_{62} & S_{63} & S_{64} & S_{65} & S_{66}\end{array}\right] \left\{\begin{array}{c}\sigma_{1}\\ \sigma_{2}\\ \sigma_{3}\\ \hdashline \tau_{23}\\ \tau_{13}\\ \tau_{12}\end{array}\right\} \tag{6}$$

A general anisotropic material under uniaxial tension undergoes axial, transverse, and shear deformations. Under pure shear loading, the material undergoes both shear and normal deformations. Thus, one applied stress will be induced to all 6 strains and 1 strain is a summation of all 6 stresses, as shown below:

$$\begin{array}{l|l} \sigma_1 = \sigma & \varepsilon_1 = s_{11}\sigma, \quad \varepsilon_2 = s_{12}\sigma, \quad \varepsilon_3 = s_{13}\sigma \\ & \gamma_{23} = s_{14}\sigma, \quad \gamma_{13} = s_{15}\sigma, \quad \gamma_{12} = s_{16}\sigma \\ \varepsilon_i = \sum_{j=1}^{6} S_{ij}\sigma_j & \varepsilon_4 = \gamma_{23} = S_{41}\sigma_1 + S_{42}\sigma_2 + S_{43}\sigma_3 + S_{44}\tau_{23} + S_{45}\tau_{13} + S_{46}\tau_{12} \end{array} \tag{7}$$

<u>Monoclinic Material</u>

When there is a symmetry plane with respect to the alignment of the fibers, the material is referred to as monoclinic. In the symmetry plane (plane 1-2), there is no coupling effect between normal stress (σ_1, σ_2) and out-of-plane shear strain (γ_{23}, γ_{13}). Consequently, this reduces the number of independent elastic constants to 13. The nonzero and zero elements of the stiffness and compliance matrices can best be seen when the stress and strain relations are written in the forms:

$$\left\{\begin{array}{c} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \hline \tau_{23} \\ \tau_{13} \\ \tau_{12} \end{array}\right\} = \left[\begin{array}{ccc|ccc} C_{11} & C_{12} & C_{13} & 0 & 0 & C_{16} \\ C_{12} & C_{22} & C_{23} & 0 & 0 & C_{26} \\ C_{13} & C_{23} & C_{33} & 0 & 0 & C_{36} \\ \hline 0 & 0 & 0 & C_{44} & S_{45} & 0 \\ 0 & 0 & 0 & C_{45} & S_{55} & 0 \\ C_{16} & C_{26} & C_{36} & 0 & 0 & C_{66} \end{array}\right] \left\{\begin{array}{c} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \hline \gamma_{23} \\ \gamma_{13} \\ \gamma_{12} \end{array}\right\} \tag{8}$$

$$\left\{\begin{array}{c} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \hline \gamma_{23} \\ \gamma_{13} \\ \gamma_{12} \end{array}\right\} = \left[\begin{array}{ccc|ccc} S_{11} & S_{12} & S_{13} & 0 & 0 & S_{16} \\ S_{12} & S_{22} & S_{23} & 0 & 0 & S_{26} \\ S_{13} & S_{23} & S_{33} & 0 & 0 & S_{36} \\ \hline 0 & 0 & 0 & S_{44} & S_{45} & 0 \\ 0 & 0 & 0 & S_{45} & S_{55} & 0 \\ S_{16} & S_{26} & S_{36} & 0 & 0 & S_{66} \end{array}\right] \left\{\begin{array}{c} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \hline \tau_{23} \\ \tau_{13} \\ \tau_{12} \end{array}\right\} \tag{9}$$

The elements of the compliance matrix can be expressed in terms of the following engineering constants:

$$\left\{\begin{array}{c}\varepsilon_1\\ \varepsilon_2\\ \varepsilon_3\\ \hline \gamma_{23}\\ \gamma_{13}\\ \gamma_{12}\end{array}\right\}=\left[\begin{array}{ccc|ccc}\frac{1}{E_1} & \frac{-\nu_{21}}{E_2} & \frac{-\nu_{31}}{E_3} & 0 & 0 & \frac{\eta_{61}}{G_{12}}\\ \frac{-\nu_{12}}{E_1} & \frac{1}{E_2} & \frac{-\nu_{32}}{E_3} & 0 & 0 & \frac{\eta_{62}}{G_{12}}\\ \frac{-\nu_{13}}{E_1} & \frac{-\nu_{23}}{E_2} & \frac{1}{E_3} & 0 & 0 & \frac{\eta_{63}}{G_{12}}\\ \hline 0 & 0 & 0 & \frac{1}{G_{23}} & \frac{\eta_{54}}{G_{13}} & 0\\ 0 & 0 & 0 & \frac{\eta_{45}}{G_{23}} & \frac{1}{G_{13}} & 0\\ \frac{\eta_{16}}{E_1} & \frac{\eta_{26}}{E_2} & \frac{\eta_{36}}{E_3} & 0 & 0 & \frac{1}{G_{12}}\end{array}\right]\left\{\begin{array}{c}\sigma_1\\ \sigma_2\\ \sigma_3\\ \hline \tau_{23}\\ \tau_{13}\\ \tau_{12}\end{array}\right\} \tag{10}$$

From the symmetry of the compliance matrix we conclude that

$$\begin{gathered}\frac{\nu_{12}}{E_1}=\frac{\nu_{21}}{E_2},\quad \frac{\nu_{13}}{E_1}=\frac{\nu_{31}}{E_3},\quad \frac{\nu_{23}}{E_2}=\frac{\nu_{32}}{E_3}\\ \frac{\eta_{16}}{E_1}=\frac{\eta_{61}}{G_{12}},\quad \frac{\eta_{26}}{E_2}=\frac{\eta_{62}}{G_{12}},\quad \frac{\eta_{36}}{E_3}=\frac{\eta_{63}}{G_{12}},\quad \frac{\eta_{45}}{G_{23}}=\frac{\eta_{54}}{G_{13}}\end{gathered} \tag{11}$$

where

E_1 = modulus of elasticity in the fiber direction,

E_2, E_3 = modulus of elasticity in the direction transverse to the fibers,

G_{12} = in-plane shear modulus,

G_{23}, G_{13} = out-of-plane shear modulus,

ν_{12} = in-plane Poisson's ratio,

ν_{23}, ν_{13} = out-of-plane Poisson's ratio,

η_{12} = in-plane shear coupling effect,

η_{23}, η_{13} = out-of-plane shear coupling effect.

Orthotropic in Principal Material Directions

Unidirectional fiber composites or fiber-reinforced composites can be regarded as orthotropic materials which possess three mutually orthogonal planes of symmetry. The directions perpendicular to these planes are called the material principal directions. It is customary to denote the fiber direction as 1-axis, and the transverse directions as 2 and 3. For an orthotropic material in principal coordinate, the stiffness and compliance matrices further reduce the number of independent elastic constants to nine in the stress and strain relations as:

$$\left\{\begin{array}{c}\sigma_1\\ \sigma_2\\ \sigma_3\\ \hline \tau_{23}\\ \tau_{13}\\ \tau_{12}\end{array}\right\}=\left[\begin{array}{ccc|ccc}C_{11} & C_{12} & C_{13} & 0 & 0 & 0\\ C_{12} & C_{22} & C_{23} & 0 & 0 & 0\\ C_{13} & C_{23} & C_{33} & 0 & 0 & 0\\ \hline 0 & 0 & 0 & C_{44} & 0 & 0\\ 0 & 0 & 0 & 0 & C_{55} & 0\\ 0 & 0 & 0 & 0 & 0 & C_{66}\end{array}\right]\left\{\begin{array}{c}\varepsilon_1\\ \varepsilon_2\\ \varepsilon_3\\ \hline \gamma_{23}\\ \gamma_{13}\\ \gamma_{12}\end{array}\right\} \tag{12}$$

$$\left\{\begin{array}{c}\varepsilon_1\\ \varepsilon_2\\ \varepsilon_3\\ \hline \gamma_{23}\\ \gamma_{13}\\ \gamma_{12}\end{array}\right\}=\left[\begin{array}{ccc|ccc}S_{11} & S_{12} & S_{13} & 0 & 0 & 0\\ S_{12} & S_{22} & S_{23} & 0 & 0 & 0\\ S_{13} & S_{23} & S_{33} & 0 & 0 & 0\\ \hline 0 & 0 & 0 & S_{44} & 0 & 0\\ 0 & 0 & 0 & 0 & S_{55} & 0\\ 0 & 0 & 0 & 0 & 0 & S_{66}\end{array}\right]\left\{\begin{array}{c}\sigma_1\\ \sigma_2\\ \sigma_3\\ \hline \tau_{23}\\ \tau_{13}\\ \tau_{12}\end{array}\right\} \tag{13}$$

$$\left\{\begin{array}{c}\varepsilon_1\\ \varepsilon_2\\ \varepsilon_3\\ \hline \gamma_{23}\\ \gamma_{13}\\ \gamma_{12}\end{array}\right\}=\left[\begin{array}{ccc|ccc}\frac{1}{E_1} & \frac{-\nu_{21}}{E_2} & \frac{-\nu_{31}}{E_3} & 0 & 0 & 0\\ \frac{-\nu_{12}}{E_1} & \frac{1}{E_2} & \frac{-\nu_{32}}{E_3} & 0 & 0 & 0\\ \frac{-\nu_{13}}{E_1} & \frac{-\nu_{23}}{E_2} & \frac{1}{E_3} & 0 & 0 & 0\\ \hline 0 & 0 & 0 & \frac{1}{G_{23}} & 0 & 0\\ 0 & 0 & 0 & 0 & \frac{1}{G_{13}} & 0\\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G_{12}}\end{array}\right]\left\{\begin{array}{c}\sigma_1\\ \sigma_2\\ \sigma_3\\ \hline \tau_{23}\\ \tau_{13}\\ \tau_{12}\end{array}\right\} \tag{14}$$

The strain-stress relation can then be expressed in terms of engineering constants as shown in equation (14).

As seen above, the relations between compliances S_{ij} and engineering constants are fairly simple. This, however, is not the case for the relations between stiffness C_{ij} and engineering constants. The elastic moduli are referenced to this particular coordinate system and denoted by $E_1, E_2, E_3, G_{23}, G_{13}, G_{12}, \nu_{23}, \nu_{13}, \nu_{12}$. Note that there is no interaction between normal stresses $\sigma_1, \sigma_2, \sigma_3$ and shear strains $\gamma_{23}, \gamma_{13}, \gamma_{12}$, which occurs in anisotropic materials. Similarly, there is no interaction between pure shear stresses and normal strains as well as none between shear stresses and shear strains in different planes when these stresses are in the 1, 2, 3 principal material directions.

Orthotropic Material with Transversely Isotropy

An orthotropic material is called transversely isotropic when one of the three mutually orthogonal planes of material symmetry is a plane of isotropy (mechanical properties are the same in all direction), as shown in Figure 1. Many unidirectional composites with fibers packed in a hexagonal array (or close to it) can be considered transversely isotropic (i.e., aramid/epoxy, carbon/epoxy, and glass/epoxy composites). In the isotropic plane (plane 2-3), which is normal to the fibers in direction 1, the subscripts 2 and 3 are interchangeable in the material constants. This further reduces the number of independent elastic constants to 5 ($E_1, E_2, G_{12}, \nu_{23}, \nu_{12}$). The elements of the stiffness and compliance matrices can best be seen when the stress and strain relations are written in the forms

$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \hline \tau_{23} \\ \tau_{13} \\ \tau_{12} \end{Bmatrix} = \left[\begin{array}{ccc|ccc} C_{11} & C_{12} & C_{12} & 0 & 0 & 0 \\ C_{12} & C_{22} & C_{23} & 0 & 0 & 0 \\ C_{12} & C_{23} & C_{22} & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & (C_{22}-C_{23})/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{66} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{66} \end{array}\right] \begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \hline \gamma_{23} \\ \gamma_{13} \\ \gamma_{12} \end{Bmatrix} \tag{15}$$

$$\begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \hline \gamma_{23} \\ \gamma_{13} \\ \gamma_{12} \end{Bmatrix} = \left[\begin{array}{ccc|ccc} S_{11} & S_{12} & S_{12} & 0 & 0 & 0 \\ S_{12} & S_{22} & S_{23} & 0 & 0 & 0 \\ S_{12} & S_{23} & S_{22} & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 2(S_{22}-S_{23}) & 0 & 0 \\ 0 & 0 & 0 & 0 & S_{66} & 0 \\ 0 & 0 & 0 & 0 & 0 & S_{66} \end{array}\right] \begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \hline \tau_{23} \\ \tau_{13} \\ \tau_{12} \end{Bmatrix} \tag{16}$$

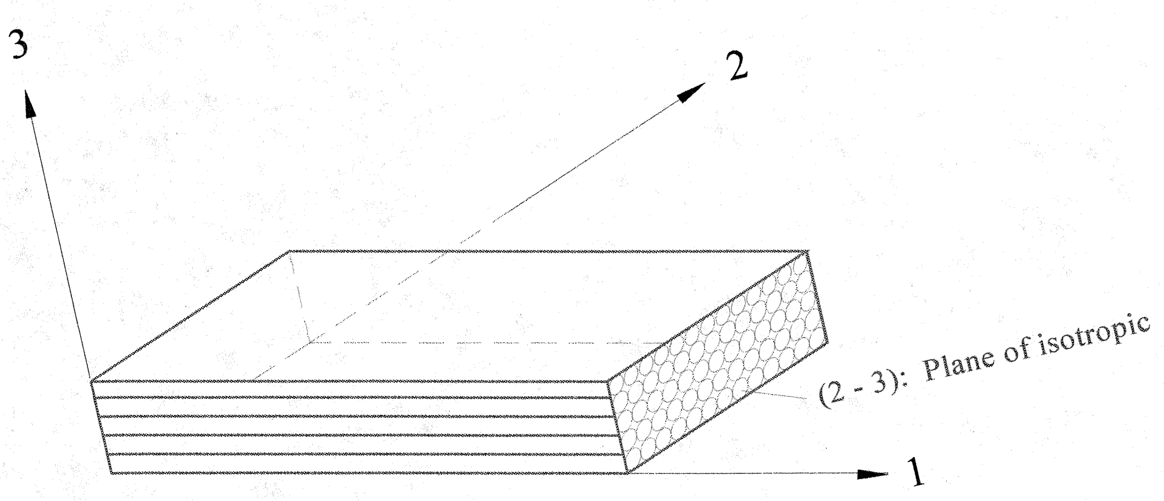

FIGURE 1. Orthotropic material with transverse isotropy.

The strain-stress relation can then be expressed in terms of engineering constants as follows:

$$\left\{\begin{array}{c}\varepsilon_1\\ \varepsilon_2\\ \varepsilon_3\\ \hline \gamma_{23}\\ \gamma_{13}\\ \gamma_{12}\end{array}\right\}=\left[\begin{array}{ccc|ccc}\frac{1}{E_1} & \frac{-\nu_{21}}{E_2} & \frac{-\nu_{21}}{E_2} & 0 & 0 & 0\\ \frac{-\nu_{12}}{E_1} & \frac{1}{E_2} & \frac{-\nu_{23}}{E_2} & 0 & 0 & 0\\ \frac{-\nu_{12}}{E_1} & \frac{-\nu_{23}}{E_2} & \frac{1}{E_2} & 0 & 0 & 0\\ \hline 0 & 0 & 0 & \frac{2(1+\nu_{23})}{E_2} & 0 & 0\\ 0 & 0 & 0 & 0 & \frac{1}{G_{12}} & 0\\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G_{12}}\end{array}\right]\left\{\begin{array}{c}\sigma_1\\ \sigma_2\\ \sigma_3\\ \hline \tau_{23}\\ \tau_{13}\\ \tau_{12}\end{array}\right\} \tag{17}$$

Special Orthotropic Lamina

Any layer is orthotropic in its own material coordinate system (1, 2, 3). A layer is called specially orthotropic when it is also orthotropic in the global coordinate system, which happens only for orientations $\theta = 0^\circ$ and $\theta = 90^\circ$, or for layers reinforced with balanced fabrics. Then, the stress and strain relations are expressed in terms of only 4 independent elastic constants (Q_{11}, Q_{22}, Q_{12}, Q_{66}) or (S_{11}, S_{22}, S_{12}, S_{66}) as shown below:

For $\theta = 0^\circ$,

$$\begin{Bmatrix}\sigma_x\\ \sigma_y\\ \tau_{xy}\end{Bmatrix}=\begin{Bmatrix}\sigma_1\\ \sigma_2\\ \tau_{12}\end{Bmatrix}=\begin{bmatrix}Q_{11} & Q_{12} & 0\\ Q_{12} & Q_{22} & 0\\ 0 & 0 & Q_{66}\end{bmatrix}\begin{Bmatrix}\varepsilon_1\\ \varepsilon_2\\ \gamma_{12}\end{Bmatrix},\quad \begin{Bmatrix}\varepsilon_x\\ \varepsilon_y\\ \gamma_{xy}\end{Bmatrix}=\begin{Bmatrix}\varepsilon_1\\ \varepsilon_2\\ \gamma_{12}\end{Bmatrix}=\begin{bmatrix}S_{11} & S_{12} & 0\\ S_{12} & S_{22} & 0\\ 0 & 0 & S_{66}\end{bmatrix}\begin{Bmatrix}\sigma_1\\ \sigma_2\\ \tau_{12}\end{Bmatrix} \tag{18}$$

For $\theta = 90^\circ$,

$$\begin{Bmatrix}\sigma_x\\ \sigma_y\\ \tau_{xy}\end{Bmatrix}=\begin{Bmatrix}\sigma_1\\ \sigma_2\\ \tau_{12}\end{Bmatrix}=\begin{bmatrix}Q_{22} & Q_{12} & 0\\ Q_{12} & Q_{11} & 0\\ 0 & 0 & Q_{66}\end{bmatrix}\begin{Bmatrix}\varepsilon_1\\ \varepsilon_2\\ \gamma_{12}\end{Bmatrix},\quad \begin{Bmatrix}\varepsilon_x\\ \varepsilon_y\\ \gamma_{xy}\end{Bmatrix}=\begin{Bmatrix}\varepsilon_1\\ \varepsilon_2\\ \gamma_{12}\end{Bmatrix}=\begin{bmatrix}S_{22} & S_{12} & 0\\ S_{12} & S_{11} & 0\\ 0 & 0 & S_{66}\end{bmatrix}\begin{Bmatrix}\sigma_1\\ \sigma_2\\ \tau_{12}\end{Bmatrix} \tag{19}$$

An orthotropic layer that is oriented at an angle not a multiple of 90° from the global coordinate system is called generally orthotropic. Therefore, $\bar{Q}_{16}$, $\bar{Q}_{26}$, $\bar{S}_{16}$ and $\bar{S}_{26}$ are different from zero for a generally orthotropic layer.

Isotropic Material

An orthotropic material has three mutually perpendicular planes of material symmetry, while an isotropic material has an infinite number of planes of material symmetry through a point. Thus, isotropy is a special case of orthotropic material. The elastic properties of isotropic materials are invariant with respect to directions. As the subscripts 1, 2, and 3 in the material constants are interchangeable, the stress and strain relations are reduced to:

$$\left\{\begin{array}{c}\sigma_1\\ \sigma_2\\ \sigma_3\\ \hline \tau_{23}\\ \tau_{13}\\ \tau_{12}\end{array}\right\}=\left[\begin{array}{ccc|ccc}C_{11} & C_{12} & C_{12} & 0 & 0 & 0\\ C_{12} & C_{11} & C_{12} & 0 & 0 & 0\\ C_{12} & C_{12} & C_{11} & 0 & 0 & 0\\ \hline 0 & 0 & 0 & (C_{11}-C_{12})/2 & 0 & 0\\ 0 & 0 & 0 & 0 & (C_{11}-C_{12})/2 & 0\\ 0 & 0 & 0 & 0 & 0 & (C_{11}-C_{12})/2\end{array}\right]\left\{\begin{array}{c}\varepsilon_1\\ \varepsilon_2\\ \varepsilon_3\\ \hline \gamma_{23}\\ \gamma_{13}\\ \gamma_{12}\end{array}\right\} \quad (20)$$

$$\left\{\begin{array}{c}\varepsilon_1\\ \varepsilon_2\\ \varepsilon_3\\ \hline \gamma_{23}\\ \gamma_{13}\\ \gamma_{12}\end{array}\right\}=\left[\begin{array}{ccc|ccc}S_{11} & S_{12} & S_{12} & 0 & 0 & 0\\ S_{12} & S_{11} & S_{12} & 0 & 0 & 0\\ S_{12} & S_{12} & S_{11} & 0 & 0 & 0\\ \hline 0 & 0 & 0 & 2(S_{11}-S_{12}) & 0 & 0\\ 0 & 0 & 0 & 0 & 2(S_{11}-S_{12}) & 0\\ 0 & 0 & 0 & 0 & 0 & 2(S_{11}-S_{12})\end{array}\right]\left\{\begin{array}{c}\sigma_1\\ \sigma_2\\ \sigma_3\\ \hline \tau_{23}\\ \tau_{13}\\ \tau_{12}\end{array}\right\} \quad (21)$$

Thus, an isotropic material is fully characterized by only two independent elastic constants, the stiffness C_{11}, C_{12} and the compliances S_{11}, S_{12}. It should be noted that the diagonal terms $(C_{ii},\ S_{ii}\quad i=1,\ 2,\ 3,\ 4,\ 5,\ 6)$ are always non zero for all kinds of composite materials.

The conclusions previously discussed regarding the required number of independent elastic constants for the various types of materials are summarized in Table 1.

TABLE 1. Independent Elastic Constants for Various Types of Materials

Material	No. of Independent Elastic Constants
General anisotropic material	81
With symmetry of stress and strain tensors	36
With elastic energy considerations	21
Monoclinic material	13
Orthotropic in principal material directions	9
Orthotropic material with transversely isotropy	5
Special orthotropic lamina	4
Isotropic material	2

Coordinate Systems

There are two coordinate systems that are used in composites design. The material coordinate system (denoted by axes 1, 2, 3) has the 1-axis aligned with the fiber direction. The 2-axis is perpendicular to the fibers in the plane of the lamina. The 3-axis is perpendicular to the plane of the lamina (plane 1-2). Each layer has its own material coordinate system aligned with the fiber direction.

The global coordinate system (denoted by axes *x, y, z*) is common to all the layers in the laminate. The orientation of the global system is chosen for convenience during the structural analysis. Therefore, it may be aligned with the boundary of the part being analyzed, with the direction of the major load, etc.

Off-Axis Stiffness

Practical composite structures have more than one layer because the properties in the transverse direction of a layer are relatively low when compared with the longitudinal properties. Therefore, several layers are stacked in different orientations so that reinforcements (fibers) are placed along all directions of loading.

A laminate is a set of layers with various fiber orientations which are bonded together to form a plate or shell. Before developing laminate properties, it is necessary to learn how to transform stresses, strains, stiffnesses, and compliances from the material coordinate system (1, 2, 3) to a global coordinate system ($x,\ y,\ z$). Once the properties of individual layers are known, the properties of a laminate can be obtained by combining the properties of the layers that form the laminate.

Stress and Strain Transformations

In stress analysis, sometimes a coordinate system *x*-*y* is set up which does not always coincide with the material principal axes, 1 and 2. The two sets of stress components with respect to these two coordinate systems are related by the transformation matrix $[T_\sigma]$. In the same manner, the strains with respect to the two coordinate systems are related by $[T_\varepsilon]$. The angle θ is measured in a positive counterclockwise fashion from the *x*-axis to the 1-axis, as shown in Figure 2. Then, the stress and strain components referred to the principal material axes (1, 2) can be expressed in terms of those referred to the loading axes (*x*, *y*) by the following transformation relations:

$$\begin{Bmatrix}\sigma_1\\ \sigma_2\\ \tau_{12}\end{Bmatrix}=[T_\sigma(\theta)]\begin{Bmatrix}\sigma_x\\ \sigma_y\\ \tau_{xy}\end{Bmatrix}=\begin{bmatrix}c^2 & s^2 & 2cs\\ s^2 & c^2 & -2cs\\ -cs & cs & c^2-s^2\end{bmatrix}\begin{Bmatrix}\sigma_x\\ \sigma_y\\ \tau_{xy}\end{Bmatrix} \qquad (22)$$

$$\begin{Bmatrix}\varepsilon_1\\ \varepsilon_2\\ \gamma_{12}\end{Bmatrix}=[T_\varepsilon(\theta)]\begin{Bmatrix}\varepsilon_x\\ \varepsilon_y\\ \gamma_{xy}\end{Bmatrix}=\begin{bmatrix}c^2 & s^2 & cs\\ s^2 & c^2 & -cs\\ -2cs & 2cs & c^2-s^2\end{bmatrix}\begin{Bmatrix}\varepsilon_x\\ \varepsilon_y\\ \gamma_{xy}\end{Bmatrix} \qquad (23)$$

where $c=\cos\theta$, and $s=\sin\theta$.

By inversion of the relations above we obtain

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = [T_\sigma(\theta)]^{-1} \begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{Bmatrix} = [T_\sigma(-\theta)] \begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{Bmatrix} = \begin{bmatrix} c^2 & s^2 & -2cs \\ s^2 & c^2 & 2cs \\ cs & -cs & c^2 - s^2 \end{bmatrix} \begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{Bmatrix} \tag{24}$$

$$\begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = [T_\varepsilon(\theta)]^{-1} \begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \gamma_{12} \end{Bmatrix} = [T_\varepsilon(-\theta)] \begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \gamma_{12} \end{Bmatrix} = \begin{bmatrix} c^2 & s^2 & -cs \\ s^2 & c^2 & cs \\ 2cs & -2cs & c^2 - s^2 \end{bmatrix} \begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \gamma_{12} \end{Bmatrix} \tag{25}$$

Note that the inverses $[T_\sigma(\theta)]^{-1}$ and $[T_\varepsilon(\theta)]^{-1}$ can be obtained by replacing θ with $-\theta$ into $[T_\sigma(\theta)]$ and $[T_\varepsilon(\theta)]$, respectively. Also, it should be noted that the laws of stress and strain transformation are independent of material properties.

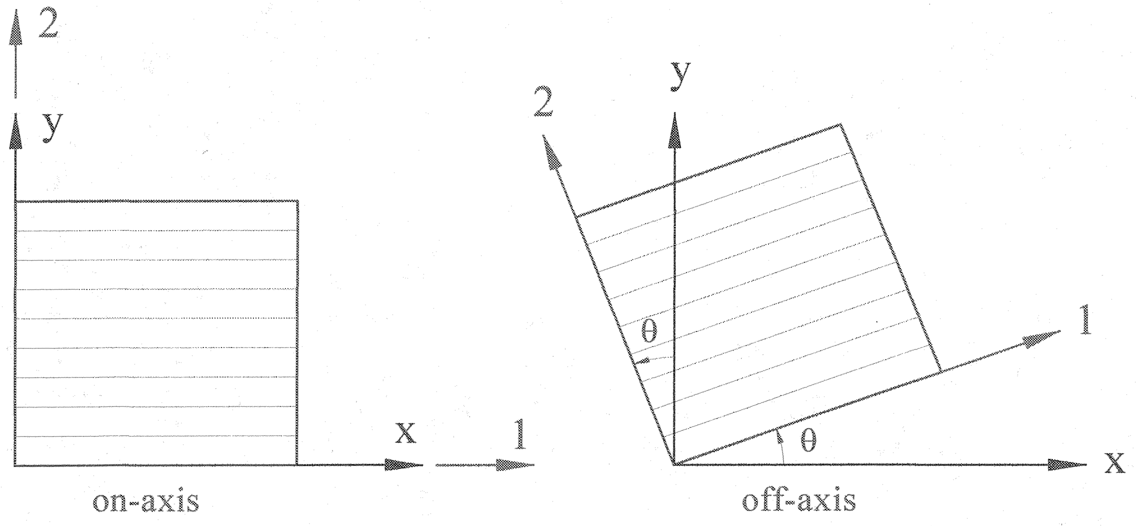

FIGURE 2. Coordinate transformations.

Stiffness and Compliance Transformations

In most structural applications, composite materials are used in the form of thin laminates loaded in the plane of the laminate. Thus, composite laminae (and laminates) can be considered to be under a condition of plane stress, with all stress components in the out-of-plane direction (3-direction) being zero, that is,

$$\sigma_3 = 0, \quad \tau_{23} = 0, \quad \tau_{13} = 0 \tag{26}$$

where the out-of-plane strains

$$\gamma_{23} = 0, \quad \gamma_{13} = 0, \quad \varepsilon_3 = S_{13}\sigma_1 + S_{23}\sigma_2 \neq 0 \tag{27}$$

The orthotropic stress-strain relations, Eq. (12), are reduced to

$$\left\{\begin{array}{c} \sigma_1 \\ \sigma_2 \\ 0 \\ \hdashline 0 \\ 0 \\ \tau_{12} \end{array}\right\} = \left[\begin{array}{ccc:ccc} C_{11} & C_{12} & C_{13} & 0 & 0 & 0 \\ C_{12} & C_{22} & C_{23} & 0 & 0 & 0 \\ C_{13} & C_{23} & C_{33} & 0 & 0 & 0 \\ \hdashline 0 & 0 & 0 & C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{66} \end{array}\right] \left\{\begin{array}{c} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \hdashline 0 \\ 0 \\ \gamma_{12} \end{array}\right\} \tag{28}$$

After eliminating ε_3, the in-plane stress and strain relations for an orthotropic layer under plane stress can be expressed in terms of only four independent elastic constants, that is, the reduced stiffnesses Q_{11}, Q_{22}, Q_{12}, and Q_{66} or the compliances S_{11}, S_{22}, S_{12}, and S_{66} as show below:

$$\left\{\begin{array}{c} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{array}\right\} = [Q]\left\{\begin{array}{c} \varepsilon_1 \\ \varepsilon_2 \\ \gamma_{12} \end{array}\right\} = \begin{bmatrix} Q_{11} & Q_{12} & 0 \\ Q_{12} & Q_{22} & 0 \\ 0 & 0 & Q_{66} \end{bmatrix} \left\{\begin{array}{c} \varepsilon_1 \\ \varepsilon_2 \\ \gamma_{12} \end{array}\right\} \tag{29}$$

$$\left\{\begin{array}{c} \varepsilon_1 \\ \varepsilon_2 \\ \gamma_{12} \end{array}\right\} = [S]\left\{\begin{array}{c} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{array}\right\} = \begin{bmatrix} S_{11} & S_{12} & 0 \\ S_{12} & S_{22} & 0 \\ 0 & 0 & S_{66} \end{bmatrix} \left\{\begin{array}{c} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{array}\right\} \tag{30}$$

The relations above can be expressed in terms of engineering constants by noting that

$$Q_{11} = \frac{E_1}{1 - \nu_{12}\nu_{21}} = \frac{S_{22}}{S_{11}S_{22} - S_{12}^2} \qquad S_{11} = \frac{1}{E_1}$$
$$Q_{22} = \frac{E_2}{1 - \nu_{12}\nu_{21}} = \frac{S_{11}}{S_{11}S_{22} - S_{12}^2} \qquad S_{22} = \frac{1}{E_2}$$

$$\begin{aligned} Q_{12} &= \frac{\nu_{12}E_2}{1-\nu_{12}\nu_{21}} = \frac{-S_{12}}{S_{11}S_{22}-S_{12}^2} & \qquad S_{12} &= \frac{-\nu_{12}}{E_1} = \frac{-\nu_{21}}{E_2} \\ Q_{66} &= G_{12} = \frac{1}{S_{66}} & \qquad S_{66} &= \frac{1}{G_{12}} \end{aligned} \tag{31}$$

The stress and strain relations in equations (29) and (30) show that, when the lamina is loaded along the principal material axes, there is no coupling between normal stresses and shear deformation and between shear stress and normal strains. This is not the case when the lamina is loaded along the arbitrary axes x and y. In that case, the stress and strain relations in global coordinates take the form:

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \left[\overline{Q}\right] \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = \begin{bmatrix} \overline{Q}_{11} & \overline{Q}_{12} & \overline{Q}_{16} \\ \overline{Q}_{12} & \overline{Q}_{22} & \overline{Q}_{26} \\ \overline{Q}_{16} & \overline{Q}_{26} & \overline{Q}_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} \tag{32}$$

$$\begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = \left[\overline{S}\right] \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \begin{bmatrix} \overline{S}_{11} & \overline{S}_{12} & \overline{S}_{16} \\ \overline{S}_{12} & \overline{S}_{22} & \overline{S}_{26} \\ \overline{S}_{16} & \overline{S}_{26} & \overline{S}_{66} \end{bmatrix} \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \begin{bmatrix} \dfrac{1}{E_x} & \dfrac{-\nu_{yx}}{E_y} & \dfrac{\eta_{xy,x}}{G_{xy}} \\ \dfrac{-\nu_{xy}}{E_x} & \dfrac{1}{E_y} & \dfrac{\eta_{xy,y}}{G_{xy}} \\ \dfrac{\eta_{x,xy}}{E_x} & \dfrac{\eta_{y,xy}}{E_y} & \dfrac{1}{G_{xy}} \end{bmatrix} \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} \tag{33}$$

where the transformed reduced stiffness $\left[\overline{Q}\right]$ are related to the principal lamina reduced stiffness $[Q]$ by:

$$\begin{aligned} \overline{Q}_{11} &= Q_{11}c^4 + 2(Q_{12}+2Q_{66})s^2c^2 + Q_{22}s^4 \\ \overline{Q}_{22} &= Q_{11}s^4 + 2(Q_{12}+2Q_{66})s^2c^2 + Q_{22}c^4 \\ \overline{Q}_{12} &= (Q_{11}+Q_{22}-4Q_{66})s^2c^2 + Q_{12}\left(s^4+c^4\right) \\ \overline{Q}_{66} &= (Q_{11}+Q_{22}-2Q_{12}-2Q_{66})s^2c^2 + Q_{66}\left(s^4+c^4\right) \\ \overline{Q}_{16} &= (Q_{11}-Q_{12}-2Q_{66})sc^3 + (Q_{12}-Q_{22}+2Q_{66})s^3c \\ \overline{Q}_{26} &= (Q_{11}-Q_{12}-2Q_{66})s^3c + (Q_{12}-Q_{22}+2Q_{66})sc^3 \end{aligned} \tag{34}$$

Similarly, the transformed compliances $\left[\overline{S}\right]$are related to the principal lamina compliances $[S]$ by:

$$
\begin{aligned}
\overline{S}_{11} &= \frac{1}{E_x} = S_{11}\,c^4 + \left(2S_{12} + S_{66}\right)s^2c^2 + S_{22}\,s^4 \\
\overline{S}_{22} &= \frac{1}{E_y} = S_{11}\,s^4 + \left(2S_{12} + S_{66}\right)s^2c^2 + S_{22}\,c^4 \\
\overline{S}_{12} &= \frac{-\nu_{xy}}{E_x} = \frac{-\nu_{yx}}{E_y} = \left(S_{11} + S_{22} - S_{66}\right)s^2c^2 + S_{12}\left(s^4 + c^4\right) \\
\overline{S}_{66} &= \frac{1}{G_{xy}} = 2\left(2S_{11} + 2S_{22} - 4S_{12} - S_{66}\right)s^2c^2 + S_{66}\left(s^4 + c^4\right) \\
\overline{S}_{16} &= \frac{\eta_{x,xy}}{E_x} = \frac{\eta_{xy,x}}{G_{xy}} = \left(2S_{11} - 2S_{12} - S_{66}\right)sc^3 + \left(2S_{12} - 2S_{22} + S_{66}\right)s^3c \\
\overline{S}_{26} &= \frac{\eta_{y,xy}}{E_y} = \frac{\eta_{xy,y}}{G_{xy}} = \left(2S_{11} - 2S_{12} - S_{66}\right)s^3c + \left(2S_{12} - 2S_{22} + S_{66}\right)sc^3
\end{aligned}
\tag{35}
$$

The transformed compliances can be expressed in terms of engineering constants as:

$$
\begin{aligned}
\overline{S}_{11} &= \frac{1}{E_x} = \frac{1}{E_1}c^4 + \left(\frac{1}{G_{12}} - \frac{2\nu_{12}}{E_1}\right)s^2c^2 + \frac{1}{E_2}s^4 \\
\overline{S}_{22} &= \frac{1}{E_y} = \frac{1}{E_1}s^4 + \left(\frac{1}{G_{12}} - \frac{2\nu_{12}}{E_1}\right)s^2c^2 + \frac{1}{E_2}c^4 \\
\overline{S}_{12} &= \frac{-\nu_{xy}}{E_x} = \frac{-\nu_{yx}}{E_y} = \left(\frac{1}{E_1} + \frac{1}{E_2} - \frac{1}{G_{12}}\right)s^2c^2 - \frac{\nu_{12}}{E_1}\left(s^4 + c^4\right) \\
\overline{S}_{66} &= \frac{1}{G_{xy}} = 2\left(\frac{2}{E_1} + \frac{2}{E_2} + \frac{4\nu_{12}}{E_1} - \frac{1}{G_{12}}\right)s^2c^2 + \frac{1}{G_{12}}\left(s^4 + c^4\right) \\
\overline{S}_{16} &= \frac{\eta_{x,xy}}{E_x} = \frac{\eta_{xy,x}}{G_{xy}} = \left(\frac{2}{E_1} + \frac{2\nu_{12}}{E_1} - \frac{1}{G_{12}}\right)sc^3 - \left(\frac{2}{E_2} + \frac{2\nu_{12}}{E_1} - \frac{1}{G_{12}}\right)s^3c \\
\overline{S}_{26} &= \frac{\eta_{y,xy}}{E_y} = \frac{\eta_{xy,y}}{G_{xy}} = \left(\frac{2}{E_1} + \frac{2\nu_{12}}{E_1} - \frac{1}{G_{12}}\right)s^3c - \left(\frac{2}{E_2} + \frac{2\nu_{12}}{E_1} - \frac{1}{G_{12}}\right)sc^3
\end{aligned}
\tag{36}
$$

Note the coefficients $\overline{Q}_{16}$, $\overline{Q}_{26} \neq 0$ and $\overline{S}_{16}$, $\overline{S}_{26} \neq 0$ after the transformation. If the global axis does not coincide with the material axis when the shear stress is applied, not only the shear strain but also the normal strains are produced.

Macromechanical Behavior of Composite Laminates

A laminate is two or more laminae bonded together to act as an integral structural element. The two basic questions of laminate analysis are: (1) What are the conditions that the laminae must meet to be a laminate? and (2) How will a laminate respond to loading, i.e., imposed forces and moments? The various laminae are oriented with (local) principal material directions at different angles to the global laminate axes to produce a structural element capable of resisting load in several directions.

The laminae are combined to create a laminate in order to achieve the largest possible bending stiffness for the materials used. For example, let's first consider that the two beams are not fastened together and loaded at midspan by a concentrated force. In contrast, the same two beams could be fastened together by nails, screws, or bonding. The deflection is less for the bonded beams than the unbonded beams by a factor of four! Thus, bonding laminae together can result in a compellingly large increase in bending resistance.

Basic Assumptions and Restrictions of Classical Lamination Theory

It is apparent that the overall behavior of a multidirectional laminate is a function of the properties and stacking sequence of the individual layers. The so-called Classical Lamination Theory predicts the behavior of the laminate within the framework of the following assumptions and restrictions:

1. Each layer (lamina) of the laminate is quasi-homogeneous and orthotropic.

2. The laminate is thin with its lateral dimensions much larger than its thickness and is loaded in its plane only, that is, the laminate and its layers (except for their edges) are in a state of plane stress $\left(\sigma_z = \tau_{xz} = \tau_{yz} = 0\right)$.

3. All displacements are small compared with the thickness of the laminate ($|u|, |v|, |w| << h$).

4. Displacements are continuous throughout the laminate.

5. In-plane displacements vary linearly through the thickness of the laminate, that is, u and v displacements in the x- and y-directions are linear functions of z.

6. Straight lines normal to the middle surface remain straight and normal to that surface after deformation. This implies that transverse shear strains γ_{xz} and γ_{yz} are zero.

7. Strain-displacement and stress-strain relations are linear.

8. Normal distances from the middle surface remain constant, that is, the transverse normal strain ε_z is zero. This implies that the transverse displacement w is independent of the thickness coordinate z.

Strain and Stress Variation in a Laminate

Knowledge of the variation of stress and strain through the laminate thickness is essential to the definition of the extensional and bending stiffnesses of a laminate. The laminate is presumed to consist of perfectly bonded laminae. Moreover, the bonds are presumed to be infinitesimally thin as well as non-shear-deformable. That is, the displacements are continuous across lamina boundaries so that no lamina can slip relative to another.

The midplane displacements u° and v° in the x- and y-directions and the out-of-plane displacement w in the z-direction are functions of x and y only:

$$u^{\circ} = u^{\circ}(x,y), \quad v^{\circ} = v^{\circ}(x,y), \quad w = f(x,y) \tag{37}$$

The rotations of the x- and y-axes are

$$\alpha_x = \frac{\partial w}{\partial x}\ , \qquad \alpha_y = \frac{\partial w}{\partial y} \tag{38}$$

Then the strain components on the midplane (or reference plane) are expressed as

$$\varepsilon_x^{\circ} = \frac{\partial u^{\circ}}{\partial x}, \quad \varepsilon_y^{\circ} = \frac{\partial v^{\circ}}{\partial y}, \quad \gamma_{xy}^{\circ} = \frac{\partial u^{\circ}}{\partial y} + \frac{\partial v^{\circ}}{\partial x} \tag{39}$$

and the midplane curvatures are

$$\begin{aligned}
\kappa_x &= \frac{-\partial \alpha_x}{\partial x} = -\frac{\partial^2 w}{\partial x^2} \\
\kappa_y &= \frac{-\partial \alpha_y}{\partial y} = -\frac{\partial^2 w}{\partial y^2} \\
\kappa_{xy} &= \frac{-\partial \alpha_x}{\partial y} - \frac{\partial \alpha_y}{\partial x} = -2\frac{\partial^2 w}{\partial x \partial y}
\end{aligned} \tag{40}$$

The laminate strains have been reduced to ε_x, ε_y, and γ_{xy} by virtue of the Kirchhoff hypothesis. That is, $\varepsilon_z = \gamma_{xz} = \gamma_{yz} = 0$. For small displacements, the classical strain-displacement relations of elasticity yield

$$\begin{aligned}
\varepsilon_x &= \frac{\partial u}{\partial x} = \frac{\partial u^{\circ}}{\partial x} - z\frac{\partial^2 w}{\partial x^2} = \varepsilon_x^{\circ} + z\kappa_x \\
\varepsilon_y &= \frac{\partial v}{\partial y} = \frac{\partial v^{\circ}}{\partial y} - z\frac{\partial^2 w}{\partial y^2} = \varepsilon_y^{\circ} + z\kappa_y \\
\gamma_{xy} &= \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} = \frac{\partial u^{\circ}}{\partial y} + \frac{\partial v^{\circ}}{\partial x} - 2z\frac{\partial^2 w}{\partial x \partial y} = \gamma_{xy}^{\circ} + z\kappa_{xy}
\end{aligned} \tag{41}$$

After substitution of equations (39) and (40) into (41), one can relate the strains at any point in the laminate to the reference plane strains and curvatures as follows:

$$\begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = \begin{Bmatrix} \varepsilon_x^{\circ} \\ \varepsilon_y^{\circ} \\ \gamma_{xy}^{\circ} \end{Bmatrix} + z\begin{Bmatrix} \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{Bmatrix} \tag{42}$$

The main reason for separating the strains into in-plane strains $(\varepsilon_x^{\circ}, \varepsilon_y^{\circ}, \gamma_{xy}^{\circ})$ and curvatures $(\kappa_x, \kappa_y, \kappa_{xy})$ is for convenience. The underlying motivation is to write all the functions (displacements, stresses, and strains) in terms of only two variables (*x* and *y*) to simplify the analysis. For most plates, if only in-plane forces are applied, only $(\varepsilon_x^\circ, \varepsilon_y^\circ, \gamma_{xy}^\circ)$ are induced and it is not necessary to consider the curvatures. Similarly, if only moments are applied to the plate, then only the curvatures need to be computed.

Since the thickness of the laminate is small compared with the in-plane dimensions of the plate, every layer is in a state of plane stress. Therefore, the stress-strain relations for a layer *k* in material coordinates are

$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{Bmatrix}^k = \begin{bmatrix} Q_{11} & Q_{12} & 0 \\ Q_{12} & Q_{22} & 0 \\ 0 & 0 & Q_{66} \end{bmatrix}^k \begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \gamma_{12} \end{Bmatrix}^k \tag{43}$$

After transformation, the stress-strain relations in global coordinates become:

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix}^k = \begin{bmatrix} \overline{Q}_{11} & \overline{Q}_{12} & \overline{Q}_{16} \\ \overline{Q}_{12} & \overline{Q}_{22} & \overline{Q}_{26} \\ \overline{Q}_{16} & \overline{Q}_{26} & \overline{Q}_{66} \end{bmatrix}^k \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix}^k \tag{44}$$

By substitution of the strain variation through the thickness (equation (42)) into the stress-strain relations (equation (44)) the stresses in the k^{th} layer can be expressed in terms of the laminate middle-surface strains and curvatures as:

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix}^k = \begin{bmatrix} \overline{Q}_{11} & \overline{Q}_{12} & \overline{Q}_{16} \\ \overline{Q}_{12} & \overline{Q}_{22} & \overline{Q}_{26} \\ \overline{Q}_{16} & \overline{Q}_{26} & \overline{Q}_{66} \end{bmatrix}^k \begin{Bmatrix} \varepsilon_x^\circ \\ \varepsilon_y^\circ \\ \gamma_{xy}^\circ \end{Bmatrix} + z \begin{bmatrix} \overline{Q}_{11} & \overline{Q}_{12} & \overline{Q}_{16} \\ \overline{Q}_{12} & \overline{Q}_{22} & \overline{Q}_{26} \\ \overline{Q}_{16} & \overline{Q}_{26} & \overline{Q}_{66} \end{bmatrix}^k \begin{Bmatrix} \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{Bmatrix} \tag{45}$$

From equations (42) and (45), it is seen that, whereas the strains vary linearly through the thickness, the stresses do not, because of the discontinuous variation of the transformed stiffness matrix $[\overline{Q}]$ from layer to layer. Instead, the stresses are piecewise linear (i.e., linear in each layer, but discontinuous at boundaries between laminae).

Resultant Laminate Forces and Moments

Because of the discontinuous variation of stresses from layer to layer, it is more convenient to deal with the integrated effect of these stresses on the laminate. The stresses are integrated over the thickness (*h*) of the plate to obtain the resultant forces and moments on a laminate as

$$\begin{Bmatrix} N_x \\ N_y \\ N_{xy} \end{Bmatrix} = \int_{-h/2}^{h/2} \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} dz, \qquad \begin{Bmatrix} M_x \\ M_y \\ M_{xy} \end{Bmatrix} = \int_{-h/2}^{h/2} \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} z\, dz \tag{46}$$

The integrations in (46) span over several layers. Therefore, the integrals can be divided into summations of integrals over each layer

$$\begin{Bmatrix} N_x \\ N_y \\ N_{xy} \end{Bmatrix} = \sum_{k=1}^{n} \int_{z_{k-1}}^{z_k} \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix}^k dz, \qquad \begin{Bmatrix} M_x \\ M_y \\ M_{xy} \end{Bmatrix} = \sum_{k=1}^{n} \int_{z_{k-1}}^{z_k} \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix}^k zdz \tag{47}$$

where

z = through-the-thickness coordinate of a point in the cross section

N_x, N_y = normal forces per unit length

N_{xy} = shear force per unit length

M_x, M_y = bending moments per unit length

M_{xy} = twisting moment per unit length

n = number of layers in the laminate

k = layer number counting from the bottom up

z_k, z_{k-1} = *z*-coordinates of the upper and lower surfaces of layer *k*.

