DISCRETE FIBER ANGLE AND CONTINUOUS FIBER PATH OPTIMIZATION IN COMPOSITE STRUCTURES

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

OF

MIDDLE EAST TECHNICAL UNIVERSITY

BY

HASAN İNCİ

IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

AEROSPACE ENGINEERING

FEBRUARY 2012

Approval of the thesis:

DISCRETE FIBER ANGLE AND CONTINUOUS FIBER PATH OPTIMIZATION IN COMPOSITE STRUCTURES

Submitted by **HASAN İNCİ** in partial fulfillment of the requirements for the degree of **Master of Science in Aerospace Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen ______________________
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Ozan Tekinalp ______________________
Head of Department, **Aerospace Engineering**

Assoc. Prof. Dr. Altan Kayran
Supervisor, **Aerospace Engineering Dept., METU** ______________________

Examining Committee Members:

Asst. Prof. Dr. Demirkan Çöker ______________________
Aerospace Engineering Dept., METU

Prof. Dr. Altan Kayran ______________________
Aerospace Engineering Dept., METU

Asst. Prof. Dr. Melin Şahin ______________________
Aerospace Engineering Dept., METU

Asst. Prof. Dr. Ferhat Akgül ______________________
Engineering Science Dept., METU

Mustafa Ekren ______________________
Structural Design Engineer, TAI

Date: February 10, 2012

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname : Hasan İNCİ
Signature :

ABSTRACT

DISCRETE FIBER ANGLE AND CONTINUOUS FIBER PATH OPTIMIZATION IN COMPOSITE STRUCTURES

İnci, Hasan

M.S., Department of Aerospace Engineering

Supervisor: Prof. Dr. Altan Kayran

February 2012, 148 pages

Fiber orientation angle stands out as one of the most effective design variables in the design optimization of composite structures. During the manufacturing of the composite structures, one can change the fiber orientation according to the specific design needs and constraints to optimize a pre-determined performance index. Fiber placement machines can place different width tows in curvilinear paths resulting in continuous change of the fiber orientation angle in a layer of the composite structure. By allowing the fibers to follow curvilinear paths in the composite structure, modification of load paths within the laminate can be obtained. Thus, more favorable stress distributions and improved laminate performance can be achieved. Such structures are called as variable stiffness composites structure. This thesis presents a fundamental study on the discrete fiber angle and continuous fiber path optimization of composite structures. In discrete fiber angle optimization, application of different analysis/optimization tools is demonstrated for optimum fiber angle optimization at the element level for both orthotropic and laminated composite structures. In the continuous fiber path optimization, which can be produced with fiber placement machines, optimized fiber paths are determined for different case studies. Continuous fiber path optimization is performed by means of an interface code that is developed.

It is hard to find the global optimum for complex optimization problems with hundreds of design variables. In order to find the global optimum solution for such complex optimization problems, a gradient based optimization algorithm is not appropriate because there will be a lot of local minima for the problem and gradient based optimization algorithms may be stuck at the local minimums. Therefore, an evolutionary algorithm is a better solver for such kind of complex optimization problems. In this thesis, genetic algorithm, an evolutionary algorithm, in MATLAB Optimization Toolbox is used for the optimizer and commercial finite element program Nastran is used for the structural solver. For the continuous fiber path optimizations these two programs are integrated with the interface code that is developed. Manufacturing constraints of a typical fiber placement machine is also included in the constraint definition of continuous fiber path optimization. By coupling of Nastran finite element solver and MATLAB genetic algorithm tool, with the manufacturing constraint for the fiber placement machines, the first buckling load of a continuous fiber composite plate is increased %22 with respect to a composite plate with zero degree orientations.

Keywords: Composite structure, fiber angle optimization, fiber path optimization, fiber placement machine, genetic algorithm, finite element method

ÖZ

KOMPOZİT YAPILARDA AYRIK FİBER AÇI VE SÜREKLİ FİBER YOLU OPTİMİZASYONU

İnci, Hasan

Yüksek Lisans, Havacılık ve Uzay Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. Altan Kayran

Aralık 2011, 148 sayfa

Kompozit yapı tasarım optimizasyonlarındaki en etkili tasarım değişkenlerden birisi fiber yönelim açısı olarak göze çarpmaktadır. Kompozit yapılar üretilirken, önceden belirlenmiş performans içeriklerini belirli tasarım gereklerine ve sınırlarına göre optimize etmek için fiber yönelimleri değiştirilebilir. Fiber yerleştirme makinaları, değişik genişlikteki kompozit kumaşları eğrisel yollara yerleştirebilmesi bir kompozit yapıdaki bir katman içerisindeki fiber yönelim açılarını sürekli değiştirebilmesine neden olur. Kompozit yapılarda fiberlerin eğrisel yollar izlemesini sağlayarak, bir plaka içindeki yük yollarında değişim sağlanabilir. Böylece, daha olumlu gerilim dağılımları ve arttırılmış katman performansı elde edilebilir. Bu tarz yapılara değişken sertlikli kompozit yapıları adı verilir. Bu tez, kompozit yapılarda ayrık fiber açısı ve sürekli fiber yolu optimizasyonu hakkında temel bir çalışma sunmaktadır. Ayrık fiber açısı optimizasyonunda, ortotropic ve katmanlı kompozit yapıların ikisi için de eleman seviyesinde farklı analiz/optimizasyon araçlarıyla optimum fiber açısı optimizasyonları uygulanmıştır. Sürekli fiber yolu, fiber yerleştirme makinalarıyla üretilebilen, optimizasyonunda, farklı durum çalışmaları

için belirlenmiştir. Sürekli fiber yolu optimizasyonu geliştirilmiş olan bir arayüz kodu ile gerçekleştirilmiştir. Yüzlerce tasarım değişkeni ile karmaşık optimizasyon problemlerinde global optimumu bulmak çok zordur. Bu tür karmaşık optimizasyon problemlerinde global optimum çözümü bulabilmek için eğim tabanlı optimizasyon algoritmaları uygun değildir çünkü, oldukça fazla yerel minimumlar olacaktır ve eğim tabanlı optimizasyon algoritmaları bu yerel minimumlarda takılı kalabilecektir. Bundan dolayı, bir evrimsel algoritma bu tür karmaşık optimizasyon problemleri için daha iyi bir çözücüdür. Bu tezde, MATLAB Optimizasyon Araç Kutusu içindeki bir evrimsel algoritma olan genetik algoritma optimizasyon yazılımı olarak kullanılmıştır ve ticari sonlu elemanlar programı Nastran da yapısal problem çözücü olarak kullanılmıştır. Sürekli fiber yolu optimizasyonu için bu iki program geliştirilmiş olan bir arayüz kodu ile bütünleştirilmiştir. Tipik bir fiber yerleştirme makinasının üretim sınır şartları da sürekli fiber yolu optimizasyonu tanımının içine eklenmiştir. Nastran sonlu elemanlar çözücüsüyle, MATLAB genetic algoritma aracının, fiber yerleştirme makinasınn üretim sınır şartları eklenerek bütünleştirilmesiyle yapılan optimizasyonun sonucunda, bir sürekli fiber kompozit plakanın birinci burulma yükünün, sıfır derece açılı bir kompozit plakaya göre %22 arttığı görülmüştür.

Anahtar Kelimeler: Kompozit yapılar, fiber açısı optimizasyonu, fiber yolu optimizasyonu, fiber yerleştirme makinası, genetik algoritma, sonlu elemenlar yöntemi

TABLE OF CONTENTS

ABSTRACT......iv

ÖZ......vi

TABLE OF CONTENTS......viii

LIST OF FIGURES......xi

LIST OF TABLES......xvi

CHAPTERS

1- Introduction......1

1.1- Variable Stiffness Concept......2

1.2- Capabilities of Tow Placement Machines......5

1.3- Optimization Concepts used in the Thesis......10

1.4- Motivation and Scope of the Thesis......18

2- Design Optimization of Variable-Stiffness 2D Orthotropic Plates......22

2.1- Design Optimization in Nastran......23

2.2- Optimization of Material Orientation of 2D Orthotropic Plate Composed of 8 Elements......28

2.2.1- 2D Orthotropic Plate Composed of 8 Elements Subject to Out of Plane Loading......29

2.2.2- 2D Orthotropic Plate Composed of 8 Elements Subject to In-Plane Shear Loading......31

2.3- Optimization of Material Orientation of 2D Orthotropic Rectangular Plate Composed of 736 Elements and in In Plane Bending Load......33

2.3.1- Static Analysis of the Rectangular Plate with Random Material Orientation Angles......35

2.3.2- Optimization using initial material orientation angle of 5^o......36

2.3.3- Optimization using initial material orientation angle of -5^o......38

2.3.4- Optimization using initial material orientation angle of 15^o 40
2.3.5- Optimization using initial material orientation angle of 25^o 42
2.3.6- Comparison of the Case Studies Employing Different Initial Material Orientation Angles 43
2.4- Optimization of Material Orientation of 2D Orthotropic Rectangular Plate with a Circular Cut-Out in Center 45
2.4.1- 2D Orthotropic Square Plate Composed with a Circular Cut-Out in Center under In-Plane Loading 49
2.4.2- 2D Orthotropic Square Plate Composed with a Circular Cut-Out in Center under In-Plane Shear Loading 53
3- Determination of Optimum Discrete Fiber Angles of Laminated Composite Structures 58
3.1- Optimization by the Nastran Design, Sensitivity and Optimization Module 59
3.1.1- Optimization of Fiber Orientation of a Single Layer Composite Rectangular Plate Composed of 736 Elements Subject to In-Plane Bending Load...60
3.1.2- Optimization of Fiber Orientation of a Four Layer Mid-Plane Symmetric Composite Rectangular Plate Composed of 736 Elements Subject to In-Plane Bending Load 62
3.1.3- Optimization of Fiber Orientation of Four Layer Mid-Plane Symmetric Composite Rectangular Plate Composed of 736 Elements Subject to In-Plane Bending Load with Layer Thicknesses and Fiber Orientations as Design Variables, and Volume as the Objective Function.. 66
3.2- Optimization by Coupling the Nastran Finite Element Solver and Genetic Algorithm Optimizer in MATLAB 70
3.2.1- Optimization of a Rectangular Single Layer Composite Plate under Transverse In-Plane Loading 73
3.2.2- Optimization of a Rectangular Single Layer Composite Plate under Transverse In-Plane Loading with a Coarser Mesh 75
4- Determination of Optimum Continuous Fiber Paths of Variable Stiffness Composite Structures 80
4.1- Description of Continuous Fiber Path Optimization 81
4.2- Solution Procedure Used for Continuous Fiber Path Optimization................ 89

4.3- Optimization of a Composite Plate by Using Genetic Algorithm with the Total Strain Energy as the Objective Function 93

4.3.1- Optimization of a Rectangular Single Layer Composite Plate under Transverse In-Plane Loading 94

4.3.2- Optimization of a Rectangular 4 Layer Composite Plate under Transverse In-Plane Loading 96

4.4- Optimization of a Composite Plate by Using Genetic Algorithm as First Buckling Load as the Objective Function 99

4.4.1- Optimization of Buckling Load of a Rectangular Single Layer Composite Plate under In-Plane Compressive Loading 103

4.4.2- Optimization of Buckling Load of a Rectangular Multilayer Composite Plate under In-Plane Compressive Loading 106

5- Conclusion 116

References 123

APPENDICES 126

Appendix A 126

Appendix B 129

Appendix C 135

LIST OF FIGURES:

FIGURES

Figure 1.1: Variables that defines the reference fiber path. Taken from Ref. [4] 4
Figure 1.2: (a) Composite panel manufactured with shifted fiber path method. 5
(b) Composite panel manufactured with parallel fiber path method taken from. 5
Ref. [2] 5
Figure 1.3: Hercules, Inc. tow placement machine, taken from Ref. [4] 6
Figure 1.4: An actual tow placement machine used in the industry, taken from Ref. [6] 7
Figure 1.5: Details of the delivery head, taken from Ref. [2] 8
Figure 1.6: An actual fiber placement machine's delivery head, taken from Ref. [6]. 8
Figure 1.7: The differential pay out of the fibers passing from the delivery head, taken from Ref. [4] 9
Figure 1.8: The ability of cutting and restarting of individual tow paths, taken from Ref. [4] 9
Figure 1.9: Manufacturing of fuselage panel for Airbus A350 aircraft, taken from Ref. [6] 10
Figure 1.10: The simple genetic algorithm structure taken from Reference [19]..... 14
Figure 1.11: Child creation of the genetic algorithm taken from Reference [5]........ 16
Figure 2.1: Early structural optimization attempts, taken from Ref. [23] 24
Figure 2.2: Coupling analysis and optimization using approximations, taken from Ref [23] 26
Figure 2.3: Nastran Implementation of structural optimization [24] 27
Figure 2.4: Material directions for the case studies 28
Figure 2.5: Loads and boundary conditions for the orthotropic plate under out of plane load 30

Figure 2.6: Optimized material orientation angles for the orthotropic plate under out of plane load. 30
Figure 2.7: Design cycle vs. Objective function graph for the out of plane load case 31
Figure 2.8: Load and boundary conditions for the in-plane shear loaded structure 32
Figure 2.9: Resultant orientations for the case study for in-plane shear load 32
Figure 2.10: Design cycle vs. objective function graph for the in-plane shear load case 33
Figure 2.11: The load and boundary condition for the following case studies performed 34
Figure 2.12: Optimal orientations for the coarse mesh (46x16) taken from Ref. [3]. 35
Figure 2.13: Random material orientations assigned to the elements of the cantilevered plate model 36
Figure 2.14: Optimized material orientation angles for the 5 degree initial material orientation 37
Figure 2.15: Total strain energy versus objective function graph for the 5 degree initial material orientation 37
Figure 2.16: Optimized material orientation angles for the -5 degree initial material orientation 38
Figure 2.17: Total strain energy versus design cycle graph for the -5 degree initial material orientation 39
Figure 2.18: Optimized material orientation angles for the 15 degree initial material orientation 40
Figure 2.19: Design cycle vs. objective function graph for the 15 degree initial material orientation 41
Figure 2.20: Optimized material orientation angles for the 25 degree initial material orientation 42
Figure 2.21: Total strain energy versus design cycle graph for the 25 degree initial material orientation 43
Figure 2.22: The mesh and the geometry for the circular cut-out square plate 45
Figure 2.23: The element and material coordinate systems defined in Nastran with CQUAD4 elements from Reference [23]. 46

Figure 2.24: Zero degree material orientations for the current case study................. 47
Figure 2.25: Optimum material orientations for the minimum compliance determined in Reference 3 (a) for in-plane normal load, (b) for in-plane shear load 48
Figure 2.26: Displacement constraints of the geometry for the case study. 50
Figure 2.27: In-plane normal load case... 51
Figure 2.28: Optimized material orientations for the in-plane normal load case....... 52
Figure 2.29: Total strain energy versus design cycle for the in-plane normal load case 52
Figure 2.30: Displacement constraints for the pure shear load case study 54
Figure 2.31: In plane shear load acting on the plate with central cut-out 55
Figure 2.32: Optimized material orientations for the in-plane shear load case 55
Figure 2.33: Total strain energy versus design cycle for the in-plane shear load case 57
Figure 3.1: Finite element model and the loads and boundary conditions for the cantilevered plate 61
Figure 3.2: Fiber orientation distribution of the single layer composite laminate after optimization 61
Figure 3.3: Objective function vs. design cycle graph for the initial fiber orientation angle of 15 degree 62
Figure 3.4: The stacking sequence of the composite plate used in the current case study. 63
Figure 3.5: Optimized fiber orientations of the outer layer of the 4 layer mid-plane symmetric composite laminate... 63
Figure 3.6: Optimized fiber orientations of the inner layer of the 4 layer mid-plane symmetric composite laminate... 64
Figure 3.7: Objective function vs. design cycle graph for the mid-plane symmetric composite laminate with the initial fiber orientation of $+15^{o}/-15^{o}$............................ 65
Figure 3.8: Optimized total thickness distribution of the cantilevered plate 67
Figure 3.9: Optimized upper layer orientations of the plate 68
Figure 3.10: Optimized lower layer orientations of the plate 68
Figure 3.11: Total volume versus design cycle.. 69

Figure 3.12: Simplified steps showing the coupling of the Nastran solver and the MATLAB genetic algorithm toolbox optimizer in the optimization process 72
Figure 3.13: Optimization process of the discrete fiber angle orientations of the single layer composite laminate 74
Figure 3.14: Mesh of the current case study with 12 finite elements 76
Figure 3.15: Optimum orientations for the 30 degree initial guess case study 77
Figure 3.16: Optimization process for the 12 element composite plate..................... 78
The optimum orientations for the elements is shown in Figure 3.17. 78
Figure 4.1: The variables in the variable stiffness composite laminate 82
Figure 4.2: The effect of the rotation angle $\emptyset$, in case (1) $\emptyset = 0°$, in case (2) $\emptyset = 45°$, in case (3) $\emptyset = 90°$ for the $0°45°$ reference fiber path, taken from Reference [4]. 83
Figure 4.3: The visualization and the fiber angles of the rotated reference fiber path for 45045 orientation, taken from Reference [28]. 84
Figure 4.4: Shifted fiber path according to the reference path, taken from Ref. [28] 86
Figure 4.5: Shifted paths for a laminate with the reference fiber path definition $0°0°45°$, taken from Ref. [4] 86
Figure 4.6 : Parallel paths for a laminate with the reference fiber path definition $0°0°45°$, taken from Ref. [4] 87
Figure 4.7: Laminate created with (a) shifted fiber path method, (b) parallel fiber path method with the reference fiber path definition $0°0°75°$, taken from Ref. [4]. 88
Figure 4.9: Loads and the displacement boundary conditions for the cantilevered plate 93
Figure 4.10: Optimized orientation for the single layer composite plate................... 94
Figure 4.11: Optimized continuous fiber paths for the case study............................. 95
Figure 4.12: Optimization process for the single layer cantilevered plate................. 95
Figure 4.13: Optimized upper layer orientations of the 4 layer composite plate 96
Figure 4.14: Optimized lower layer orientations of the 4 layer composite plate 97
Figure 4.15: Optimized fiber paths through the 4 layer composite plate................... 97
Figure 4.16: Optimization process for the 4 layer cantilevered plate 98
Figure 4.17: Loads and the displacement boundary conditions for the case study on maximization of buckling load.. 101

Figure 4.18: Optimized reference fiber path for the single layer composite plate... 104
Figure 4.19: Optimized fiber paths of the composite plate 104
Figure 4.20: Optimization process for the single layer composite plate 106
Figure 4.21: Optimized reference fiber path for the lamina with the positive fiber orientation angle 107
Figure 4.22: Optimized reference fiber path for the lamina with the negative fiber orientation angle 108
Figure 4.23: Optimized continuous fiber orientations on the plate (a) fiber paths with positive fiber orientation angles and (b) fiber paths with negative fiber orientation angles 109
Figure 4.24: Optimized continuous fiber paths through the composite plate 109
Figure 4.25: Optimization process for the multilayer composite plate 110
Figure 4.26: Buckling and stiffness perfomance of case 1 in Reference 30, taken from Reference 30 112
Figure 4.27: Optimization process of unconstrained buckling maximization case study 114

LIST OF TABLES:

TABLES:

Table 2.1: Table for the static solution for the random orientations and the optimum solutions for different initial values..43

Table 3.1: Optimum total strain energy values for the composite laminates........65

Table 3.2: Optimum total strain energy values for different initial guesses in Nastran Optimization module..76

Table 4.1: Optimized total strain energy values for the discrete fiber orientation angles and continuous fiber path definitions...99

Table 4.2: The maximum eigenvalues and the increase percentages of the case studies performed..115

CHAPTER 1

1- Introduction

Day by day, composite structures are gaining more importance in aerospace structures, as well in structures used in other industries. Beside its high strength to weight ratio, with the ability to tailor the fibers along the desired orientation, composites can be manufactured with respect to the needs of the end users. Composite structures can be modified to improve its performance, measured through increased load carrying capability or lower weight for a specified design load. Traditionally, design of composite laminates is limited to thickness optimization of stacking sequences of thin laminates. In this traditional method the fibers are straight. However, it is been realized that in-plane variations of fiber directions gives a better efficiency for the placement of the fibers. [1] There are tow placement machines, capable of tailoring the composite structure with in-plane variations. In-plane variations of continuous fiber paths are given as input to tow placement machines. This manufacturing concept creates the so called variable stiffness composite plate concept. [2; 3; 4] On the other hand, curvilinear fiber paths bring on several important constraints to the design process. The turning radius of the delivery head of the tow placement machine is the major limiting factor for the design process. [1] In this thesis, these concepts are all being discussed and applied to the some certain problems.

1.1- Variable Stiffness Concept

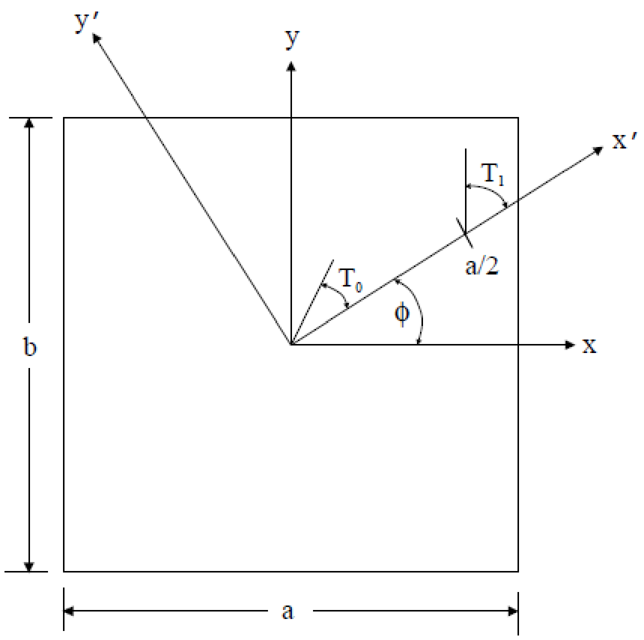
Composite structures are made of layers of tailored fibers and a matrix which holds fibers together. Thin layers of fiber and matrix mixtures make up the plies. The laminate is made of by stacking thin plies until a certain thickness is reached. During the stacking of the plies, each ply can be manufactured with different fiber orientations which usually vary within the plies in the so called variable stiffness laminates. Composites have the greatest strength and stiffness along its fiber directions. The orientations can be designed with respect to the given loading condition to have higher stiffness and strength properties in the higher stress areas. [5] In many classical design applications, laminates are composed of plies with straight fiber paths. Therefore, in classical applications, the desired laminate stiffness is achieved by stacking plies with different constant fiber orientations. However, in the variable stiffness concept, fiber path is changed continuously in a ply by special machines. In this respect, the variable stiffness concept makes the composite structures more efficient than the classical approach, because fiber orientation angle in a ply can be changed continuously to tailor the stiffness of the composite structure for a desired structural response. Composite structure can be made lighter and stiffer. However, variable stiffness concept comes with a major drawback which is the need for the automated manufacturing which increases the cost of the manufacturing system. Although the automated manufacturing system that is required to manufacture variable stiffness composite structure is expensive, it can be argued that for serial production the initial investment for the automated manufacturing system can be compensated.

Variable stiffness implies structures that have different mechanical properties at different points in the structure. This would give us stiffer parts on more stressed areas. This concept can be used to optimize a particular structural response or combination of different structural responses in a multi-objective framework. For instance, maximization of the buckling load or natural frequency, and minimization of the displacement or weight of a composite structure can be achieved by employing the variable stiffness concept in the design. Therefore, multi-objective structural optimization problems are out of the scope of this thesis.

While designing the variable stiffness composite structure, one can also use discrete fiber distribution. In a finite element framework discrete fiber orientation implies the determination of optimum fiber orientation angles at the element level. However, in case of discrete fiber orientation, the manufacturing of such composite structure would be almost impossible. One would need to make approximations to smooth out the discrete changes in the fiber orientation so that the part would be manufacturable. Determination of optimum discrete fiber angles at the element level would be the ideal solution since local changes in the fiber orientation angle are taken into account. Thus, discrete fiber angle optimization would give the designer an insight about the ideal optimum structure. However, to make the part manufacturable, continuous fiber path optimization is required so that fiber orientation angle can be adjusted along the predefined path by means of a computer controlled machine such as fiber placement machine.

With the fiber placement machines fiber tows can be placed along curvilinear fiber paths. [2; 4] This concept offers continuous fiber paths on a composite structure. The designed continuous fiber paths are easier to model and to manufacture.

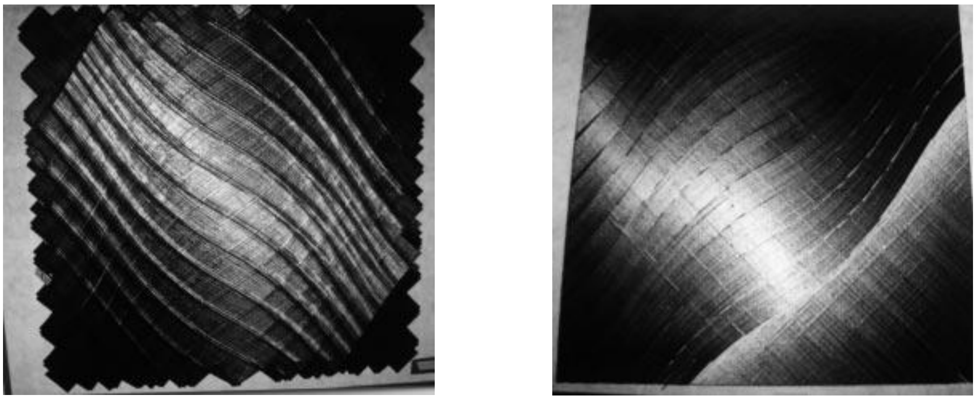
Continuous fiber paths are produced with two basic methods known as shifted and parallel paths. In both methods, one has a reference fiber path, and surface of the laminate is steered according to this reference fiber path [4]. This reference fiber path is commonly determined with three variables, $T_0, T_1, \emptyset$. T_0 is the fiber orientation angle at a side edge angle of the composite laminate, T_1 is the midpoint fiber orientation angle and $\emptyset$ is the angle about which the reference fiber path is rotated with respect to the original coordinate system of the composite laminate. [4] These variables are also defined in Figure 1.1.



T_0 → *angle measured at the centre of the plate*
T_1 → *angle measured at* $x' = a/2$
$\emptyset$ → *rotation angle*

Figure 1.1: Variables that defines the reference fiber path. Taken from Ref. [4]

In shifted path method, fibers are shifted by a fixed amount in a predefined direction. In this method first a reference fiber path is drawn and the remaining paths are obtained by shifting the reference path. In the shifted fiber method overlaps or gaps may occur since fiber paths may not always be parallel to each other. Thus, parts made by shifted path method are usually thicker due to the overlapping regions. On the other hand, in the parallel path method, each new path is parallel to the centerline of the adjacent path. Therefore, in the parallel path method there is no overlapping, but it is harder to manufacture due to the need to satisfy the curvature constraint for each path definition. Figure 1.2 shows examples of manufactured composite panels with shifted and parallel path method.



(a) (b)

Figure 1.2: (a) Composite panel manufactured with shifted fiber path method.
(b) Composite panel manufactured with parallel fiber path method taken from.
Ref. [2]

Shifted path method and the detailed description of the reference path are discussed in Chapter 4. Only shifted path method is defined since it is used in this thesis.

1.2- Capabilities of Tow Placement Machines

CNC-controlled fiber placement machines first become commercially available in late 1980s. In the mid-1990s, tow placement machines are introduced and they are started to be used. These machines offer, high volume production rates with the computer controlled laying system. Since tow placement machines are computer controlled, they offer a much more reliable, consistent and cost effective fabrication. Although they are best suited for plain and relatively flat surfaces, tow placement machines can also be used to manufacture much more complex geometries because of the robotic features of the machines which provide high manufacturing capabilities. It is claimed in Reference 6 that, complete composite fuselages are made in a fraction of time required for equivalent metal structures. Tow placement machines, also minimizes the wastage and removal operations for the composite structure.

With the tow placement machine, one can manufacture variable stiffness composite structure. Tow placement machine's ability of placing the fibers along a continuous curvilinear path makes it possible to manufacture the variable stiffness composite structures. By means of fiber placement machines tows can be laid in any pre-programmed orientations and positions so that laminate can be tailored to deliver the strength and stiffness parameters required by designers at various parts of the structure, fibers being aligned with the local forces expected in service.

Tow placement machines are computer controlled, multi-axis fiber placement systems. These machines consist three main parts which are mandrel, moving platform and delivery head. Figure 1.3 shows main parts of a typical tow placement machine. Figure 1.4 shows an actual tow placement machine used in the industry.

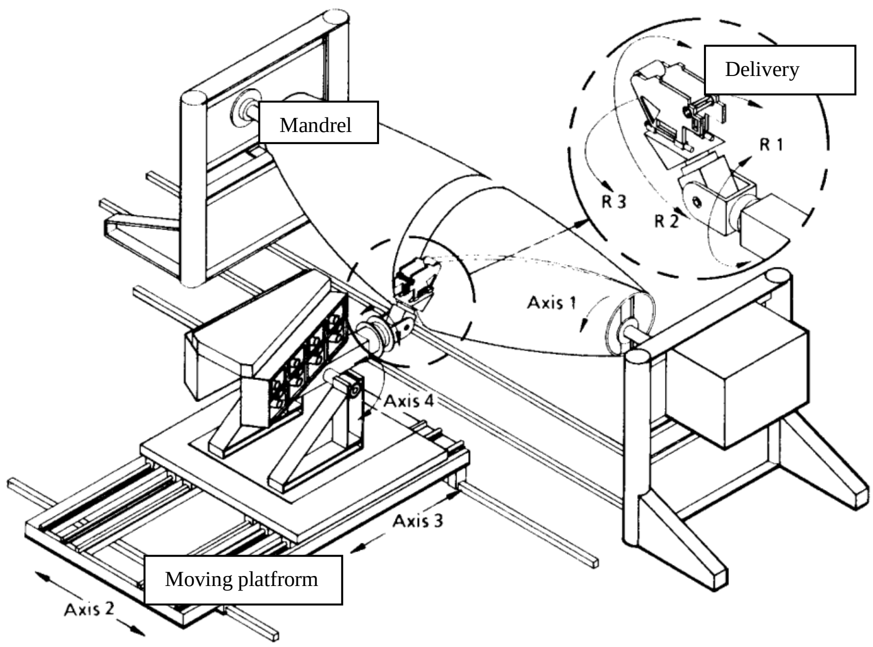


Figure 1.3: Hercules, Inc. tow placement machine, taken from Ref. [4]

Head of an automated fiber placement machine can be mounted on a multi-axis articulating arm that moves around the tool/mandrel, or can be carried by a gantry.

Alternatively, the tool can be rotated under a static head or both the head and the mandrel can move in an organized way controlled by a software program.

Figure 1.4: An actual tow placement machine used in the industry, taken from Ref. [6]

Figure 1.5 shows the detailed schematic view of the delivery head of a typical fiber placement machine. This delivery head can make curvilinear paths, cut and continue individual fibers according to the needs of the user. With the fiber placement machines composite material is laid free of tension and folds, with precisely defined pressure. Heads can also provide consolidation with compaction rollers besides carrying out all necessary cutting and re-start operations. Figure 1.6 shows an actual delivery head which is used in the industry.

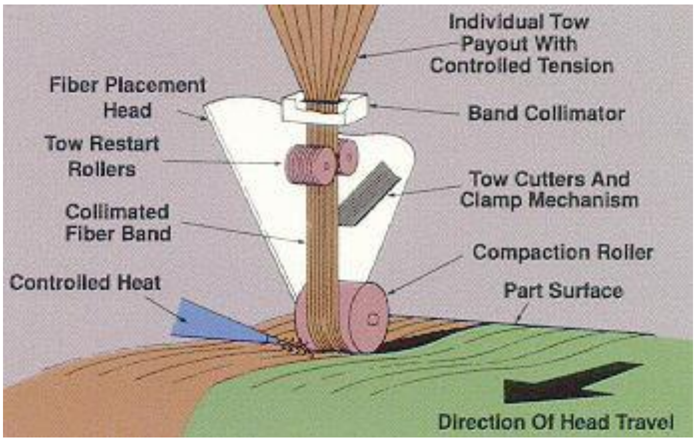


Figure 1.5: Details of the delivery head, taken from Ref. [2]

Figure 1.6: An actual fiber placement machine's delivery head, taken from Ref. [6]

In case of placing tows along a curved contour the speed at which each tow passes through the head must be individually controlled. Figure 1.7 shows a curved path created by the fiber tows. In this application, since path A is longer than path B fiber tow is laid with different speeds. This phenomenon is called a differential tow pay out. [4] Fiber placement machines must therefore control the rates at which the tows are placed on the structure. The speed at which each tow passes through the head is individually controlled.

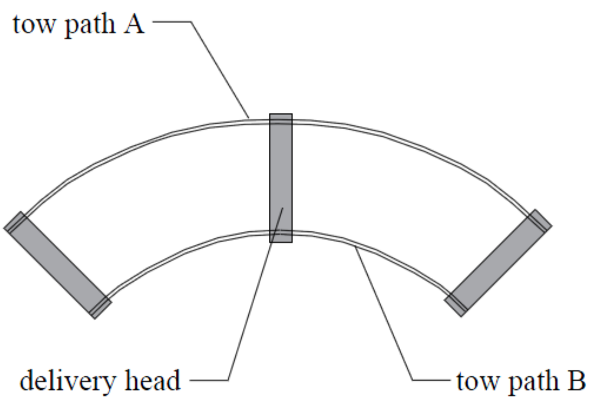


Figure 1.7: The differential pay out of the fibers passing from the delivery head, taken from Ref. [4]

Figure 1.8 shows the schematic view which depicts the cutting and restarting ability of the tow placement machine. In Fig. 1.8 the grey region represents the omitted region where the tow was cut and restarted by the fiber placement machine. It should be noted that with the cutting and restarting ability of the fiber placement machines, composite structures with cut-outs can be easily manufactured. In addition, the boundaries of the composite structure can also be handled easily.

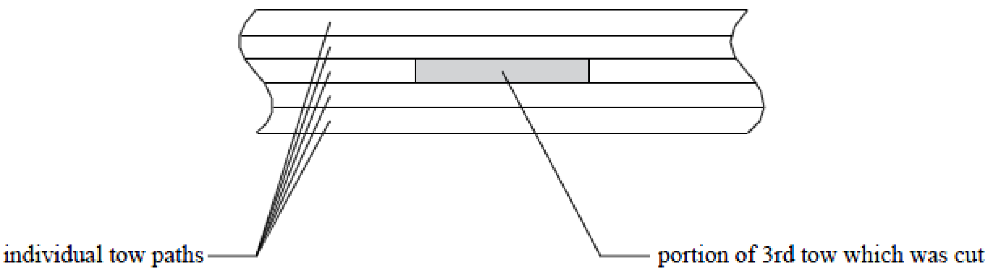


Figure 1.8: The ability of cutting and restarting of individual tow paths, taken from Ref. [4]

Since composites can be produced with a wide range of material properties, the tow placement machines must deal with wide range of materials as well. For most of the composite parts in Boeing 787 and Airbus A350 aircrafts, resins with low viscosities are used, because low viscosity resins offer the best structural properties after curing. [6] Therefore, tow placement machines must be capable of heating and pressurizing the point of compaction. Figure 1.9 shows the production of Airbus A350 aircraft fuselage panel which is, manufactured by a tow placement machine [6].

Figure 1.9: Manufacturing of fuselage panel for Airbus A350 aircraft, taken from Ref. [6]

Today, tow placement machines are widely used in the aerospace structures, wind turbines, automotive and performance yachting. Military aerospace applications also make use of fiber placement technology. Lockheed Martin F-22 Raptor's inlet duct and other certain parts are manufactured by tow placement machines. As another example for the application of the tow placement machines in military aircraft, F-35 JSF's 35-feet-long upper composite wing surface can be given.

1.3- Optimization Concepts used in the Thesis

In daily life, optimization has a very significant effect on almost any kind of problem. One faces with problems do not have to be a numerical problem. This can be the road to take, or the daily routines, one does every day. Everyone uses the resources that are available and try to have the fastest, economical and most efficient use of the resources. In today's world, everything must be optimum. Because there are limited resources and people try to have the best performance from the limited resources. Every optimization problem has some constraints. Constraints limit the usage of the resources that one has for the particular problem at hand. Constraints can be of three types, inequality, equality and side constraints. Inequality constraints set the group for the problem when the resources properties are bigger or lower from some value. Equality constraints set the group for the problem when the resources properties are equal to some value. Side constraints set the group when the resources are between two defined values. The real time optimization problems might be described with simple mathematical formulas for most of the cases. The resources are counted as variables and the product is the objective function. The constraints imposed prevent one to select the design variables freely. The design variables are selected such that the constraints imposed are satisfied. Thus, resources are used in the most effective way.

A simple optimization problem can be mathematically formulated as follows;

$$F(X) \quad objective\ function\ to\ be\ optimized$$

Subject to:

$$g_j(X) \leq 0; \quad j = 1, \dots, n_g \quad inequality\ constraints$$

$$h_k(X) = 0; \quad k = 1, \dots, n_h \quad equality\ constraints$$

$$x_i^L \leq x_i \leq x_i^U; \quad i = 1, \dots, n \quad side\ constraints$$

$$X = \{x_1, x_2, \dots, x_n\} \quad design\ variables$$

where inequality and equality constraints are usually expressed in terms of design variables as linear or nonlinear functions.

In daily life, the problems are usually solved with the experiences gained about the problem. But this approach is time and money consuming in real life. Because there are plenty of solutions for the problem and one must try all the possibilities and make a final decision for the optimum solution. To make this process easier and less time consuming, different algorithms are created to solve the problems. According to the problems, one can choose an appropriate algorithm for each case. The algorithms works fine with certain cases, which may involve linear or nonlinear objective functions, linear or nonlinear constraints, or the combinations of these cases.

People try to optimize everything, but engineers will lead the way to optimization. When the composites are considered, the design variables are usually defined as thickness and orientation of fibers. In the early studies, fibers are considered to be straight, and in a ply orientation angle is fixed. Straight fiber paths are preferred because of the less time and labor requirement to manufacture the part. With the improvements in automated manufacturing technology, today fibers can be layed along curvilinear paths on the composite structure and even faster compared to conventional tape laying. The variable stiffness concept became practically applicable with the improvements achieved in robotic technology. Nonlinear variable stiffness composite structure problem is a complicated optimization problem. Therefore, to solve the variable stiffness composite structure problem, which has lots of local minimum, one needs an optimization algorithm. This optimization algorithm must be able to solve nonlinear complex problems.

Two basic methods are used in this thesis for the solution of the optimization problem. These are gradient-based optimization of in non-linear constrained optimization problem and evolutionary genetic algorithm. These methods are used in different case studies depending on their stability for the particular problem at hand.

In the literature, there are many gradient based fiber orientation optimization algorithms [7; 8]. These algorithms solve the problems analytically. Several objective functions, such as minimum strain energy, minimum structural compliance, minimum deflection and minimum weight can be defined in a typical optimization problem. Analytical solution is the exact solution for the cases, but it restricts the engineer. Complex geometries and load cases are hard to solve analytically. Therefore, optimization programs with embedded finite element solver, such as Nastran, offer a good tool for optimizing complex optimization problems [9].

In this thesis, as the gradient-based numerical optimization solver, the optimization module of Nastran is used. Nastran applies concepts that make the structural response quantities nearly linear function of the design variables.

In Nastran, linearization of the nonlinear constrained problem is achieved by building an approximate model. Nastran uses formal approximations to the finite element analysis and the sensitivity analysis to avoid the high cost associated with repeated finite element analysis during design optimization. The use of approximate model and the convergence concepts used in Nastran is explained in detail in Chapter 2.

In the thesis, Nastran optimization module is used in discrete fiber orientation angle optimization for both two dimensional orthotropic material and laminated composite structure. However, in general whether discrete fiber angle or continuous path optimization is considered, such problems are highly nonlinear in nature such that one may expect to have many local minimums. Therefore, gradient based optimization method may not be the most appropriate method for the optimization of variable stiffness composites. It should be noted that solution for the field variables for the variable stiffness composite structures necessitates the use of a finite element solver, because variable stiffness composites are hard to be modeled analytically. Optimization of discrete fiber angles or continuous path optimization would thus require the use of a finite element solver coupled with and optimizer. In the thesis, for the function and constraint evaluations Nastran finite element program is used. As for the optimizer, to eliminate the inherent drawback of gradient based optimization method, in this thesis evolutionary genetic algorithm is used. There are also other evolutionary algorithms such as, simulated annealing, differential evolution, memetic algorithms [10]. Some of these methods are used in composite structure optimization [11]. However, genetic algorithm is widely used in optimization of composite structures [12; 13; 14; 15; 16]. There are also studies which compares the performance of the optimization algorithms [17]. Nastran optimization module and the MATLAB genetic algorithm module are compared with each other by Benford and Tinker [18]. Based on the information provided in the literature, genetic algorithm is decided to be used in this thesis. Thus, besides the optimization module of Nastran, Nastran finite element solver is coupled with the genetic algorithm toolbox of MATLAB to solve the discrete fiber orientation angle and continuous fiber path optimization problems of composite structures.

