

Marwan Maurizio CHEDID

Optimisation of Composite Materials using a Multilevel
Decomposition Approach

School of Engineering

MSc Thesis by Research

School of Engineering

MSc Thesis by Research

2009

Marwan Maurizio CHEDID

Optimisation of Composite Materials using a Multilevel
Decomposition Approach

Supervisors: Dr K. Hughes / Prof. R. Vignjevic

Academic Year 2008 to 2009

Abstract

The design optimisation of thick composite components requires dealing with large number of design variables, highly non-linear equations and huge computational demand.

A multi-level decomposition process has been developed to optimise these elements, investigating the potentialities of pre-existing approaches of the bioengineering field of hip and femoral implants.

The strength of exploiting a multi-level decomposition process is related to the flexibility of choosing at each level specific optimisation methods which modulate according to the design variables appointed and to the difficulty level of the application. Specifically the proposed framework is developed through two main applications, Basic and Intermediate Example, with increasing difficulty.

The Basic Example uses Graphical Optimisation tools selecting the typical design variables such as thicknesses and orientation angles. Then, the Intermediate Example considers numerical optimisation conditions with Kuhn- Tucker and Lagrange Multipliers and again Graphical Methods, but using lamination parameters. The optimisation analyses and classical lamination theory calculations are performed through a Matlab code linked with an external solver, Nastran, to perform the numerical structural analysis.

These two applications, designed as 2-level decomposition approach, do not consider any inter-laminar effect and out-of-plane loading that it will be taken into account in the proposed final frame work application, Advanced Example.

A single level optimisation analysis is performed by commercial software optimiser, Nastran Sol 200, as alternative mean to validate the two numerical applications.

Results show the potentiality of this technique, mainly related to the capability to control each design cycle, though the full understanding will come after a numerical application of the advanced example.

In fact, adding within the current Matlab code genetic algorithms, sequential linear programming, and gradient based methods capable to deal with out-of-plane loadings, quadratic failure criteria taking into account the through-thickness stress effects is still a complex direction of enhancement recommended to be explored.

Table of Contents

Abstract	3
List of Symbols.....	7
List of Tables.....	10
List of Figures	11
Acknowledgment	14
1 Introduction	15
1.1 General background	15
1.2 Research Objectives	18
1.3 Multi-Level Decomposition	19
1.4 Research Methodology	21
1.5 Software Methodologies	27
1.6 Summary	29
2 Literature Review.....	31
2.1 Introduction	31
2.2 Brief Review of Structural Optimisation and Composite Materials.....	31
2.3 The Three Axes of Complexity.....	34
2.4 Bioengineering Applications.....	38
2.5 Decomposition and Multi-Level Optimisation	40
2.6 Parallel Computation.....	44
2.7 Finite Element Modelling of Laminated Structures	48
2.8 Inter-laminar Stresses: Analytical and Numerical Methods.....	55
2.9 Summary	57
3 Laminated Composite Materials	59
3.1 General	59
3.2 Classification of Composite Materials.....	59
3.3 Fibre-Reinforced Composite Materials	62
3.4 Fibre-Reinforced Composite Materials: Properties and Applications.....	62
3.5 Laminated Composite Definition	63
3.6 Laminated Composites Structural Analysis.....	65
3.6.1 Classical Analysis	65
3.6.2 Limitations of Classical Laminate Theory	82