Laminate Stiffnesses

Substituting equation (45) for the layer stresses in equation (47), we obtain

$$\begin{Bmatrix} N_x \\ N_y \\ N_{xy} \end{Bmatrix} = \sum_{k=1}^{n} \left\{ \begin{bmatrix} \overline{Q}_{11} & \overline{Q}_{12} & \overline{Q}_{16} \\ \overline{Q}_{12} & \overline{Q}_{22} & \overline{Q}_{26} \\ \overline{Q}_{16} & \overline{Q}_{26} & \overline{Q}_{66} \end{bmatrix}^k \begin{Bmatrix} \varepsilon^\circ_x \\ \varepsilon^\circ_y \\ \gamma^\circ_{xy} \end{Bmatrix} \int_{z_{k-1}}^{z_k} dz + \begin{bmatrix} \overline{Q}_{11} & \overline{Q}_{12} & \overline{Q}_{16} \\ \overline{Q}_{12} & \overline{Q}_{22} & \overline{Q}_{26} \\ \overline{Q}_{16} & \overline{Q}_{26} & \overline{Q}_{66} \end{bmatrix}^k \begin{Bmatrix} \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{Bmatrix} \int_{z_{k-1}}^{z_k} zdz \right\} \tag{48}$$

and

$$\begin{Bmatrix} M_x \\ M_y \\ M_{xy} \end{Bmatrix} = \sum_{k=1}^{n} \left\{ \begin{bmatrix} \overline{Q}_{11} & \overline{Q}_{12} & \overline{Q}_{16} \\ \overline{Q}_{12} & \overline{Q}_{22} & \overline{Q}_{26} \\ \overline{Q}_{16} & \overline{Q}_{26} & \overline{Q}_{66} \end{bmatrix}^k \begin{Bmatrix} \varepsilon^\circ_x \\ \varepsilon^\circ_y \\ \gamma^\circ_{xy} \end{Bmatrix} \int_{z_{k-1}}^{z_k} zdz + \begin{bmatrix} \overline{Q}_{11} & \overline{Q}_{12} & \overline{Q}_{16} \\ \overline{Q}_{12} & \overline{Q}_{22} & \overline{Q}_{26} \\ \overline{Q}_{16} & \overline{Q}_{26} & \overline{Q}_{66} \end{bmatrix}^k \begin{Bmatrix} \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{Bmatrix} \int_{z_{k-1}}^{z_k} z^2dz \right\} \tag{49}$$

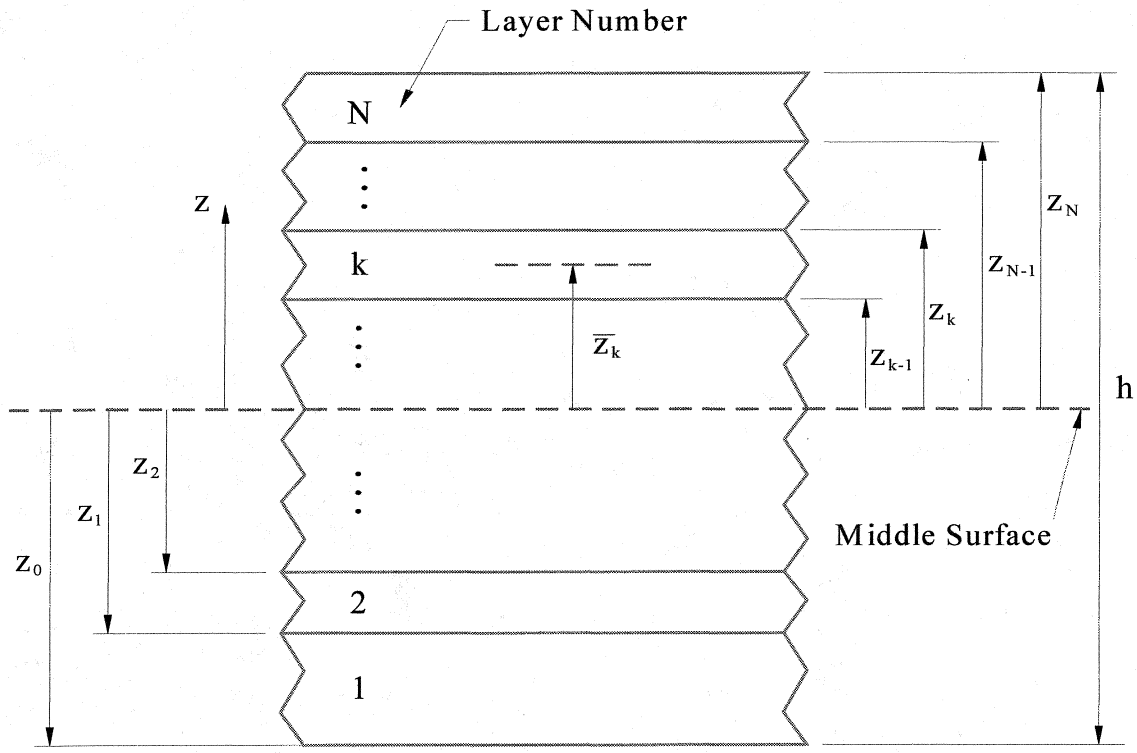

FIGURE 3. Geometry of an N-layer laminate.

In the expressions above, the stiffnesses $[\overline{Q}]^k$, reference plane strains $[\varepsilon^\circ]_{xy}$, and curvatures $[\kappa]_{xy}$ are taken outside the integration operation since they are not functions of z. Of these quantities, only the stiffnesses are unique for each layer k, whereas the reference plane strains and curvatures refer to the entire laminate and are the same for all plies. Thus, $[\varepsilon^{\circ}]_{xy}$ and $[\kappa]_{xy}$ can be factored outside the summation sign as follows:

$$
\begin{aligned}
[N]_{xy} &= \left[\sum_{k=1}^{n}[\overline{Q}]^{k}\int_{z_{k-1}}^{z_k} dz\right][\varepsilon^{\circ}]_{xy} + \left[\sum_{k=1}^{n}[\overline{Q}]^{k}\int_{z_{k-1}}^{z_k} z\,dz\right][\kappa]_{xy} \\
&= \left[\sum_{k=1}^{n}[\overline{Q}]^{k}(z_k - z_{k-1})\right][\varepsilon^{\circ}]_{xy} + \left[\frac{1}{2}\sum_{k=1}^{n}[\overline{Q}]^{k}(z_k^2 - z_{k-1}^2)\right][\kappa]_{xy} \qquad (50) \\
&= [A][\varepsilon^{\circ}]_{xy} + [B][\kappa]_{xy}
\end{aligned}
$$

$$
\begin{aligned}
[M]_{xy} &= \left[\sum_{k=1}^{n}[\overline{Q}]^{k}\int_{z_{k-1}}^{z_k} z\,dz\right][\varepsilon^{\circ}]_{xy} + \left[\sum_{k=1}^{n}[\overline{Q}]^{k}\int_{z_{k-1}}^{z_k} z^2 dz\right][\kappa]_{xy} \\
&= \left[\frac{1}{2}\sum_{k=1}^{n}[\overline{Q}]^{k}(z_k^2 - z_{k-1}^2)\right][\varepsilon^{\circ}]_{xy} + \left[\frac{1}{3}\sum_{k=1}^{n}[\overline{Q}]^{k}(z_k^3 - z_{k-1}^3)\right][\kappa]_{xy} \qquad (51) \\
&= [B][\varepsilon^{\circ}]_{xy} + [D][\kappa]_{xy}
\end{aligned}
$$

with

$$
\begin{aligned}
A_{ij} &= \sum_{k=1}^{n}\overline{Q}_{ij}^{k}(z_k - z_{k-1}) = \sum_{k=1}^{n}\overline{Q}_{ij}^{k}\,t_k && i, j = 1, 2, 6 \\
B_{ij} &= \frac{1}{2}\sum_{k=1}^{n}\overline{Q}_{ij}^{k}\left(z_k^2 - z_{k-1}^2\right) = \sum_{k=1}^{n}\overline{Q}_{ij}^{k}\,t_k \overline{z}_k && i, j = 1, 2, 6 \qquad (52) \\
D_{ij} &= \frac{1}{3}\sum_{k=1}^{n}\overline{Q}_{ij}^{k}\left(z_k^3 - z_{k-1}^3\right) = \sum_{k=1}^{n}\overline{Q}_{ij}^{k}\left(t_k \overline{z}_k^2 + \frac{t_k^3}{12}\right) && i, j = 1, 2, 6
\end{aligned}
$$

The coefficients A_{ij}, B_{ij} and D_{ij} are function of the thickness, orientation, stacking sequence, and material properties of the layers. Each coefficient has a particular role in the analysis of the laminate, as follows:

A_{ij} = extensional stiffnesses, or in-plane stiffnesses because it directly relates in-plane strains $(\varepsilon_x^\circ, \varepsilon_y^\circ, \gamma_{xy}^\circ)$ to in-plane forces (N_x, N_y, N_{xy}).

B_{ij} = bending-extension coupling stiffnesses because it relates in-plane strains to bending moments and curvatures to in-plane forces. This coupling effect does not exist for homogeneous plates.

D_{ij} = bending or flexural stiffnesses because it relates curvatures $(\kappa_x, \kappa_y, \kappa_{xy})$ to bending moments (M_x, M_y, M_{xy}).

$\bar{z}_k$ = coordinate of the middle surface of the k^{th} layer (Figure 3).

t = layer thickness.

Thus, in full form the force-deformation and the moment-deformation relations are

$$\begin{Bmatrix} N_x \\ N_y \\ N_{xy} \end{Bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{16} \\ A_{21} & A_{22} & A_{26} \\ A_{61} & A_{62} & A_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_x^\circ \\ \varepsilon_y^\circ \\ \gamma_{xy}^\circ \end{Bmatrix} + \begin{bmatrix} B_{11} & B_{12} & B_{16} \\ B_{21} & B_{22} & B_{26} \\ B_{61} & B_{62} & B_{66} \end{bmatrix} \begin{Bmatrix} \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{Bmatrix} \tag{53}$$

$$\begin{Bmatrix} M_x \\ M_y \\ M_{xy} \end{Bmatrix} = \begin{bmatrix} B_{11} & B_{12} & B_{16} \\ B_{21} & B_{22} & B_{26} \\ B_{61} & B_{62} & B_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_x^\circ \\ \varepsilon_y^\circ \\ \gamma_{xy}^\circ \end{Bmatrix} + \begin{bmatrix} D_{11} & D_{12} & D_{16} \\ D_{21} & D_{22} & D_{26} \\ D_{61} & D_{62} & D_{66} \end{bmatrix} \begin{Bmatrix} \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{Bmatrix} \tag{54}$$

The expressions above can be combined into one general expression relating in-plane forces and moments to reference plane strains and curvatures:

$$\left\{ \begin{array}{c} N_x \\ N_y \\ N_{xy} \\ \hdashline M_x \\ M_y \\ M_{xy} \end{array} \right\} = \left[\begin{array}{ccc:ccc} A_{11} & A_{12} & A_{16} & B_{11} & B_{12} & B_{16} \\ A_{21} & A_{22} & A_{26} & B_{21} & B_{22} & B_{26} \\ A_{61} & A_{62} & A_{66} & B_{61} & B_{62} & B_{66} \\ \hdashline B_{11} & B_{12} & B_{16} & D_{11} & D_{12} & D_{16} \\ B_{21} & B_{22} & B_{26} & D_{21} & D_{22} & D_{26} \\ B_{61} & B_{62} & B_{66} & D_{61} & D_{62} & D_{66} \end{array} \right] \left\{ \begin{array}{c} \varepsilon_x^\circ \\ \varepsilon_y^\circ \\ \gamma_{xy}^\circ \\ \hdashline \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{array} \right\} \tag{55}$$

or, in brief,

$$\left\{\begin{array}{c} N \\ \hline M \end{array}\right\}_{6x1} = \left[\begin{array}{c|c} A & B \\ \hline B & D \end{array}\right]_{6x6} \left\{\begin{array}{c} \varepsilon^{\circ} \\ \hline \kappa \end{array}\right\}_{6x1} \tag{56}$$

The 6x6 stiffness matrix in (55) is composed of three submatrices, [*A*], [*B*], and [*D*], each 3x3 in size. All three are symmetric matrices, that is,

$$A_{ij} = A_{ji}, \qquad B_{ij} = B_{ji}, \qquad D_{ij} = D_{ji} \qquad \left(i,\ j = 1,\ 2,\ 6\right) \tag{57}$$

Equations (55) are called stiffness equations because of the analogy of Hooke's law $\sigma = E\varepsilon$, where *E* is the modulus or stiffness of the material.

Laminate Compliances

Since multidirectional laminates are characterized by stress discontinuities from ply to ply, it is preferable to work with strains, which are continuous through the thickness. For this reason it is necessary to invert the load-deformation relations, equation (55), and express strains and curvatures as a function of applied loads and moments, as follows:

$$\left\{\begin{array}{c} \varepsilon_x^{\circ} \\ \varepsilon_y^{\circ} \\ \gamma_{xy}^{\circ} \\ \hline \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{array}\right\} = \left[\begin{array}{ccc|ccc} a_{11} & a_{12} & a_{16} & b_{11} & b_{12} & b_{16} \\ a_{12} & a_{22} & a_{26} & b_{21} & b_{22} & b_{26} \\ a_{16} & a_{26} & a_{66} & b_{61} & b_{62} & b_{66} \\ \hline b_{11} & b_{21} & b_{61} & d_{11} & d_{12} & d_{16} \\ b_{12} & b_{22} & b_{62} & d_{12} & d_{22} & d_{26} \\ b_{16} & b_{26} & b_{66} & d_{16} & d_{26} & d_{66} \end{array}\right] \left\{\begin{array}{c} N_x \\ N_y \\ N_{xy} \\ \hline M_x \\ M_y \\ M_{xy} \end{array}\right\} \tag{58}$$

$$\left\{\begin{array}{c} \varepsilon_x^{\circ} \\ \varepsilon_y^{\circ} \\ \gamma_{xy}^{\circ} \\ \hline \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{array}\right\} = \left[\begin{array}{ccc|ccc} & & & b_{11} & b_{12} & b_{16} \\ & \text{sym} & & b_{21} & b_{22} & b_{26} \\ & & & b_{61} & b_{62} & b_{66} \\ \hline b_{11} & b_{21} & b_{61} & & & \\ b_{12} & b_{22} & b_{62} & & \text{sym} & \\ b_{16} & b_{26} & b_{66} & & & \end{array}\right] \left\{\begin{array}{c} N_x \\ N_y \\ N_{xy} \\ \hline M_x \\ M_y \\ M_{xy} \end{array}\right\} \tag{59}$$

or, in brief,

$$\left\{\begin{array}{c} \varepsilon^{\circ} \\ \hline \kappa \end{array}\right\}_{6x1} = \left[\begin{array}{c|c} A & B \\ \hline B & D \end{array}\right]_{6x6}^{-1} \left\{\begin{array}{c} N \\ \hline M \end{array}\right\}_{6x1} = \left[\begin{array}{c|c} a & b \\ \hline b^{T} & d \end{array}\right]_{6x6} \left\{\begin{array}{c} N \\ \hline M \end{array}\right\}_{6x1} \tag{60}$$

Here, matrices $\left[a\right]$, $\left[b\right]$, and $\left[d\right]$ are the laminate compliance matrices and $\left[b\right]^T$ is the transpose of matrix $\left[b\right]$ obtained from it by interchanging columns and rows.

Computation of Stresses

Once the $\left[A\right]$, $\left[B\right]$, and $\left[D\right]$ matrices have been formulated, strains and curvatures at the middle surface can be computed by solving (58) for a given set of stress resultants. Once the middle surface strains $(\varepsilon_x^\circ, \varepsilon_y^\circ, \gamma_{xy}^\circ)$ and the curvatures $(\kappa_x, \kappa_y, \kappa_{xy})$ are known, the strains $(\varepsilon_x, \varepsilon_y, \gamma_{xy})$ can be computed at any point through the thickness of the plate using (42), which is repeated here for convenience

$$\begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = \begin{Bmatrix} \varepsilon^\circ_x \\ \varepsilon^\circ_y \\ \gamma^\circ_{xy} \end{Bmatrix} + z \begin{Bmatrix} \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{Bmatrix} \tag{61}$$

The stresses can be computed on each layer using the constitutive equations of that particular layer, (44), which is repeated here

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix}^k = \begin{bmatrix} \overline{Q}_{11} & \overline{Q}_{12} & \overline{Q}_{16} \\ \overline{Q}_{12} & \overline{Q}_{22} & \overline{Q}_{26} \\ \overline{Q}_{16} & \overline{Q}_{26} & \overline{Q}_{66} \end{bmatrix}^k \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} \tag{62}$$

where the superscript (*k*) indicates that these equations apply to the *k*th layer.

Failure Criteria

Failure of fiber-reinforced composites may be caused by fiber buckling, fiber breakage, matrix cracking, delamination, or by a combination of these factors.

Local fiber buckling, or microbuckling, reduces the compressive stiffness and strength of the laminate. Microbuckling does not necessarily lead to immediate failure because the surrounding matrix supports the fibers. The properties of the fibers and the matrix greatly affect the onset and magnitude of fiber buckling and result in losses in the compressive properties of the laminate.

One of the main roles of the fibers is to carry tensile loads. When dry fibers (with no matrix surrounding them) break, they, of course, can no longer carry tensile loads. When the fibers are embedded in a matrix, the matrix acts as a bridge across the break and transmits the load across the gap created by the breakage as well as from the broken to the adjacent fibers. Fiber bridging, as this phenomenon is called, is the main reason that the tensile strengths of unidirectional, continuous fiber-reinforced composites are higher than the tensile strengths of dry fiber bundles.

Matrix cracking frequently occurs in composite laminates. In itself, matrix cracking generally does not result in the ultimate failure of a laminate. Nonetheless, matrix cracks have many detrimental effects: they facilitate moisture absorption, reduce the matrix-dominated stiffnesses of the laminate and, last but not least, may propagate into the interface between adjacent layers, initiating delamination.

Delamination is a separation of adjacent layers that may be introduced either during manufacture or subsequently by loads applied to the laminate. For example, loads due to transverse impact by an object on the laminate are a frequent cause of delamination. Delamination reduces the bending stiffness and strength as well as the load carrying capability of the laminate under compression. Significantly, under repeated loading the size of the delamination may increase to a critical point. Like the behavior of a crack in metal, once the critical size is reached, the growth of the delamination becomes unstable, leading to a rapid loss of compressive strength.

Procedures of Classical Lamination Theory Analysis

In conclusion, Classical Lamination Theory consists of a comprehensive set of deformation hypotheses leading to the force-strain-curvature and moment-strain-curvature relations of equation (55), where the physical significances of A_{ij}, B_{ij}, and D_{ij} have already been labeled. The procedures of Classical Lamination Theory are summarized in Table 2.

TABLE 2. Procedures of Classical Laminate Theory Analysis

Step	Statements	Procedures of each step	Notes
1	$E_1, E_2, G_{12}, v_{12}, t_{\text{ply}}$ $F_{1t}, F_{1c}, F_{2t}, F_{2c}, F_6$ $\theta_k \ (k=1, 2, ..., n)$ $\{N\}, \{M\}$	→ σ, ε analysis → failure analysis → stacking sequence → given or calculate	failure analysis use: Tsai-Hill, Tsai-Wu Max σ, Max ε Safety factor, FPF
2	calculate $[Q]$	use E_1, E_2, G_{12}, v_{12}	
3	calculate $[\overline{Q}]^k$	use $[Q]$, and θ_k	
4	calculate $[A], [B], [D]$	use $[\overline{Q}]^k, t_{\text{ply}}$, and θ_k	$k=1, 2, \ldots, n$
5	calculate $[a], [d]$	$[a]=[A]^{-1}, \quad [d]=[D]^{-1}$	inverse matrix
6	solve $\{\varepsilon^\circ\}, \{\kappa\}$	$\begin{Bmatrix} \varepsilon^\circ \\ \kappa \end{Bmatrix} = \begin{bmatrix} a & b \\ b^T & d \end{bmatrix} \begin{Bmatrix} N \\ M \end{Bmatrix}$	if $[B]=0 \Rightarrow \begin{matrix} \{\varepsilon^\circ\}=[a]\{N\} \\ \{\kappa\}=[d]\{M\} \end{matrix}$
7	calculate $\{\varepsilon\}_g^k$	$\{\varepsilon\}_g^k = \{\varepsilon^\circ\} + z\{\kappa\}$	$\{M\}=0 \ \Rightarrow \{\varepsilon\}_g^k = \{\varepsilon^\circ\}$ $\{M\} \neq 0 \ \Rightarrow \{\varepsilon\}_g^k = \{\varepsilon^\circ\} + z\{\kappa\}$
8	calculate 2 stresses for each layer	$\{\sigma\}_g^k = [\overline{Q}]^k \{\varepsilon\}_g^k$	$\theta_{k-1} \neq \theta_k \ \Rightarrow [\overline{Q}]^{k-1} \neq [\overline{Q}]^k$
9	calculate 2 stresses for each layer	$\{\sigma\}_m^k = [T_\sigma(\theta_k)]\{\sigma\}_g^k$	$\{\sigma\}_g^{k-1} \neq \{\sigma\}_g^k$
10	calculate 2 strains	$\{\varepsilon\}_m^k = [S]\{\sigma\}_m^k$	
11	failure analysis	safety factor of each fly	

Note: z can be at mid-layer, or at interface between two layers. When $\{M\}=0$, strain is a constant through the laminate thickness. When $\{M\} \neq 0$, strain is a linear function of z.

Classical Lamination Theory allows for the calculation of forces and moments if the strains and curvatures of the middle surface (or vice versa) are known (step-6). This enables the calculation of laminae stresses in the laminate coordinates (step-8). Next, one can transform the laminae stresses from the laminate coordinates to the lamina principal material directions (step-9). Finally, one would expect to apply a failure criterion to each lamina in its own principal material directions (step-10 and 11). This process seems straightforward in principle, but the force-strain-curvature and moment-strain-curvature relations are difficult to completely understand.

Common Laminate Types

The constitutive equations of the most general type of laminate are given by (55) with the coefficients computed according to (52). There are 18 different coefficients in the $[A\,B\,D]$matrix and if all coefficients are different from zero, the laminate response is fully coupled, meaning that the application of just one load (say, N_x) makes all six strains different from zero. A desire to eliminate some extent of the coupling terms has motivated the use of particular types of laminates, which are described in this section.

Laminate Description

The notation used to describe laminates has its roots in the description used to specify the lay-up sequence for hand lay-up using prepreg. When using prepreg, all layers have the same thickness and only the angles need to be specified. In the hand lay-up process, the laminate is built starting at the tool surface or bottom of the laminate, adding subsequent layers on top. Therefore, the layers are numbered starting at the bottom and the angles are given from bottom up.

If the laminate is symmetric, like$[30/0/0/30]$ and $[0/90/0]$, an abbreviated notation is used where only half of the stacking sequence is given and a subscript (S) is added to specify symmetry. The last examples become $[30/0]_S$ and $[0/\overline{90}]_S$. A subscript (T) may be used to indicate that the sequence given is the total (e.g., $[45/-45/45/-45]_T$) or $[\pm 45/\pm 45]_T$. A subscript may be used to indicate a repeating pattern. In such a way the last example becomes $[(\pm 45)_2]_T$.

If the thicknesses of the layers are different, they are specified for each layer. For example $[\theta_{t1}/-\theta_{t2}]$, if all layers have the same thickness, the laminate is called regular.

Balanced Laminates

In a balanced laminate, for every layer at $+\theta$ there is another at $-\theta$ with the same thickness and material; and for each 0°-layer there is a complementary 90°-layer, also of the same thickness and material. For each balanced pair of layers k and k',

$$t_k = t_{k'}, \qquad \theta_k = -\theta_{k'} \tag{63}$$

By virtue of (34),

$$\begin{aligned} &\text{Even function}: \ \overline{Q}_{ij}^k(\theta_k) = \overline{Q}_{ij}^k(-\theta_k) \qquad (ij = 11,\ 22,\ 12,\ 66) \\ &\text{Odd function}: \ \overline{Q}_{i6}(\theta) = -\overline{Q}_{i6}(-\theta) \qquad (i = 1,\ 2) \end{aligned} \tag{64}$$

Then,

$$A_{i6} = \sum_{k=1}^{n} \overline{Q}_{i6}^k t_k = 0 \qquad (i = 1, 2) \tag{65}$$

The general load-deformation relations for this class of laminates are:

$$\begin{Bmatrix} N_x \\ N_y \\ N_{xy} \\ \hline M_x \\ M_y \\ M_{xy} \end{Bmatrix} = \left[\begin{array}{ccc:ccc} A_{11} & A_{12} & 0 & B_{11} & B_{12} & B_{16} \\ A_{21} & A_{22} & 0 & B_{21} & B_{22} & B_{26} \\ 0 & 0 & A_{66} & B_{61} & B_{62} & B_{66} \\ \hdashline B_{11} & B_{12} & B_{16} & D_{11} & D_{12} & D_{16} \\ B_{21} & B_{22} & B_{26} & D_{21} & D_{22} & D_{26} \\ B_{61} & B_{62} & B_{66} & D_{61} & D_{62} & D_{66} \end{array}\right] \begin{Bmatrix} \varepsilon_x^\circ \\ \varepsilon_y^\circ \\ \gamma_{xy}^\circ \\ \hline \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{Bmatrix} \tag{66}$$

Thus, a balanced laminate always has $A_{16} = A_{26} = 0$.

A balanced laminate can be symmetric, anti-symmetric, or asymmetric. For example, a laminate consisting of pairs of θ_1, $-\theta_1$ and θ_2, $-\theta_2$ plies can be arranged in the following layups:

$$\begin{array}{l:l:l:l}
\text{Symmetric} & [\pm\theta_1/\pm\theta_2]_S & A_{16} = A_{26} = 0 & [B] = 0 \\
\text{Anti-symmetric} & [\theta_1/\theta_2/-\theta_2/-\theta_1] & A_{16} = A_{26} = D_{16} = D_{26} = 0 & [B] \neq 0 \\
\text{Asymmetric} & [\theta_1/\theta_2/-\theta_1/-\theta_2] & A_{16} = A_{26} = 0 & [B] \neq 0
\end{array} \qquad (67)$$

Symmetric Laminates

A laminate is symmetric if layers of the same material, thickness, and orientation are symmetrically located with respect to the middle surface of the laminate. For example, $[30/0/0/30]$ is symmetric.

Symmetric laminates experience no bending-extension coupling $B_{ij} = 0$. This is very important during fabrication because curing and subsequent cooling of the composite induces thermal forces N_x and N_y. If the laminate is not symmetric, these forces will induce warping (κ_x, κ_y) of the final part. Since residual thermal forces due to curing are unavoidable, only a symmetric laminate will remain flat after cool-down.

Consider the identical layers k and k' are symmetrically situated about the reference plane (figure 4). Then

$$t_k = t_{k'}, \qquad \bar{z}_k = -\bar{z}_{k'}, \qquad \overline{Q}_{ij}^{k} = \overline{Q}_{ij}^{k'} \qquad (i,\, j = 1, 2, 6) \qquad (68)$$

and according to the definition of equation (52), the bending-extension coupling coefficients are

$$B_{ij} = \frac{1}{2}\sum_{k=1}^{n} \overline{Q}_{ij}^{k}\left(z_k^2 - z_{k-1}^2\right) = \sum_{k=1}^{n} \overline{Q}_{ij}^{k}\, t_k \bar{z}_k = 0 \qquad (i,\, j = 1, 2, 6) \qquad (69)$$

With reference to Figure 4, note that for a symmetric laminate there is a negative $\bar{z}_k$ for each positive $\bar{z}_k$ (due to symmetry) corresponding to the same material properties and same thickness. Therefore, pairs of terms in the equation above corresponding to symmetrically located layers cancel each other, resulting in $B_{ij} = 0$. The general load-deformation relations for symmetric laminates are

$$\begin{Bmatrix} N_x \\ N_y \\ N_{xy} \\ \hline M_x \\ M_y \\ M_{xy} \end{Bmatrix} = \left[\begin{array}{ccc|ccc} A_{11} & A_{12} & A_{16} & 0 & 0 & 0 \\ A_{12} & A_{22} & A_{26} & 0 & 0 & 0 \\ A_{16} & A_{26} & A_{66} & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & D_{11} & D_{12} & D_{16} \\ 0 & 0 & 0 & D_{12} & D_{22} & D_{26} \\ 0 & 0 & 0 & D_{16} & D_{26} & D_{66} \end{array}\right] \begin{Bmatrix} \varepsilon_x^\circ \\ \varepsilon_y^\circ \\ \gamma_{xy}^\circ \\ \hline \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{Bmatrix} \tag{70}$$

and the load-deformation relations are reduced to

$$\begin{Bmatrix} N_x \\ N_y \\ N_{xy} \end{Bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{16} \\ A_{12} & A_{22} & A_{26} \\ A_{16} & A_{26} & A_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_x^\circ \\ \varepsilon_y^\circ \\ \gamma_{xy}^\circ \end{Bmatrix}, \qquad \begin{Bmatrix} M_x \\ M_y \\ M_{xy} \end{Bmatrix} = \begin{bmatrix} D_{11} & D_{12} & D_{16} \\ D_{12} & D_{22} & D_{26} \\ D_{16} & D_{26} & D_{66} \end{bmatrix} \begin{Bmatrix} \kappa_x \\ \kappa_Y \\ \kappa_{xy} \end{Bmatrix} \tag{71}$$

The fact that $[B]$ matrix is zero uncouples all bending terms from the extension terms in (55). This means that a symmetric laminate subjected to bending moments will not stretch or shear. Also, a symmetric laminate will not bend or twist if subject to in-plane forces. Symmetric laminates are used whenever it is possible because the analysis is simplified and they do not behave in extraneous ways. For example, unsymmetrical laminates bend and warp as a result of thermal expansion. In a symmetric laminate, the symmetric layers expand the same amount, and the laminate will expand but not warp. The first three equations of (55) can be solved independently of the last three. If the compliances (58) are desired, the 3x3 $[A]$ matrix can be inverted to get the $[a]$ matrix and the 3x3 $[D]$ matrix can be inverted to get the $[d]$ matrix without having to invert the whole 6x6 $[A\ B\ D]$ matrix. Some special types of symmetric laminates are discussed below.

Symmetric Laminates Under In-Plane Loading

Symmetric laminates are most popular in applications mainly because they are free from warping induced by thermal residual stresses resulting from curing at elevated temperatures. Laminates provide excellent stiffness and strength properties for in-plane loading. They can be used with great structural efficiency in skins and stiffeners in aircraft structures.

In other words, we assume that the laminate deforms uniformly over the thickness if it is under in-plane loading. Although the strains are uniform and continuous over the thickness of the laminate, the stresses in the laminate are, in general, discontinuous across the interfaces due to different material properties resulting from different fiber orientations. If the laminate is thin, then we can assume with good accuracy that the strain components ε_x, ε_y, and γ_{xy} in all the laminae are the same over the thickness of the laminate. Under in-plane loading,

$$\overline{Q}_{16} = \overline{Q}_{26} = 0 \qquad \Rightarrow \quad A_{16} = A_{26} = 0 \tag{72}$$

$$\{M\} = 0,\ \{\kappa\} = 0 \ \Rightarrow \ \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix}^k = \begin{Bmatrix} \varepsilon_x^\circ \\ \varepsilon_y^\circ \\ \gamma_{xy}^\circ \end{Bmatrix} = \text{constant} \tag{73}$$

The overall load-deformation relations for this class of laminates are:

$$\begin{Bmatrix} N_x \\ N_y \\ N_{xy} \\ \hdashline M_x \\ M_y \\ M_{xy} \end{Bmatrix} = \left[\begin{array}{ccc:ccc} A_{11} & A_{12} & 0 & 0 & 0 & 0 \\ A_{12} & A_{22} & 0 & 0 & 0 & 0 \\ 0 & 0 & A_{66} & 0 & 0 & 0 \\ \hdashline 0 & 0 & 0 & D_{11} & D_{12} & D_{16} \\ 0 & 0 & 0 & D_{12} & D_{22} & D_{26} \\ 0 & 0 & 0 & D_{16} & D_{26} & D_{66} \end{array}\right] \begin{Bmatrix} \varepsilon_x^\circ \\ \varepsilon_y^\circ \\ \gamma_{xy}^\circ \\ \hdashline \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{Bmatrix} \tag{74}$$

Therefore the load-deformation relations are reduced to:

$$\begin{Bmatrix} N_x \\ N_y \end{Bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{12} & A_{22} \end{bmatrix} \begin{Bmatrix} \varepsilon_x^\circ \\ \varepsilon_y^\circ \end{Bmatrix}, \qquad \begin{Bmatrix} M_x \\ M_y \\ M_{xy} \end{Bmatrix} = \begin{bmatrix} D_{11} & D_{12} & D_{16} \\ D_{12} & D_{22} & D_{26} \\ D_{16} & D_{26} & D_{66} \end{bmatrix} \begin{Bmatrix} \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{Bmatrix} \tag{75}$$

$$N_{xy} = A_{66}\gamma_{xy}^\circ$$

Symmetric and Balanced Laminates

On a macroscopic scale, this type of laminate can be treated as a homogeneous orthotropic material. For example, $[45/-45/0/-45/45]=[\pm 45/\overline{0}]_S$ is symmetric and balanced laminates. By virtue of (34),

$$\overline{Q}_{i6} = 0 \quad \Rightarrow \quad A_{i6} = 0 \qquad (i=1,\,2) \tag{76}$$

Thus the defining characteristics of symmetric and balanced laminates are

$$A_{16} = A_{26} = 0, \quad B_{ij} = 0 \quad (i,\, j = 1,\, 2,\, 6) \tag{77}$$

The general load-deformation relations for symmetric and balanced laminates are as follows:

$$\begin{Bmatrix} N_x \\ N_y \\ N_{xy} \\ \hdashline M_x \\ M_y \\ M_{xy} \end{Bmatrix} = \left[\begin{array}{ccc:ccc} A_{11} & A_{12} & 0 & 0 & 0 & 0 \\ A_{12} & A_{22} & 0 & 0 & 0 & 0 \\ 0 & 0 & A_{66} & 0 & 0 & 0 \\ \hdashline 0 & 0 & 0 & D_{11} & D_{12} & D_{16} \\ 0 & 0 & 0 & D_{12} & D_{22} & D_{26} \\ 0 & 0 & 0 & D_{16} & D_{26} & D_{66} \end{array}\right] \begin{Bmatrix} \varepsilon_x^\circ \\ \varepsilon_y^\circ \\ \gamma_{xy}^\circ \\ \hdashline \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{Bmatrix} \tag{78}$$

and the load-deformation relations are reduced to

$$\begin{Bmatrix} N_x \\ N_y \end{Bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{12} & A_{22} \end{bmatrix} \begin{Bmatrix} \varepsilon_x^{\circ} \\ \varepsilon_y^{\circ} \end{Bmatrix}, \qquad \begin{Bmatrix} M_x \\ M_y \\ M_{xy} \end{Bmatrix} = \begin{bmatrix} D_{11} & D_{12} & D_{16} \\ D_{12} & D_{22} & D_{26} \\ D_{16} & D_{26} & D_{66} \end{bmatrix} \begin{Bmatrix} \kappa_x \\ \kappa_Y \\ \kappa_{xy} \end{Bmatrix} \tag{79}$$

$$N_{xy} = A_{66} \gamma_{xy}^{\circ}$$

The in-plane strains can be obtain as:

$$\begin{Bmatrix} \varepsilon_x^{\circ} \\ \varepsilon_y^{\circ} \end{Bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{12} & A_{22} \end{bmatrix}^{-1} \begin{Bmatrix} N_x \\ N_y \end{Bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{12} & a_{22} \end{bmatrix} \begin{Bmatrix} N_x \\ N_y \end{Bmatrix} = \begin{bmatrix} \dfrac{1}{E_1} & \dfrac{-\nu_{21}}{E_2} \\ \dfrac{-\nu_{12}}{E_1} & \dfrac{1}{E_2} \end{bmatrix} \begin{Bmatrix} N_x \\ N_y \end{Bmatrix} \tag{80}$$

$$\gamma_{xy}^{\circ} = \frac{N_{xy}}{A_{66}} = a_{66} N_{xy} = \frac{1}{G_{12}} N_{xy}$$

Symmetric Cross-Ply Laminates

A cross-ply laminate has only layers oriented at 0 and 90 degrees. If the construction is symmetric, it is called a symmetric cross-ply laminate. The main advantage of a cross-ply laminate is simplicity of construction, plywood being the typical example. Since all layers in the laminate have $\overline{Q}_{16}$ and $\overline{Q}_{26}$ equal to zero, the 16 and 26 entries in $[A]$ and $[D]$ are also zero. Therefore, cross-ply laminates do not have coupling between shear and extension in the $[A]$ matrix, nor do they have coupling between twisting and bending in the $[D]$ matrix. This is not a really great advantage from an analysis or a performance point of view, unless the laminate is also symmetric. For example, $[0/90/90/0] = [0/90]_S$ and $[0/90/0] = [0/\overline{90}]_S$ are symmetric cross-ply laminates with even and odd layers, respectively. By virtue of specially orthotropic layers,

$$\overline{Q}_{i6} = 0 \quad \Rightarrow \quad A_{i6} = B_{i6} = D_{i6} = 0 \quad (i = 1,\ 2) \tag{81}$$

Thus, the characteristics of symmetric cross-ply laminates are defined as:

$$A_{16} = A_{26} = 0, \quad D_{16} = D_{26} = 0, \quad [B] = 0 \tag{82}$$

$$\begin{Bmatrix} N_x \\ N_y \\ N_{xy} \\ \hline M_x \\ M_y \\ M_{xy} \end{Bmatrix} = \left[\begin{array}{ccc|ccc} A_{11} & A_{12} & 0 & 0 & 0 & 0 \\ A_{12} & A_{22} & 0 & 0 & 0 & 0 \\ 0 & 0 & A_{66} & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & D_{11} & D_{12} & 0 \\ 0 & 0 & 0 & D_{12} & D_{22} & 0 \\ 0 & 0 & 0 & 0 & 0 & D_{66} \end{array}\right] \begin{Bmatrix} \varepsilon_x^\circ \\ \varepsilon_y^\circ \\ \gamma_{xy}^\circ \\ \hline \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{Bmatrix} \tag{83}$$

The load-deformation relations for symmetric cross-ply laminates are shown in equation (83). Thus the load-deformation relations are reduced to

$$\begin{gathered} \begin{Bmatrix} N_x \\ N_y \end{Bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{12} & A_{22} \end{bmatrix} \begin{Bmatrix} \varepsilon_x^\circ \\ \varepsilon_y^\circ \end{Bmatrix}, \qquad \begin{Bmatrix} M_x \\ M_y \end{Bmatrix} = \begin{bmatrix} D_{11} & D_{12} \\ D_{12} & D_{22} \end{bmatrix} \begin{Bmatrix} \kappa_x \\ \kappa_y \end{Bmatrix} \\ N_{xy} = A_{66}\gamma_{xy}^\circ \qquad\qquad M_{xy} = D_{66}\kappa_{xy} \end{gathered} \tag{84}$$

For a specially symmetric cross-ply laminate that has alternating orientations of 0° and 90° plies with odd layers, with $\overline{\theta}\,(0^\circ \text{ or } 90^\circ)$ as the orientation of the middle layer, the magnitude of the 12 and 66 entries in $[A]$ and $[D]$ can be described as:

$$\begin{array}{l|l|l} A_{16} = A_{26} = 0 & D_{16} = D_{26} = 0 & [B] = 0 \\ A_{12} = Q_{12}(\overline{\theta})h & D_{12} = Q_{12}(\overline{\theta})(h^3/12) & \\ A_{66} = Q_{66}(\overline{\theta})h & D_{66} = Q_{66}(\overline{\theta})(h^3/12) & \end{array} \tag{85}$$

Symmetric Angle-Ply Laminates

Angle-ply laminates are built with layers of the same material and thickness, oriented at $+\theta$ and $-\theta$ directions. They can be symmetric or asymmetric. An angle-ply laminate with an even number of plies is also balanced, for example, $[30/-30/-30/30]$ is a symmetric angle-ply laminate and $[30/-30/30/-30]$ is an anti-symmetric angle-ply laminate. Then, the in-plane shear coupling terms are

$$A_{i6} = \sum_{k=1}^{n} \overline{Q}_{i6}^{k}\, t = \sum_{k=1}^{n/2} \overline{Q}_{i6}^{k}(\theta)t + \sum_{k=1}^{n/2} \overline{Q}_{i6}^{k}(-\theta)t = 0 \qquad (i = 1, 2) \tag{86}$$

since $\overline{Q}_{i6}(\theta) = -\overline{Q}_{i6}(-\theta)$ from equation (34). Thus, a symmetric angle-ply laminate with an even number of layers has

$$A_{16} = A_{26} = 0, \quad [D] \neq 0, \quad [B] = 0 \tag{87}$$

The load-deformation relations for this class of laminates are

$$\begin{Bmatrix} N_x \\ N_y \\ N_{xy} \\ \hline M_x \\ M_y \\ M_{xy} \end{Bmatrix} = \left[\begin{array}{ccc|ccc} A_{11} & A_{12} & 0 & 0 & 0 & 0 \\ A_{12} & A_{22} & 0 & 0 & 0 & 0 \\ 0 & 0 & A_{66} & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & D_{11} & D_{12} & D_{16} \\ 0 & 0 & 0 & D_{12} & D_{22} & D_{26} \\ 0 & 0 & 0 & D_{16} & D_{26} & D_{66} \end{array}\right] \begin{Bmatrix} \varepsilon_x^\circ \\ \varepsilon_y^\circ \\ \gamma_{xy}^\circ \\ \hline \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{Bmatrix} \tag{88}$$

Then the load-deformation relations are reduced to

$$\begin{Bmatrix} N_x \\ N_y \end{Bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{12} & A_{22} \end{bmatrix} \begin{Bmatrix} \varepsilon_x^\circ \\ \varepsilon_y^\circ \end{Bmatrix}, \qquad \begin{Bmatrix} M_x \\ M_y \\ M_{xy} \end{Bmatrix} = \begin{bmatrix} D_{11} & D_{12} & D_{16} \\ D_{12} & D_{22} & D_{26} \\ D_{16} & D_{26} & D_{66} \end{bmatrix} \begin{Bmatrix} \kappa_x \\ \kappa_Y \\ \kappa_{xy} \end{Bmatrix}$$
$$N_{xy} = A_{66}\gamma_{xy}^\circ \tag{89}$$

If a specially angle-ply laminate consists of an odd number of alternating $+\theta$ and $-\theta$ plies of equal thickness, then it is also symmetric, for example, $[\theta/-\theta/\theta/-\theta/\theta] = [\pm\theta/\overline{\theta}]_S$. For a ply thickness t, laminate thickness $h = nt$, and orientation of middle layer $\overline{\theta}$, then the in-plane shear coupling terms are:

$$A_{i6} = \sum_{k=1}^{n} \overline{Q}_{i6}^k t = \sum_{k=1}^{(n-1)/2} \overline{Q}_{i6}^k(\theta)t + \sum_{k=1}^{(n-1)/2} \overline{Q}_{i6}^k(-\theta)t + \overline{Q}_{i6}(\overline{\theta})t = \overline{Q}_{i6}(\overline{\theta})\frac{h}{n} \qquad (i = 1, 2) \tag{90}$$

Thus, a specially symmetric angle-ply laminate with an odd number of layers has

$$[A] \neq 0, \quad [D] \neq 0, \quad [B] = 0 \tag{91}$$

Even though the full coupling terms of $[A]$ and $[D]$ matrices are not zero, but their magnitude can be shown as

$$\begin{array}{l|l|l} (ij=11,\,22,\,12,\,66) & A_{ij}=\overline{Q}_{ij}(\theta)h & D_{ij}=\overline{Q}_{ij}(\theta)\dfrac{h^3}{12} \\ (i=1,\,2) & A_{i6}=\overline{Q}_{i6}(\theta)\dfrac{h}{n} & D_{i6}=\overline{Q}_{i6}(\theta)\dfrac{h^3}{12}\left(\dfrac{3n^2-2}{n^3}\right) \end{array} \qquad (92)$$

Anti-Symmetric Laminates

An anti-symmetric laminate is a special case of a balanced laminate, has pairs of layers of opposite orientation but the same material and thickness symmetrically located with respect to the middle surface of the laminate. A laminate has an even number of plies. For example, $[30/-30/30/-30]$ and $[0/90/0/90]$ are anti-symmetric laminates.

For the symmetrically situated balanced pair of k and k' (or θ and $-\theta$ layers),

$$\begin{gathered} t_k=t_{k'} \\ z_k^3-z_{k-1}^3=z_{k'}^3-z_{k'-1}^3 \\ \overline{Q}_{i6}(\theta)=-\overline{Q}_{i6}(-\theta) \qquad (i=1,\,2) \end{gathered} \qquad (93)$$

the in-plane shear coupling stiffnesses and bending/twisting coupling stiffnesses are

$$\begin{aligned} A_{i6}&=\sum_{k=1}^{n}\overline{Q}_{i6}^{k}t_k=0 && (i=1,\,2) \\ D_{i6}&=\frac{1}{3}\sum_{k=1}^{n}\overline{Q}_{i6}^{k}(\theta_k)\left(z_k^3-z_{k-1}^3\right)=0 && (i=1,\,2) \end{aligned} \qquad (94)$$

Thus, anti-symmetric laminates have $A_{16}=A_{26}=D_{16}=D_{26}=0$, but they are not particularly useful nor are they easier to analyze than general laminates because the bending extension coefficients B_{16} and B_{26} are not always zero for these laminates.

The purpose of the remainder of this section is to discuss two important classes of anti-symmetric laminates, the anti-symmetric cross-ply laminate and the anti-symmetric angle-ply laminate. Neither laminate is used much in practice, but both add to our understanding of laminates.

Anti-Symmetric Cross-Ply Laminates

Anti-symmetric cross-ply laminates consist of pairs of 0° and 90° plies symmetrically situated about the middle surface with identical thickness and material properties. A laminate has an even number of plies. For example, $\left[0/90_2/0_2/90\right]_T$ is an anti-symmetric cross-ply laminate.

By virtue of specially orthotropic layers,

$$\overline{Q}_{i6}=0 \;\;\Rightarrow\;\; A_{i6}=B_{i6}=D_{i6}=0 \quad (i=1,2) \tag{95}$$

Thus, an anti-symmetric cross-ply laminate with an even number of layers has

$$\begin{array}{l|l|l} A_{16}=A_{26}=0 & D_{16}=D_{26}=0 & B_{16}=B_{26}=0 \\ A_{11}\approx A_{22} & D_{11}\approx D_{22} & B_{11}=-B_{22} \\ A_{12},\ A_{66}\neq 0 & D_{12},\ D_{66}\neq 0 & B_{12}=B_{66}=0 \end{array} \tag{96}$$

The overall load-deformation relations for this class of laminates are

$$\begin{Bmatrix} N_x \\ N_y \\ N_{xy} \\ \hline M_x \\ M_y \\ M_{xy} \end{Bmatrix} = \left[\begin{array}{ccc|ccc} A_{11} & A_{12} & 0 & B_{11} & 0 & 0 \\ A_{12} & A_{11} & 0 & 0 & -B_{11} & 0 \\ 0 & 0 & A_{66} & 0 & 0 & 0 \\ \hline B_{11} & 0 & 0 & D_{11} & D_{12} & 0 \\ 0 & -B_{11} & 0 & D_{12} & D_{11} & 0 \\ 0 & 0 & 0 & 0 & 0 & D_{66} \end{array}\right] \begin{Bmatrix} \varepsilon_x^{\circ} \\ \varepsilon_y^{\circ} \\ \gamma_{xy}^{\circ} \\ \hline \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{Bmatrix} \tag{97}$$

The load-deformation relations are then reduced to

$$\begin{Bmatrix} N_x \\ N_y \\ \hline M_x \\ M_y \end{Bmatrix} = \left[\begin{array}{cc|cc} A_{11} & A_{12} & B_{11} & 0 \\ A_{12} & A_{11} & 0 & -B_{11} \\ \hline B_{11} & 0 & D_{11} & D_{12} \\ 0 & -B_{11} & D_{12} & D_{11} \end{array}\right] \begin{Bmatrix} \varepsilon_x^{\circ} \\ \varepsilon_y^{\circ} \\ \hline \kappa_x \\ \kappa_y \end{Bmatrix}, \qquad \begin{Bmatrix} N_{xy} \\ M_{xy} \end{Bmatrix} = \begin{bmatrix} A_{66} & 0 \\ 0 & D_{66} \end{bmatrix} \begin{Bmatrix} \gamma_{xy}^{\circ} \\ \kappa_{xy} \end{Bmatrix} \tag{98}$$

A regular anti-symmetric cross-ply laminate is defined to have all laminae of equal thickness and is commonly due to its simplicity of fabrication. As the number of layers in the laminate increases for a fixed laminate thickness, the bending-extension coupling stiffness B_{11} can be shown to approach zero.

Using (58), it can be shown that the laminate bends when loaded with only an inplane force N_x. That is, a curvature $\kappa_x = b_{11} N_x$ is induced in addition to stretching $\varepsilon_x^\circ = a_{11} N_x$. The anti-symmetric cross-ply laminate is subject to in-plane forces N_x and N_y caused by thermal contraction during cool-down after curing. It can be shown that b_{11} and b_{22} (as well as B_{11} and B_{22}) have opposite signs, resulting in the saddle shape deformation.

For specially anti-symmetric cross-ply laminates with an even number of alternating 0° and 90° plies, the magnitude of the 12 and 66 entries in $[A]$ and $[D]$ can be shown as

$$\begin{array}{l|l|l}
A_{16} = A_{26} = 0 & D_{16} = D_{26} = 0 & B_{16} = B_{26} = 0 \\
A_{11} \approx A_{22} & D_{11} \approx D_{22} & B_{11} = -B_{22} \\
A_{12} = Q_{12} h & D_{12} = Q_{12}\left(h^3/12\right) & B_{12} = 0 \\
A_{66} = Q_{66} h & D_{66} = Q_{66}\left(h^3/12\right) & B_{66} = 0
\end{array} \qquad (99)$$

Anti-Symmetric Angle-Ply Laminates

Anti-symmetric angle-ply laminates consist of pairs of $+\theta_i$ and $-\theta_i$ plies $\left(0^\circ < \theta_i < 90^\circ\right)$, symmetrically situated about the middle surface and having the same thickness and elastic properties.