It should be noted that genetic algorithm emulates nature. Figure 1.10 shows the basic structure of the genetic algorithm. As in the science of genetics, the population, sub-population, organism, chromosomes and the genes are present in the basic structure. The genes are the optimization variables in the genetic algorithm. Differing from the real life, the genes are represented as binary codes. [19] These binary codes are processed to have best solution in the design domain. In Figure 1.10, the cross-over and mutation operators, which process on the binary codes are shown.

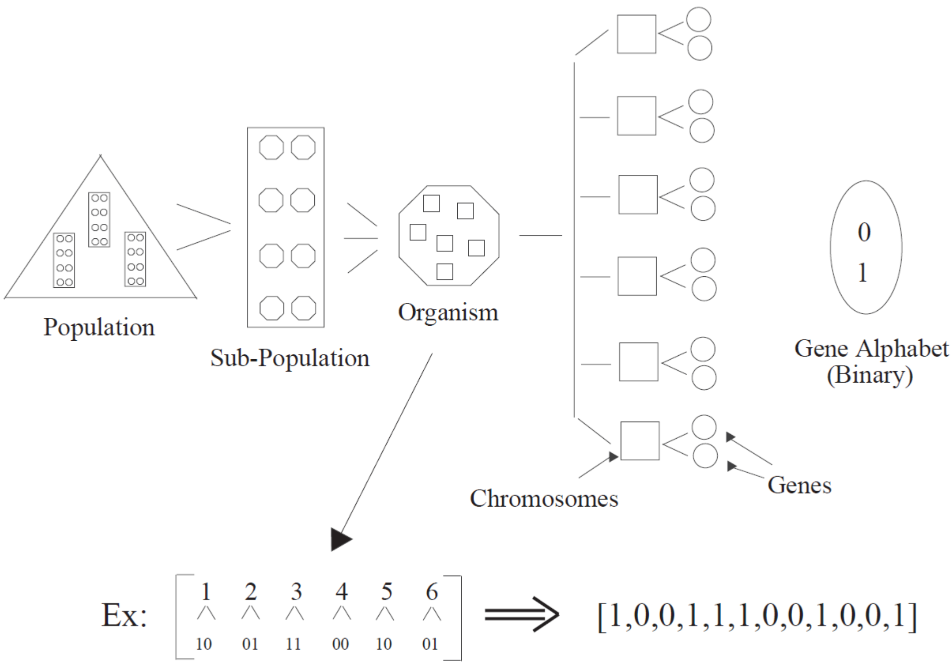


Figure 1.10: The simple genetic algorithm structure taken from Reference [19].

In the genetic algorithm, at the beginning, random initial population is created in the whole design domain. The objective function is evaluated for every design alternative within the population. The evaluated objective function values are compared with each other. The best individuals are selected among the others. In a way, best individuals survive in the optimization process. The genes that generate the best values for the objective functions are processed to have new pairs of children. Cross-over operator affects the performance of the genetic algorithm. Cross over is simply combining a pair of parents [20]. After that, the mutation operation is employed. Mutation operator makes random changes in the genes. Mutation avoids losing whole information coming from the heritage, and it is used to increase the randomness in the solution space. This is the simple design cycle of the genetic algorithm. The cycle is repeated until the stopping criterion is reached and a global optimum solution is found. The child creation process is shown in detail in the Figure 1.11. Since there is no differentiation in this process, genetic algorithm is not stuck in a local optimum and finds the global minimum of the problem directly.

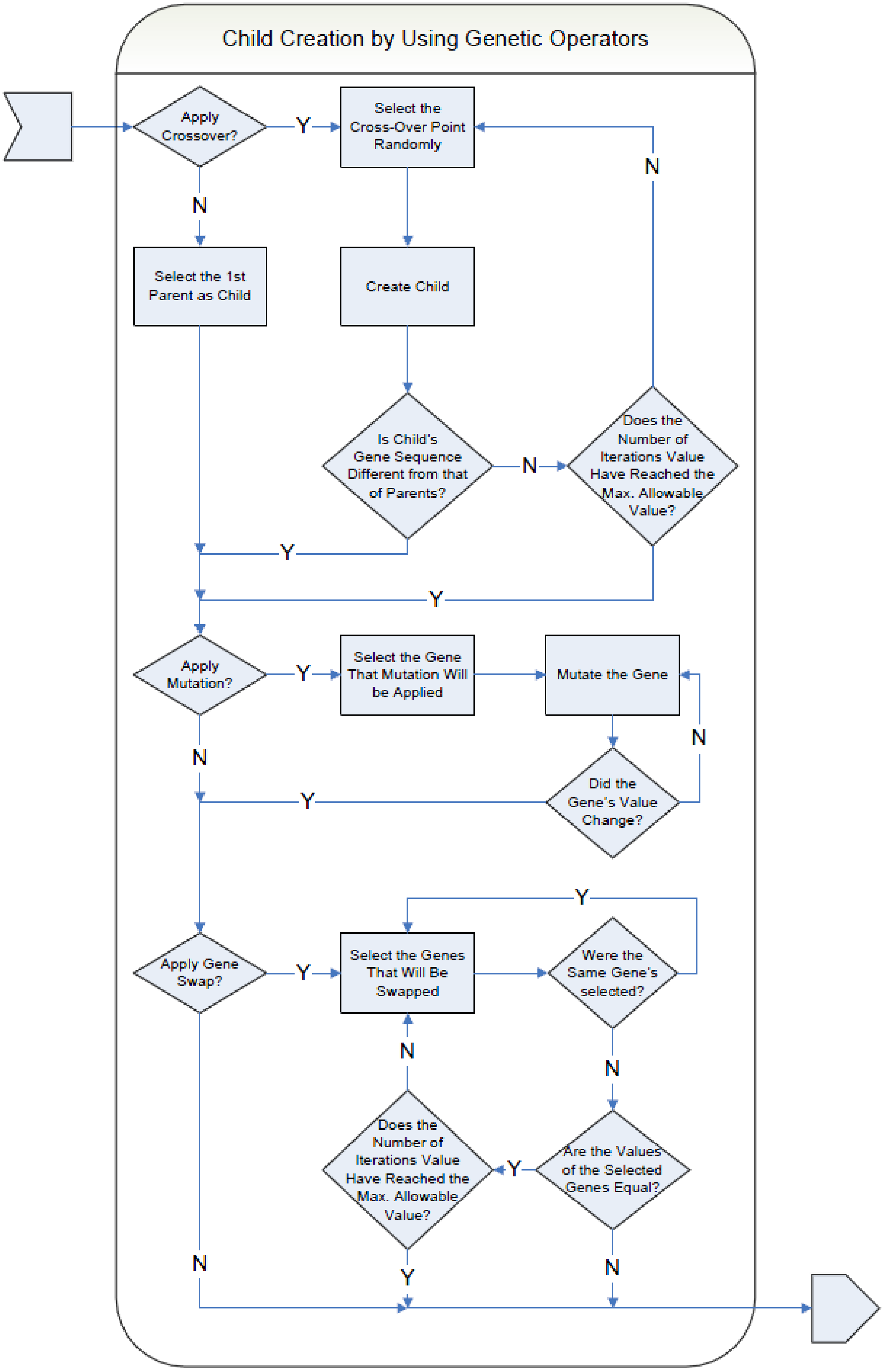


Figure 1.11: Child creation of the genetic algorithm taken from Reference [5].

The most important parts for the child creation are the mutation and the cross-over phases of the genetic algorithm. The motivation of using mutation is that, after several generations it is possible to lose some converging genes in the individuals. If this happens, without the genetic algorithm converging to a satisfactory solution, the algorithm has prematurely converged [21]. This case may be a problem in small populations. Without a mutation operator, there is no chance to introduce the missing gene to the individuals. The mutation option in MATLAB genetic algorithm is controlled with two options which are “shrink” and “scale” [20]. “Shrink” controls the rate at which the average amount of mutation decreases. “Shrink” has the default value of 1, then the amount of mutation decreases to 0 at the final step. “Scale” controls the standard deviation of the mutation at the first generation. “Scale” is multiplied by the range of the initial population. Differing from these two options, mutation has different handles in MATLAB genetic algorithm. There are uniform, adaptive feasible and Gaussian mutation handles in MATLAB genetic algorithm [22]. Gaussian handle is the default handle for MATLAB genetic algorithm. For the continuous fiber orientation optimization case studies the mutation handle is changed to adaptive feasible which can be seen in Appendix C. This modification is done to improve the efficiency of the algorithm for the particular case study.

The crossover operator in genetic algorithm changes genes between the best individuals and the worst individuals. Best individual randomly give their genes to the worst individuals to make the worst individuals to survive in the next generation of the population. There are several crossover handlers in MATLAB genetic algorithm. These handlers are heuristic, scattered, intermediate, single point, two point and arithmetic handlers. Scattered crossover handler is the default handler for MATLAB genetic algorithm. The crossover handler is kept as default for the continuous fiber orientation case studies.

1.4- Motivation and Scope of the Thesis

In this thesis, discrete fiber orientation angle and continuous fiber optimization of composite structures is performed by employing two different optimization methods. First method employed is the gradient based optimization performed by the optimization module of Nastran. The second method employed is the coupling of MATLAB genetic algorithm code and Nastran finite element solver. For case studies, different two dimensional problems are defined and discrete fiber orientation angle and continuous fiber path optimizations are carried out. The following paragraphs give brief descriptions of the different problems that are solved in the thesis.

Chapter 2 is devoted to discrete orientation angle optimization of plates made of 2D orthotropic material. It is considered that before dealing with laminated composite structures, it would be more appropriate to study material orientation angle optimization of composite structures made of 2D orthotropic materials. First a simple 2D orthotropic material with 8 elements is considered and solved by the optimization module of Nastran. Design variables are taken as material orientation angles which are let to vary in the range of -90^o to 90^o which are the side constraints the objective function is taken as the total strain energy of the structure, and it is aimed at minimizing the total strain energy of the structure which also implies maximizing the stiffness of the structure. Eight discrete orientation angles are defined as the design variables and discrete material orientation angle optimization is performed for different simple load cases for verification purposes.

The second problem tackled is a 736 element rectangular structure under in-plane bending. This problem is again solved by the optimization module of Nastran. In this problem 736 discrete material orientation angles are defined as design variables and discrete orientation angle optimization is performed. This example structure and the loading condition are taken from Reference 3 and changed to 2D orthotropic case. Again only side constraints are defined for the variables such that design variables restricted to be in the -90^o to 90^o degree ranges and the objective function is again as the total strain energy. It is required that for the load case defined, the total strain energy is minimized by the proper selection of the material orientation angles. Optimization is performed by taking different initial values for the orientation angles. Different initial values are tried to examine the effect of initial orientation angles on the final optimized configuration. It should be noted that the outcome of gradient based optimization problems highly depend on the initial conditions.

As a last case study for the initial part, a square 2D orthotropic laminate with a circular cut out is considered, and again discrete orientation angle optimization is carried out. For this problem different loading and initial values of orientation angles are also considered to see how the optimized solutions change with respect to loading and initial values of design variables. Chapter 2 describes these case studies in detail.

In Chapter 3, laminated composite structures are considered. First problem in Reference 3 is solved again as a laminated composite structure, instead of single layer two dimensional orthotropic materials. Discrete fiber orientation angle optimization is performed by the optimization module of Nastran. For this sample problem, fiber orientation angles are considered as variables. For this problem side constraint for the fiber orientation angles is taken as -90^o to 90^o degrees and the objective function is the total strain energy. Case studies performed in this section are discussed in detail.

In the second part of Chapter 3, genetic algorithm toolbox of MATLAB is used as the optimizer and Nastran is used as the finite element solver to perform the case studies related to the discrete fiber orientation angle optimization. The Nastran sample cases are performed to see if MATLAB would give reasonable results. The procedures followed are explained in Chapter 3 in detail.

In Chapter 4, optimization of continuous fiber path is performed. For the case studies in this section, again the MATLAB genetic algorithm is used as the optimizer and Nastran is used as the finite element solver. Firstly, the continuous fiber path concept and the manufacturing methods of the variable stiffness composite plates are introduced. The equations for determining the reference fiber path are given, and the variables, which are used in the definition of the reference fiber path, are introduced. After the introduction of the concepts used in the continuous fiber path optimization, the optimization procedure of MATLAB genetic algorithm is again explained, since for the continuous fiber path optimization some changes are made in the main MATLAB code. In this section, continuous fiber orientation formulations are added to the main code. Manufacturing constraint is imposed to the problem in terms of curvature constraints, since at the end it is desired to design a composite structure which can be manufactured with the tow placement machines. The tow placement machines have inherent manufacturing constraints which must be considered during the design optimization phase of the composite structure. Curvature constraint is one of the most critical constraints which must be considered in the optimization process. The heads of the tow placement machines usually have a minimum turning radius which imposes a constraint on the fiber path. In addition, tows may tend to wrinkle if they are curved too much. Therefore, it is necessary to place a limit on fiber curvature. In this present study, curvature constraint is also added to the continuous fiber path optimization process.

The case studies in Chapter 4 can be divided into main groups. In the first group objective function is taken as the total strain energy. The first two case studies of first group have the same objective function and the same cantilever plate example introduced in Reference 3. In this section, optimization results for the case studies, with the total strain energy as the objective function, are compared with the results determined in the previous case studies.

In the second group of case studies, the objective function is taken as the first buckling load. In the optimization process, the first buckling load is desired to be maximized. Therefore, different geometry, loads and displacement boundary conditions are created for these case studies conducted under the second group. For this purpose, the main code coupling the Nastran finite element solver and the MATLAB genetic algorithm is updated. The objective function is changed from total strain energy to first buckling load. All the procedures and the case studies are explained in detail in Chapter 4.

The final goal of the thesis is to design manufacturable composite laminates by employing continuous fiber paths in an optimization framework. It is considered that in the near future fiber placement technology will be a popular manufacturing method in the manufacturing of aerospace vehicles.

CHAPTER 2

2- Design Optimization of Variable-Stiffness 2D Orthotropic Plates

In this chapter, optimization of composite structures, made of 2D orthotropic materials is performed. Primarily, discrete fiber orientation angle optimization is carried out. Study of 2D orthotropic structures are easier to handle since only a single orientation angle has to be defined per element in the finite element model. In this section, optimization and finite element analysis is performed by the optimization module of Nastran. Different load conditions, different structures, different initial values of design variables are considered and sample problems are solved. The aim is to have better mechanical properties than the same fiber orientation case for all elements. “Better mechanical properties” are defined as objective functions of the sample problems studied. For instance, minimum total strain energy, which maximizes the stiffness of the plate, and the volume, which makes the structure lighter are some examples of better mechanical properties. In this chapter, some of the simple problems are solved are used to verify that Nastran optimization is working correctly.

2.1- Design Optimization in Nastran

Optimization module of Nastran solves the structural optimization problems by constructing approximate models. In that respect Nastran does not perform finite element analysis in each design iteration. By using the approximate model, sub-steps of the non-linear optimization problem is converted into series of simple linear programming problems [23].

The first and the simplest attempts at linking structural analysis with numerical optimization were based on a direct coupling or “black box” type of approach as seen in Figure 2.1 [23]. Whenever the optimizer needs a function evaluation, the finite element analysis would be performed to get the necesary output values. This method requieres great amount of computer resource since in every iteration one must perform a finite element analysis. Therefore, the great number of analyses that is required makes this approach useless. The numerical optimizer may not only require response derivatives with respect to the design variables, but a number of function evaluations must also be performed during one-dimensional searches. This situation could quite easily lead to hundreds of analyses. Nastran creates concepts that limit the number of required finite element analyses. It is remarked that some commercial structural analysis codes that include an optimization capability still perform their optimization tasks in the “black box” manner shown in Figure 2.1. Although the high performance capabilities of present day computers, the optimization capabilities are still limited in the size of the design task (in terms of number of design variables and responses and the cost of finite element analysis), which they can address relative to what can be achieved using Nastran.

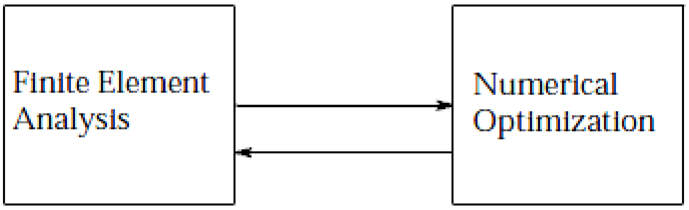


Figure 2.1: Early structural optimization attempts, taken from Ref. [23]

The principal complicating factor in structural optimization is that the response quantities of interest are usually implicit functions of the design variables. When it is considered that the optimizer may be asked to deal with perhaps hundreds of design variables and thousands of constraints, it is not efficient to perform a full finite element analysis each time the optimizer proposes an incremental design change. To avoid these difficulties, certain approximations can be implemented to increase the computational efficiency.

Approximation concepts are actually nothing more than the computational implementation of methods. In many cases, an engineer is handed a stack of analysis data and asked to propose an improved design. This raw data usually contains much more information than is necessary to suggest possible design improvements. The question becomes one of how to reduce the problem sufficiently so that only the most important information is considered in the process of generating a better design. A first step may be to narrow the design task to that of determining the best combination of just a few design variables. It is really hard for a designer to consider fifty or one hundred variables simultaneously and expect to find a suitable combination from the group. It is much more efficient to link these together for the design problem if possible. It would be advantageous if all the design variables could be varied in a suitably proportional manner. Describing a shape defining polynomial in terms of just a few parameters, or allowing only linear variations of plate element thicknesses, are both examples of types of so called “design variable linking” [23]. The next step might then be to identify the few constraints that are violated or nearly violated. There may be just a few constraints which are currently guiding the design, such as stress concentrations, while others, such as the first bending mode, may be nowhere near critical and can be temporarily disregarded. In structural optimization, this is a "constraint deletion" process [23]. Constraint deletion allows the optimizer to consider a reduced set of constraints, simplifying the numerical optimization phase. This also reduces the computational effort associated with determining the required structural response derivatives. Thus, sensitivity analysis costs are reduced as well.

Once the engineer has determined the constraint set that seems to be driving the design, the next step might be to perform some sort of parametric analysis in order to determine how these constraints vary as the design is modified. The results of just a few analyses might be used to propose a design change based on a compromise among the various trial designs. A parametric study of the problem is carried one step further in structural optimization with formal approximations, or series expansions of response quantities in terms of the design variables. Formal approximations make use of the results of the sensitivity analysis to construct an approximation to the true design space which, only locally valid, is explicit in the design variables. The resultant explicit representation can then be used by the optimizer whenever function or gradient evaluations are required instead of the costly, implicit finite element structural analysis.

This coupling is illustrated in Figure 2.2. The finite element analysis forms the basis for creation of the approximate model which is subsequently used by the optimizer. The approximate model includes the effects of design variable linking, constraint deletion, and formal approximations. Design variable linking is established by the engineer, while constraint deletion and formal approximations are performed automatically in Nastran.

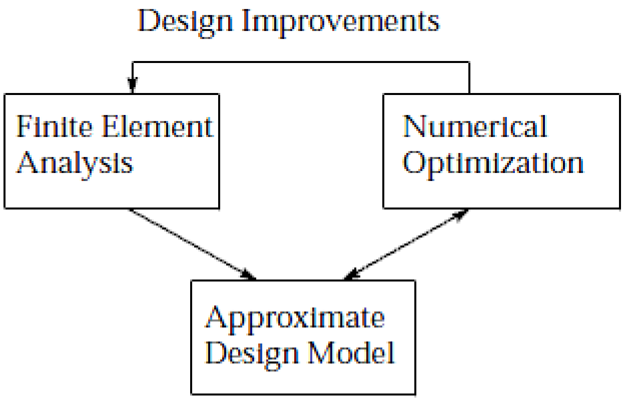


Figure 2.2: Coupling analysis and optimization using approximations, taken from Ref [23]

Once a new design, based on the approximate model, has been proposed by the optimizer the next step would most likely be to perform a detailed analysis of the new configuration to see if it has actually managed to satisfy the various design constraints and reduce the objective function. This reanalysis update of the proposed designs is represented by the upper segment of the loop in Figure 2.2. If a subsequent approximate optimization is necessary, the finite element analysis serves as the new baseline from which to construct another approximate subproblem. This cycle may be repeated as necessary until convergence is achieved. These loops are referred to as design cycles.

As Figure 2.3 shows the optimizer interacts with the approximate model rather than the finite element model and produces an improved design. Once the improved design is obtained the finite element model is updated based on the results from the optimizer so that a new finite element analysis can be performed.

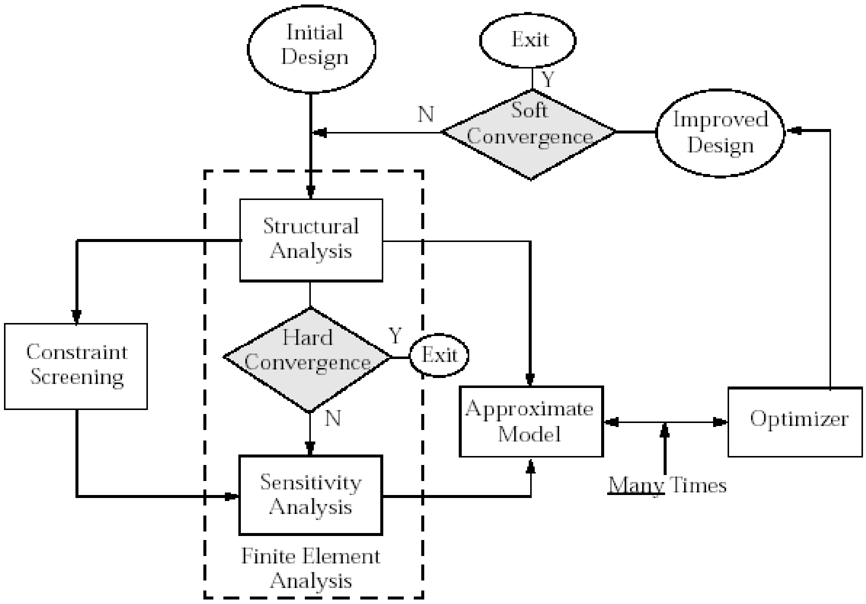


Figure 2.3: Nastran Implementation of structural optimization [24]

A key part of the implementation of the optimization process is to determine when to stop the iterations. There are two levels at which convergence is tested. The lower level is at the optimizer level, and it is at this level where the optimizer decides on the optimized solution based on the output of the approximate model. The second and higher level is with respect to the overall design cycles. Figure 2.3 shows the locations of higher level of convergence tests. As shown in Figure 2.3 hard convergence compares the most recent finite element analysis with those from the previous design cycle. Since this test compares exact results from two consecutive analyses, it is named as hard convergence. This test is used as the default test for determining whether or not to terminate the design-cycle process. On the other hand soft convergence compares the design variables and properties output from the approximate optimization with those of the input to the approximate optimization. If design variables and properties have not changes appreciably another finite element analysis may not be asked for.

2.2- Optimization of Material Orientation of 2D Orthotropic Plate Composed of 8 Elements

In this section, optimization of material orientation of a 2D orthotropic plate is performed by the optimization module Nastran, SOL 200. As a test case a plate composed of eight elements is considered and plate, which is subjected to different loading conditions, is optimized. For the case studies with the 2D orthotropic materials in Chapter 2 the material properties are taken from Reference 3. The directions for the material properties are shown in Figure 2.4.

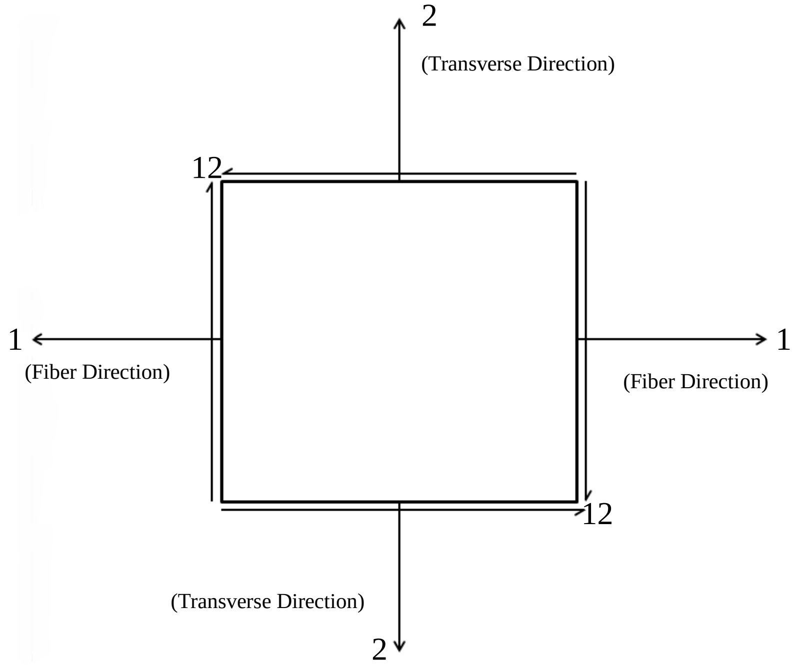


Figure 2.4: Material directions for the case studies

$$\frac{E_1}{E_2} = 8.0, \qquad \frac{G_{12}}{E_2} = 0.35, \qquad \gamma_{12} = 0.3$$

Unidirectional composites act like 2D orthotropic materials. In the later chapters of this thesis composite materials are going to be considered. Therefore, in this chapter a composite material is selected and the same material properties are used in later chapters. In this study, carbon\epoxy unidirectional composite material's longitudinal elastic modulus is taken as the reference. The rest of the properties are calculated according to the longitudinal elastic modulus. The values are below.

$$E_1 = 134\, GPa, \qquad E_2 = 16.75\, GPa, \qquad G_{12} = 5.8625\, GPa, \qquad \gamma_{12} = 0.3$$

$$t = 0.2\, m$$

These material properties are used in this 8 element case studies and the following case study. The total plate thickness is $0.2\, m$ for the problems.

2.2.1- 2D Orthotropic Plate Composed of 8 Elements Subject to Out of Plane Loading

In this case study objective function is taken as the total strain energy. When total strain energy is minimum, the stiffness of the structure is maximum or the compliance is minimum. Design variables are taken as the material orientation angles of each element. Since the plate is composed of 8 elements, there are 8 design variables. The initial values of the design variables are taken as 30 degrees. The load is a pressure perpendicular to the upper surface of the plate. The load and the boundary conditions are shown in Figure 2.5. Loads are in Pascal. Plate is clamped at the left edge. Figure 2.6 shows the graphical representation of the optimized material orientation angles.

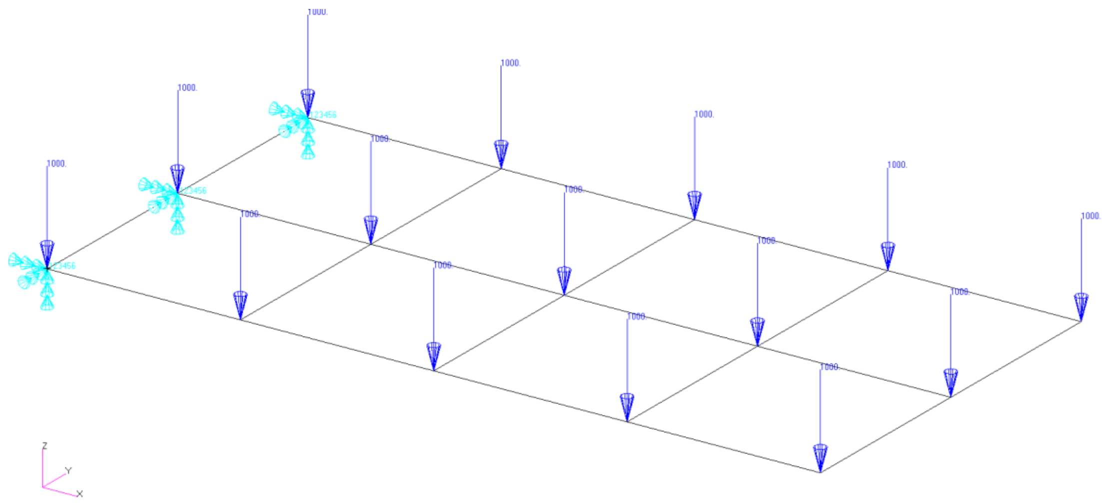


Figure 2.5: Loads and boundary conditions for the orthotropic plate under out of plane load

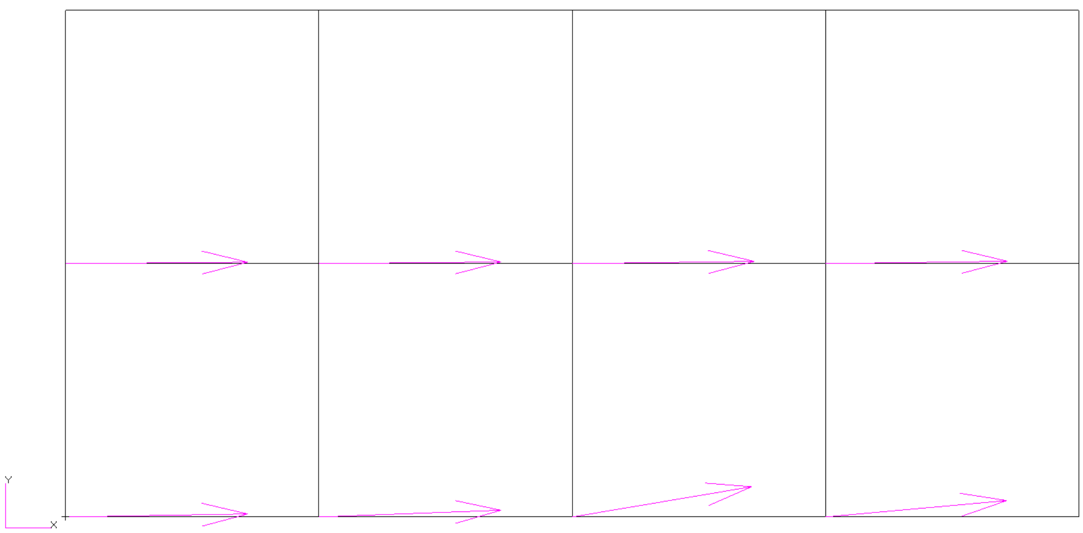


Figure 2.6: Optimized material orientation angles for the orthotropic plate under out of plane load.

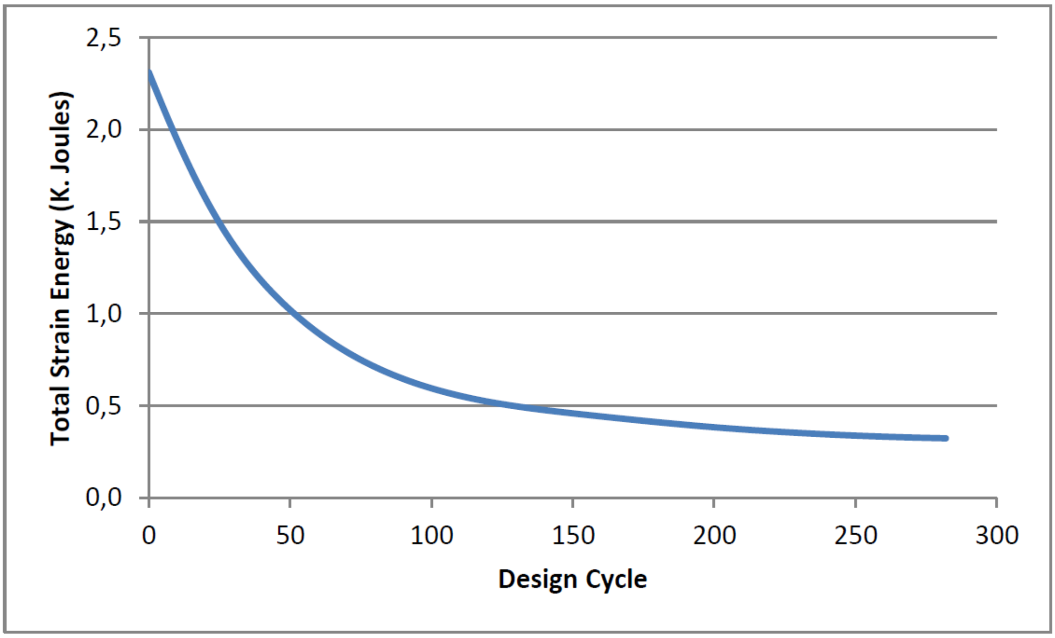


Figure 2.7: Design cycle vs. Objective function graph for the out of plane load case

From Figure 2.6 it is seen that optimum material orientation angle that minimizes the strain energy is determined as 0 degree by the Nastran optimization module. It should be noted that for this problem, since the plate is under transverse loading with long side edges free, minimum compliant structure is the one which has material orientation in the longer dimension of the plate. Thus, when the material orientation coincides with maximum stress direction, structure has highest stiffness and lowest total strain energy. Figure 2.7 shows that the Nastran optimization minimizes the objective function since the optimum criterion is reached after a certain number of design cycles. The optimum value of the total strain energy for this case study is 323 Joules.

2.2.2- 2D Orthotropic Plate Composed of 8 Elements Subject to In-Plane Shear Loading

For the in-plane shear loading objective function is again taken as the total strain energy. The initial values of the material orientation angles are taken as 30 degrees as before. The structure is clamped on the force edges of the geometry. Figure 2.8 show the loads and boundary conditions for the geometry. The loads are in Newton.

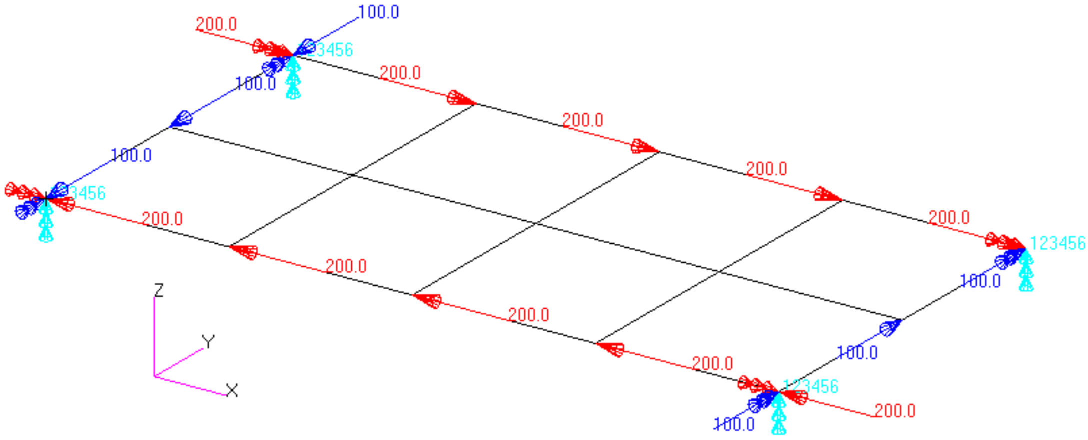


Figure 2.8: Load and boundary conditions for the in-plane shear loaded structure

By running Nastran optimization module the orientations in Figure 2.9 are found.

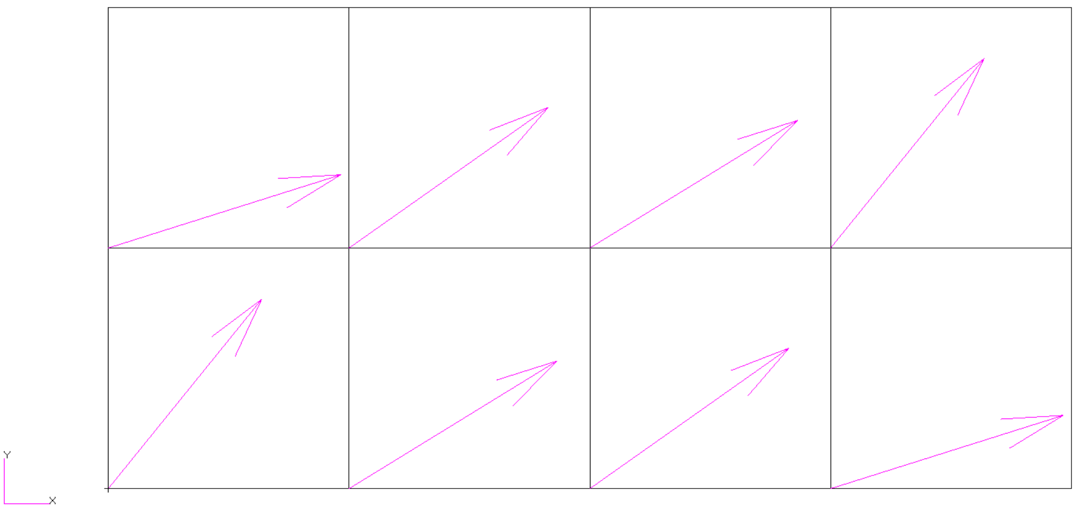


Figure 2.9: Resultant orientations for the case study for in-plane shear load

The optimization process can be seen from Figure 2.10, which shows variation of the objective function with respect to the design cycle.

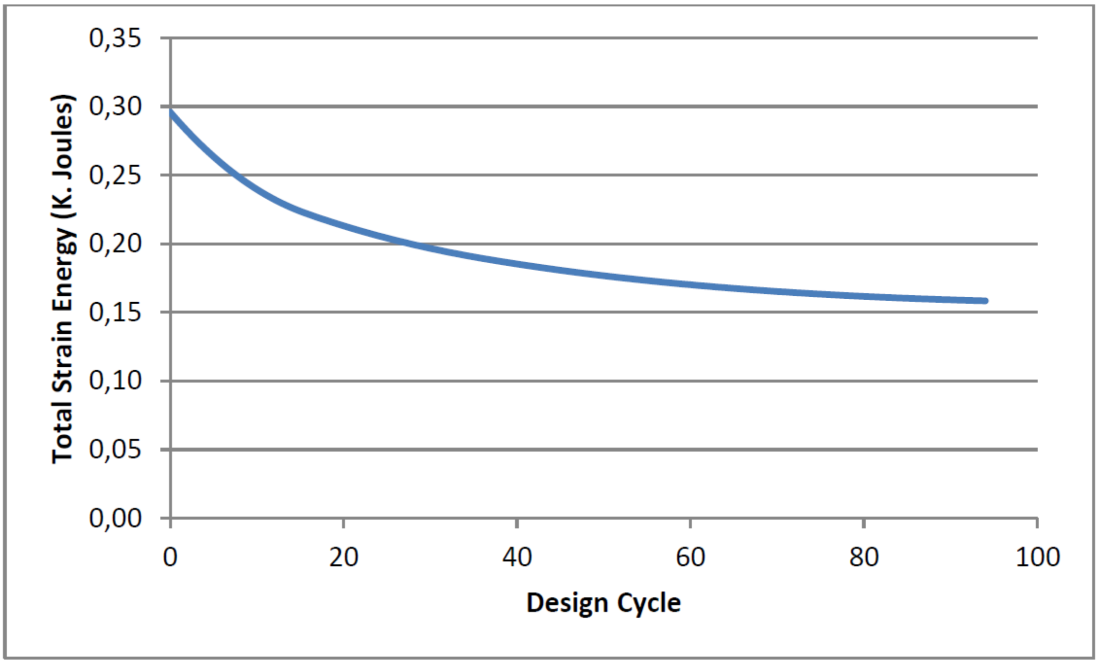


Figure 2.10: Design cycle vs. objective function graph for the in-plane shear load case

The minimum value for the total strain energy is determined as 158.42 Joules. These two case studies show that the Nastran optimization module works fine for the optimization of the material orientation angles of the 2D orthotropic materials. However, it should be noted that in present case studies there are 8 design variables only. The following case studies are done for higher number of design variable case and, more complex shapes such as a plate with a center cut-out.

2.3- Optimization of Material Orientation of 2D Orthotropic Rectangular Plate Composed of 736 Elements and in In Plane Bending Load

This case study is taken from the work of Setoodeh et.al. [3]. In Reference 3, cantilevered plate, shown in Figure 2.10, is optimized for minimum compliance by employing cellular automata based strategy for optimal design of curvilinear fiber paths to improve in-plane response of composite laminate within the context of structural design. Plate of aspect ratio three is subjected to a uniformly distributed load of 1.0 kN/m as shown in Figure 2.11.

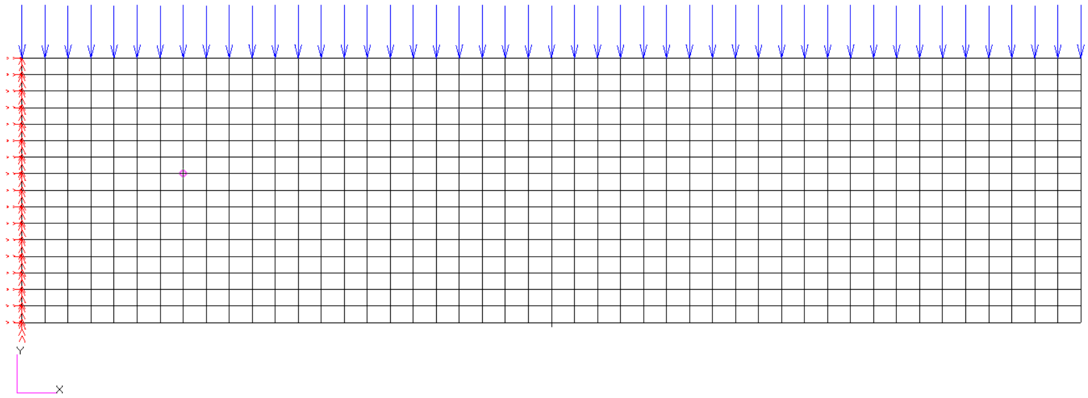


Figure 2.11: The load and boundary condition for the following case studies performed

The same material properties with the total thickness for the structure in the previous section are used in the simulations in this part which are:

$$E_1 = 134\ GPa, \qquad E_2 = 16.75\ GPa, \qquad G_{12} = 5.8625\ GPa, \qquad \gamma_{12} = 0.3$$

$$t = 0.2\ m$$

In Reference 3, optimal distribution of material orientation at the element level is given for three different mesh sizes. Figure 2.12 gives the optimal orientations by the coarse mesh which has 46x16 elements. Therefore, there are no numerical total strain energy values for the case study. Only material orientations of the performed case studies and the case study in Reference 3 are compared.