3.6.3	Finite Element Analysis	83
3.6.4	Properties of Laminates Made of Sub-Laminates	84
3.6.5	Failure Criteria for Laminated Composites	85
3.7	Summary	86
4	Optimisation.....	88
4.1	General.....	88
4.2	Optimisation Fundamentals.....	89
4.3	Optimisation Classification.....	92
4.4	System Decomposition and Optimisation.....	93
4.4.1	Graphical Solution	94
4.4.2	Lamination Parameters	95
4.4.3	Numerical Techniques.....	102
4.4.4	Sequential Linear Programming.....	110
4.4.5	Genetic Algorithm	112
4.5	Summary	114
5	Basic Example.....	116
5.1	Optimisation Procedure	116
5.1.1	Analytical Model (A).....	121
5.1.2	Graphical Optimiser (B).....	130
5.1.3	Numerical Model: CLT & FEM (C).....	151
5.1.4	Validation Test: Constraint Checks (V)	170
5.1.5	Convergence Test (CT).....	176
5.1.6	Optimum Design (OD)	179
5.2	Numerical Application Results and Discussion	184
5.3	Summary	199
6	Intermediate Example	201
6.1	Optimisation Procedure	202
6.1.1	Analytical Model (A)	206
6.1.2	Numerical and Graphical Optimiser (B)	213
6.2	Basic Example and Intermediate Example: Similarities and Differences	236
6.3	Numerical Application Results and Discussion	244
6.4	Summary	267
7	Commercial Optimiser Comparison.....	269

7.1	Optimiser Description	270
7.2	“Commercial Optimiser” VS “2-Level Decomposition Technique”	273
7.3	RUN1 and RUN2: Results and Discussion	277
7.4	Summary	293
8	Conclusions	295
8.1	Objectives and Outcomes	295
8.2	Future Work	298
8.2.1	Advanced Example	299
	References	308
	Appendix A	317
	Appendix B	319
	Appendix C	321

List of Symbols

PSE = Principle Structural Elements
 ALCAS = Advanced Low-Cost Aircraft Structure
 MLG = main landing gear
 MD = Multi-level Decomposition
 2-LD = A two-level decomposition
 3-LD = A three-level decomposition
 MA = Mathematical Programming
 CLT=Classical Lamination Theory
 FEM = Finite Element Model
 PP = Parallel Programming
 DOF = degrees of freedom
 2D = bi-dimensional
 3D = three-dimensional
 NP = non-linear problem
 DSM=Design Structure Matrix
 PLATON = Parallel Adaptive Techniques for Optimisation
 CTL = Component Template Library
 CORBA = Common Object Request Broker Architecture
 FSDT = first-order shear deformation plate theory
 TSDT = third-order shear deformation plate theory
 ISSCT = Inter-laminar Shear Stress Continuity Theory
 FRC = Fibre Re-inforced Composites
 [] =Matrix Notation
 $[\vartheta_1 / \vartheta_2 / \dots / \vartheta_N]$ = Laminate Orientation Angles Configuration
 $[\vartheta_1 / \vartheta_2 / \dots / \vartheta_N]_S$ = Symmetric Laminate Orientation Angles Configuration
 1,2,3=Laminate Fibre Direction (1) and its Orthogonal Direction (2) in-the ply plane and out-of-the ply laminate plane (3)
 x,y,z= laminated General Axes of the laminated medium
 $\sum$ =Summatory Operator
 $\varepsilon_x, \varepsilon_y, \varepsilon_z$ = Normal Strain along X, Y, and Z axes
 $\gamma_{xy}, \gamma_{yz}, \gamma_{zx}$ =Shear Strains in the X,Y, and direction
 [C]= Material Stiffness Matrix
 σ =Vector of Stresses
 ε =Vector of Strain
 $\sigma = C\varepsilon$ = Hooke's Law
 U = Energy Density
 $\frac{\partial}{\partial g}$ =Partial Derivative along "g"
 E=Elastic Modulus
 G=Shear Modulus
 ν =Poisson's Ratio
 $\sigma_x, \sigma_y, \tau_{xy}$ =In-plane Stresses
 $\sigma_z, \tau_{yz}, \tau_{zx}$ = Out-of-Plane Stresses
 [Q]= Reduced Material Stiffness Matrix
 K = Number of Layers