An anti-symmetric angle-ply laminate has $A_{16} = A_{26} = D_{16} = D_{26} = 0$, but $B_{16} \neq 0$ and $B_{26} \neq 0$. It can be shown that for this laminate, the only zero terms are

$$\begin{array}{l|l|l}
A_{16} = A_{26} = 0 & D_{16} = D_{26} = 0 & B_{11} = B_{22} = B_{12} = B_{66} = 0 \\
a_{16} = a_{26} = 0 & d_{16} = d_{26} = 0 & b_{11} = b_{22} = b_{12} = b_{66} = 0
\end{array} \qquad (100)$$

The load-deformation relations for an anti-symmetric angle-ply laminate are

$$\begin{Bmatrix} N_x \\ N_y \\ N_{xy} \\ \hline M_x \\ M_y \\ M_{xy} \end{Bmatrix} = \left[\begin{array}{ccc:ccc} A_{11} & A_{12} & 0 & 0 & 0 & B_{16} \\ A_{12} & A_{22} & 0 & 0 & 0 & B_{26} \\ 0 & 0 & A_{66} & B_{16} & B_{26} & 0 \\ \hdashline 0 & 0 & B_{16} & D_{11} & D_{12} & 0 \\ 0 & 0 & B_{26} & D_{12} & D_{22} & 0 \\ B_{16} & B_{26} & 0 & 0 & 0 & D_{66} \end{array}\right] \begin{Bmatrix} \varepsilon_x^{\circ} \\ \varepsilon_y^{\circ} \\ \gamma_{xy}^{\circ} \\ \hline \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{Bmatrix} \tag{101}$$

The load-deformation relations are then reduced to

$$\begin{Bmatrix} N_x \\ N_y \\ M_{xy} \end{Bmatrix} = \begin{bmatrix} A_{11} & A_{12} & B_{16} \\ A_{12} & A_{22} & B_{26} \\ B_{16} & B_{26} & D_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_x^{\circ} \\ \varepsilon_y^{\circ} \\ \kappa_{xy} \end{Bmatrix}, \quad \begin{Bmatrix} M_x \\ M_y \\ N_{xy} \end{Bmatrix} = \begin{bmatrix} D_{11} & D_{12} & B_{16} \\ D_{12} & D_{22} & B_{26} \\ B_{16} & B_{26} & A_{66} \end{bmatrix} \begin{Bmatrix} \kappa_x \\ \kappa_Y \\ \gamma_{xy}^{\circ} \end{Bmatrix} \tag{102}$$

The in-plane strains and curvatures can be obtain as

$$\begin{Bmatrix} \varepsilon_x^{\circ} \\ \varepsilon_y^{\circ} \\ \kappa_{xy} \end{Bmatrix} = \begin{bmatrix} A_{11} & A_{12} & B_{16} \\ A_{12} & A_{22} & B_{26} \\ B_{16} & B_{26} & D_{66} \end{bmatrix}^{-1} \begin{Bmatrix} N_x \\ N_y \\ M_{xy} \end{Bmatrix} = \begin{bmatrix} a_{11} & a_{12} & b_{16} \\ a_{12} & a_{22} & b_{26} \\ b_{16} & b_{26} & d_{66} \end{bmatrix} \begin{Bmatrix} N_x \\ N_y \\ M_{xy} \end{Bmatrix} \tag{103}$$

$$\begin{Bmatrix} \kappa_x \\ \kappa_Y \\ \gamma_{xy}^{\circ} \end{Bmatrix} = \begin{bmatrix} D_{11} & D_{12} & B_{16} \\ D_{12} & D_{22} & B_{26} \\ B_{16} & B_{26} & A_{66} \end{bmatrix}^{-1} \begin{Bmatrix} M_x \\ M_y \\ N_{xy} \end{Bmatrix} = \begin{bmatrix} d_{11} & d_{12} & b_{61} \\ d_{12} & d_{22} & b_{62} \\ b_{61} & b_{62} & a_{66} \end{bmatrix} \begin{Bmatrix} M_x \\ M_y \\ N_{xy} \end{Bmatrix} \tag{104}$$

Therefore, an inplane load N_x causes the laminate twist ($\kappa_{xy} = b_{16} N_x$). A regular anti-symmetric angle-ply laminate has laminae of all the same material and thickness for ease of fabrication. The bending-extension coupling stiffnesses B_{16} and B_{26} can be shown to approach zero as the number of layers in the laminate increases for a fixed laminate thickness.

A more special case of this class of laminates is the anti-symmetric regular angle-ply laminate, consisting of an even number of plies alternating between θ and $-\theta$ in orientation, that is, $[\theta/-\theta/\theta/-\theta/\theta/-\theta]$. However, the same coupling terms are not zero, but their magnitude can be shown as

$$\begin{array}{l|l|l|l} ij = 16,\, 26 & A_{ij} = 0 & D_{ij} = 0 & B_{ij} = -\overline{Q}_{ij}\left(h^2/2n\right) \\ ij = 11,\, 22,\, 12,\, 66 & A_{ij} = \overline{Q}_{ij} h & D_{ij} = \overline{Q}_{ij}\left(h^3/12\right) & B_{ij} = 0 \end{array} \qquad (105)$$

Specially Orthotropic Laminates

Each fiber-reinforced composite layer is orthotropic in material axes, resulting in $Q_{16} = Q_{26} = 0$. If the layer is oriented at 0 or 90 degrees, it is called specially orthotropic because the $\left[\overline{Q}\right]$ matrix has the same zero entries as the $\left[Q\right]$ matrix; that is, $\overline{Q}_{16} = \overline{Q}_{26} = 0$. A layer reinforced with woven or stitched bidirectional fabric is also specially orthotropic if the amount of fabric in both directions ($\pm\theta$) is the same.

A specially orthotropic laminate is constructed with specially orthotropic layers. It can be symmetric or not. For example, $\left[0/90\right]_S$ is a symmetric cross-ply laminate and $\left[0/90\right]$ is an anti-symmetric cross-ply laminate. Thus, specially orthotropic laminates are not limited to cross-ply types (only 0° and 90° layers), as they may include balanced $\pm\theta$ fabrics as well. The advantage of a specially orthotropic laminate is that it has the same zero terms for laminate stiffnesses and compliances as

$$\begin{array}{l|l|l} A_{16} = A_{26} = 0 & D_{16} = D_{26} = 0 & B_{16} = B_{26} = 0 \\ a_{16} = a_{26} = 0 & d_{16} = d_{26} = 0 & b_{16} = b_{26} = 0 \end{array} \qquad (106)$$

Orthotropic Laminates

A laminate is orthotropic when every ply is orthotropic and the orthotropic directions coincide with the x and y directions. Fiber-reinforced plies are orthotropic under the following conditions:

1. For unidirectional fibers, all fibers are aligned with one of the laminate's orthotropic directions

2. For woven fabric, ply's symmetry axes are aligned with the laminate's orthotropic directions

3. For adjacent unidirectional plies, the symmetry axes of this combined layer are aligned with the laminate's orthotropic directions.

On a macroscopic scale, a balanced and symmetric laminate about the principal laminate axes 1 and 2 can be treated as a homogeneous orthotropic material. This type of laminate is called an orthotropic laminate. By definition, the inplane/flexure coupling stiffnesses and the inplane shear coupling stiffnesses are zero, that is,

$$A_{16} = A_{26} = 0, \quad D_{16} = D_{26} = 0, \quad [B] = 0 \tag{107}$$

The overall load-deformation relations for the orthotropic laminates are

$$\left\{\begin{array}{c} N_1 \\ N_2 \\ N_{12} \\ \hline M_1 \\ M_2 \\ M_{12} \end{array}\right\} = \left[\begin{array}{ccc|ccc} A_{11} & A_{12} & 0 & 0 & 0 & 0 \\ A_{12} & A_{22} & 0 & 0 & 0 & 0 \\ 0 & 0 & A_{66} & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & D_{11} & D_{12} & 0 \\ 0 & 0 & 0 & D_{12} & D_{22} & 0 \\ 0 & 0 & 0 & 0 & 0 & D_{66} \end{array}\right] \left\{\begin{array}{c} \varepsilon_1^{\circ} \\ \varepsilon_2^{\circ} \\ \gamma_{12}^{\circ} \\ \hline \kappa_1 \\ \kappa_2 \\ \kappa_{12} \end{array}\right\} \tag{108}$$

Thus, the force-deformation relations referred to the $1-2$ system of coordinates are

$$\begin{Bmatrix} N_1 \\ N_2 \\ N_{12} \end{Bmatrix} = \begin{bmatrix} A_{11} & A_{12} & 0 \\ A_{12} & A_{22} & 0 \\ 0 & 0 & A_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_1^{\circ} \\ \varepsilon_2^{\circ} \\ \gamma_{12}^{\circ} \end{Bmatrix} \tag{109}$$

These relations, when referred to any arbitrary system, take the form

$$\begin{Bmatrix} N_x \\ N_y \\ N_{xy} \end{Bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{16} \\ A_{12} & A_{22} & A_{26} \\ A_{16} & A_{26} & A_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_x^{\circ} \\ \varepsilon_y^{\circ} \\ \gamma_{xy}^{\circ} \end{Bmatrix} \tag{110}$$

10-Percent Rule of Orthotropic Laminates

Plates are often made according to the 10-percent rule, and such plates behave similarly to orthotropic plates. Therefore, solutions for orthotropic plates provide good approximations of the deflections, maximum bending moments, buckling loads, and natural frequencies of non-orthotropic plates that have symmetrical lay-up and are constructed according to the 10-percent rule.

The 10-percent rule requires that the plate satisfy the following conditions:

1. The plate is made of unidirectional plies.
2. There are at least three ply orientations, $(n \geq 3)$.
3. The angles between the fibers are at least 15°, $(\Delta\theta \geq 15^\circ)$.
4. The number of plies in each fiber direction is at least 10 percent of the total number of plies $(\mathrm{n}_\theta \geq 0.1\ \mathrm{n}_{\text{total}})$.
5. Symmetric layup.

For example, $[45_6 / 90_3 / 0_{10}]_S \quad \Rightarrow \quad \frac{6}{19} \geq 1.9, \quad \frac{3}{19} \geq 1.9, \quad \frac{10}{19} \geq 1.9$

Plates conforming to the 10-percent rule have better load bearing capabilities than unidirectional or angle-ply laminates. The deflections, maximum bending moments, buckling loads, and natural frequencies of unsymmetrical plates can be approximated by replacing $[D]$ by $[D]^*$ in the expressions derived for symmetrical plates as shown below:

$$[D]^* = [D] - [B][A]^{-1}[B] \qquad (111)$$

Quasi-Isotropic Laminates

Quasi-isotropic laminates are constructed in an attempt to create a composite laminate that behaves like an isotropic plate. One such laminate has equal percentages of 0°, 45°, -45° and 90° layers placed symmetrically with respect to the laminate mid-plane, for example $[\pm 45/90/0]_s$. The in-plane behavior of quasi-isotropic laminates is similar to that of isotropic plates, but the bending behavior of quasi-isotropic laminates is quite different from the bending behavior of isotropic plates. Quasi-isotropic laminates were a convenient replacement for isotropic materials in weight critical applications. In a quasi-isotropic laminate, each layer has an orientation given by

$$\theta_k = k\theta_0, \quad \theta_0 = \frac{\pi}{n} \tag{112}$$

where k indicates the layer number, n is the number of layers (at least 3), and θ_0 is an initial angle. When writing the stacking sequence, any angle larger than 90 degrees is replaced by its complement (for $\theta_k > 90 \Rightarrow \theta_k = \theta_k - 180$). For example, $[60/120/180]$ is written as $[60/-60/0]$. The layers can be ordered in any order, like $[60/0/-60]$, and the laminae is still quasi-isotropic. Most quasi-isotropic laminates are balanced.

Quasi-isotropic laminates are generally not symmetric, but they can be made symmetric by doubling the number of layers in a mirror (symmetric) fashion. For example, the $[60/-60/0]$ can be made into a $[60/-60/0/0/-60/60]$, which is still quasi-isotropic. The advantage of symmetric quasi-isotropic laminates is that $[B] = 0$.

In quasi-isotropic laminates, the in-plane stiffness matrix $[A]$ behaves like that of an isotropic material. The reason for such a laminate being called quasi-isotropic and not isotropic is that the other stiffness matrixes $[B]$ and $[D]$ may not behave like an isotropic material. The in-plane stiffness matrix $[A]$ and the bending stiffness matrix $[D]$ of isotropic plates can be written in terms of the thickness (h) of the plate with only two material properties, E and ν, as:

$$[A]=h[Q]=\frac{Eh}{1-\nu^2}\begin{bmatrix}1 & \nu & 0\\ \nu & 1 & 0\\ 0 & 0 & (1-\nu)/2\end{bmatrix} \tag{113}$$

$$[D]=\frac{h^3}{12}[Q]=\frac{Eh^3}{12(1-\nu^2)}\begin{bmatrix}1 & \nu & 0\\ \nu & 1 & 0\\ 0 & 0 & (1-\nu)/2\end{bmatrix} \tag{114}$$

Note that for isotropic plates, $A_{11}=A_{22}=Eh/(1-\nu^2)$ and $A_{16}=A_{26}=0$. Also, $D_{11}=D_{22}=Eh^3/12(1-\nu^2)$ and $D_{16}=D_{26}=0$. On the other hand, a quasi-isotropic laminate has

$$[A]=\begin{bmatrix}A_{11} & A_{12} & 0\\ A_{12} & A_{11} & 0\\ 0 & 0 & A_{66}\end{bmatrix} \tag{115}$$

$$[D]=\begin{bmatrix}D_{11} & D_{12} & D_{16}\\ D_{12} & D_{22} & D_{26}\\ D_{16} & D_{26} & D_{66}\end{bmatrix} \tag{116}$$

Like isotropic plates, quasi-isotropic laminates have $A_{11}=A_{22}$, but the later have $D_{11}\neq D_{22}$, $D_{16}\neq 0$, and $D_{26}\neq 0$, which makes quasi-isotropic laminates quite different from isotropic plates, as shown below.

$$\text{Isotropic:}\qquad \begin{array}{|l|l|l} A_{16}=A_{26}=0 & D_{16}=D_{26}=0 & [B]=0\\ A_{11}=A_{22} & D_{11}=D_{22} & \\ A_{12}=\nu A_{11} & D_{12}=\nu D_{11} & \\ 2A_{66}=A_{11}-A_{22} & 2D_{66}=D_{11}-D_{22} & \end{array} \tag{117}$$

$$\text{Quasi-isotropic}: \left|\begin{array}{l} A_{16} = A_{26} = 0 \\ A_{11} = A_{22} \\ A_{12} = \nu A_{11} \\ 2A_{66} = A_{11} - A_{22} \end{array}\right| \begin{array}{l} D_{16} \neq 0, \quad D_{26} \neq 0 \\ D_{11} \neq D_{22} \\ \\ \\ \end{array} \tag{118}$$

Therefore, formulas for bending, buckling, and vibrations of isotropic plates can be used for quasi-isotropic laminates only as an approximation. However, a laminate can be designed trying to approach the characteristics of isotropic plates, with $D_{11} \approx D_{22}$ and D_{16} and D_{26} as small as possible. This can be accomplished by using a symmetric quasi-isotropic laminate with balanced $0/90$ and $\pm\theta$ layers and a large number of layers. In that case, the formulas for isotropic plates provide a reasonable approximation.

The conclusions discussed before regarding the defining characteristics for the various types of laminates are summarized in Tables 3 and 4.

TABLE 3. The Defining Characteristics for Various Types of Orthotropic and Isotropic Laminates

Orthotropic			
	$A_{16} = A_{26} = 0$	$D_{16} = D_{26} = 0$	$[B] = 0$
Specially Orthotropic			
	$A_{16} = A_{26} = 0$ $a_{16} = a_{26} = 0$	$D_{16} = D_{26} = 0$ $d_{16} = d_{26} = 0$	$B_{16} = B_{26} = 0$ $b_{16} = b_{26} = 0$
Isotropic			
	$A_{16} = A_{26} = 0$ $A_{11} = A_{22}$ $A_{12} = \nu A_{11}$ $2A_{66} = A_{11} - A_{22}$	$D_{16} = D_{26} = 0$ $D_{11} = D_{22}$ $D_{12} = \nu D_{11}$ $2D_{66} = D_{11} - D_{22}$	$[B] = 0$
Quasi-Isotropic			
	$A_{16} = A_{26} = 0$ $A_{11} = A_{22}$ $A_{12} = \nu A_{11}$ $2A_{66} = A_{11} - A_{22}$	$D_{16} \neq 0, \quad D_{26} \neq 0$ $D_{11} \neq D_{22}$	

TABLE 4. The Defining Characteristics for Various Types of Symmetric and Antisymmetric Laminates

Symmetric and Balanced			
	$A_{16} = A_{26} = 0$	$[D] \neq 0$	$[B] = 0$
Symmetric Cross-Ply			
	$A_{16} = A_{26} = 0$	$D_{16} = D_{26} = 0$	$[B] = 0$
Specially Symmetric Cross-Ply			
	$A_{16} = A_{26} = 0$ $A_{12} = Q_{12}(\overline{\theta})h$ $A_{66} = Q_{66}(\overline{\theta})h$	$D_{16} = D_{26} = 0$ $D_{12} = Q_{12}(\overline{\theta})(h^3/12)$ $D_{66} = Q_{66}(\overline{\theta})(h^3/12)$	$[B] = 0$
Symmetric Angle-Ply (n even)			
	$A_{16} = A_{26} = 0$	$[D] \neq 0$	$[B] = 0$
Specially Symmetric Angle-Ply (n odd)			
$i, j = 1, 2, 6$ $ij = 11, 22, 12, 66$ $i = 1, 2$	$A_{ij} \neq 0$ $A_{ij} = \overline{Q}_{ij}(\overline{\theta})h$ $A_{i6} = \overline{Q}_{i6}(\overline{\theta})\frac{h}{n}$	$D_{ij} \neq 0$ $D_{ij} = \overline{Q}_{ij}(\overline{\theta})(h^3/12)$ $D_{i6} = \overline{Q}_{i6}(\overline{\theta})\frac{h^3(3n^2-2)}{12n^3}$	$B_{ij} = 0$
Antisymmetric Cross-Ply (n even)			
	$A_{16} = A_{26} = 0$ $A_{11} \approx A_{22}$ $A_{12},\ A_{66} \neq 0$	$D_{16} = D_{26} = 0$ $D_{11} \approx D_{22}$ $D_{12},\ D_{66} \neq 0$	$B_{16} = B_{26} = 0$ $B_{11} = -B_{22}$ $B_{12} = B_{66} = 0$
Specially Antisymmetric Cross-Ply (n even)			
	$A_{16} = A_{26} = 0$ $A_{11} \approx A_{22}$ $A_{12} = Q_{12}h$ $A_{66} = Q_{66}h$	$D_{16} = D_{26} = 0$ $D_{11} \approx D_{22}$ $D_{12} = Q_{12}(h^3/12)$ $D_{66} = Q_{66}(h^3/12)$	$B_{16} = B_{26} = 0$ $B_{11} = -B_{22}$ $B_{12} = 0$ $B_{66} = 0$
Antisymmetric Angle-Ply			
	$A_{16} = A_{26} = 0$ $a_{16} = a_{26} = 0$	$D_{16} = D_{26} = 0$ $d_{16} = d_{26} = 0$	$B_{11} = B_{22} = B_{12} = B_{66} = 0$ $b_{11} = b_{22} = b_{12} = b_{66} = 0$
Specially Antisymmetric Angle-Ply (n even)			
$ij = 16, 26$ $ij = 11, 22, 12, 66$	$A_{ij} = 0$ $A_{ij} = \overline{Q}_{ij}h$	$D_{ij} = 0$ $D_{ij} = \overline{Q}_{ij}(h^3/12)$	$B_{ij} = -\overline{Q}_{ij}(h^2/2n)$ $B_{ij} = 0$

CHAPTER 3

METHODOLOGY

The methodology used in this research consists of three key elements: a function evaluator to calculate the dynamic responses of delaminated composite laminates based on analysis, a function approximation to simulate the dynamic responses from the analysis, and a function optimizer to search for the delamination pattern associated with measured dynamic responses that were simulated by function approximation.

The delamination detection problem comprises three computing modules, namely, finite element analysis, counterpropagation neural network, and genetic algorithm. The finite element models generate the dynamic responses (nature frequencies) of various delamination sizes and locations of composite plates. The counterpropagation neural network models are trained by the input-output data collected from the finite element analysis, a function evaluator. The detection process is then formulated as an interactive optimization between two modules, namely, the trained counterpropagation neural network, which is a function approximation, and the genetic algorithm, which is essentially a function optimizer. Figure 4 shows a flowchart of the current methodology which combines the three modules together.

The formulation of this delamination detection as an optimization problem is given below:

For a set of given measured natural frequencies ω_{mi}, $i = 1,\ldots, n_m$

find the delamination pattern corresponding to parameters x_1, x_2,..., x_n

to minimize an objective function *f*.

where the delamination parameters x_i, $i = 1,..., n$ are design variables, which define the delamination patterns (referred to as PIN in the MATLAB codes). The parameters include the sequence of thickness location (Z_d), in-plane locations (x_d, y_d), and the sizes of the delamination (a_x, a_y), in such an order as $\lfloor Z_d, x_d, y_d, a_x, a_y \rfloor$.

Objective Function

The objective function is defined to minimize the output error between the measured frequencies from FEA and predicted frequencies from CPNN for each delamination pattern (individual in the population). It is essentially a convergence or a termination criterion in the form of the optimal valuation. Although finite element analysis is the most time-consuming part of the entire delamination detection method, the value of the objective function is the key for GA to reproduce the offspring and plays an important role in the entire optimization procedure. Thus, defining an appropriate objective function is essential to the success of the delamination detection problem. There are two formulations that can be used as the objective function. They are the weighted sum of squares of the relative errors and the maximum of relative errors (in absolute values), which can be defined respectively as follows:

$$f = \sum_{j=1}^{n_m} W_j \left(\frac{\omega_{pj} - \omega_{mj}}{\omega_{mj}} \right)^2 \tag{119}$$

$$f = \max_j \left(\left| \frac{\omega_{pj} - \omega_{mj}}{\omega_{mj}} \right| \right) \tag{120}$$

where ω_{mj} and ω_{pj} are the measured and predicted j^{th} natural frequency of the delaminated laminate, W_j is the weight associated with j^{th} frequency, and n_m is the number of measured frequencies. The value of the objective function f depends on the specified delamination pattern and is a function of the design variables identifying the delamination pattern. Numerical experiments in previous research studies by Hieu The Le and Jin-Hwan Kim show that the definition of the objective function given in equation (120) leads to more accurate delamination detection results [7,8].

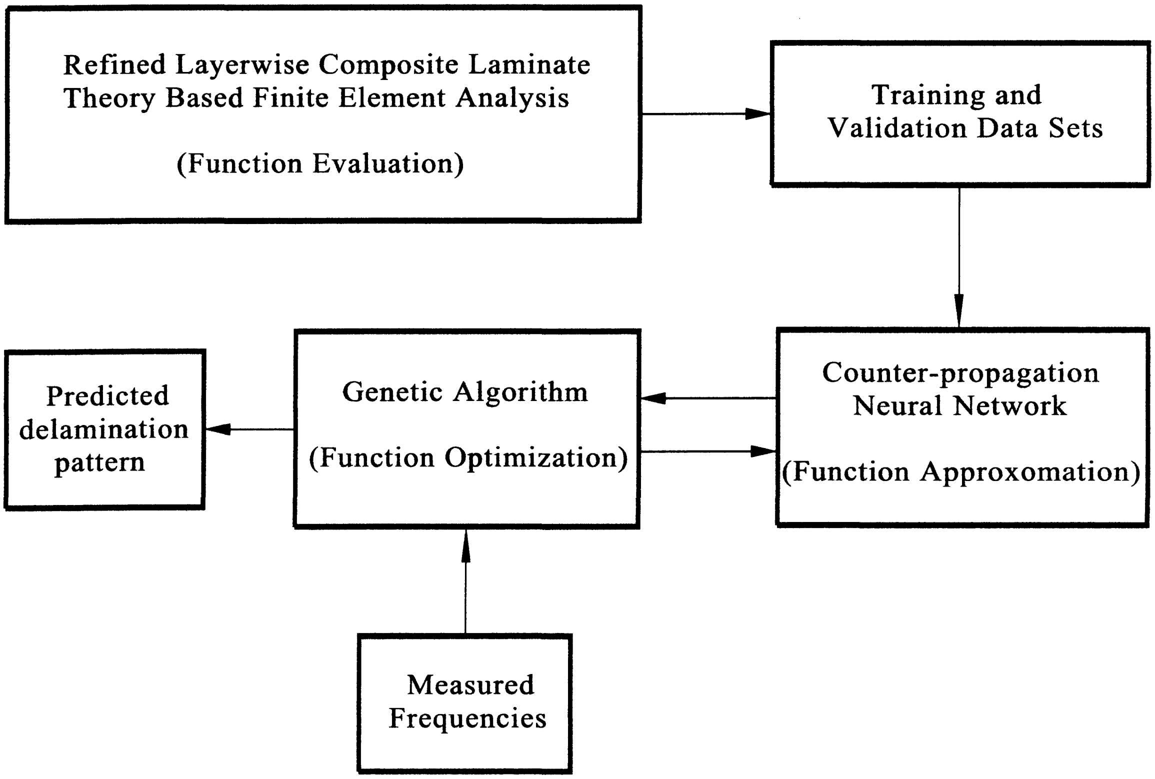
Refined Layerwise Composite Laminate Theory Based Finite Element Analysis
(Function Evaluation)
Training and Validation Data Sets
Counter-propagation Neural Network
(Function Approxomation)
Genetic Algorithm
(Function Optimization)
Predicted delamination pattern
Measured Frequencies

FIGURE 4. Flow chart of combined methodology of FEA, CPNN, and GA.

CHAPTER 4

FINITE ELEMENT ANALYSIS

Dynamics of Damaged Structures

This section is devoted to the construction of the dynamics of damaged structures, which provides a basis for damage identification [7]. Modal properties, i.e. natural frequencies and mode shapes, are used for delamination detection. Modal properties are related to the physical properties of the structure. The equations governing the dynamics of a multi-degree freedom system can be written in the time domain through the following assumed expression:

$$[M]\{\ddot{X}(t)\}+[K]\{X(t)\}=F(t) \tag{121}$$

where $[M]$and $[K]$ are the mass and stiffness matrices of the system, $X(t)$ and $\ddot{X}(t)$are the physical displacement and acceleration vectors, respectively, while $F(t)$ is the applied load vector. If equation (121) is transformed into the modal domain to form an eigenvalue equation for the jth mode, then:

$$[K]\{\phi_j\}-\lambda_j[M]\{\phi_j\}=0 \tag{122}$$

where λ_j is eigenvalue and $\{\phi_j\}$ is corresponding normalized eigenvector.

When damage occurs in a structure, the stiffness matrix, natural frequencies, and mode shapes of the damaged structure can be expressed as $[K_d]$, λ_{jd}, and $\{\phi_{jd}\}$, respectively. In addition, the natural frequencies and mode shapes of the damaged structure continue to satisfy the eigenvalue equation. The jth mode of such damaged structure therefore satisfies the equation:

$$[K_d]\{\phi_{jd}\} - \lambda_{jd}[M]\{\phi_{jd}\} = 0 \tag{123}$$

From equation (123), it may be deduced that the changes in the mass and stiffness matrices cause changes in the modal properties of the structure. Since the mass matrix $[M]$ is unaltered even in damaged condition, therefore, the modal properties can be identified through the identification of the correct stiffness matrix. In general, delamination decreases the natural frequency and causes changes to the mode shape of the composite laminate. This is due to the reduction of the stiffness caused by the delamination. Thus, the measurement of natural frequencies of a structure at two or more stages of its life offers the possibility of detecting the presence and location of delaminations.

Improved Layerwise Composite Laminate Theory

For the deformation of laminated structures, the layerwise distribution of mechanical properties in the thickness direction leads to the fact that displacement and transverse stress fields are continuous, but their derivatives with respect to the thickness coordinate at layer interfaces are discontinuous. Modeling of this phenomenon, that is, a zigzag-like form of displacements and an interlaminar continuity of transverse stresses, remains a challenging problem. Failure to capture these characterizations will result in unreasonable predictions of stress and strain fields in the analysis of multilayered composite structures.

Modeling and detection of delamination in composite structures has primarily been based on Classical Laminate Theory (CLT) and First-Order Shear Deformation Theory

(FSDT). However, the CLT analysis, which is based on the Love-Kirchhoff assumptions, is inadequate for structures with a high ratio of in-plane Young's modulus to transverse-shear modulus. Therefore, CLT is not capable of predicting the overall response for thick structures where the effect of transverse-shear deformation is significant. This means that transverse shears are completely ignored (CLT) or are modeled using shear correction factor (FSDT). Shear deformation plays an important role in the response analysis of composite structures due to the material discontinuities at each interface of the laminae. To address this issue with ply level accuracy, the in-plane displacement field is modeled using the superposition of overall first-order shear deformation and layerwise functions [1,10]. The first-order shear deformation based displacement field was used to address the overall response of the entire laminate and the layerwise functions were used to accommodate the complexity of zigzag-like in-plane deformation through the laminate thickness. The layerwise theory was modified to include the discontinuities in the in-plane and through-the-thickness of displacements induced by delamination. The displacement fields are supplemented with Heaviside unit step functions that allow discontinuities in the displacements.

Consider an N-layered laminated composite plate with multiple delaminations. The displacements of a point with the coordinates (x, y, z) are described using the superposition of first-order shear deformation and layerwise functions, as follows

$$\begin{aligned} U_i^k(x,y,z,t) &= u_i(x,y,t) + \phi_i(x,y,t)z + \theta_i^k(x,y,t)g(z) \\ &\quad + \psi_i^k(x,y,t)h(z) + \sum_{j=1}^{N-1} \overline{u}_i^j(x,y,t)H(z-z_j) \\ U_3^k(x,y,z,t) &= w(x,y,t) + \sum_{j=1}^{N-1} \overline{w}^j(x,y,t)H(z-z_j) \end{aligned} \tag{124}$$

where U_i^k denotes in-plane displacement and U_3^k denotes transverse deflection. The superscript k denotes the kth layer of the laminate and the subscript i denotes the coordinate x or y. The unknowns are $u_i, w, \phi_i, \theta_i^k, \psi_i^k, \overline{u}_i^j$ and $\overline{w}^j$. Note that u_i and w denote the displacement of the reference plane, ϕ_i are rotations of the normal to the reference plane about the x and y axes. θ_i^k and ψ_i^k are layerwise structural unknowns defined at each laminae. The terms $\overline{u}_i^j$ and $\overline{w}^j$ represent possible jumps in the displacement field due to delamination, allowing slipping and separation between sublaminates, and z_j denotes the delaminated interface. It must be noted that at the perfectly bonded interfaces, transverse shear stresses are continuous. At the delaminated interface, transverse shear stresses are zero. The function $H(z - z_j)$ is Heaviside unit step function. All interfaces between layers are initially assumed to be delaminated and perfectly bonded interfaces can be easily simulated by setting $\overline{u}_i^j$ and $\overline{w}^j$ to be zero in equation (124). Therefore, the number of delaminated layer interfaces is initially equal to the total number of plies in the laminate. The through-laminated-thickness functions, $g(z)$ and $h(z)$ are used to address the characteristics of in-plane zigzag deformation and have the following form:

$$\begin{aligned} g(z) &= sinh(z/h) \\ h(z) &= cosh(z/h) \end{aligned} \tag{125}$$

The above displacement fields lead to a total of $5 + 4N + 3(N - 1)$ structural unknowns, where N is number of layers. The total number of structural unknowns is dependent on the number of layers and delaminations, implying that computational effort will increase greatly if multilayered laminates are used. To further reduce the number of variables, the conditions of zero surface traction at top and bottom surfaces, and continuity of transverse shear stresses and in-plane displacements at interlaminar surfaces are imposed.

In this section, finite element methods are used to develop the mathematical models of structures. Natural frequencies have been computed for cross-ply laminates with delaminations placed at different locations. Numerical results indicate excellent correlation with analytical solutions and experimental results.

CHAPTER 5

COUNTERPROPAGATION NEURAL NETWORKS

Artificial Neural Networks

Introduction to Neural Networks

Neural networks are characterized by many of the inherent properties of the human thinking process. While the human way of thinking is less robust than the correct execution of a computer program, it has some very attractive features such as intelligence, creativity, intuition, adaptability, and spontaneous association. Mathematical models and algorithms, designed to mimic the information processing and knowledge acquisition of the human brain, are called neural networks. Such a neural network is a non-linear system consisting of a large number of highly interconnected processing units, nodes or artificial neurons (or simply neurons), which perform identical tasks. The basic elements of a neural network are the various nodes, the connectivity pattern of these nodes through directional links, the response mechanism of the node to its input signals, and the learning rule for training such a network [11].

A node collects a series of input signals and transforms them into an output signal via a transfer function, as shown in Figure 5. When the weight value at each link and the connection pattern are determined, the neural network is trained. This process is accomplished by learning from the training set and by applying certain learning rules. The trained network can be used to generalize for any of those inputs that are not included in the training set.

Nodes

Neurobiological methods of computation are distinctly different from traditional computation. The basic unit or computational processor in neural networks is a node. The node receives a sum of weighted inputs from various input sources, and processes this weighted sum through an activation function, as shown in Figure 5.

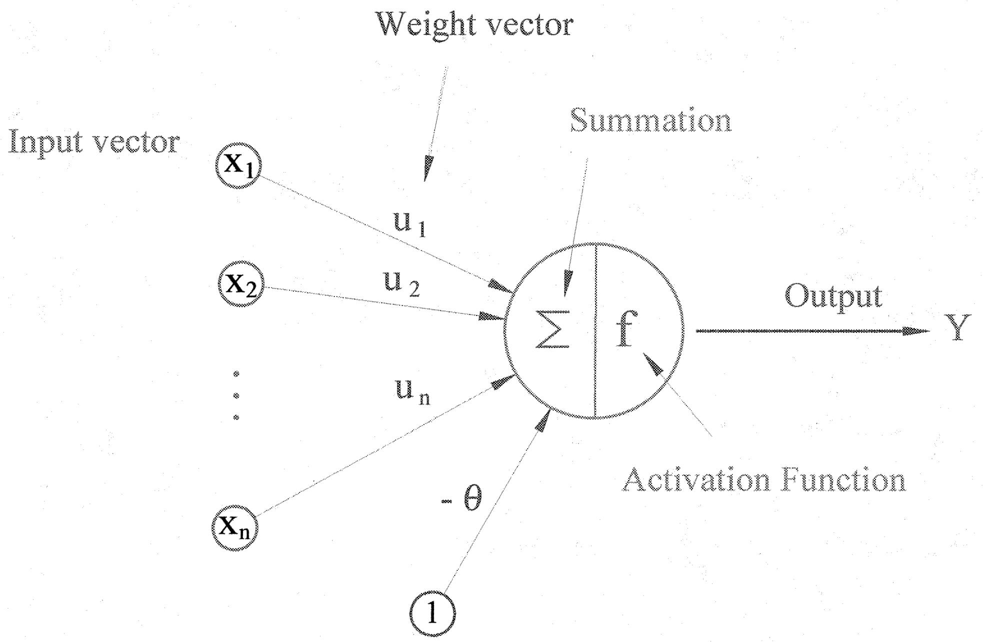

FIGURE 5. Structure of a node.

The input vector X , which is used to simulate the stimuli, is multiplied to its individual input x_i by the associate weight value u_i along its edge to the node. The products are summed at the node. The node fires only if the sum builds to the threshold or bias θ , which is estimated as a trainable weight for an additional input signal attached to each node having a constant input value of $x = 1$. In this way, a node carries out the computation and the output is $Y = f\left(\sum_{i=1}^{m} u_i x_i - \theta\right)$, where f is the activation function.

Architectures of Neural Networks

The computational capacity of one node may have some limitations. However, when many nodes are put together to form neural networks, a complex computational task can be performed. The arrangement of nodes and the pattern of connections between them are called the architecture of NN (type, structure, or topology of NN). The three main types of architecture are feedforward, recurrent, and cellular NN. In the feedforward NN, signals are transmitted in one direction, only from inputs to outputs.

Learning of Neural Networks

In the course of training, such a network of neurons 'learns' by changing the weights of its neurons. Two different learning methods can be distinguished: supervised and unsupervised learning. When learning is unsupervised, the neural network is provided with the input patterns. The network itself decides what output is best for a given input and reorganizes accordingly. After some iteration, it should settle to a stable state. All unsupervised learning systems have a general optimization criterion, such as the minimization of energy or distance, maximization of profit, and so on, which is used for the evaluation of the result at the end of each cycle. The goal of the supervised learning method is to find a model that correctly associates the inputs (representation of the objects) with the targets (representation of the responses). In supervised learning, the outputs of the neural network are compared with the desired outputs or target outputs, and the error is calculated. The weights are adjusted so as to minimize this error. Thus, the targets serve not only as a criterion for how well the system has been trained, but they also influence the correction of each weight.

Generalization

After "learning," the network should extract "regularities" or "rules" from the training data and be able to generalize data, i.e., to produce the output correctly for the input patterns that are never used in creating and training the neural network. Neural networks can be applied to four basic types of applications: association, classification (clustering), transformation (different representation), and modeling.

Neural Network Modeling Techniques

Neural networks are capable of modeling input-output functional relations, even when mathematically explicit formulas are unavailable. To create such models, it suffices to present a network of interconnected artificial neurons with a set of known input-output pairs, and adjust the strength of connections among neurons during the training process. These interconnection strengths, also referred to as the interconnection weights in the network, are computed according to some learning rules; once trained, the network can be used as a model of a given relation. Thus, well-trained neural networks represent a knowledge base where knowledge is distributed in the form of weighted interconnections in which a learning algorithm is used to modify the knowledge base from a set of given input-output pairs.

Kohonen Neural Networks

Introduction to Kohonen Neural Networks

In this section, a general overview of how to use a Kohonen neural network (K-NN) is given [12,13]. Learning in a Kohonen network is unsupervised, that is, the property to be investigated is not used in the training process. Thus, in principle, Kohonen networks can be used for similarity perception or for the clustering of objects. A counter-propagation network, which will be discussed in the next section, is a combination of supervised and unsupervised mapping of neural networks. K-NN and CPNN are quite flexible for adaptation to different types of problems and can be custom-designed to almost any type of data representation. In comparison to other neural network methods, the K-NN and CPNN are transparent and the results are easily interpretable. An advantage of K-NN and CPNN is that we can follow the predictions. When an object is situated into the trained network, its neighbors determine the prediction.

Supervised and Unsupervised Learning

Artificial neural networks (or simply neural networks) have several advantages over statistical techniques. They allow building models without knowing the actual modeling functions. From K-NN and CPNN, useful information about input and output variables can be extracted, respectively.

Lets consider the neural network (NN) with n inputs and m outputs. After the training, the network will be called a trained NN, which should be able to achieve one of the following tasks:

1. to generate, with no prior knowledge about the intrinsic property of input objects, a 2-dimensional topological distribution (a 2D map) of most active output signals (neurons) as the answer to all r input objects $X_s = (x_{s1}, x_{s2}, \ldots, x_{sn})$, $s = 1, \ldots, r,$

2. to yield, for any given signal $X_s = (x_{s1}, x_{s2}, \ldots, x_{sn}),$ the output signal $Y_s = (y_{s1}, y_{s2}, \ldots, y_{sm})$ as similar to the predefined target vector $T_s = (t_{s1}, t_{s2}, \ldots, t_{sm})$ as possible.

The first goal (2D map) calls for 'unsupervised' learning which requires only the input vector X_s and no target T_s to be known in advance. In such learning, the associated target vector T_s is required merely for checking the results after the learning is finished and not for the learning itself. In the unsupervised learning, the results, i.e., the target values T_s, are implied by the position of the vector X_s in a 2D map of neurons. Therefore, the obtained NN serves as a pointing device to the results.

In order to achieve the second goal, the so called "supervised" learning should be applied. For such learning, a set of input-output pairs $\{X_s,T_s\}$ with $X_s(x_{s1},x_{s2},...,x_{sn})$ being n-variable input objects and $T_s(t_{s1},t_{s2},...,t_{sm})$ the m -response output, or set of m targets associated with each X_s, is mandatory. During the supervised learning, the output vector Y_s is calculated for each individual input X_s and is compared to the target T_s. After comparison, a corrective measure is taken according to the particular NN strategy to change the weights in such a way that the corrections will assure better agreement between the Y_s and T_s in the final model. Thus, the goal of supervised learning methods is to find a model that correctly associates the inputs with the targets. The results of any supervised learning are the values of output vector coordinates as close to target values as possible. Supervised learning can be compared to fitting.

In any case, in order to generate (to 'train' or to 'teach') the final NN means to input all objects X_s (and all targets T_s if required) to the network a number of times. The input of all objects to the NN is repeated until either the agreements between all T_s and the produced outputs Y_s are acceptable or until the number of pre-specified number of epochs is exceeded.

Data Clustering and Classification

Clustering of data is a method by which large sets of data are grouped into clusters of smaller sets of similar data. The classification is conceptually different. It is an ordering of objects into exactly predefined classes. Thus, clustering is an unsupervised method designed to find relations between objects on the basis of their input representations, while classification is a supervised method designed to establish relations between the objects and the corresponding targets. Usually, clustering is the first pre-processing step in many data handling procedures. If a set of objects is separable by any clustering method on easily identifiable clusters, then classification of unknown objects can be achieved on the basis of already established clusters.

Division in Training and Validation Set

For testing of models, the data set should be divided into two subsets. The training set is used to build the model and the validation set to test it. To obtain reasonable predictions for the validation set, the training set must contain information of an entire descriptor space, and the final validation of models should be done with an external validation set.

Uncertain Decision Areas

To optimize the classification function, a training pattern distribution should be created with a well-defined decision class that exists [14]. To achieve this goal, the uncertain decision regions should be omitted. Uncertainty regions may originate from two different situations: low sampling regions, i.e., regions where the training pattern probability density is low and is scarcely represented in the training set, and regions where the probabilities of the different classes are very similar.

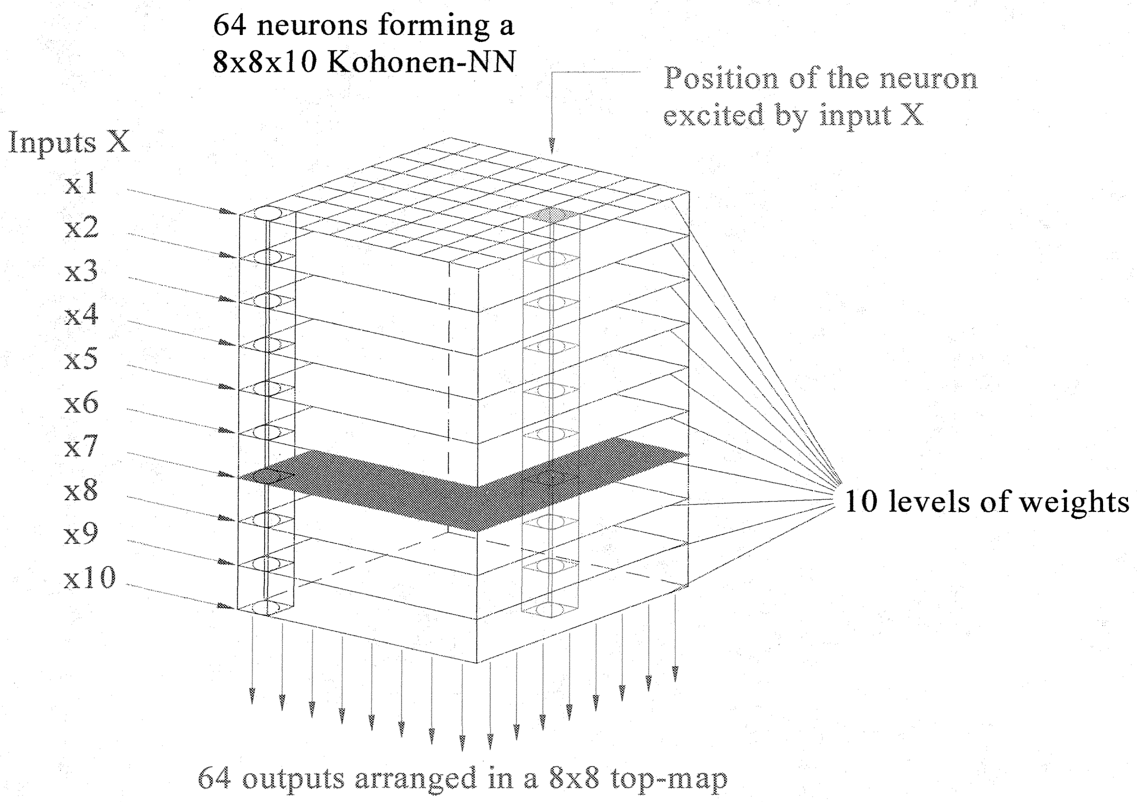

FIGURE 6. Kohonen neural network architecture.

Note: Kohonen neural network architecture represented as the 8x8 block of neurons. Each neuron is represented as a column of 10 weights. The level of weight which handles the seventh input variable x_7 is shown in green. The seventh weight in the neuron at the position (1, 1) is also shaded.

Architecture and Learning Strategy of Kohonen Neural Networks

SOM or Kohonen neural network is one of the basic types of artificial neural networks. SOM is a 'self-organizing maps' system which is capable of solving unsupervised rather than supervised problems. It consists of only one layer of neurons in which each neuron contains as many weights as there are elements in the input vector X_s;

that is, it contains n variables. Therefore, the number of weights in each neuron coincides with the number of input variables. Neurons are, in the Kohonen NN, ordered in a two-dimensional array (see Figure 6). Dimensions are specified as $N_{\text{net}} \times N_{\text{net}}$.

The learning of SOM is the projection from multi-dimensional space onto a two-dimensional array of neurons. The projection or learning of the network runs in two steps. The first step is the selection of the winning neuron and the second step is the self-organization of the map. In details, it runs as follows: the multi-dimensional vector X_s, which represents an object or an input pattern, is passed to all neurons as input. Because of the competitive nature of Kohonen layer, the larger value input vectors overpower the smaller vectors. For the network to operate properly, the input vector must be normalized. At each neuron, the comparison between vector X_s and the neuron's weights are made. As a rule, in the Kohonen NN, after input of training pattern, only one neuron is stimulated. Therefore, the neurons are competing among themselves to be stimulated and learning is usually referred to as competitive learning. The one selected from among $N_{\text{net}} \times N_{\text{net}}$ neurons is called the winning neuron. The actual selection of the winning neuron is based on the comparison between all weight vectors $U_j = (u_{j1}, u_{j2}, ..., u_{jn})$ and the input vector $X_s = (x_{s1}, x_{s2}, ..., x_{sn})$:

$$\text{winning neuron} \leftarrow \min\left\{\sum_{i=1}^{n}\left(x_{si} - u_{ji}\right)^2\right\}, \; j = 1,\, 2, ..., N_{\text{net}} \times N_{\text{net}} \tag{126}$$

After the winning neuron is found, the correction (adaption or learning) of weights starts. The weights of the winning neuron are modified to the input vector values and in the same time the neighboring neurons are modified to become similar to it. The corrections do not cover the entire network, not even the same number of neurons at different stages of the learning process. The number and the extension of corrections change during the learning. This means that the correction of weights do not affect all neurons in the NN, but only a small number of them: the ones that are topologically close to the winning one. Such corrections cause the topologically close neurons to start acting similar if similar training patterns are presented to the network. This means that two similar training patterns will excite two topologically very close neurons and vice versa; two very different training patterns will excite (select) the wining neurons topologically far away from each other. Thus, the neurons learn to pinpoint the location of the neuron in the network that is most 'similar' to the input vector X_s. After all training patterns are presented to the network, one learning epoch (cycle) is over. This procedure repeats until the weights are stabilized.

Weight Maps

Since the number of weights in each neuron is equal to the dimension n of the input vector $X_s = (x_{s1}, x_{s2}, \ldots, x_{sn})$, therefore each $N_{\text{net}} \times N_{\text{net}}$ Kohonen NN consists of $(N_{\text{net}} \times N_{\text{net}}) \times n$ weights. Before learning starts, all weights in the NN are randomized in the interval [0, 1]. Each neuron has the same number of weights and to each weight at a fixed position in the neuron, always the same variable is passed, i.e., the first weight u_{j1} handles only the first variable x_{s1}, the second weight u_{j2} handles the second variable x_{s2}, etc. Hence, in each level of weights only data of one specific variable is handled. Consequently, at the end of learning in each level, the visualization of the distribution of values of the particular variable can be seen on a table called a top-map.

Counterpropagation Neural Networks

Introduction to Counterpropagation Neural Networks

Hecht-Nielsen proposed CPNN as an alternate function approximator which can be developed on the available input-output data [15,16]. The underlying principle for CPNN is simple: for a given independent variable vector I not present in the available data set, find the independent variable vector in the data set closest to I. The criterion of closeness in the n-dimensional Euclidean space can either be distance based (minimum Euclidean distance) or angle based (minimum angle between vectors of normalized lengths). If X_k is the vector found closest in the data set then the value of $f(I)$ can be approximated as the dependent variable value corresponding to X_k. This technique runs into problems when the data set becomes very large. An important component of training in the CPNN is reduction of the data set into a respective data set of lesser, specified size [17]. This is attempted using Kohonen's learning algorithm, where large sets of data are grouped into clusters of smaller sets of similar data. Once such a reduction of the independent variable vector set is achieved, the estimates of the dependent variable values corresponding to the reduced independent variable vectors can also be calculated. Thus CPNN actually operates as a closest-matched lookup table and training a CPNN is an attempt to appropriately reduce the size of this lookup table.

The first counterpropagation network consists of a bi-directional mapping between input and output layers. In essence, while data is presented to the input layer to generate a classification pattern on the output layer, the output layer in turn would accept an additional input vector and generate an output classification on the network's input layer. The network received its name from this counter-posing flow of information through its structure. Most developers use a uni-flow variant of this formal representation of counterpropagation. In other words, there is only one feed-forward path from input layer to output layer. Therefore, the forward-only counterpropagation network is discussed.

Architectures of Counterpropagation Neural Networks

The topology of a counterpropagation network consists of three primary layers: input layer, competition layer or Kohonen layer, and interpolation layer or Grossberg layer, as shown in Figure 7.