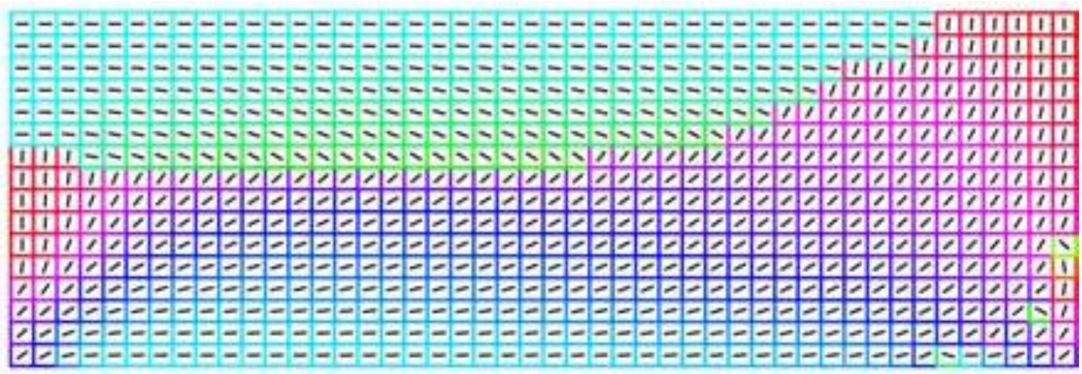


Figure 2.12: Optimal orientations for the coarse mesh (46x16) taken from Ref. [3].

Before presenting the optimal orientation results obtained by the optimization module of Nastran, a static analysis is performed for random material orientation case, and the total strain energy is determined. Such an analysis is performed so that the total strain energies obtained at the end of the optimization analyses performed by Nastran can be compared with the random material orientation case. Although such a comparison does not totally verify that results obtained by Nastran correspond to global optimum, comparing the strain energies obtained by the optimization analysis with the random material orientation case allows one to interpret the results of the optimized solutions.

2.3.1- Static Analysis of the Rectangular Plate with Random Material Orientation Angles

Static analysis of the plate is performed by assigning random material orientation to the 736 elements which constitute the finite element model of the cantilevered plate. No optimization is performed and only static finite element solution is conducted and total strain energy of the plate is determined. Figure 2.13 shows the random material orientation angles assigned to each element of the finite element model.

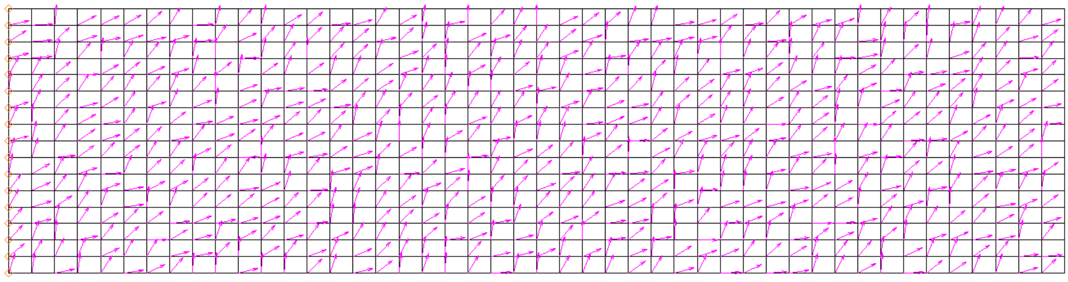


Figure 2.13: Random material orientations assigned to the elements of the cantilevered plate model

As a result of the static analysis, the total strain energy is determined as 29,02 K.Joules.

2.3.2- Optimization using initial material orientation angle of 5^o

The first optimization for the material orientation angles is performed with the initial value of 5 degrees for the material orientation angles of the elements. Since the finite element model has 736 elements, there are 736 design variables. The lower boundary of the design variables is taken as -90^o degrees and upper boundary is taken as 90^o degrees. The objective function is defined as the minimization of the total strain energy. The loading and the boundary conditions are given in the beginning of this section. Optimized material orientation angles predicted by Nastran are shown in Figure 2.14 below. For the particular case, variation of the total strain energy with the design cycles is shown in Figure 2.15.

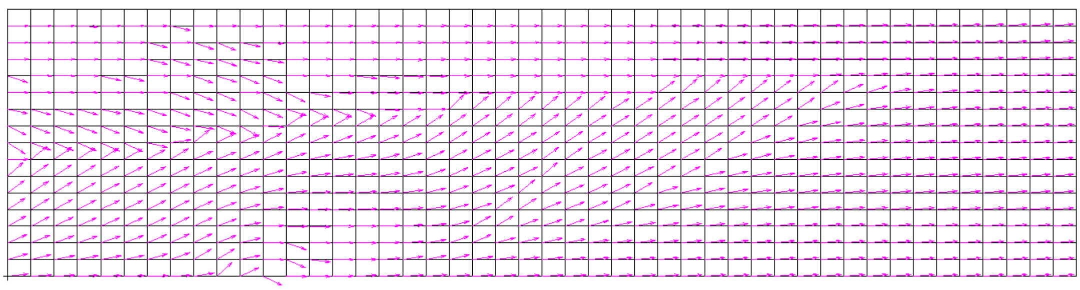


Figure 2.14: Optimized material orientation angles for the 5 degree initial material orientation

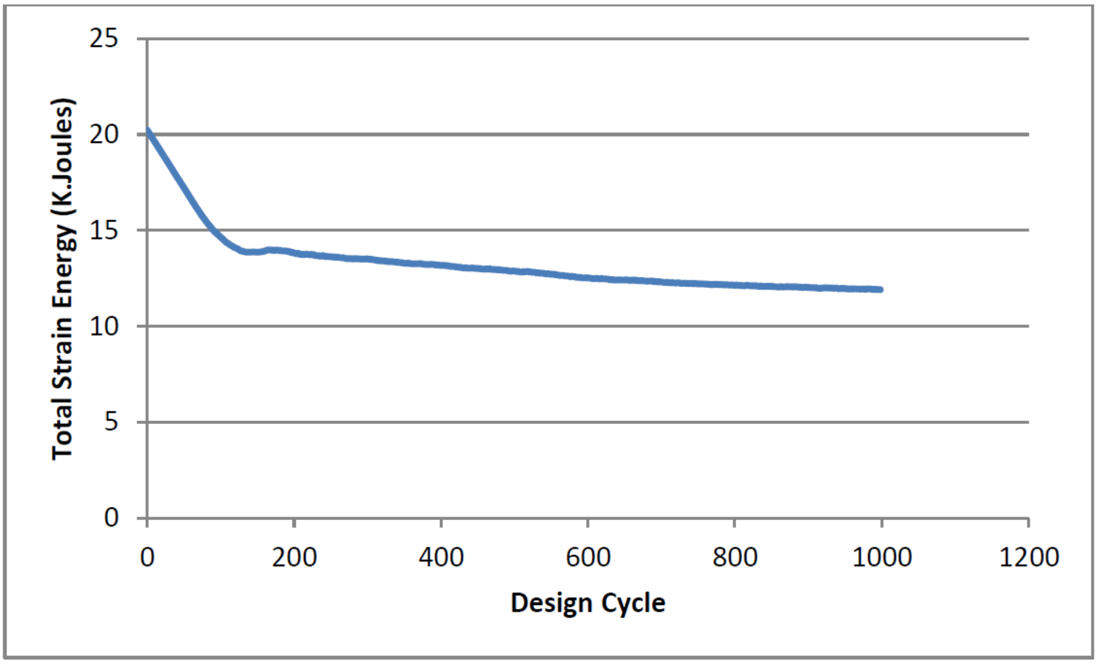


Figure 2.15: Total strain energy versus objective function graph for the 5 degree initial material orientation

For the 5 degree initial material orientation angle case, in the optimized configuration the total strain energy is determined as 11.9 K.Joules. It is seen that optimization performed with the 5 degree initial material orientation angle yielded a total strain energy which is approximately 2.5 times less compared to the total strain energy of the random material orientation case. The total run time for Nastran in this case study is about 7 hours in 2.8 Ghz dual processor with 8 Gb memory computer. However, it should be noted that the optimized material orientation angle distribution shown in Figure 2.14 may not give the global minimum total strain energy, because gradient based optimization is sensitive to the initial values of the design variables, especially when the number of design variables is very high. Therefore, in the following optimization is performed by assigning different initial values to the material orientation angles.

Comparison of Figure 2.14 with Figure 2.12 of Reference 3 reveals that for the 5 degree initial material orientation angles determined by Nastran do not identically agree with the optimized material orientation angles determined in Reference 3. The main deviation is seen near the free end where in the present study optimized orientations are close to 0 degree whereas Reference 3 predicts higher material orientation angles. However, it should also be noted that cellular automata optimization strategy determines negative material orientation angles in some elements near the free end of the plate. In general, it is seen that there are sharp changes in material orientation angles along certain borders such as the one near the top edge of the cantilevered plate.

2.3.3- Optimization using initial material orientation angle of -5^{o}

For the -5 degree initial material orientation angle, Figure 2.16 gives the optimized material orientation angle distribution determined by Nastran. It is seen that for the -5 degree initial material orientation angle, the optimized material angle distribution is very different from the result given in Figure 2.12. The large deviation is an indication that the orientation angle distribution given in Figure 2.16 gives a local minimum for the total strain energy.

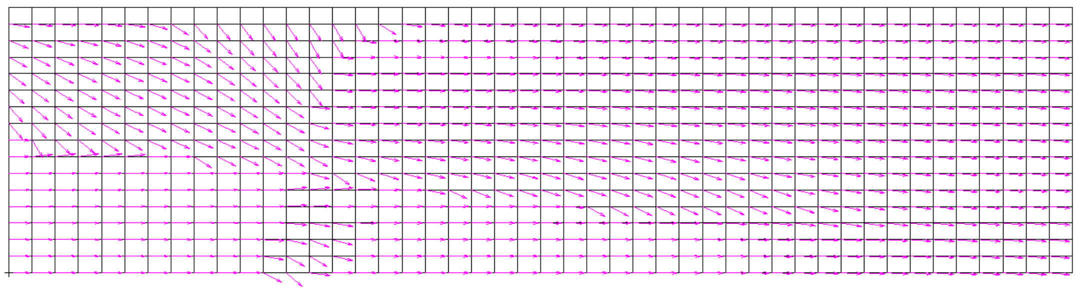


Figure 2.16: Optimized material orientation angles for the -5 degree initial material orientation

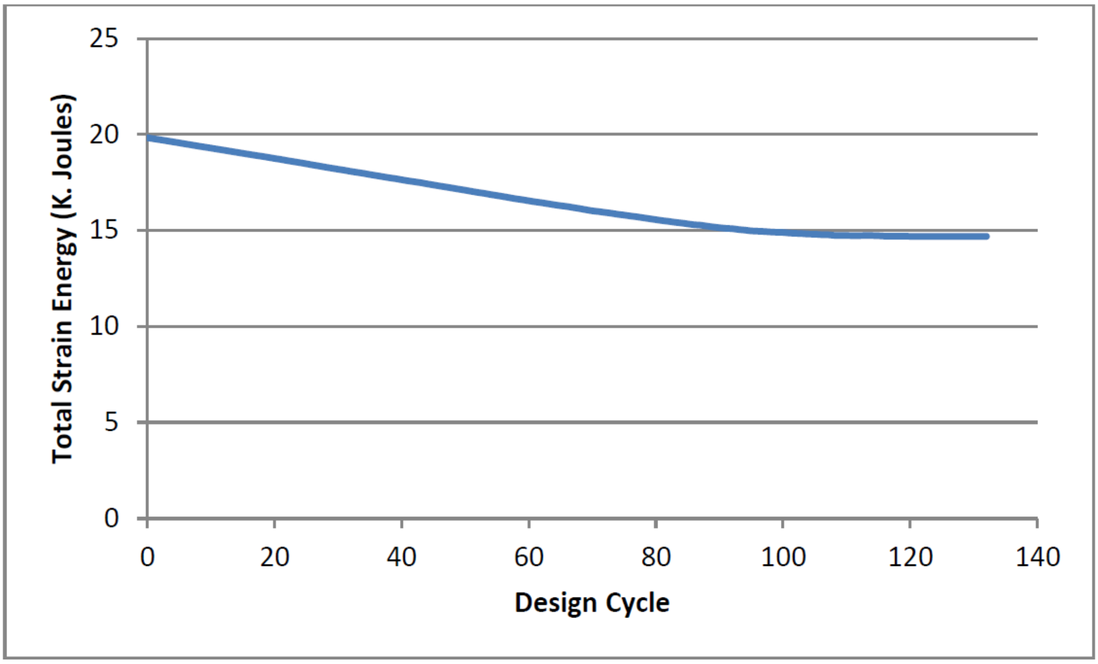


Figure 2.17: Total strain energy versus design cycle graph for the -5 degree initial material orientation

For the -5 degree initial material orientation angle distribution, the total strain energy in the optimized configuration is determined as 14.7 K.Joules. Figure 2.17 shows the result is not much different than the total strain energy obtained for the positive 5 degree initial material orientation case. Thus, it can be commented that for the 5 and -5 degree initial material orientation, the optimized configurations may be stuck in local minimums. However, $+5^{o}$ initial fiber orientation case results in lower strain energy which is an indication that -5^{o} initial fiber orientation case definitely results in local minimum. Therefore, in gradient based optimization strategy, different solutions must be performed by assigning different initial values to the design variables so that the variation of the objective function with respect to the initial value of the design variables can be obtained. Obviously, for the problems with many design variables, it is impossible to carry out optimization for every possible combination of initial values for the design variables. However, depending on the particular problem studied, one can base the selections of the initial values of the design variables on engineering judgment, and come up with different initial values sets with which the global optimum can be encompassed. The total run time of

Nastran in particular case study is about 3 hours with a 2.8 Ghz dual processor with 8 Gb memory computer.

2.3.4- Optimization using initial material orientation angle of 15°

For the 15 degree initial material orientation angle, Figure 2.18 gives the optimized material orientation angle distribution determined by Nastran. It is seen that for the 15 degree initial material orientation angle, the optimized material angle distribution is similar to the material orientation shown in Figure 2.12. Better agreement of the optimum material orientation distribution determined in the present study and in Reference 3 could be due to the use of 15 degree initial for the design variables which probably prevented the optimum solution to get stuck in a worse local minimum. It is noted that the optimized solution is now much near the global optimum.

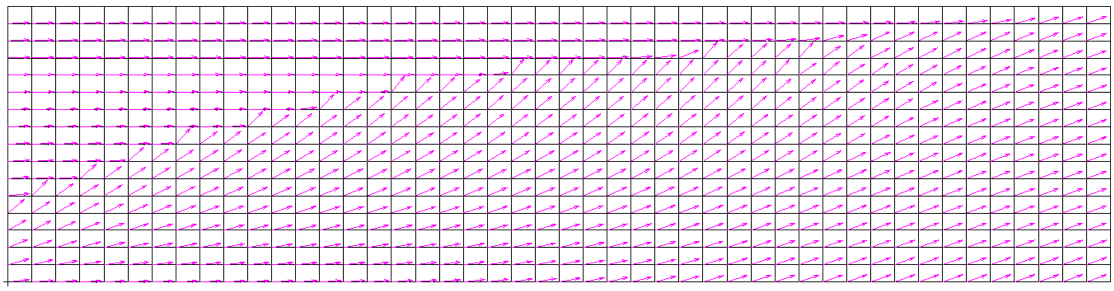


Figure 2.18: Optimized material orientation angles for the 15 degree initial material orientation

For the 15 degree initial material orientation case, Figure 2.19 gives the variation of the total strain energy with the design cycles. Figure 2.19 shows that in each design iteration, the total strain energy decreases until no significant improvement is observed and hard convergence is achieved.

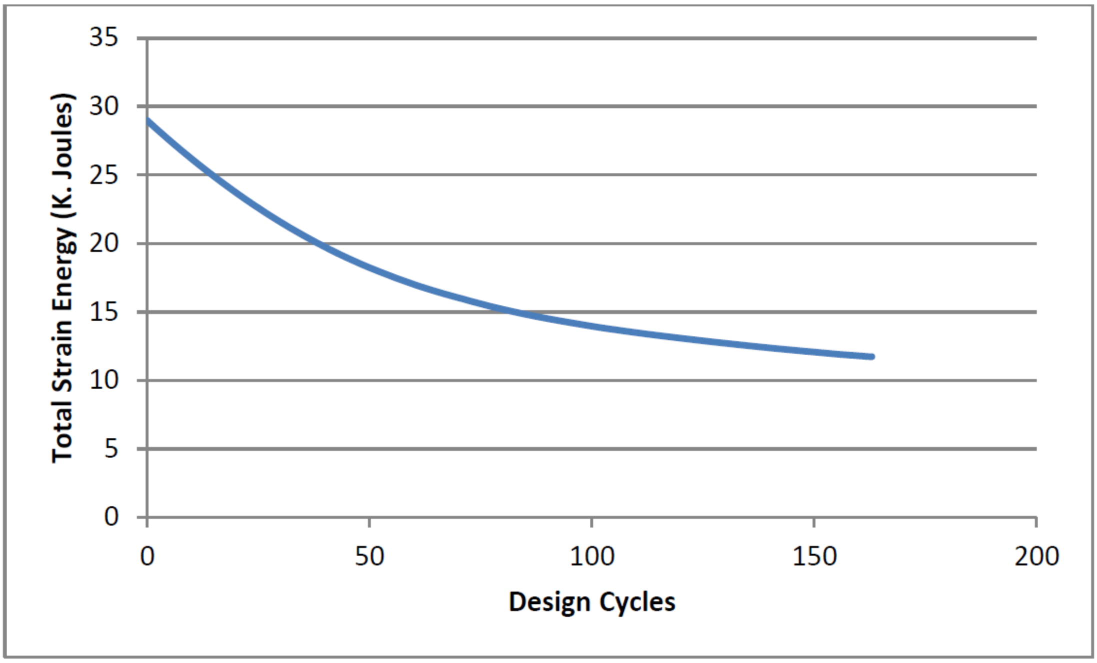


Figure 2.19: Design cycle vs. objective function graph for the 15 degree initial material orientation

The resulting total strain energy value for this case is 11.3 K. Joules. It is seen that for the 15 degree initial material orientation case optimized value of the total strain energy is smaller than the previous cases which is an indication that the material orientation distribution shown in Figure 2.18 is close to the global optimum. It should also be noted that this solution gave a smoother material orientation distribution over the most part of the cantilevered plate. The total run time of Nastran in the 15^{o} initial material orientation is about 26 minutes 2.8 Ghz dual processor with 8 Gb memory computer.

From Figure 2.18, it can be seen that near the most of the top and bottom edges of the plate material orientation angles are horizontal. Since near the top and bottom edges of the cantilevered plate maximum principal strains are along the longer edge of the plate, it is reasonable that the material orientations are aligned with the principal strain direction to reduce the total strain energy. However, away from the edges since the plate is two dimensional, material orientations deviate from being purely horizontal. Near the mid plane of the plate, it is seen that material orientation angles are in the diagonal direction which more or less coincide with the maximum principal strain directions. However, it should be noted that based on the study of Pedersen [25], in orthotropic materials principal strain directions do not always correspond to material orientations in order to minimize the total strain energy. Therefore, one should be cautious when interpreting the results of the optimization.

2.3.5- Optimization using initial material orientation angle of 25^o

This case is performed to see the effect of further increase in the initial material orientation angle on the optimized configuration. The resulting optimized material orientation distribution is shown in Figure 2.20.

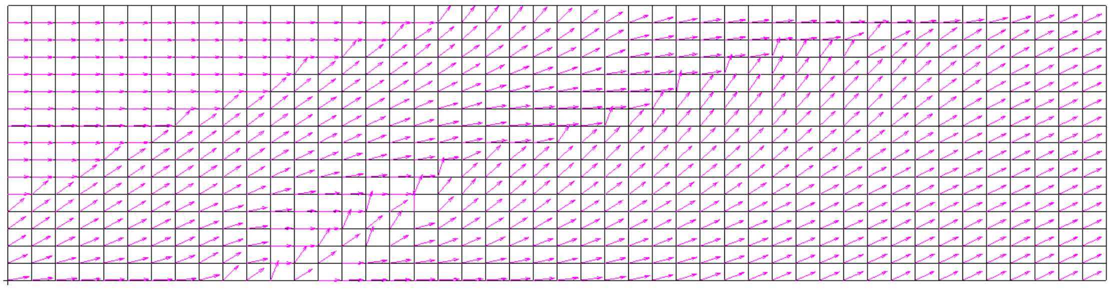


Figure 2.20: Optimized material orientation angles for the 25 degree initial material orientation

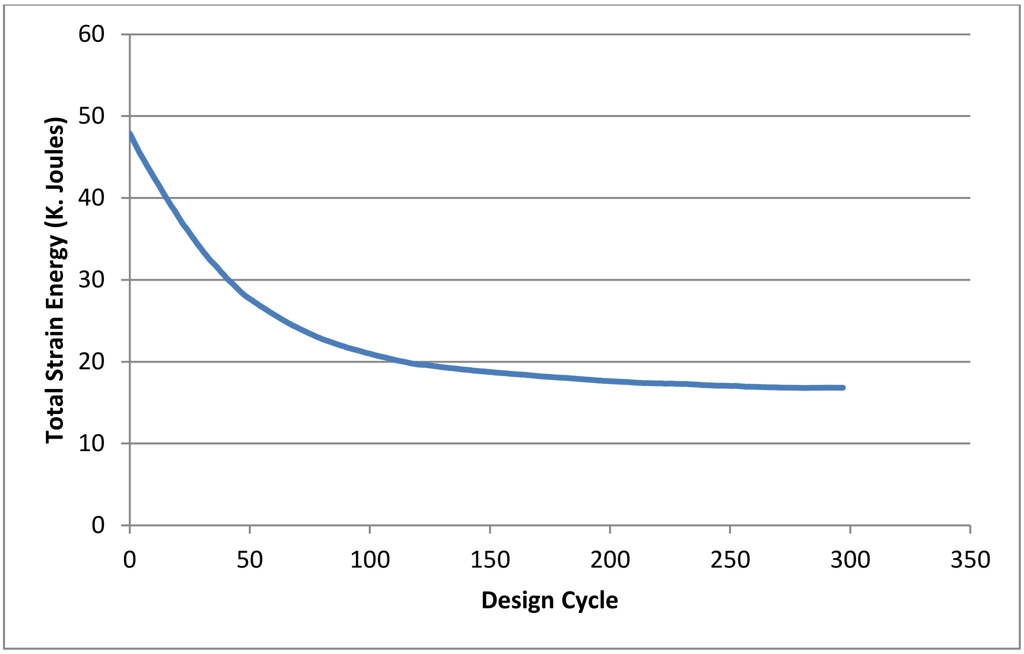


Figure 2.21: Total strain energy versus design cycle graph for the 25 degree initial material orientation

For this case the optimized total strain energy is determined as 16.8 K. Joules. It should be noted that although Figure 2.21 shows that convergence is achieved, the optimized total strain energy is higher than the minimum total strain energies obtained by using lower initial material orientation angles. For instance, it can be seen that for the 25 degree initial material orientation case, the optimized total strain energy is higher than the 15 degree initial material orientation case. Therefore, it can be said that the resulting configuration is a local optimum not a global one. The total run time for Nastran is 1 hour for the particular case study.

2.3.6- Comparison of the Case Studies Employing Different Initial Material Orientation Angles

It can be seen from the above results that the use of different initial material orientation angles in the optimization process gives different optimum material orientation angle distributions with different optimized total strain energies. It is concluded that with the different initial value of the material orientation angles, optimization problem approaches to different local minima. Among the case studies studied, 15 degree initial material orientation case gave the lowest strain energy with material orientation angle distribution which is somewhat closer to the distribution predicted in Reference 3. However, it should be noted that in the case studies, same initial material orientation angles are assigned to all elements. By trying different combination of initial values, optimized results which are closer to the global optimum can be obtained. Table 2.1 shows the results for the optimum total strain energy values for different initial values.

	Optimum Energy
Random Orientation Angle	29.02 K.Joules
Initial Orienation of 5 Degrees	11.9 K.Joules
Initial Orienation of -5 Degrees	14.7 K.Joules
Initial Orientation of 15 degrees	11.3 K.Joules
Initial Orientation of 25 degrees	16.8 K.Joules

Table 2.1: Table for the static solution for the random orientations and the optimum solutions for different initial values.

Based on the case studies conducted, it can be concluded that gradient based optimization method is highly sensitive to the initial values of the design variables. In optimization problems with many design variables, such as the one studied above with 736 design variables, gradient based methods can be stuck in local minima. The run times are also changing with respect to the initial material orientations. For better initial guesses the run time is getting shorter. Therefore, for problems involving high number design variables, gradient based optimization methods are not very suitable in determining the global optimum solution.

2.4- Optimization of Material Orientation of 2D Orthotropic Rectangular Plate with a Circular Cut-Out in Center

There are several case studies with the composite structures [14; 26]. Thus, this case is again taken from Reference 3. In this problem, a square plate with a central circular cut-out is analyzed, and optimum discrete material orientation angles at the element level are determined with the objective being the minimization of the total strain energy. In the analysis and optimization process, the square plate is divided into four equal sections, and only one quarter of the plate is analyzed. However, in the present study a different mesh is used than the one used in Reference 3. Figure 2.22 shows the geometry and mesh used for the case study. The mesh is the same as Reference [27].

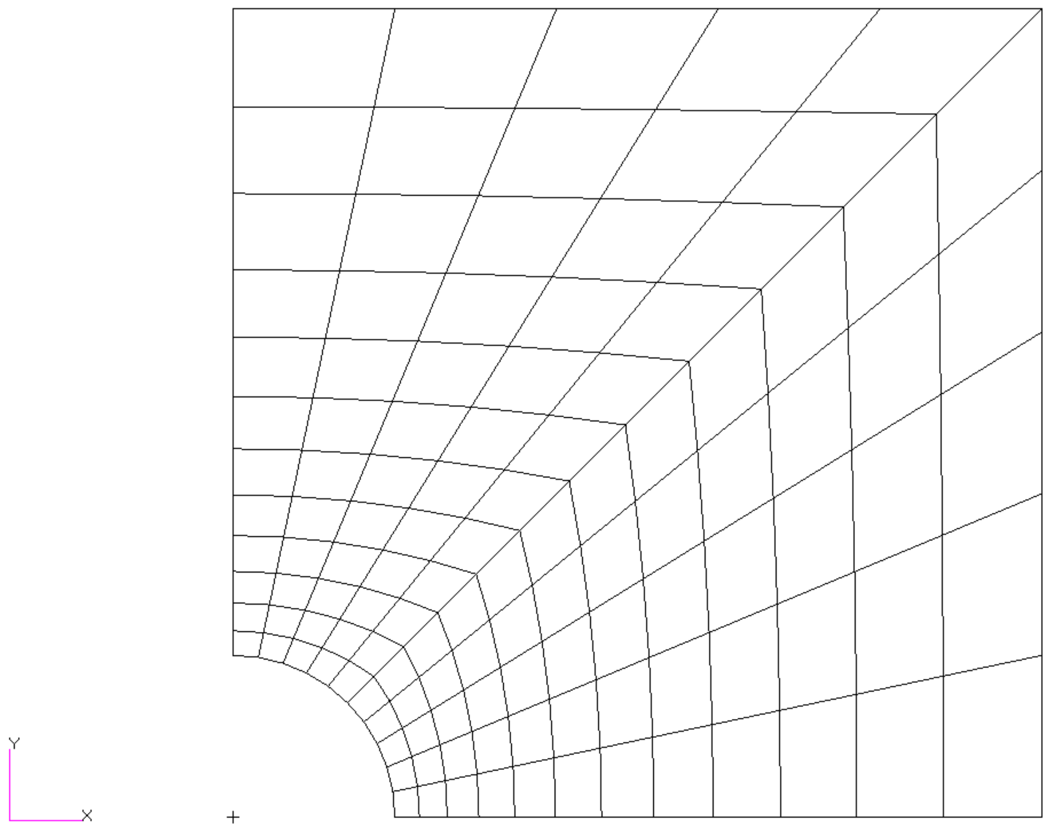


Figure 2.22: The mesh and the geometry for the circular cut-out square plate

Nastran has its unique material and element orientations. The material and element orientations are defined with respect to the coordinate systems defined within Nastran. The material and element coordinate systems can be the same or different with respect to the geometry of the element. Figure 2.23 shows the material and element material axes for a CQUAD4 element, which is the mesh type used for the calculations. Using different material and element coordinate systems is useful when using orthotropic or anisotropic material properties [23].

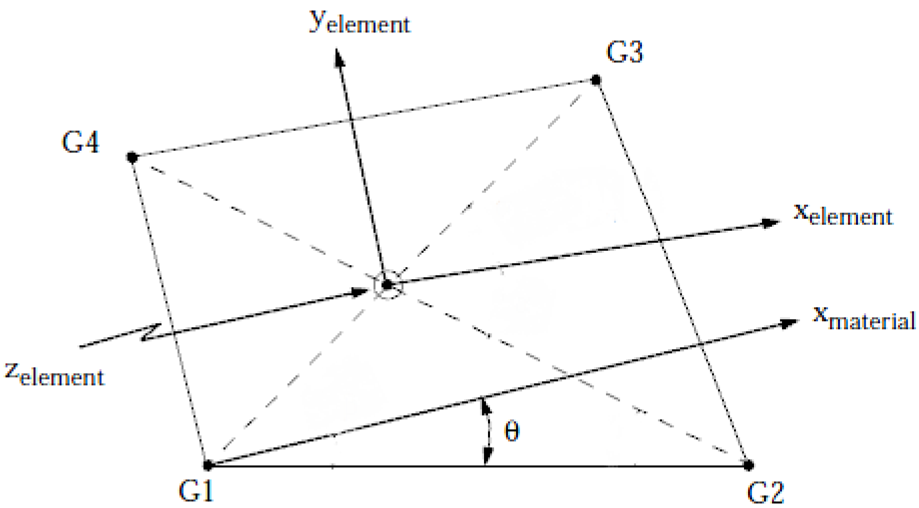


Figure 2.23: The element and material coordinate systems defined in Nastran with CQUAD4 elements from Reference [23].

The material orientations are defined as vectors in Nastran so that all the elements have their zero orientations along the *x* axis. The orientation of the element coordinate system is defined by the order of the connectivity for the grid points. Therefore, if the order of the grid points connectivity are changed, the direction of the positive normal also reverse. When the elements are rectangular CQUAD4 elements, the element and material axes coincide. For this case study, the elements are not rectangular. If there is no rotation angle input for the material orientation, the *x* material axis is defined as the G1G2 line. In the previous case studies and upcoming case studies the orientations are shown from the G1 point of the element.

Figure 2.24 shows the material orientations for this case study. All analysis in this section is going to be performed with respect to the orientations, parallel to the global *x* axis, shown in Figure 2.24. If this modification is not performed, every element has its own material orientation with respect to the grid point's connectivity. In this case, while inputting an initial value Nastran calculates rest of the problem with respect to the material coordinate system. This would affect the solution for the problem. On the other hand the visualization of the problem would be difficult since every element has different material coordinate systems.

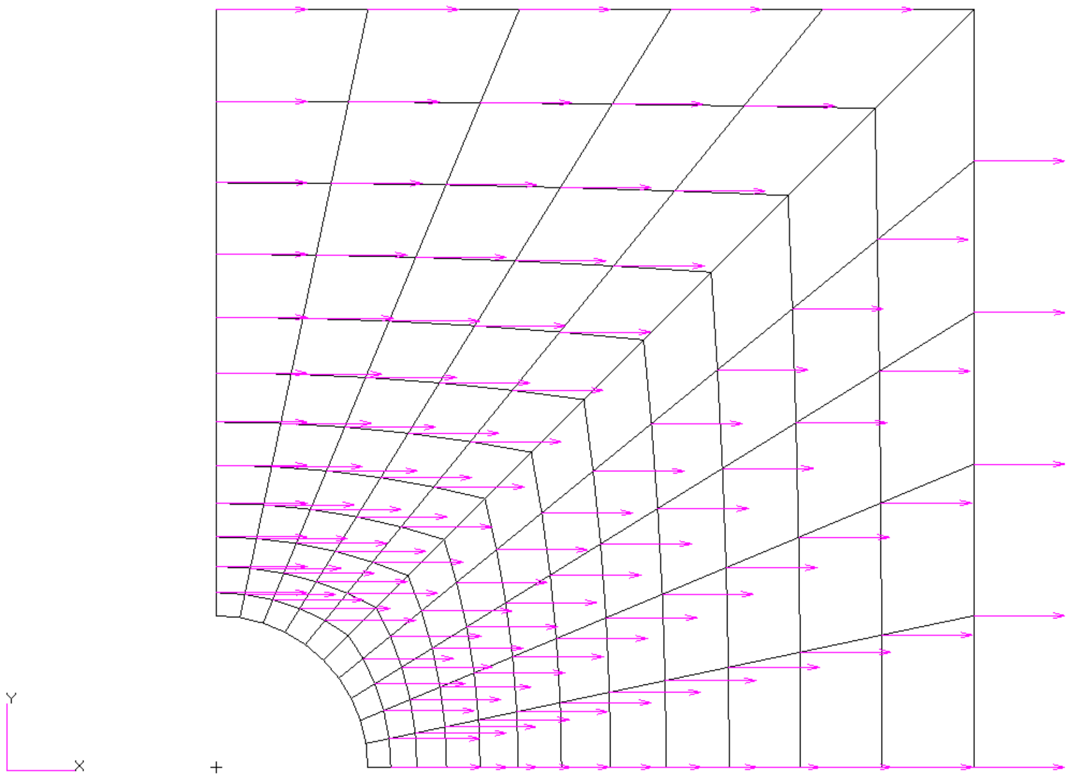


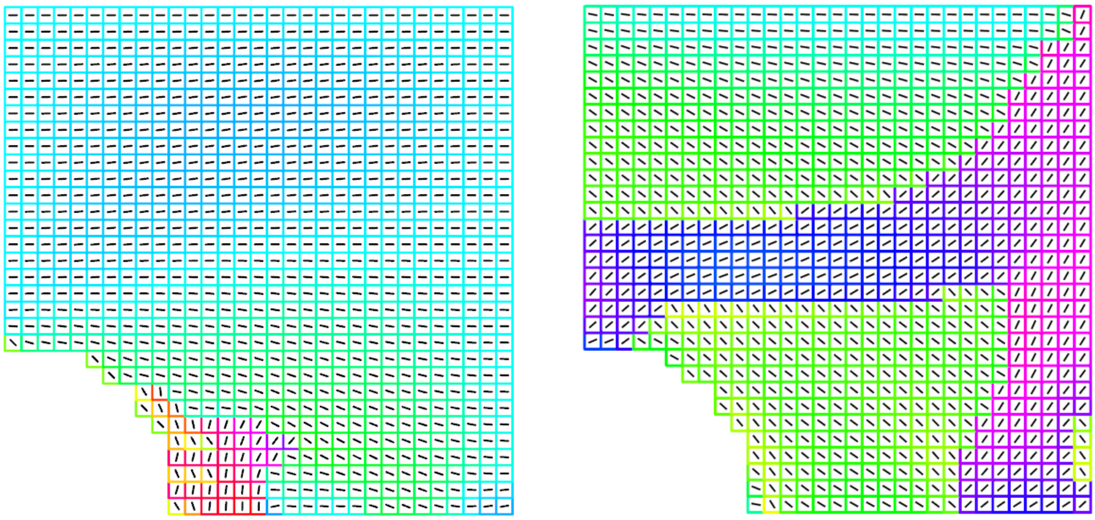
Figure 2.24: Zero degree material orientations for the current case study

In this section the same material properties given in the previous section are used. The same elastic, shear and Young's modulus are used. The thickness is given below.

$$t = 0.02\, m$$

The actual geometry is a 1 m to 1 m square. The working geometry for this case study is 0.5 m to 0.5 m with a circular cut-out with 0.1 m radius.

In Reference 3 all the elements are square elements, as shown in Figure 2.25. In this case study, CQUAD4 elements of Nastran are used to model the plate with a central hole, and the edges of all elements are not parallel to the *x* and *y* coordinates, as seen in Figure 2.22. The optimum material orientations, to achieve minimum compliance for the in-plane axial and shear load cases studied in Reference 3, are given in Figure 2.25. In Figure 2.25, the lines show the material orientations determined for the elements at the end of the optimization process using cellular automata [3].



(a) (b)

Figure 2.25: Optimum material orientations for the minimum compliance determined in Reference 3 (a) for in-plane normal load, (b) for in-plane shear load

It is aimed to have similar results with respect to the cases in Reference 3. In Reference 3 the compliance is minimized which means that the total strain energy has to be minimized. For the case studies in this section the total strain energy is taken as objective function to minimize. Again, Nastran is used for the case studies performed in this section.

2.4.1- 2D Orthotropic Square Plate Composed with a Circular Cut-Out in Center under In-Plane Loading

The first case study is the in-plane load case. The load is applied to the right edge of the cut-out plate, which is shown in Figure 2.27. The total force acting is 5000 N which is distributed along the right edge.

For the in-plane loading, boundary conditions shown in Figure 2.26 are used. Left edge of the plate is fixed in the x direction to create symmetry condition with respect to y axis passing through the center of the hole. Similarly, the bottom edge of the plate is fixed in y direction to create symmetry condition with respect to x axis passing through the center of the hole. To prevent singularity in the finite element analysis, the lower bottom rightmost node is also prevented to move in the out-of-plane direction by fixing the z displacement and rotations with respect to x and y axes.

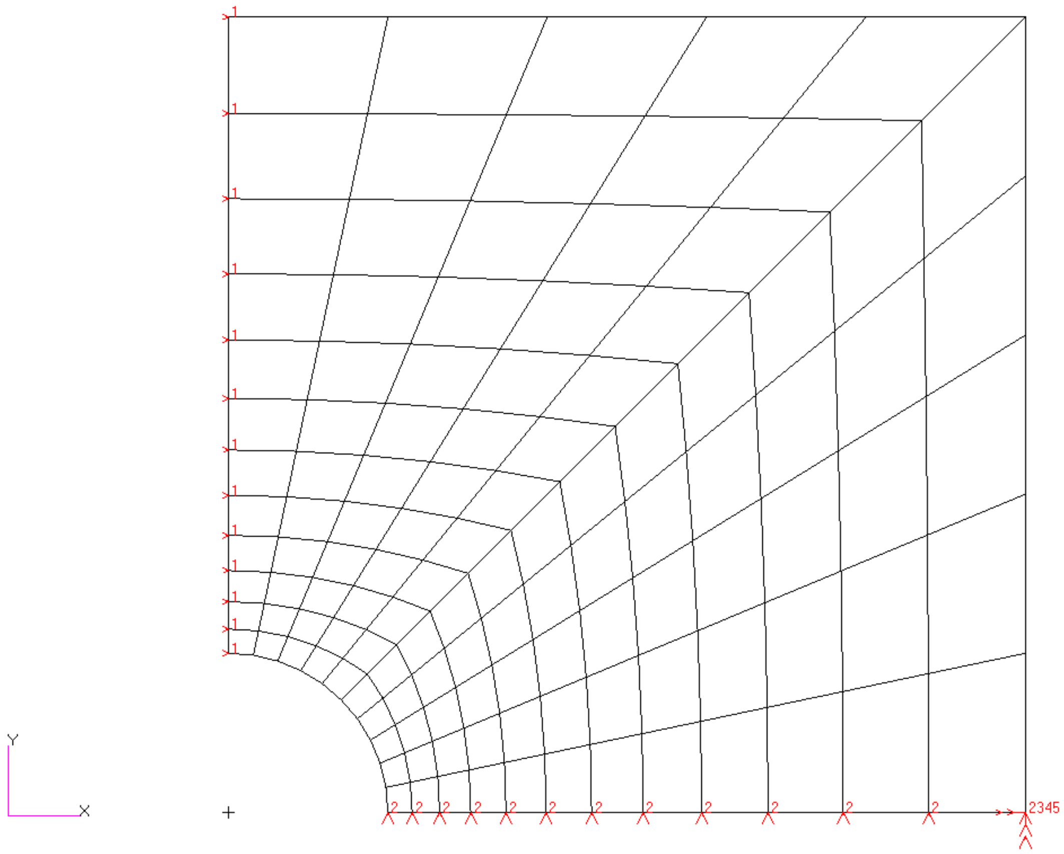


Figure 2.26: Displacement constraints of the geometry for the case study.