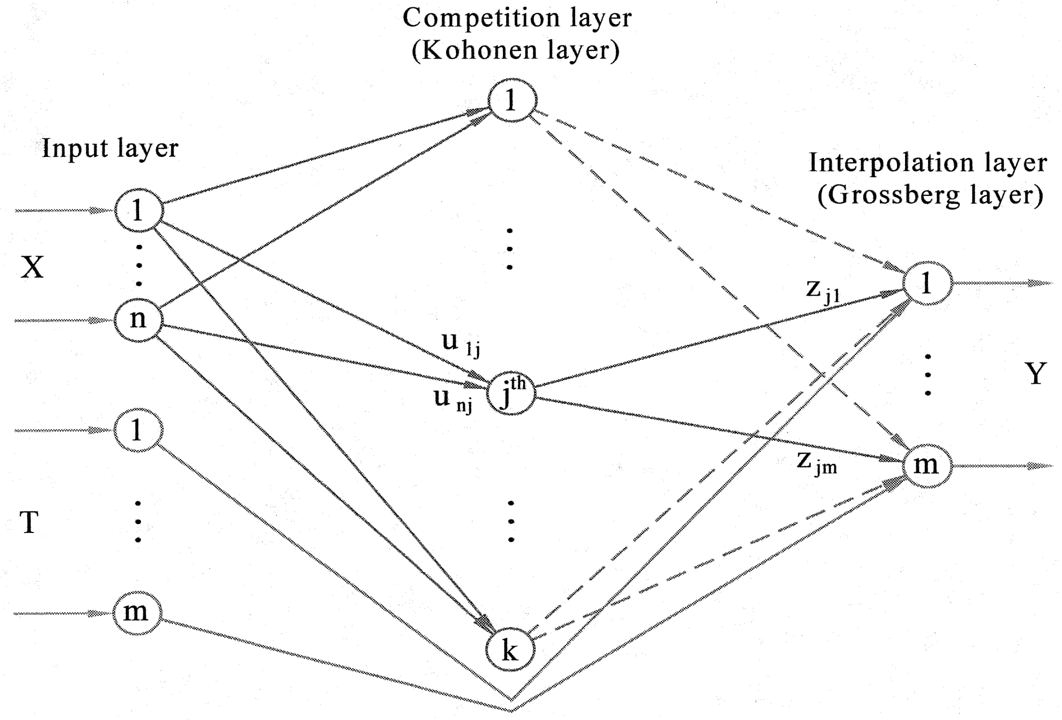

FIGURE 7. Counterpropagation neural network architecture.

The input layer of neurons is one to which the external stimuli, in this case the delamination patterns, are presented. The competition layer is a hidden layer with competitive nodes performing unsupervised learning. The hidden layer has the same structure and the same learning as in SOM stand alone. The interpolation layer is the output layer, which is fully connected with the hidden layer and is not competitive. Thus the CPNN is an up-grade of the Kohonen-NN. The main intention for setting up the CPNNs is to enable the Kohonen type of NN to solve the supervised type of problems. In fact, CPNN is a Kohonen-NN augmented by an output layer of neurons placed exactly below the neurons of the Kohonen layer. Although the output layer of neurons has exactly the same number and the same layout of neurons as the upper one, its neurons contain a different number of weights compared to the neurons in Kohonen layer. The output layer of neurons, which has as many weight planes as the target vectors have responses, is corrected separately and in a very similar way to the Kohonen layer. However, there is an important difference. The most distinguished neuron in the output layer around which the corrections are made is not the one that is closest to the target, but the one exactly below the selected neuron in the Kohonen layer, as shown in Figure 8. This means that the topological position of the winning neuron is identical in both layers: in the Kohonen and in the output one, respectively. The weights in the output layer are then adapted to the components t_{si} of the target vector $T_s = (t_{s1}, t_{s2}, \ldots, t_{si}, \ldots, t_{sm})$ instead of to the values x_{si} of the input vector.

Before network training is performed, the network architecture needs to be constructed by choosing the number of hidden units, M. If M is too small, the neural network will be insufficiently flexible and will give poor generalization of the data because of high bias. However, if M is too large, the neural network will be unnecessarily flexible and will give poor generalization due to a phenomenon known as over-fitting caused by high variance. Some researchers propose a rule of thumb, which states that the number of neurons should be one to three times the number of training patterns in a training set.

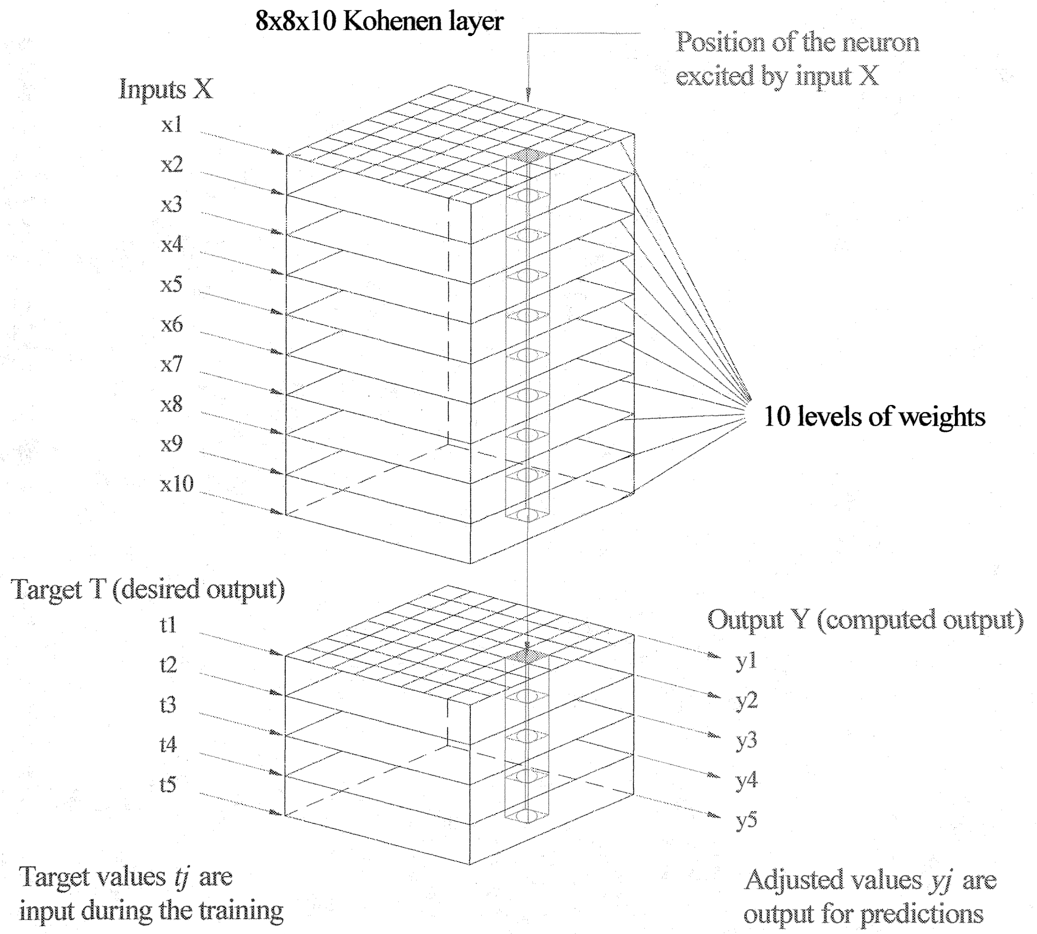

FIGURE 8. Counterpropagation neural network in array model.

Note: CPNN architecture consists of two layers of neurons: the Kohonen and the output layers. The central neuron (neuron excited by input X) is selected in the Kohonen layer and then corrections of weights are made around its position (arrow pointing) in the Kohonen and in the output layer.

Learning of Counterpropagation Neural Networks

For training the CPNN, the vectors of input patterns X and target or desired output T are presented to the network at the input and interpolation layers, respectively [18]. Thus, the number of nodes in the input and interpolation layers correspond to the number of elements in the vectors X and T, respectively. The vector of computed output is represented by Y. The training of the CPNN is performed in two consequently phases. The first phase adjusts the competitive layer and the second phase adjusts the output layer. When trained, the network works as follows: after presentation of an input to the input layer, the nodes in the hidden layer sum their inputs and compete to respond to that input. The node with the highest input wins and its activation is set to 1 while all others are set to 0. After the competition, the output layer calculates a weighted-sum on the outputs of the hidden layer. In details, it runs as follows.

For each pair of input pattern and target (X, T), each component of input pattern, x_i, is presented to the corresponding node of the input layer. Let U_j be the arbitrary initial weight vector assigned to the links between input nodes and the jth node in the competition layer. The transfer function for the competition layer is defined by the Euclidean distance between the weight vector U_j and the input pattern X as follows:

$$d_j = \left\| U_j - X \right\| \tag{127}$$

where $\left\| X \right\| = \sqrt{\sum_i \left| x_i \right|^2}$. For the given input pattern X, each node in the competition layer competes with other nodes and the node with the shortest Euclidean distance to X wins. As a result of the competition, the output of the winning node is set to 1.0 and outputs of the other nodes are set to 0. Thus the output of the jth node in the competition layer, Z_j, is given by

$$Z_j = \begin{cases} 1.0 & \text{if } d_j < d_i \text{ for all } i \\ 0 & \text{otherwise} \end{cases} \tag{128}$$

A weight u_{ji} assigned to the link connecting the node j in the competition layer and the node i in the input layer is adjusted according to the Kohonen (1988) learning rule

$$u_{ji}(n+1) = u_{ji}(n) + a\lfloor x_i - u_{ji}(n) \rfloor Z_j \tag{129}$$

where n is the iteration number and a is the learning coefficient. For the learning coefficient, Hecht-Nielsen (1988) suggests a number in the range of $0 < a \leq 0.8$.

These training steps mentioned above are repeated until all input vectors have been classified properly. After the weight vector U_j of the competition layer stabilizes, the interpolation layers start to learn the target. Thus, clusters and output vectors are mapped. The weight assigned to the link between the winning node j in the competition layer and the lth node in the interpolation layer, v_{lj}, is adjusted according to the learning rule suggested by Grossberg (1982)

$$v_{lj}(n+1) = v_{lj}(n) + b\lfloor T_l + v_{lj}(n) \rfloor Z_j \tag{130}$$

where b is the learning coefficient. Hecht-Nielsen (1988) suggests a number in the range of $0 < b \leq 1.0$. To circumvent the arbitrary trial-and-error selection of the learning coefficients a and b encountered in the counterpropagation algorithm, a simple formula is proposed as a function of the iteration number in the following form

$$a = b = 1/(n+1)^2 \tag{131}$$

The interpolation layer uses a weighted summation function as a transfer function. The lth element of the computed output of the network, Y_l, is determined by

$$Y_l = \sum_j v_{lj} Z_j \tag{132}$$

During training, only one node of the competition layer can win and the corresponding output is set to 1.0. After connection weights in the network stabilize, the performance of the network can be tested by using untrained patterns. During the verification of untrained patterns, the number of winning nodes in the competition layer can be more than one. The nonzero outputs of winning nodes are set so that the node associated with the weight vector closest to the given untrained pattern has the largest output. However, the sum of the outputs of winning nodes in the competition layer remains equal to 1.0. By letting n_{win} be the number of winning nodes in the competition layer and S_w be the set of winning nodes, we define Z_j as follows:

$$Z_j = \begin{cases} \dfrac{\left(\sum_{n=1}^{n_{\mathrm{win}}} d_k\right) - d_j}{2\sum_{n=1}^{n_{\mathrm{win}}} d_k} & \text{if } Z_j \in S_w \\ 0 & \text{if } Z_j \notin S_w \end{cases} \tag{133}$$

where d_k is the Euclidean distance between the weight vector assigned to the kth winning node and the untrained pattern. Thus, the output of the network for untrained patterns is calculated by (132) with more than one nonzero output from the competition layer, Z_j.

The selection of the number of the winning nodes plays an important role in the performance of a counterpropagation network. Different numbers of winning nodes in the competition layer show different performances. If the number of winning nodes is set to one, the winning node is the node whose weights are the closest to the input vector in a Euclidean sense. In this case, the CPNN works as a simple nearest-neighbor classifier. If the number of winning nodes is more than one, the accuracy of the mapping approximation can be improved significantly. In this case, the CPNN works as an interpolator with multiple winning nodes. The number of winning nodes in the competition layer is selected based on the set of input vectors (training patterns) in any given problem. The optimum number of winning nodes can be obtained from experience and through numerical experimentation.

Operation of Counterpropagation Neural Networks

After the training is completed, the Kohonen layer in the CPNN acts as a pointer device determining for any X_s the position of the neuron in the output layer in which the answer is stored. During the learning procedure, the answers (components of target vectors T_s) are distributed throughout the assembly of output neuron weights: target components t_i are distributed only in the ith level of output weights. Because each output component is distributed throughout the level of corresponding weights, each output neuron contains the answers for all classes even if its counterpart in the Kohonen layer above it was never excited during the training. The values of the weights in each output neuron j are distributed between 0 and 1 in such a way that the sum of all m output weights is equal to one.

When compared to backpropagation neural network (BPNN), the superior convergence property and a substantial decrease in the central processing unit time were found for CPNN. The superior convergence of CPNN is due to the fact that it is less sensitive to the learning coefficients. Also, the specific connection weights associated with a winning node, instead of all connection weights, are adjusted to minimize the error for a given training pattern. Unfortunately, the current structure of CPNN is not suitable for performing continuous mappings. In the next section, a new type of CPNN architecture is introduced, which possesses the desirable characteristics of the existing CPNN as well as an improvement of continuous function mapping property.

Improved-Counterpropagation Neural Networks

Introduction to Improved-CPNN

Detection of damage in structural systems can be formulated as a direct or an inverse problem and solved by means of a new approach utilizing feature-sensitive neural networks [19-22]. In this section, a modified or improved version of a counterpropagation neural network, which belongs to a class of feature-sensitive neural networks, has been selected for solving the damage-detection problem. The major reasons for using a feature-sensitive type of network are the ease with which the CPNN can be trained, its ability to model data for a large domain of a functional relation, and development of approximation for direct and inverse relations at the same time, and work with incomplete and noisy input data. The CPNN functions as an associative memory device that meets this robustness requirement. The advantage of using associative memories stems from the fact that even a partial knowledge of certain features may suffice to make a perfect recall.

In the neural network approach, possible damage and its extent is detected by relating changes in structural response directly to degradation of structural components. To diagnose the condition of a structure, the input vector, representing damage, is provided to the network, and the output vector, representing the system response, is returned. In absence of an exact analytical solution to the problem, approximation yielded by an improved-counterpropagation neural network is examined. Improvements to the CPNN network include a dynamic adjustment of the network size, the use of averaging operators for training, and an increased accuracy of approximations based on a nonlinear blend of interconnection weights.

Disadvantages of Original CPNN

The original CPNN architecture is basically a combination of the self-organizing map of Kohonen and the outstar structure of Grossberg [19]. The activation function of Kohonen neurons is important because it affects the output of the CPNN directly. The ordinary CPNN uses a threshold logic function in its Kohonen neurons. It is the threshold logic function that makes CPNN well suited for pattern classification applications. On the other hand, the threshold logic makes CPNN incapable of implementing continuous function approximations. In this case, the continuous functions approximated by CPNN are quantized or truncated. Also, the CPNN does not have continuous contributions to its output from Kohonen neurons. It "switches" the firing Kohonen neuron (or neuron group) from one to another. This switching results in different contributions to the output through the Grossberg weights. This is the reason why the CPNN output always has small non-smooth regions. There are several ways to overcome this problem. One way is to use continuous activation functions for Kohonen neurons. The second method is to let the CPNN operate in the interpolative mode, that is, to ask more than one neuron (a group of neurons), instead of a single neuron, to fire. The third way to improve the CPNN performance is to employ nonstandard activation functions. These functions should have a gradual decay property after each firing. Therefore, the network output depends not only on the current network information but on past information as well.

Also, an ordinary CPNN architecture generally requires a large number of Kohonen units to achieve accurate mapping approximation. Since CPNN adjusts its weights to conform to the statistics of the input vectors, the number of Kohonen units required to reach a desirable level of accuracy depends also on the complexity of the mapping and the statistics of the input-output vector selections (the density of the training sets). Generally, the mapping relation (complexity) of CPNN models and the data representation of the input-output map are specified. Hence, the accuracy of the mapping approximation will basically depend on the number of Kohonen neurons in the Kohonen layer. A direct way to improve the function approximation accuracy would be to simply increase this number which will result in a larger size network and correspondingly greater computation time. This cannot, of course, constitute a feasible solution. Therefore, a new CPNN architecture will be introduced, which avoids these disadvantages. The improved-CPNN is proposed as a tool for fast function approximations (FFA) [20-22]. In essence, the improved-CPNN is a clustering device capable of optimal distribution of cluster examplars in a uniform sample space with an integrated interpolation scheme to improve the accuracy of approximations.

Architectures of Improved-CPNN

There are three distinct components in the improved-CPNN architecture as shown in Figure 9. These are a fan-out layer, a hidden layer of feature-sensitive neurons, and an interpolator of outstar.

The hidden layer of neurons is essentially a clustering device which for continuous mappings classifies input patterns presented to them on the basis of some distance measure. The unsupervised learning of the feature-sensitive neurons is based on the minimum disturbance principle. According to this principle only that weight vector which is the closest in the sense of the selected norm to an input pattern is updated. To describe a flow of information through the network, assume that a set of training data contains M samples of some mapping, $\phi: X \in R^n \rightarrow T \in R^m$, and that a sample space is uniform. As the structure of the fan-out layer shows in Figure 9, an input pattern is simply a juxtaposition of both input and output vectors, and, therefore, can be defined as a vector $P_i \equiv [X, T]^T = [\text{PIN}, \text{target}]^T \in R^{n+m},\ i = 1, \ldots, M.$ Since the input patterns are directly passed to the first processing layer of the network, the weight vectors of the feature-sensitive neurons that are to be determined during the training are defined as $z_j \equiv [z_1, \ldots, z_n, z_{n+1}, \ldots, z_{n+m},]^T \in R^{n+m}$, and are elements of the same space as the input patterns. Such a definition of the input patterns, and consequently the weight vectors, has interesting ramifications. First, the improved-CPNN may model a direct relation, its inverse, or both at the same time. This can be easily achieved by specifying with respect to what part of the training pattern, P, is clustering to be performed. The direct relation would require the first through n^{th} component, the inverse relation, the $(n+1)^{th}$ through $(n+m)^{th}$, and for both relations the whole pattern would have to be considered. The distinctive feature of the neural network approach is that the data required to model an inverse relation are simply obtained as the solution of the direct problem. By interchanging the input and output vectors during the training process, an approximate model of the inverse relation is developed. Second, the clustering procedure can virtually use any part of the training pattern, allowing for further flexibility such as an incomplete direct or incomplete inverse relation, and distorted patterns. This property may be explored in the use of the network as an associative memory device.

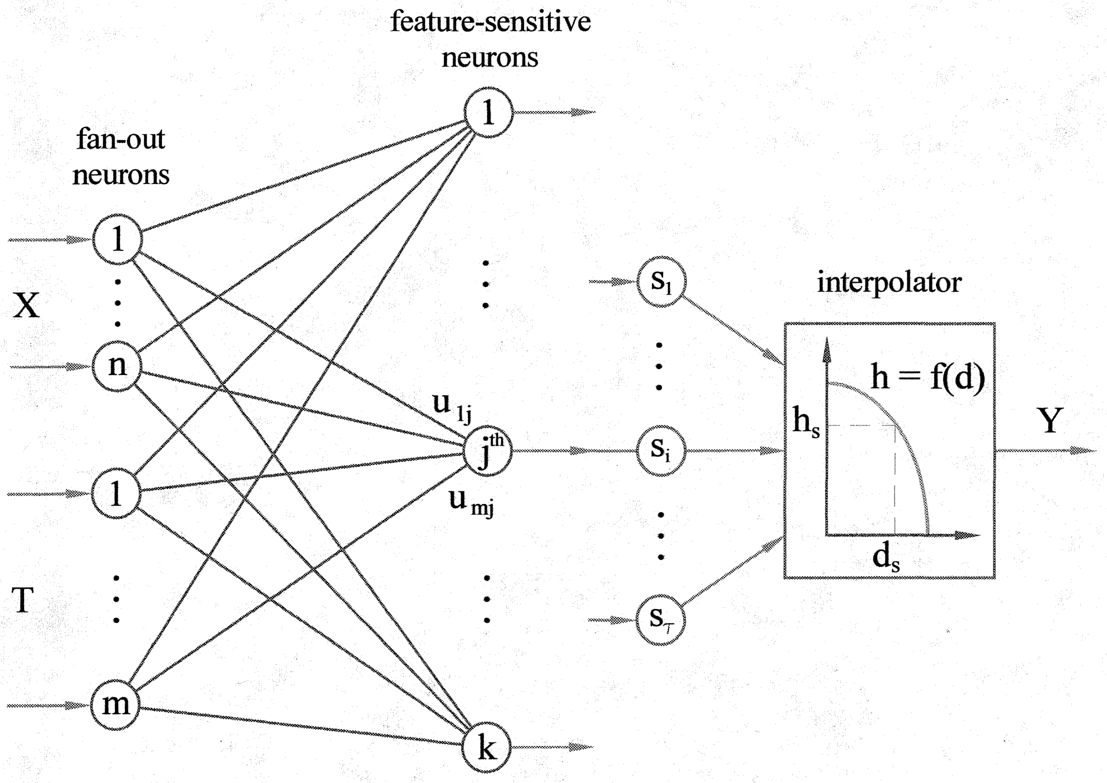

FIGURE 9. Improved-counterpropagation neural network architecture.

Note: Improved-CPNN architecture of mapping $X \in R^n \mapsto T \in R^m$

The number of the feature-sensitive neurons, or the size of the network, k, depends on a value of the network resolution, δ_r, and changes dynamically during the training process. The resolution may be thought of as the size of a mesh in a multidimensional sample space, or approximately the distance between the closest cluster examplars, and, therefore, directly related to the accuracy of network approximations.

The purpose of the interpolator is to take τ cluster examplars found to be the closest to an input vector (the δ-neighborhood), and to blend them into a single output vector on the basis of a nonlinear averaging scheme. Therefore, network response is defined as a vector, $\mathrm{Y} = f(z_s),\ s = 1, \ldots, \tau$. The number of examplars participating in a network response is determined by a parameter, δ (size of neighborhood), which is essentially a distance measure. A value of this parameter (size of neighborhood) is set during testing to minimize a response error.

Learning of Improved-CPNN

There are several new features in the modified version of the CPNN that improve its performance and facilitate its use. A control parameter δ_r, referred to as a network resolution, is the only arbitrary parameter required to determine the size of a network and control accuracy of approximations. An averaging operator (an arithmetic mean) is used to determine the outstar weights during the training and to places the weight vector at the geometric center of all outputs for which the connection is updated; in addition, it eliminates a learning rate required by the original network. Finally, and perhaps most importantly, a nonlinear interpolation scheme is introduced to increase the accuracy of network estimates. This scheme blends several outstar weights in constructing a network response.

There are several measures that may be used for defining the network resolution value. In problems where the network is employed as a function estimator, the absolute value or Manhattan norm defined for two arbitrary vectors x and y as, $\| \,.\, \|_1 = \sum_{i=1}^{n} |x_i - y_i|$, is preferred. This norm is equally sensitive to each vector component, a feature shown to be essential in the vector matching procedure used in the learning rule of the network. It is also clear that this norm does not require normalized vectors to be processed. This is distinctly different from the distance measure used in the original CPNN, where the dot product of the normalized vectors was used. For normalized vector components and the length of a vector equal to n, the range of a resolution is 0 to $0.8\,n$. These limit values produce a network of the size equal to the total number of the training samples and unity, respectively.

Operation of Improved-CPNN

In operation, the network returns an approximation of the T vector when presented with some input vector, $X \in R^n$. In the improved version of the CPNN, a nonlinear interpolation scheme based on a membership function has been introduced to increase accuracy of network response. The process of developing approximations can be summarized as follows. First, the minimum operator finds the best match u_{i^*} for the vector X among all k weight vectors of the hidden layer; it also computes the distance δ_0 between the weight vector of the winning neuron u_{i^*} and the input vector. Then, the δ - neighborhood of the closest neuron is defined as containing all neurons for which distance is less than $(\delta_0 + \delta)$ from the current input. Next, an index set, S, is determined that contains all indices, s, for which the following relation holds, $\delta_s = \sum_{l=1}^{m} |x_l - z_{ls}| \le \delta_o + \delta$, where $1 \le s \le k$. Since each neuron in the hidden layer is connected to all outstarts, the network response may be calculated as a nonlinear blend of the outstars interconnection weights $\sum_{s=1}^{\tau} h_s z_s$. To compute an individual contribution h_s of each neuron to the network response, Y, a membership function, $f(d_s)$, defined as a power function, $f(d_s) = 1 - d_s^{\,r}$, is used. Here d_s is a normalized distance calculated as $d_s = |\delta_o - \delta_s|/\delta$. The contribution, h_s, of each neuron is normalized as $h_s = (1 - d_s^{\,r}) \Big/ \sum_{s=1}^{\tau} (1 - d_s^{\,r})$, so that the contribution of all neurons adds up to one. It is worth noting that the number of contributing neurons depends on the value of δ (size of neighborhood), and may vary for different input vectors. In the case where the winning neuron coincides with the input vector or is very close to it, the winner almost certainly becomes the sole contributor to the response. But in the case where there is no decisive winner, the blend of many weight vectors is created as the output. The control parameter δ is set during the testing to minimize a response error. The shape of the membership function, and hence the relative contribution of the participating neurons is controlled by the power r. Its value has to be adjusted to minimize the error; it is usually in the range 0.1 – 6.0. Figure 10 illustrates how the individual contribution changes for different values of the exponent, *r*. In all applications, this interpolating scheme significantly reduced about 30–40% of error in the approximations. This indicates that the network response is developed through a nonlinear interpolation procedure with an advantage of removing the sharp boundaries among clusters.

Thus, the neural network approach to the damage detection problem proved to be a promising alternative to more traditional techniques, particularly for on-line damage detection when the processing efficiency becomes an important issue.

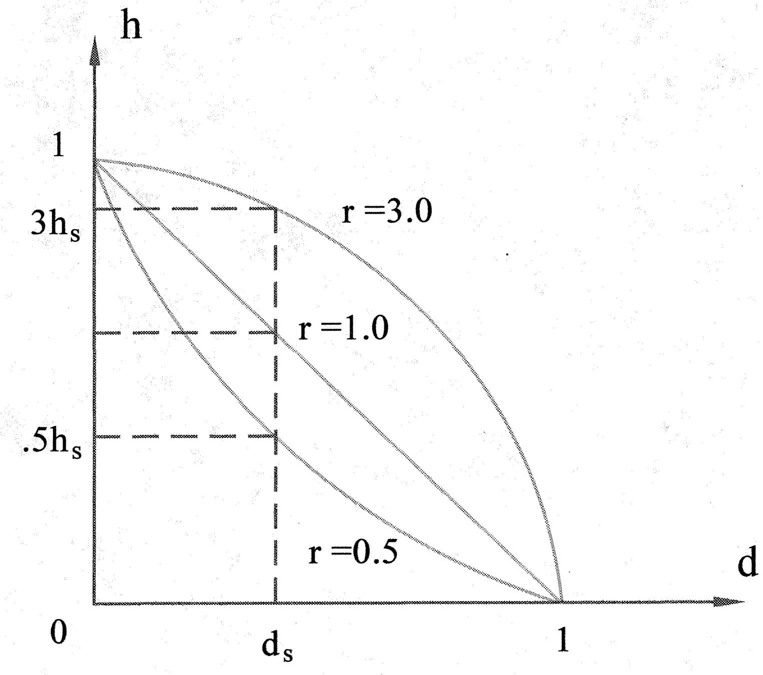

FIGURE 10. The individual contribution of each neuron changes for different values of the exponent, r.

CHAPTER 6

GENETIC ALGORITHMS

Detection of structural damage is an inverse problem in structural engineering. The solution space of this problem is multi-modal, which exist at many local optima. In a multi-modal problem, the solution using the hill climbing method may often be trapped into a local optimum. There are three main questions in the damage detection: the existence, location, and extent of damage. The detection of delamination sizes and locations of delaminated laminates can be formulated as an optimization problem. In this section, GA is employed to find the optimal solution by minimizing the error of the response characteristics (frequencies) between the analytical model and the measured data of the delaminated laminates. Unlike the traditional mathematical methods, which guide the direction of hill climbing by the derivatives of objective functions, GA searches the problem domain by the objective function itself at multiple points. Thus, GA is an alternative class of optimization method but does not rely on the slope for optimization.

Genetic algorithm (GA) was inspired by Darwin's theory of natural evolution and selection [23-26]. Genetic algorithm is a simulation of natural evolution where the law of survival of the fittest is applied to a population of individuals. The philosophy of survival of the fittest facilitates arrival at the globally optimal solution; the methodology is implemented numerically and developed for optimization problems, where natural evaluation and adaptation to environmental variation is simulated mathematically by using GAs. Thus, GAs are stochastic optimization methods and provide a powerful means to perform directed random searches in a large problem space as encountered in damage detection. Analogous to genes in genetics, GA represents the parameters in a given problem by encoding them in a string. Instead of finding the optimum from a single point in traditional mathematical optimization methods, in GA a set of points, that is a population of coded strings, is used to search for the optimal solution.

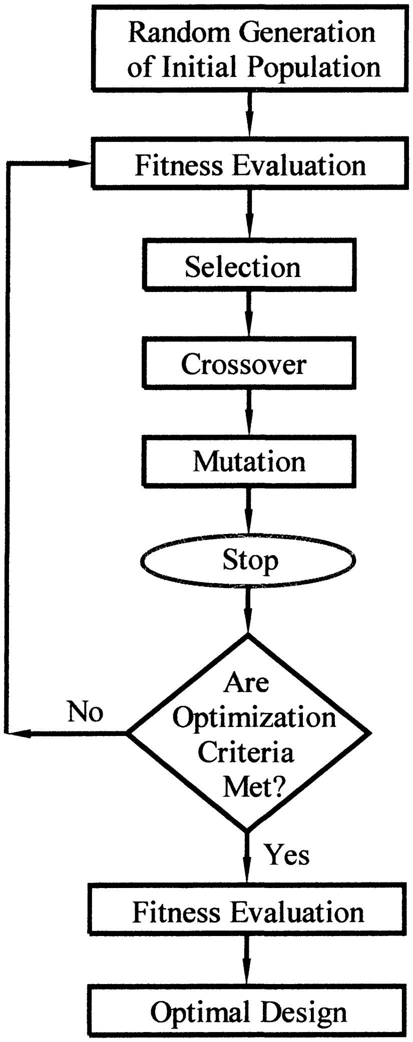

FIGURE 11. Structure of a simple genetic algorithm.

The search is initiated by selecting a number of candidate design variables either randomly or heuristically in order to create an initial population (possible solutions, or first generation), which is then encouraged to evolve over generations to produce new designs which are better or fitter. This improvement is achieved through the processes of evaluation, reproduction, breeding, and mutation. A flow chart diagram of the GA process is shown in Figure 11.

Initial Population

GA starts with encoding design variables which converts the design variables into bit strings that are counterparts of chromosomes in biological genetics. The bit strings can be coded by binary or real numbers that are most widely used in GAs. Before applying a GA to a task, the selection of an initial population must first be developed, which consists of a set of points selected from within the search space. The size of the initial population is maintained constantly through all generations and is increased with the size of design problems. The larger population maintains its diversity longer and finds better solutions. The smaller population seems to become homogeneous quickly, and from that point on improvement is slow, driven primarily by mutation. However, the smaller population arrives at better answers much faster than the larger population. This indicates that large populations are less influenced by good potential solutions early on, analogous to having a greater degree of inertia. Consequently, there is a tradeoff between the convergence rate and the fitness of the final answer. In summary, if the population is oversized, this will adversely impact the speed of convergence, thereby diminishing the efficiency of the search. Conversely, an undersized population will enhance the chances of premature convergence on a non-optimal solution, thus impacting search reliability. The initial population is sometimes artificially manipulated by various means in an attempt to improve the performance of the genetic search. One special technique is seeding, whose effect is to nudge the search onto a favorable path through the search space [27]. When seeded, the GA quickly improves the approximation and finds a network that gives the desired output. Seeding verifies that there is a correct answer and that the GA could find it.

Mechanics of Genetic Algorithms

After initialization of the first generation, the fitness of each individual is evaluated by an objective function (step 1). In the reproduction step (step 2), the genetic operators of parent selection, breeding (step 3) and mutation (step 4) are applied, thereby providing the first offspring generation. Iterations of steps 1 to 4 are performed until the objective function converges or the process is externally terminated. In detail, the four-step processes are shown below [28].

Evaluation

The first step in each generation is to evaluate the current chromosomes. This is the only step where we use the interpretation of the chromosome; in all other steps, the chromosome is treated as a bit string. Each chromosome in the population is decoded and the resulting network is tested with the training data. To evaluate networks, we calculate the maximum of relative error (MRE) which is defined as an objective function for the training set between measured and predicted by FEA and CPNN, respectively. With the fitness of the chromosome equal to $1/(1+\mathrm{MRE})$, this means that the better a network performs, the higher its fitness, with a perfect network having a fitness of 1.

Reproduction

The next step in each generation is to create a new population based on an evaluation of the current one. To generate fitter strings, GA reproduces the population according to their relative fitness; the strings with higher fitness have a better chance of passing their genes to the next generation. Thus, reproduction takes successful chromosomes and reproduces them in accordance to their fitness functions. Every chromosome generates a number of copies of itself based on its performance, with the best chromosomes producing several copies of themselves and the worst not producing any. This is the step that allows GAs to take advantage of a survival-of-the-fittest strategy.

The selection of mating pairs for reproduction is a crucial step in GA. There are several ways to calculate the number of offspring generated by each chromosome. The most common technique is rationing: each chromosome produces a number of offspring proportional to its fitness, with the restriction that the total number of chromosomes per generation remains constant. Thus, if one chromosome's fitness is twice that of another, the superior chromosome would produce twice as many offspring. However, there are two major problems with this method. First, if all the chromosomes have a similar fitness, each member in the population would produce one offspring. This results in little pressure to improve the fitness of the population. Second, if one chromosome has fitness much larger than any other, that chromosome would create most, if not all, of the new offspring. The chromosome would dominate the population, resulting in a loss of genetic diversity. This problem has been labeled premature convergence.

The other method is ranking, in which the whole population is sorted by fitness. The number of offspring each chromosome generates is determined by how it ranks in the population. A typical example of the ranking method can be described as follows. The top 20 percent of the population generates two offspring each, the bottom 20 percent generates no offspring, and the rest generates one offspring each. No one chromosome can overpower the population in a single generation, and no matter how close the actual fitness values are, there is always pressure to improve. The primary disadvantage of ranking is speed, because better chromosomes cannot easily guide the population, forcing good answers to develop more slowly.

The Tournament Selection (TS) method provides good selective pressure by holding a tournament competition among $N(=3)$ individuals. The best individual (winner) from this tournament is the one with the highest fitness and the winner is then inserted into the mating pool. The tournament competition is repeated until the mating pool for generating new offspring is filled. The mating pool comprising the tournament winners has a single average fitness.

<u>Breeding (crossover)</u>

The previous step creates a population whose members are currently the best at solving the problem; however, many of the chromosomes are identical, and none differ from those in the previous generation. Breeding combines chromosomes from the population and produces new chromosomes that, while they did not exist in the previous generation, maintain the same gene pool. In natural evolution, breeding and reproduction are the same step, but in GAs they have been separated to allow different methods for each to be experimented with and independently evaluated. In biological reproduction, the chromosomal pattern of the child is derived from the chromosomal strings of the two parents and consequently the child inherits the characteristics of both. In GA, breeding can exploit knowledge of the gene pool by allowing good chromosomes to combine with chromosomes that are not as good. This is based on the assumption that each individual, no matter how good it is, does not contain the answer to the problem. The correct answer is contained in the population as a whole, and can only be found by combining chromosomes.

There are several methods for breeding, the most common being crossover. The crossover operator mixes genetic information in the population by cutting pairs of chromosomes at random points along their length and exchanging over the cut sections. Many variations on crossover have been used, but there is no consensus as to which is best. The process of crossover ensures that design information is transferred from one generation to the next, essentially by a simple swapping of one (single-point) or two sections (two-points) of bit string representation of two parent-designs to obtain two offspring design solutions. The positioning and extent of crossover time is chosen at random and may be different for each mating couple in each generation.

Mutation

Following crossover, the natural evolution concept of mutation is introduced into GA via the occasional switching of the bit value at a randomly selected location of the generated strings. This action is important since it guards against premature convergence of the design towards an optimal solution. The procedure is repeated until the new generation ceases to improve according to the objective function. When this occurs, the fittest individual of the youngest generation represents optimal design solution.

GAs usually require a large number of iterations and thus has a high computational cost. The genetic search procedure requires a proper selection of crossover and mutation operators. Goldberg's study of genetic algorithms in function optimization suggests that good genetic algorithm performance requires the choice of a high crossover rate, a low mutation rate (inversely proportional to the population size), and a moderate population size. In summary, after reproduction, a one-point crossover with the probability of p_c is performed to evolve new offspring. The probability p_c in this study is 0.5. In addition, to inhibit premature convergence during the reproduction and crossover, mutation is implemented to maintain the genetic variability of the strings. Mutation is conducted with the probability of $p_m = 0.005 - 0.02$.

CHAPTER 7

RESULTS AND DISCUSSION

The detection of single internal delamination has been addressed in the present study and only the rectangular delamination configuration was considered. To save computational time, an eight-ply carbon-epoxy laminate with stacking sequence $[0/90]_{2s}$ was adopted. The ply thickness is 0.0127 cm and the material properties are $E_1 = 134.4\text{GPa}$, $E_2 = 10.34\text{GPa}$, $G_{12} = 5.0\text{GPa}$, $\nu_{12} = 0.33$, and $\rho = 1477\text{kg}/\text{m}^3$. Its in-plane geometry is 8 cm long and 4 cm wide. In order to prevent the occurrence of frequency-identical delaminations which are symmetric to the central line, the left and lower edges of the laminate plate are fixed while the upper and right edges are free. The geometry and boundary conditions for the laminate plate are shown in Figure 12.

FEA Model

To construct a CPNN, finite element analysis was used to calculate the natural frequencies of laminate plate with different delamination patterns. FEA provides all the training and validation data sets for both neural network and genetic algorithms. Therefore, the accuracy of FEA model directly affects the outcome of delamination location detection. The finite element mesh consists of four-node quadrilateral plate elements and delamination patterns in the central area of the plate are considered. The data sets (including delamination patterns and associated natural frequencies) have been generated to train and validate neural networks. Each delamination pattern is represented by a mixed type of data structure expressed in the form $\lfloor z_d, x_d, y_d, a_x, a_y \rfloor$ which consists of four real (continuous) variables and one integer (discrete) variable. These variables comprise two real variables (x_d, y_d) specifying the in-plane coordinates of the left-lower corner of the delamination, two real variables (a_x, a_y) specifying the length and width of the delamination, and one integer variable (z_d) specifying the thickness location of the delamination which is defined as the number of layers above the mid-plane where delamination occurs. When delamination occurs at the mid-plane, the value of z_d is taken as 0. Due to the symmetry of the layup, only delamination occurring in the upper half of the laminate has been considered. Thus, the range of z_d is limited to [0, 1, 2, 3] for the laminate layup considered in the present study.

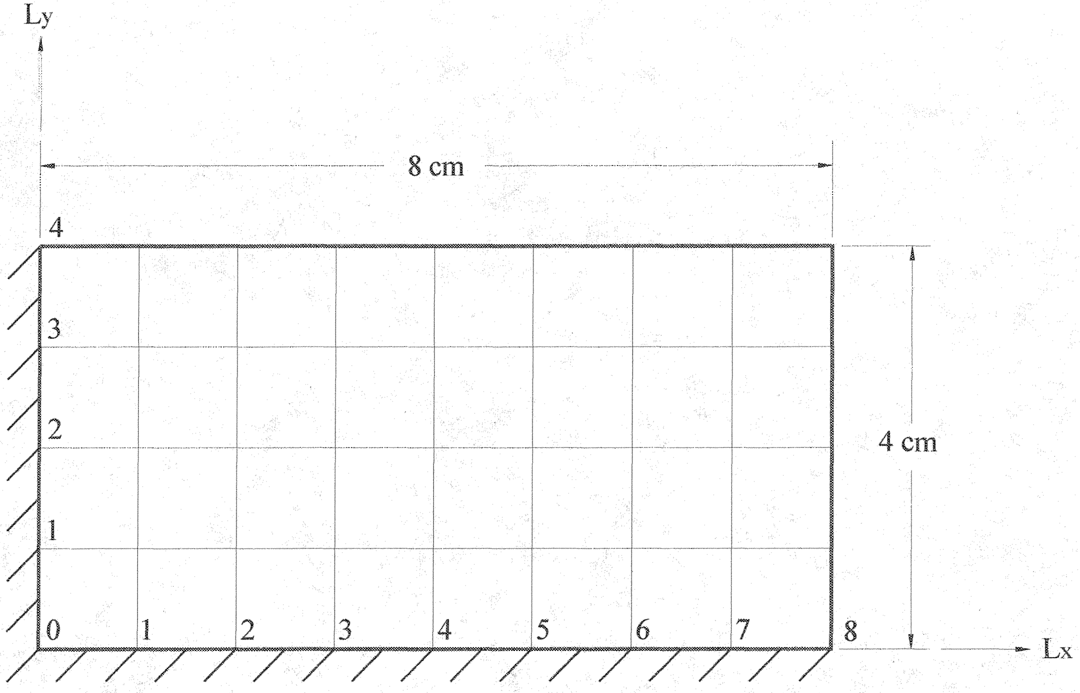

FIGURE 12. Geometry and boundary conditions for the internal delamination.

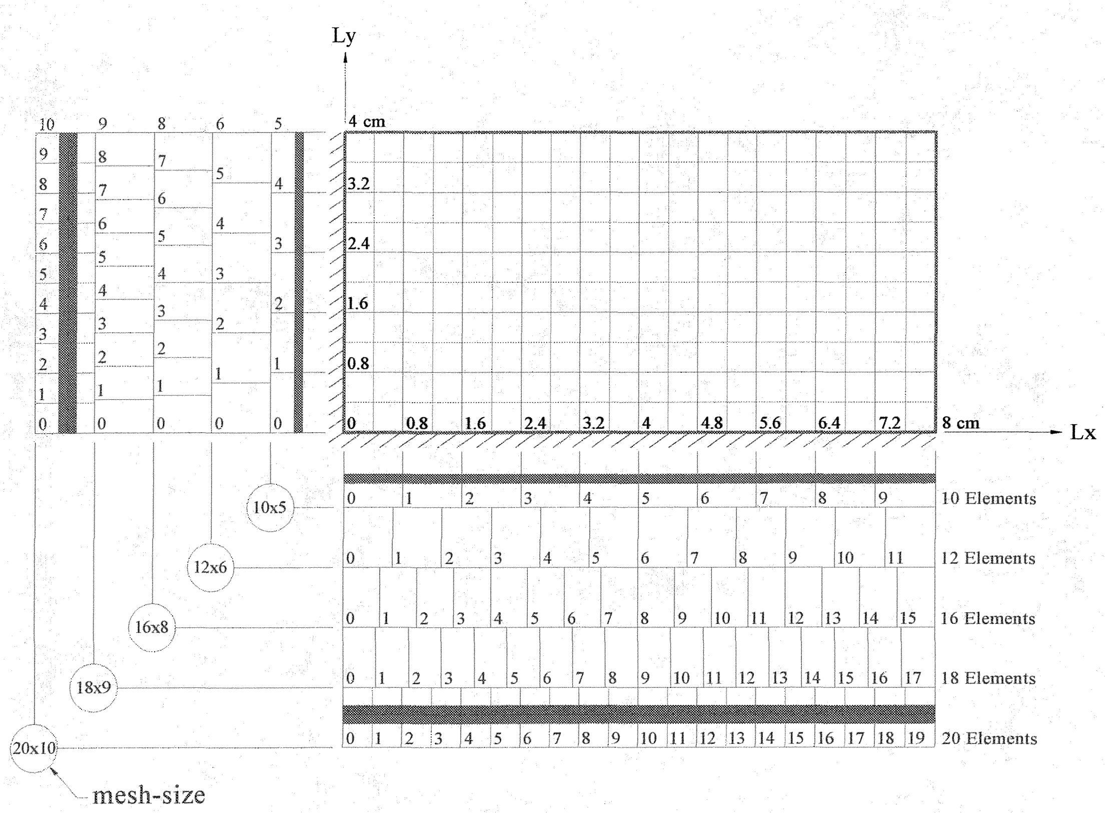

FIGURE 13. A laminate plate of dimension 8cm x 4cm with five different mesh sizes: 10x5, 12x6, 16x8, 18x9, and 20x10.

To choose the best set of training and validation data sets, five different finite element mesh sizes have been used to calculate the natural frequencies of the 8 cm long by 4 cm wide laminate plate as shown in Figure 13. The finite element mesh size, the percentage of delamination, and the number of data sets are shown in Table 5. The mesh size 20x10 was used for training data sets, while the other four mesh sizes (18x9, 16x8, 12x6, 10x5) were used for validation data sets. After being validated with the same trained CPNN, different validation mesh sizes show different values of errors. As shown in Table 6, mesh size 18x9 has the smallest value of the overall relative mean error. Therefore, mesh size 18x9 was chosen as the validation mesh for mesh size 20x10 and as the training mesh of neural network in the present study.

TABLE 5. FEA Data Sets Training Time for Plate Dimension 8cm x 4cm

Mesh Size	Percentage of Delamination	Data Sets	Training Time
10x5	20%	12	49 sec
	60%	504	39 min
	100%	3,300	5 hrs 07 min
12x6	100%	6,552	25 hrs 10 min
16x8	100%	19,584	14 days 17 hrs 37 min
18x9	100%	30,780	40 days 17 hrs 19 min
20x10	20%	120	5 hrs 09 min
	40%	1,440	2 days 17 hrs 31 min
	60%	6,552	13 days 9 hrs 15 min
	80%	19,584	43 days 16 hrs 25 min
	100%	46,200	114 days 2 hrs 25 min

TABLE 6. Trained CPNN Errors Between Different Validation Mesh Sizes

Validation mesh size (100%)	10x5	12x6	16x8	18x9
Validation data sets (100%)	3,300	6,552	19,584	30,780
Overall relative mean error	0.0529	0.0294	0.0086	0.0039

Note: A trained CPNN with mesh size 20x10 (100%) has 46,200 data sets. After being validated with different validation mesh sizes, the values of overall relative mean error (rmeanALL) decreased as the mesh size increased.

Neural Networks

Various CPNNs have been tested and trained using different parameter settings. To find the best CPNN, the first ten natural frequencies have been adopted (i.e., $n_m = 10$). The natural frequencies of these delamination patterns calculated by finite element analysis were compared with those simulated by the trained CPNN. An overall relative mean error (referred to MATLAB codes as rmeanALL) has been used to indicate the accuracy of the trained CPNN and is defined as follows:

$$\varepsilon = \frac{\sum_{k=1}^{n_v}\sum_{j=1}^{n_m}\left|\frac{\omega_{kj,\mathrm{FEM}} - \omega_{kj,\mathrm{CPNN}}}{\omega_{kj,\mathrm{FEM}}}\right|}{n_v \times n_m} \qquad (134)$$

where the subscripts k and j denote the j-th natural frequency of the k-th delamination pattern, n_m is the number of output frequencies used to detect the delamination and n_v is the number of validation data sets. From equation (134), the value of overall relative mean error is disproportional to the number of training data sets, but inversely proportional to the number of validation data sets. Thus, the mesh size of validation data sets that are coupled with the mesh size of training data sets will affect the performance of the trained neural network, which is indicated by the value of overall relative mean error, as shown in Table 6. The MATLAB codes were developed by Dr. H. P. Chen and used in the present study to construct a CPNN for function approximation.

CPNN Parameters Selection

Network Resolution (δr) and Size of Neighborhood (referred to MATLAB codes as interpolationSize) are two parameters which play an important role in the performance of CPNN. The value of overall relative mean error (rmeanALL) and total number of neurons in layer 1 (nuL1, or size of the network) in the neuron networks can be used as indicators to select the suitable parameters. First, gradually decrease the value of δr to decrease the value of overall relative mean error (rmeanALL) and stop when the total number of neurons in layer 1 (nuL1) is equal to the number of training data sets (46,200). Then further decrease δr until rmeanALL reach the minimum. Table 7 shows that different network resolution (δr) settings provide a variety of total number of neurons in layer 1 (nuL1) and overall relative mean error (rmeanALL). In Table 7, $\delta r = 0.003$ is selected.

TABLE 7. Selection of Control Parameter, δr

Run	δr	nuL1	rmeanALL	rrmeanALL	Training Time
1	0.08	134	0.0187	0.0232	27.91 sec
2	0.04	1,041	0.0123	0.0157	181.87 sec
3	0.02	5,987	0.0089	0.0114	991.14 sec
4	0.01	18,033	0.0074	0.0095	3,111.61 sec
5	0.008	22,914	0.0070	0.0091	3,408.91 sec
6	0.006	29,483	0.0065	0.0086	4,350.83 sec
7	0.004	38,672	0.0061	0.0081	5,308.97 sec
8	0.003	46,200	0.0057	0.0076	6,271.22 sec
9	0.002	46,200	0.0057	0.0076	6,279.53 sec

Note: The value of preset parameters are: interpolationSize = 0.04, and rInt = 1.

TABLE 8. Selection of Size of Neighborhood (interpolationSize)

Run	interpolationSize	rmeanALL	rrmeanALL	Training Time
0	0.04	0.0057	0.0076	6,271.22 sec
1	0.03	0.0048	0.0064	6,290.24 sec
2	0.02	0.0040	0.0054	6,222.02 sec
3	0.01	0.0039	0.0052	6,248.31 sec
4	0.009	0.0040	0.0053	6,250.68 sec
5	0.008	0.0041	0.0055	6,249.37 sec

After control parameter δr has been determined, decreasing the value of size of neighborhood (interpolationSize) will decrease the value of overall relative mean error (rmeanALL). Further decreasing the value of size of neighborhood will allow overall relative mean error to reach a minimum value, as shown in Table 8. In Table 8, interpolationSize = 0.01 is selected. Thus, the combination of network resolution (δr) equal to 0.003 and size of neighborhood (interpolationSize) equal to 0.01 will produce a CPNN with an overall relative mean error equal to 0.0039. The summary of the best set of parameters is shown in Table 9.

TABLE 9. Summary of Best Set of Parameters

	Training	Validation
Mesh size	20x10 (100%)	18x9 (100%)
Samples (data sets)	46,200	30,780
nuL1	46,200	30,780
rInt	1	1
δr	0.003	0.003
interpolationSize	0.01	0.01
rmeanALL	0.0039	0.0039
rrmeansALL	0.0052	0.0052

Genetic Algorithms

A tailored genetic algorithm for the detection of single internal delamination has been developed. This GA consists of the tournament selection operator with size of 3, two genetic operators (crossover and mutation) with associated probabilities, population size, and termination criterion. In addition, an elitist strategy, in which the best individual found so far will survive and be selected by the next generation, is adopted in the current GA. Since there is no constraint to be satisfied, the objective function f of equation (120) is taken as the fitness function in the genetic algorithm.

GA Paramenters

A genetic algorithm has been developed to suit the present delamination detection problem. Several numerical investigations have been conducted to determine the best GA parameters setting and their combination in the previous research studies by Hieu The Le and Jin-Hwan Kim [7, 8]. It is believed that not only each parameter setting but also a combination of parameters has a strong effect on GA performance. Since many variables should be considered in GA parametric study, thus GA parametric study is complicated and time-consuming. GA results are satisfactory when the tournament selection is adopted as the selection operator in which a set of three individuals is picked at random and the best individual in this set will be selected by the mating group. For genetic operators, one-point crossover with probability $p_c = 0.5$ and uniform mutation with probability $p_m = 0.02$ are adopted. The population size is chosen to be 200 and the total number of generations before termination is set to 100. In general, uniform mutation performs better than non-uniform mutation and 1-point crossover performs better than 2-points crossover.

Results

GAs results and errors are listed from Tables 10 to 21; even numbered tables are for results and odd numbered tables are for errors. Their natural frequencies are simulated using the best trained CPNN mentioned above. Ten runs using the GA have been performed for each delamination pattern. The actual delamination pattern corresponding to the given frequencies, the mean predicted delamination pattern, and the best predicted delamination pattern among the 10 runs are listed in the even numbered tables. In Tables 10 and 12, eighteen delamination patterns at the center of the plate are considered and to be detected for layer 0 and layer 3, respectively. It is observed that cases 1 to 7, which are considered as small size delaminations (less than 17% of the plate area), are difficult to be detected.

TABLE 10. Mean and Best Predicted Delamination Patterns in Central Area of Plate at Layer 0

Case	Actual Delamination Patterns $\lfloor z_d, x_d, y_d, a_x, a_y \rfloor_{\text{actual}}$	Mean Predicted Delamination Patterns $\lfloor z_d, x_d, y_d, a_x, a_y \rfloor_{\text{mean}}$	Best Predicted Delamination Patterns $\lfloor z_d, x_d, y_d, a_x, a_y \rfloor_{\text{best}}$
1	[0, 3.56, 1.33, 0.89, 1.33]	[2.0, 6.13, 1.73, 0.75, 0.98]	[2, 3.59, 1.60, 0.66, 1.30]
2	[0, 3.56, 0.89, 0.89, 2.22]	[1.4, 4.66, 1.13, 2.16, 1.41]	[0, 3.53, 1.48, 1.04, 1.25]
3	[0, 3.56, 0.44, 0.89, 3.11]	[1.7, 3.70, 0.60, 1.93, 2.20]	[1, 3.64, 0.12, 0.66, 3.79]
4	[0, 3.11, 1.33, 1.78, 1.33]	[1.2, 4.98, 1.08, 2.02, 1.72]	[1, 3.23, 1.06, 1.71, 1.43]
5	[0, 3.11, 0.89, 1.78, 2.22]	[1.1, 4.51, 1.18, 4.94, 1.73]	[1, 3.16, 0.80, 1.80, 2.32]
6	[0, 3.11, 0.44, 1.78, 3.11]	[1.1, 3.30, 0.38, 4.59, 2.22]	[0, 3.26, 0.54, 1.67, 3.35]
7	[0, 2.67, 1.33, 2.67, 1.33]	[0.9, 5.53, 1.03, 2.89, 1.69]	[1, 2.88, 1.24, 2.47, 1.43]
8	[0, 2.67, 0.89, 2.67, 2.22]	[0.7, 3.46, 0.82, 3.69, 2.15]	[0, 2.31, 0.67, 2.53, 2.27]
9	[0, 2.67, 0.44, 2.67, 3.11]	[1.0, 3.58, 0.98, 4.86, 2.91]	[0, 3.26, 0.26, 2.26, 3.42]
10	[0, 2.22, 1.33, 3.56, 1.33]	[1.3, 3.29, 0.98, 3.02, 1.99]	[0, 2.55, 1.32, 3.11, 1.41]
11	[0, 2.22, 0.89, 3.56, 2.22]	[0.9, 3.39, 0.92, 4.36, 2.54]	[0, 2.13, 0.91, 3.40, 2.25]
12	[0, 2.22, 0.44, 3.56, 3.11]	[0.2, 4.30, 1.89, 4.93, 3.67]	[0, 2.56, 0.53, 3.36, 3.33]
13	[0, 1.78, 1.33, 4.44, 1.33]	[0.5, 3.55, 1.19, 3.67, 1.61]	[0, 1.34, 1.39, 3.52, 1.45]
14	[0, 1.78, 0.89, 4.44, 2.22]	[0.3, 3.67, 1.13, 6.17, 2.16]	[0, 2.12, 1.08, 5.49, 1.90]
15	[0, 1.78, 0.44, 4.44, 3.11]	[0.0, 4.05, 1.38, 5.45, 2.93]	[0, 4.17, 1.07, 5.60, 2.97]
16	[0, 1.33, 1.33, 5.33, 1.33]	[1.1, 4.22, 1.18, 5.07, 1.72]	[2, 1.13, 1.37, 5.50, 1.75]
17	[0, 1.33, 0.89, 5.33, 2.22]	[0.1, 1.42, 0.99, 6.44, 2.12]	[0, 1.20, 0.98, 5.54, 2.19]
18	[0, 1.33, 0.44, 5.33, 3.11]	[0.2, 3.29, 0.97, 6.77, 3.21]	[0, 1.47, 0.34, 5.05, 3.07]

TABLE 11. Detection Errors in Central Area of Plate at Layer 0

Case	Finite Element Pattern Size	Actual Ply Location z_{actual}	Mean Predicted Ply Location z_{mean}	Best Predicted Ply Location z_{best}	Mean RRMSE	Best RRMSE	Elapsed Time (seconds)
1	2x3	0	2.0	2	0.00168	0.00152	1944.06
2	2x5	0	1.4	0	0.00224	0.00186	1873.40
3	2x7	0	1.7	1	0.00242	0.00125	1812.77
4	4x3	0	1.2	1	0.00587	0.00368	1761.47
5	4x5	0	1.1	1	0.01382	0.00668	1707.13
6	4x7	0	1.1	0	0.01412	0.00392	1844.51
7	6x3	0	0.9	1	0.00705	0.00406	1799.91
8	6x5	0	0.7	0	0.01380	0.00699	1812.00
9	6x7	0	1.0	0	0.01893	0.00753	1768.68
10	8x3	0	1.3	0	0.00989	0.00370	1860.71
11	8x5	0	0.9	0	0.01703	0.00642	1672.56
12	8x7	0	0.2	0	0.02075	0.00711	1736.58
13	10x3	0	0.5	0	0.00782	0.00689	1687.55
14	10x5	0	0.3	0	0.01259	0.00766	1683.43
15	10x7	0	0.0	0	0.01805	0.01638	1701.27
16	12x3	0	1.1	2	0.00710	0.00821	1678.29
17	12x5	0	0.1	0	0.01120	0.00703	1679.43
18	12x7	0	0.2	0	0.01592	0.00584	1684.39

Comparison between Tables 10 and 12 reveals that the small size delaminations are even more difficult to be detected precisely at near surface ($z_d = 3$, where the frequency of near surface delamination is close to that of healthy one). For all other cases, results are satisfactory. An exception is when a delamination with length to width ratio is close to 4, the detection is normally off by two layers, or difficult to be detected precisely, as in case

case 16 in Table 10.

TABLE 12. Mean and Best Predicted Delamination Patterns in Central Area of Plate at Layer 3

Case	Actual Delamination Patterns $\lfloor z_d, x_d, y_d, a_x, a_y \rfloor_{\text{actual}}$	Mean Predicted Delamination Patterns $\lfloor z_d, x_d, y_d, a_x, a_y \rfloor_{\text{mean}}$	Best Predicted Delamination Patterns $\lfloor z_d, x_d, y_d, a_x, a_y \rfloor_{\text{best}}$
1	[3, 3.56, 1.33, 0.89, 1.33]	[1.4, 3.01, 1.39, 3.37, 0.48]	[3, 5.39, 0.84, 2.87, 0.48]
2	[3, 3.56, 0.89, 0.89, 2.22]	[1.4, 3.01, 1.39, 3.37, 0.48]	[3, 5.39, 0.84, 2.87, 0.48]
3	[3, 3.56, 0.44, 0.89, 3.11]	[1.4, 3.51, 1.21, 3.42, 0.48]	[3, 5.39, 0.84, 2.87, 0.48]
4	[3, 3.11, 1.33, 1.78, 1.33]	[1.3, 3.01, 1.45, 3.21, 0.48]	[3, 5.39, 0.84, 2.87, 0.48]
5	[3, 3.11, 0.89, 1.78, 2.22]	[2.6, 0.83, 0.70, 1.57, 1.47]	[3, 2.72, 1.07, 2.40, 0.93]
6	[3, 3.11, 0.44, 1.78, 3.11]	[2.5, 2.30, 2.05, 1.30, 3.30]	[3, 3.03, 2.52, 1.64, 3.15]
7	[3, 2.67, 1.33, 2.67, 1.33]	[2.7, 0.70, 1.06, 2.31, 1.27]	[3, 0.35, 1.03, 2.00, 1.38]
8	[3, 2.67, 0.89, 2.67, 2.22]	[2.5, 2.80, 0.74, 3.65, 2.16]	[3, 2.46, 0.94, 2.78, 1.94]
9	[3, 2.67, 0.44, 2.67, 3.11]	[2.7, 2.66, 0.72, 2.66, 2.92]	[3, 2.59, 0.53, 2.39, 3.14]
10	[3, 2.22, 1.33, 3.56, 1.33]	[3.0, 1.84, 1.14, 4.43, 1.18]	[3, 1.90, 1.32, 3.55, 1.17]
11	[3, 2.22, 0.89, 3.56, 2.22]	[2.7, 2.18, 0.78, 3.69, 2.42]	[3, 2.43, 0.91, 3.21, 2.45]
12	[3, 2.22, 0.44, 3.56, 3.11]	[2.5, 2.01, 1.34, 3.48, 2.79]	[3, 2.40, 0.49, 3.37, 3.07]
13	[3, 1.78, 1.33, 4.44, 1.33]	[3.0, 2.64, 1.09, 4.69, 1.45]	[3, 1.94, 1.37, 3.91, 1.38]
14	[3, 1.78, 0.89, 4.44, 2.22]	[3.0, 2.42, 0.84, 5.26, 2.26]	[3, 1.78, 1.01, 4.27, 2.16]
15	[3, 1.78, 0.44, 4.44, 3.11]	[3.0, 3.26, 0.56, 5.35, 3.34]	[3, 1.50, 0.53, 4.66, 3.10]
16	[3, 1.33, 1.33, 5.33, 1.33]	[3.0, 2.36, 1.07, 4.34, 1.64]	[3, 1.27, 1.35, 4.81, 1.37]
17	[3, 1.33, 0.89, 5.33, 2.22]	[3.0, 3.26, 0.60, 5.80, 2.67]	[3, 1.56, 0.67, 5.01, 2.36]
18	[3, 1.33, 0.44, 5.33, 3.11]	[3.0, 2.90, 0.73, 6.33, 2.96]	[3, 1.30, 0.76, 5.25, 2.71]

TABLE 13. Detection Errors in Central Area of Plate at Layer 3

Case	Finite Element Pattern Size	Actual Ply Location z_{actual}	Mean Predicted Ply Location z_{mean}	Best Predicted Ply Location z_{best}	Mean RRMSE	Best RRMSE	Elapsed Time (seconds)
1	2x3	3	1.4	3	0.00337	0.00337	1706.55
2	2x5	3	1.4	3	0.00324	0.00324	1626.47
3	2x7	3	1.4	3	0.00312	0.00312	1786.53
4	4x3	3	1.3	3	0.00331	0.00331	1701.79
5	4x5	3	2.6	3	0.00367	0.00336	1702.90
6	4x7	3	2.5	3	0.00420	0.00396	1694.42
7	6x3	3	2.7	3	0.00389	0.00359	1729.97
8	6x5	3	2.5	3	0.00597	0.00433	1705.77
9	6x7	3	2.7	3	0.00663	0.00317	1670.35
10	8x3	3	3.0	3	0.00429	0.00377	1624.48
11	8x5	3	2.7	3	0.00575	0.00427	1727.39
12	8x7	3	2.5	3	0.00972	0.00426	1740.69
13	10x3	3	3.0	3	0.00478	0.00400	1703.07
14	10x5	3	3.0	3	0.00778	0.00358	1655.82
15	10x7	3	3.0	3	0.00710	0.00439	1739.12
16	12x3	3	3.0	3	0.00465	0.00376	1640.24
17	12x5	3	3.0	3	0.00567	0.00402	1685.57
18	12x7	3	3.0	3	0.00600	0.00331	1666.48

In Tables 14 and 16, eighteen small size delamination patterns (less than 17% of the plate area) are considered and to be detected for layer 0 and layer 3, respectively. Cases 1 to 6 are at the center of the plate, cases 7 to 12 are at the left-center (one element away from the left boundary), and cases 13 to 18 are at the right-center (one element away from the free edge). After comparing between center, left, and right, it is observed that the delamination patterns on the left side are more difficult to be detected. Five out of six cases are off by three layers because of the effect by the left side boundary, as shown in Table 14. Similarly, Table 16 shows that the best predictions on the left side are less precise than the right side.

TABLE 14. Mean and Best Predicted for Small Size Delamination Patterns at Layer 0

Case	Actual Delamination Patterns $\lfloor z_d, x_d, y_d, a_x, a_y \rfloor_{\text{actual}}$	Mean Predicted Delamination Patterns $\lfloor z_d, x_d, y_d, a_x, a_y \rfloor_{\text{mean}}$	Best Predicted Delamination Patterns $\lfloor z_d, x_d, y_d, a_x, a_y \rfloor_{\text{best}}$
1	[0, 3.56, 1.33, 0.89, 1.33]	[2.0, 6.13, 1.73, 0.75, 0.98]	[2, 3.59, 1.60, 0.66, 1.30]
2	[0, 3.56, 0.89, 0.89, 2.22]	[1.4, 4.66, 1.13, 2.16, 1.41]	[0, 3.53, 1.48, 1.04, 1.25]
3	[0, 3.56, 0.44, 0.89, 3.11]	[1.7, 3.40, 0.60, 1.93, 2.20]	[1, 3.64, 0.12, 0.66, 3.79]
4	[0, 3.11, 1.33, 1.78, 1.33]	[1.2, 4.98, 1.08, 2.02, 1.72]	[1, 3.23, 1.06, 1.71, 1.43]
5	[0, 3.11, 0.89, 1.78, 2.22]	[1.1, 4.51, 1.18, 4.94, 1.73]	[1, 3.16, 0.80, 1.80, 2.32]
6	[0, 3.11, 0.44, 1.78, 3.11]	[1.1, 3.30, 0.38, 4.59, 2.22]	[0, 3.26, 0.54, 1.67, 3.35]
7	[0, 0.44, 1.33, 0.89, 1.33]	[3.0, 1.47, 1.14, 1.64, 0.78]	[3, 0.36, 1.26, 0.81, 1.29]
8	[0, 0.44, 0.89, 0.89, 2.22]	[3.0, 0.09, 1.21, 1.86, 1.64]	[3, 0.08, 0.52, 1.07, 2.26]
9	[0, 0.44, 0.44, 0.89, 3.11]	[3.0, 0.47, 1.74, 3.00, 1.67]	[3, 0.34, 1.07, 1.19, 3.14]
10	[0, 0.44, 1.33, 1.33, 1.33]	[2.8, 0.07, 1.04, 2.25, 1.30]	[3, 0.06, 1.31, 1.36, 1.38]
11	[0, 0.44, 0.89, 1.33, 2.22]	[2.7, 0.90, 0.88, 3.21, 2.06]	[3, 0.15, 0.94, 2.95, 2.11]
12	[0, 0.44, 0.44, 1.33, 3.11]	[2.3, 0.80, 0.69, 2.24, 3.09]	[0, 0.47, 0.63, 1.26, 2.98]
13	[0, 6.67, 1.33, 0.89, 1.33]	[1.6, 5.64, 1.42, 1.13, 1.60]	[0, 5.98, 1.44, 0.93, 1.16]
14	[0, 6.67, 0.89, 0.89, 2.22]	[1.4, 5.51, 1.29, 1.52, 1.78]	[1, 6.12, 0.68, 0.92, 2.51]
15	[0, 6.67, 0.44, 0.89, 3.11]	[1.6, 5.87, 1.78, 1.27, 2.97]	[0, 6.70, 0.51, 0.76, 3.31]
16	[0, 6.22, 1.33, 1.33, 1.33]	[1.2, 5.80, 1.13, 1.29, 2.05]	[0, 5.49, 1.32, 1.66, 1.11]
17	[0, 6.22, 0.89, 1.33, 2.22]	[1.0, 5.78, 1.01, 1.68, 1.98]	[0, 5.93, 0.92, 1.43, 2.07]
18	[0, 6.22, 0.44, 1.33, 3.11]	[0.4, 6.60, 0.65, 1.47, 3.19]	[0, 6.51, 0.25, 1.08, 3.37]

TABLE 15. Detection Errors for Small Size Delamination Patterns at Layer 0

Case	Finite Element Pattern Size	Actual Ply Location z_{actual}	Mean Predicted Ply Location z_{mean}	Best Predicted Ply Location z_{best}	Mean RRMSE	Best RRMSE	Elapsed Time (seconds)
1	2x3	0	2.0	2	0.00168	0.00152	1944.06
2	2x5	0	1.4	0	0.00224	0.00186	1873.40
3	2x7	0	1.7	1	0.00242	0.00125	1812.77
4	4x3	0	1.2	1	0.00587	0.00368	1761.47
5	4x5	0	1.1	1	0.01382	0.00668	1707.13
6	4x7	0	1.1	0	0.01412	0.00392	1844.51
7	2x3	0	3.0	3	0.00424	0.00417	1724.05
8	2x5	0	3.0	3	0.00407	0.00402	1601.38
9	2x7	0	3.0	3	0.00438	0.00386	1796.30
10	3x3	0	2.8	3	0.00355	0.00318	1598.47
11	3x5	0	2.7	3	0.00428	0.00399	1681.72
12	3x7	0	2.3	0	0.00536	0.00303	1618.74
13	2x3	0	1.6	0	0.00188	0.00193	1764.89
14	2x5	0	1.4	1	0.00205	0.00257	1729.65
15	2x7	0	1.6	0	0.00288	0.00281	1764.54
16	3x3	0	1.2	0	0.00340	0.00249	1771.05
17	3x5	0	1.0	0	0.00477	0.00345	1780.97
18	3x7	0	0.4	0	0.00535	0.00475	1643.62

TABLE 16. Mean and Best Predicted for Small Size Delamination Patterns at Layer 3

Case	Actual Delamination Patterns $\lfloor z_d, x_d, y_d, a_x, a_y \rfloor_{\text{actual}}$	Mean Predicted Delamination Patterns $\lfloor z_d, x_d, y_d, a_x, a_y \rfloor_{\text{mean}}$	Best Predicted Delamination Patterns $\lfloor z_d, x_d, y_d, a_x, a_y \rfloor_{\text{best}}$
1	[3, 3.56, 1.33, 0.89, 1.33]	[1.4, 3.01, 1.39, 3.37, 0.48]	[3, 5.39, 0.84, 2.87, 0.48]
2	[3, 3.56, 0.89, 0.89, 2.22]	[1.4, 3.01, 1.39, 3.37, 0.48]	[3, 5.39, 0.84, 2.87, 0.48]
3	[3, 3.56, 0.44, 0.89, 3.11]	[1.4, 3.51, 1.21, 3.42, 0.48]	[3, 5.39, 0.84, 2.87, 0.48]
4	[3, 3.11, 1.33, 1.78, 1.33]	[1.3, 3.01, 1.45, 3.21, 0.48]	[3, 5.39, 0.84, 2.87, 0.48]
5	[3, 3.11, 0.89, 1.78, 2.22]	[2.6, 0.83, 0.70, 1.57, 1.47]	[3, 2.72, 1.07, 2.40, 0.93]
6	[3, 3.11, 0.44, 1.78, 3.11]	[2.5, 2.30, 2.05, 1.30, 3.30]	[3, 3.03, 2.52, 1.64, 3.15]
7	[3, 0.44, 1.33, 0.89, 1.33]	[1.3, 2.46, 1.23, 3.40, 0.47]	[2, 0.79, 1.62, 3.61, 0.45]
8	[3, 0.44, 0.89, 0.89, 2.22]	[1.5, 2.70, 0.99, 3.77, 0.47]	[2, 0.79, 1.62, 3.61, 0.45]
9	[3, 0.44, 0.44, 0.89, 3.11]	[1.5, 2.58, 0.87, 3.97, 0.48]	[2, 0.40, 1.27, 3.18, 0.50]
10	[3, 0.44, 1.33, 1.33, 1.33]	[1.5, 2.70, 0.99, 3.77, 0.47]	[2, 0.40, 1.27, 3.18, 0.50]
11	[3, 0.44, 0.89, 1.33, 2.22]	[3.0, 0.50, 1.32, 0.87, 0.81]	[3, 0.38, 0.99, 0.65, 1.34]
12	[3, 0.44, 0.44, 1.33, 3.11]	[3.0, 0.38, 2.79, 1.12, 1.76]	[3, 0.65, 1.56, 0.90, 2.71]
13	[3, 6.67, 1.33, 0.89, 1.33]	[1.8, 6.16, 1.79, 1.25, 0.48]	[3, 7.22, 1.23, 0.85, 0.51]
14	[3, 6.67, 0.89, 0.89, 2.22]	[1.7, 6.16, 1.79, 1.24, 0.48]	[3, 7.22, 1.23, 0.85, 0.51]
15	[3, 6.67, 0.44, 0.89, 3.11]	[1.7, 6.16, 1.79, 1.24, 0.48]	[0, 6.40, 0.51, 1.20, 0.49]
16	[3, 6.22, 1.33, 1.33, 1.33]	[1.8, 5.96, 1.55, 1.28, 0.48]	[3, 4.00, 1.09, 1.63, 0.48]
17	[3, 6.22, 0.89, 1.33, 2.22]	[1.8, 6.16, 1.78, 1.25, 0.48]	[2, 7.25, 1.16, 0.94, 0.46]
18	[3, 6.22, 0.44, 1.33, 3.11]	[1.8, 5.84, 1.77, 1.31, 0.49]	[0, 6.40, 0.51, 1.20, 0.49]

TABLE 17. Detection Errors for Small Size Delamination Patterns at Layer 3

Case	Finite Element Pattern Size	Actual Ply Location z_{actual}	Mean Predicted Ply Location z_{mean}	Best Predicted Ply Location z_{best}	Mean RRMSE	Best RRMSE	Elapsed Time (seconds)
1	2x3	3	1.4	3	0.00337	0.00337	1706.55
2	2x5	3	1.4	3	0.00324	0.00324	1626.47
3	2x7	3	1.4	3	0.00312	0.00312	1786.53
4	4x3	3	1.3	3	0.00331	0.00331	1701.79
5	4x5	3	2.6	3	0.00367	0.00336	1702.90
6	4x7	3	2.5	3	0.00420	0.00396	1694.42
7	2x3	3	1.3	2	0.00424	0.00417	1724.05
8	2x5	3	1.5	2	0.00407	0.00402	1601.38
9	2x7	3	1.5	2	0.00438	0.00386	1796.30
10	3x3	3	1.5	2	0.00355	0.00318	1598.47
11	3x5	3	3.0	3	0.00428	0.00399	1681.72
12	3x7	3	3.0	3	0.00536	0.00303	1619.74
13	2x3	3	1.8	3	0.00327	0.00327	1707.88
14	2x5	3	1.7	3	0.00313	0.00313	1678.20
15	2x7	3	1.7	0	0.00305	0.00305	1692.44
16	3x3	3	1.8	3	0.00298	0.00298	1674.17
17	3x5	3	1.8	2	0.00300	0.00300	1734.19
18	3x7	3	1.8	0	0.00347	0.00347	1732.11

In Table 18, fifteen larger sized delamination patterns (less than 35% of the plate area) are considered and to be detected. Cases 1 to 5 are at the center of the plate, cases 6 to 10 are at the left-center, and cases 11 to 15 are at the right-center. It is observed that the detections are much easier detected. As the best predicted delamination patterns in Table 18 shows, only case 15 is off by one layer; all other cases are satisfactory.

TABLE 18. Mean and Best Predicted for Larger Sized Delamination Patterns at Layer 0

Case	Actual Delamination Patterns $\lfloor z_d, x_d, y_d, a_x, a_y \rfloor_{\text{actual}}$	Mean Predicted Delamination Patterns $\lfloor z_d, x_d, y_d, a_x, a_y \rfloor_{\text{mean}}$	Best Predicted Delamination Patterns $\lfloor z_d, x_d, y_d, a_x, a_y \rfloor_{\text{best}}$
1	[0, 2.67, 0.89, 2.67, 2.22]	[0.7, 3.46, 0.82, 3.69, 2.15]	[0, 2.31, 0.67, 2.53, 2.27]
2	[0, 2.67, 0.44, 2.67, 3.11]	[1.0, 3.58, 0.98, 4.86, 2.91]	[0, 3.26, 0.26, 2.26, 3.42]
3	[0, 2.22, 0.89, 3.56, 2.22]	[0.9, 3.39, 0.92, 4.36, 2.54]	[0, 2.13, 0.91, 3.40, 2.25]
4	[0, 2.22, 0.44, 3.56, 3.11]	[0.2, 4.30, 1.89, 4.93, 3.67]	[0, 2.56, 0.53, 3.36, 3.33]
5	[0, 1.78, 1.33, 4.44, 1.33]	[0.5, 3.55, 1.19, 3.67, 1.61]	[0, 1.34, 1.39, 3.52, 1.45]
6	[0, 0.44, 0.44, 2.22, 3.11]	[1.4, 2.06, 2.11, 3.34, 3.48]	[0, 0.57, 0.54, 2.12, 3.07]
7	[0, 0.44, 0.89, 2.67, 2.22]	[0.6, 0.81, 1.03, 3.02, 2.06]	[0, 0.53, 0.95, 2.69, 2.18]
8	[0, 0.44, 0.44, 2.67, 3.11]	[1.2, 0.46, 1.25, 3.55, 2.95]	[0, 0.54, 0.31, 2.52, 3.30]
9	[0, 0.44, 0.89, 3.11, 2.22]	[0.6, 1.05, 0.90, 3.29, 2.36]	[0, 0.47, 1.07, 3.44, 1.83]
10	[0, 0.44, 0.44, 3.11, 3.11]	[1.1, 0.52, 0.87, 3.69, 3.42]	[0, 0.45, 0.29, 2.95, 3.29]
11	[0, 5.33, 0.89, 2.22, 2.22]	[1.6, 5.58, 0.96, 3.86, 2.12]	[0, 5.35, 0.70, 2.12, 2.25]
12	[0, 5.33, 0.44, 2.22, 3.11]	[1.5, 5.65, 0.71, 4.19, 2.88]	[0, 5.24, 0.14, 1.86, 3.71]
13	[0, 4.89, 0.89, 2.67, 2.22]	[0.3, 5.34, 1.13, 3.98, 1.93]	[0, 5.03, 1.26, 2.88, 2.11]
14	[0, 4.44, 0.89, 3.11, 2.22]	[0.5, 4.94, 0.94, 3.74, 2.16]	[0, 3.98, 0.95, 3.28, 2.12]
15	[0, 4.44, 0.44, 3.11, 3.11]	[0.5, 5.83, 1.40, 4.57, 3.28]	[1, 4.27, 0.59, 3.84, 2.67]

TABLE 19. Detection Errors for Larger Sized Delamination Patterns at Layer 0

Case	Finite Element Pattern Size	Actual Ply Location z_{actual}	Mean Predicted Ply Location z_{mean}	Best Predicted Ply Location z_{best}	Mean RRMSE	Best RRMSE	Elapsed Time (seconds)
1	6x5	0	0.7	0	0.01380	0.00699	1812.00
2	6x7	0	1.0	0	0.01893	0.00753	1768.68
3	8x5	0	0.9	0	0.01703	0.00642	1672.56
4	8x7	0	0.2	0	0.02075	0.00711	1736.58
5	10x3	0	0.5	0	0.00782	0.00689	1687.55
6	5x7	0	1.4	0	0.01874	0.00344	1599.95
7	6x5	0	0.6	0	0.00836	0.00363	1602.28
8	6x7	0	1.2	0	0.01523	0.00445	1699.36
9	7x5	0	0.6	0	0.01352	0.00968	1667.78
10	7x7	0	1.1	0	0.01572	0.00421	1599.80
11	5x5	0	1.6	0	0.00822	0.00500	1790.31
12	5x7	0	1.5	0	0.01458	0.01374	1615.54
13	6x5	0	0.3	0	0.00693	0.00457	1686.71
14	7x5	0	0.5	0	0.01177	0.00952	1604.16
15	7x7	0	0.5	1	0.01890	0.01189	1750.52

Comprehensive Case Studies

In Tables 20 and 21, twenty-seven delamination patterns of mesh size 18x9 were used to observe the effects of density of training data sets of mesh size 20x10, as shown in Figures 14 and 15. The delamination patterns are divided into four groups. The first group consists of cases 1 to 7 with delamination size 4x3 and their locations are shown in Figures 16 and 17. The second group consists of cases 8 to 15 with delamination sizes of 2x5 and 3x5 and their locations are shown in Figures 18 and 19. The third group consists of cases 16 to 23 with delamination size 3x6 and their locations are shown in Figures 20 and 21. The fourth group consists of cases 24 to 27 with delamination sizes 6x3 and 10x3 and their locations are shown in Figures 22 and 23.

TABLE 20. Mean and Best Predicted Delamination Patterns for Comprehensive Case Studies at Layer 0

Case	Actual Delamination Patterns $\lfloor z_d, x_d, y_d, a_x, a_y \rfloor_{\text{actual}}$	Mean Predicted Delamination Patterns $\lfloor z_d, x_d, y_d, a_x, a_y \rfloor_{\text{mean}}$	Best Predicted Delamination Patterns $\lfloor z_d, x_d, y_d, a_x, a_y \rfloor_{\text{best}}$
1	[0, 3.11, 1.33, 1.78, 1.33]	[1.2, 4.98, 1.08, 2.02, 1.72]	[1, 3.23, 1.06, 1.71, 1.43]
2	[0, 1.78, 1.33, 1.78, 1.33]	[0.5, 3.26, 1.50, 4.16, 0.99]	[0, 1.82, 1.06, 1.75, 1.37]
3	[0, 4.44, 1.33, 1.78, 1.33]	[0.5, 4.68, 1.11, 1.89, 1.37]	[0, 4.56, 1.06, 1.72, 1.37]
4	[0, 0.89, 1.33, 1.78, 1.33]	[1.8, 2.16, 1.39, 4.64, 1.16]	[2, 2.52, 0.97, 1.44, 1.44]
5	[0, 5.33, 1.33, 1.78, 1.33]	[1.8, 5.56, 1.25, 1.47, 1.92]	[2, 5.10, 1.31, 1.71, 1.44]
6	[0, 0.44, 1.33, 1.78, 1.33]	[2.0, 0.69, 0.84, 2.63, 1.86]	[0, 0.40, 1.36, 1.68, 1.13]
7	[0, 5.78, 1.33, 1.78, 1.33]	[1.6, 5.75, 1.28, 1.68, 1.85]	[0, 5.97, 1.32, 1.32, 1.71]
8	[0, 1.78, 0.89, 0.89, 2.22]	[1.2, 1.71, 1.29, 1.55, 1.24]	[1, 1.97, 0.95, 0.76, 1.87]
9	[0, 5.33, 0.89, 0.89, 2.22]	[2.0, 5.29, 1.48, 0.83, 2.49]	[2, 5.27, 1.15, 0.75, 2.25]
10	[0, 1.78, 0.89, 1.33, 2.22]	[0.3, 2.30, 1.27, 3.09, 1.44]	[0, 1.90, 0.64, 1.24, 2.27]
11	[0, 4.89, 0.89, 1.33, 2.22]	[0.7, 4.50, 1.05, 1.56, 2.08]	[0, 4.76, 0.61, 1.23, 2.53]
12	[0, 0.89, 0.89, 1.33, 2.22]	[1.1, 1.37, 0.88, 1.69, 2.88]	[0, 1.13, 0.62, 1.07, 2.68]
13	[0, 5.78, 0.89, 1.33, 2.22]	[1.9, 5.69, 0.95, 1.43, 2.41]	[1, 5.66, 1.02, 1.43, 2.07]
14	[0, 0.44, 0.89, 1.33, 2.22]	[2.7, 0.90, 0.88, 3.21, 2.06]	[3, 0.15, 0.94, 2.95, 2.11]
15	[0, 6.22, 0.89, 1.33, 2.22]	[1.0, 5.78, 1.01, 1.68, 1.98]	[0, 5.96, 0.68, 1.32, 2.28]
16	[0, 0.89, 0.44, 1.33, 2.67]	[1.3, 0.83, 0.42, 3.09, 1.98]	[2, 0.56, 0.30, 1.49, 2.64]
17	[0, 5.78, 0.44, 1.33, 2.67]	[1.5, 5.74, 0.88, 1.40, 3.13]	[0, 5.84, 0.48, 1.20, 2.72]
18	[0, 0.44, 0.44, 1.33, 2.67]	[3.0, 1.65, 0.68, 3.54, 2.09]	[3, 0.07, 0.51, 2.97, 2.56]
19	[0, 6.22, 0.44, 1.33, 2.67]	[0.9, 6.05, 1.01, 1.69, 2.75]	[0, 6.20, 0.55, 1.26, 2.61]
20	[0, 0.89, 0.89, 1.33, 2.67]	[0.8, 0.95, 1.65, 1.34, 3.05]	[1, 0.99, 0.99, 1.24, 2.65]
21	[0, 5.78, 0.89, 1.33, 2.67]	[2.0, 5.59, 1.20, 1.50, 2.87]	[2, 5.41, 1.01, 1.53, 2.82]
22	[0, 0.44, 0.89, 1.33, 2.67]	[1.8, 1.27, 1.30, 2.31, 2.78]	[1, 0.51, 0.61, 1.08, 3.45]
23	[0, 6.22, 0.89, 1.33, 2.67]	[1.0, 6.11, 1.34, 1.62, 2.59]	[0, 6.04, 0.94, 1.34, 2.94]
24	[0, 2.67, 1.33, 2.67, 1.33]	[0.9, 5.53, 1.03, 2.89, 1.69]	[1, 2.88, 1.24, 2.47, 1.43]
25	[0, 0.44, 1.33, 2.67, 1.33]	[0.9, 1.80, 1.20, 2.94, 1.44]	[0, 0.93, 1.40, 3.48, 1.02]
26	[0, 4.89, 1.33, 2.67, 1.33]	[0.5, 6.87, 1.24, 2.40, 1.63]	[0, 5.48, 1.48, 3.05, 1.12]
27	[0, 1.78, 1.33, 4.44, 1.33]	[0.5, 1.19, 1.19, 3.67, 1.61]	[0, 1.34, 1.39, 3.52, 1.45]

TABLE 21. Detection Errors for Comprehensive Case Studies at Layer 0

Case	Finite Element Pattern Size	Actual Ply Location z_{actual}	Mean Predicted Ply Location z_{mean}	Best Predicted Ply Location z_{best}	Mean RRMSE	Best RRMSE	Elapsed Time (seconds)
1	4x3	0	1.2	1	0.00587	0.00368	1761.47
2	4x3	0	0.5	0	0.00568	0.00318	1817.65
3	4x3	0	0.5	0	0.00491	0.00403	1697.61
4	4x3	0	1.8	2	0.00270	0.00241	1648.38
5	4x3	0	1.8	2	0.00391	0.00308	1715.46
6	4x3	0	2.0	0	0.00388	0.00181	1700.85
7	4x3	0	1.6	0	0.00411	0.00286	1790.61
8	2x5	0	1.2	1	0.00139	0.00108	1666.12
9	2x5	0	2.0	2	0.00377	0.00372	1667.98
10	3x5	0	0.3	0	0.00637	0.00416	1688.77
11	3x5	0	0.7	0	0.00625	0.00552	1660.46
12	3x5	0	1.1	0	0.00288	0.00170	1794.51
13	3x5	0	1.9	1	0.00443	0.00366	1710.63
14	3x5	0	2.7	3	0.00428	0.00399	1681.72
15	3x5	0	1.0	0	0.00477	0.00375	1780.97
16	3x6	0	1.3	2	0.00458	0.00388	1624.77
17	3x6	0	1.5	0	0.00458	0.00338	1670.09
18	3x6	0	3.0	3	0.00525	0.00400	1674.26
19	3x6	0	0.9	0	0.00505	0.00350	1641.00
20	3x6	0	0.8	1	0.00315	0.00261	1714.49
21	3x6	0	2.0	2	0.00435	0.00378	1610.03
22	3x6	0	1.8	1	0.00573	0.00269	1851.59
23	3x6	0	1.0	0	0.00551	0.00391	1687.05
24	6x3	0	0.9	1	0.00705	0.00406	1590.10
25	6x3	0	0.9	0	0.00479	0.00305	1601.86
26	6x3	0	0.5	0	0.00374	0.00463	1621.15
27	10x3	0	0.5	0	0.00782	0.00689	1687.55

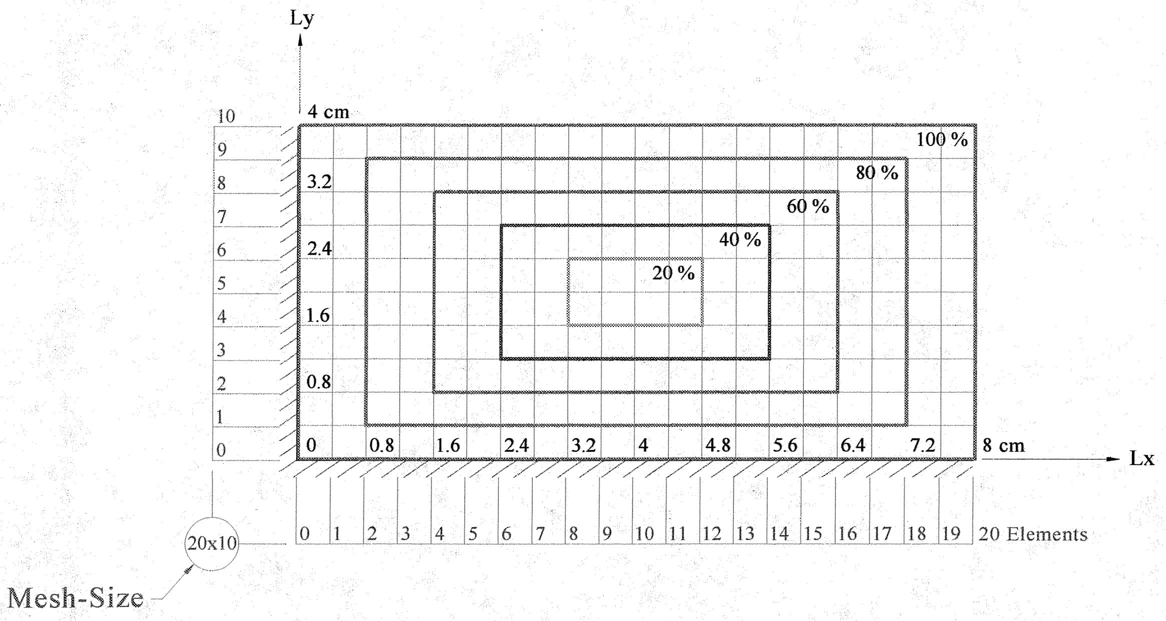

FIGURE 14. Area of delamination for plate dimension 8cm x 4cm with mesh-size 20x10.

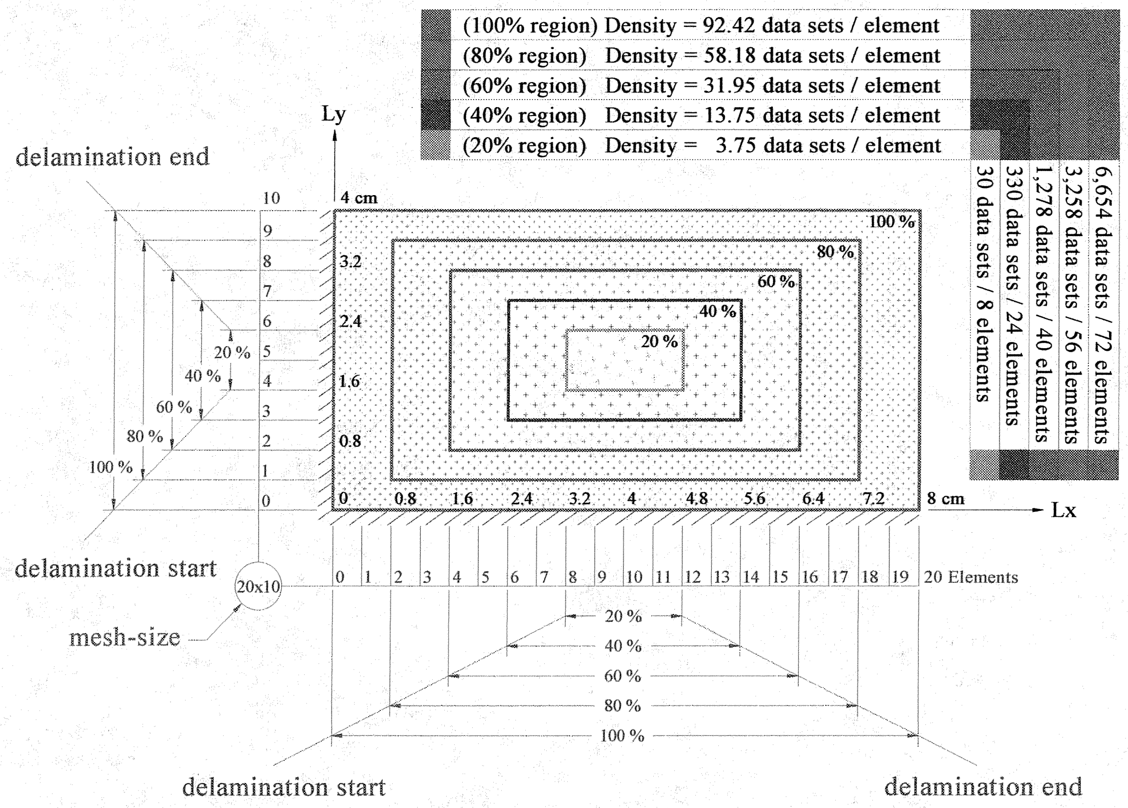
(100% region) Density = 92.42 data sets / element
(80% region) Density = 58.18 data sets / element
(60% region) Density = 31.95 data sets / element
(40% region) Density = 13.75 data sets / element
(20% region) Density = 3.75 data sets / element
6,654 data sets / 72 elements
3,258 data sets / 56 elements
1,278 data sets / 40 elements
330 data sets / 24 elements
30 data sets / 8 elements
Ly
4 cm
delamination end
delamination start
10
9
8
7
6
5
4
3
2
1
0
20 %
40 %
60 %
80 %
100 %
3.2
2.4
1.6
0.8
0
0.8
1.6
2.4
3.2
4
4.8
5.6
6.4
7.2
8 cm
Lx
100 %
80 %
60 %
40 %
20 %
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
20 Elements
20x10
mesh-size
20 %
40 %
60 %
80 %
100 %
delamination start
delamination end

FIGURE 15. Density of data sets for plate dimension 8cm x 4cm with mesh size 20x10.

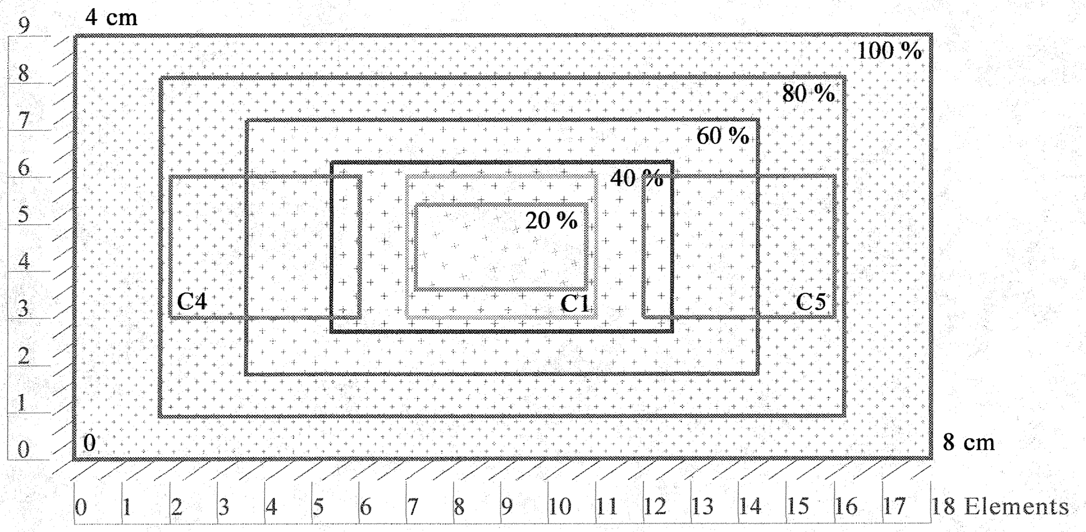

FIGURE 16. Delamination patterns location of cases 1, 4, and 5 (group 1).

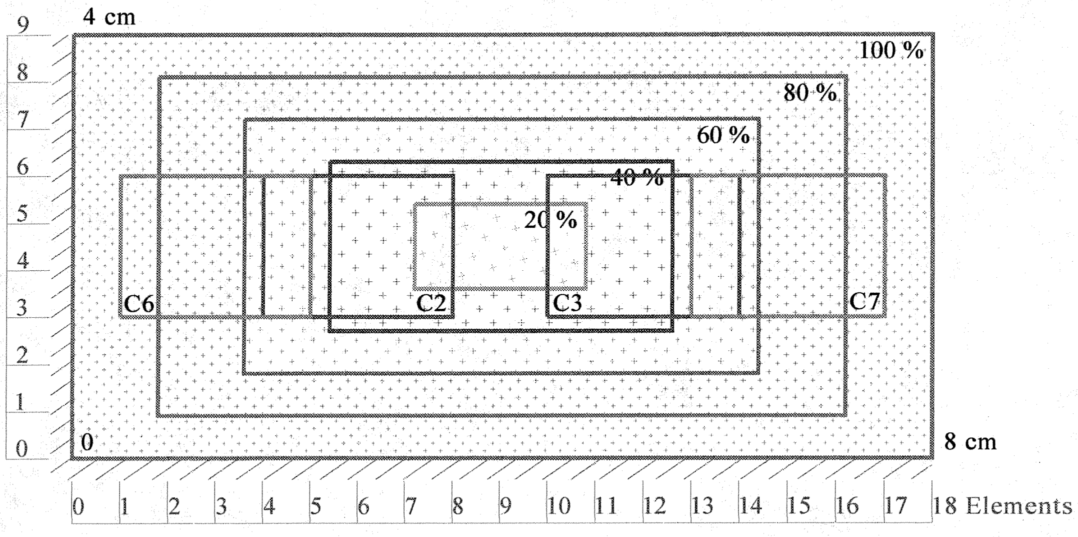

FIGURE 17. Delamination patterns location of cases 2, 3, 6, and 7 (group 1).