The initial material orientations are given as 15^o. From the previous case studies, it is seen that 15^o gives good results. Therefore, as the initial value for the design variables, 15^o is used. Figure 2.28 shows the resultant orientations for the in-plane normal load case study. Figure 2.29 shows the total strain energy versus design cycle graph for the optimization process.

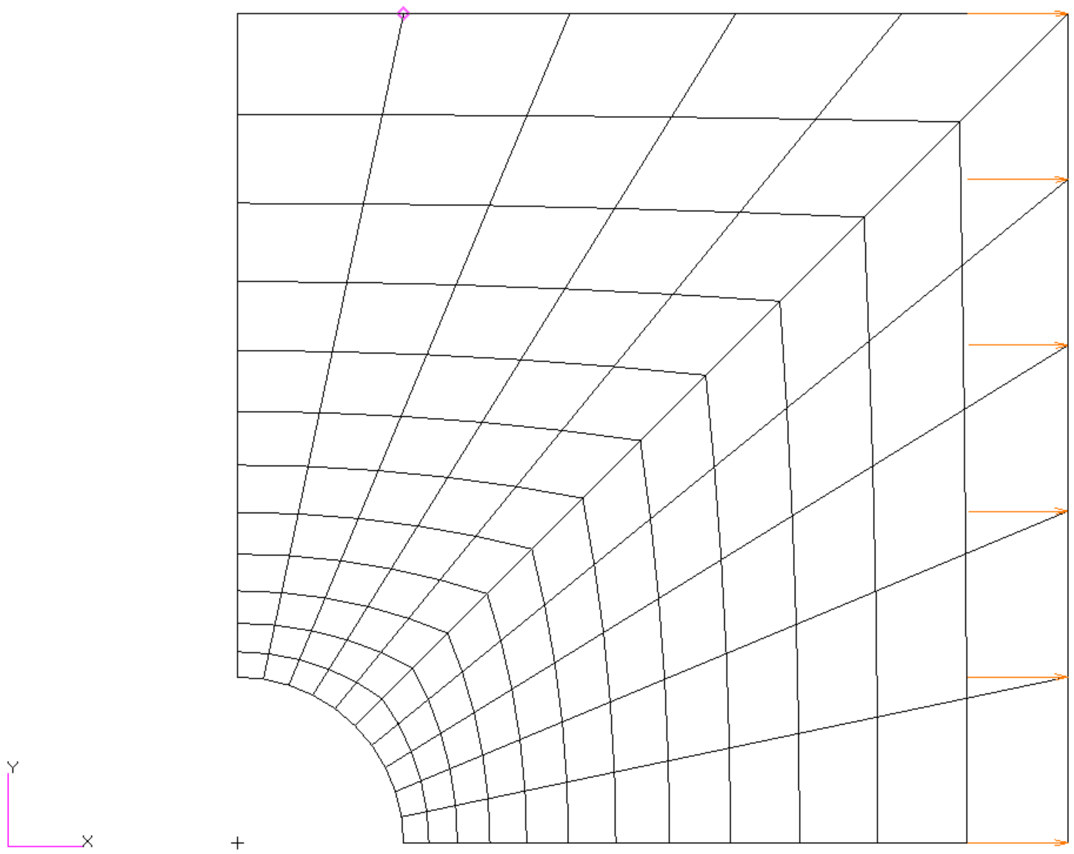


Figure 2.27: In-plane normal load case

It is noticed that the material orientations plotted in Figure 2.28 are similar to the material orientations, determined in Reference 3, shown in Figure 2.26(a). For the in-plane load case, for the upper part of the geometry the orientations are almost horizontal. For the lower part the material orientations are seen to turn clockwise. The elements that are close to the cut-out have discontinuity in material orientations. These material orientation characteristics are more or less same as the material orientation characteristics of the case study in Reference 3.

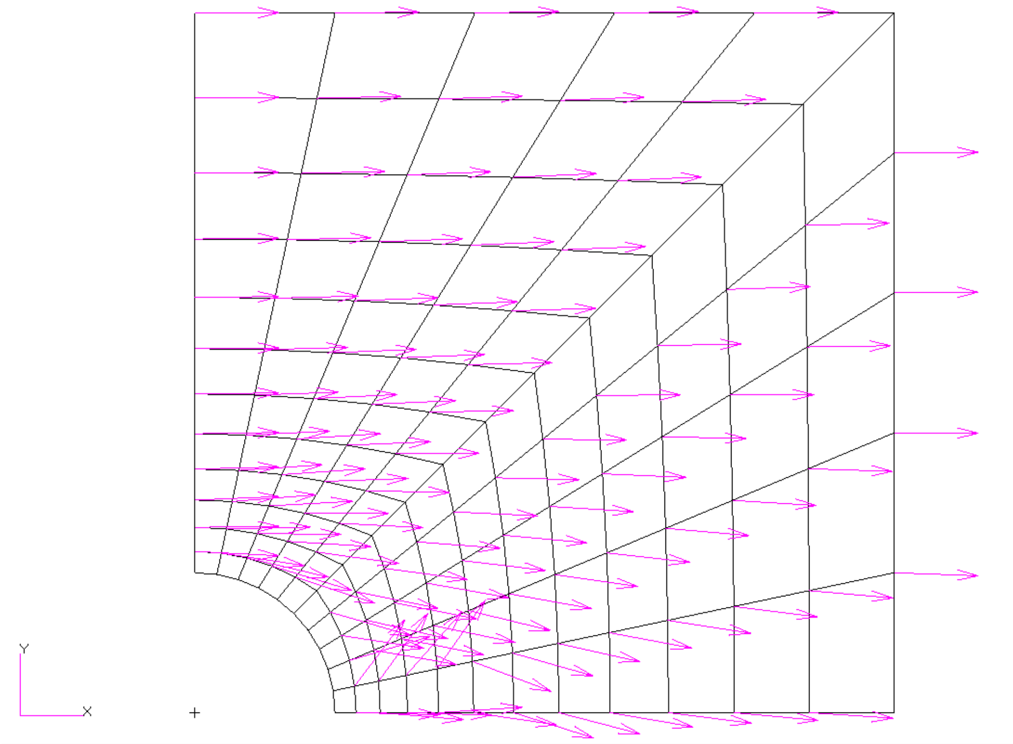


Figure 2.28: Optimized material orientations for the in-plane normal load case

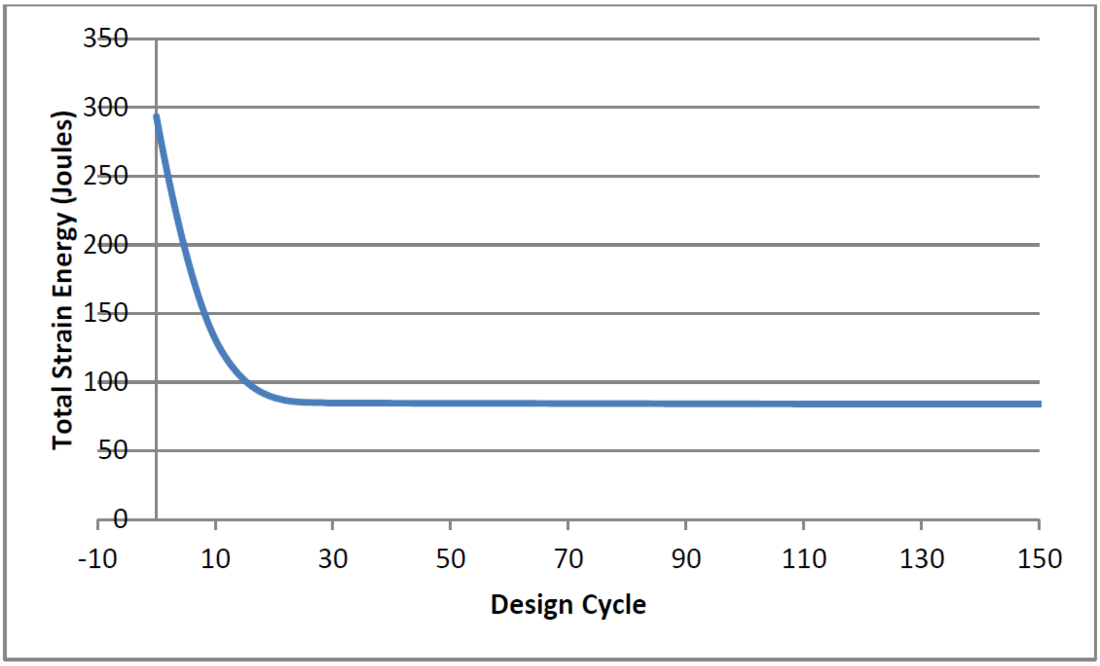


Figure 2.29: Total strain energy versus design cycle for the in-plane normal load case

It should be noted that for the present case study, actual design cycle number is approximately 500 cycles. However, since no significant change is seen in the remaining design cycles, in Figure 2.29 the objective function is drawn up to the 150^{th} design cycle. The optimum value for the total strain energy is determined as 84.08 Joules. In this particular example, optimization is performed for the 15^{o} initial material orientation case. The total run time for the particular case in Nastran is 9 minutes. By performing other optimization runs with different initial for the material orientations, the optimum value of the total strain energy can be checked whether global optimum is achieved or not. Comparison of Figure 2.25 (a) with Figure 2.28 reveals that the present solution performed by Nastran is very close to the global optimum.

2.4.2- 2D Orthotropic Square Plate Composed with a Circular Cut-Out in Center under In-Plane Shear Loading

So far, the in-plane load case studies had satisfactory results when compared to the results obtained in Reference 3. In present example, the plate with a central cut-out is subjected to pure shear loading and strain energy is minimized by optimizing the material orientations of each finite element. For the shear loading example, the displacement constraints are as shown in Figure 2.30. The lower edge of the plate is restricted to move in the *x* direction but free to move in the *y* direction. The material properties are the same as the previous case study. The shear load applied to the geometry is shown in Figure 2.31. The total shear load is 5000 N in the positive *y* direction along the right edge.

In the optimization study, the initial material orientations are again given as 15^{o}, same as the in-plane normal load case study. The optimization is done in Nastran and the optimized material orientations are shown in Figure 2.32.

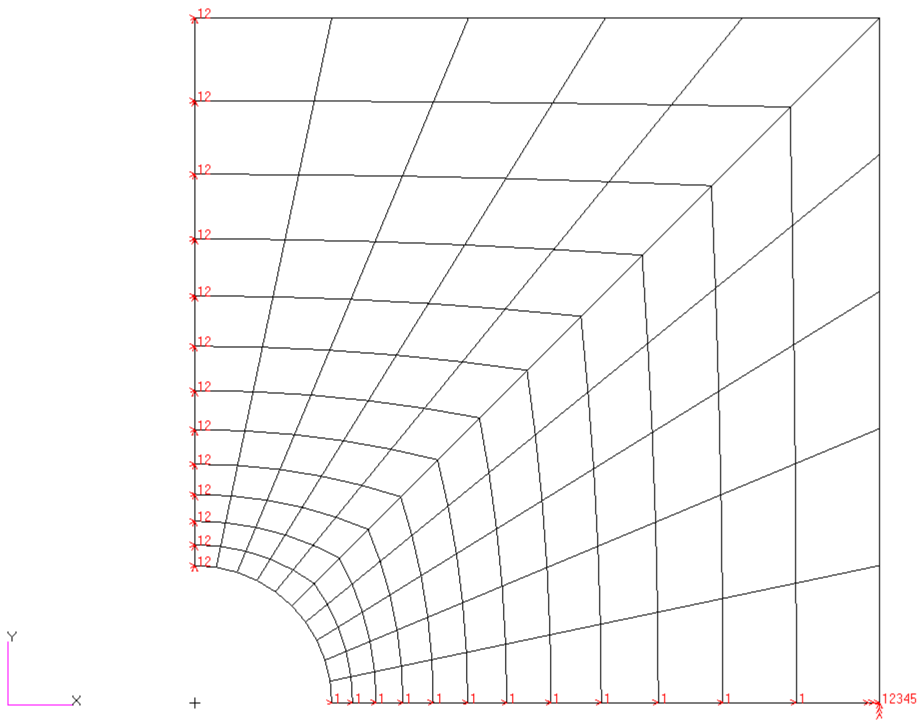


Figure 2.30: Displacement constraints for the pure shear load case study

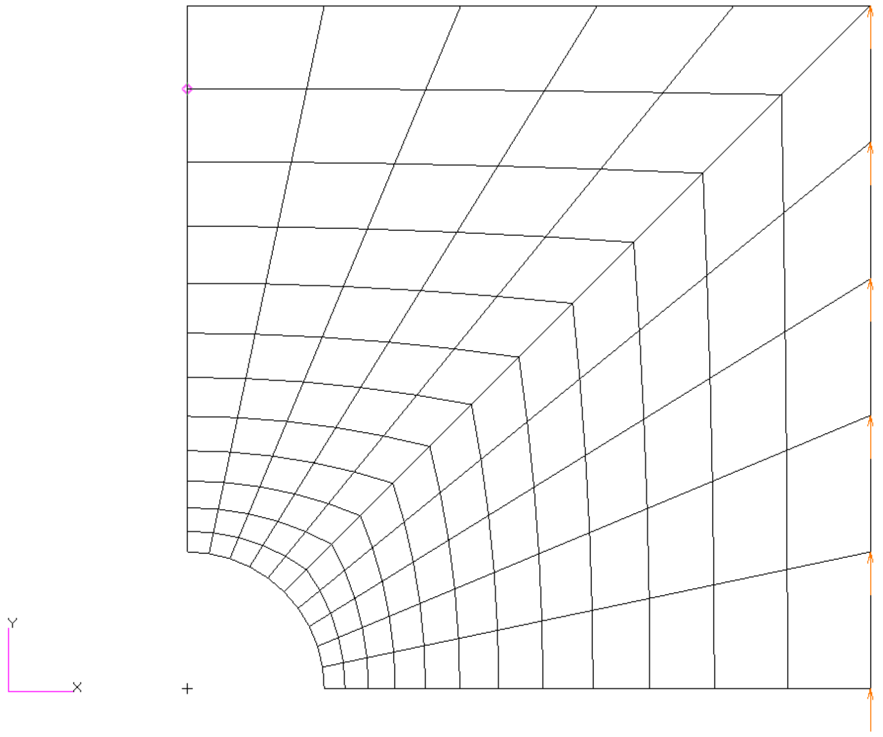


Figure 2.31: In plane shear load acting on the plate with central cut-out

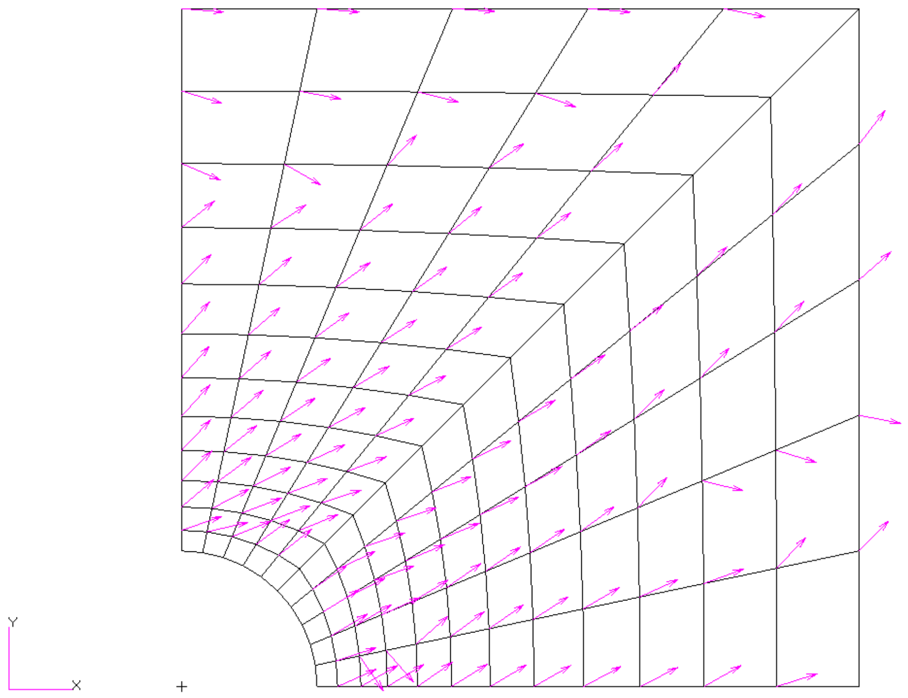


Figure 2.32: Optimized material orientations for the in-plane shear load case

It is noted that the resultant optimized orientations are not exactly same as the orientations obtained in Reference 3 for the shear load case. The material orientations of upper elements are similar in both results. The lower left part, close to the cut-out is different from the material orientations obtained in Reference 3. However, in the middle portion of the plate, material orientations aim at the right upper edge of the geometry. The aiming of the material orientations towards the upper right edge of the plate is logical, since the shear force acts vertically up along the right edge of the plate. For this example, total strain energy versus the design cycle graph is shown in Figure 2.33.

For the pure shear load case, as in the previous in-plane load normal case study, there are about 1600 design cycles in the optimization. But after the design cycle number reaches 300, not much change is observed in the objective function value, so to visualize the drop of the objective function value in the graph only first 300 design cycles are shown in the graph. The total run time for the particular case study in Nastran in about 1 hour. The optimized total strain energy value for the shear load case is determined as 252.22 Joules. This value is about 3 times higher than the optimized total strain energy for the in-plane load case. However it should be noted that in the particular example, the load is shear load and two dimensional orthotropic materials has lower in-plane shear stiffness. Therefore, it is normal that the total strain energy value is about 3 times higher for the shear load case.

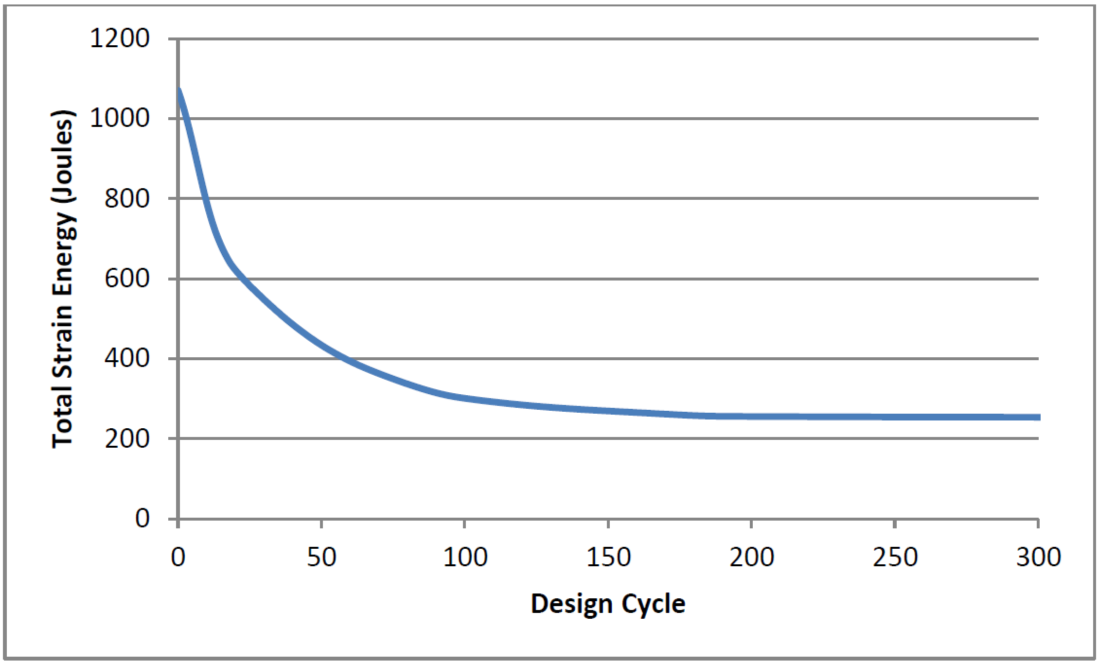


Figure 2.33: Total strain energy versus design cycle for the in-plane shear load case

With the last optimizations performed, it can be commented that in general Nastran seems to be a reliable optimization tool for the optimization of two dimensional orthotropic materials. All the case studies performed in this chapter gives satisfactory and logical results when compared with the results of the identical case studies performed in Reference 3. However as it was noted before, when there are too many design variables, gradient based optimization algorithms may stuck in local optimum. Therefore, it is recommended to repeat the solutions with different initial values of the design variables to avoid the local optimums. It is also noted that for the optimization problems involving many design variables, evolutionary algorithms such as genetic algorithms may perform better in reaching the global optimums. It should also be noted that for the case studies for the plate with central circular cut-out, in the present study the finite element mesh used is coarser than the one used in Reference 3 which is shown in Figure 2.25. Therefore, to make a more reliable comparison of the results of the present study and Reference 3, the same mesh should be used.

Chapter 3

3- Determination of Optimum Discrete Fiber Angles of Laminated Composite Structures

In this chapter, instead of 2D orthotropic materials, single layered composite structures are considered. The previous rectangular plate case in Reference 3 is solved again to show that for the layered composite plate, the optimization process works fine as the single layered two dimensional orthotropic case. In the first part of this chapter Nastran is used as the finite element solver and the optimizer. The rectangular composite plate example is first solved for a single layered composite material to check that Nastran optimization module can handle the optimization of layered composites. Then, a mid-plane symmetric composite laminate problem is solved. Differing from the previous chapter, in the second part of this chapter MATLAB genetic algorithm is used for the optimizer and Nastran as the finite element solver. These two programs are coupled with an interface MATLAB code and they are executed in a loop to perform the required optimization task. It should be noted that all the case studies in this chapter are performed for the optimization of discrete fiber angle orientations.

3.1- Optimization by the Nastran Design, Sensitivity and Optimization Module

For this case study, the geometry is again chosen as the same geometry as in Reference 3. As shown in Figure 2.10, composite laminate of aspect ratio three is taken, as the sample geometry. Differing from the previous chapter, in this example the plate is assumed to be composed of layers. Therefore, finite elements have composite laminate material properties. In this part of the chapter, only Nastran optimization module is used to calculate the optimum fiber orientation angles and in another case study thickness is also defined as the design variable. First to check the consistency of the Nastran optimization module on composite structures, a single layered composite structure is taken and the optimization problem is defined such that the objective function is taken as the total strain energy. As the second case study, a multi layered mid-plane symmetric composite structure is solved for optimum fiber orientation angles only, again taking the total strain energy as the objective function. As another case study, volume is taken as the objective function, and optimization is performed to see the thicknesses effect. In this case study, the objective function is changed because it would be meaningless if the objective function is total strain energy. Since the minimum total strain energy requirement maximizes the stiffness of the composite structure during the optimization process, the highest thickness would be determined as the optimum thickness.

In the case studies, the composite material properties are taken as the same as in Chapter 2. The layer properties are repeated here.

$$E_1 = 134\ GPa, \qquad E_2 = 16.75\ GPa, \qquad G_{12} = 5.8625\ GPa, \qquad \gamma_{12} = 0.3$$

$$t = 0.2\ m$$

3.1.1- Optimization of Fiber Orientation of a Single Layer Composite Rectangular Plate Composed of 736 Elements Subject to In-Plane Bending Load

In this part, the example in Reference 3 is studied again with a laminated composite plate model with a single layer. With this example it is aimed to see if Nastran would give a similar optimum fiber orientation distribution as that determined in Reference 3 and as the two dimensional orthotropic case studied in Chapter 2. The geometry, loads and boundary conditions are taken as same as the one in Chapter 2. In Figure 3.1, the geometry, loads and boundary conditions are shown with the finite elements. The total force is applied 1000 N.

The optimization is performed in Nastran. This case study is performed to see if Nastran works fine in the solution of the optimization problem when the element property is defined as a composite laminate instead of two dimensional orthotropic materials. In this case study, 15^{o} is chosen to be the initial fiber orientation angle for all elements.

After the optimization, it is seen that optimum fiber orientation angles are more or less similar to the ones that are obtained in this thesis and referenced in Reference 3. Figure 3.2 shows the fiber orientation distribution in the composite laminate after the optimization. It can be easily seen that the fiber orientations angles are in accordance with the results given in section 2.3, and the one in given in Reference 3. Little differences are seen in the fiber orientation angles which can only be identified when the output file is examined. Figure 3.3 shows the total strain energy versus design cycle graph for the particular case study.

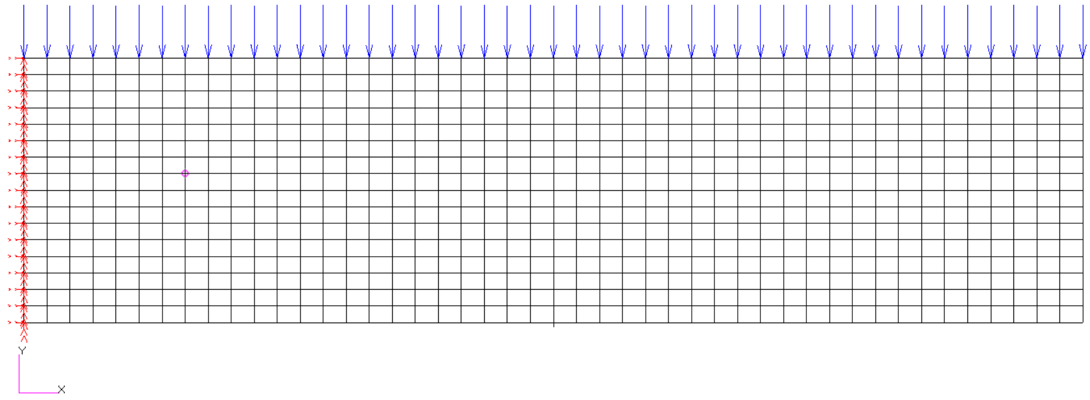


Figure 3.1: Finite element model and the loads and boundary conditions for the cantilevered plate

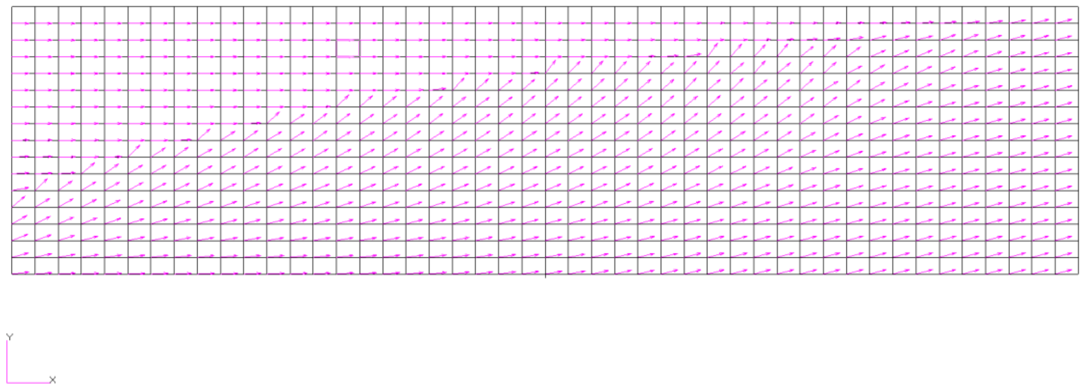


Figure 3.2: Fiber orientation distribution of the single layer composite laminate after optimization

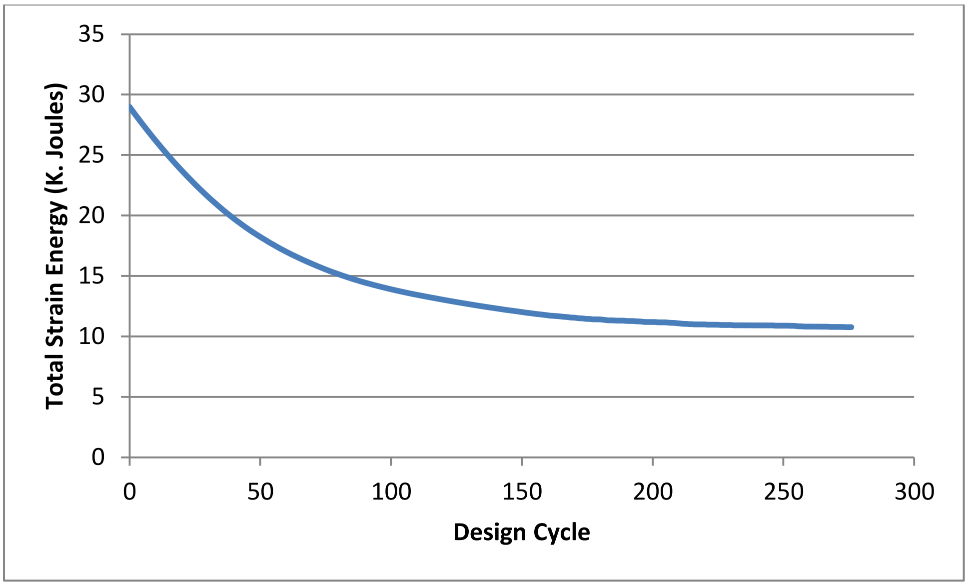


Figure 3.3: Objective function vs. design cycle graph for the initial fiber orientation angle of 15 degree

The optimum value for the total strain energy is determined as 10.8 K.Joules. This result is very close to the case study performed in Chapter 2 using two dimensional orthotropic material models. This example in a way verified that optimization of structures composed of composite laminates. The total run time of Nastran for the particular case is about 1 hour.

3.1.2- Optimization of Fiber Orientation of a Four Layer Mid-Plane Symmetric Composite Rectangular Plate Composed of 736 Elements Subject to In-Plane Bending Load

In this case study, mid-plane symmetric 4 layer composite plate is taken. This case study is performed to see how Nastran optimization module handles multi-layered composite material. Again the geometry is taken as same as in Reference 3. The loads and the boundary conditions are the same as shown in Figure 3.1. The material has the same unidirectional material properties as in the previous case study. The initial values of fiber orientations are taken as 15° for the outer layers and -15° degrees for the inner layers. Figure 3.4 shows the stacking sequence of the mid-plane symmetric composite plate which is composed of four layers. For the mid-plane symmetric laminate composed of four layers, there are twice more design variables compared to the single layer case. Therefore, in this case study there are 1472 design variables.

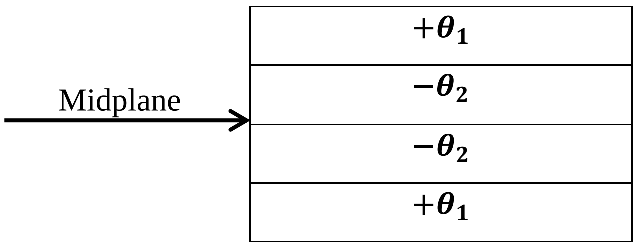


Figure 3.4: The stacking sequence of the composite plate used in the current case study.

The total thickness of the laminate is kept constant for this case study. Every layer has a thickness of 0.05 m. Figure 3.5 shows optimized fiber orientations of the outer layer of the cantilevered plate. Figure 3.6 shows the optimized fiber orientations of the inner layer of the cantilevered plate.

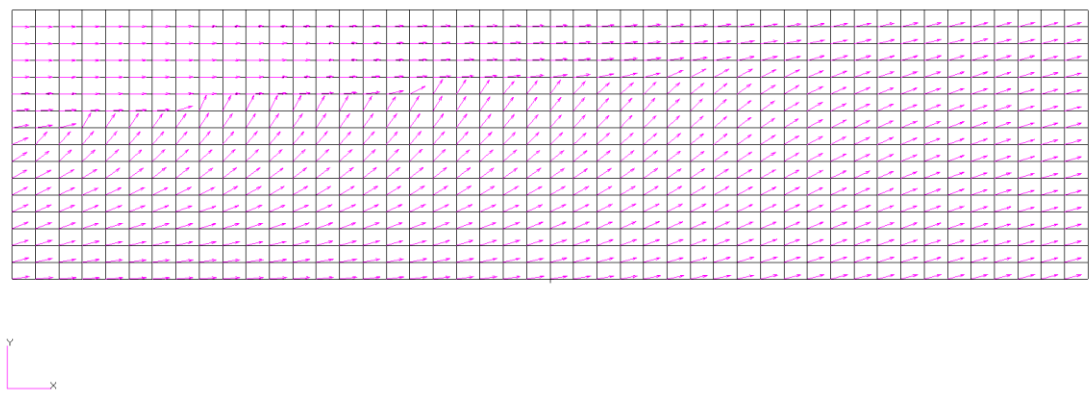


Figure 3.5: Optimized fiber orientations of the outer layer of the 4 layer mid-plane symmetric composite laminate

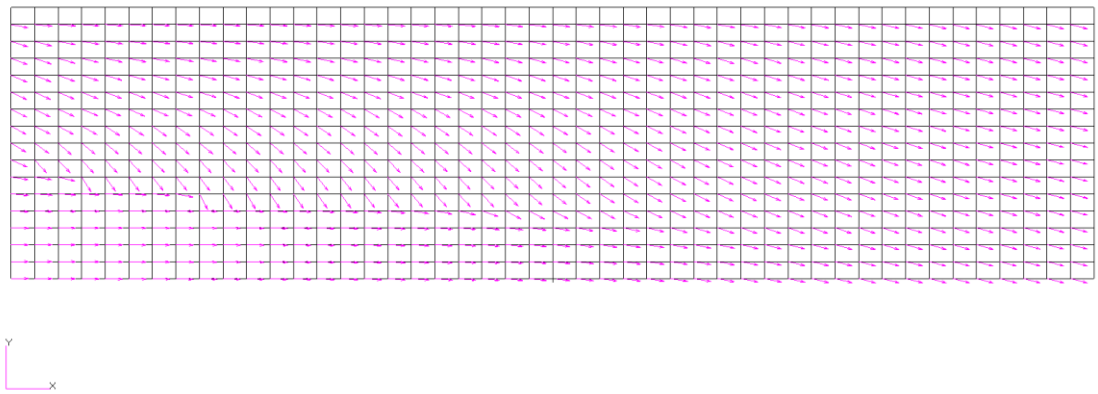


Figure 3.6: Optimized fiber orientations of the inner layer of the 4 layer mid-plane symmetric composite laminate

It is noticed that optimum fiber orientations are somewhat similar to the single layered composite case study. For instance, fiber orientations of the outer layer are very much similar to the fiber orientations of the single layer case.

Figure 3.7 show the design optimization process for the 4 layered composite plate case study. The final total strain energy is determined as 9.47 K.Joules.

From Figure 3.7, it can be seen that in each design iteration there is an improvement in the objective function which is the total strain energy. The total run time in Nastran for the particular case is about 2 hours. Based on this one can comment that the resulting fiber orientation distributions provides a global minimum total strain energy. However, as it is stated in Chapter 2, in gradient based optimization process, one can never be sure whether the final optimized configuration is global optimum solution or not. Therefore, different initial values for the design variables must be tried to seek for better solutions, if there is any.

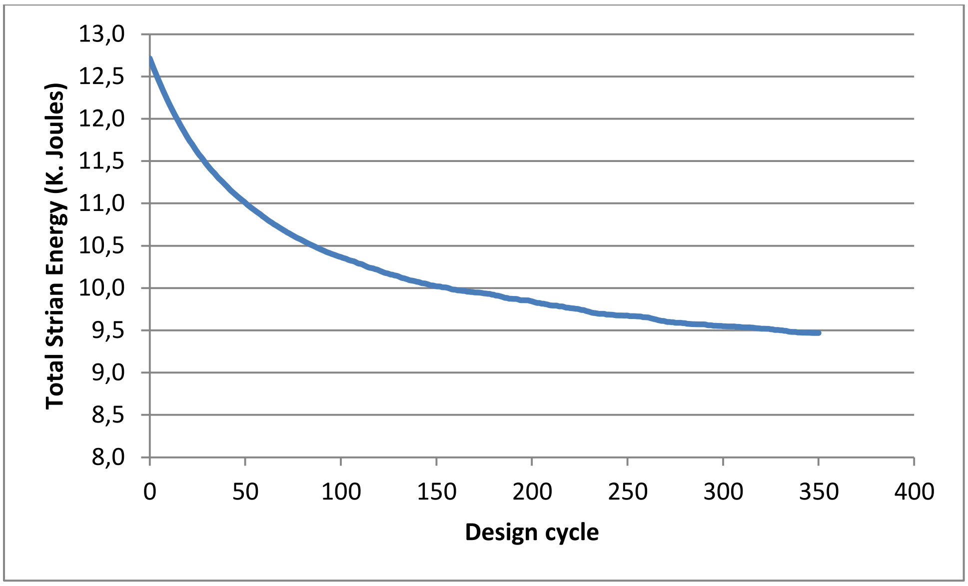


Figure 3.7: Objective function vs. design cycle graph for the mid-plane symmetric composite laminate with the initial fiber orientation of $+15^{o}/-15^{o}$

For both composite plates, the optimum total strain energy values are summarized in Table 3.1.

	Optimum Energy
Initial Orientation of 15 Degrees for the Single Layer Composite Plate	10.8 K.Joules
Initial Orientation of +15/-15 degrees for the Four Layer Composite Plate	9.47 K.Joules

Table 3.1: Optimum total strain energy values for the composite laminates

It should be noted that for the particular cantilevered mid-plane symmetric laminate the optimum total strain energy is the lowest among the total strain energies obtained for the same geometry and load cases studied in the previous sections. This example shows that with a different selection of the laminate configuration, one can obtain stiffer structure without any weight penalty.

3.1.3- Optimization of Fiber Orientation of Four Layer Mid-Plane Symmetric Composite Rectangular Plate Composed of 736 Elements Subject to In-Plane Bending Load with Layer Thicknesses and Fiber Orientations as Design Variables, and Volume as the Objective Function

For this case study, to see the thickness effect on the composite plate, the volume of the composite plate is taken as the objective function. Since, total strain energy is directly related to the stiffness of the material, if one sets the total strain energy as the objective function with the thickness as the design variable, it would be meaningless. In such an optimization problem, thickness variables would directly be set to the upper boundary, if the total strain energy is taken as the objective function. Therefore, in this example objective function is taken as the volume of the composite plate.

The geometry, load and boundary condition, material properties and the stacking sequence for this case study are all same as the previous case study. But the total number of design variables is 4 times the total number of design variables of the single layered composite plate case, because every layer has its orientation and thickness defined as design variables. Therefore, this case study has 2944 design variables. But only, minimizing the volume without a constraint would be meaningless. Therefore, in this case study a displacement constraint is added to the optimization problem. Maximum displacement of the upper right corner of the cantilevered plate is constrained to be 2.5 % of the plate length.

By taking volume as the objective function, in the optimization problem weight is indirectly minimized since volume is directly related with the weight of the plate. It should be noted that weight is one of the most important parameter in structural engineering problems. In this case study Nastran copes with the fiber orientations and the thickness of the plate at the same time.

The initial values for the thicknesses are selected as 0.05 m for each layer and 0.2 m for the total of the 4 layers. Also, initial fiber orientation angles are taken as 15^o for the upper layer and -15^o for the lower layer of the plate.

Since the volume is the objective function, first the optimized total thickness distribution of the plate is shown in Figure 3.8.

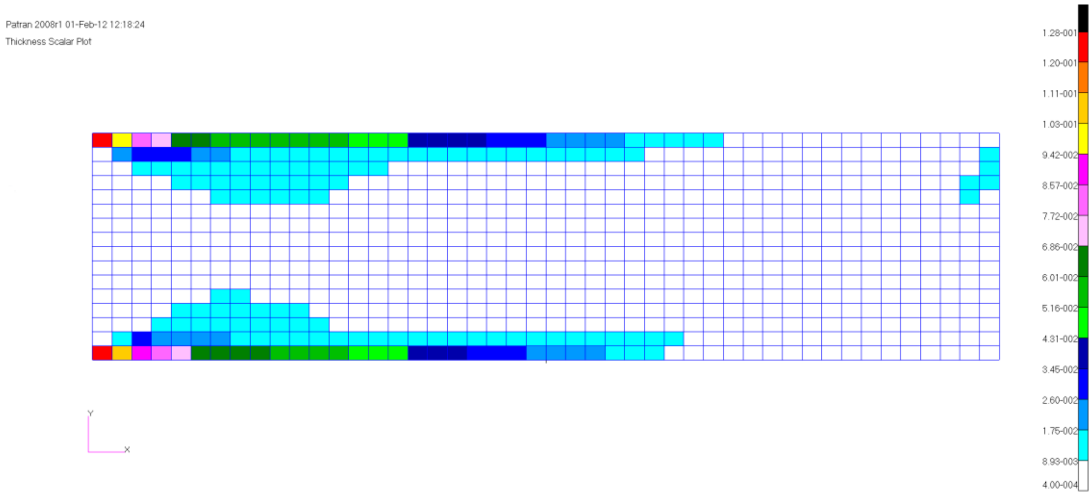


Figure 3.8: Optimized total thickness distribution of the cantilevered plate

From Figure 3.8, it is seen that the thicknesses increase from white color to red color which can be seen in the color scale on the side. Considering that the external load is in-plane bending load in this case study the thickness distribution is reasonable. Since the bending stress is highest in the upper and lower edges of the plate, thickness is higher in those areas. It is also observed that near the root of the plate, where the bending moment is higher, thickness is also higher, as expected.

Figures 3.9 and 3.10 give the optimized fiber orientation angles in the upper and lower layers, respectively.

It is noted that the fiber orientations also affect the thickness distribution, since they both are parameters affecting the stiffness.