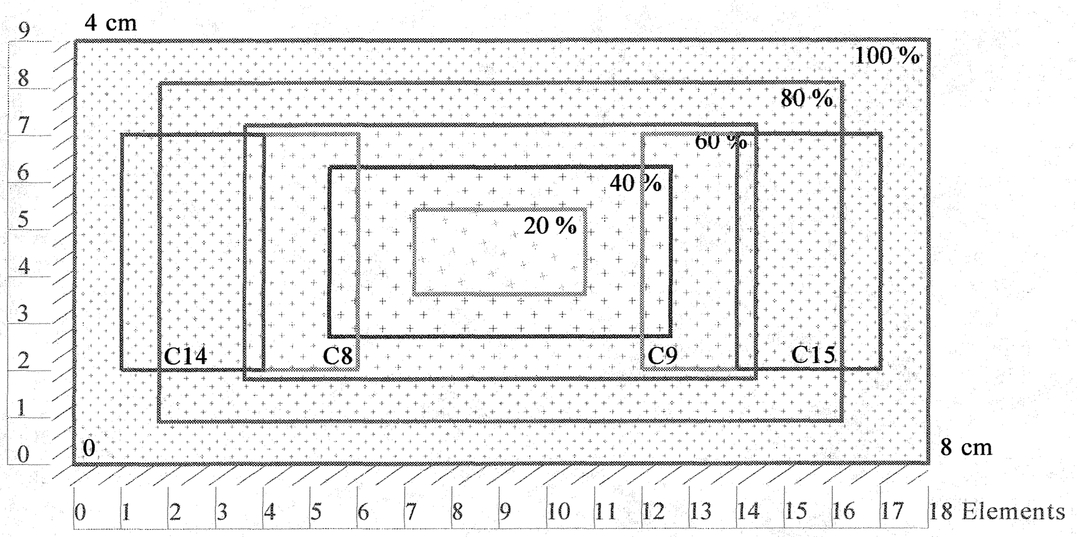

FIGURE 18. Delamination patterns location of cases 8, 9, 14, and 15 (group 2).

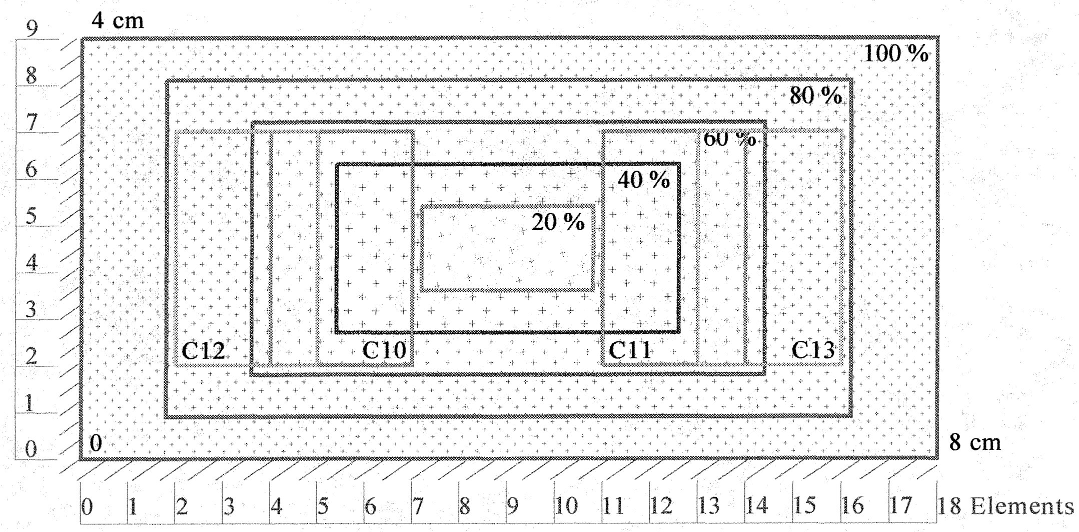

FIGURE 19. Delamination patterns location of cases 10, 11, 12, and 13 (group 2).

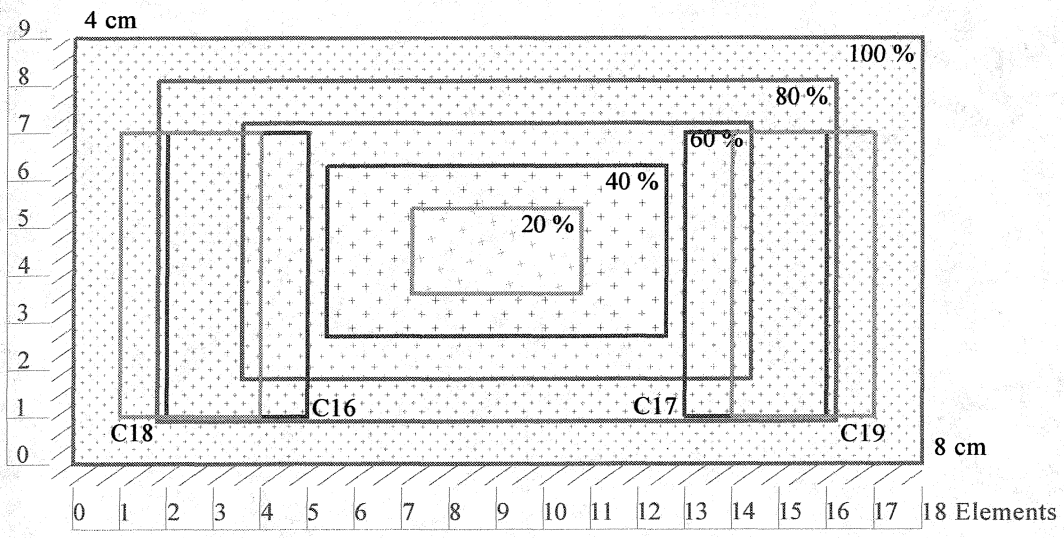

FIGURE 20. Delamination patterns location of cases 16, 17, 18, and 19 (group 3).

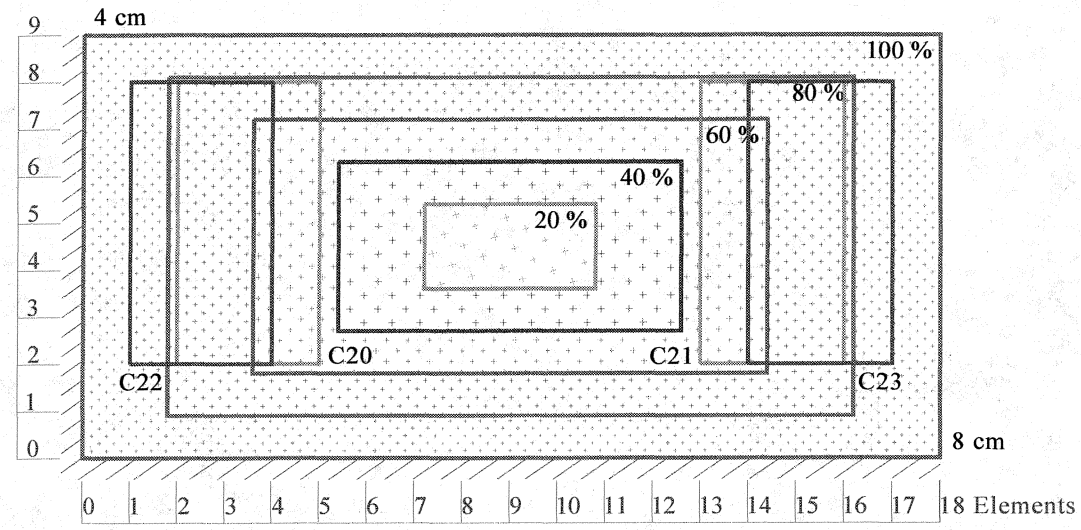

FIGURE 21. Delamination patterns location of cases 20, 21, 22, and 23 (group 3).

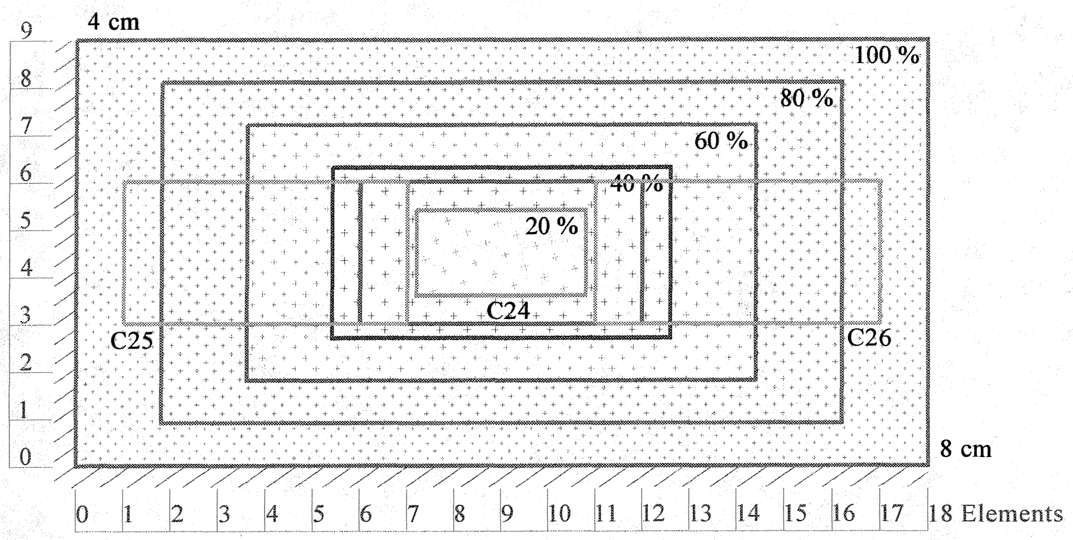

FIGURE 22. Delamination patterns location of cases 24, 25, and 26 (group 4).

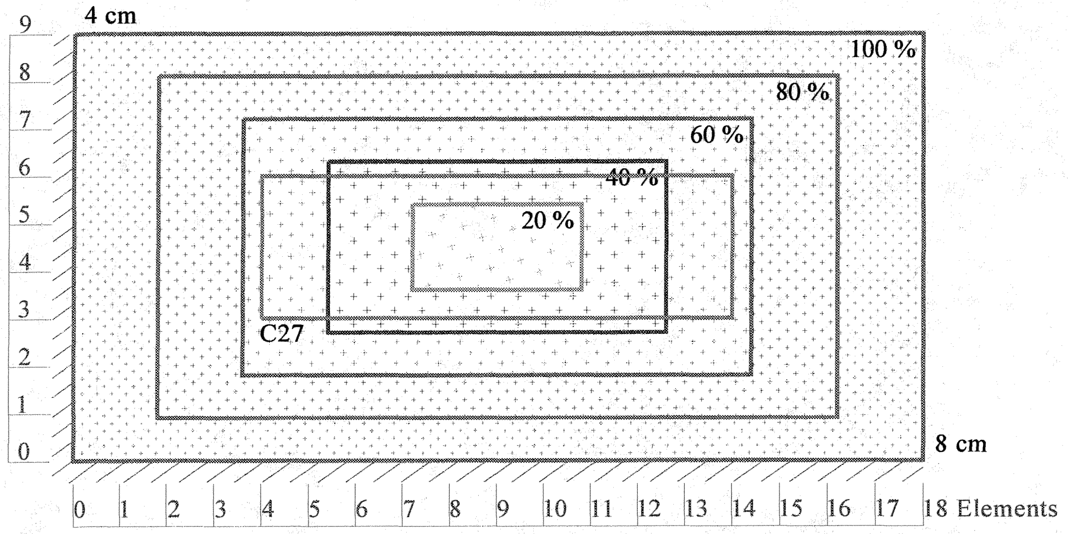

FIGURE 23. Delamination pattern location of case 27 (group 4).

In this comprehensive case study, the delamination patterns will shift between each group from one location to another, such as center to left, center to right, upper-left, lower-left, upper-right, or lower-right. When the delamination patterns relocate, it lands into a region of different density of training data sets. Besides the effect of density of data sets on the detection of the delamination, the effects of size on delamination, its locations, and boundary conditions are also discussed. Below are the observations of the effects on the detection when the delamination patterns are relocated:

C1 - As shown in Figure 16, a delamination size 4x3 (7% of the plate area) fully covers the 20% region, a region of low density of data sets. C1 locates at the central area of the plate where there is no boundary conditions effect. The prediction of C1 is off by one layer, while x_d, y_d, a_x, and a_y are satisfactory.

C2, C3 - As shown in Figure 17, C1 shifts three elements to the left and right, respectively. Each covers a small portion of the 20% region, one-third of the 60% region, and the rest of the 40% region. Since both C2 and C3 fall inside a 60% region, boundary conditions are thus not totally in effect. With changes in densities of data sets, the detections are much better than detection of C1and are all satisfactory.

C4, C5 - As shown in Figure 16, C2 and C3 shift two elements to the left and right, respectively. Each covers mainly the 60% and 80% regions, which are not the favored regions of prediction. With both barely falling inside the 80% region, this may induce slight boundary effects. The predictions are off by two layers for both cases. C4 with x_d is totally off, y_d and a_x are slightly off. C5 with x_d, y_d, a_x, and a_y are satisfactory.

C6, C7 - As shown in Figure 17, C4 and C5 shift one element to the left and right, respectively. Each covers one-third of the 60% region, one-fifth of the 100% region, and the rest of the 80% region. Both fall outside of the 80% region. Since the short edge of delamination is parallel to the left boundary, this may reduce the boundary conditions effect. With changes in densities of data sets, the detection should be better than detection of C4 and C5. For the prediction of C7, a_x and a_y are slightly off.

C8, C9 - As shown in Figure 18, each delamination size 2x5 (6% of the plate area) covers a small portion of the 40% region and the rest of the 60% region. Both fall inside the 60% region and thus boundary conditions are not totally in effect. With size effect (less than 7% of the plate area), the prediction of C8 is off by one layer while C9 is off by two layers. C8 with a_y is slightly off.

C10, C11 - As shown in Figure 19, C8 and C9 enlarge the size from 2x5 to 3x5 (6% to 9% of the plate area). With the size effect, the predictions are much improved and are all satisfactory.

C12, C13 - As shown in Figure 19, C10 and C11 shift two elements to the left and right, respectively. Each covers a half of the 60% region and a half of the 80% region, which are not the favored regions of prediction. With both barely falling inside the 80% region, this may induce slight boundary conditions effect. For the predictions of C12, a_y is slightly off. The prediction of C13 is off by one layer.

C14, C15 - As shown in Figure 18, C12 and C13 shift one element to the left and right, respectively. Each covers a small portion of the 60% region, one-fourth of the 100% region, and the rest of the 80% region. Both fall outside the 80% region. Since the long edge of delamination is parallel to the left boundary, this will increase the boundary conditions effect. The prediction of C14 is off by three layers, x_d is slightly off, a_x is totally off; C15 is satisfactory.

C16, C17 - As shown in Figure 20, each covers two-fifth of the 60% region and the rest of the 80% region, which are not the favored regions of prediction. With both barely falling inside the 80% region, this may increase the boundary conditions effect. The prediction of C16 is off by two layers and $x_\mathrm{d}$is slightly off. The prediction of C17 is satisfactory.

C18, C19 - As shown in Figure 20, C16 and C17 shift one element to the left and right, respectively. Each covers one-third of the 100% region and most of the 80% region. Both fall outside the 80% region and thus boundary conditions are totally in effect. With both lower and left boundary effects, the detection should be more difficult than detection of C16 and C17. The prediction of C18 is off by three layers,x_d is slightly off, anda_x is totally off. The prediction of C19 is satisfactory.

C20, C21 - As shown in Figure 21, C16 and C17 shift one element up and further away from the lower boundary. Each covers two-fifth of the 60% and the rest of the 80% regions, which are not the favored regions of prediction. Both barely fall inside the 80% region and the long edge of delamination is parallel to the left boundary. These may increase the boundary conditions effect. The prediction of C20 is off by one layer, while x_d, y_d, a_x, and a_y are satisfactory. The prediction of C21 is off by two layers, and x_d is slightly off.

C22, C23 - As shown in Figure 21, C18 and C19 shift one element up and further away from the lower boundary, and cover the same regions of density of data sets. This shifting further reduces the lower boundary effect. Thus, the detection of C22 and C23 must be easier than detection of C18 and C19. The prediction of C22 is off by one layer and a_y is totally off. The prediction of C23 is satisfactory.

C24 - As shown in Figure 22, the size of C1 is enlarged from 4x3 to 6x3 (7% to11% of the plate area). It fully covers the 20% region, and most of the 40% region, which are regions of low density of data sets. Since it falls inside the 40% region and locates at the central area of the plate, there is no boundary conditions effect. The preciseness of detection of C24 is similar to C1. With the density of data sets effect, the prediction of C24 is off by one layer, while x_d, y_d, a_x, and a_y are satisfactory.

C25, C26 - As shown in Figure 22, C24 shifts five elements to left and right, respectively. Each covers three-tenth of 40%, 60%, 80% and one-tenth of 100% regions. Both fall outside the 80% region. Since the short edge of delamination is parallel to the left boundary, this further reduces the boundary conditions effect. The predictions are not very precise for both. C25 with x_d is slightly off and a_x is totally off. C26 with x_d is totally off and a_x is slightly off.

C27 - As shown in Figure 23, the size of C24 is enlarged from 6x3 to 10x3 (11% to 19% of the plate area). It fully covers the 20% region, the majority of the 40% region, and some of the 60% region, which are regions of low density of data sets. Since it falls inside the 60% region and locates at the central area of the plate, there is no boundary conditions effect. With the size effect (greater than 17% of the plate area), the prediction of C27 must be easier than prediction of C24, where x_d is slightly off and a_x is totally off.

CHAPTER 8

CONCLUSION

The most widely used method for damage assessment is to identify the occurrence, location, and extent of the damage from measured structural dynamic characteristics. The presence of delamination decreases the natural frequency and causes changes to the mode shape of the composite laminate. This is due to the reduction of the stiffness caused by the delamination. In this paper, a methodology of combined techniques of finite element analysis, counterpropagation neural networks, and genetic algorithms has been introduced and used to solve the delamination detection problem.

An improved-layerwise composite laminate theory is extended to model composite laminates with delamination. This new layerwise finite element model is employed to calculate natural frequencies of cross-ply laminates with given delamination patterns placed at different locations. Therefore, the detection of delamination is first formulated as a simulation and second as an optimization problem and solved by the approach utilizing improved-counterpropagation neural networks and genetic algorithms, respectively. Counterpropagation neural networks are trained to simulate natural frequencies from the finite element analysis. These artificial neural networks are chosen as function approximations which are developed on the available input-output data from a finite element model. Genetic algorithms with mixed type design variables are used to search the optimum delamination patterns associated with the given natural frequencies. Frequencies are used to formulate the objective function. The differences between the measured and computed frequencies are used to formulate the optimization problem that is solved efficiently by genetic algorithms.

A technique has been developed to select the best set of parameters for counterpropagation neural network. The effects of using different validation data sets have found that the mesh size of validation data sets that are coupled with the mesh size of training data sets will affect the performance of the trained neural network, which is indicated by the value of overall relative mean error. Besides the effects of size on delamination, its location, and boundary conditions on the detection of delamination in laminated composites, the density of training data sets also has significant effects on the precise detection of delamination. From many comprehensive case studies, these important observations have been made: at the central area of the plate, there is no boundary conditions effect. But that is the region of low density of data sets, which affects the preciseness of prediction. When the delamination patterns fall into the regions of 60% and 80%, which are not the favored regions of prediction, the detection becomes difficult to predict. When the delamination patterns fall outside the 80% region, then boundary conditions are totally in effect. The degree of effect also depends on the length (short or long edge) of delamination pattern that is parallel to the boundary. A delamination size less than 7% of the plate area is unable to be detected. Small sized delaminations (between 7% to 17% of the plate area) are difficult to detect. Larger sized delaminations (but less than 35% of the plate area) are much easier to detect.

Single internal delamination has been considered in the present studies and has been used to validate the methodology for delamination detection. It is found that internal delamination has been detected with remarkable accuracy using these combined techniques of finite element analysis, counterpropagation neural networks, and genetic algorithms.

APPENDICES

APPENDIX A

FLOW CHART OF THESIS OVERVIEW

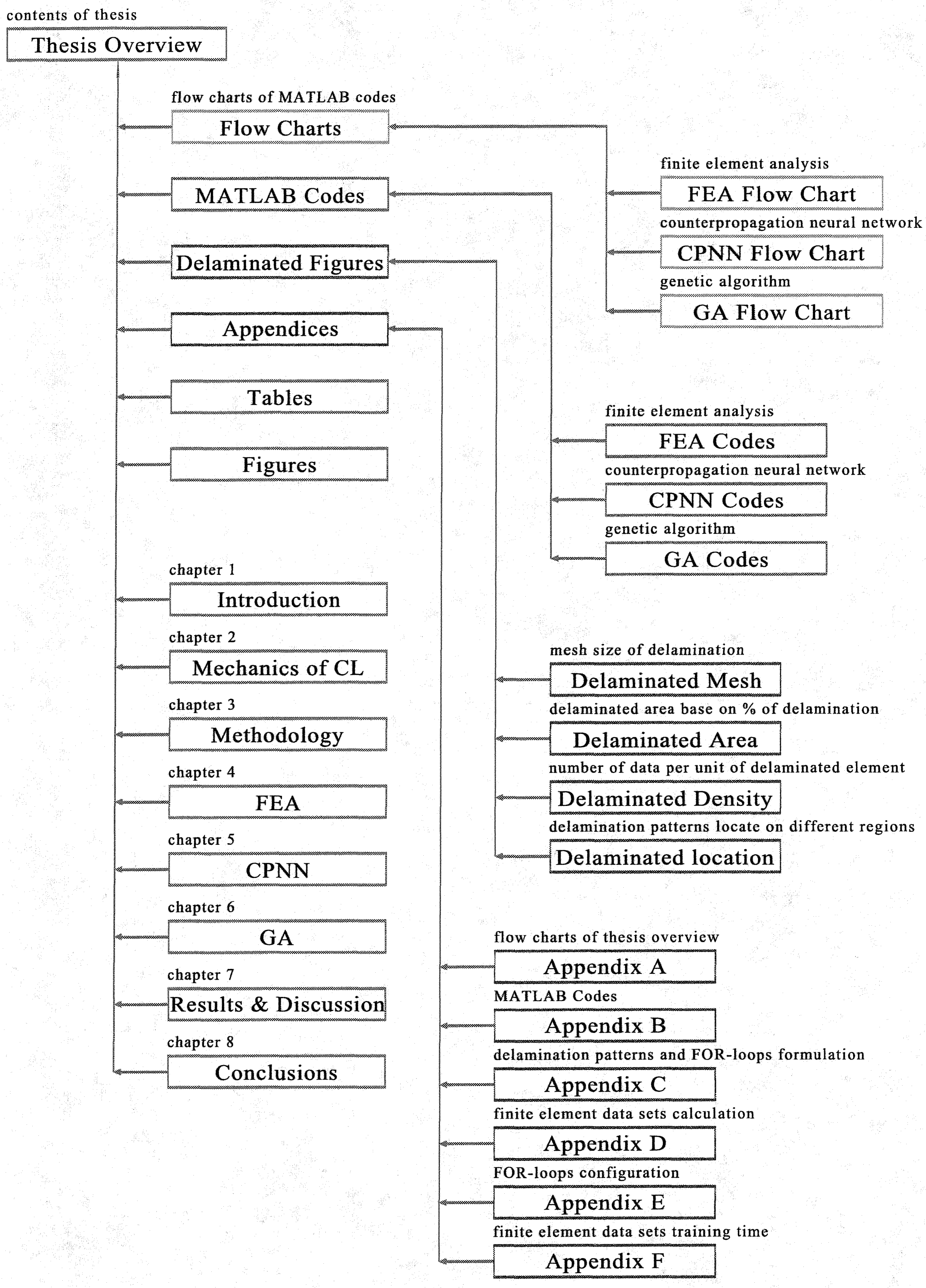

FIGURE 24. Flow chart of thesis overview.

APPENDIX B

MATLAB CODES

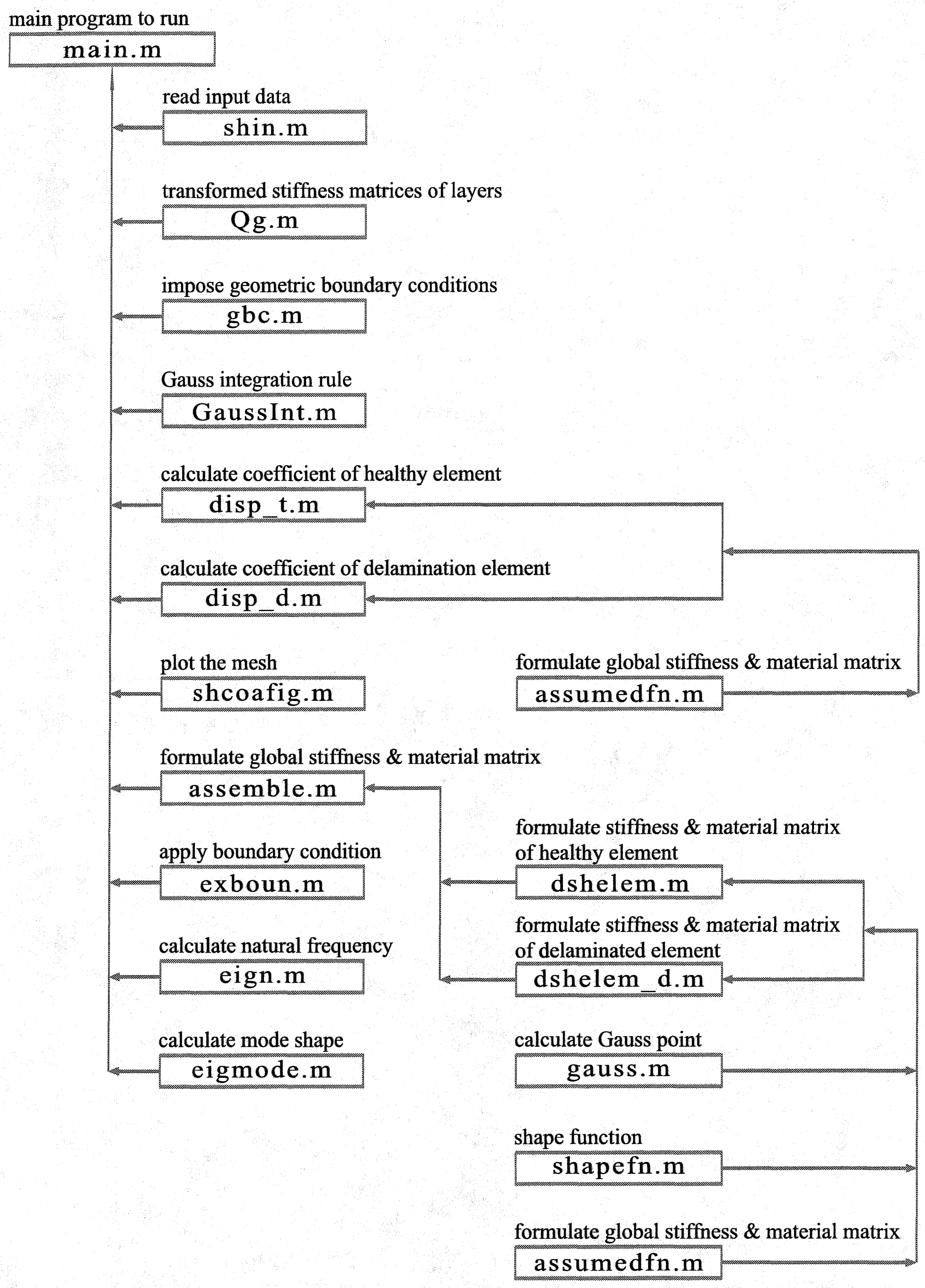
main program to run
main.m
read input data
shin.m
transformed stiffness matrices of layers
Qg.m
impose geometric boundary conditions
gbc.m
Gauss integration rule
GaussInt.m
calculate coefficient of healthy element
disp_t.m
calculate coefficient of delamination element
disp_d.m
plot the mesh
shcoafig.m
formulate global stiffness & material matrix
assumedfn.m
formulate global stiffness & material matrix
assemble.m
formulate stiffness & material matrix
of healthy element
dshelem.m
apply boundary condition
exboun.m
formulate stiffness & material matrix
of delaminated element
dshelem_d.m
calculate natural frequency
eign.m
calculate mode shape
eigmode.m
calculate Gauss point
gauss.m
shape function
shapefn.m
formulate global stiffness & material matrix
assumedfn.m

FIGURE 25. Flow chart of FEA MATLAB codes.

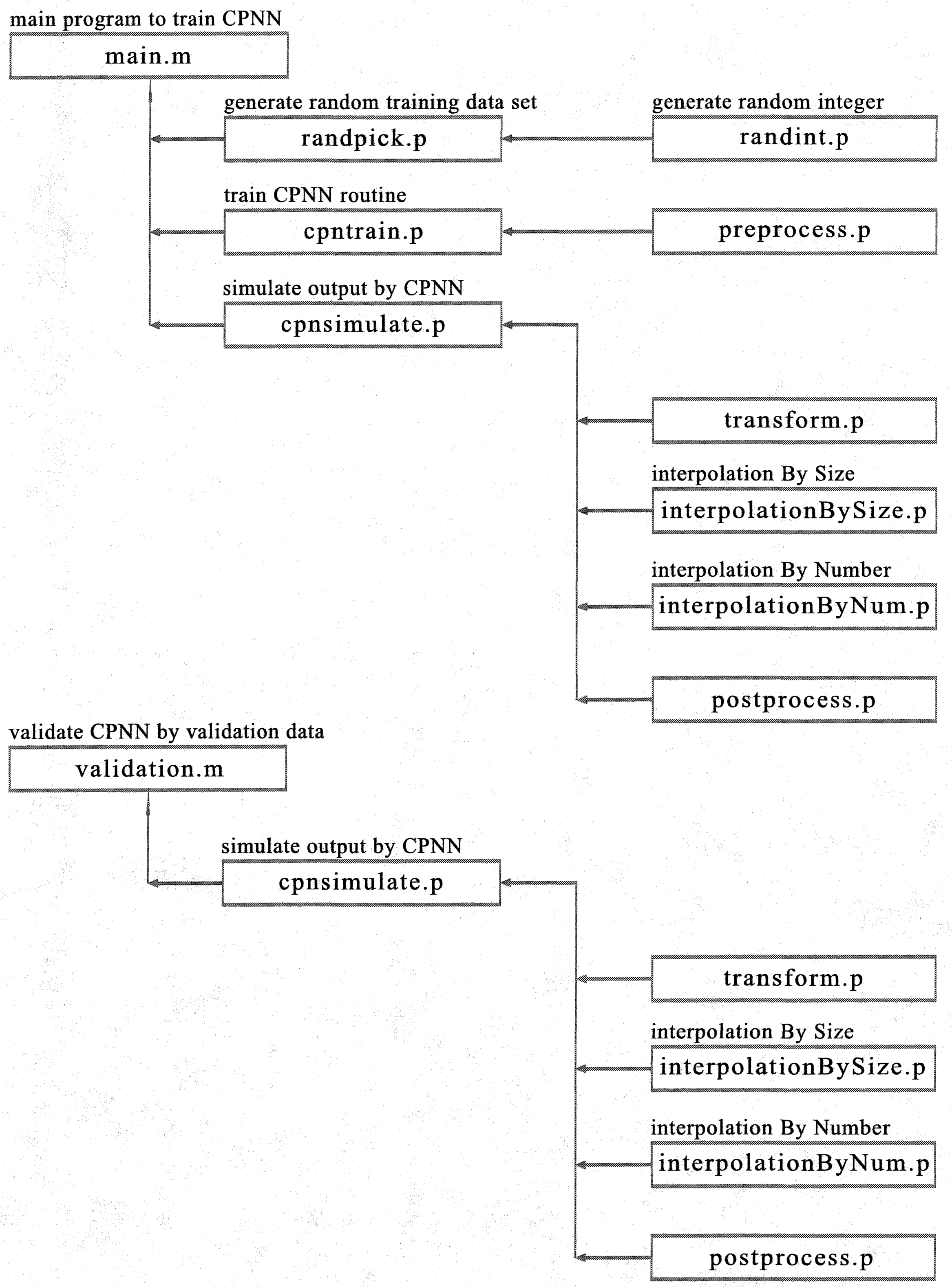

FIGURE 26. Flow chart of CPNN MATLAB codes.

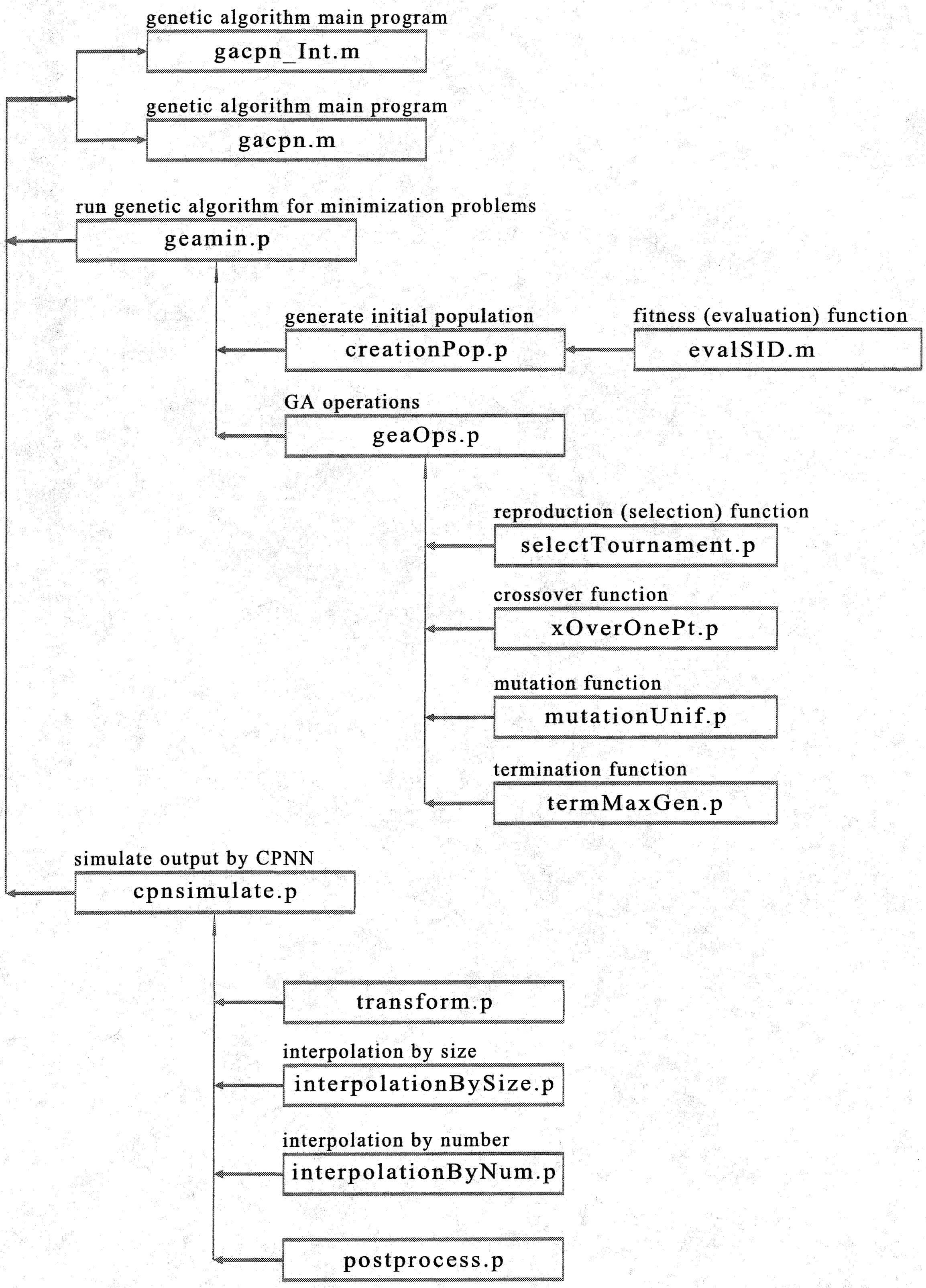

FIGURE 27. Flow chart of GA MATLAB codes.

```
% FILE NAME: main_Tran.m
%
%
% FINITE ELEMENT ANALYSIS
% Solve: Single Internal Delamination Detection Problem (SID)
% Solve: Single Through-the-Width Delamination Detection Problem (STWD)
%    * Theory
%       - Layerwise Shell Theory
%         DOUBLY CURVED ELEMENT, Cross-ply laminates only
%    * Shape function
%       - 4-noded quadratic shape function
%
% PROGRAMMER : Heung Soo Kim
% Arizona State University
% DATE            : 02 / 14 / 2002
%
% MODIFIED BY   : Tran Ngoc Phuong
% Mechanical and Aerospace Engineering Department, CSULB
% DATE            : 02 / 09 / 2006
%
% -----------------------------------------------------------------------------------------------


% -----------------------------------------------------------------------------------------------
% OPEN TEXT FILE AND DATA FILE
% -----------------------------------------------------------------------------------------------
clear all
close all
ti = cputime;                  % initial time

fid = fopen ('fptfreq20x10_SID8x4_100pc_z0123.txt', 'a+');
dataFN = 'fptfreq20x10_SID8x4_100pc_z0123.mat';
delete(dataFN)

% -----------------------------------------------------------------------------------------------
%  USER DIRECT INPUT
% -----------------------------------------------------------------------------------------------
delcase = 2                    % 1 = healthy laminate,  2 = delaminated laminate
gbccase = 2;                   % 1 = STWD,  2 = SID
delpercent =20;                % percentage of delamination
zdLL = 0;  zdUL = 0;           % zdLL = zd Lower Limit,  zdUL = zd Upper Limit
Lo = 0;  Lx = 0.1;  Ly = 0.1;  % lower left corner (Lo),  length (Lx), width (Ly)  (meter)
xMesh = 20;  yMesh = 10;        % mesh-size : (xMesh) x (yMesh)
modeN = 10;                    % number of mode
npe = 4;                       % number of nodes per element
vpn = 7;                       % number of variables per node
```

```
% ------------------------------------------------------------------------------------------
% POINTS, NODES AND ELEMENTS INFORMATIONS
% ------------------------------------------------------------------------------------------
%
% Coordinates of each corner point
PA = Lo;  PD = Lx;  PE = Ly;          % plate dimension (m)
x_eps = xMesh;  y_eps = yMesh;        % mesh-size

% Node and element
x_nps = x_eps+1;        % x_nps : nodes Per X Sides,  x_eps : elements Per X Sides
y_nps = y_eps+1;        % y_nps : nodes Per Y Sides,  y_eps : elements Per Y Sides
nnod = x_nps*y_nps;     % nnod : number of nodes
nelm = x_eps*y_eps;     % nelm : number of elements

% ------------------------------------------------------------------------------------------
% LAMINA MATERIAL PROPERTIES AND STACKING SEQUENCE
% ------------------------------------------------------------------------------------------
%
% Material properties of composite
E1 = 134.4e9;  E2 = 10.34e9;                        % Young's modulus,  E3 = 16.6e9
G12 = 5e9;  G13 = G12;  G23 = G12;                  % Shear modulus
nu12 = 0.33;  nu13 = nu12;  nu21 = E2/E1*nu12;      % nu23 = 0.42
rho = 1477;                                         % density (kg/m^3)
tply = 1.27e-4;                                     % lamina thickness (meter)

% Layup configurations
%
nlam = 8;                             % number of layers
ang = [0 90 0 90 90 0 90 0];          % orientation of layers
h = tply*nlam;                        % laminate thickness
zt = h/nlam*[-nlam/2 : nlam/2];       % thickness distribution of layers
zm = (zt(2 : end) + zt(1 : end-1))/2;

% ------------------------------------------------------------------------------------------
% LIMITS OF FOR-LOOPS
% ------------------------------------------------------------------------------------------
%
delstart = (100 - delpercent) / (2*100);        % delstart : delamination starting

x1L = delstart*x_eps;  x1U = x_eps - x1L - 1;  x2U = x1U+1;

if gbccase == 1                                 % STWD
   y1L = 0;  y1U = 0;  y2U = y_eps;
elseif gbccase == 2                             % SID
   y1L = delstart*y_eps;  y1U = y_eps - y1L - 1;  y2U = y1U+1;
end
```

```
%  ----------------------------------------------------------------------------------------------
%  FOR-LOOPS OF DELAMINATION
%
%          zd :  thickness location of delamination
%                0 – delamination at midplane
%                1 – delamination at first layer above midplane
%                2 – delamination at second layer above midplane
%                3 – delamination at third layer above midplane
% ----------------------------------------------------------------------------------------------
zdL = zdLL;  zdU = zdUL;          % Lower Limit and Upper Limit of zd
for zd = zdL : zdU                % zd: thickness location of delamination (0 : 3)

   for x1 = x1L : x1U             % x1: starting point of delamination (x-axis)
      for y1 = y1L : y1U          % y1: starting point of delamination (y-axis)
         for x2 = x1+1 : x2U      % x2: ending point of delamination (x-axis)

            if gbccase == 1       % 1: through-the-width delamination
               y2L = y_eps;
            elseif gbccase == 2   % 2:  internal delamination
               y2L = y1+1;
            end

            for y2 = y2L : y2U    % y2: ending point of delamination (y-axis)
               tic

               %  ---------------------------------------------------------------------------
               %  READ FILES :
               %      shin_Tran          disp_t          exboun
               %      Qg                 disp_d          eign
               %      gbc                shcoafig        eigmode
               %      GaussInt           assemble
               %  ---------------------------------------------------------------------------
               shin_Tran          % Input data content
               Qg                 % Transformed stiffness matrices of layers
               gbc                % Impose geometric boundary conditions
               GaussInt           % Gauss integration rule
               disp_t             % Calculate the coefficient of healthy element
               disp_d             % Calculate the coefficent of delamination element
               %  shcoafig        % Coordinate x,y,z and Figure

               %  ---------------------------------------------------------------------------
               %  FORMULATE STIFFNESS AND MATERIAL MATRIX
               %
               %  assemble:  Formulate Global Stiffness and Material Matrix
               %          GM : Global Mass Matrix
               %          GK : Global Stiffness Matrix
```

```
%
%          dshelem :  Formulate stiffness and material matrix of healthy
%                      element
%          dshelem_d :  Formulate stiffness and material matrix of
%                        delaminated element
%
%               gauss     :  Calculate Gauss point
%               shapefn :  Shape function
% ---------------------------------------------------------------------------------
DELU = zeros (vpn*nnod,1);
assemble          % Formulate Global Stiffness and Material Matrix
exboun            % Boundary Condition Apply

% ---------------------------------------------------------------------------------
%  SOLVE THE SYSTEM OF EIGENVECTORS AND EIGENVALUES
%
%  eign :  solve the generalized Eigenvalue problem, order and
%           normalize the Eigenvectors
%        [Lambda,Phi,Psi] = eign(A,B)
%                        A x = [Lambda] B x
%
%  eigmode :  calculate the mode shape
%        modeshape = eigvec(: , mm)
%        eigvec : eigenvector
%        eigval : eigenvalue
% ---------------------------------------------------------------------------------
save temp1 GK GM nnod NODEINDEX elemin nodin nodetype nelm
load temp1

 [eig_val, eig_vec] = eign(GK,GM);
nmode = modeN;                          % number of mode
eigvec = eig_vec(:, 1:nmode);
eigval = eig_val(1:nmode);
nopn = 0;
omega = sqrt(eigval)/2/pi;
target = omega;                         % natural frequency
 % eigmode                                % mode shape

% ---------------------------------------------------------------------------------
%  PRINTOUT THE RESULTS AND SAVE FILE
% ---------------------------------------------------------------------------------
%
%  Printout the results
if delcase == 1
   fprintf('\nhealthy laminate with deltype = %2.0f\n',gbccase)
   target
```

```
                  return
               elseif delcase == 2
                  if gbccase == 1
                     fprintf(fid,'%4.0f %7.4f %7.4f %12.4f %12.4f %12.4f %12.4f %12.4f
                     %12.4f %12.4f %12.4f %12.4f %12.4f \n',zd,x1_r,ax_r,target');
                     delamination_pattern = [zd,x1,x2]'
                  elseif gbccase == 2
                     fprintf(fid,'%4.0f %7.4f %7.4f %7.4f %7.4f %12.4f %12.4f %12.4f
                     %12.4f %12.4f %12.4f %12.4f %12.4f %12.4f %12.4f \n',...
                     zd,x1_r,y1_r,ax_r,ay_r,target');
                     delamination_pattern = [zd,x1,y1,x2,y2]'
                  end
               end

               ann_pin = pin                        % ann: artificial neural network
               ann_target = target

               %  Save data file
               %
               if exist(dataFN,'file') == 2         % if exist([dataFN '.mat'],'file') == 2
                  pt = load(dataFN);
                  pin = [pt.pin,pin];
                  target = [pt.target,target];
               end
               save(dataFN,'pin','target')

               toc
            end
         end
      end
   end
end

fclose(fid);

% ----------------------------------------------------------------------------------------------------
% COMPUTER ELAPSED TIME
% ----------------------------------------------------------------------------------------------------
tf = cputime;              % final time
dt = tf -  ti;             % dt : total computer elapsed time
disp(' ')
fprintf('Total elapsed time is %g seconds.\n',dt)
```

```
% FILE NAME:  shin_Tran.m
%
%
% FINITE ELEMENT ANALYSIS INPUT DATA
% Solve:  Single Internal Delamination Detection Problem (SID)
% Solve:  Single Through-the-Width Delamination Detection Problem (STWD)
%
%
% PROGRAMMER : Heung Soo Kim
% Arizona State University
% DATE              : 02 / 14 / 2002
%
% MODIFIED BY   : Tran Ngoc Phuong
% Mechanical and Aerospace Engineering Department, CSULB
% DATE              : 02 / 09 / 2006
%
% ---------------------------------------------------------------------------------------------
% nnod          : Number of nodes
% nelm          : Number of elements
% ndel          : Number of delaminations
% a             : Shell length
% delin         : Delamination location
% delid         : Delamination elements
% dnod          : Number of delamination nodes
% nodetype   : Node type (node in delamination or health element)
% bnodetype  : Node type (node in the boundary of delamination element)
% zt            : Thickness distribution of layers
% zd            : Thickness location of delamination
% ---------------------------------------------------------------------------------------------


% ---------------------------------------------------------------------------------------------
% SYLINDRICAL SHELL SURFACE GENERATION
%     Curvature R1,R2
% ---------------------------------------------------------------------------------------------
%
% R1 = 1.0e+30;  R2 = 1.0e+30;


% ---------------------------------------------------------------------------------------------
%  NODE DATA
%     nodin(:,1) : node number
%     nodin(:,2) : x-coordinate
%     nodin(:,3) : y-coordinate
%     nodin(:,4) : z-coordinate
% ---------------------------------------------------------------------------------------------
```

```
nodin = zeros(nnod,4);
for i = 1:nnod,  nodin(i,1) = i;,  end

for i = 1 : x_nps
   for j = 1 : y_nps
      nodin((i-1)*y_nps+j,2) = (i-1)*PD/(x_nps-1);
      nodin((i-1)*y_nps+j,3) = (j-1)*PE/(y_nps-1);
      nodin((i-1)*y_nps+j,4) = 0;
   end
end


% -------------------------------------------------------------------------------------------------
% ELEMENT DATA
%      elemin : ELEMent INput data [eid, nid1, nid2, nid3, nid4]
% -------------------------------------------------------------------------------------------------
elemin = zeros(nelm,5);
for i = 1:nelm,  elemin(i,1) = i;,  end

for i = 1:x_eps
   for j = 1:y_eps
      elemin((i-1)*y_eps+j,2) = (i-1)*y_nps+j;
      elemin((i-1)*y_eps+j,3) = i*y_nps+j;
      elemin((i-1)*y_eps+j,4) = i*y_nps+j+1;
      elemin((i-1)*y_eps+j,5) = (i-1)*y_nps+j+1;
   end
end


% -------------------------------------------------------------------------------------------------
% DELAMINATION INFORMATION
% -------------------------------------------------------------------------------------------------
ndel = 1;                         % 1 = single delamination,  2 = multiple delaminations
dvpn = 7+5*ndel;                  % degrees of freedom for delaminated node

if delcase == 1                   % 1 = healthy laminate,  2 = delaminated laminate
   delid = zeros(nelm,1);
   dnod = 0;
   nodetype = zeros(nnod,1);
   bnodetype = zeros(nnod,1);
   delin = zeros(1,nlam);
elseif delcase == 2               % 1 = healthy laminate,  2 = delaminated laminate
   delin = zeros(1,nlam);
   nmid = nlam/2;
   delin(nmid+zd) = 1;
   delin(nmid+zd+1:end) = 2;
```

```
element_num = x1*y_eps+1+y1;
ax = x2-x1;                        % x length of delamination
ay = y2-y1;                        % y length of delamination

x1_r = x1*(PD/x_eps)*100;          % real x1 coordinate
y1_r = y1*(PE/y_eps)*100;          % real y1 coordinate
ax_r = ax*(PD/x_eps)*100;          % real ax length
ay_r = ay*(PE/y_eps)*100;          % real ay length

if gbccase == 1                    % 1 = through-the-width delamination
   pin = [];
   pin(1,1) = zd;
   pin(2,1) = x1_r;
   pin(3,1) = ax_r;

   delid = zeros(nelm,1);
   for i = 1 : ax
       for j = x1*y_eps+1+y1+(i-1)*y_eps  :  x1*y_eps+y2+(i-1)*y_eps
          delid(j) = 1;
      end
   end

   dnod = (ax+1)*(ay+1);
   nodetype = zeros(nnod,1);
   for i = 1 : ax+1
      for j = x1*y_nps+1+y1+(i-1)*y_nps : x1*y_nps+1+y2+(i-1)*y_nps
         nodetype(j) = 1;
      end
   end

   bnodetype = zeros(nnod,1);
   for i = x1*y_nps+1+y1 : x1*y_nps+1+y2,  bnodetype(i) = 1;  end

   for i = x1*y_nps+1+y1+ax*y_nps  :  x1*y_nps+1+y2+ax*y_nps
      bnodetype(i) = 1;
   end

elseif gbccase == 2                % 2 = internal delamination
   pin = [];
   pin(1,1) = zd;
   pin(2,1) = x1_r;
   pin(3,1) = y1_r;
   pin(4,1) = ax_r;
   pin(5,1) = ay_r;

   delid = zeros(nelm,1);
```

```
        for i = 1 : ax
            for j = x1*y_eps+1+y1+(i-1)*y_eps  :  x1*y_eps+y2+(i-1)*y_eps
                delid(j) = 1;
            end
        end

        dnod = (ax+1)*(ay+1);
        nodetype = zeros(nnod,1);
        for i = 1 : ax+1
            for j = x1*y_nps+1+y1+(i-1)*y_nps  :  x1*y_nps+1+y2+(i-1)*y_nps
                nodetype(j) = 1;
            end
        end

        bnodetype = zeros(nnod,1);
        for i = x1*y_nps+1+y1 : x1*y_nps+1+y2,  bnodetype(i) = 1;  end

        for i = x1*y_nps+1+y1+ax*y_nps  :  x1*y_nps+1+y2+ax*y_nps
            bnodetype(i) = 1;
        end

        for i = 1 : ax-1
            for j = x1*y_nps+1+y1+y_nps*i,  bnodetype(j) = 1;  end
        end

        for i = 1 : ax-1
            for j = x1*y_nps+1+y2+y_nps*i,  bnodetype(j) = 1;  end
        end

    end
end

pointer = 1;
NODEINDEX(1,1) = 1;  NODEINDEX(1,2) = 1;
for i = 1:nnod-1
    if nodetype(i) == 0
        pointer = pointer+vpn;
        NODEINDEX(i+1,1) = i+1;
        NODEINDEX(i+1,2) = pointer;
    elseif nodetype(i) == 1
        pointer = pointer+dvpn;
        NODEINDEX(i+1,1) = i+1;
        NODEINDEX(i+1,2) = pointer;
    end
end
```

```
% --------------------------------------------------------------------------------------------------
% USER INFORMATION
% --------------------------------------------------------------------------------------------------
disp(' ')
disp(' Four-Noded Element ')
fprintf(' NLAM = %g \n',nlam)
fprintf(' Mesh Size = %g x %g \n',x_eps,y_eps)
fprintf(' Global matrix Size = %g x %g \n',vpn*nnod+dnod*(dvpn-vpn),vpn*nnod . . .
+dnod*(dvpn-vpn))
dis
```

APPENDIX C

DELAMINATION PATTERNS (PIN) AND MATLAB FOR-LOOPS FORMULATION

Delamination Patterns (PIN) and MATLAB FOR-Loops Formulation
(Refer to Figure 28)

Notations of Delamination

$\wp$: Delamination in x-direction or y-direction

$\wp_{\%}$: Percentage of delamination in x-direction or y-direction

$\wp_{\text{start}}$: Delamination start in x-direction or y-direction

$\wp_{\text{end}}$: Delamination end in x-direction or y-direction

$x1$: Length of delamination start in $x1$-direction

$y1$: Length of delamination start in $y1$-direction

$x2$: Length of delamination start in $x2$-direction

$y2$: Length of delamination start in $y2$-direction

Z_d : Layer of delamination

x_d : Length of delamination start in x-direction

y_d : Length of delamination start in y-direction

a_x : Length of actual delamination in x-direction

a_y : Length of actual delamination in y-direction

L_x : Length of the plate in x-direction (8 cm)

L_y : Length of the plate in y-direction (4 cm)

PIN : $\lfloor Z_d \ \ x_d \ \ y_d \ \ a_x \ \ a_y \rfloor$

Note: FOR-loop is independent from the plate dimensions but depends on the mesh size and the percentage of the delamination.