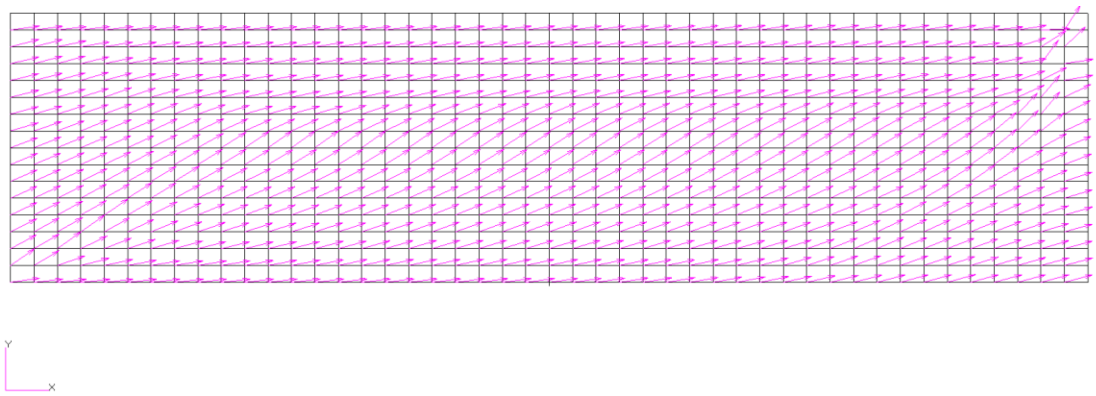


Figure 3.9: Optimized upper layer orientations of the plate

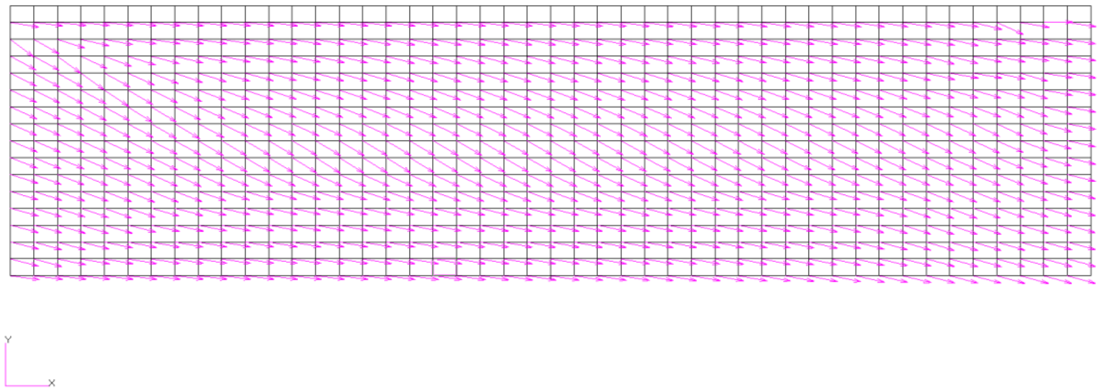


Figure 3.10: Optimized lower layer orientations of the plate

The fiber orientations for both upper and lower layers show a continuous pattern for the solution except the discontinuity at the upper right corner of both layers. The discontinuity can be due to the displacement constraint defined in the optimization problem.

The total plate volume versus design cycle graph is shown in Figure 3.11. With the given conditions and the constraints the objective function value has decreased from

0.8 m^3 to 0.037 m^3. The weight of the composite plate decreases with decreasing volume of the plate.

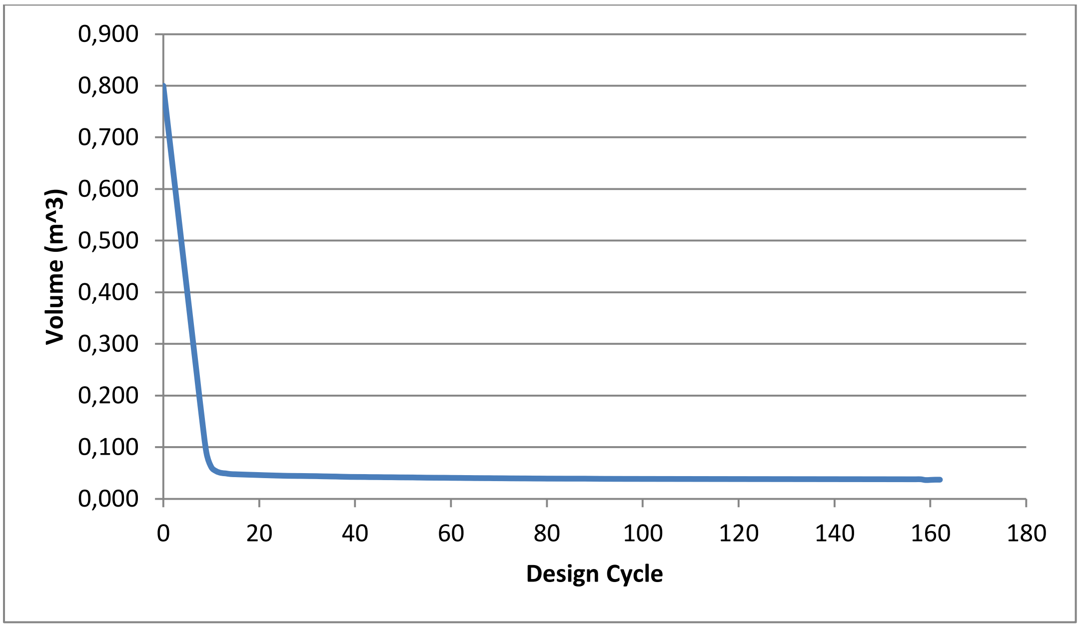


Figure 3.11: Total volume versus design cycle

From Figure 3.11, it is seen that total volume of the composite plate rapidly decreases with the design cycles. The reason that the optimization lasts about 160 iterations is because of the convergence criteria that are set in Nastran optimization module. The default values are changed to have the optimum solution at once. It should be noted that the present case study is performed without any stress constraint and only displacement constraint is used. This case study is performed to demonstrate that both thickness and fiber orientation angle of each element can be defined as design variables, and optimization can be conducted accordingly. Adding stress constraints brings no additional complexity to the optimization problem. Therefore, the total run time of Nastran for the particular case study is about 20 minutes, and the tip deflection constraint might help Nastran to converge very fast to the optimum solution.

With this example, it is again shown that Nastran optimization module works fine and fast even with large number of design variables. However, since Nastran optimization module works with gradient based optimization algorithm, the global optimum can be missed. Therefore, in the proceeding sections, optimization case studies are performed by employing MATLAB genetic algorithm as the optimizer.

3.2- Optimization by Coupling the Nastran Finite Element Solver and Genetic Algorithm Optimizer in MATLAB

So far, in the previous case studies, only Nastran is used for both for the optimizer and the finite element solver. Since, the optimization of the composite material orientations is a complex problem involving many design variables and the optimizer in Nastran is gradient based optimizer, the optimization algorithm of Nastran can be stuck in local minimums easily. Every engineer wants the global solution for the optimization problems. For this problem too, the global optimum is desired. For problems involving many design variables, evolutionary algorithms such as genetic algorithm works better in reaching the global optimum. Optimization process by the genetic algorithm is explained in Chapter 1. In this thesis, MATLAB Genetic Algorithm Optimizer is used for the case studies presented in this chapter. The objective function is given as input to the algorithm as a MATLAB code. The MATLAB code is prepared to couple the Nastran finite element solver and the MATLAB genetic algorithm optimizer. The MATLAB code gets the population that the genetic algorithm creates. In discrete orientation cases, MATLAB creates the population as “beta” variables within the given boundary conditions. The “beta” variables are the design variables for the particular case studies of discrete fiber orientation case studies. They are created to reduce the design space for the problems. The created “beta” values are multiplied with the numbers written in the Nastran input file. These “beta” values are shown in the optimization process figures. The “beta” values are in the lower part of the figures. After that, the calculated initial population values are written in the Nastran input file. After this step, MATLAB code calls for a Nastran execution. The input file of Nastran, which is prepared, is used in the Nastran run to produce the output file which has all the results. For the case studies in this chapter linear static analysis is performed for the composite laminates. After the Nastran execution, MATLAB code then extracts the objective function values from the output file of Nastran one by one at each function evaluation, and genetic algorithm ranks the output values and passes to the next iteration step for the problem. The iterations continue until the optimization criterion is met.

It should be noted that in the optimization process by genetic algorithm, the population size must be carefully selected with respect to the number of design variables. If the population size is small, then the genetic algorithm is going to act like a gradient based solution algorithm. The solution would probably be stuck around a local minimum and all the individuals in the population would converge to the local minimum. The population must be big enough to reach the global minimum and small enough to perform the optimization in a reasonable time.

Coupling of the Nastran finite element solver and MATLAB genetic algorithm toolbox in the optimization process is described in the flowchart given in Figure 3.12. The detailed explanation of the MATLAB codes used in the optimization procedure is given in the Appendix B.

The following case studies are performed according to the flowchart in Figure 3.12. In these case studies, the material orientation angles are discrete angles, same as the previous studies performed by the optimization module of Nastran (SOL 200).

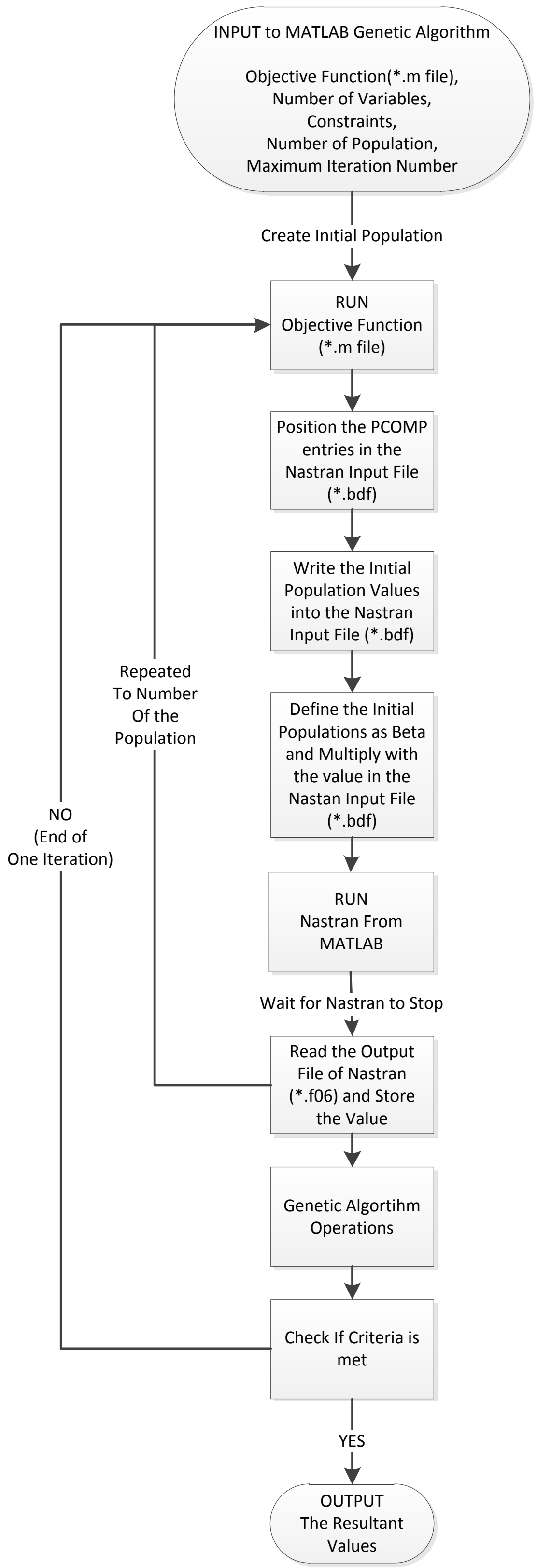


Figure 3.12: Simplified steps showing the coupling of the Nastran solver and the MATLAB genetic algorithm toolbox optimizer in the optimization process

3.2.1- Optimization of a Rectangular Single Layer Composite Plate under Transverse In-Plane Loading

In this section, the case study in section 3.1.1 is repeated. Material and geometric properties, which are used in this section, are same as the previous 2D orthotropic and composite plate case studies. The thickness of the plate is also taken as 0.2 m, same as before.

The load and displacement boundary conditions are also same as defined in section 3.1.1. Composite plate has one layer and all the elements have discrete fiber orientation angles which are also defined as design variables.

All the design characteristics are same as in section 3.1.1, but in this case study MATLAB Genetic Algorithm is used as the optimizer and Nastran is used as the finite element solver. Again as before, the total strain energy is the objective function for this case study.

The population number for the single layered composite plate case study is taken as 100. The reason for selecting this population number is that in the particular example, there are 736 design variables which are defined as the fiber orientation angles of each element. For genetic algorithm problems, it is highly recommended to use a higher population size. This population number should at least be equal to the number of design variables used. For the particular case the population number is smaller than the total number of design variables, because otherwise the optimization process would be extremely long. But, these drawbacks can be compensated with creating an approximate model for the finite element. With the approximate model, there would be no need for a new finite element analysis in each iteration.

The optimization process for the single layer composite plate is given in Figure 3.13.

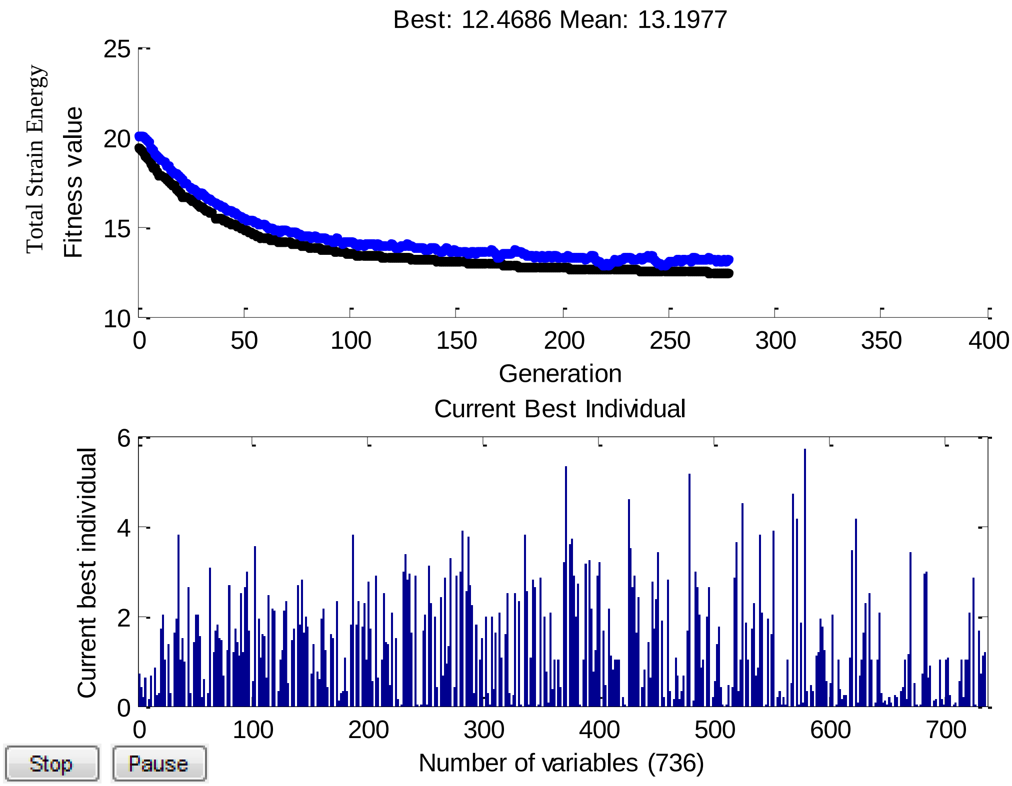


Figure 3.13: Optimization process of the discrete fiber angle orientations of the single layer composite laminate

The upper graph in Figure 3.13 shows the best objective function value for each design cycle with the black dots and the mean objective function value of whole population for each design cycle with the blue dots. The below, bar chart shows best individual values for the current design cycle. The values are between 0 and 5 in the current design cycle. These beta values created by genetic algorithm are multiplied by the orientation value specified in the Nastran input file. These orientation values are defined in composite property entries for the structure. The value in the Nastran input file is 10 for the particular case study. For example, if a variable has a value of 3.5, this is written to the Nastran input file as 35^o for the fiber orientation.

The optimization is terminated, because the run time is very long and the population number seems to be inadequate. The orientations for the best individuals cannot be shown, because the run must stop to have those values. However, based on the trend seen in Figure 3.13, it can be said that the algorithm works fine for the optimization problem. Up to the 275^{th} design cycle the best total strain energy value is obtained as 12.47 K.Joules and it is still decreasing. The optimized value for the same case study in section 3.1.1 is 10.8 K.Joules. It should be noted that with a population size of 100, the finite element analysis and modifying the Nastran input files take significant time. Therefore, optimization run lasted about two weeks and a half. Therefore the optimization is terminated because of the long optimization time. The performance of the computer is also affecting the run time for the problem. This drawback can be compensated by parallel computing and faster CPUs.

It is decided to check the MATLAB genetic algorithm with a coarser mesh using the same geometry and load condition as the particular case study. This study is presented in the next section.

3.2.2- Optimization of a Rectangular Single Layer Composite Plate under Transverse In-Plane Loading with a Coarser Mesh

The same material properties, given in the previous case study, are also used in this case study. The load and displacement boundary conditions are the same as the previous case study as well. The mesh is different in the current case study. The mesh is shown in Figure 3.14 with the load and displacement boundary conditions. There are 12 finite elements for the current case study. On the other hand, the composite plate has a single layer, and the optimization problem has 12 independent design variables.

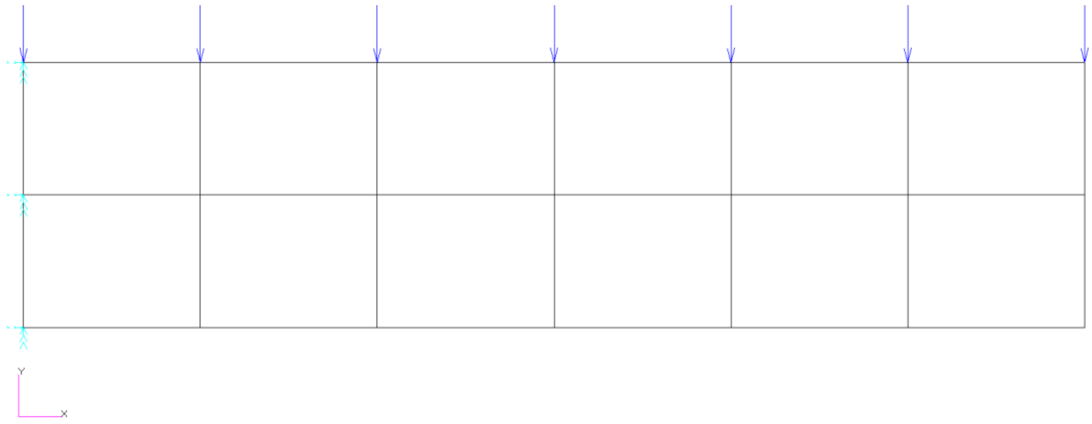


Figure 3.14: Mesh of the current case study with 12 finite elements

The problem considered in this section is not solved before. To compare Nastran optimization module and MATLAB genetic algorithm, first a series of Nastran optimizations are performed with different initial conditions because the gradient based algorithms are dependent on the initial guesses. Table 3.2 shows the optimization results for the Nastran optimization. It should be noted that zero degree initial guess cannot be performed because of the specifications of Nastran.

Initial Guess (degrees)	Minimum Total Strain Energy (K. Joules)	# of iterations
-1	6,318787	196
1	6,291005	263
5	6,292220	174
10	6,295078	174
15	6,296534	212
20	6,292673	249
25	4,616500	1107
30	4,606064	1035
35	4,678367	532
40	5,230337	526

Table 3.2: Optimum total strain energy values for different initial guesses in Nastran Optimization module

It is seen from Table 3.2 that 30 degree initial guess for the optimum solution among the other solutions performed with different initial guesses. After trying initial guesses in the range of -1^o and 40^o, it is observed that the total strain energy values increase with no further improvement in the objective function. Therefore, no further trials are performed. Figure 3.15 shows the optimum orientations for the 30 degree initial guess case study.

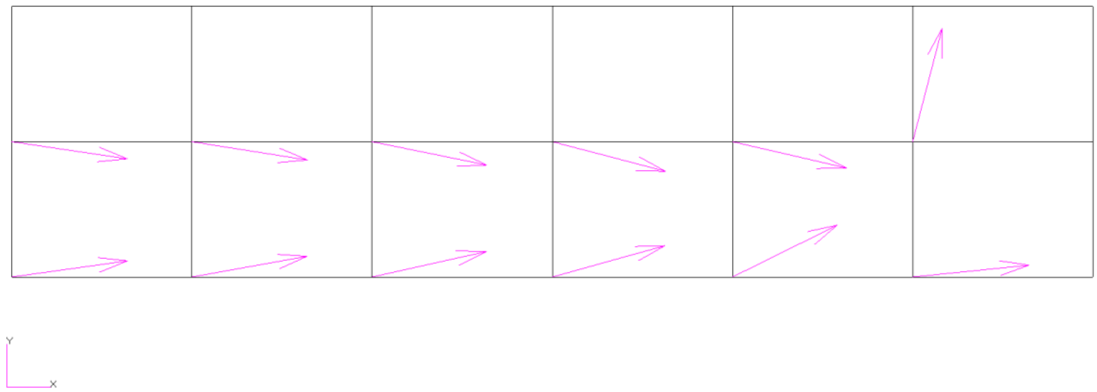


Figure 3.15: Optimum orientations for the 30 degree initial guess case study

After performing the optimizations in Nastran optimization module, the same problem is solved by employing MATLAB genetic algorithm as the optimizer. As mentioned earlier, the MATLAB code generated, multiplies the property orientation with the design variables defined for MATLAB genetic algorithm. In the Nastran input file the material property orientation is given as 30 degrees such that the lower and upper boundaries for the design variables are assigned as -3 and 3. Thus, the design space covers the orientation angles of -90^o to 90^o. The population size selected for the current case study is 60. After 3 runs for the MATLAB genetic algorithm the lowest resultant optimum value for the current case study is determined as 4.6 K. Joules. The optimization process for this case study is shown in Figure 3.16. The optimized fiber orientations for the current case is also shown in Figure 3.17.

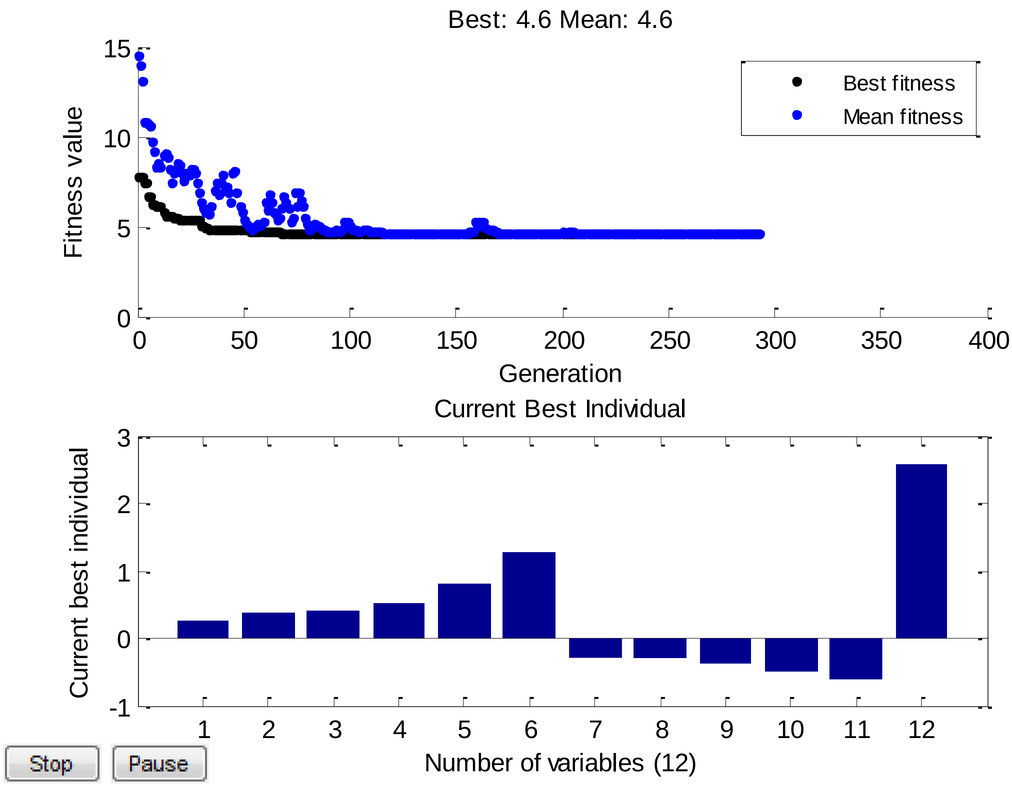


Figure 3.16: Optimization process for the 12 element composite plate

The optimum orientations for the elements is shown in Figure 3.17.

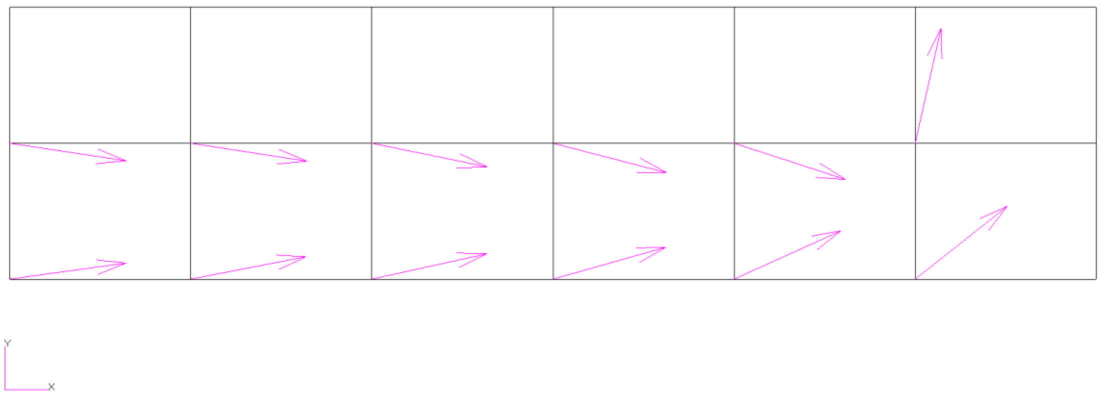


Figure 3.17: Optimum orientations of the fibers for MATLAB genetic algorithm

It is seen that, the optimum value of the total strain energy obtained from MATLAB genetic algorithm is slightly lower than the optimum total strain energy determined by employing Nastran optimization module in the optimization process. The final orientations show slight differences. It should be noted that in the particular study, since there are only 12 design variables, gradient based optimization algorithm of the optimization modlu of Nastran may also find the global optimum. However, this sample problem showed that even with 12 design variables, in order to reach the global optimum many trials, with different initial values for the design variables, had to be performed when the gradient based optimization solver of Nastran is employed in the solution of the optimization problem. Even then, genetic algorithm based optimization performed by MATLAB finds a slightly better solution.

Chapter 4

4- Determination of Optimum Continuous Fiber Paths of Variable Stiffness Composite Structures

In theory, discrete fiber orientation optimization gives the optimum orientation for the composite structures. However, in practical applications manufacturing of composite structures based on discrete fiber orientation angles, determined in an optimization framework, for the composite structures is not easy. There would be discontinuities in the fiber orientation angles in the composite structure, and this would make it almost impossible to manufacture the composite structure. If the design of the laminate is done according to the manufacturing constraints at the very beginning of the design process, then it would be possible to manufacture the composite structure.

In this thesis the final goal is to optimize continuous fiber paths such that composite laminate can be manufactured. Differing from the other case studies, the case studies in this chapter have continuous fiber paths. As mentioned in Chapter 1, variable stiffness concept is used to determine the fiber paths along the composite plate [2] [4] [28].

4.1- Description of Continuous Fiber Path Optimization

In order to create variable-stiffness composite laminate, a reference fiber path method, which is commonly employed in the literature, must be defined first [2; 4; 28; 29]. The reference path serves as the basis for creating other fiber paths in a ply. In the literature, two common methods have been proposed to create other fiber paths utilizing the reference fiber path. One technique used is to produce the remaining fiber paths by shifting the reference path along one of the axis of the composite laminate. In the other method, the remaining fiber paths are created such that all the fiber paths are truly parallel to the reference fiber path. The continuous fiber paths are defined for square or rectangular shapes.

Variable stiffness feature of a ply is achieved by curvilinear fiber paths which cannot be described by a single orientation angle. In the literature, a convenient continuous fiber path is described such that it is assumed that the fiber orientation of the reference fiber path varies linearly from one value at the center of the panel to another at a specified distance [2].

The advantage of the linearly varying fiber orientation angle is that, the fiber orientation of the composite lamina can be determined with just three variables. These variables are defined as $T_0, T_1, \emptyset$ [2; 28]. T_0 is the fiber orientation angle of the reference fiber path measured at the center of the plate, T_1 is the fiber orientation angle measured at $x' = a/2$, where a is the reference dimension of the composite laminate and x' and y' are the rotated axes of the plate shown in Figure 4.1. Finally, as shown in Figure 4.1, $\emptyset$ is the rotation angle of the tow placement axis [28]. For a composite lamina, the representation of this concept is denoted by "$\emptyset\langle T_0|T_1\rangle$". A symmetric composite laminate with a zero degree of rotation angle $\emptyset$ is represented as $[\pm\langle T_0|T_1\rangle]_s$ [28]. A "$\pm$" sign in front of the rotation angle $\emptyset$ represents that reference fiber paths for two successive layers are rotated by equal and opposite amount such that the fiber orientation angle of the outer and inner lamina have positive and negative signs, and “s” represents that the composite laminate is mid-plane symmetric. Figure 4.2 shows the effect of the rotation angle ∅ on the composite laminate.

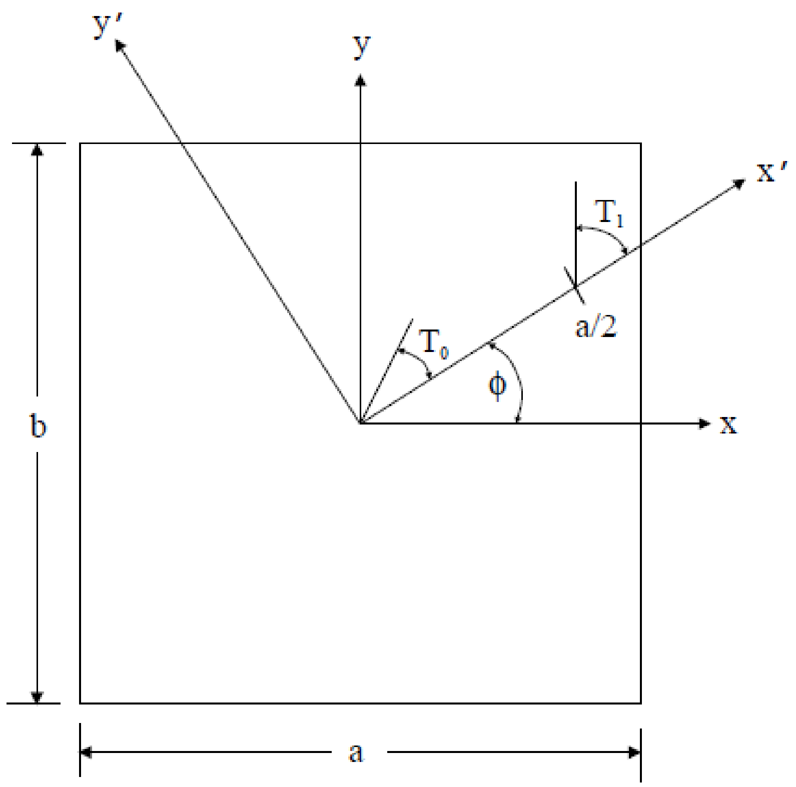
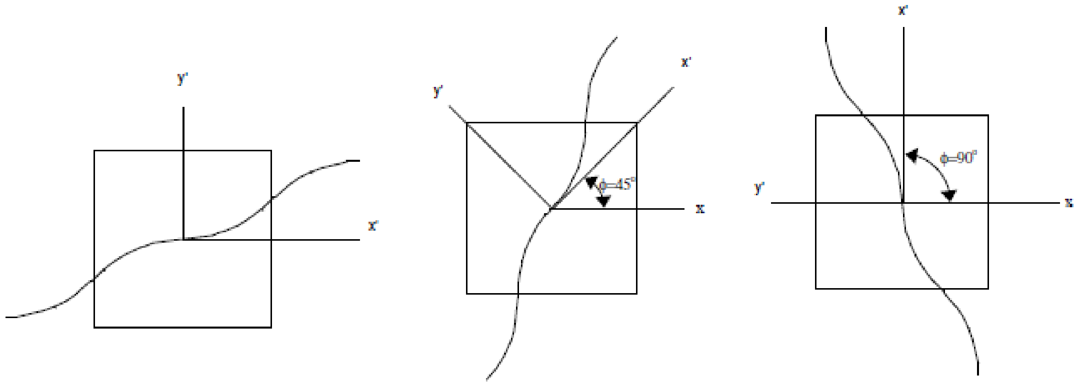


Figure 4.1: The variables in the variable stiffness composite laminate



(1) (2) (3)

Figure 4.2: The effect of the rotation angle $\emptyset$, in case (1) $\emptyset = 0°$, in case (2) $\emptyset = 45°$, in case (3) $\emptyset = 90°$ for the $\langle 0°|45° \rangle$ reference fiber path, taken from Reference [4].

Equations 4.1, 4.2, 4.3 define the continuous reference fiber path written in terms of the variables "$\emptyset\langle T_0|T_1\rangle$". Equations are written such that fiber orientation angle given by the slope of the reference fiber path varies with respect to the x' axis of the composite laminate. Equations 4.1 and 4.2 are written for the rotated path axis [4; 29].

$$
y'(x') = \begin{cases}
\dfrac{a}{2(T_0 - T_1)}\left\{\ln[\cos T_0] - 2\ln[\cos T_1] + \ln\left[\cos\left(-T_0 + 2T_1 + \dfrac{2(T_1 + T_0)}{a}x'\right)\right]\right\}, \\
\qquad for \quad -a \le x' \le -\dfrac{a}{2} \\
\dfrac{a}{2(T_1 - T_0)}\left\{-\ln[\cos T_1] + \ln\left[\cos\left(T_1 + \dfrac{2(T_1 - T_0)}{a}x'\right)\right]\right\}, \\
\qquad for \quad -\dfrac{a}{2} \le x' \le 0 \\
\dfrac{a}{2(T_0 - T_1)}\left\{-\ln[\cos T_0] + \ln\left[\cos\left(T_0 + \dfrac{2(T_1 - T_0)}{a}x'\right)\right]\right\}, \\
\qquad for \quad 0 \le x' \le \dfrac{a}{2} \\
\dfrac{a}{2(T_1 - T_0)}\left\{\ln[\cos T_0] - 2\ln[\cos T_1] + \ln\left[\cos\left(-T_0 + 2T_1 + \dfrac{2(T_0 + T_1)}{a}x'\right)\right]\right\}, \\
\qquad for \quad \dfrac{a}{2} \le x' \le a
\end{cases} \tag{4.1}
$$

$$\theta(x') = \begin{cases} \emptyset + \frac{2}{a}(T_1 - T_0)x' + T_0 - 2(T_0 - T_1), & for \ -a \leq x' \leq -\frac{a}{2} \\ \emptyset + \frac{2}{a}(T_0 - T_1)x' + T_0, & for \ -\frac{a}{2} \leq x' \leq 0 \\ \emptyset + \frac{2}{a}(T_1 - T_0)x' + T_0, & for \ 0 \leq x' \leq \frac{a}{2} \\ \emptyset + \frac{2}{a}(T_0 - T_1)x' + T_0 - 2(T_0 - T_1), & for \ \frac{a}{2} \leq x' \leq a \end{cases} \quad (4.2)$$

where;

$$x' = x \cos \emptyset + y \sin \emptyset \quad (4.3)$$

It is noted that for the case of $\emptyset = 0^{\circ}$, which is used in this thesis, x' and x axes coincide. In this thesis, rotation angle at the center of the laminate is taken as zero, because otherwise, the calculations for the fiber paths become more complicated and time consuming. Therefore, for the ease of the calculations and the speed of the analysis rotation angle is set to zero. However as future work, it is recommended that non-zero rotation angle cases be studied also. The θ and $\emptyset$ angles are visualized in Figure 4.3 for a $45\langle 0|45 \rangle$ reference fiber path. In figure 4.3, β is the actual fiber angle for the given points in the rotated coordinate frame.

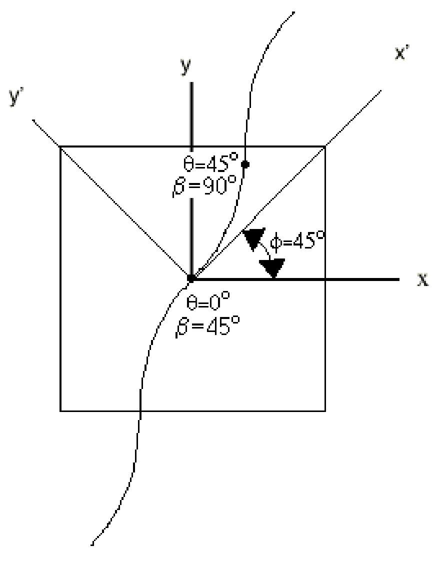


Figure 4.3: The visualization and the fiber angles of the rotated reference fiber path for $45\langle 0|45 \rangle$ orientation, taken from Reference [28].

The final goal for this thesis is to design composite plates by employing continuous fiber paths such that the composite plate can also be manufactured. Therefore, in the definition of the optimization problem appropriate manufacturing constraints must also be taken into consideration. For this purpose, y' equations given by Eqns. 4.1 and 4.3 are used to define curvature constraint. A general curvature constraint for the tow placement machines is taken from the References 2 and 4. The maximum allowable curvature for the tow placement machine that is used in References 2 and 4 is 1/(12 inches). In SI units the maximum allowable curvature is 1/ (30.48 cm). According to the curvature value the minimum radius of curvature is 30.38 cm. For a function of single variable, the curvature equation is given by

$$K = \frac{f''(x)}{(1+\left(f'(x)\right)^2)^{3/2}} \qquad (4.4)$$

For the continuous fiber path orientation case studies, function $f(x)$ is the position equation, $y'(x')$. The curvatures must be calculated at all points on the composite laminate. Since, the reference fiber path is anti-symmetric with respect to the y axis, only the curvature of this path in the positive x portion of the composite laminate needs to be calculated.

In Chapter 1, two production methods were considered. The so called, shifted and parallel fiber paths are two common ways to define the remaining fiber paths based on the reference fiber path definition. In this thesis, shifted fiber path definition is used in the continuous fiber path optimization study. Shifted fiber path can simply be defined as the shifting of the reference fiber path. The remaining fiber paths are created by shifting the reference fiber path by a fixed amount in the y' direction. Figure 4.4 shows a single shifted path which is created by shifting the reference fiber path in the y' direction.

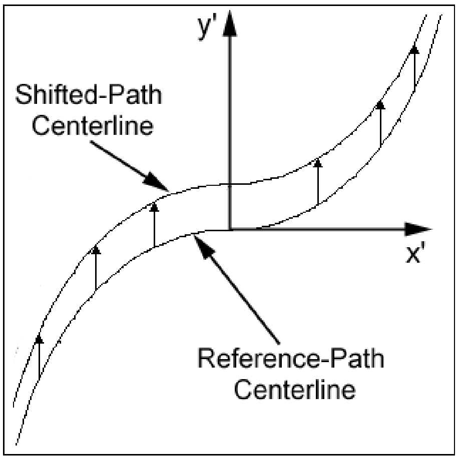


Figure 4.4: Shifted fiber path according to the reference path, taken from Ref. [28]

From Figure 4.5, it is seen that the reference fiber path is shifted along the y' axis to create the other paths. This is how the composite laminate is manufactured. The completed shifted fiber lamina $0°\langle 0°|45°\rangle$ is shown Figure 4.5.

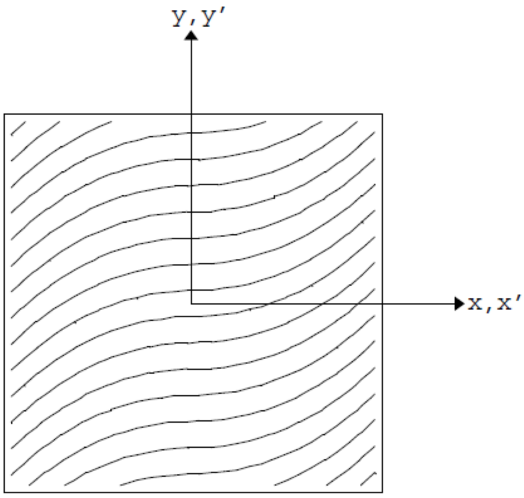


Figure 4.5: Shifted paths for a laminate with the reference fiber path definition $0°\langle 0°|45°\rangle$, taken from Ref. [4]

In this thesis, as shown in Figure 4.5 the center rotation $\emptyset$ is kept zero, for the optimization case studies performed in this chapter. Thus, for the shifted fiber paths fiber orientations only change along the x' axis. Since the reference fiber path is shifted to create the other fiber paths, it is also sufficient to calculate the curvature of the reference fiber paths to check for the curvature constraint during the path optimization. Curvatures of the remaining fiber paths are same as the curvatures of the reference fiber path at the same x' location and for any y' coordinate.

The other fiber path definition is the parallel fiber path method. In this method fiber paths are generated such that, each path is defined as a set of points lying at a constant distance from the reference fiber path. Therefore, parallel fiber paths do not have an analytical expression [4]. Every path is defined through a numerical procedure. Figure 4.6 shows the $0°\langle 0°|45°\rangle$ lamina created with parallel fiber path definition.

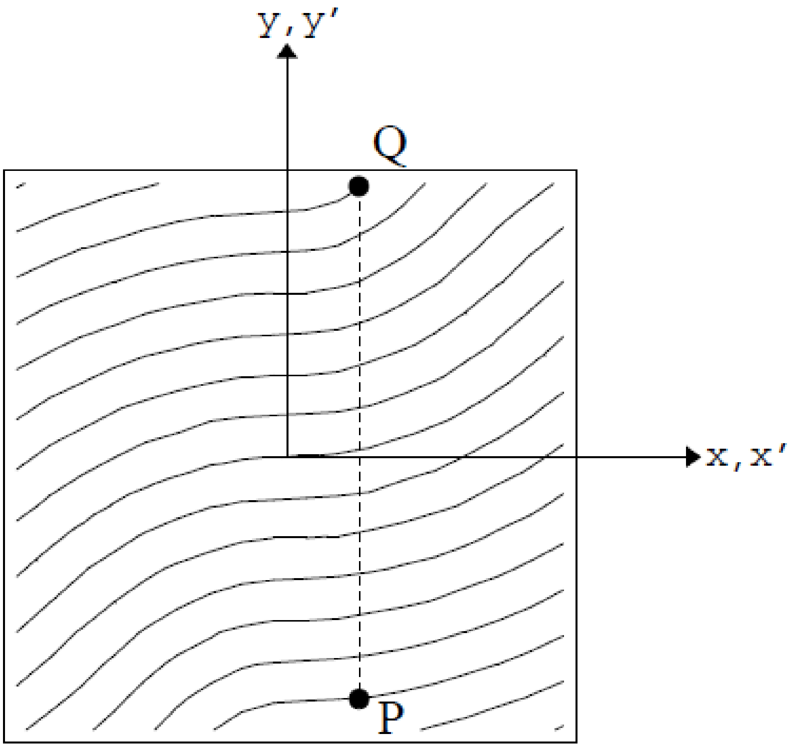


Figure 4.6 : Parallel paths for a laminate with the reference fiber path definition $0°\langle 0°|45°\rangle$, taken from Ref. [4]

If parameters T_0 and T_1 are close to each other, parallel and shifted fiber paths look alike. Figure 4.7 shows a lamina with the reference fiber path definition $0°\langle 0°|75°\rangle$.

It is noticed that when the parameters T_0 and T_1 are not close to each other, the differences in the fiber paths for the shifted and parallel path cases can be seen more clearly as depicted in Figure 4.7.

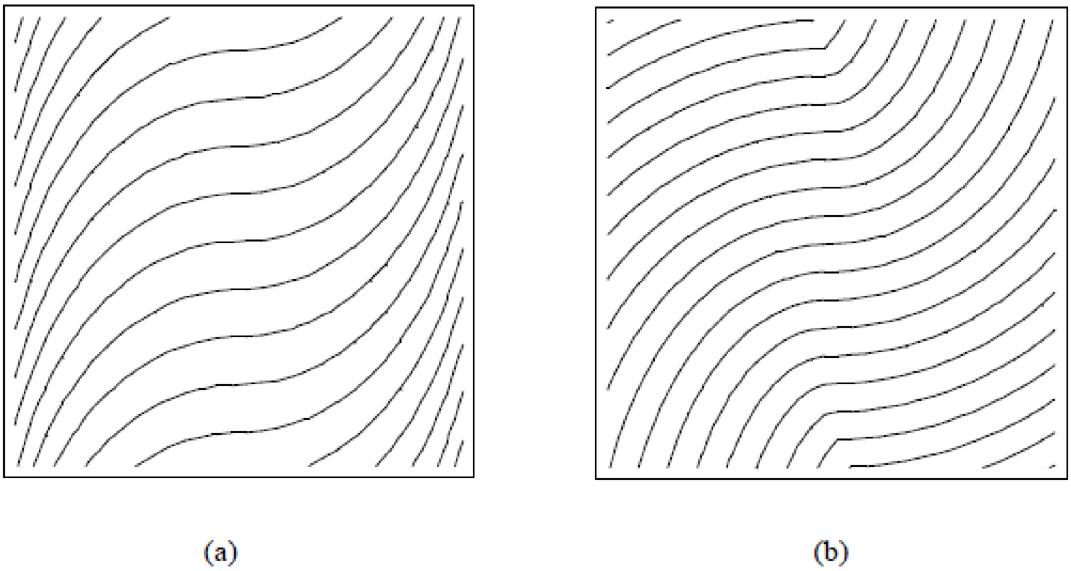


Figure 4.7: Laminate created with (a) shifted fiber path method, (b) parallel fiber path method with the reference fiber path definition $0°\langle 0°|75°\rangle$, taken from Ref. [4]

As mentioned earlier in this chapter shifted fiber path definition is used in the case studies which are presented in this chapter. Because shifted fiber paths can be defined with analytical expressions with respect to the reference fiber path. On the other hand parallel fiber path definition requires numerical procedures to define the remaining parallel fiber paths with respect to the reference fiber path.

4.2- Solution Procedure Used for Continuous Fiber Path Optimization

As described in section 2 of Chapter 3, MATLAB genetic algorithm toolbox and Nastran finite element solver are coupled for the continuous fiber path optimization. In this part of the thesis the continuous fiber path and curvature constraint definitions are added to the optimization procedure. Addition of the fiber orientation definitions and curvature constraints affect the optimization procedure which was used in Chapter 3 on the optimization of discrete fiber orientation angles. Since in the shifted fiber path method, fiber orientation angle is defined by means of the analytical expressions given by equations 4.2 and 4.3, in the continuous fiber path optimization, the total number of design variables decreases to two. These design variables are T_0, T_1 and the rotation angle is taken as zero.

The coupling procedure of Nastran and MATLAB is done in the same manner as explained in Chapter 3. However, in the continuous fiber path optimization, while calculating the fiber orientation angles for each element, the element locations must be read in so that fiber orientations angle of each element can be calculated. Therefore, nodal information of all elements is needed to calculate the fiber orientation angle using Eqns. 4.2 and 4.3. Since for the zero rotation angle ($\emptyset$) case x' is same as x, fiber orientation angles are dependent on only the x' coordinates of the finite elements. In the main optimization code, for a more robust solution procedure, all the grid and element information are read by a different MATLAB input ,"*.m", file and processed with a different "*.m" file to calculate the coordinates of the centroids of each element. Then, fiber orientation angles of every element are calculated at the element centroids.

For the finite element mesh composed of quadrilateral elements, the centroids of the finite elements are calculated by averaging the x coordinates of the corner grid points of each element.

In the upcoming section of this chapter continuous fiber orientation is performed for buckling load maximization. With the buckling analysis and the added routines to create the continuous fiber path, the previous MATLAB input file which is written for the discrete fiber orientation angle is changed. The code used and more detailed explanations are given in Appendix C.

A flowchart for the continuous fiber path optimization procedure is given in Figure 4.8. For the optimization of continuous fiber orientation, the steps for reading the geometry and element properties, calculating the mid-point of the elements and calculation of the fiber orientation angles of the elements are added to the analysis and optimization procedure. The variables are written directly to the Nastran input file, and this is different from the discrete fiber orientation angle case studies. In discrete fiber orientation optimization case studies, the orientation value in the Nastran input file is multiplied with the variable determined in genetic algorithm. The multiplied value is written to the orientation field for the composite property entry. Also in the continuous fiber path optimization, curvature constraint of the fiber path is also added to the solution procedure.

Curvature constraints are processed in the genetic algorithm operations. Constraints are defined in a MATLAB file (*.m), which includes a simple code to calculate the curvatures at the centroids of the finite elements. As noted before, centroids are calculated in another subroutine, and extracted to the MATLAB file which calculates curvatures. The subroutine which calculates the centroidal locations is written for the different case studies performed for continuous fiber path optimization. The same code is also used to calculate the fiber orientation angles of all the elements. The code outputs the curvature values calculated at the midpoints of all elements. The genetic algorithm then creates the populations by taking the curvature constraints calculated at each element into account.

The population must be selected carefully to have a better solution for the case studies as mentioned in section 3.2. There are two variables in the cases studies performed in this chapter. Therefore, 30 populations is good enough to reach the global optimum for the case studies.

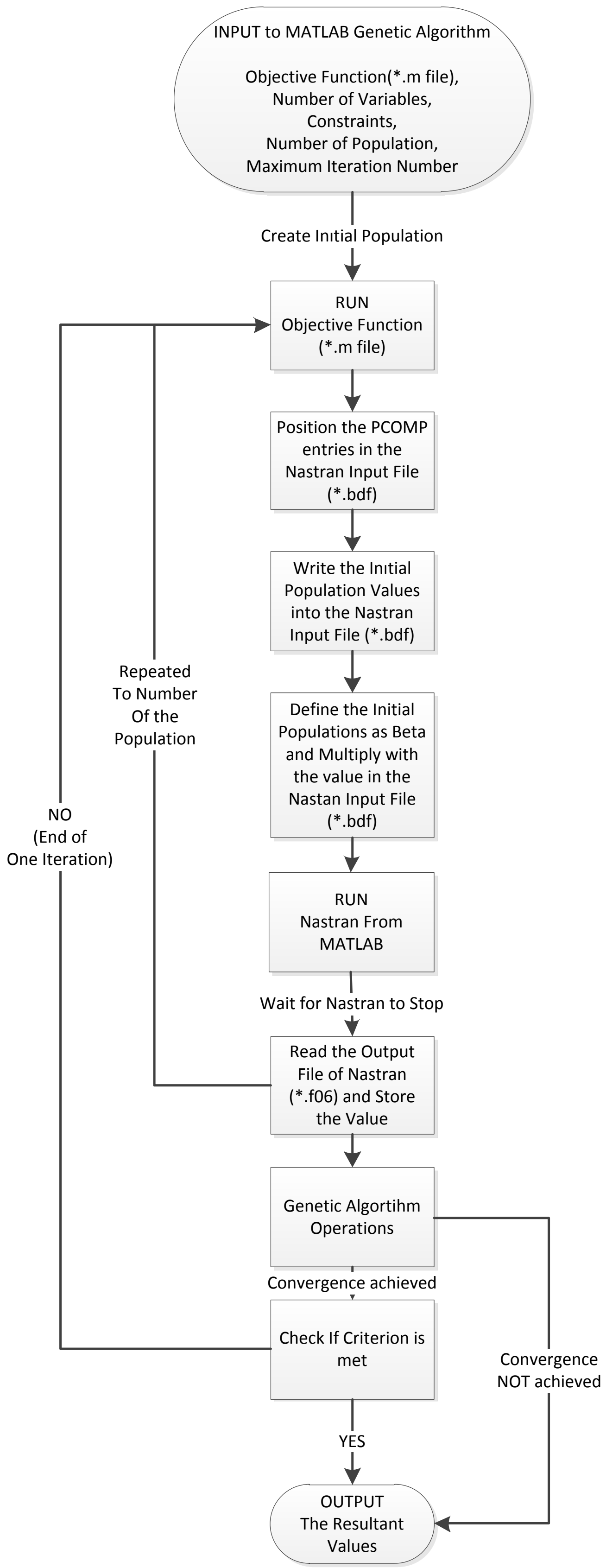


Figure 4.8: Flowchart for the continuous fiber path optimization procedure

4.3- Optimization of a Composite Plate by Using Genetic Algorithm with the Total Strain Energy as the Objective Function

In this section, the case studies performed with the discrete fiber orientation are repeated by employing continuous fiber orientation in the optimization procedure. The case study on the cantilevered rectangular plate studied in Reference 3 is taken here as the first example. The load and displacement boundary conditions are same as described in sections 3.1.1 and 3.2.1. Figure 4.9 shows the cantilevered plate and the load applied.

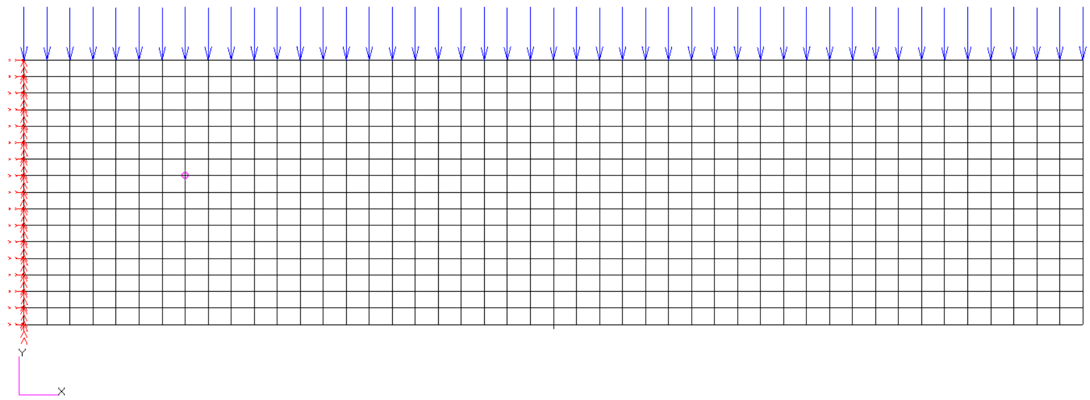


Figure 4.9: Loads and the displacement boundary conditions for the cantilevered plate

Material properties and the thickness of the plate are given Chapter 3, so they are not repeated here.

For the case studies in this section, differing from the previous case studies in this chapter which are discrete fiber orientation case studies, there is also curvature constraint applied during the optimization procedure. Curvature constraint for the fiber path is applied because; the aim of this section is to perform continuous fiber path optimization of composite plates which can be manufactured in practice by a suitable fiber placement machine.

4.3.1- Optimization of a Rectangular Single Layer Composite Plate under Transverse In-Plane Loading

To see the effect of the continuous fiber paths on the composite plate stiffness, the case study in section 3.2.1, which is taken from Reference 3, is repeated. By employing the optimization procedure shown in the flowchart in Figure 4.8, the optimization is performed for the composite plate under in-plane load. The load and the displacement boundary conditions are shown in Figure 4.9. The resultant optimized fiber orientations at the element level are shown in Figure 4.10.

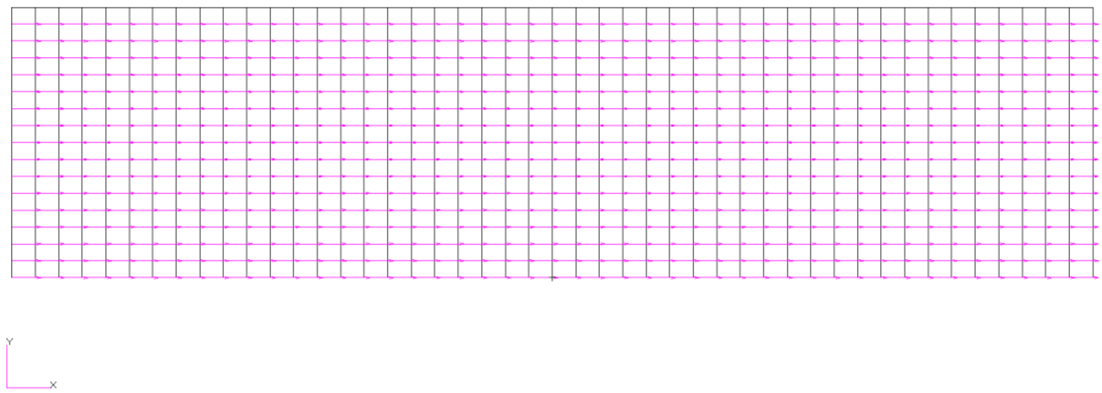


Figure 4.10: Optimized orientation for the single layer composite plate

The optimized angles for the continuous fiber paths are $\langle 0.0012|0.000156\rangle$. The angles are very small. The minimum value for the total strain energy is obtained with the zero degree straight fibers. The continuous fiber paths are shown in Figure 4.11. The continuous fiber orientation for the composite plate is sketched with “Fiber Path Visualization Tool” in Reference 32.

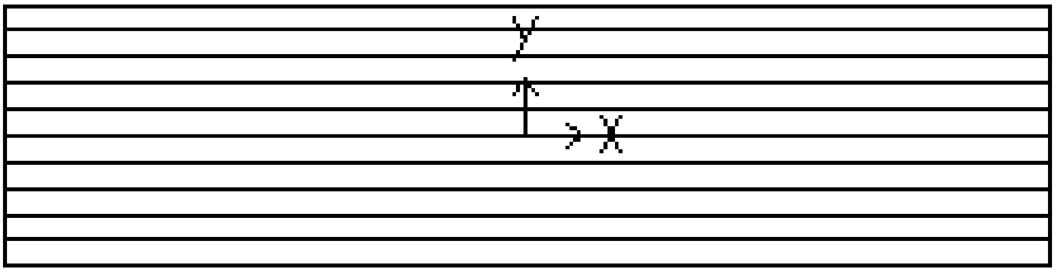


Figure 4.11: Optimized continuous fiber paths for the case study

The optimization steps are shown in Figure 4.12. The optimization is finished fast because the optimized values for the variables are zero.

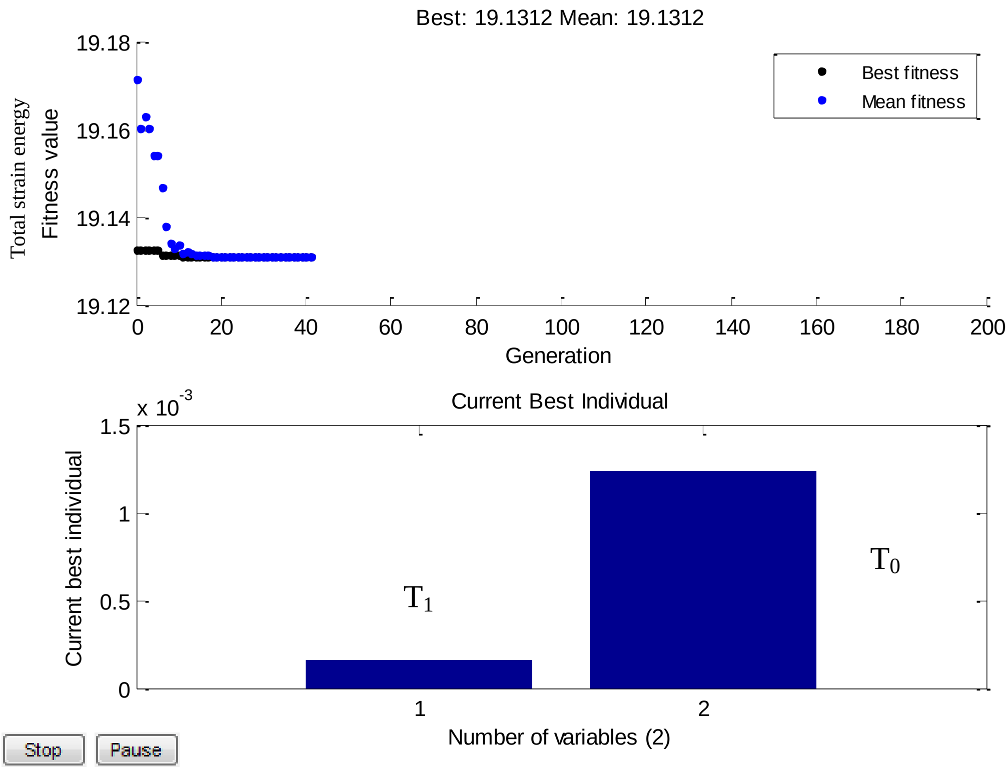


Figure 4.12: Optimization process for the single layer cantilevered plate

Optimum value for the case study can be seen from Figure 4.12. The optimum total strain energy value for the case study is 19.13 K. Joules. This value is as twice as the case study performed in section 3.1.1. The total strain energy can be twice as higher because the fiber orientations are constraint with linear equations. The total run time for the particular case is about 2 hours. After the optimization, one can easily say that for a manufacturable single layer composite plate under in-plane bending load, the best orientation is zero degrees.

4.3.2- Optimization of a Rectangular 4 Layer Composite Plate under Transverse In-Plane Loading

Case study on a four layer composite plate under in-plane load, is performed again to see the effect of the continuous fiber paths. For the particular case study, a four layer laminate with a stacking sequence of $[+\theta/-\theta]_s$ is considered. The response of the optimization might be different with respect to the previous study performed in 4.3.1. The genetic algorithm is started with 30 populations to have a better solution in the design space. Since there are 2 variables in the particular problem the population number is considered to be enough to reach global minimum. Figure 4.13 and 4.14 shows the optimized orientations for each element. Figure 4.15 shows both optimized positive and negative orientations for the 4 layer composite plate.

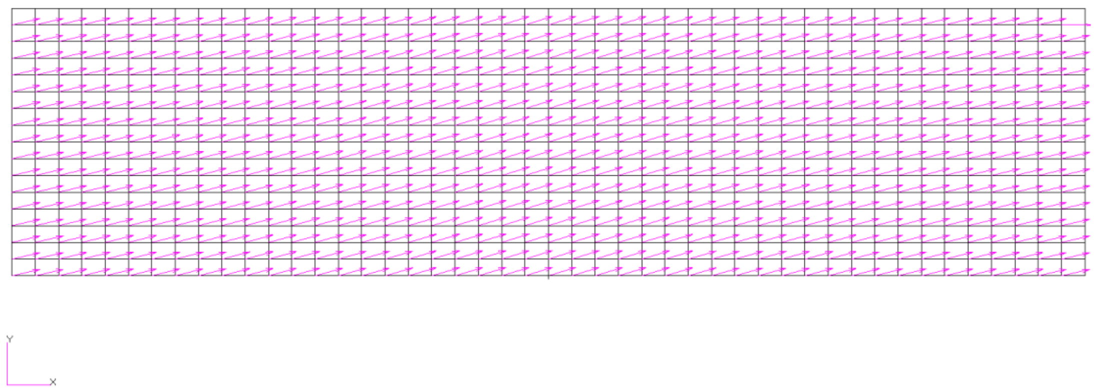


Figure 4.13: Optimized upper layer orientations of the 4 layer composite plate

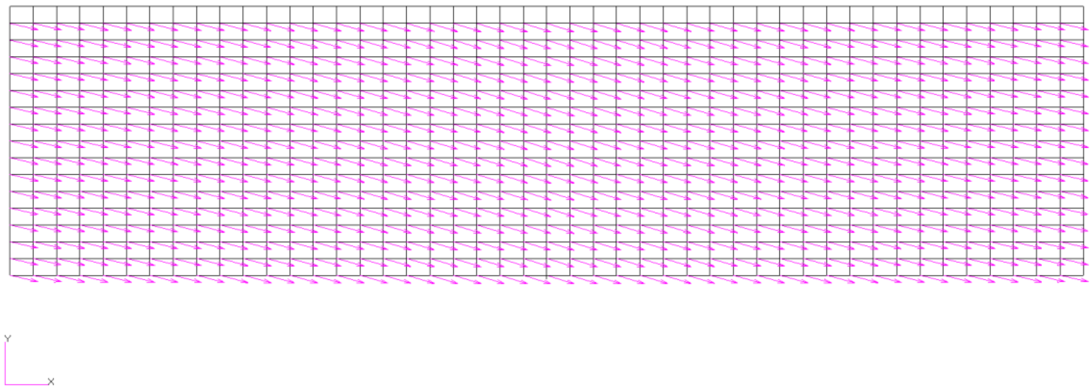


Figure 4.14: Optimized lower layer orientations of the 4 layer composite plate

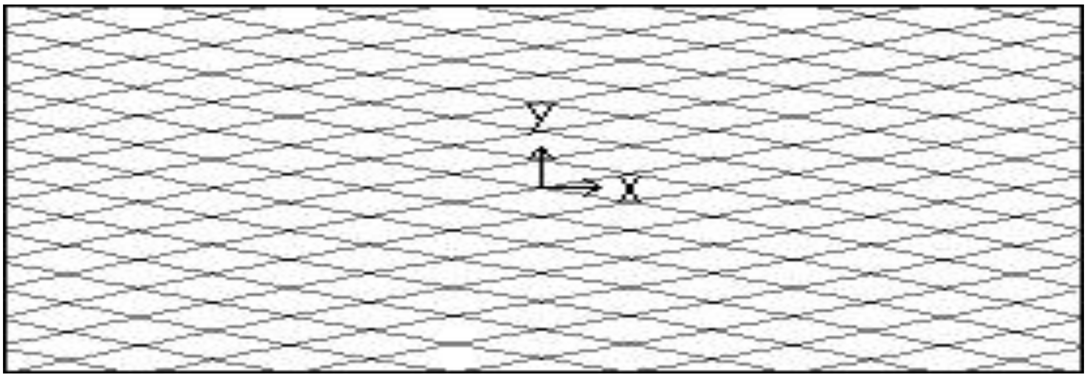


Figure 4.15: Optimized fiber paths through the 4 layer composite plate

The optimized fiber path definition for this case is $\pm\langle 12.88|17.46\rangle$. The optimization process is shown in Figure 4.16. The best and the mean objective function values can be seen from Figure 4.16. Both values are same because all the population converged to the solution individual. The optimum value for total strain energy for the particular case is determined as 12.65 K.Joules. In Figure 4.16, the upper graph gives the total strain energy versus design cycle. The blue dots show the mean value for the population created in each design cycle. The black dots show the best individual's total strain energy in the population for each iteration. In Figure 4.16, during the optimization process, bar chart shows the values of the best individuals in each iteration. Figure 4.16 shows the final optimized solution, so the values shown in the bar chart is the best solution for the particular problem. It should be noted that solution in Figure 4.16 is obtained after a series of separate runs. Figure

4.16 gives the best solution selected among 3 separate runs. The total run time for the particular case is about 7 hours; therefore the algorithm reached the minimum at about the 130th iteration and the run time for that iteration number is about 4.5 hours.

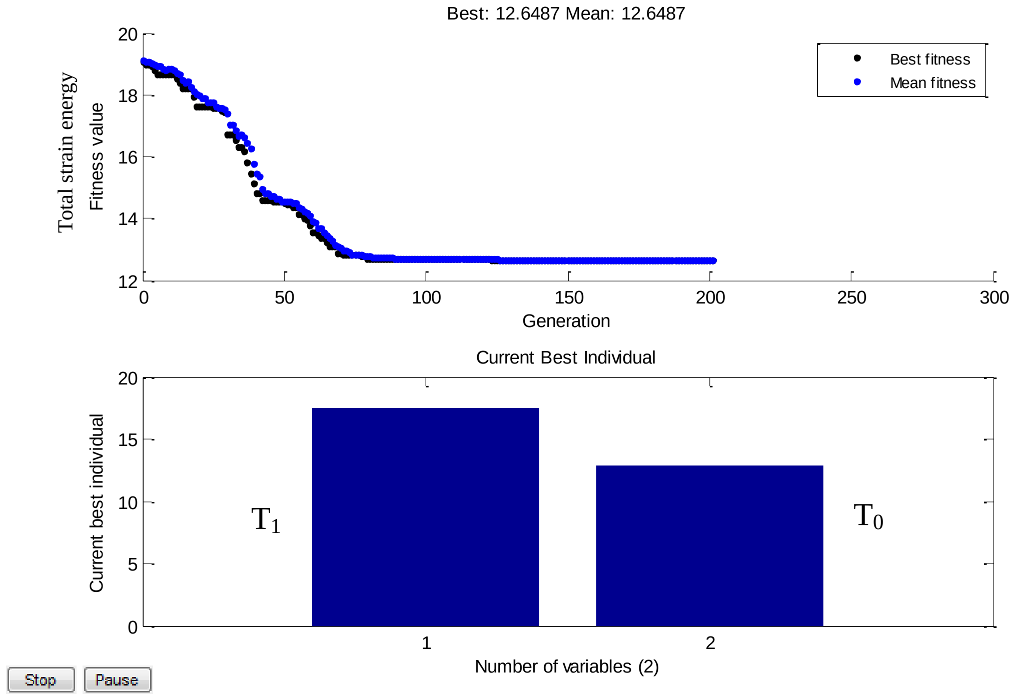


Figure 4.16: Optimization process for the 4 layer cantilevered plate

Since the two angles T_0 and T_1 are close to each other, from Eqn. 4.2, which gives the fiber orientation angle, it can be deduced the fiber orientation looks like a straight fiber distribution. But for the particular case, such a result seems to be logical since the load is a distributed in-plane transverse load. The optimized total strain energy value is also reasonable when compared to the optimized total strain energy obtained for the discrete fiber orientation case in section 3.1.2 which is 9.47 K.Joules. It should be noted that in the previous discrete orientation case studies, every element has its unique orientation. Thus, orientation angle of every element can be adjusted to minimize the total strain energy. With unique fiber orientations for each element, in theory the plate can be made stiffer. However, it is almost impossible to manufacture a composite plate with distinct fiber orientations for each element. When one compares the results of four layer composite plate case studies with discrete fiber orientations, given in section 3.1.2 and 3.2.2, with the results of the current continuous fiber orientation case study, it can be said that the continuous fiber path optimization gives reasonable minimized total strain energy value. Table 4.1 shows the optimum total strain energy values for discrete fiber orientation angles and continuous fiber path definitions. It should be noted that the continuous fiber path definitions are created with the manufacturing constraints.

	Single Layer Composite Laminate	4 Layer Mid Plane Symmetric Composite Plate
Discrete Fiber Orientation Angles	10.8 K.Joules	9.47 K.Joules
Continuous Fiber Path Definitions	19.13 K.Joules	12.65 K.Joules

Table 4.1: Optimized total strain energy values for the discrete fiber orientation angles and continuous fiber path definitions

4.4- Optimization of a Composite Plate by Using Genetic Algorithm as First Buckling Load as the Objective Function

In the literature, the variable stiffness concept is mostly used for maximizing the buckling load [2; 30; 31]. The aim of this section of the thesis is to design a plate which can be manufactured in practice, and at the same time buckling load is maximized by optimizing the parameters of the continuous fiber equations given by Eq. 4.2. Again for the case studies performed in this section of the thesis, to improve the manufacturability, the curvature constraint is also added to the genetic algorithm.

The procedure for the implementing the constraint is explained in first part of this chapter.

For the buckling case studies, the geometry of the composite plate in Reference 30 is taken as reference. In Reference 30 there are two cases. In both cases, composite plate is subjected to a uniform displacement, u_o. In Reference 30, for case 1 the rotation angle $\emptyset$ is taken as zero, and for case 2 the rotation angle $\emptyset$ is taken as $90°$. In this thesis, for all the case studies the rotation is angle is taken as zero. Therefore, only case 1 of Reference 30 is considered as the reference solution to compare with in this part of the thesis.

For the maximization of the buckling load, a square plate of 4 meter by 4 meter is taken. Four hundred CQUAD4 elements are created on the geometry. The loads and the displacement boundary conditions are illustrated in Figure 4.16. At the left edge, all the translations are prevented, $(u = w = v = 0)$, as well as the z axis rotation $(Rz = 0, Rx \neq 0, Ry \neq 0)$. At the upper and the lower edges, the translations in the y and z directions are prevented $(u \neq 0, w = z = 0)$, and the rotation about the z axis is prevented $(Rz = 0, Rx \neq 0, Ry \neq 0)$. At the right edge of the plate, only the translation in the y direction is prevented, all the other translations and rotations are let free, $(v = 0, u \neq 0, w \neq 0$ and $Rx \neq 0, Ry \neq 0, Rz \neq 0)$. A uniformly distributed load of total value 1 kN is applied along the right edge of the plate. The given load and displacement boundary conditions are taken as same for the both case studies in this section.

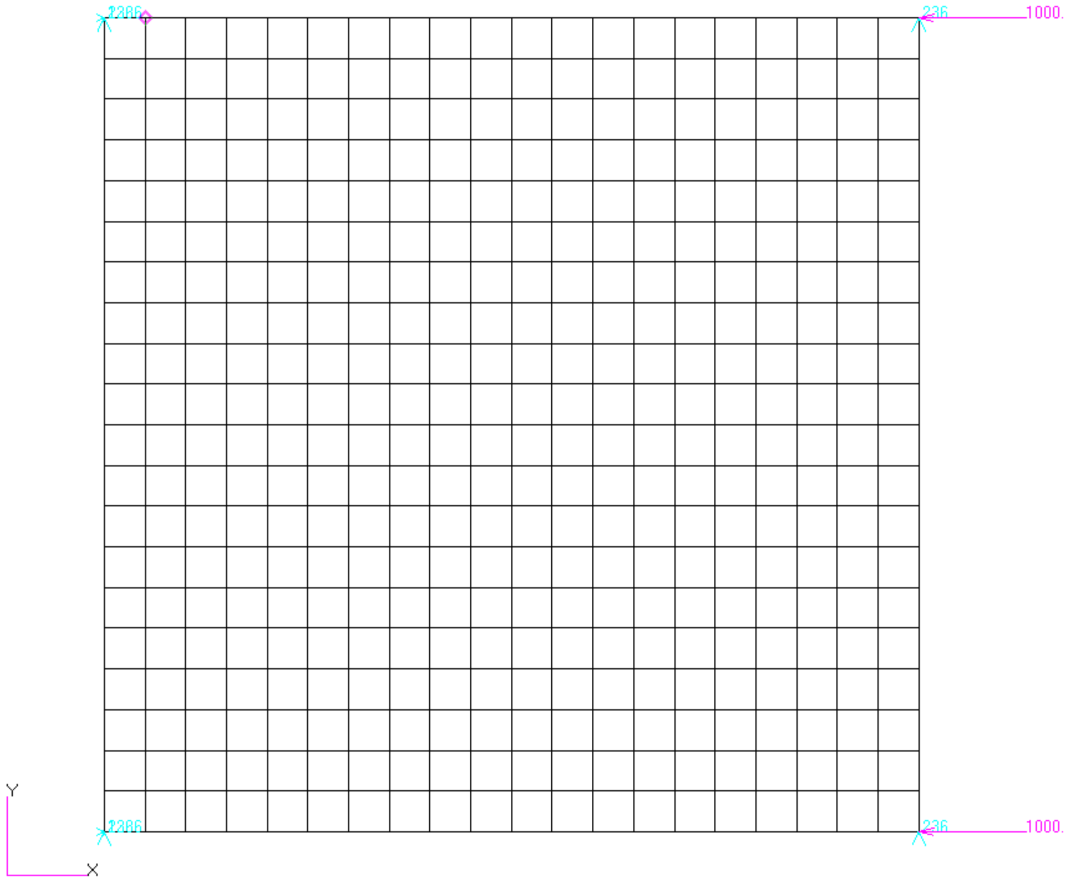


Figure 4.17: Loads and the displacement boundary conditions for the case study on maximization of buckling load

The material properties provided in Reference 30 are also used for these case studies. In the Reference 30 the calculations are done in British units. Therefore, in the present study, the values are converted into SI units for the calculations. The material properties and the total thickness used in the case studies are given below.

$$E_1 = 181\ GPa, \qquad E_2 = 10.27\ GPa, \qquad G_{12} = 7.17\ GPa, \qquad \gamma_{12} = 0.28$$

$$t = 0.1524\ m$$

The first case study is a single layer composite plate with the total thickness of the plate given above. The single layer case is performed to see that how the developed code works in the maximization of buckling load by optimizing continuous fiber path.

The second case study is a 4 layer composite plate with the same total thickness given above. The case study in Reference 30 is performed with a stacking sequence of $[\emptyset \pm \langle T_0|T_1\rangle]_{3s}$, which is the same as in the case study. Therefore, the optimized value is expected to be similar with the analysis based optimum results given in Reference 30. It should be noted that in Reference 30 finite element solution is not performed in the analyses of the composite plate. The results given in Reference 30 are the analytical solutions for different cases of the continuous fiber path definitions.

In this section, the objective function is different from all the case studies performed earlier. Maximizing the buckling load is taken as the objective function for the case studies. To assign the buckling load as, the objective function, some changes are made in the main code. For instance, the reading of the total strain energy is omitted and a new subroutine to read the eigenvalue in the Nastran output file (*.f06) is added to the main code. For the case studies in this section, solution type of Nastran is selected as linear buckling analysis rather than linear static analysis, which is used in the previous case studies. The eigenvalue is formulated as shown in Equation 4.5 [23]. "P_a" is the load applied to the structure and "P_{cr_i}" is the critical buckling load for the structure. "λ_i" is the eigenvalue, which is the coefficient relating the actual load to the buckling load.

$$P_{cr_i} = \lambda_i * P_a \tag{4.5}$$

By looking at Equation 4.5, it is obvious that if the eigenvalue is high, then the critical buckling load is also high. Therefore, since the buckling load eigenvalue is selected as the objective function, optimization problem is defined as the maximization of buckling load. At this point it should be noted that since the genetic algorithm works for objective functions which are to be minimized, a minus sign is added in front of the eigenvalue, and the minimization problem is converted into a maximization problem.

4.4.1- Optimization of Buckling Load of a Rectangular Single Layer Composite Plate under In-Plane Compressive Loading

For a simpler case study, different from the case study given in Reference 30, a single layer composite plate is optimized for the maximum buckling load. Case study is solved with the geometry, load and displacement boundary conditions given in the previous section. The manufacturing constraint, which is the curvature of the reference fiber path, is added to the optimization procedure. Since there are only two design variables, the population is selected as 30 to reach the global optimum.

The optimized continuous reference fiber orientation is given in Figure 4.18. The reference fiber path is drawn with the "Fiber Path Visualization Tool" which is given in Reference 32. The optimized fiber path definition shown in Figure 4.18 is $\langle 14.97|24.30 \rangle$. The *x* and *y* coordinates shows the midpoint of the plate. Since the manufacturing constraint is included in the optimization process, the minimum radius of curvature condition for all the points on the composite plate is satisfied. The maximum allowed curvature for the tow placement machines is 1/12 inches.

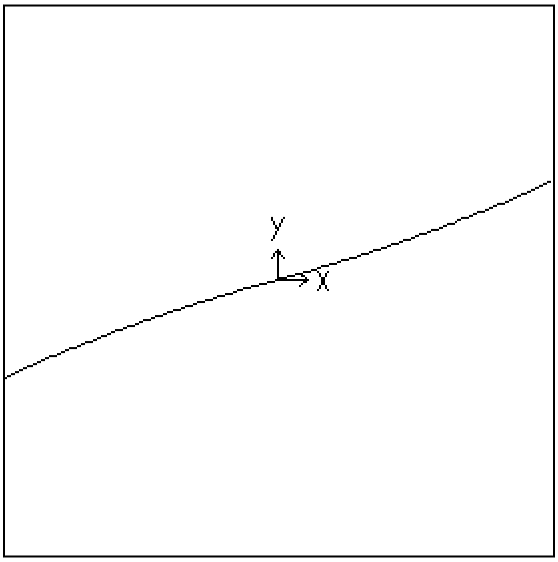


Figure 4.18: Optimized reference fiber path for the single layer composite plate

In Figure 4.19, the remaining fiber paths on the composite plate are also drawn using the tool provided in Reference 32.

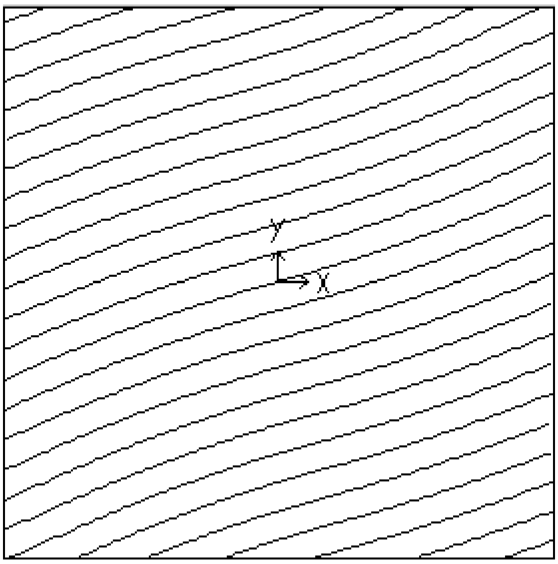


Figure 4.19: Optimized fiber paths of the composite plate

The optimization steps for the single layer buckling load maximization problem are shown in Figure 4.20. The upper graph in Figure 4.20 shows the objective function value versus design cycles. For this particular case, the objective function is the buckling load. The blue dots show the mean objective function value for the population for each design cycle. The black dots show the best individual in the population for each design cycle.

In this case study, buckling load of the composite plate is aimed to be maximized. Since, the genetic algorithm of MATLAB is a minimization algorithm the eigenvalue for the problem is multiplied by minus one to make the optimization a minimization problem. Thus, the value for the best and the mean values for the objective function have to be multiplied by -1 to get the true value of the optimized buckling load. The final objective function value for the single layer composite plate buckling load maximization problem is determined as 4.0649. Figure 4.20 is obtained after 4 separate runs. The optimum value is the best solution among all the solutions. This value is the first eigenvalue for the composite plate. It means that the buckling load for the resultant fiber orientation is nearly 4 times higher than the applied load. The lower graph shows the best individual values for the design cycle. T_1 and T_0 values are shown in the bar graph. The total run time for the particular case is about 2 hours.