PIN is independent from the mesh size but depends on the plate dimension and the percentage of the delamiation.

Mesh-Size = 10 x 5

$$\text{Mesh size} = \text{xMesh} \times \text{yMesh} \quad \Rightarrow \quad \text{xMesh} = 10\,\text{elements}, \ \ \text{yMesh} = 5\,\text{elements}$$

20 percent delamination $\left(20\wp_{\%}\right)$

$$\wp_{\text{start}} : \ \frac{100\% - 20\%}{2}(\text{Mesh}) = 40\%(\text{Mesh}) = 0.4(\text{Mesh})$$

$$\wp_{\text{end}} : \ (40\% + 20\%)\text{Mesh} = 60\%(\text{Mesh}) = 0.6(\text{Mesh})$$

$$20\,\wp_{\%} \ \Rightarrow \begin{cases} x1 = 0.4(10) \to 0.6(10) - 1 \\ y1 = 0.4(5) \ \to 0.6(5) - 1 \\ x2 = x1 + 1 \ \ \to 0.6(10) \\ y2 = y1 + 1 \ \ \to 0.6(5) \end{cases} \xrightarrow{\text{FOR-Loops}} \boxed{\begin{array}{l} \text{for x1} = 4 \to 5 \\ \quad \text{for y1} = 2 \to 2 \\ \qquad \text{for x2} = \text{x1} + 1 \to 6 \\ \qquad\quad \text{for y2} = \text{y1} + 1 \to 3 \end{array}}$$

$$\text{PIN}_{20\wp_{\%}} = \begin{cases} Z_d = 0,\,1,\,2,\,3 \\ x_d = \dfrac{L_x - a_x}{2} = \dfrac{8-1.6}{2} = 3.2\,\text{cm} \\ y_d = \dfrac{L_y - a_y}{2} = \dfrac{4-0.8}{2} = 1.6\,\text{cm} \\ a_x = 20\%(L_x) = 0.2(8\,\text{cm}) = 1.6\,\text{cm} \\ a_y = 20\%(L_y) = 0.2(4\,\text{cm}) = 0.8\,\text{cm} \end{cases}$$

$$\text{PIN}_{20\wp_{\%}} = [Z_d \quad 3.2\,\text{cm} \quad 1.6\,\text{cm} \quad 1.6\,\text{cm} \quad 0.8\,\text{cm}]$$

60 percent delamination $(60\wp_{\%})$

$$\wp_{\text{start}} : \ \frac{100\% - 60\%}{2}(\text{Mesh}) = 20\%(\text{Mesh}) = 0.2(\text{Mesh})$$

$$\wp_{\text{end}} : \ (20\% + 60\%)\text{Mesh} = 80\%(\text{Mesh}) = 0.8(\text{Mesh})$$

$$60\wp_{\%} \Rightarrow \begin{cases} x1 = 0.2(10) \rightarrow 0.8(10) - 1 \\ y1 = 0.2(5) \rightarrow 0.8(5) - 1 \\ x2 = x1 + 1 \rightarrow 0.8(10) \\ y2 = y1 + 1 \rightarrow 0.8(5) \end{cases} \xrightarrow{\text{FOR-Loops}} \boxed{\begin{array}{l} \text{for x1 } = 2 \rightarrow 7 \\ \quad \text{for y1 } = 1 \rightarrow 3 \\ \qquad \text{for x2 } = \text{x1} + 1 \rightarrow 8 \\ \qquad\quad \text{for y2 } = \text{y1} + 1 \rightarrow 4 \end{array}}$$

$$\text{PIN}_{60\wp_{\%}} = \begin{cases} Z_d = 0,\,1,\,2,\,3 \\ x_d = \dfrac{L_x - a_x}{2} = \dfrac{8-4.8}{2} = 1.6\,\text{cm} \\ y_d = \dfrac{L_y - a_y}{2} = \dfrac{4-2.4}{2} = 0.8\,\text{cm} \\ a_x = 60\%(L_x) = 0.6(8\,\text{cm}) = 4.8\,\text{cm} \\ a_y = 60\%(L_y) = 0.6(4\,\text{cm}) = 2.4\,\text{cm} \end{cases}$$

$$\text{PIN}_{60\wp_{\%}} = [Z_d \quad 1.6\,\text{cm} \quad 0.8\,\text{cm} \quad 4.8\,\text{cm} \quad 2.4\,\text{cm}]$$

100 percent delamination $(100\wp_{\%})$

$$\wp_{\text{start}} : \ \frac{100\% - 100\%}{2}(\text{Mesh}) = 0\%(\text{Mesh}) = 0$$

$$\wp_{\text{end}} : \ (0\% + 100\%)\text{Mesh} = 100\%(\text{Mesh}) = 1(\text{Mesh})$$

$$100\wp_{\%} \Rightarrow \begin{cases} x1 = 0(10) \rightarrow 1(10) - 1 \\ y1 = 0(5) \rightarrow 1(5) - 1 \\ x2 = x1 + 1 \rightarrow 1(10) \\ y2 = y1 + 1 \rightarrow 1(5) \end{cases} \xrightarrow{\text{FOR-Loops}} \boxed{\begin{array}{l} \text{for x1 } = 0 \rightarrow 9 \\ \quad \text{for y1 } = 0 \rightarrow 4 \\ \qquad \text{for x2 } = \text{x1} + 1 \rightarrow 10 \\ \qquad\quad \text{for y2 } = \text{y1} + 1 \rightarrow 5 \end{array}}$$

$$\text{PIN}_{100\wp_{\%}} = \begin{cases} Z_d = 0, 1, 2, 3 \\ x_d = \dfrac{L_x - a_x}{2} = \dfrac{8-8}{2} = 0\,\text{cm} \\ y_d = \dfrac{L_y - a_y}{2} = \dfrac{4-4}{2} = 0\,\text{cm} \\ a_x = 100\%(L_x) = 1(8\,\text{cm}) = 8\,\text{cm} \\ a_y = 100\%(L_y) = 1(4\,\text{cm}) = 4\,\text{cm} \end{cases}$$

$$\text{PIN}_{100\wp_{\%}} = [Z_d \quad 0\,\text{cm} \quad 0\,\text{cm} \quad 8\,\text{cm} \quad 4\,\text{cm}]$$

<u>Mesh-Size = 12 x 6</u>

$\text{Mesh size} = \text{xMesh} \times \text{yMesh} \quad \Rightarrow \quad \text{xMesh} = 12\,\text{elements}, \ \text{yMesh} = 6\,\text{elements}$

100 percent delamination $(100\wp_{\%})$

$$\wp_{\text{start}} : \ \frac{100\% - 100\%}{2}(\text{Mesh}) = 0\%(\text{Mesh}) = 0$$

$$\wp_{\text{end}} : \ (0\% + 100\%)\text{Mesh} = 100\%(\text{Mesh}) = 1(\text{Mesh})$$

$$100\wp_{\%} \Rightarrow \begin{cases} x1 = 0(12) \rightarrow 1(12) - 1 \\ y1 = 0(6) \rightarrow 1(6) - 1 \\ x2 = x1 + 1 \rightarrow 1(12) \\ y2 = y1 + 1 \rightarrow 1(6) \end{cases} \xrightarrow{\text{FOR-Loops}}$$

```
for x1 = 0 → 11
   for y1 = 0 → 5
      for x2 = x1 + 1 → 12
         for y2 = y1 + 1 → 6
```

$$\text{PIN}_{100\wp_{\%}} = \begin{cases} Z_d = 0, 1, 2, 3 \\ x_d = \dfrac{L_x - a_x}{2} = \dfrac{8-8}{2} = 0\,\text{cm} \\ y_d = \dfrac{L_y - a_y}{2} = \dfrac{4-4}{2} = 0\,\text{cm} \\ a_x = 100\%(L_x) = 1(8\,\text{cm}) = 8\,\text{cm} \\ a_y = 100\%(L_y) = 1(4\,\text{cm}) = 4\,\text{cm} \end{cases}$$

$$\text{PIN}_{100\wp_{\%}} = [Z_d \quad 0\,\text{cm} \quad 0\,\text{cm} \quad 8\,\text{cm} \quad 4\,\text{cm}]$$

<u>Mesh-Size = 16 x 8</u>

$\text{Mesh size} = \text{xMesh} \times \text{yMesh} \quad \Rightarrow \quad \text{xMesh} = 16\,\text{elements}, \ \text{yMesh} = 8\,\text{elements}$

100 percent delamination $(100\wp_{\%})$

$$\wp_{\text{start}} : \ \frac{100\% - 100\%}{2}(\text{Mesh}) = 0\%(\text{Mesh}) = 0$$

$$\wp_{\text{end}} : \ (0\% + 100\%)\text{Mesh} = 100\%(\text{Mesh}) = 1(\text{Mesh})$$

$$100\wp_{\%} \Rightarrow \begin{cases} x1 = 0(16) \rightarrow 1(16)-1 \\ y1 = 0(8) \rightarrow 1(8)-1 \\ x2 = x1+1 \rightarrow 1(16) \\ y2 = y1+1 \rightarrow 1(8) \end{cases} \xrightarrow{\text{FOR-Loops}} \boxed{\begin{array}{l} \text{for x1 } = 0 \rightarrow 15 \\ \quad \text{for y1 } = 0 \rightarrow 7 \\ \qquad \text{for x2 } = \text{x1}+1 \rightarrow 16 \\ \qquad\quad \text{for y2 } = \text{y1}+1 \rightarrow 8 \end{array}}$$

$$\text{PIN}_{100\wp_{\%}} = \begin{cases} Z_d = 0, 1, 2, 3 \\ x_d = \dfrac{L_x - a_x}{2} = \dfrac{8-8}{2} = 0\,\text{cm} \\ y_d = \dfrac{L_y - a_y}{2} = \dfrac{4-4}{2} = 0\,\text{cm} \\ a_x = 100\%(L_x) = 1(8\,\text{cm}) = 8\,\text{cm} \\ a_y = 100\%(L_y) = 1(4\,\text{cm}) = 4\,\text{cm} \end{cases}$$

$$\text{PIN}_{100\wp_{\%}} = [Z_d \quad 0\,\text{cm} \quad 0\,\text{cm} \quad 8\,\text{cm} \quad 4\,\text{cm}]$$

Mesh-size = 18 x 9

Mesh size = xMesh × yMesh ⇒ xMesh = 18 elements, yMesh = 9 elements

100 percent delamination $(100\wp_{\%})$

$$\wp_{\text{start}} : \frac{100\% - 100\%}{2}(\text{Mesh}) = 0\%(\text{Mesh}) = 0$$

$$\wp_{\text{end}} : (0\% + 100\%)\text{Mesh} = 100\%(\text{Mesh}) = 1(\text{Mesh})$$

$$100\wp_{\%} \Rightarrow \begin{cases} x1 = 0(18) \rightarrow 1(18)-1 \\ y1 = 0(9) \rightarrow 1(9)-1 \\ x2 = x1+1 \rightarrow 1(18) \\ y2 = y1+1 \rightarrow 1(9) \end{cases} \xrightarrow{\text{FOR-Loops}} \boxed{\begin{array}{l} \text{for x1 } = 0 \rightarrow 17 \\ \quad \text{for y1 } = 0 \rightarrow 8 \\ \qquad \text{for x2 } = \text{x1}+1 \rightarrow 18 \\ \qquad\quad \text{for y2 } = \text{y1}+1 \rightarrow 9 \end{array}}$$

$$\text{PIN}_{100\wp_{\%}} = \begin{cases} Z_d = 0, 1, 2, 3 \\ x_d = \dfrac{L_x - a_x}{2} = \dfrac{8-8}{2} = 0\,\text{cm} \\ y_d = \dfrac{L_y - a_y}{2} = \dfrac{4-4}{2} = 0\,\text{cm} \\ a_x = 100\%(L_x) = 1(8\,\text{cm}) = 8\,\text{cm} \\ a_y = 100\%(L_y) = 1(4\,\text{cm}) = 4\,\text{cm} \end{cases}$$

$$\text{PIN}_{100\wp_{\%}} = [Z_d \quad 0\,\text{cm} \quad 0\,\text{cm} \quad 8\,\text{cm} \quad 4\,\text{cm}]$$

Mesh Size = 20 x 10

Mesh size = xMesh × yMesh ⇒ xMesh = 20 elements, yMesh = 10 elements

20 percent delamination $(20\wp_{\%})$

$$\wp_{\text{start}} : \frac{100\% - 20\%}{2}(\text{Mesh}) = 40\%(\text{Mesh}) = 0.4(\text{Mesh})$$

$$\wp_{\text{end}} : (40\% + 20\%)\text{Mesh} = 60\%(\text{Mesh}) = 0.6(\text{Mesh})$$

$$20\wp_{\%} \Rightarrow \left\{\begin{array}{l} x1 = 0.4(20) \rightarrow 0.6(20)-1 \\ y1 = 0.4(10) \rightarrow 0.6(10)-1 \\ x2 = x1+1 \quad \rightarrow 0.6(20) \\ y2 = y1+1 \quad \rightarrow 0.6(10) \end{array}\right. \xrightarrow{\text{FOR-Loops}} \boxed{\begin{array}{l} \text{for x1 } = 8 \rightarrow 11 \\ \quad \text{for y1 } = 4 \rightarrow 5 \\ \quad\quad \text{for x2 } = \text{x1}+1 \rightarrow 12 \\ \quad\quad\quad \text{for y2 } = \text{y1}+1 \rightarrow 6 \end{array}}$$

$$\text{PIN}_{20\wp_{\%}} = \left\{\begin{array}{l} Z_d = 0, 1, 2, 3 \\ x_d = \dfrac{L_x - a_x}{2} = \dfrac{8-1.6}{2} = 3.2\,\text{cm} \\ y_d = \dfrac{L_y - a_y}{2} = \dfrac{4-0.8}{2} = 1.6\,\text{cm} \\ a_x = 20\%(L_x) = 0.2(8\,\text{cm}) = 1.6\,\text{cm} \\ a_y = 20\%(L_y) = 0.2(4\,\text{cm}) = 0.8\,\text{cm} \end{array}\right.$$

$$\text{PIN}_{20\wp_{\%}} = [Z_d \quad 3.2\,\text{cm} \quad 1.6\,\text{cm} \quad 1.6\,\text{cm} \quad 0.8\,\text{cm}]$$

40 percent delamination $(40\wp_{\%})$

$$\wp_{\text{start}} : \frac{100\% - 40\%}{2}(\text{Mesh}) = 30\%(\text{Mesh}) = 0.3(\text{Mesh})$$

$$\wp_{\text{end}} : (30\% + 40\%)\text{Mesh} = 70\%(\text{Mesh}) = 0.7(\text{Mesh})$$

$$40\wp_{\%} \Rightarrow \left\{\begin{array}{l} x1 = 0.3(20) \rightarrow 0.7(20)-1 \\ y1 = 0.3(10) \rightarrow 0.7(10)-1 \\ x2 = x1+1 \quad \rightarrow 0.7(20) \\ y2 = y1+1 \quad \rightarrow 0.7(10) \end{array}\right. \xrightarrow{\text{FOR-Loops}} \boxed{\begin{array}{l} \text{for x1 } = 6 \rightarrow 13 \\ \quad \text{for y1 } = 3 \rightarrow 6 \\ \quad\quad \text{for x2 } = \text{x1}+1 \rightarrow 14 \\ \quad\quad\quad \text{for y2 } = \text{y1}+1 \rightarrow 7 \end{array}}$$

$$\text{PIN}_{40\wp_{\%}} = \left\{\begin{array}{l} Z_d = 0, 1, 2, 3 \\ x_d = \dfrac{L_x - a_x}{2} = \dfrac{8-3.2}{2} = 2.4\,\text{cm} \\ y_d = \dfrac{L_y - a_y}{2} = \dfrac{4-1.6}{2} = 1.2\,\text{cm} \\ a_x = 40\%(L_x) = 0.4(8\,\text{cm}) = 3.2\,\text{cm} \\ a_y = 40\%(L_y) = 0.4(4\,\text{cm}) = 1.6\,\text{cm} \end{array}\right.$$

$$\text{PIN}_{40\wp_{\%}} = [Z_d \quad 2.4\,\text{cm} \quad 1.2\,\text{cm} \quad 3.2\,\text{cm} \quad 1.6\,\text{cm}]$$

60 percent delamination $(60\wp_{\%})$

$$\wp_{\text{start}} : \frac{100\% - 60\%}{2}(\text{Mesh}) = 20\%(\text{Mesh}) = 0.2(\text{Mesh})$$

$$\wp_{\text{end}} : (20\% + 60\%)\text{Mesh} = 80\%(\text{Mesh}) = 0.8(\text{Mesh})$$

$$60\wp_{\%} \Rightarrow \left\{\begin{array}{l} x1 = 0.2(20) \rightarrow 0.8(20)-1 \\ y1 = 0.2(10) \rightarrow 0.8(10)-1 \\ x2 = x1+1 \quad \rightarrow 0.8(20) \\ y2 = y1+1 \quad \rightarrow 0.8(10) \end{array}\right. \xrightarrow{\text{FOR-Loops}} \boxed{\begin{array}{l} \text{for x1 } = 4 \rightarrow 15 \\ \quad \text{for y1 } = 2 \rightarrow 7 \\ \quad\quad \text{for x2 } = \text{x1}+1 \rightarrow 16 \\ \quad\quad\quad \text{for y2 } = \text{y1}+1 \rightarrow 8 \end{array}}$$

$$\text{PIN}_{60\wp_{\%}} = \begin{cases} Z_d = 0,\ 1,\ 2,\ 3 \\ x_d = \dfrac{L_x - a_x}{2} = \dfrac{8-4.8}{2} = 1.6\,\text{cm} \\ y_d = \dfrac{L_y - a_y}{2} = \dfrac{4-2.4}{2} = 0.8\,\text{cm} \\ a_x = 60\%\left(L_x\right) = 0.6\left(8\,\text{cm}\right) = 4.8\,\text{cm} \\ a_y = 60\%\left(L_y\right) = 0.6\left(4\,\text{cm}\right) = 2.4\,\text{cm} \end{cases}$$

$$\text{PIN}_{60\wp_{\%}} = \left[Z_d \quad 1.6\,\text{cm} \quad 0.8\,\text{cm} \quad 4.8\,\text{cm} \quad 2.4\,\text{cm}\right]$$

80 percent delamination $\left(80\wp_{\%}\right)$

$$\wp_{\text{start}} : \ \frac{100\% - 80\%}{2}\left(\text{Mesh}\right) = 10\%\left(\text{Mesh}\right) = 0.1\left(\text{Mesh}\right)$$

$$\wp_{\text{end}} : \ \left(10\% + 80\%\right)\text{Mesh} = 90\%\left(\text{Mesh}\right) = 0.9\left(\text{Mesh}\right)$$

$$80\wp_{\%} \Rightarrow \begin{cases} x1 = 0.1(20) \rightarrow 0.9(20) - 1 \\ y1 = 0.1(10) \rightarrow 0.9(10) - 1 \\ x2 = x1 + 1 \ \rightarrow 0.9(20) \\ y2 = y1 + 1 \ \rightarrow 0.9(10) \end{cases} \xrightarrow{\text{FOR-Loops}} \boxed{\begin{array}{l} \text{for x1 } = \ 2 \rightarrow 17 \\ \quad \text{for y1 } = \ 1 \rightarrow 8 \\ \qquad \text{for x2 } = \ \text{x1} + 1 \rightarrow 18 \\ \qquad\quad \text{for y2 } = \ \text{y1} + 1 \rightarrow 9 \end{array}}$$

$$\text{PIN}_{80\wp_{\%}} = \begin{cases} Z_d = 0,\ 1,\ 2,\ 3 \\ x_d = \dfrac{L_x - a_x}{2} = \dfrac{8-6.4}{2} = 0.8\,\text{cm} \\ y_d = \dfrac{L_y - a_y}{2} = \dfrac{4-3.2}{2} = 0.4\,\text{cm} \\ a_x = 80\%\left(L_x\right) = 0.8\left(8\,\text{cm}\right) = 6.4\,\text{cm} \\ a_y = 80\%\left(L_y\right) = 0.8\left(4\,\text{cm}\right) = 3.2\,\text{cm} \end{cases}$$

$$\text{PIN}_{80\wp_{\%}} = \left[Z_d \quad 0.8\,\text{cm} \quad 0.4\,\text{cm} \quad 6.4\,\text{cm} \quad 3.2\,\text{cm}\right]$$

100 percent delamination $\left(100\wp_{\%}\right)$

$$\wp_{\text{start}} : \ \frac{100\% - 100\%}{2}\left(\text{Mesh}\right) = 0\%\left(\text{Mesh}\right) = 0$$

$$\wp_{\text{end}} : \ \left(0\% + 100\%\right)\text{Mesh} = 100\%\left(\text{Mesh}\right) = 1\left(\text{Mesh}\right)$$

$$100\wp_{\%} \Rightarrow \begin{cases} x1 = 0(20) \ \rightarrow 1(20) - 1 \\ y1 = 0(10) \ \rightarrow 1(10) - 1 \\ x2 = x1 + 1 \rightarrow 1(20) \\ y2 = y1 + 1 \rightarrow 1(10) \end{cases} \xrightarrow{\text{FOR-Loops}} \boxed{\begin{array}{l} \text{for x1 } = \ 0 \rightarrow 19 \\ \quad \text{for y1 } = \ 0 \rightarrow 9 \\ \qquad \text{for x2 } = \ \text{x1} + 1 \rightarrow 20 \\ \qquad\quad \text{for y2 } = \ \text{y1} + 1 \rightarrow 10 \end{array}}$$

$$\text{PIN}_{100\wp\%} = \begin{cases} Z_d = 0, 1, 2, 3 \\ x_d = \dfrac{L_x - a_x}{2} = \dfrac{8-8}{2} = 0\,\text{cm} \\ y_d = \dfrac{L_y - a_y}{2} = \dfrac{4-4}{2} = 0\,\text{cm} \\ a_x = 100\%(L_x) = 1(8\,\text{cm}) = 8\,\text{cm} \\ a_y = 100\%(L_y) = 1(4\,\text{cm}) = 4\,\text{cm} \end{cases}$$

$$\text{PIN}_{100\wp\%} = \begin{bmatrix} Z_d & 0\,\text{cm} & 0\,\text{cm} & 8\,\text{cm} & 4\,\text{cm} \end{bmatrix}$$

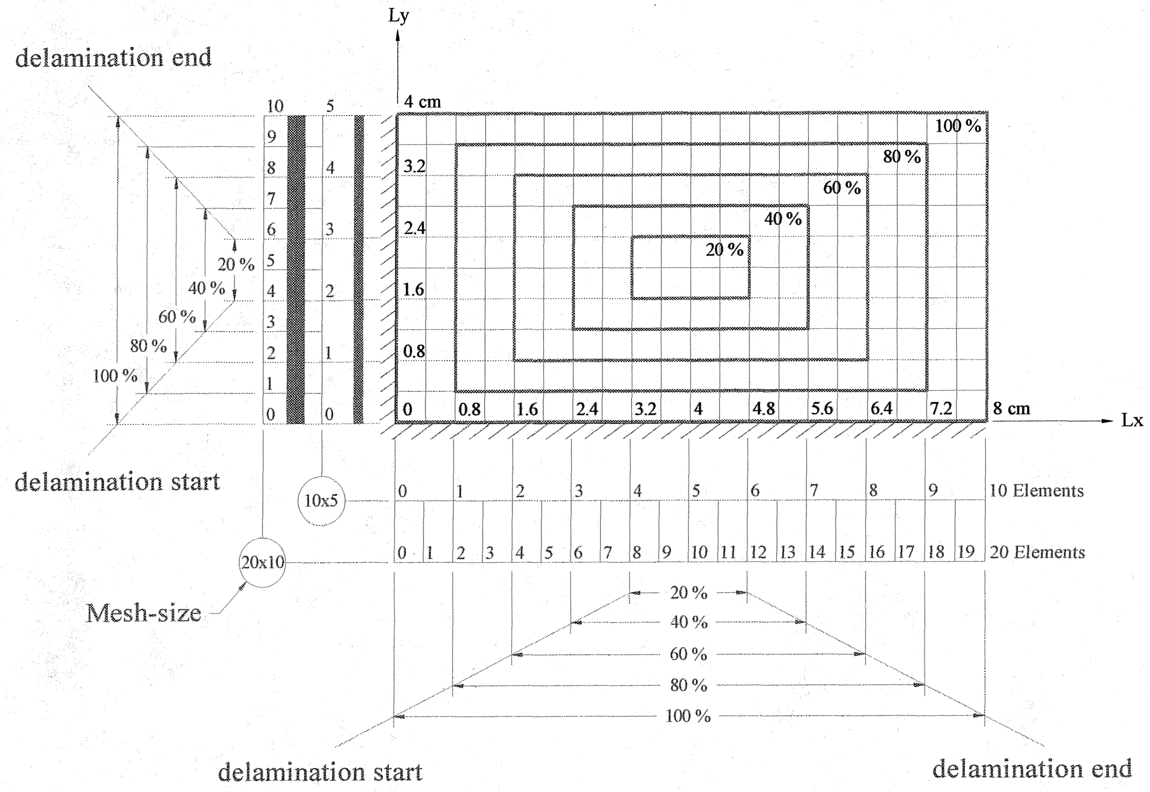

FIGURE 28. Area of delamination for plate dimension 8cm x 4cm with mesh size 10x5 and 20x10.

APPENDIX D

FINITE ELEMENT DATA SETS CALCULATION

Finite Element Data Sets Calculation of Plate Dimension 8cm x 4cm
(Refer to Figure 28)

Mesh Size = 10 x 5

20 percent delamination

```
for $x1 = 4 \rightarrow 5$
    for $y1 = 2 \rightarrow 2$
        for $x2 = x1+1 \rightarrow 6$
            for $y2 = y1+1 \rightarrow 3$
```

$$\xrightarrow{\text{Formula}} \boxed{\sum_{x1=4}^{5}(6-x1)\sum_{y1=2}^{2}(3-y1)}$$

$$\sum_{x1=4}^{5}(6-x1)=2+1=3$$

$$\sum_{y1=2}^{2}(3-y1)=1$$

For one layer: $3(1)=3$ data sets
For four layers: $4(3)=12$ data sets

60 percent delamination

```
for $x1 = 2 \rightarrow 7$
    for $y1 = 1 \rightarrow 3$
        for $x2 = x1+1 \rightarrow 8$
            for $y2 = y1+1 \rightarrow 4$
```

$$\xrightarrow{\text{Formula}} \boxed{\sum_{x1=2}^{7}(8-x1)\sum_{y1=1}^{3}(4-y1)}$$

$$\sum_{x1=2}^{7}(8-x1)=6+5+4+3+2+1=3(7)=21$$

$$\sum_{y1=1}^{3}(4-y1)=3+2+1=6$$

For one layer: $21(6)=126$ data sets
For four layers: $4(126)=504$ data sets

100 percent delamination

```
for $x1 = 0 \rightarrow 9$
    for $y1 = 0 \rightarrow 4$
        for $x2 = x1+1 \rightarrow 10$
            for $y2 = y1+1 \rightarrow 5$
```

$$\xrightarrow{\text{Formula}} \boxed{\sum_{x1=0}^{9}(10-x1)\sum_{y1=0}^{4}(5-y1)}$$

$$\sum_{x1=0}^{9}(10-x1)=10+9+8+7+6+5+4+3+2+1=5(11)=55$$

$$\sum_{y1=0}^{4}(5-y1)=5+4+3+2+1=2.5(6)=15$$

For one layer: $55(15)=825$ data sets
For four layers: $4(825)=3{,}300$ data sets

Mesh Size = 12 x 6

100 percent delamination

```
for x1 = 0 → 11
    for y1 = 0 → 5
        for x2 = x1+1 → 12
            for y2 = y1+1 → 6
```

$$\xrightarrow{\text{Formula}} \boxed{\sum_{x1=0}^{11}(12-x1)\sum_{y1=0}^{5}(6-y1)}$$

$$\sum_{x1=0}^{11}(12-x1)=12+11+10+9+8+7+6+5+4+3+2+1=6(13)=78$$

$$\sum_{y1=0}^{5}(6-y1)=6+5+4+3+2+1=3(7)=21$$

For one laye: $78(21)=1{,}638$ data sets
For four layers: $4(1{,}638)=6{,}552$ data sets

Mesh Size = 16 x 8

100 percent delamination

```
for x1 = 0 → 15
    for y1 = 0 → 7
        for x2 = x1+1 → 16
            for y2 = y1+1 → 8
```

$$\xrightarrow{\text{Formula}} \boxed{\sum_{x1=0}^{15}(16-x1)\sum_{y1=0}^{7}(8-y1)}$$

$$\sum_{x1=0}^{15}(16-x1)=16+15+14+13+12+11+10+9+8+7+6+5+4+3+2+1=8(17)=136$$

$$\sum_{y1=0}^{7}(8-y1)=8+7+6+5+4+3+2+1=4(9)=36$$

For one layer: $136(36)=4{,}896$ data sets
For four layers: $4(4{,}896)=19{,}584$ data sets

Mesh Size = 18 x 9

100 percent delamination

```
for x1 = 0 → 17
    for y1 = 0 → 8
        for x2 = x1+1 → 18
            for y2 = y1+1 → 9
```

$$\xrightarrow{\text{Formula}} \boxed{\sum_{x1=0}^{17}(18-x1)\sum_{y1=0}^{8}(9-y1)}$$

$$\sum_{x1=0}^{17}(18-x1)=18+17+16+15+14+13+12+11+10+9+8+7+6+5+4+3+2+1$$
$$=9(19)=171$$

$$\sum_{y1=0}^{8}(9-y1)=9+8+7+6+5+4+3+2+1=4.5(10)=45$$

For one layer: $171(45)=7{,}695$ data sets
For four layers: $4(7{,}695)=30{,}780$ data sets

Mesh Size = 20 x 10

20 percent delamination

```
for x1 = 8 → 11
    for y1 = 4 → 5
        for x2 = x1+1 → 12
            for y2 = y1+1 → 6
```

$$\xrightarrow{\text{Formula}} \boxed{\sum_{x1=8}^{11}(12-x1)\sum_{y1=4}^{5}(6-y1)}$$

$$\sum_{x1=8}^{11}(12-x1)=4+3+2+1=2(5)=10$$

$$\sum_{y1=4}^{5}(6-y1)=2+1=3$$

For one layer: 10(3) = 30 data sets
For four layers: 4(30) = 120 data sets

40 percent delamination

```
for x1 = 6 → 13
    for y1 = 3 → 6
        for x2 = x1+1 → 14
            for y2 = y1+1 → 7
```

$$\xrightarrow{\text{Formula}} \boxed{\sum_{x1=6}^{13}(14-x1)\sum_{y1=3}^{6}(7-y1)}$$

$$\sum_{x1=6}^{13}(14-x1)=8+7+6+5+4+3+2+1=4(9)=36$$

$$\sum_{y1=3}^{6}(7-y1)=4+3+2+1=2(5)=10$$

For one layer: 36(10) = 360 data sets
For four layers: 4(360) = 1,440 data sets

60 percent delamination

```
for x1 = 4 → 15
    for y1 = 2 → 7
        for x2 = x1+1 → 16
            for y2 = y1+1 → 8
```

$$\xrightarrow{\text{Formula}} \boxed{\sum_{x1=4}^{15}(16-x1)\sum_{y1=2}^{7}(8-y1)}$$

$$\sum_{x1=4}^{15}(16-x1)=12+11+10+9+8+7+6+5+4+3+2+1=6(13)=78$$

$$\sum_{y1=2}^{7}(8-y1)=6+5+4+3+2+1=3(7)=21$$

For one layer: 78(21) = 1,638 data sets
For four layers: 4(1,638) = 6,552 data sets

80 percent delamination

```
for x1 = 2 → 17
    for y1 = 1 → 8
        for x2 = x1+1 → 18
            for y2 = y1+1 → 9
```

$$\xrightarrow{\text{Formula}} \boxed{\sum_{x1=2}^{17}(18-x1)\sum_{y1=1}^{8}(9-y1)}$$

$$\sum_{x1=2}^{17}(18-x1)=16+15+14+13+12+11+10+9+8+7+6+5+4+3+2+1=8(17)=136$$

$$\sum_{y1=1}^{8}(9-y1)=8+7+6+5+4+3+2+1=4(9)=36$$

For one layer: $136(36)=4{,}896$ data sets
For four layers: $4(4{,}896)=19{,}584$ data sets

100 percent delamination

```
for x1 = 0 → 19
    for y1 = 0 → 9
        for x2 = x1+1 → 20
            for y2 = y1+1 → 10
```

$$\xrightarrow{\text{Formula}} \boxed{\sum_{x1=0}^{19}(20-x1)\sum_{y1=0}^{9}(10-y1)}$$

$$\begin{aligned}\sum_{x1=0}^{19}(20-x1)&=20+19+18+17+16+15+14+13+12+11+10+9+8+7+6+5+4\\&\quad+3+2+1\\&=10(21)=210\end{aligned}$$

$$\sum_{y1=0}^{9}(10-y1)=10+9+8+7+6+5+4+3+2+1=5(11)=55$$

For one layer: $210(55)=11{,}550$ data sets
For four layers: $4(11{,}550)=46{,}200$ data sets

APPENDIX E

MATLAB FOR-LOOPS CONFIGURATION

MATLAB FOR-Loops Configuration of Plate Dimension 8cm x 4cm with Mesh Size 20x10 (100 Percent Delamination)

(Refer to Figure 29)

FOR-Loops

for $x1 = 0 \rightarrow 19$; for $y1 = 0 \rightarrow 9$; for $x2 = x1+1 \rightarrow 20$; for $y2 = y1+1 \rightarrow 10$

For x1=0

$$\left\{\begin{array}{lcccccccccccccccccccc}
y1= & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
x2= & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\
y2= & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10
\end{array}\right.$$

$$\left\{\begin{array}{lcccccccccccccccccccc}
y1= & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
x2= & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\
y2= & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10
\end{array}\right.$$

$$\left\{\begin{array}{lcccccccccccccccccccc}
y1= & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\
x2= & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\
y2= & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10
\end{array}\right.$$

$$\left\{\begin{array}{lcccccccccccccccccccc}
y1= & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\
x2= & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\
y2= & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10
\end{array}\right.$$

$$\left\{\begin{array}{lcccccccccccccccccccc}
y1= & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\
x2= & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\
y2= & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10
\end{array}\right.$$

$$\left\{\begin{array}{lcccccccccccccccccccc}
y1= & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\
x2= & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\
y2= & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10
\end{array}\right.$$

$$\left\{\begin{array}{lcccccccccccccccccccc}
y1= & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\
x2= & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\
y2= & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 \\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10
\end{array}\right.$$

$$\left\{\begin{array}{lcccccccccccccccccccc}
y1= & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 \\
x2= & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\
y2= & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10
\end{array}\right.$$

$$\left\{\begin{array}{lcccccccccccccccccccc}
y1= & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\
x2= & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\
y2= & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 \\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10
\end{array}\right.$$

$$\left\{\begin{array}{lcccccccccccccccccccc}
y1= & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 \\
x2= & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\
y2= & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10
\end{array}\right.$$

$$(20-x1)\sum_{y1=0}^{9}(10-y1)=20(55)=1100$$

For x1=1

$$\left\{\begin{array}{lccccccccccccccccccc}
y1= & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
x2= & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\
y2= & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10
\end{array}\right.$$

$$\left\{\begin{array}{lccccccccccccccccccc}
y1= & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
x2= & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\
y2= & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10
\end{array}\right.$$

$$\left\{\begin{array}{lccccccccccccccccccc}
y1= & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\
x2= & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\
y2= & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10
\end{array}\right.$$

$$\left\{\begin{array}{lccccccccccccccccccc}
y1= & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\
x2= & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\
y2= & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10
\end{array}\right.$$

$$\left\{\begin{array}{lccccccccccccccccccc}
y1= & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\
x2= & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\
y2= & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10
\end{array}\right.$$

$$\left\{\begin{array}{lccccccccccccccccccc}
y1= & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\
x2= & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\
y2= & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10
\end{array}\right.$$

$$\left\{\begin{array}{lccccccccccccccccccc}
y1= & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\
x2= & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\
y2= & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 \\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10
\end{array}\right.$$

$$\left\{\begin{array}{lccccccccccccccccccc}
y1= & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 \\
x2= & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\
y2= & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10
\end{array}\right.$$

$$\left\{\begin{array}{lccccccccccccccccccc}
y1= & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\
x2= & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\
y2= & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 \\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10
\end{array}\right.$$

$$\left\{\begin{array}{lccccccccccccccccccc}
y1= & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 \\
x2= & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\
y2= & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10
\end{array}\right.$$

$$(20-x1)\sum_{y1=0}^{9}(10-y1)=19(55)=1045$$

For x1=2

$$\left\{\begin{array}{lcccccccccccccccccc} y1 = & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ x2 = & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2 = & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcccccccccccccccccc} y1 = & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ x2 = & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2 = & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcccccccccccccccccc} y1 = & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ x2 = & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2 = & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcccccccccccccccccc} y1 = & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\ x2 = & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2 = & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcccccccccccccccccc} y1 = & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ x2 = & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2 = & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcccccccccccccccccc} y1 = & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\ x2 = & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2 = & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcccccccccccccccccc} y1 = & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ x2 = & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2 = & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcccccccccccccccccc}
y1= & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 \\
x2= & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\
y2= & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10
\end{array}\right.$$

$$\left\{\begin{array}{lcccccccccccccccccc}
y1= & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\
x2= & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\
y2= & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 \\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10
\end{array}\right.$$

$$\left\{\begin{array}{lcccccccccccccccccc}
y1= & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 \\
x2= & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\
y2= & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10
\end{array}\right.$$

$$(20-x1)\sum_{y1=0}^{9}(10-y1)=18(55)=990$$

For x1=3

$$\left\{\begin{array}{lccccccccccccccccc}
y1= & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
x2= & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\
y2= & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10
\end{array}\right.$$

$$\left\{\begin{array}{lccccccccccccccccc}
y1= & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
x2= & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\
y2= & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10
\end{array}\right.$$

$$\left\{\begin{array}{lccccccccccccccccc}
y1= & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\
x2= & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\
y2= & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10
\end{array}\right.$$

$$\left\{\begin{array}{lccccccccccccccccc}
y1= & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\
x2= & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\
y2= & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10
\end{array}\right.$$

$$\left\{\begin{array}{lccccccccccccccccc}
y1= & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\
x2= & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\
y2= & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10
\end{array}\right.$$

$$\left\{\begin{array}{lccccccccccccccccc}
y1= & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\
x2= & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\
y2= & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10
\end{array}\right.$$

$$\left\{\begin{array}{lccccccccccccccccc}
y1= & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\
x2= & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\
y2= & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 \\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10
\end{array}\right.$$

$$\left\{\begin{array}{lccccccccccccccccc}
y1= & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 \\
x2= & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\
y2= & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10
\end{array}\right.$$

$$\left\{\begin{array}{lccccccccccccccccc}
y1= & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\
x2= & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\
y2= & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 \\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10
\end{array}\right.$$

$$\left\{\begin{array}{lccccccccccccccccc}
y1= & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 \\
x2= & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\
y2= & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10
\end{array}\right.$$

$$(20-x1)\sum_{y1=0}^{9}(10-y1)=17(55)=935$$

For x1=4

$$\left\{\begin{array}{lcccccccccccccccc}
y1= & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
x2= & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\
y2= & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10
\end{array}\right.$$

$$\left\{\begin{array}{lcccccccccccccccc}
y1 = & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\
x2 = & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20\\
y2 = & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2\\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\\
 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10
\end{array}\right.$$

$$\left\{\begin{array}{lcccccccccccccccc}
y1 = & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2\\
x2 = & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20\\
y2 = & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3\\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\\
 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10
\end{array}\right.$$

$$\left\{\begin{array}{lcccccccccccccccc}
y1 = & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3\\
x2 = & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20\\
y2 = & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4\\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\\
 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10
\end{array}\right.$$

$$\left\{\begin{array}{lcccccccccccccccc}
y1 = & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4\\
x2 = & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20\\
y2 = & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5\\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\\
 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10
\end{array}\right.$$

$$\left\{\begin{array}{lcccccccccccccccc}
y1 = & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5\\
x2 = & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20\\
y2 = & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6\\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\\
 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10
\end{array}\right.$$

$$\left\{\begin{array}{lcccccccccccccccc}
y1 = & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6\\
x2 = & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20\\
y2 = & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7\\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\\
 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10
\end{array}\right.$$

$$\left\{\begin{array}{lcccccccccccccccc}
y1 = & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7\\
x2 = & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20\\
y2 = & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8\\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\\
 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10
\end{array}\right.$$

$$\left\{\begin{array}{lcccccccccccccccc}
y1 = & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8\\
x2 = & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20\\
y2 = & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9\\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\\
 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10
\end{array}\right.$$

$$\left\{\begin{array}{lcccccccccccccccc} y1= & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 \\ x2= & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$(20-x1)\sum_{y1=0}^{9}(10-y1)=16(55)=880$$

<u>For x1=5</u>

$$\left\{\begin{array}{lccccccccccccccc} y1= & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ x2= & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lccccccccccccccc} y1= & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ x2= & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lccccccccccccccc} y1= & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ x2= & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lccccccccccccccc} y1= & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\ x2= & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lccccccccccccccc} y1= & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ x2= & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lccccccccccccccc} y1= & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\ x2= & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lccccccccccccccc} y1= & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ x2= & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lccccccccccccccc} y1= & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 \\ x2= & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lccccccccccccccc} y1= & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\ x2= & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lccccccccccccccc} y1= & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 \\ x2= & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$(20-x1)\sum_{y1=0}^{9}(10-y1)=15(55)=825$$

<u>For x1=6</u>

$$\left\{\begin{array}{lcccccccccccccc} y1= & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ x2= & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcccccccccccccc} y1= & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ x2= & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcccccccccccccc} y1= & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ x2= & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcccccccccccccc} y1= & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\ x2= & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcccccccccccccc} y1= & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ x2= & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcccccccccccccc} y1= & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\ x2= & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcccccccccccccc} y1= & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ x2= & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcccccccccccccc} y1= & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 \\ x2= & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcccccccccccccc} y1= & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\ x2= & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcccccccccccccc} y1= & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 \\ x2= & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$(20-x1)\sum_{y1=0}^{9}(10-y1)=14(55)=770$$

For x1=7

$$\left\{\begin{array}{lccccccccccccc} y1= & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ x2= & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lccccccccccccc} y1= & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ x2= & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lccccccccccccc} y1= & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ x2= & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lccccccccccccc} y1= & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\ x2= & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lccccccccccccc} y1= & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ x2= & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lccccccccccccc} y1= & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\ x2= & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lccccccccccccc} y1= & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ x2= & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lccccccccccccc} y1= & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 \\ x2= & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lccccccccccccc} y1= & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\ x2= & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lccccccccccccc} y1= & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 \\ x2= & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$(20-x1)\sum_{y1=0}^{9}(10-y1)=13(55)=715$$

<u>For x1=8</u>

$$\left\{\begin{array}{lcccccccccccc} y1= & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ x2= & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcccccccccccc} y1= & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ x2= & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcccccccccccc} y1= & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ x2= & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcccccccccccc} y1= & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\ x2= & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcccccccccccc} y1= & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ x2= & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcccccccccccc} y1= & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\ x2= & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcccccccccccc} y1= & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ x2= & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcccccccccccc} y1= & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 \\ x2= & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcccccccccccc} y1= & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\ x2= & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcccccccccccc} y1= & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 \\ x2= & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$(20-x1)\sum_{y1=0}^{9}(10-y1)=12(55)=660$$

<u>For x1=9</u>

$$\begin{array}{lccccccccccc} y1= & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ x2= & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}$$

$$\left\{\begin{array}{lccccccccccc} y1 = & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ x2 = & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2 = & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lccccccccccc} y1 = & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ x2 = & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2 = & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lccccccccccc} y1 = & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\ x2 = & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2 = & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lccccccccccc} y1 = & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ x2 = & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2 = & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lccccccccccc} y1 = & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\ x2 = & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2 = & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lccccccccccc} y1 = & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ x2 = & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2 = & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lccccccccccc} y1 = & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 \\ x2 = & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2 = & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lccccccccccc} y1 = & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\ x2 = & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2 = & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lccccccccccc} y1= & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 \\ x2= & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$(20-x1)\sum_{y1=0}^{9}(10-y1)=11(55)=605$$

For x1=10

$$\left\{\begin{array}{lcccccccccc} y1= & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ x2= & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcccccccccc} y1= & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ x2= & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcccccccccc} y1= & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ x2= & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcccccccccc} y1= & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\ x2= & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcccccccccc} y1= & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ x2= & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcccccccccc} y1= & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\ x2= & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcccccccccc} y1= & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ x2= & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2\ = & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcccccccccc} y1= & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 \\ x2= & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2\ = & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcccccccccc} y1= & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\ x2= & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2\ = & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcccccccccc} y1= & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 \\ x2= & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2\ = & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$(20-x1)\sum_{y1=0}^{9}(10-y1)=10(55)=550$$

<u>For x1=11</u>

$$\left\{\begin{array}{lccccccccc} y1= & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ x2= & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2\ = & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lccccccccc} y1= & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ x2= & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2\ = & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lccccccccc} y1= & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ x2= & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2\ = & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{llllllllll} y1= & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\ x2= & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2\ = & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{llllllllll} y1= & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ x2= & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2\ = & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{llllllllll} y1= & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\ x2= & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2\ = & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{llllllllll} y1= & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ x2= & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2\ = & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{llllllllll} y1= & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 \\ x2= & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2\ = & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{llllllllll} y1= & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\ x2= & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2\ = & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{llllllllll} y1= & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 \\ x2= & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2\ = & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$(20-x1)\sum_{y1=0}^{9}(10-y1)=9(55)=495$$

For x1=12

$$\left\{\begin{array}{lcccccccc} y1= & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ x2= & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcccccccc} y1= & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ x2= & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcccccccc} y1= & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ x2= & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcccccccc} y1= & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\ x2= & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcccccccc} y1= & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ x2= & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcccccccc} y1= & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\ x2= & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcccccccc} y1= & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ x2= & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcccccccc} y1= & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 \\ x2= & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcccccccc} y1= & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\ x2= & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcccccccc} y1= & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 \\ x2= & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$(20-x1)\sum_{y1=0}^{9}(10-y1)=8(55)=440$$