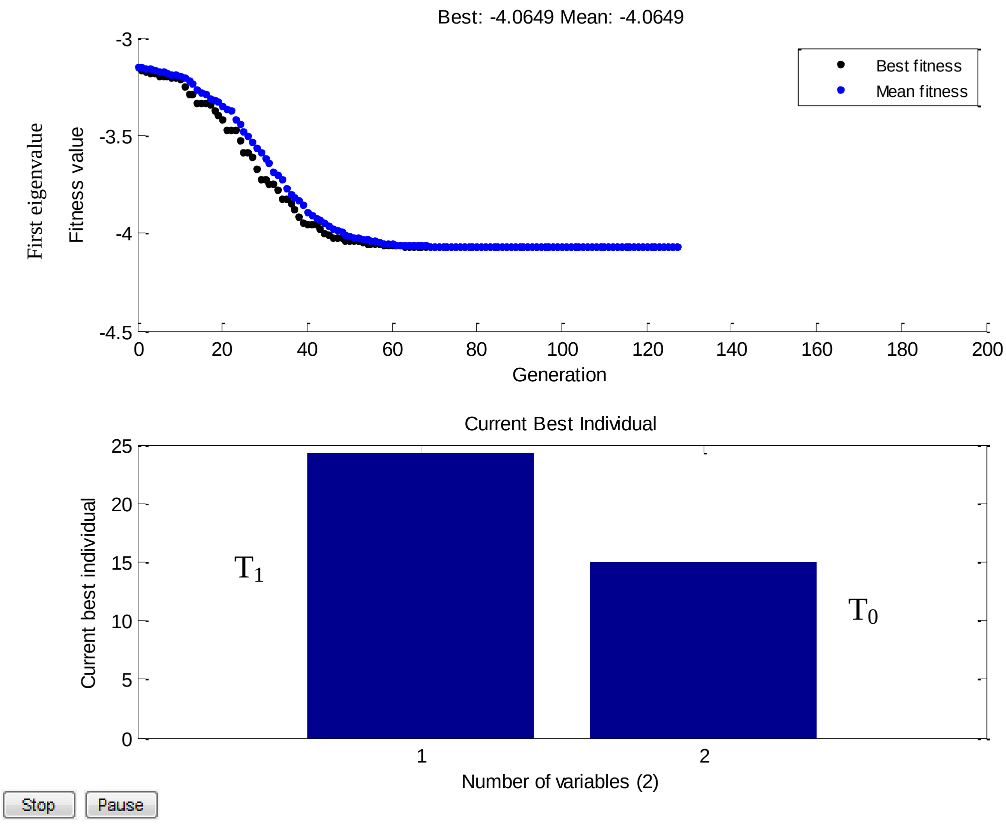


Figure 4.20: Optimization process for the single layer composite plate

The eigenvalue for the zero degree orientation for all the elements is 3.1464. For the particular case, there is almost 30 % increase in the buckling load with respect to the zero degree orientation case. These results show that the developed code works for the buckling load maximization problem.

4.4.2- Optimization of Buckling Load of a Rectangular Multilayer Composite Plate under In-Plane Compressive Loading

The multilayer composite plate example is prepared to compare the maximized buckling load determined in the present study with the buckling load obtained by Gürdal et.al. [30]. For this example, reference fiber path and stacking sequence definition is given by$[\pm\langle T_0|T_1\rangle]_{3S}$. The total thickness is kept as the same, which is given in the beginning of this chapter. Again the curvature constraint is applied in the algorithm to have a manufacturable composite plate.

The optimized reference fiber paths for the positive and the negative orientations are shown in Figure 4.21 and Figure 4.22, respectively. The reference fiber paths are again drawn using “Fiber Path Visualization Tool” [32]. The optimized reference fiber path definition for this case is determined as $\pm\langle 1.38|23.23\rangle$.

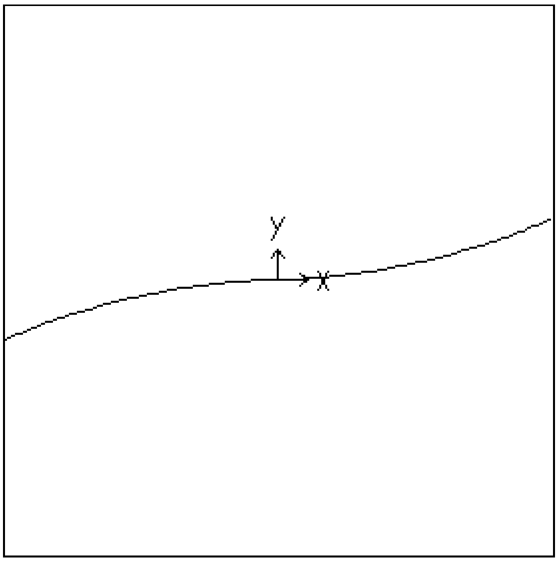


Figure 4.21: Optimized reference fiber path for the lamina with the positive fiber orientation angle

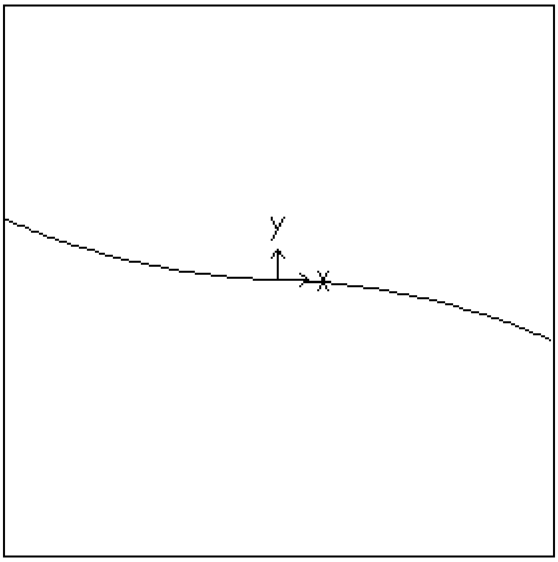
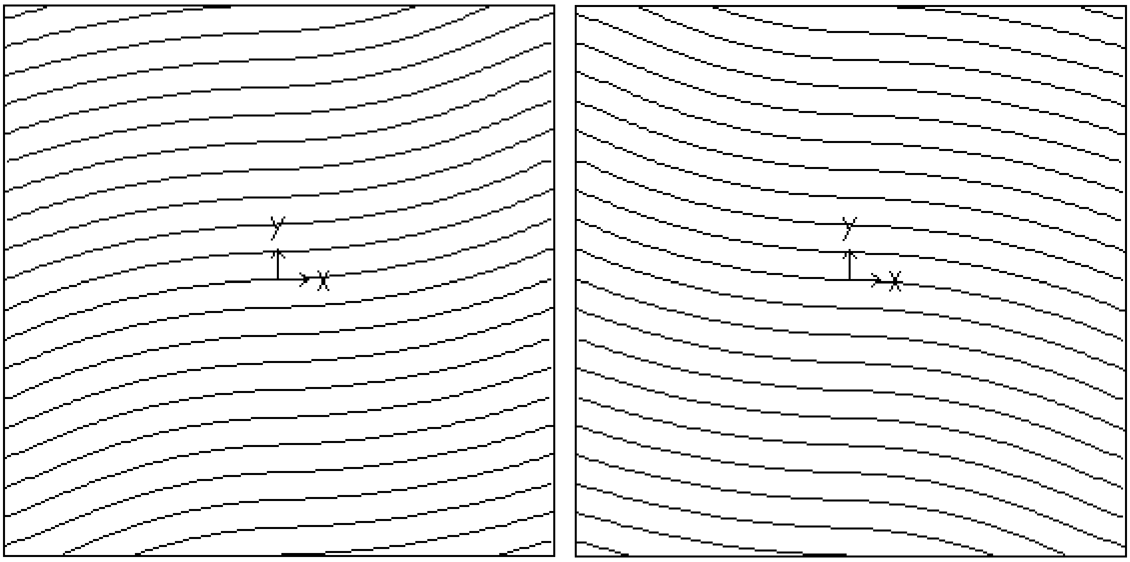


Figure 4.22: Optimized reference fiber path for the lamina with the negative fiber orientation angle

For better visualization, remaining fiber paths with the positive and the negative fiber orientations are drawn on the whole composite plate in Figure 4.23. The graph is prepared with the “Fiber Path Visualization Tool” again with the shifted fiber path method.



(a) (b)

Figure 4.23: Optimized continuous fiber orientations on the plate (a) fiber paths with positive fiber orientation angles and (b) fiber paths with negative fiber orientation angles

Figure 4.24 shows the overall optimized fiber orientation in the composite plate.

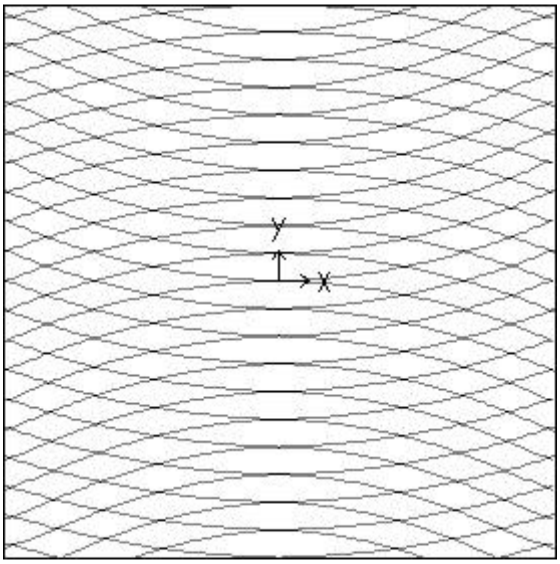


Figure 4.24: Optimized continuous fiber paths through the composite plate

The optimization steps are shown in Figure 4.25. Again the upper graph in Figure 4.25 shows the variation of the objective function with the design cycle. The blue dots show the mean objective function values for the current population for each design cycle. The black dots show the best individual in the population for each design cycle. The bar chart in Figure 4.25 shows the best values for T_1 and T_0 values which are the design variables of the optimization problem. With optimized reference fiber path definition given by $\pm\langle 1.38|23.23\rangle$, the minimum radius of curvature constraint is satisfied for all the points on the composite plate. The total run time for the particular case study is again about 2 hours.

Again for this case, the objective is to maximize the buckling load for the multilayer composite plate. The actual buckling load is determined using the first eigenvalue in the Nastran output file. The optimized eigenvalue is determined as 3.8424 for the particular case study. This means that the maximum buckling load is about 3.8424 times the applied load to the structure.

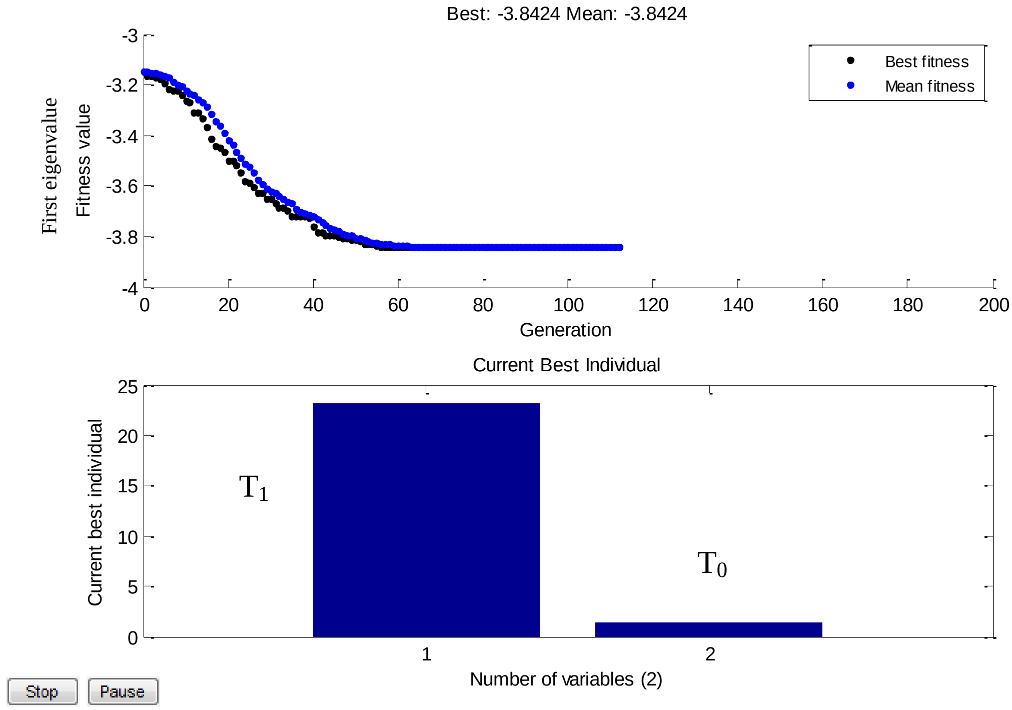


Figure 4.25: Optimization process for the multilayer composite plate

It should be noted that the lowest eigenvalue for the zero degree orientation for all the layers is 3.1464. Thus, this results shows that with the optimized fiber path there is a 22 % increase in the buckling load.

The first buckling load maximization problems last shorter than the in-plane bending total strain energy minimization case studies. The number of elements used is different in both cases. The finite element model of the buckling load maximization case studies has about half of the elements compared to the total number of elements used for the total strain energy case studies.

Therefore, the run times are long for the genetic algorithm case studies, because in every function evaluation genetic algorithm reads the Nastran input file, places the variables in the Nastran input file and analyzes in Nastran. After the analysis, again MATLAB searches the objective function value in the output file of Nastran. The time costs are added up and made the run longer. For larger mesh numbers the run time of the MATLAB genetic algorithm increases significantly.

The results in Reference 30 are compatible with results determined in this case study. With the analytical studies on the composite plates in Reference 30, it is determined that the best fiber orientation for the buckling load response is $\langle 0° | 50° \rangle$. However, in Reference 30 the manufacturing constraint is not taken into consideration. In Reference 30, for case 1 the best fiber orientation is obtained for the fiber path definitions with $T_0 = 0°$. However, T_1 value of 50^{o} is significantly higher than the optimum value of 23.23^{o} determined in the present study. This difference is considered to be due to imposing manufacturing constraint in the current study. By looking at the buckling performance graph for case 1 in Reference 30 shown in Figure 4.26, it is seen that the normalized buckling load obtained is similar to the buckling load obtained in this thesis.

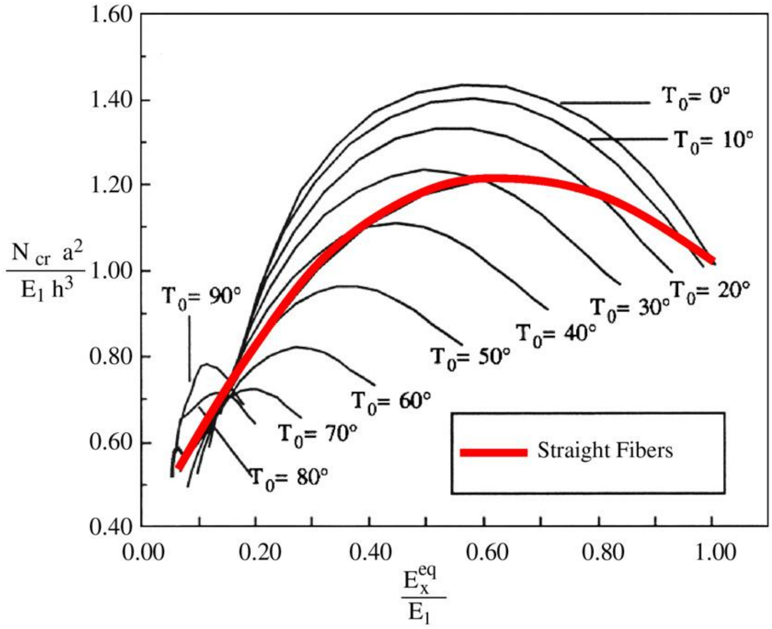


Figure 4.26: Buckling and stiffness perfomance of case 1 in Reference 30, taken from Reference 30

The vertical axis represents the way that Gürdal et.al. [30] show the normalized critical buckling load. In Figure 4.26, “N_{cr}” is the critical buckling load, “a” is the panel length, “h” is thickness of the panel and “E_1” is the longitudinal fiber modulus. The horizontal axis represents the normalized axial stiffness. The red line on the graph shows the variation of the buckling load of constant-stiffness straight fiber panels. The fiber orientation angle is changed from 0^o (from right of the red line) to 90^o (the left end). For the variable stiffness panels, the curves are created by changing T_0 from 0^o to 90^o with the increments of 10^o, and for a given T_0 value, T_1 is varied between 0^o and 90^o. The point (1.00, 1.00) in Figure 4.26 corresponds to the 0^o straight fiber orientation. It is mentioned in Reference 30, that the maximum buckling load is achieved by a fiber path definition of $\langle 0° | 50° \rangle$ for the variable stiffness panels and a $\pm 32°$ straight fiber laminate at the normalized stiffness value of 0.65. When optimized buckling load of the present study is compared with the best buckling load determined by Gürdal et.al. [30], it can be seen that in the present study the buckling load is increased by 22 % compared to the straight fiber case with zero degree fiber orientation. On the other hand, in Reference 30, from Figure 4.26 it can be seen that buckling load of variable stiffness panel is increased by approximately 44 % compared to the straight fiber case with zero degree fiber orientation. As it is mentioned before, the difference is considered to be due to the application of the curvature constraint in the present study.

To see the curvature effect more clearly, an additional unconstrained buckling load maximization case is studied. The case study has the same geometry, load and properties as the constrained buckling load maximization case study. Figure 4.27 shows the optimization process of the unconstrained optimization of the square plate. From Figure 4.27, it is seen that the maximum value for the eigenvalue is determined as 4.51. The maximum value of the eigenvalue is increased about 43 % with respect to the zero degree straight fiber orientation for the square plate. The resultant orientation of the composite fibers is $\pm\langle 1.1038|48.5896\rangle$. The resultant fiber orientation is also compatible with the analytic study in Reference 30. The lowest optimum eigenvalue, which is determined analytically in Reference 30, is about 44 % percent higher than the lowest eigenvalue of the zero degree straight fiber orientation case. The little difference may be the difference between the finite element solution and the analytical solution. The results of the constrained and the unconstrained optimization problems show that when the curvature constraint is removed, there is a definite improvement in the optimized lowest eigenvalue. Such an improvement is expected, because curvature constraint restricts the design space. However, in order to be able to manufacture the composite laminate, it is necessary to apply curvature constraint since the fiber placement machines have a capability to lay tows above a minimum turning radius.

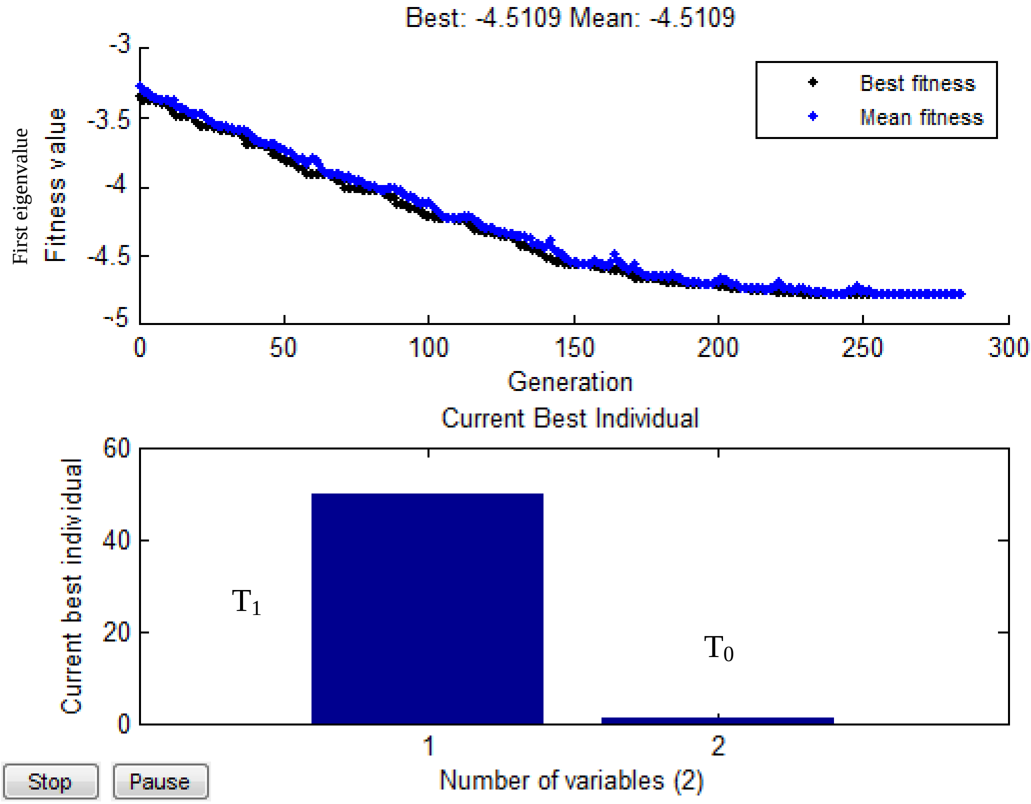


Figure 4.27: Optimization process of unconstrained buckling maximization case study

It is noted that the unconstrained optimization solution converges with higher number of iterations compared to the constrained optimization solution. Figure 4.27 also shows that the mean fitness value also does not exhibit a linear variation differing from the previous case study. It is obvious that unconstrained optimization takes more time since the design space is wider, and on the contrary, with the constrained optimization, the optimization process lasts shorter. Table 4.2 summarizes all the maximum eigenvalues of the case studies performed including the zero degree straight fiber orientation. The percent increase of the buckling load is also given in a separate column in Table 4.2. Table 4.2 shows that with the curvilinear fiber orientation, the buckling load can be increased compared to straight fiber case resulting in better structural performance. Table 4.2 also reveals that the use of single or multilayer also makes a difference in the optimized buckling load. However, it should be noted that in structural problems other considerations such as failure response must also be taken into consideration in deciding on a laminate configuration.

	Maximum Eigenvalue	% of increase compared to zero degree fiber orientation case
Zero Degree Straight Fiber Orientation Composite Plate	3,1464	
Constrained Single Layer Continuous Fiber Composite Plate	4,0649	29,19
Constrained Multi Layer Continuous Fiber Composite Plate	3,8424	22,12
Unconstrained Multi Layer Continuous Fiber Composite Plate	4,5109	43,37

Table 4.2: The maximum eigenvalues and the increase percentages of the case studies performed

Chapter 5

5- Conclusion

Fiber orientation angle is one the most significant variables affecting the mechanical behavior of composite laminates. As explained in Chapter 1, the curvilinear fiber paths in composite structures allow the modification of load paths within the composite laminate. With the curvilinear fiber paths more favorable stress distributions and improved laminate performance can be obtained. The work done in this thesis presents a fundamental study on the discrete fiber angle and continuous fiber path optimization of composite structures.

In the first part of the thesis, material orientation angles of 2D orthotropic materials are optimized to minimize the total strain energy under different load cases. In these case studies, material orientation angles at the element level are taken as the design variables. Since material orientation angles are defined as design variables at the element level, optimization of discrete fiber orientation is performed. Firstly, two simple case studies are performed to check the Nastran's performance. Then, the case studies in Reference 3 are solved to compare the solutions of optimized fiber orientations. The objective function in Reference 3 is defined as the stiffness which is to be maximized. In the case studies performed with Nastran, the objective function is defined as the minimization of the total strain energy, which maximizes the stiffness of the structure. Nastran uses gradient based optimization algorithm and an approximate model for the finite element models. As the case study, rectangular plate under in-plane bending load case is solved for different initial angles assigned to the element orientations. With initial angle trials, the best initial angle for the elements is determined as 15^o. For the rest of the case studies which are optimized with Nastran optimization module, 15^o is used as the initial element orientation angle. It is observed that Nastran gives satisfactory results when compared with the optimized fiber orientation angles determined in different case studies in Reference 3.

In Chapter 3, optimization of discrete fiber orientation angles is performed for composite laminates. In the case studies performed, the objective function is again taken as the total strain energy and discrete fiber orientation angles are used at the ply level. In the first section of Chapter 3, Nastran is used for the optimization of single layered composite laminate and 4 layered mid plane symmetric composite laminate with a stacking sequence of $[\theta/-\theta]_s$. These case studies are compared to the previous 2D orthotropic case studies. The optimized discrete fiber orientations are found to be similar to the optimized material orientations of the 2D orthotropic case studies. The minimum objective function is also determined to be similar to the total strain energy determined in the 2D orthotropic case studies. In the case studies performed for optimization of discrete fiber orientation angles, if the problem involves many design variables, it is noted that gradient based optimization algorithm has drawbacks in reaching the global minimum. Therefore, it is decided to try genetic algorithm as the optimizer in the remaining parts of the thesis.

In the second section of Chapter 3, Nastran finite element solver and MATLAB genetic algorithm toolbox are coupled to eliminate the drawbacks of gradient based optimization algorithm that is used in Nastran. For this purpose, an interface code is developed in MATLAB environment. For these case studies, Nastran is used for the finite element solver and MATLAB as the optimizer. The first case study is performed with the same mesh and properties as the rectangular case studies performed in Chapter 2. The population number is determined as 100 for the current case. This population number is assumed to be enough before starting the optimization run. Therefore, the run took very long because of the large number of design variables and the population number. In genetic algorithm, every design cycle needs function evaluations equal to the total number of individuals in the population. Every function evaluation means a finite element analysis with Nastran. For the specified population numbers, it is observed that MATLAB gives satisfactory results, but the optimization lasts long. Therefore, with higher population values and with parallel computing, the results would be better and the optimization will last shorter. Even with a population number of 100, the algorithm works fine. To shorten the time of the analysis, the second case study is created with a coarser mesh with the same geometry and properties for the rectangular case studies in Chapter 2. Coarser mesh reduces the number of variables for the problem. For the new case study several Nastran optimizations are performed to form a baseline for the genetic algorithm optimizations. Because Nastran optimization module is a gradient based optimization algorithm and needs different initial guesses to reach the global minimum. After performing the genetic algorithm optimization runs, it is seen that MATLAB genetic algorithm give better results than the Nastran optimization module. It is noted that with discrete fiber orientation since every element is assigned a fiber orientation angle, it is possible to obtain a better optimum solution compared to the continuous fiber path optimization. In the continuous fiber optimization, fiber orientation angle of a finite element is based on the equation of the reference fiber path which restricts the free selection of the fiber orientation angle of the individual elements.

Discrete fiber orientations are satisfactory in theory, but in practice they cannot be used in manufacturing since in actual manufacturing continuity of the curvature of the fiber path is required. In discrete fiber orientation case, there are discontinuities in the fiber orientations. To avoid the discontinuities, continuous fiber orientation concept is introduced in Chapter 4. In the first part of Chapter, the case studies in section 3.2 are repeated to see the effect of continuous fiber orientations. Again the objective function is taken as the total strain energy of the composite plate. In the continuous fiber path optimization, there are two design variables in the definition of the optimization problem. These two variables are used to define the reference fiber path on the composite structure. In case continuous fiber path optimization, curvature constraint must be applied in order to be able to manufacture the composite part. Curvature constraint is imposed because fiber placement machines which are used in manufacturing have limitations on the minimum radius of curvature that they can handle. In addition, in order to prevent the wrinkling of the fiber, a constraint must be applied on the minimum radius of curvature. Therefore, in the case studies performed in continuous fiber path optimization, manufacturing constraint is also applied in the optimization process.

In the case studies involving the minimization of total strain energy, a population size of 30 individuals is used. The stacking sequences used in the case studies are taken to be same as those used for the discrete fiber orientation case studies. It is seen that the optimized total strain energies determined by employing continuous fiber paths are very close to the optimized strain energies obtained by discrete fiber orientation which are presented in section 3.2. However, it is noted that the total strain energy values are not lower than those determined in section 3.2. The main reason for this is the use of continuous fiber paths which restricts the freedom of assigning independent fiber orientation angles to each element, unlike discrete fiber orientation angle optimization. Curvature constraints, which are added to the optimization process, restrict the allowable range of fiber orientation angles.

The second part of Chapter 4 is devoted to the maximization of the first buckling load of a square plate. These case studies are prepared by referencing the plate geometry used in Reference 30. The material properties and geometrical properties are also taken from Reference 30 for the case studies performed in this section. Curvature constraints are taken as same as the previous continuous fiber path optimization problems studied in section 4.1. First, a single layer composite plate, which is not studied in Reference 30, is taken to try the algorithm. As a result of continuous fiber path optimization nearly 30% increase is obtained in the first buckling load with respect to the zero degree straight fiber composite plate. In the second case study, the same stacking sequence which used in Reference 30 is considered. In this case study, the increase in the buckling load is determined to be nearly 22% with respect to zero degree straight fiber composite plate. This increase is lower than what has been determined in Reference 30. The main reason for this difference is considered to be due to the use of curvature constraint in the present study. Therefore, in the present study the optimized values for T_0 and T_1 which define the reference fiber path did not come out to be same as the corresponding values determined in Reference 30. To check the effect of the curvature constraint, a further study is conducted by removing the curvature constraint from the definition of the optimization problem for the four layered composite laminate. It is seen that when the curvature constraint is removed, 43.4% increase in the lowest buckling load is obtained. This increase is very close to the 44 % increase in the lowest buckling load determined in Ref. 30.

Case study on the maximization of the buckling load by optimizing continuous fiber paths shows the great potential that the variable stiffness concept presents. This thesis presents a fundamental study to introduce the potentials of the variable stiffness composite structures which can be manufactured by the fiber placement machines. It is considered that in the future fiber placement technology will be the major production method of composite structures due to the automation that they offer. However more importantly, with the fiber placement machines it is possible to optimize a specific response or responses of a composite structure much more effectively compared to the straight fiber case using stacking sequence optimization.

As for the future work, the current work can be improved in many respects. Some of future work items are listed below:

- In the present study shifted fiber paths are used in the continuous fiber path optimization. The study can be extended such that parallel fiber paths can be employed in the optimization process. As it was noted before, parallel fiber paths are more suitable in terms manufacturing.

- In the present study, reference fiber path is defined without considering the rotation angle at the origin of the composite laminate. The current study can be improved to include the rotation angle as the third design variable besides T_0 and T_1. Such a study will give the opportunity to explore more alternative design which may give better optimum solutions.

- In the present study, the reference fiber path is defined such that fiber orientation varies linearly along one of the coordinate axis. As one candidate future work, the reference fiber path definition can be made such that more variables can be added to the reference fiber path equation resulting in nonlinear change of the fiber orientation along axis of the composite laminate. Better orientations can be obtained by increasing the number of the variables in continuous fiber path definitions. This would increase the efficiency of the continuous fiber paths for meeting the load paths within the composite structure.

- Present study deals with straight panels. As one of the future work, curved panels can be studied. Optimization of continuous fiber paths on curved panels is very critical because in aerospace vehicles most panels are curved. In addition, fiber placement technology is also used in the manufacturing of wind turbine blades. Optimization of continuous fiber paths on the wind turbine skins is a very worthwhile study to conduct. Optimization of continuous fiber paths on curved surfaces will give the opportunity to aeroelastically tailor the composite structure. Aeroleastic tailoring is a current research area which relies on accurate positioning of fibers on curved contours to achieve a desired structural and/or aerodynamic response.

Aeroelastic tailoring can be achieved most accurately by the use of fiber placement machines

- Coupling of Nastran and MATLAB requires a finite element analysis for each function evaluation. This can be prevented by creating an approximate model same as in Nastran optimization module. This improvement will end the necessity of finite element analysis in each function evaluation. In each iteration, an approximate model, described by a simple equation, may be employed, and used for the function evaluations. This improvement will give the opportunity to solve problems with high number of variables in a shorter time improving the efficiency.

References

1. **Tatting, B. F. and Gürdal, Z.** *Automated Finite Element Analysis of Elastically-Tailored Plates.* s.l. : NASA/CR-2003-212679, December 2003.

2. **Gürdal, Z. and Tatting, B. F.** *Design and Manufacture of Tow-Placed Variable Stiffness Composite Laminates with Manufacturing Considerations.* s.l. : 13th U.S. National Congress of Applied Mechanics, June 25th 1998.

3. **Setoodeh, S., Gürdal, Z. and L.T., Watson.** *Design of Variable-Stiffness Composite Layers Using Cellular Automata.* s.l. : Computational Methods in Applied Mechanics and Engineering 195, 2006. Vol. 195. 836-851.

4. **Waldhart, C.** *Analysis of Tow-Placed, Variable-Stiffness Laminates.* s.l. : Thesis Submitted to the Faculty of the Virgina Polytechnic Institute and State University, June 5th 1996.

5. **Kutay, H.** *Stacking Sequence Optimization of a Composite Pressure Vessel by Genetic Algorithm.* s.l. : Thesis Submitted to the Graduate School of Natural and Applied Sciences of METU, September 2007.

6. **Marsh, G.** *Automating Aerospace Composite Production with Fiber Placement.* s.l. : http://www.reinforcedplastics.com/view/17236/automating-aerospace-composites-production-with-fibre-placement/, June 28th 2011.

7. *Optimization of Fiber Reinforced Composites.* **Duvaut, G., et al.** s.l. : Composite Structures, 2000, Vol. 48.

8. *Optimal Design of Symmetrically Laminated Plates for Minimum Deflection and Weight.* **Walker, Mark, Reiss, Talmon and Adali, Sarp.** s.l. : Composite Structures, 1997, Vol. 39.

9. *Structural Optimization of an Aeroelastically Tailoring Composite Flat Plate Made of Woven Fiberglass/Epoxy.* **Attaran, A., et al.** s.l. : Jurnal Mekanikal, 2006, Vol. 22.

10. **Hasançebi, Oğuzhan.** *CE 587 Structural Optimization - Lecture notes.* 2009.

11. *Optimum Design of Composite Laminates for Maximum Buckling Load Capacity Using Simulated Annealing.* **Erdal, Özgür and Sönmez, Fazıl O.** s.l. : Composite Structures, 2005, Vol. 71.

12. **Kogiso, Nozomu, et al.** *Genetic Algorithms with Local Improvement for Composite Laminate Design.* Blacksburg : s.n., 2003.

13. *Stacking Sequence Design of Composite Laminates for Maximum Strength Using Genetic Algorithm.* **Park, J. H., et al.** s.l. : Composite Structures, 2001, Vol. 52.

14. *Optimum Design of Laminated Composite Plates with Cutouts Using a Genetic Algorithm.* **Sivakumar, K., Iyengar, N. G. R. and Deb, Kalyanmoy.** s.l. : Composite Structures, 1998, Vol. 42.

15. *Stacking Sequence Blending of Multiple Composite Laminates Using Genetic Algorithms.* **Soremekun, Grant, et al.** s.l. : Composite Structures, 2002, Vol. 56.

16. *A Fortran 90 Genetic Algorithm Module for Composite Laminate Structure Design.* **McMahon, M. T., et al.** s.l. : Engineering with Computers, 1998, Vol. 14.

17. *Application of Genetic Algoritms to Optimal Tailoring of Composite Materials.* **Sadagopan, D. and Pitchumani, R.** s.l. : Composites Science and Technology, 1998, Vol. 58.

18. **Benford, Andrew and Tinker, Micheal L.** *Comparison of Structural Optimization Techniques for a Nuclear Electric Space Vehicle.*

19. **Soremekun, Grant A. E.** *Genetic Algorithms for Composite Laminate Design and Optimization.* Blacksburg : Thesis Submitted to the Faculty of the Virginia Polytechnic Institute and State University, 1997.

20. **MATLAB.** *Genetic Algorithm and Direct Search Toolbox.* s.l. : The MathWorks, January 2004.

21. **Darrel, Whitley.** *A Genetic Algorithm Tutorial.*

22. **MathWorks.** *MATLAB Product Help.* s.l. : Math Works.

23. *MD Nastran R3 Documentation.* s.l. : Design Sensitivity and Optimization User's Guide.

24. **Ekren, M. and Kayran, A.** *Structural Optimization of Wing Torque Boxes Using Different Idealization in the Finite Element Model.* s.l. : 5th Ankara Aerospace Conference, 17-19 August 2009.

25. **Pedersen, P.** *On optimal Orientation of Orthotropic Materials.* s.l. : Structural Optimization 1, 1989. 101-106.

26. *Optimization of Elastically Tailored Tow-Placed Plates With Holes.* **Jegley, Dawn C., Tatting, Brian F. and Gürdal, Zafer.** s.l. : AIAA.

27. *Use of Curvilinear Fiber Format in Composite Structure Design.* **Hyer, M. W. and Charette, R. F.** 6, s.l. : AIAA Journal, June 1991, Vol. 29.

28. **Langley, Patrick Tyler.** *Finite Element Modeliing of Tow-Placed Variable-Stiffness Composite Laminates.* s.l. : Thesis Submitted to the Faculty of the Virgina Ploytechnis Institute and State University, 1999.

29. *In-plane Response of Laminates with Spatially Varying Fiber Orientations: Variable Stiffness Concept.* **Gürdal, Zafer and Olmedo, Reynaldo.** 4, s.l. : AIAA Journal, April 1993, Vol. 31.

30. *Variable Stiffness Composite Panels: Effects of Stiffness Variation on the In-Plane and Buckling Response.* **Gürdal, Z., Tatting, B. F. and Wu, C. K.** s.l. : Composites Journal, 2008, Vol. 39.

31. *Variable Stiffness Composite Panels: Buckling and First-Ply Failure İmprovements over Straight-Fiber Laminates.* **Lopes, C. S., Gürdal, Z. and Camanho, P. P.** s.l. : Computers & Structures, 2008, Vol. 86.

32. *Adoptech.* [Online] [Cited: February 6, 2012.] http://www.adoptech.com/fibersteering/FiberSteerExample.htm.

APPENDICES

Appendix A

Description of the Nastran input file which is prepared to perform fiber orientation angle and thickness of a composite laminate

Nastran's optimization solver

Shows that which design design response is the objective function and the design objective tobe minimized.

Defines constraint sets for the problem.

With the PUNCH(NEWBULK) in the ECHO entry, just for the design optimization case, it is available to provide an unsorted Bulk Data file following a design optimization task that has updated entries replacing the original entries. The output of this is a *.pch file.

```
SOL 200
$ Direct Text Input for Executive Control
CEND
TITLE = MD Nastran job created on 19-Apr-10 at 11:24:42
ECHO = SORT,PUNCH(NEWBULK)
DESOBJ(MIN)  =  1
DESSUB       =  21
ANALYSIS = STATICS
$ Direct Text Input for Global Case Control Data
SUBCASE 1
$ Subcase name : Default
   SUBTITLE=Default
   SPC = 2
   LOAD = 2
   DISPLACEMENT(SORT1,REAL)=ALL
   SPCFORCES(SORT1,REAL)=ALL
   STRESS(SORT1,REAL,VONMISES,BILIN)=ALL
BEGIN BULK
PARAM    POST     -1
PARAM    PRTMAXIM YES
PARAM    NASPRT   1
```

Results in an .op2 file that is read onto the MSC.Patran data Base.

Specifies how often data recovery is performed and printed in SOL 200

Controls the printout of the maximums of applied loads, single-point forces of constraint, multipoint forces of constraint, and displacements. The printouts are titled "MAXIMUM APPLIED LOADS", "MAXIMUM SPCFORCES", "MAXIMUM MPCFORCES", and "MAXIMUM DISPLACEMENTS"

```
$ Elements and Element Properties for region : p1
$ Composite Property Record created from P3/PATRAN composite material
$ record : comp_el1
$ Composite Material Description :
PCOMP    1                                       0.      0.      SYM
         1       .1      -30.    YES     1       .1      30.     YES
$ Pset: "p1" will be imported as: "pcomp.1"
CQUAD4   1       1       1       2       7       6
$ Elements and Element Properties for region : p2
$ Composite Property Record created from P3/PATRAN composite material
$ record : comp_el2
$ Composite Material Description :
PCOMP    2                                       0.      0.      SYM
         1       .1      -30.    YES     1       .1      30.     YES
$ Pset: "p2" will be imported as: "pcomp.2"
CQUAD4   2       2       2       3       8       7
$ Elements and Element Properties for region : p3
$ Composite Property Record created from P3/PATRAN composite material
$ record : comp_el3
$ Composite Material Description :
PCOMP    3                                       0.      0.      SYM
         1       .1      -30.    YES     1       .1      30.     YES
$ Pset: "p3" will be imported as: "pcomp.3"
CQUAD4   3       3       3       4       9       8
$ Elements and Element Properties for region : p4
$ Composite Property Record created from P3/PATRAN composite material
$ record : comp_el4
$ Composite Material Description :
PCOMP    4                                       0.      0.      SYM
         1       .1      -30.    YES     1       .1      30.     YES
$ Pset: "p4" will be imported as: "pcomp.4"
CQUAD4   4       4       4       5       10      9
$ Elements and Element Properties for region : p5
$ Composite Property Record created from P3/PATRAN composite material
$ record : comp_el5
$ Composite Material Description :
PCOMP    5                                       0.      0.      SYM
         1       .1      -30.    YES     1       .1      30.     YES
$ Pset: "p5" will be imported as: "pcomp.5"
CQUAD4   5       5       6       7       12      11
$ Elements and Element Properties for region : p6
$ Composite Property Record created from P3/PATRAN composite material
$ record : comp_el6
$ Composite Material Description :
PCOMP    6                                       0.      0.      SYM
         1       .1      -30.    YES     1       .1      30.     YES
$ Pset: "p6" will be imported as: "pcomp.6"
CQUAD4   6       6       7       8       13      12
```

PCOMP	PID	Z0	NSM	SB	FT	TREF	GE	LAM	
	MID1	T1	THETA1	SOUT1	MID2	T2	THETA2	SOUT2	
	MID3	T3	THETA3	SOUT3	-etc.-				

PCOMP entry has a unique card to input.
Simply this example is a 2 layers mid-plane symmetric composite material.
But every element has its own PCOMP property since, it is desired to have unique element properties at the end of the optimization.

Giving every element a seperate property since every element will has its unique orientation.

```
$ Elements and Element Properties for region : p7
$ Composite Property Record created from P3/PATRAN composite material
$ record : comp_el7
$ Composite Material Description :
PCOMP    7                                         0.      0.      SYM
         1       .1      -30.    YES     1       .1      30.     YES
$ Pset: "p7" will be imported as: "pcomp.7"
CQUAD4   7       7       8       9       14      13
$ Elements and Element Properties for region : p8
$ Composite Property Record created from P3/PATRAN composite material
$ record : comp_el8
$ Composite Material Description :
PCOMP    8                                         0.      0.      SYM
         1       .1      -30.    YES     1       .1      30.     YES
$ Pset: "p8" will be imported as: "pcomp.8"
CQUAD4   8       8       9       10      15      14
$ Referenced Material Records
$ Material Record : ortho
$ Description of Material : Date: 03-Nov-10           Time: 19:09:55
MAT8     1       1.352+111.69+10 .3      5.915+9 5.915+9 5.915+9 1.6
$ Nodes of the Entire Model
GRID     1                0.      0.      0.
GRID     2               .5       0.      0.
GRID     3               1.       0.      0.
GRID     4               1.5      0.      0.
GRID     5               2.       0.      0.
GRID     6                0.     .5       0.
GRID     7               .5      .5       0.
GRID     8               1.      .5       0.
GRID     9               1.5     .5       0.
GRID     10              2.      .5       0.
GRID     11               0.     1.       0.
GRID     12              .5      1.       0.
GRID     13              1.      1.       0.
GRID     14              1.5     1.       0.
GRID     15              2.      1.       0.
$ Loads for Load Case : Default
SPCADD   2       1
LOAD     2       1.      1.      1
$ Displacement Constraints of Load Set : disp1
SPC1     1       123456  1       6       11
$ Nodal Forces of Load Set : force1
FORCE    1       15      0       100000. 0.      -1.     0.
```

Continued for the remaining elements.

Material Properties

Showing the grids of the plate elements.

The applied loads.

```
$ ...DESIGN VARIABLE DEFINITION
DESVAR  1       or11    -30.    -90.    90.     0.01
DESVAR  2       or12    30.     -90.    90.     0.01
DESVAR  3       or21    -30.    -90.    90.     0.01
DESVAR  4       or22    30.     -90.    90.     0.01
DESVAR  5       or31    -30.    -90.    90.     0.01
DESVAR  6       or32    30.     -90.    90.     0.01
DESVAR  7       or41    -30.    -90.    90.     0.01
DESVAR  8       or42    30.     -90.    90.     0.01
DESVAR  9       or51    -30.    -90.    90.     0.01
DESVAR  10      or52    30.     -90.    90.     0.01
DESVAR  11      or61    -30.    -90.    90.     0.01
DESVAR  12      or62    30.     -90.    90.     0.01
DESVAR  13      or71    -30.    -90.    90.     0.01
DESVAR  14      or72    30.     -90.    90.     0.01
DESVAR  15      or81    -30.    -90.    90.     0.01
DESVAR  16      or82    30.     -90.    90.     0.01
DESVAR  17      th11    0.1     0.05    1.      0.01
DESVAR  18      th12    0.1     0.05    1.      0.01
DESVAR  19      th21    0.1     0.05    1.      0.01
DESVAR  20      th22    0.1     0.05    1.      0.01
DESVAR  21      th31    0.1     0.05    1.      0.01
DESVAR  22      th32    0.1     0.05    1.      0.01
DESVAR  23      th41    0.1     0.05    1.      0.01
DESVAR  24      th42    0.1     0.05    1.      0.01
DESVAR  25      th51    0.1     0.05    1.      0.01
DESVAR  26      th52    0.1     0.05    1.      0.01
DESVAR  27      th61    0.1     0.05    1.      0.01
DESVAR  28      th62    0.1     0.05    1.      0.01
DESVAR  29      th71    0.1     0.05    1.      0.01
DESVAR  30      th72    0.1     0.05    1.      0.01
DESVAR  31      th81    0.1     0.05    1.      0.01
DESVAR  32      th82    0.1     0.05    1.      0.01
```

DESVAR entry, defines a design variable for design optimization

Below chart shows the entries in DESVAR.

DESVAR	ID	LABEL	XINIT	XLB	XUB	DELXV	DDVAL		

Field	Contents
ID	Unique design variable identification number. (Integer > 0)
LABEL	User-supplied name for printing purposes. (Character)
XINIT	Initial value. (Real, XLB ≤ XINIT ≤ XUB)
XLB	Lower bound. (Real, default = -1.0E+20)
XUB	Upper bound. (Real, default = +1.0E+20)
DELXV	Move limit for the design variable during approximate optimization. (Real > 0.0)
DDVAL	ID of a DDVAL entry that provides a set of allowable discrete values. (Blank or integer > 0, Default = blank for continuous design variables.)

In this example, there are 32 design variables because, every element has 2 independent orientations and thicknesses.

The thickness will show how many plies are going to use while producing the designed plate.

```
$ ...DEFINITION OF DESIGN VARIABLE TO ANALYSIS MODEL PARAMETER RELATIONS
DVPREL1 1       PCOMP   1       THETA1
        1       1.
DVPREL1 2       PCOMP   1       THETA2
        2       1.
DVPREL1 3       PCOMP   2       THETA1
        3       1.
DVPREL1 4       PCOMP   2       THETA2
        4       1.
DVPREL1 5       PCOMP   3       THETA1
        5       1.
DVPREL1 6       PCOMP   3       THETA2
        6       1.
DVPREL1 7       PCOMP   4       THETA1
        7       1.
DVPREL1 8       PCOMP   4       THETA2
        8       1.
DVPREL1 9       PCOMP   5       THETA1
        9       1.
DVPREL1 10      PCOMP   5       THETA2
        10      1.
DVPREL1 11      PCOMP   6       THETA1
        11      1.
DVPREL1 12      PCOMP   6       THETA2
        12      1.
DVPREL1 13      PCOMP   7       THETA1
        13      1.
DVPREL1 14      PCOMP   7       THETA2
        14      1.
DVPREL1 15      PCOMP   8       THETA1
        15      1.
DVPREL1 16      PCOMP   8       THETA2
        16      1.
DVPREL1 17      PCOMP   1       T1
        17      1.
DVPREL1 18      PCOMP   1       T2
        18      1.
DVPREL1 19      PCOMP   2       T1
        19      1.
DVPREL1 20      PCOMP   2       T2
        20      1.
DVPREL1 21      PCOMP   3       T1
        21      1.
DVPREL1 22      PCOMP   3       T2
        22      1.
DVPREL1 23      PCOMP   4       T1
        23      1.
```

DVPREL1 defines the relation between a element property and design variables.

DVPREL1	ID	TYPE	PID	PNAME/ FID	PMIN	PMAX	C0		
	DVID1	COEF1/ 'PVAL'	DVID2	COEF2	DVID3	-etc.-			

Field	Contents
ID	Unique identification number. (Integer > 0)
TYPE	Name of a property entry, such as "PBAR", "PBEAM", etc. (Character)
PID	Property entry identification number. (Integer > 0)
PNAME/FID	Property name, such as "T", "A", or field position of the property entry, or word position in the element property table of the analysis model. (Character or Integer ≠ 0)
PMIN	Minimum value allowed for this property. If FID references a stress recovery location, then the default value for PMIN is -1.0 + 35. PMIN must be explicitly set to a negative number for properties that may be less than zero (for example, field ZO on the PCOMP entry). (Real; Default = 1.0E-20)
PMAX	Maximum value allowed for this property. (Real; Default = 1.0E+20)
C0	Constant term of relation. (Real; Default = 0.0)
DVIDi	DESVAR entry identification number. (Integer > 0)
COEFi	Coefficient of linear relation. (Real)
'PVAL'	Flag to indicate COEF1 is to be set to the current property value (Character = 'PVAL')

THETAi represents the layers orientations.
Ti represents the layers thicknesses.

```
DVPREL1 24      PCOMP   4       T2
        24      1.
DVPREL1 25      PCOMP   5       T1
        25      1.
DVPREL1 26      PCOMP   5       T2
        26      1.
DVPREL1 27      PCOMP   6       T1
        27      1.
DVPREL1 28      PCOMP   6       T2
        28      1.
DVPREL1 29      PCOMP   7       T1
        29      1.
DVPREL1 30      PCOMP   7       T2
        30      1.
DVPREL1 31      PCOMP   8       T1
        31      1.
DVPREL1 32      PCOMP   8       T2
        32      1.
$ ...STRUCTURAL RESPONSE IDENTIFICATION
DRESP1  1       VOL     VOLUME
DRESP1  27      U2      DISP                    2               15
$ ...CONSTRAINTS
DCONSTR 21      27      -0.000050.00005
$ ...OPTIMIZATION CONTROL
DOPTPRM DESMAX  10000   FSDMAX  0       P1      1       P2      1
        CONV1   .0001   CONV2   1.-20   CONVDV  .0001   CONVPR  .001
        DELP    .2      DELX    1.      DPMIN   .01     DXMIN   .05
$ Referenced Coordinate Frame
ENDDATA d0837cc6
```

Continued to the thickness desing variable relations.

DRESP1 Bulk Data entries define type-1, or first-level responses. These responses are available directly from an MSC.Nastran analysis.

DRESP1	ID	LABEL	RTYPE	PTYPE	REGION	ATTA	ATTB	ATT1	
	ATT2	-etc.-							

Field	Contents
ID	Unique entry identifier. (Integer > 0)
LABEL	User-defined label. (Character)
RTYPE	Response type. See the DRESP1 listing in "The Bulk Data Section" on page 895 of the *MSC.Nastran Quick Reference Guide*. (Character)
PTYPE	Element flag (PTYPE = "ELEM") or property entry name. Used with element type responses (stress, strain, force, etc.) to identify the property type, since property entry IDs are not unique across property types. (Character: "ELEM", "PBAR", "PSHELL", etc.)
REGION	Region identifier for constraint screening. (Integer > 0)
ATTA, ATTB, ATTi	Response attributes. See the DRESP1 listing in Bulk Data entries in Appendix C. (Integer > 0 or Real or blank)

First DRESP1 entry numbered as "1" is the objective function which is stated in the beginning of the file. It is the volume, stated as VOLUME.

Second desing response is the displacement (DISP) numbered as "27".

DOPTPRM overrides default values of parameters used in design optimization.

DOPTPRM	PARAM1	VAL1	PARAM2	VAL2	PARAM3	VAL3	PARAM4	VAL4	
	PARAM5	VAL5	-etc.-						

Field	Contents
PARAMi	Name of the design optimization parameter. For allowable names, see the DOPTPRM listing in "The Bulk Data Section" on page 895 of the *MSC.Nastran Quick Reference Guide*. (Character)
VALi	Value of the parameter. (Real or Integer; see the DOPTPRM listing in "The Bulk Data Section" on page 895 of the *MSC.Nastran Quick Reference Guide*.)

DCONSTR	DCID	RID	LALLOW/ LID	UALLOW/ UID	LOWFQ	HIGHFQ			

Field	Contents
DCID	Design constraint set identification number. (Integer > 0)
RID	DRESPi entry identification number. (Integer > 0)
LALLOW	Lower bound on the response quantity. (Real, Default = -1.0E20)
LID	Set identification of a TABLEDi entry that supplies the lower bound as a function of frequency.
UALLOW	Upper bound on the response quantity. (Real, Default = 1.0E20)
UID	Set identification of a TABLEDi entry that supplies the upper bound as a function of frequency.
LOWFQ	Low end of frequency range in Hertz. (Real ≥ 0.0, Default = 0.0)
HIGHFQ	High end of frequency range in Hertz. (Real ≥ LOWFQ, Default = 1.0E+20)

DCONSTR places limits on a design response. Numbered as "21" and defined as the constraint set in the beginning of the code.

Appendix B

Developed MATLAB code for the fitness function input to the genetic algorithm for the discrete fiber angle orientations of the composite plates

- First, from an input file, which calls another input file that sets the algorithm options.

```
%inputting the variable number
kacElement=1472;
%inputting the boundary conditions, maximum number of design cycles,
%population size
[x,fval] =
GA6(kacElement,ones(1,kacElement)*(0.01),ones(1,kacElement)*9,400,20
0);
```

- GA6 is the file that sets the options for the algorithm

```
function [x,fval,exitflag,output,population,score] =
GA6(nvars,lb,ub,Generations_Data,PopulationSize_Data)
% This is an auto generated M-file from Optimization Tool.

% Start with the default options
options = gaoptimset;
% Modify options setting
options = gaoptimset(options,'Display', 'off');
options = gaoptimset(options,'OutputFcns', { [] });
options = gaoptimset(options,'Display', 'iter');
options = gaoptimset(options,'Generations', Generations_Data);
options = gaoptimset(options,'PlotFcns', {  @gaplotbestf
@gaplotbestindiv });
options = gaoptimset(options,'PopulationSize', PopulationSize_Data);
options = gaoptimset(options,'TolFun', 1*10^-12);
options = gaoptimset(options,'StallGenLimit',200);
%calling the genetic algorithm with the given inputs and desired
outputs
[x,fval,exitflag,output,population,score] = ...
ga(@NastMat_discrete,nvars,[],[],[],[],lb,ub,[],options);
```

- After that the genetic algorithm operations are started. The NastMat_discrete file is inputted as the objective function to the algorithm. When algorithm does a function evaluation, it calls NastMat_discrete function with the created population. For the discrete fiber orientation cases this populations are multiplied with the value in the *.bdf file.

```
function [z5]=NastMat_discrete(Beta)
% opening the Nastran input file which will be optimized
fid = fopen('twolayer_symm.bdf','rt');
fod = fopen('a_original2.bdf','w');

%defining the positions in the PCOMP entry for the material
%orientation angles
% FEOF is used as the
% loop control to read
% until the file ends:
posArrayIndex = 1;
while ~feof(fid)
  % FSCANF is used to
  % read in data one
  % character at a time :
  position = ftell(fid);
  data = fgetl(fid);
  if ~isempty(data)
  % the data is stored in
  % vector 'a' :'
      if length(data) > 5
          if data(1) == 'P' && data(2) == 'C' && data(3) == 'O' &&
data(4) == 'M' && data(5) == 'P';
              posArray(posArrayIndex) = position+length(data)+2;
              posArrayIndex = posArrayIndex + 1;
          end
      end
      fwrite(fod,data);
   end
   fprintf(fod,'\r\n');
end

fclose(fid);
fclose(fod);

%one of the below ones are chosen
%for single layer case
kacinciWhiteSpace=[3];
%for 4 layer case
kacinciWhiteSpace=[3 8];
numberofvariable=length(kacinciWhiteSpace);
%inputting the number of elements
numberofMaterial=736;
currentMaterialNumber = 1;
%openning and start writing the .bdf file which will be modified
during the
%optimization
t = 1;
     while currentMaterialNumber < length(posArray)

            fid2 = fopen('a_original2.bdf','rt');
            fod2 = fopen('a.bdf','w');

                 while ~feof(fid2)
              position = ftell(fid2);
              data = fgetl(fid2);
```

```
                y=1;

                positionArrayindeBuPozisyonVarMi = 0;

                for tmpIndex=1:1:length(posArray)
                    if(position == posArray(tmpIndex))
                        positionArrayindeBuPozisyonVarMi = 1;
                        break;
                    end
                end


                            if positionArrayindeBuPozisyonVarMi ~= 1
                               fwrite(fod2,data);
                               fprintf(fod2,'\r\n');

                            else% If the line is starting with PCOMP
                            j=1;
                            s = '';
                     % Read and write the file with respect to the
                     % data inputted in which the number which will be
                     % multiplied with the created population
                            whiteSpaceCount = 0;
                               while j <= length(data)
                                  c = data(j);
                                  if c == ' '
                           whiteSpaceCount = whiteSpaceCount + 1;
                           fwrite(fod2,c);
                           k = j + 1;
                           while k < length(data)
                               c = data(k);
                               if c == ' '
                                   fwrite(fod2,c);
                               else
                                   break;
                               end
                               k = k + 1;
                           end
                           j = k - 1;
                                  else

                       % prepare the numbers according to the Nastran
                       % input terminology

[r11,c11]=find(kacinciWhiteSpace==whiteSpaceCount);
                           if isempty(r11)~=1;

                               l = 0;
                               sayiString = '';
                               while l < 8
                                   c = data(j + l);
                                   if c ~= ' '
                                       if c == '+'
                                           sayiString =
strcat(sayiString,'e');
                                       end
```

```
                                    sayiString =
strcat(sayiString,c);
                                else
                                    j = j + l - 1;
                                    break;
                                end
                                l = l + 1;
                            end

                            if length(findstr(sayiString,'e')) == 0
                                j = j - 1;
                            end

                            if l == 8
                                j = j + l - 1;
                            end

                            sayi = str2double(sayiString);

                            sayi = sayi *
Beta((currentMaterialNumber-1)*numberofvariable+y);
                            y=y+1;
                            if y==numberofvariable &&
currentMaterialNumber<numberofMaterial;

currentMaterialNumber=currentMaterialNumber+1;
                            end
                            sayiStringDonusum = sprintf('%g',sayi);
                            if mod(y,2)==0;
                                sayiStringDonusum =
strrep(sayiStringDonusum, '0.', '.');
                            end
                            if ispc
                                sayiStringDonusum =
strrep(sayiStringDonusum, 'e+0', 'e+');
                            end
                            sayiStringYazilacak = '';
                            m = 1;
                            while m <= length(sayiStringDonusum)
                                c = sayiStringDonusum(m);
                                if c == 'e'
                                else
                                    sayiStringYazilacak =
strcat(sayiStringYazilacak,c);
                                end
                                m = m + 1;
                            end
                            dummyyy=str2num(sayiStringYazilacak);
                            dummy=sprintf('%0.3f',dummyyy);
                            fwrite(fod2,dummy);

                            boslukFarki = length(dummy) - (
length(sayiString) - 1);
                            if length(dummy) > length(sayiString) -
1
                                while boslukFarki > 0
                                    j = j + 1;
```

```
                                boslukFarki = boslukFarki - 1;
                            end
                        else
                            while boslukFarki < 0
                                fwrite(fod2,' ');
                                boslukFarki = boslukFarki + 1;
                            end
                        end
                        whiteSpaceCount = whiteSpaceCount + 1;
                    else
                  % Not the place for the variable number write
                  % directly
                        fwrite(fod2,c);
                    end


                            end
                            j = j + 1;
                         end
                         fprintf(fod2,'\r\n');
                    end
                end

            fclose(fid2);
            fclose(fod2);


    end
                toc
% The preparation for the input file is done, now call Nastran for
finite
% element analysis for the given variable values
system('C:\Users\sony\AppData\Roaming\MSC.Software\MD_Nastran\bin\md
nastran.exe a.bdf');

                delete a.f04
                delete a.xdb
                delete a.log



% open the results file *.f06
fid3 = fopen('a.f06','rt');
fod3 = fopen('a_original3.f06','w');

% read the output file where the objective function value is stored
within
% the output file and read that number

% FEOF is used as the
% loop control to read
% until the file ends:
posArrayIndex = 1;
while ~feof(fid3)
  % FSCANF is used to
  % read in data one
  % character at a time :
```

```
  position = ftell(fid3);
  data = fgetl(fid3);
  if ~isempty(data)
  % the data is stored in
  % vector 'a' :'
  for i=1:length(data)-6;
      c=data(i);
      if length(data) > 3
          if data(i) == 'T' && data(i+1) == 'O' && data(i+2) == 'T'
&& data(i+3) == 'A' && data(i+4) == 'L'&& data(i+5) == ' '&&
data(i+6) == 'E';
              posArray(posArrayIndex) = position;
              posArrayIndex = posArrayIndex + 1;
              tline=data;
              C = textscan(tline, '%s %s %s' );
              strainenergy=str2num(C{1,1}{5,1});
          end
      end
      fwrite(fod3,c);
  end

   end
   fprintf(fod3,'\r\n');
end

fclose(fid3);
fclose(fod3);
delete a.f06
%finally get the desired objective function value total strain
energy
z5=strainenergy;
end
```

- The genetic algorithm receives the objective function value

Appendix C

Developed MATLAB codes for the fitness function input to the genetic algorithm and curvature constraint for the continuous fiber path definitions of the composite plates

- Again to a MATLAB input file, the boundary conditions, population data and the variable number is inputted. From the following file again a options input file is called.

```
% input the number of variables
kacvariable=2;
% calls the option creating file with lower and upper boundaries,
total
% number of design cycles and population number
[x,fval] =
GA6(kacvariable,ones(1,kacvariable)*(0),ones(1,kacvariable)*90,200,3
0);
%saving the resultant graphs
h=get(0,'children');
saveas(h,'hasan','fig');
```

- Options creating input file opened and the followings are inputted

```
function [x,fval,exitflag,output,population,score] =
GA6(nvars,lb,ub,Generations_Data,PopulationSize_Data)
% This is an auto generated M-file from Optimization Tool.

% Start with the default options
options = gaoptimset;
% Modify options setting
options = gaoptimset(options,'Display', 'diagnose');
options = gaoptimset(options,'OutputFcns', { [] });
options = gaoptimset(options,'Generations', Generations_Data);
options = gaoptimset(options,'PlotFcns', {  @gaplotbestf
@gaplotbestindiv });
options = gaoptimset(options,'PopulationSize', PopulationSize_Data);
options = gaoptimset(options,'TolFun', 1*10^-12);
%mutation and selection methods are selected
options = gaoptimset(options,'MutationFcn', @mutationadaptfeasible);
```

```
options = gaoptimset(options,'SelectionFcn', @selectionroulette);
options = gaoptimset(options,'StallGenLimit',200);
[x,fval,exitflag,output,population,score] = ...
ga(@NastMat,nvars,[],[],[],[],lb,ub,@curvature,options);
```

- Genetic algorithm is started working with the given options and the boundary conditions. Curvature constraint is an input file as the following.

```
function [curve]=curvature(a,T0,T1);
% a is length of the plate
a=4;
%T0, T1 (in degrees) is the constants of the function
%converting into radians
T0=T0*pi()/180;
T1=T1*pi()/180;
%creating the functions
elementnumber = 400;
centre = midpoint;
%calculating the curvatures with respecto the Eqns. 4.1 4.2 4.3
%curvatures are calcuted with MATLAB
for elem=1:1:elementnumber;
    x = centre(elem);
    if (x <= -a/2) && (x >= -a);
        h= -(2*(tan(2*T1 - T0 - (x*(2*T0 - 2*T1))/a)^2 +
1)*((tan(2*T1 - T0 - (x*(2*T0 - 2*T1))/a) + 1)^2)^(1/2)*(T0 -
T1))/(a*(tan(2*T1 - T0 - (x*(2*T0 - 2*T1))/a) + 1)^4);
    elseif (x > -a/2) && (x <= 0);
        h= (((cos(T0 + (x*(2*T0 - 2*T1))/a)^2 - 1)*(2*T0 -
2*T1))/(a*cos(T0 + (x*(2*T0 - 2*T1))/a)^2*(cos(T0) + log(cos(T0 +
(x*(2*T0 - 2*T1))/a)))) - (2*T0 - 2*T1)/(a*(cos(T0) + log(cos(T0 +
(x*(2*T0 - 2*T1))/a)))) + ((cos(T0 + (x*(2*T0 - 2*T1))/a)^2 -
1)*(2*T0 - 2*T1))/(a*cos(T0 + (x*(2*T0 - 2*T1))/a)^2*(cos(T0) +
log(cos(T0 + (x*(2*T0 - 2*T1))/a)))^2))/((tan(T0 + (x*(2*T0 -
2*T1))/a)/(cos(T0) + log(cos(T0 + (x*(2*T0 - 2*T1))/a))) -
1)^2)^(3/2);
    elseif (x > 0) && (x <= a/2);
        h= -(((cos(T0 - (x*(2*T0 - 2*T1))/a)^2 - 1)*(2*T0 -
2*T1))/(a*cos(T0 - (x*(2*T0 - 2*T1))/a)^2*(cos(T0) + log(cos(T0 -
(x*(2*T0 - 2*T1))/a)))) - (2*T0 - 2*T1)/(a*(cos(T0) + log(cos(T0 -
(x*(2*T0 - 2*T1))/a)))) + ((cos(T0 - (x*(2*T0 - 2*T1))/a)^2 -
1)*(2*T0 - 2*T1))/(a*cos(T0 - (x*(2*T0 - 2*T1))/a)^2*(cos(T0) +
log(cos(T0 - (x*(2*T0 - 2*T1))/a)))^2))/((tan(T0 - (x*(2*T0 -
2*T1))/a)/(cos(T0) + log(cos(T0 - (x*(2*T0 - 2*T1))/a))) -
1)^2)^(3/2);
    else (x > a/2) && (x <= a);
        h= (2*(tan(2*T1 - T0 + (x*(2*T0 - 2*T1))/a)^2 +
1)*((tan(2*T1 - T0 + (x*(2*T0 - 2*T1))/a) + 1)^2)^(1/2)*(T0 -
T1))/(a*(tan(2*T1 - T0 + (x*(2*T0 - 2*T1))/a) + 1)^4);
    end
    curve(elem,1) = h;
end
end
```

- After the constraints the main code is called for the function evaluation.

- Objective function value is obtained from Nastran finite element solver. Now, MATLAB genetic algorithm is using the value.

```
function [z5]=NastMat(a)
%opening the desired case study input file
fid = fopen('twolayer_symm.bdf','rt');
fod = fopen('a_original2.bdf','w');

% defining the positions for the places of the variables in the
PCOMP entry
% FEOF is used as the
% loop control to read
% until the file ends:
posArrayIndex = 1;
while ~feof(fid)
  % FSCANF is used to
  % read in data one
  % character at a time :
  position = ftell(fid);
  data = fgetl(fid);
  if ~isempty(data)
  % the data is stored in
  % vector 'a' :'
      if length(data) > 5
          if data(1) == 'P' && data(2) == 'C' && data(3) == 'O' &&
data(4) == 'M' && data(5) == 'P';
              posArray(posArrayIndex) = position+length(data)+2;
              posArrayIndex = posArrayIndex + 1;
          end
      end
      fwrite(fod,data);
   end
   fprintf(fod,'\r\n');
end

fclose(fid);
fclose(fod);

% followings are selected with respect to
% single layer case studies
kacinciWhiteSpace = [3];
% multilayer case studies
kacinciWhiteSpace = [3 7];
numberofvariable=length(kacinciWhiteSpace);

% material orientation angles with the fiber path definitions are
created
```

```
% 0 as the axis rotation angle, 4 is the length of the plate, a(1)
and a(2)
% are the variables which are T0 and T1
  [teta]=fiberpath(0,4,a(1),a(2));
% resort the values to write with respect to the code
 counter=1;
 for i=1:size(teta,1);
     for j=1:size(teta,2);
         newteta(counter)=teta(i,j);
         counter=counter+1;
     end
 end
 clear counter i j

numberofMaterial=736;
currentMaterialNumber = 1;
t = 1;
     while currentMaterialNumber < length(posArray)

            fid2 = fopen('a_original2.bdf','rt');
            fod2 = fopen('a.bdf','w');

                 while ~feof(fid2)
              position = ftell(fid2);
              data = fgetl(fid2);
              y=1;

              positionArrayindeBuPozisyonVarMi = 0;

              for tmpIndex=1:1:length(posArray)
                  if(position == posArray(tmpIndex))
                      positionArrayindeBuPozisyonVarMi = 1;
                      break;
                  end
              end



                            if positionArrayindeBuPozisyonVarMi ~= 1
                               fwrite(fod2,data);
                               fprintf(fod2,'\r\n');

                            else% If the line is starting with PCOMP
                            j=1;
                            s = '';
                     % Read and write the file with respect to the
                     % data inputted in which the number which will be
                     % multiplied with the created population
                            whiteSpaceCount = 0;
                               while j <= length(data)
                                  c = data(j);
                                  if c == ' '
                          whiteSpaceCount = whiteSpaceCount + 1;
                          fwrite(fod2,c);
                          k = j + 1;
```

```
                    while k < length(data)
                        c = data(k);
                        if c == ' '
                            fwrite(fod2,c);
                        else
                            break;
                        end
                        k = k + 1;
                    end
                    j = k - 1;
                           else

                % prepare the numbers according to the Nastran
                % input terminology

                    fwrite(fod2,c);
                    k = j + 1;
                    while k < length(data)
                        c = data(k);
                        if c == ' '
                            fwrite(fod2,c);
                        else
                            break;
                        end
                        k = k + 1;
                    end
                    j = k - 1;

                                if isequal(data(j),' ')==1 &&
j>=j1;
                                    if
exist('notifier','var')==1 && notifier==1;

whiteSpaceCount=whiteSpaceCount;
                                        notifier=0;
                                    else
                                        whiteSpaceCount =
whiteSpaceCount + 1;
                                    end

                                end

                else

% write the value into the Nastran input file
                    if y<=length(kacinciWhiteSpace)&&
whiteSpaceCount == kacinciWhiteSpace(y);

                        l = 0;
                        sayiString = '';
                        while l < 8 && j+l<length(data);
                            c = data(j + l);
                            if c ~= ' '
                                if c == '+'
                                    sayiString =
strcat(sayiString,'e');
                                end
```

```
                                sayiString =
strcat(sayiString,c);
                            else
                                j = j + l - 1;
                                break;
                            end
                            l = l + 1;
                        end

                        if length(findstr(sayiString,'e')) == 0
                            j = j - 1;
                        end

                        if l == 8
                            j = j + l - 1;
                        end

                        sayi = str2double(sayiString);


                        sayi=newteta((currentMaterialNumber-
1)*numberofvariable+y);
                        y=y+1;

                        if y==numberofvariable+1 &&
currentMaterialNumber<numberofMaterial;   %

currentMaterialNumber=currentMaterialNumber+1;
                        end
                        sayiStringDonusum = sprintf('%g',sayi);
                        if mod(y,2)==0;
                        end
                        if ispc
                            sayiStringDonusum =
strrep(sayiStringDonusum, 'e+0', 'e+');
                        end
                        sayiStringYazilacak = '';
                        m = 1;
                        while m <= length(sayiStringDonusum)
                            c = sayiStringDonusum(m);
                            if c == 'e'
                            else
                                sayiStringYazilacak =
strcat(sayiStringYazilacak,c);
                            end
                            m = m + 1;
                        end
                        dummyyy=str2num(sayiStringYazilacak);
                        dummy=sprintf('%0.3f',dummyyy);
                        fwrite(fod2,dummy);

                        boslukFarki = length(dummy) - (
length(sayiString) - 1);
                        if length(dummy) > length(sayiString) -
1
                            while boslukFarki > 0
                                j = j + 1;
```

```
                        boslukFarki = boslukFarki - 1;
                    end
                else
                    while boslukFarki < 0
                        fwrite(fod2,' ');
                        boslukFarki = boslukFarki + 1;
                    end
                end
                whiteSpaceCount = whiteSpaceCount + 1;
                notifier=1;
            else
                % not the place to write the number,
write
                % directly
                fwrite(fod2,c);
            end


                    end
                    j = j + 1;
                 end
                 fprintf(fod2,'\r\n');
            end
        end

    fclose(fid2);
    fclose(fod2);


     end
% call Nastran for analyzing the input file

system('C:\Users\sony\AppData\Roaming\MSC.Software\MD_Nastran\bin\md
nastran.exe a.bdf');

% Matlab waits until the finite element analysis finish


                delete a.f04
                delete a.xdb
                delete a.log

%% This part is used for reading the total strain energy of the
particular
%% case studies

% open the output file to read the objective value
fid3 = fopen('a.f06','rt');
fod3 = fopen('a_original3.f06','w');

% FEOF is used as the
% loop control to read
% until the file ends:
posArrayIndex = 1;
while ~feof(fid3)
  % FSCANF is used to
```

```
  % read in data one
  % character at a time :
  position = ftell(fid3);
  data = fgetl(fid3);
  if ~isempty(data)
  % the data is stored in
  % vector 'a' :'
  for i=1:length(data)-6;
      c=data(i);
      if length(data) > 3
          if data(i) == 'T' && data(i+1) == 'O' && data(i+2) == 'T'
&& data(i+3) == 'A' && data(i+4) == 'L'&& data(i+5) == ' '&&
data(i+6) == 'E';
              posArray(posArrayIndex) = position;
              posArrayIndex = posArrayIndex + 1;
              tline=data;
              C = textscan(tline, '%s %s %s' );
              strainenergy=str2num(C{1,1}{5,1});
          end
      end
      fwrite(fod3,c);
  end

   end
   fprintf(fod3,'\r\n');
end

fclose(fid3);
fclose(fod3);

%% This subroutine is used to read the eigenvalue for the buckling
problem
%% for the particular cases
[out]=eigenvalue(1);
delete a.f06
%%% output for the strain energy case studies
z5=strainenergy;
%%% output for the buckling load case studies
z5=(-1)*(out);
end
```

- The fiber path subroutine is given below in detail

```
function [theta]=fiberpath(phi,a,T0,T1)
elementnumber = 400;
%call the midpoint function for the mid point coordinates as
function
%centre
centre = midpoint;
%for the orientations formulation used in reference [3], for
%+theta/-theta orientation
%theta(elem,2) is used for the multilayer case studies
for elem=1:1:elementnumber
```

```
    x = centre(elem);
    if (x <= -a/2) && (x >= -a);
        theta(elem,1) = phi + 2/a*(T1-T0)*x + T0 - 2*(T0-T1);
        theta(elem,2) = -theta(elem,1);
    elseif (x > -a/2) && (x <= 0);
        theta(elem,1) = phi + 2/a*(T1-T0)*x + T0;
        theta(elem,2) = -theta(elem,1);
    elseif (x > 0) && (x <= a/2);
        theta(elem,1) = phi + 2/a*(T1-T0)*x + T0;
        theta(elem,2) = -theta(elem,1);
    else (x > a/2) && (x <= a);
        theta(elem,1) = phi + 2/a*(T0-T1)*x + T0 - 2*(T0-T1);
        theta(elem,2) = -theta(elem,1);
    end
end
% save the orientations
save ('theta.mat','theta');
end
```

- The midpoint subroutine is as follows

```
function [centre] = midpoint(mesh,grid)
% calling the function to read the CQUAD4 and GRID entries in the
Nastran
% input file, which writes the data in the variables.mat file
cquad4
load variables.mat
elementnumber = 400;
% calculating the x axis midpoint locations for each element
for elem = 1:1:elementnumber
    point1 = mesh(elem,3);
    point2 = mesh(elem,4);
    point3 = mesh(elem,5);
    point4 = mesh(elem,6);

    centre(elem) = (grid(point1,2) + grid(point2,2) + grid(point3,2)
+ grid(point4,2))/4;
end
end
```

- The subroutine for reading the element and geometry data, cquad4 is as follows

```
% input file for the case study
inputname='twolayer_symm.bdf';
% for each run delete the variables
delete variables.mat;
if exist('variables.mat','file')==0;
```

```
fid1 = fopen(eval('inputname'),'rt');
fod1 = fopen('a_originall.bdf','w');

% find the CQUAD4 data entries in the *.bdf file for the case study
% FEOF is used as the
% loop control to read
% until the file ends:
clear posArray;
posArrayIndex = 1;
while ~feof(fid1)
  % FSCANF is used to
  % read in data one
  % character at a time :
  position = ftell(fid1);
  data = fgetl(fid1);
  if ~isempty(data)
  % the data is stored in
  % vector 'a' :'
      if length(data) > 4
          if data(1) == 'C' && data(2) == 'Q' && data(3) == 'U' &&
data(4) == 'A' && data(5) == 'D' && data(6) == '4'
              posArray(posArrayIndex) = position;
              posArrayIndex = posArrayIndex + 1;
          end
      end
      fwrite(fod1,data);
   end
   fprintf(fod1,'\r\n');
end

fclose(fid1);
fclose(fod1);

kacinciWhiteSpace = 2;

    fid2 = fopen('a_originall.bdf','rt');
    fod2 = fopen('a.txt','w');

    while ~feof(fid2)
      position = ftell(fid2);
      datal = fgetl(fid2);

      positionArrayindeBuPozisyonVarMi = 0;

      for tmpIndex=1:1:length(posArray)
          if(position == posArray(tmpIndex))
              positionArrayindeBuPozisyonVarMi = 1;
              break;
          end
      end


      if positionArrayindeBuPozisyonVarMi ~= 1
          continue
      else
```

```
             fwrite(fod2,data1);
            fprintf(fod2,'\r\n');

       end
     end

     fclose(fid2);
     fclose(fod2);


         fid3=fopen('a.txt');
         B = textscan(fid3, ' %s %f %f %f %f %f %f ');
         fclose(fid3);

         clear sayi1
         sayi1(:,1) = B{1,2};
         sayi1(:,2)= B{1,3};
         sayi1(:,3) = B{1,4};
         sayi1(:,4) = B{1,5};
         sayi1(:,5) = B{1,6};
         sayi1(:,6) = B{1,7};


         counter=1;
         for ih=1:size(sayi1,1);
             if isnan(sayi1(ih,1))==1;
                 yer(counter)=ih;
                 counter=counter+1;
             end
         end
         if exist('yer','var')==1;
             yer=yer';
             sayi1(yer,:)=[];
         end

         mesh=[sayi1];
% CQUAD4 Data are read and stored
% Now read the GRID data
                     fid4 = fopen(eval('inputname'),'rt');
                     fod4 = fopen('a_original.txt','w');

                     posArrayIndex = 1;
                     while ~feof(fid4)
                       % FSCANF is used to
                       % read in data one
                       % character at a time :
                       position = ftell(fid4);
                       data = fgetl(fid4);
                       if ~isempty(data)
                       % the data is stored in
                       % vector 'a' :
                           if length(data) > 3
                               if data(1) == 'G' && data(2) == 'R' &&
data(3) == 'I' && data(4) == 'D'
                                   posArray(posArrayIndex) =
position;
```

```
                        posArrayIndex = posArrayIndex + 1;
                    end
                end
                fwrite(fod4,data);
            end
            fprintf(fod4,'\r\n');
        end

        fclose(fid4);
        fclose(fod4);

        kacinciWhiteSpace = 2;

            fid5 = fopen('a_original.txt','rt');
            fod5 = fopen('grid.txt','w');

            while ~feof(fid5)
              position = ftell(fid5);
              data = fgetl(fid5);

              positionArrayindeBuPozisyonVarMi = 0;

              for tmpIndex=1:1:length(posArray)
                  if(position == posArray(tmpIndex))
                      positionArrayindeBuPozisyonVarMi =
1;

                      break;
                  end
              end


              if positionArrayindeBuPozisyonVarMi ~= 1
                  continue
              else

                     fwrite(fod5,data);
                    fprintf(fod5,'\r\n');

              end
            end

            fclose(fid5);
            fclose(fod5);

fid6=fopen('grid.txt');
B = textscan(fid6, '%s %f %f %f %f');
fclose(fid6);



clear sayi1
sayi1(:,1) = B{1,2};
sayi1(:,2) = B{1,3};
sayi1(:,3) = B{1,4};
sayi1(:,4) = B{1,5};
```

```
        grid=[sayi1];
% GRID data is read and stored
        kacElement=length(mesh);
        save('variables.mat','mesh','grid','kacElement');
end
```

- With the subroutines given above the fiber orientations are calculated.
- The eigenvalue subroutine, which reads the eigenvalue number in the Nastran output file (*.f06), is given in detail below

```
function [out] = eigenvalue(number)
% number is the desired number of eigenvalues
fid1=fopen('a.f06','rt');
fod1 = fopen('eigenvalues.txt','w');
posArrayIndex = 1;
count=0;
% The place where the objective function value is written in the
Nastran
% output file is found and the eigenvalue is read from the Nastran
output
% file
while ~feof(fid1)
  % FSCANF is used to
  % read in data one
  % character at a time :
  position = ftell(fid1);
  data = fgetl(fid1);
  if ~isempty(data)
  % the data is stored in
  % vector 'a' :'
      if strcmp(data,'   MODE    EXTRACTION      EIGENVALUE
RADIANS             CYCLES            GENERALIZED
GENERALIZED')==1 || count>=1;
          if count==0;
                count=1;
          else
              count=count+1;
          end
          if count>2
              if strfind(data,'1     ')==1
                  count=0;
                  continue
              end
          fwrite(fod1,data);
          fprintf(fod1,'\r\n');

          end
              posArray(posArrayIndex) = position;
              posArrayIndex = posArrayIndex + 1;
      end
```

```
    end
end
fclose(fid1);
fclose(fod1);

dummy=textread('eigenvalues.txt');
eigenvalues=dummy(:,3);
% subroutine outputs the desired values
out=eigenvalues(1:number);
```