For x1=13

$$\left\{\begin{array}{lccccccc} y1= & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ x2= & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lccccccc} y1= & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ x2= & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lccccccc} y1= & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ x2= & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lccccccc} y1= & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\ x2= & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lccccccc} y1= & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ x2= & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lccccccc} y1= & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\ x2= & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lccccccc} y1= & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ x2= & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 7 & 7 & 7 & 7 & 7 & 7 & 7 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lccccccc} y1= & 7 & 7 & 7 & 7 & 7 & 7 & 7 \\ x2= & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lccccccc} y1= & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\ x2= & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 9 & 9 & 9 & 9 & 9 & 9 & 9 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lccccccc} y1= & 9 & 9 & 9 & 9 & 9 & 9 & 9 \\ x2= & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 10 & 10 & 10 & 10 & 10 & 10 & 10 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$(20-x1)\sum_{y1=0}^{9}(10-y1)=7(55)=385$$

For x1=14

$$\left\{\begin{array}{lcccccc} y1= & 0 & 0 & 0 & 0 & 0 & 0 \\ x2= & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 1 & 1 & 1 & 1 & 1 & 1 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcccccc} y1= & 1 & 1 & 1 & 1 & 1 & 1 \\ x2= & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 2 & 2 & 2 & 2 & 2 & 2 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcccccc} y1= & 2 & 2 & 2 & 2 & 2 & 2 \\ x2= & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 3 & 3 & 3 & 3 & 3 & 3 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcccccc} y1= & 3 & 3 & 3 & 3 & 3 & 3 \\ x2= & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 4 & 4 & 4 & 4 & 4 & 4 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcccccc} y1= & 4 & 4 & 4 & 4 & 4 & 4 \\ x2= & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 5 & 5 & 5 & 5 & 5 & 5 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcccccc} y1= & 5 & 5 & 5 & 5 & 5 & 5 \\ x2= & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 6 & 6 & 6 & 6 & 6 & 6 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcccccc} y1= & 6 & 6 & 6 & 6 & 6 & 6 \\ x2= & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 7 & 7 & 7 & 7 & 7 & 7 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcccccc} y1= & 7 & 7 & 7 & 7 & 7 & 7 \\ x2= & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 8 & 8 & 8 & 8 & 8 & 8 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcccccc} y1= & 8 & 8 & 8 & 8 & 8 & 8 \\ x2= & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 9 & 9 & 9 & 9 & 9 & 9 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcccccc} y1= & 9 & 9 & 9 & 9 & 9 & 9 \\ x2= & 15 & 16 & 17 & 18 & 19 & 20 \\ y2= & 10 & 10 & 10 & 10 & 10 & 10 \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$(20-x1)\sum_{y1=0}^{9}(10-y1)=6(55)=330$$

For x1=15

$$\left\{\begin{array}{llllll} y1= & 0 & 0 & 0 & 0 & 0 \\ x2= & 16 & 17 & 18 & 19 & 20 \\ y2= & 1 & 1 & 1 & 1 & 1 \\ & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{llllll} y1= & 1 & 1 & 1 & 1 & 1 \\ x2= & 16 & 17 & 18 & 19 & 20 \\ y2= & 2 & 2 & 2 & 2 & 2 \\ & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{llllll} y1= & 2 & 2 & 2 & 2 & 2 \\ x2= & 16 & 17 & 18 & 19 & 20 \\ y2= & 3 & 3 & 3 & 3 & 3 \\ & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{llllll} y1= & 3 & 3 & 3 & 3 & 3 \\ x2= & 16 & 17 & 18 & 19 & 20 \\ y2= & 4 & 4 & 4 & 4 & 4 \\ & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{llllll} y1= & 4 & 4 & 4 & 4 & 4 \\ x2= & 16 & 17 & 18 & 19 & 20 \\ y2= & 5 & 5 & 5 & 5 & 5 \\ & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{llllll} y1= & 5 & 5 & 5 & 5 & 5 \\ x2= & 16 & 17 & 18 & 19 & 20 \\ y2= & 6 & 6 & 6 & 6 & 6 \\ & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{llllll} y1= & 6 & 6 & 6 & 6 & 6 \\ x2= & 16 & 17 & 18 & 19 & 20 \\ y2= & 7 & 7 & 7 & 7 & 7 \\ & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{llllll} y1= & 7 & 7 & 7 & 7 & 7 \\ x2= & 16 & 17 & 18 & 19 & 20 \\ y2= & 8 & 8 & 8 & 8 & 8 \\ & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{llllll} y1= & 8 & 8 & 8 & 8 & 8 \\ x2= & 16 & 17 & 18 & 19 & 20 \\ y2= & 9 & 9 & 9 & 9 & 9 \\ & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{llllll} y1= & 9 & 9 & 9 & 9 & 9 \\ x2= & 16 & 17 & 18 & 19 & 20 \\ y2= & 10 & 10 & 10 & 10 & 10 \\ & \vdots & \vdots & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 & 10 & 10 \end{array}\right.$$

$$(20-x1)\sum_{y1=0}^{9}(10-y1)=5(55)=275$$

For x1=16

$$\left\{\begin{array}{lcccc} y1= & 0 & 0 & 0 & 0\\ x2= & 17 & 18 & 19 & 20\\ y2= & 1 & 1 & 1 & 1\\ & \vdots & \vdots & \vdots & \vdots\\ & 10 & 10 & 10 & 10\end{array}\right.$$

$$\left\{\begin{array}{lcccc} y1= & 1 & 1 & 1 & 1\\ x2= & 17 & 18 & 19 & 20\\ y2= & 2 & 2 & 2 & 2\\ & \vdots & \vdots & \vdots & \vdots\\ & 10 & 10 & 10 & 10\end{array}\right.$$

$$\left\{\begin{array}{lcccc} y1= & 2 & 2 & 2 & 2\\ x2= & 17 & 18 & 19 & 20\\ y2= & 3 & 3 & 3 & 3\\ & \vdots & \vdots & \vdots & \vdots\\ & 10 & 10 & 10 & 10\end{array}\right.$$

$$\left\{\begin{array}{lcccc} y1= & 3 & 3 & 3 & 3\\ x2= & 17 & 18 & 19 & 20\\ y2= & 4 & 4 & 4 & 4\\ & \vdots & \vdots & \vdots & \vdots\\ & 10 & 10 & 10 & 10\end{array}\right.$$

$$\left\{\begin{array}{lcccc} y1= & 4 & 4 & 4 & 4\\ x2= & 17 & 18 & 19 & 20\\ y2= & 5 & 5 & 5 & 5\\ & \vdots & \vdots & \vdots & \vdots\\ & 10 & 10 & 10 & 10\end{array}\right.$$

$$\left\{\begin{array}{lcccc} y1= & 5 & 5 & 5 & 5\\ x2= & 17 & 18 & 19 & 20\\ y2= & 6 & 6 & 6 & 6\\ & \vdots & \vdots & \vdots & \vdots\\ & 10 & 10 & 10 & 10\end{array}\right.$$

$$\left\{\begin{array}{lcccc} y1= & 6 & 6 & 6 & 6\\ x2= & 17 & 18 & 19 & 20\\ y2= & 7 & 7 & 7 & 7\\ & \vdots & \vdots & \vdots & \vdots\\ & 10 & 10 & 10 & 10\end{array}\right.$$

$$\left\{\begin{array}{lcccc} y1= & 7 & 7 & 7 & 7\\ x2= & 17 & 18 & 19 & 20\\ y2= & 8 & 8 & 8 & 8\\ & \vdots & \vdots & \vdots & \vdots\\ & 10 & 10 & 10 & 10\end{array}\right.$$

$$\left\{\begin{array}{lcccc} y1= & 8 & 8 & 8 & 8\\ x2= & 17 & 18 & 19 & 20\\ y2= & 9 & 9 & 9 & 9\\ & \vdots & \vdots & \vdots & \vdots\\ & 10 & 10 & 10 & 10\end{array}\right.$$

$$\left\{\begin{array}{lcccc} y1= & 9 & 9 & 9 & 9\\ x2= & 17 & 18 & 19 & 20\\ y2= & 10 & 10 & 10 & 10\\ & \vdots & \vdots & \vdots & \vdots\\ & 10 & 10 & 10 & 10\end{array}\right.$$

$$(20-x1)\sum_{y1=0}^{9}(10-y1)=4(55)=220$$

For x1=17

$$\left\{\begin{array}{llll} y1= & 0 & 0 & 0 \\ x2= & 18 & 19 & 20 \\ y2= & 1 & 1 & 1 \\ & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{llll} y1= & 1 & 1 & 1 \\ x2= & 18 & 19 & 20 \\ y2= & 2 & 2 & 2 \\ & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{llll} y1= & 2 & 2 & 2 \\ x2= & 18 & 19 & 20 \\ y2= & 3 & 3 & 3 \\ & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{llll} y1= & 3 & 3 & 3 \\ x2= & 18 & 19 & 20 \\ y2= & 4 & 4 & 4 \\ & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{llll} y1= & 4 & 4 & 4 \\ x2= & 18 & 19 & 20 \\ y2= & 5 & 5 & 5 \\ & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{llll} y1= & 5 & 5 & 5 \\ x2= & 18 & 19 & 20 \\ y2= & 6 & 6 & 6 \\ & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{llll} y1= & 6 & 6 & 6 \\ x2= & 18 & 19 & 20 \\ y2= & 7 & 7 & 7 \\ & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{llll} y1= & 7 & 7 & 7 \\ x2= & 18 & 19 & 20 \\ y2= & 8 & 8 & 8 \\ & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{llll} y1= & 8 & 8 & 8 \\ x2= & 18 & 19 & 20 \\ y2= & 9 & 9 & 9 \\ & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{llll} y1= & 9 & 9 & 9 \\ x2= & 18 & 19 & 20 \\ y2= & 10 & 10 & 10 \\ & \vdots & \vdots & \vdots \\ & 10 & 10 & 10 \end{array}\right.$$

$$(20-x1)\sum_{y1=0}^{9}(10-y1)=3(55)=165$$

For x1=18

$$\left\{\begin{array}{lcc} y1= & 0 & 0 \\ x2= & 19 & 20 \\ y2= & 1 & 1 \\ & \vdots & \vdots \\ & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcc} y1= & 1 & 1 \\ x2= & 19 & 20 \\ y2= & 2 & 2 \\ & \vdots & \vdots \\ & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcc} y1= & 2 & 2 \\ x2= & 19 & 20 \\ y2= & 3 & 3 \\ & \vdots & \vdots \\ & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcc} y1= & 3 & 3 \\ x2= & 19 & 20 \\ y2= & 4 & 4 \\ & \vdots & \vdots \\ & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcc} y1= & 4 & 4 \\ x2= & 19 & 20 \\ y2= & 5 & 5 \\ & \vdots & \vdots \\ & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcc} y1= & 5 & 5 \\ x2= & 19 & 20 \\ y2= & 6 & 6 \\ & \vdots & \vdots \\ & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcc} y1= & 6 & 6 \\ x2= & 19 & 20 \\ y2= & 7 & 7 \\ & \vdots & \vdots \\ & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcc} y1= & 7 & 7 \\ x2= & 19 & 20 \\ y2= & 8 & 8 \\ & \vdots & \vdots \\ & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcc} y1= & 8 & 8 \\ x2= & 19 & 20 \\ y2= & 9 & 9 \\ & \vdots & \vdots \\ & 10 & 10 \end{array}\right.$$

$$\left\{\begin{array}{lcc} y1= & 9 & 9 \\ x2= & 19 & 20 \\ y2= & 10 & 10 \\ & \vdots & \vdots \\ & 10 & 10 \end{array}\right.$$

$$(20-x1)\sum_{y1=0}^{9}(10-y1)=2(55)=110$$

For x1=19

$$\begin{cases} y1= & 0 \\ x2= & 20 \\ y2 = & 1 \\ & \vdots \\ & 10 \end{cases}$$

$$\begin{cases} y1= & 1 \\ x2= & 20 \\ y2 = & 2 \\ & \vdots \\ & 10 \end{cases}$$

$$\begin{cases} y1= & 2 \\ x2= & 20 \\ y2 = & 3 \\ & \vdots \\ & 10 \end{cases}$$

$$\begin{cases} y1= & 3 \\ x2= & 20 \\ y2 = & 4 \\ & \vdots \\ & 10 \end{cases}$$

$$\begin{cases} y1= & 4 \\ x2= & 20 \\ y2 = & 5 \\ & \vdots \\ & 10 \end{cases}$$

$$\begin{cases} y1= & 5 \\ x2= & 20 \\ y2 = & 6 \\ & \vdots \\ & 10 \end{cases}$$

$$\begin{cases} y1= & 6 \\ x2= & 20 \\ y2 = & 7 \\ & \vdots \\ & 10 \end{cases}$$

$$\begin{cases} y1= & 7 \\ x2= & 20 \\ y2 = & 8 \\ & \vdots \\ & 10 \end{cases}$$

$$\begin{cases} y1= & 8 \\ x2= & 20 \\ y2 = & 9 \\ & \vdots \\ & 10 \end{cases}$$

$$\begin{cases} y1= & 9 \\ x2= & 20 \\ y2 = & 10 \\ & \vdots \\ & 10 \end{cases}$$

$$(20 - x1)\sum_{y1=0}^{9}(10 - y1) = 1(55) = 55$$

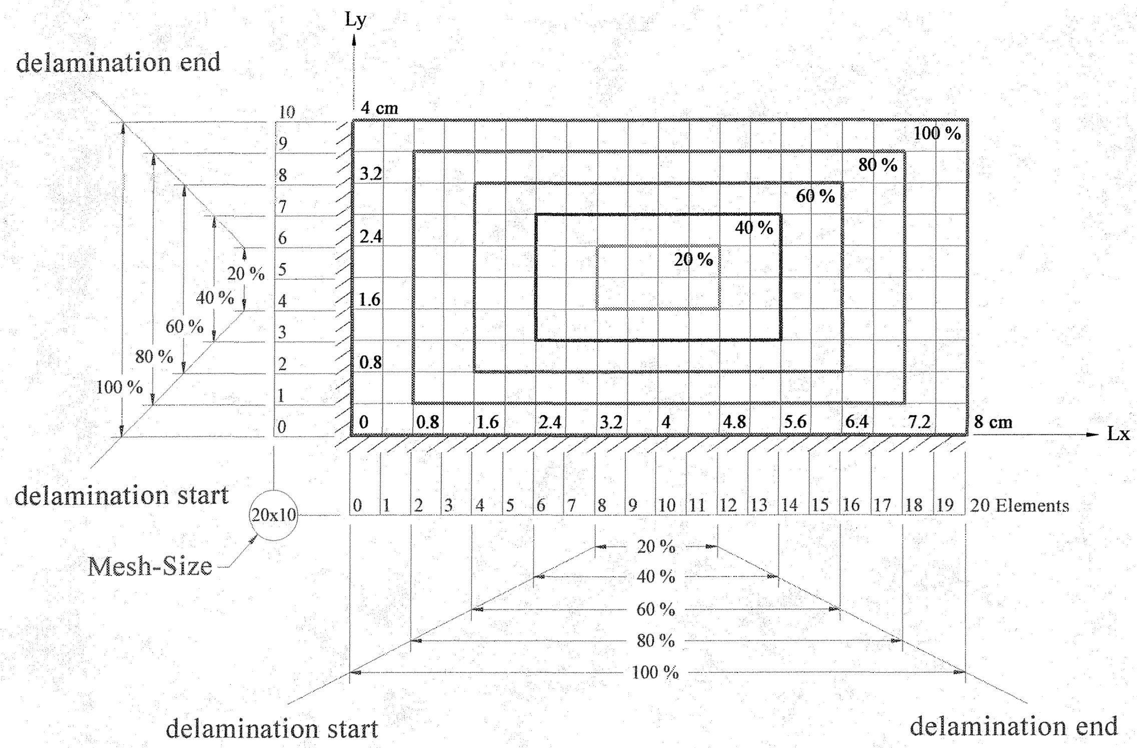

FIGURE 29. Area of delamination of plate dimension 8cm x 4cm with mesh size 20x10.

APPENDIX F

FINITE ELEMENT DATA SETS TRAINING TIME

Table 22. FEA Data Sets Collection for Plate Dimension 8 cm x 4 cm

Mesh Size	Percentage of Delamination	Files Name	Data Sets	Training Time
10x5	20%	fptfreq10x5_SID8x4_20pc_z0123	12	49 sec
	60%	fptfreq10x5_SID8x4_60pc_z0123	504	39 min
	100%	fptfreq10x5_SID8x4_100pc_z0123	3,300	5 hrs 07 min
12x6	100%	fptfreq12x6_SID8x4_100pc_z0123	6,552	25 hrs 10 min
16x8	100%	fptfreq16x8_SID8x4_100pc_z0123	19,584	14 days 17 hrs 37 min
18x9	100%	fptfreq18x9_SID8x4_100pc_z0123	30,780	40 days 17 hrs 19 min
20x10	20%	fptfreq20x10_SID8x4_20pc_z0123	120	5 hrs 09 min
	40%	fptfreq20x10_SID8x4_40pc_z0123	1,440	2 days 17 hrs 31 min
	60%	fptfreq20x10_SID8x4_60pc_z0123	6,552	13 days 9 hrs 15 min
	80%	fptfreq20x10_SID8x4_80pc_z0123	19,584	43 days 16 hrs 25 min
	100%	fptfreq20x10_SID8x4_100pc_z0123	46,200	114 days 2 hrs 25 min

Finite Element Data Sets Collection for Plate Dimension 8cm x 4cm

Mesh-size = 10x5

	File name	Data sets	Training time	Training time
20 percent				
1	fptfreq10x5_SID8x4_20pc_z0123	12	49 sec	49 sec
60 percent				
1	fptfreq10x5_SID8x4_60pc_z0123	504	2,324.4 sec	39 min
100 percent				
1	fptfreq10x5_SID8x4_100pc_z0123	3,300	18,418 sec	5 hrs 07 min

Mesh-size = 12x6

	File name	Data sets	Training time	Training time
100 percent				
1	fptfreq12x6_SID8x4_100pc_z0123	6,552	90,577 sec	25 hrs 10 min

Mesh-size = 16x8

	File name	Data sets	Training time	Training time
100 percent				
1	fptfreq16x8_SID8x4_100pc_z0_x0-2	1,620	114,700 sec	31 hrs 52 min
2	fptfreq16x8_SID8x4_100pc_z0_x3-6	1,656	105,500 sec	29 hrs 18 min
3	fptfreq16x8_SID8x4_100pc_z0_x7-15	1,620	91,277 sec	25 hrs 22 min
	Data and elapsed time for layer 0	4, 896	311,477 sec	86 hrs 31 min
4	fptfreq16x8_SID8x4_100pc_z1_x0-2	1,620	116,050 sec	32 hrs 14 min
5	fptfreq16x8_SID8x4_100pc_z1_x3-6	1,656	108,340 sec	30 hrs 06 min
6	fptfreq16x8_SID8x4_100pc_z1_x7-15	1,620	92,474 sec	25 hrs 41 min
	Data and elapsed time for layer 1	4, 896	316,864 sec	88 hrs 01 min
7	fptfreq16x8_SID8x4_100pc_z2_x0-2	1,620	115,110 sec	31 hrs 59 min
8	fptfreq16x8_SID8x4_100pc_z2_x3-6	1,656	108,670 sec	30 hrs 11 min
9	fptfreq16x8_SID8x4_100pc_z2_x7-15	1,620	91,407 sec	25 hrs 23 min
	Data and elapsed time for layer 2	4, 896	315,187 sec	87 hrs 33 min
10	fptfreq16x8_SID8x4_100pc_z3_x0-2	1,620	116,190 sec	32 hrs 17 min
11	fptfreq16x8_SID8x4_100pc_z3_x3-6	1,656	120,990 sec	33 hrs 36 min
12	fptfreq16x8_SID8x4_100pc_z3_x7-15	1,620	92,332 sec	25 hrs 39 min
	Data and elapsed time for layer 3	4, 896	329,512 sec	91 hrs 32 min
	Total data and elapsed time for 4 layers	19,584	1,273,040 sec	14 days 17 hrs 37 min

Mesh-size = 18x9

	File name	Data sets	Training time	Training time
100 percent				
1	fptfreq18x9_SID8x4_100pc_z0_x0-1	1,575	200,080 sec	55 hrs 35 min
2	fptfreq18x9_SID8x4_100pc_z0_x2-4	2,025	242,960 sec	67 hrs 29 min

3	fptfreq18x9_SID8x4_100pc_z0_x5-8	2,070	228,410 sec	63 hrs 27 min
4	fptfreq18x9_SID8x4_100pc_z0_x9-10	765	84,098 sec	23 hrs 22 min
5	fptfreq18x9_SID8x4_100pc_z0_x11-12	585	60,934 sec	16 hrs 56 min
6	fptfreq18x9_SID8x4_100pc_z0_x13-17	675	66,777 sec	18 hrs 33 min
	Data and elapsed time for layer 0	7,695	883,259 sec	245 hrs 21 min
7	fptfreq18x9_SID8x4_100pc_z1_x0-1	1,575	205,870 sec	57 hrs 11 min
8	fptfreq18x9_SID8x4_100pc_z1_x2-4	2,025	247,620 sec	68 hrs 47 min
9	fptfreq18x9_SID8x4_100pc_z1_x5-8	2,070	233,460 sec	64 hrs 51 min
10	fptfreq18x9_SID8x4_100pc_z1_x9-10	765	85,385 sec	23 hrs 43 min
11	fptfreq18x9_SID8x4_100pc_z1_x11-12	585	62,246 sec	17 hrs 17 min
12	fptfreq18x9_SID8x4_100pc_z1_x13-17	675	66,364 sec	18 hrs 26 min
	Data and elapsed time for layer 1	7,695	900,945 sec	250 hrs 16 min
13	fptfreq18x9_SID8x4_100pc_z2_x0-1	1,575	184,020 sec	51 hrs 07 min
14	fptfreq18x9_SID8x4_100pc_z2_x2-4	2,025	223,390 sec	62 hrs 03 min
15	fptfreq18x9_SID8x4_100pc_z2_x5-8	2,070	211,110 sec	58 hrs 39 min
16	fptfreq18x9_SID8x4_100pc_z2_x9-10	765	83,587 sec	23 hrs 13 min
17	fptfreq18x9_SID8x4_100pc_z2_x11-12	585	61,241 sec	17 hrs 00 min
18	fptfreq18x9_SID8x4_100pc_z2_x13-17	675	66,092 sec	18 hrs 22 min
	Data and elapsed time for layer 2	7,695	829,440 sec	230 hrs 24 min
19	fptfreq18x9_SID8x4_100pc_z3_x0-1	1,575	206,480 sec	57 hrs 21 min
20	fptfreq18x9_SID8x4_100pc_z3_x2-4	2,025	249,420 sec	69 hrs17 min
21	fptfreq18x9_SID8x4_100pc_z3_x5-8	2,070	234,670 sec	65 hrs 11 min
22	fptfreq18x9_SID8x4_100pc_z3_x9-10	765	85,140 sec	23 hrs 39 min
23	fptfreq18x9_SID8x4_100pc_z3_x11-12	585	62,643 sec	17 hrs 24 min
24	fptfreq18x9_SID8x4_100pc_z3_x13-17	675	66,347 sec	18 hrs 26 min
	Data and elapsed time for layer 3	7,695	904,700 sec	251 hrs 18 min
	Total data and elapsed time for 4 layers	30,780	3,518,344 sec	40 days 17 hrs 19 min

Mesh-size = 20x10

	File name	Data sets	Training time	Training time
20 percent				
1	fptfreq20x10_SID8x4_20pc_z0123	120	18,523.9 sec	5 hrs 09 min
40 percent				
1	fptfreq20x10_SID8x4_40pc_z0	360	58,638.9 sec	16 hrs 17 min
2	fptfreq20x10_SID8x4_40pc_z1	360	58,786.7 sec	16 hrs 20 min
3	fptfreq20x10_SID8x4_40pc_z2	360	59,814.6 sec	16 hrs 40 min
4	fptfreq20x10_SID8x4_40pc_z3	360	58,591.0 sec	16 hrs 17 min
	Total data and elapsed time for 4 layers	1,440	235,831.2 sec	2 days 17 hrs 31 min
60 percent				
1	fptfreq20x10_SID8x4_60pc_z0_x4-5	483	88,742 sec	24 hrs 39 min
2	fptfreq20x10_SID8x4_60pc_z0_x6-7	399	71,769.8 sec	19 hrs 56 min
3	fptfreq20x10_SID8x4_60pc_z0_x8-10	441	76,139.5 sec	21 hrs 9 min
4	fptfreq20x10_SID8x4_60pc_z0_x11-15	315	51,005.4 sec	14 hrs 10 min
	Data and elapsed time for layer 0	1,638	287,656.7 sec	79 hrs 54 min

5	fptfreq20x10_SID8x4_60pc_z1_x4-5	483	89,503.9 sec	24 hrs 52 min
6	fptfreq20x10_SID8x4_60pc_z1_x6-7	399	73,241.4 sec	20 hrs 21 min
7	fptfreq20x10_SID8x4_60pc_z1_x8-10	441	76,727.3 sec	21 hrs 19 min
8	fptfreq20x10_SID8x4_60pc_z1_x11-15	315	51,203.0 sec	14 hrs 13 min
	Data and elapsed time for layer 1	1,638	290,675.6 sec	80 hrs 45 min
9	fptfreq20x10_SID8x4_60pc_z2_x4-5	483	88,941.4 sec	24 hrs 42 min
10	fptfreq20x10_SID8x4_60pc_z2_x6-7	399	72,182.8 sec	20 hrs 03 min
11	fptfreq20x10_SID8x4_60pc_z2_x8-10	441	76,363.5 sec	21 hrs 13 min
12	fptfreq20x10_SID8x4_60pc_z2_x11-15	315	51,493.8 sec	14 hrs 18 min
	Data and elapsed time for layer 2	1,638	288,981.5 sec	80 hrs 16 min
13	fptfreq20x10_SID8x4_60pc_z3_x4-5	483	89,190.7 sec	24 hrs 47 min
14	fptfreq20x10_SID8x4_60pc_z3_x6-7	399	72,252.6 sec	20 hrs 04 min
15	fptfreq20x10_SID8x4_60pc_z3_x8-10	441	76,605.0 sec	21 hrs 17 min
16	fptfreq20x10_SID8x4_60pc_z3_x11-15	315	51,135.7 sec	14 hrs 12 min
	Data and elapsed time for layer 3	1,638	289,184.0 sec	80 hrs 20 min
	Total data and elapsed time for 4 layers	6,552	1,156,497.8 sec	13 days 9 hrs 15 min
80 percent				
1	fptfreq20x10_SID8x4_80pc_z0_x2	576	120,148.0 sec	33 hrs 22 min
2	fptfreq20x10_SID8x4_80pc_z0_x3	540	110,607.0 sec	30 hrs 43 min
3	fptfreq20x10_SID8x4_80pc_z0_x4	504	100,845.0 sec	28 hrs 01 min
4	fptfreq20x10_SID8x4_80pc_z0_x5	468	92,269.3 sec	25 hrs 38 min
5	fptfreq20x10_SID8x4_80pc_z0_x6	432	85,194.6 sec	23 hrs 40 min
6	fptfreq20x10_SID8x4_80pc_z0_x7	396	76,477.5 sec	21 hrs 15 min
7	fptfreq20x10_SID8x4_80pc_z0_x8	360	67,236.2 sec	18 hrs 41 min
8	fptfreq20x10_SID8x4_80pc_z0_x9	324	59,507.6 sec	16 hrs 32 min
9	fptfreq20x10_SID8x4_80pc_z0_x10	288	52,214.3 sec	14 hrs 30 min
10	fptfreq20x10_SID8x4_80pc_z0_x11	252	44,904.1 sec	12 hrs 29 min
11	fptfreq20x10_SID8x4_80pc_z0_x12	216	38,006.4 sec	10 hrs 33 min
12	fptfreq20x10_SID8x4_80pc_z0_x13	180	30,401.3 sec	8 hrs 27 min
13	fptfreq20x10_SID8x4_80pc_z0_x14	144	24,058.5 sec	6 hrs 41 min
14	fptfreq20x10_SID8x4_80pc_z0_x15	108	17,470.7 sec	4 hrs 51 min
15	fptfreq20x10_SID8x4_80pc_z0_x16	72	11,348.7 sec	3 hrs 09 min
16	fptfreq20x10_SID8x4_80pc_z0_x17	36	5,592.7 sec	1 hrs 33 min
	Data and elapsed time for layer 0	4,896	936,281.9 sec	260 hrs 05 min
1	fptfreq20x10_SID8x4_80pc_z1_x2	576	121,609.0 sec	33 hrs 47 min
2	fptfreq20x10_SID8x4_80pc_z1_x3	540	112,499.0 sec	31 hrs 15 min
3	fptfreq20x10_SID8x4_80pc_z1_x4	504	103,467.0 sec	28 hrs 44 min
4	fptfreq20x10_SID8x4_80pc_z1_x5	468	93,688.9 sec	26 hrs 01 min
5	fptfreq20x10_SID8x4_80pc_z1_x6	432	86,417.1 sec	24 hrs 00 min
6	fptfreq20x10_SID8x4_80pc_z1_x7	396	78,265.4 sec	21 hrs 44 min
7	fptfreq20x10_SID8x4_80pc_z1_x8	360	67,945.8 sec	18 hrs 52 min
8	fptfreq20x10_SID8x4_80pc_z1_x9	324	60,206.4 sec	16 hrs 43 min
9	fptfreq20x10_SID8x4_80pc_z1_x10	288	52,722.6 sec	14 hrs 39 min
10	fptfreq20x10_SID8x4_80pc_z1_x11	252	45,267.5 sec	12 hrs 34 min
11	fptfreq20x10_SID8x4_80pc_z1_x12	216	38,227.0 sec	10 hrs 37 min

12	fptfreq20x10_SID8x4_80pc_z1_x13	180	30,622.4 sec	8 hrs 30 min
13	fptfreq20x10_SID8x4_80pc_z1_x14	144	24,517.5 sec	6 hrs 49 min
14	fptfreq20x10_SID8x4_80pc_z1_x15	108	17,540.2 sec	4 hrs 52 min
15	fptfreq20x10_SID8x4_80pc_z1_x16	72	11,401.3 sec	3 hrs 10 min
16	fptfreq20x10_SID8x4_80pc_z1_x17	36	5,573.1 sec	1 hrs 33 min
	Data and elapsed time for layer 1	4,896	949,970.2 sec	263 hrs 53 min
1	fptfreq20x10_SID8x4_80pc_z2_x2	576	120,825.0 sec	33 hrs 34 min
2	fptfreq20x10_SID8x4_80pc_z2_x3	540	111,410.0 sec	30 hrs 57 min
3	fptfreq20x10_SID8x4_80pc_z2_x4	504	101,610.0 sec	28 hrs 14 min
4	fptfreq20x10_SID8x4_80pc_z2_x5	468	93,553.5 sec	25 hrs 59 min
5	fptfreq20x10_SID8x4_80pc_z2_x6	432	85,704.4 sec	23 hrs 48 min
6	fptfreq20x10_SID8x4_80pc_z2_x7	396	77,242.1 sec	21 hrs 27 min
7	fptfreq20x10_SID8x4_80pc_z2_x8	360	68,394.5 sec	19 hrs 00 min
8	fptfreq20x10_SID8x4_80pc_z2_x9	324	59,687.8 sec	16 hrs 35 min
9	fptfreq20x10_SID8x4_80pc_z2_x10	288	52,384.5 sec	14 hrs 33 min
10	fptfreq20x10_SID8x4_80pc_z2_x11	252	44,884.3 sec	12 hrs 28 min
11	fptfreq20x10_SID8x4_80pc_z2_x12	216	37,983.1 sec	10 hrs 33 min
12	fptfreq20x10_SID8x4_80pc_z2_x13	180	30,782.5 sec	8 hrs 33 min
13	fptfreq20x10_SID8x4_80pc_z2_x14	144	24,149.9 sec	6 hrs 43 min
14	fptfreq20x10_SID8x4_80pc_z2_x15	108	17,475.5 sec	4 hrs 51 min
15	fptfreq20x10_SID8x4_80pc_z2_x16	72	11,489.9 sec	3 hrs 11 min
16	fptfreq20x10_SID8x4_80pc_z2_x17	36	5,569.8 sec	1 hrs 33 min
	Data and elapsed time for layer 2	4,896	943,146.8 sec	261 hrs 59 min
1	fptfreq20x10_SID8x4_80pc_z3_x2	576	121,710.0 sec	33 hrs 48 min
2	fptfreq20x10_SID8x4_80pc_z3_x3	540	111,874.0 sec	31 hrs 05 min
3	fptfreq20x10_SID8x4_80pc_z3_x4	504	102,299.0 sec	28 hrs 25 min
4	fptfreq20x10_SID8x4_80pc_z3_x5	468	93,168.0 sec	25 hrs 53 min
5	fptfreq20x10_SID8x4_80pc_z3_x6	432	85,238.5 sec	23 hrs 41 min
6	fptfreq20x10_SID8x4_80pc_z3_x7	396	77,079.0 sec	21 hrs 25 min
7	fptfreq20x10_SID8x4_80pc_z3_x8	360	68,054.4 sec	18 hrs 54 min
8	fptfreq20x10_SID8x4_80pc_z3_x9	324	59,893.9 sec	16 hrs 38 min
9	fptfreq20x10_SID8x4_80pc_z3_x10	288	53,063.8 sec	14 hrs 44 min
10	fptfreq20x10_SID8x4_80pc_z3_x11	252	45,196.0 sec	12 hrs 33 min
11	fptfreq20x10_SID8x4_80pc_z3_x12	216	38,109.0 sec	10 hrs 35 min
12	fptfreq20x10_SID8x4_80pc_z3_x13	180	30,547.6 sec	8 hrs 29 min
13	fptfreq20x10_SID8x4_80pc_z3_x14	144	24,174.3 sec	6 hrs 43 min
14	fptfreq20x10_SID8x4_80pc_z3_x15	108	17,534.3 sec	4 hrs 52 min
15	fptfreq20x10_SID8x4_80pc_z3_x16	72	11,388.5 sec	3 hrs 10 min
16	fptfreq20x10_SID8x4_80pc_z3_x17	36	5,565.3 sec	1 hrs 33 min
	Data and elapsed time for layer 3	4,896	944,895.6 sec	262 hrs 28 min
	Total data and elapsed time for 4 layers	19,584	3,774,294.5 sec	43 days 16 hrs 25 min
100 percent				
1	fptfreq20x10_SID8x4_100pc_z0_x0	1,100	260,700 sec	72 hrs 25 min
2	fptfreq20x10_SID8x4_100pc_z0_x1	1,045	242,860 sec	67 hrs 28 min
3	fptfreq20x10_SID8x4_100pc_z0_x2	990	226,080 sec	62 hrs 48 min
4	fptfreq20x10_SID8x4_100pc_z0_x3	935	208,520 sec	57 hrs 55 min

5	fptfreq20x10_SID8x4_100pc_z0_x4	880	172,430 sec	47 hrs 54 min
6	fptfreq20x10_SID8x4_100pc_z0_x5	825	192,860 sec	53 hrs 34 min
7	fptfreq20x10_SID8x4_100pc_z0_x6	770	176,230 sec	48 hrs 57 min
8	fptfreq20x10_SID8x4_100pc_z0_x7	715	149,410 sec	41 hrs 30 min
9	fptfreq20x10_SID8x4_100pc_z0_x8	660	134,580 sec	37 hrs 23 min
10	fptfreq20x10_SID8x4_100pc_z0_x9	605	120,370 sec	33 hrs 26 min
11	fptfreq20x10_SID8x4_100pc_z0_x10	550	108,890 sec	30 hrs 15 min
12	fptfreq20x10_SID8x4_100pc_z0_x11	495	94,919 sec	26 hrs 22 min
13	fptfreq20x10_SID8x4_100pc_z0_x12	440	82,554 sec	22 hrs 56 min
14	fptfreq20x10_SID8x4_100pc_z0_x13	385	70,822 sec	19 hrs 40 min
15	fptfreq20x10_SID8x4_100pc_z0_x14	330	60,022 sec	16 hrs 40 min
16	fptfreq20x10_SID8x4_100pc_z0_x15	275	48,460 sec	13 hrs 28 min
17	fptfreq20x10_SID8x4_100pc_z0_x16	220	40,726 sec	11 hrs 19 min
18	fptfreq20x10_SID8x4_100pc_z0_x17	165	29,154 sec	08 hrs 06 min
19	fptfreq20x10_SID8x4_100pc_z0_x18	110	17,613 sec	04 hrs 54 min
20	fptfreq20x10_SID8x4_100pc_z0_x19	55	8,632 sec	02 hrs 24 min
	Data and elapsed time for layer 0	11,550	2,445,832 sec	679 hrs 24 min
21	fptfreq20x10_SID8x4_100pc_z1_x0	1,100	266,300 sec	73 hrs 58 min
22	fptfreq20x10_SID8x4_100pc_z1_x1	1,045	248,280 sec	68 hrs 58 min
23	fptfreq20x10_SID8x4_100pc_z1_x2	990	230,760 sec	64 hrs 06 min
24	fptfreq20x10_SID8x4_100pc_z1_x3	935	212,850 sec	59 hrs 08 min
25	fptfreq20x10_SID8x4_100pc_z1_x4	880	176,520 sec	49 hrs 02 min
26	fptfreq20x10_SID8x4_100pc_z1_x5	825	194,470 sec	54 hrs 01 min
27	fptfreq20x10_SID8x4_100pc_z1_x6	770	179,670 sec	49 hrs 54 min
28	fptfreq20x10_SID8x4_100pc_z1_x7	715	152,180 sec	42 hrs 16 min
29	fptfreq20x10_SID8x4_100pc_z1_x8	660	136,360 sec	37 hrs 53 min
30	fptfreq20x10_SID8x4_100pc_z1_x9	605	122,170 sec	33 hrs 56 min
31	fptfreq20x10_SID8x4_100pc_z1_x10	550	109,360 sec	30 hrs 23 min
32	fptfreq20x10_SID8x4_100pc_z1_x11	495	95,897 sec	26 hrs 38 min
33	fptfreq20x10_SID8x4_100pc_z1_x12	440	83,386 sec	23 hrs 10 min
34	fptfreq20x10_SID8x4_100pc_z1_x13	385	71,541 sec	19 hrs 52 min
35	fptfreq20x10_SID8x4_100pc_z1_x14	330	60,480 sec	16 hrs 48 min
36	fptfreq20x10_SID8x4_100pc_z1_x15	275	48,337 sec	13 hrs 26 min
37	fptfreq20x10_SID8x4_100pc_z1_x16	220	40,375 sec	11 hrs 13 min
38	fptfreq20x10_SID8x4_100pc_z1_x17	165	29,562 sec	08 hrs 13 min
39	fptfreq20x10_SID8x4_100pc_z1_x18	110	17,694 sec	04 hrs 55 min
40	fptfreq20x10_SID8x4_100pc_z1_x19	55	8,555 sec	02 hrs 23 min
	Data and elapsed time for layer 1	11,550	2,484,747 sec	690 hrs 12 min
41	fptfreq20x10_SID8x4_100pc_z2_x0	1,100	266,500 sec	74 hrs 02 min
42	fptfreq20x10_SID8x4_100pc_z2_x1	1,045	248,060 sec	68 hrs 54 min
43	fptfreq20x10_SID8x4_100pc_z2_x2	990	230,370 sec	64 hrs 00 min
44	fptfreq20x10_SID8x4_100pc_z2_x3	935	211,240 sec	58 hrs 41 min
45	fptfreq20x10_SID8x4_100pc_z2_x4	880	174,630 sec	48 hrs 30 min
46	fptfreq20x10_SID8x4_100pc_z2_x5	825	195,950 sec	54 hrs 26 min
47	fptfreq20x10_SID8x4_100pc_z2_x6	770	177,040 sec	49 hrs 11 min
48	fptfreq20x10_SID8x4_100pc_z2_x7	715	150,950 sec	41 hrs 56 min
49	fptfreq20x10_SID8x4_100pc_z2_x8	660	135,440 sec	37 hrs 37 min

50	fptfreq20x10_SID8x4_100pc_z2_x9	605	121,140 sec	33 hrs 39 min
51	fptfreq20x10_SID8x4_100pc_z2_x10	550	109,420 sec	30 hrs 24 min
52	fptfreq20x10_SID8x4_100pc_z2_x11	495	94,960 sec	26 hrs 23 min
53	fptfreq20x10_SID8x4_100pc_z2_x12	440	82,777 sec	23 hrs 00 min
54	fptfreq20x10_SID8x4_100pc_z2_x13	385	70,963 sec	19 hrs 43 min
55	fptfreq20x10_SID8x4_100pc_z2_x14	330	60,245 sec	16 hrs 44 min
56	fptfreq20x10_SID8x4_100pc_z2_x15	275	48,689 sec	13 hrs 31 min
57	fptfreq20x10_SID8x4_100pc_z2_x16	220	40,237 sec	11 hrs 11 min
58	fptfreq20x10_SID8x4_100pc_z2_x17	165	29,270 sec	08 hrs 08 min
59	fptfreq20x10_SID8x4_100pc_z2_x18	110	17,618 sec	04 hrs 54 min
60	fptfreq20x10_SID8x4_100pc_z2_x19	55	8,550 sec	02 hrs 22 min
	Data and elapsed time for layer 2	11,550	2,474,049 sec	687 hrs 14 min
61	fptfreq20x10_SID8x4_100pc_z3_x0	1,100	238,660 sec	66 hrs 18 min
62	fptfreq20x10_SID8x4_100pc_z3_x1	1,045	247,470 sec	68 hrs 45 min
63	fptfreq20x10_SID8x4_100pc_z3_x2	990	230,060 sec	63 hrs 54 min
64	fptfreq20x10_SID8x4_100pc_z3_x3	935	212,530 sec	59 hrs 02 min
65	fptfreq20x10_SID8x4_100pc_z3_x4	880	176,570 sec	49 hrs 03 min
66	fptfreq20x10_SID8x4_100pc_z3_x5	825	194,260 sec	53 hrs 58 min
67	fptfreq20x10_SID8x4_100pc_z3_x6	770	177,880 sec	49 hrs 25 min
68	fptfreq20x10_SID8x4_100pc_z3_x7	715	150,510 sec	41 hrs 49 min
69	fptfreq20x10_SID8x4_100pc_z3_x8	660	136,550 sec	37 hrs 56 min
70	fptfreq20x10_SID8x4_100pc_z3_x9	605	121,880 sec	33 hrs 51 min
71	fptfreq20x10_SID8x4_100pc_z3_x10	550	108,810 sec	30 hrs 14 min
72	fptfreq20x10_SID8x4_100pc_z3_x11	495	95,959 sec	26 hrs 39 min
73	fptfreq20x10_SID8x4_100pc_z3_x12	440	83,251 sec	23 hrs 08 min
74	fptfreq20x10_SID8x4_100pc_z3_x13	385	71,286 sec	19 hrs 48 min
75	fptfreq20x10_SID8x4_100pc_z3_x14	330	59,972 sec	16 hrs 40 min
76	fptfreq20x10_SID8x4_100pc_z3_x15	275	52,061 sec	14 hrs 28 min
77	fptfreq20x10_SID8x4_100pc_z3_x16	220	40,283 sec	11 hrs 11 min
78	fptfreq20x10_SID8x4_100pc_z3_x17	165	28,943 sec	08 hrs 02 min
79	fptfreq20x10_SID8x4_100pc_z3_x18	110	17,650 sec	04 hrs 54 min
80	fptfreq20x10_SID8x4_100pc_z3_x19	55	9,113 sec	02 hrs 32 min
	Data and elapsed time for layer 3	11,550	2,453,698 sec	681 hrs 35 min
	Total data and elapsed time for 4 layers	46,200	9,858,326 sec	114 days 2 hrs 25 min

REFERENCES

REFERENCES

1. Della, C.N., and Shu, D.W., 2007, "Vibration of Delaminated Composite Laminates: A Review," Trans. of the ASME, **60**, pp. 1-20.

2. Islam, A.S., and Craig, K.C., 1994, "Damage Detection in Composite Structures Using Piezoelectric Materials," Smart Materials and Structures, **3**, pp. 318-328.

3. Daniel, I., and Ishai, O., 2006, *Engineering Mechanics of Composite Materials*, 2nd ed., Oxford University Press, Oxford, UK.

4. Barbero, E.J., 1999, *Introduction to Composite Materials Design*, Taylor and Francis, Philadelphia, PA.

5. Jones, R.M., 1999, *Mechanics of Composite Materials*, 2nd ed., Taylor and Francis, Philadelphia, PA.

6. Kollar, L.P., and Springer, G.S., 2003, *Mechanics of Composite Structures*, Cambridge University Press, Cambridge, UK.

7. Le, T.H., 2004, "A Method to Detect Single and Multiple Delamination Problems Using a Combined Neural Network Technique and Genetic Algorithm Optimization," Ph.D. thesis, Department of Engineering and Applied Mathematics, Claremont Graduate University and California State University of Long Beach.

8. Kim, J.H., 2004, "Delamination Detection Problems Using a Combined Genetic Algorithm and Neural Network Technique," Master's thesis, Department of Mechanical and Aerospace Engineering, California State University of Long Beach.

9. Rao, M.A., Srinivas, J., and Murthy, B.S.N., 2004, "Damage Detection in Vibrating Bodies Using Genetic Algorithms," Computers and Structures, **82**, pp. 963-968.

10. Kim, H.S., Chattopadhyay, A., and Ghoshal, A., 2003, "Characterization of Delamination Effect on Composite Laminates Using a New Generalized Layerwise Approach," Computers and Structures, **81,** pp. 1555-1566.

11. Lu, W., 2000, "Neural Network Model for Distortional Buckling Behaviour of Cold-Formed Steel Compression Members," Master's thesis, Department of Civil and Environmental Engineering, Helsinki University of Technology.

12. Vracko, M., 2005, "Kohonen Artificial Neural Network and Counterpropagation Neural Network in Molecular Structure-Toxicity Studies," Current Computer-Aided Drug Design, **1**(1), pp. 73-78.

13. Zupan, J., Novic, M., and Ruisanchez, I., 1997, "Kohonen and Counterpropagation Artificial Neural Networks in Analytical Chemistry," Chemometrics and Intelligent Laboratory Systems, **38**, pp. 1-23.

14. Kovacs, L., and Terstyanszky, G., 2000, "Boundary Region Sensitive Classification for the Counterpropagation Neural Network," *Neural Networks, IJCNN, Proceedings of the IEEE-INNS-ENNS International Joint Conference on*, **1**, 2000, pp. 90-94.

15. Fidencio, P.H., Ruisanchez, I., and Poppi, R.J., 2001, "Application of Artificial Neural Networks to the Classification of Soils from Sao Paulo State Using Near-Infrared Spectroscopy," Analyst, **126**(12), pp. 2194-2200.

16. Zafar, M.F., Mohamad, D., and Othman, R.M., 2005, "On-Line Handwritten Character Recognition: An Implementation of Counterpropagation Neural Net," Proceedings of World Academy of Science, Engineering and Technology (PWASET), **10**, pp. 232-237.

17. Zafar, M.F., and Mohamad, D., 2001, "Counterpropagation Neural Networks for Trademark Recognition," *International Symposium on Signal Processing and its Applications (ISSPA)*, **2**, 2001, pp. 751-752.

18. Adeli, H., and Park, H.S., 1995, "Counterpropagation Neural Networks in Structural Engineering," Journal of Structural Engineering, **121**(8), pp. 1205-1212.

19. Lin, Z., Khorasani, K., and Patel, R.V., 1990, "A Counterpropagation Neural Network for Function Approximation," *Systems, Man and Cybernetics, IEEE International Conference on*, 1990, pp. 382-384.

20. Szewczyk, Z.P., and Hajela, P., 1992, "Feature-Sensitive Neural Networks in Structural Response Estimation," Department of Mechanical Engineering, Aeronautical Engineering, and Mechanics, Rensselaer Polytechnic Institute, Troy, New York.

21. Szewczyk, Z.P., and Hajela, P., 1994, "Damage Detection in Structures Based On Feature-Sensitive Neural Networks," Journal of Computing in Civil Engineering, **8**(2), pp. 163-178.

22. Szewczyk, Z.P., and Hajela, P., 1993, "Neural Network Based Selection of Dynamic System Parameters," Trans. of the CSME, **17**(4A), pp. 567-584.

23. Marwala, T., and Chakraverty, S., 2006, "Fault Classification in Structures with Incomplete Measured Data Using Autoassociative Neural Networks and Genetic Algorithm," Current Science, **90**(4), pp. 542-548.

24. Lee, S.Y., and Wooh, S.C., 2005, "Detection of Stiffness Reductions in Laminated Composite Plates from Their Dynamic Response Using the Microgenetic Algorithm," Computational Mechanics, **36**(4), pp. 320-330.

25. Chou, J.H., and Ghaboussi, J., 2001, "Genetic Algorithm in Structural Damage Detection," Computers and Structures, **79**, pp. 1335-1353.

26. Terfloth, L., and Gasteiger, J., 2001, "Neural Networks and Genetic Algorithms in Drug Design," Drug Discovery Today, **6**(15), pp. 102-108.

27. Karunaratne, R., 2001, "Optimal Design of Composite Laminates Using Genetic Algorithms," Master's thesis, Department of Mechanical and Aerospace Engineering, California State University of Long Beach.

28. Janson, D.J., and Frenzel, J.F., 1993, "Training Product Unit Neural Networks with Genetic Algorithms," *IEEE Expert*, **8**(5), pp. 26-33.