N_x, N_y = Normal Loadings, force over length
 N_{xy} = Shear Loading, force over length
 ϵ = mid-plane strains
 N = number of layers
 Z_k = Through the Thickness Locations
 t_k = Thickness of the k layer
 $[A]$ = extensional material stiffness matrix or in-plane stiffness matrix
 $E_1, E_2, E_3, G_{12}, G_{23}, G_{31}$ = Three Elastic and Three Shear Moduli in Fibre Axes
 $[T]$ = Transformation Matrix
 $m = \cos \theta$
 $n = \sin \theta$
 $[]^{-1}$ = Inverse Matrix
 $[\bar{Q}]$ = Transformed Reduced Stiffness Matrix
 U_i = Material Invariants
 n_M = Plies of Sub-laminate M
 n_N = Plies of Sub-laminate N
 n_{MN} = Plies of a Laminate made of two sub-laminates M and N
 X = Design Variable Vector
 $f(x)$ = Objective Function in term of x design variables
 $h(x)$ = Equality Constraint Function in term of x design variables
 $g(x)$ = Inequality Constraint Function in term of x design variables
 x^L = Design Variable Lower Bound
 x^U = Design Variable Upper Bound
 $V_1^* V_3^*$ = Lamination Parameters
 h = Laminate Thickness
 v_i = Volume Fraction
 X^* = Optimum Value
 ∇ = Gradient Operator
 FOC = First Order Condition
 $[H]$ = Hessian Matrix
 SOC = Second Order Condition
 λ = Lagrange Multiplier for Equality Problem
 $[]^T$ = Traspose Matrix
 β = Lagrange Multiplier for Inequality Problems
 SLP = Sequential Linear Programming
 OD = Optimum Design
 $L1$ = Level 1
 $L2$ = Level 2
 $(\epsilon_x)_{MAX}$ = Normal Strain Function to be maximised
 $(\gamma_{xy})_{min}$ = Shear Strain Function to be minimised
 A = Analytical Model Box
 B = Numerical and Graphical Optimiser Box
 C = Numerical Model Box
 V = Validation Test Box
 P = Design Parameters

RUN1 = Basic Example Numerical Application

RUN2 = Intermediate Example Numerical Application

$(RI_N)_k$ = Normal Reliability Index at every iteration k

$(RI_S)_k$ = Shear Reliability Index at every iteration k

$(EI_1)_k$ = Level 1 Effectiveness Index at every iteration k

$(EI_2)_k$ = Level 2 Effectiveness Index at every iteration k

List of Tables

TABLE 1-1 - BASIC EXAMPLE 2-LEVEL DECOMPOSITION SUMMARY	24
TABLE 1-2 - INTERMEDIATE EXAMPLE 2-LEVEL DECOMPOSITION SUMMARY	25
TABLE 1-3 - ADVANCED EXAMPLE 3-LEVEL DECOMPOSITION SUMMARY	26
TABLE 5-1 "BASIC EXAMPLE": OPTIMISATION PROBLEM DEFINITION	121
TABLE 5-2 BASIC EXAMPLE: FUNCTION M-FILE LABELS AT LEVEL1 AND LEVEL2, 2D PLOTS	132
TABLE 5-3 BASIC EXAMPLE: FUNCTION M-FILE LABELS AT LEVEL2, 3D PLOTS	133
TABLE 5-4 - DIFFERENT STRUCTURAL IDEALISATION BETWEEN FIRST AND FINAL BOX C	153
TABLE 5-5 – LAMINATE CONFIGURATIONS AT LEVEL 1 AND LEVEL 2, 1 ST ITERATION	160
TABLE 5-6 – LAMINATE EFFECTIVE MATERIAL PROPERTIES	165
TABLE 5-7 – ITERATION1, LEVEL 1 AND LEVEL 2 DESIGN VARIABLES	185
TABLE 5-8 – ITERATION2, LEVEL 1 AND LEVEL 2 DESIGN VARIABLES	186
TABLE 5-9 – ITERATION3, LEVEL 1 AND LEVEL 2 DESIGN VARIABLES	187
TABLE 5-10 – ITERATION4, LEVEL 1 AND LEVEL 2 DESIGN VARIABLES	188
TABLE 5-11 – ITERATION5, LEVEL 1 AND LEVEL 2 DESIGN VARIABLES	189
TABLE 5-12 – ITERATION6, LEVEL 1 AND LEVEL 2 DESIGN VARIABLES	190
TABLE 5-13 – RUN1, LEVEL1 AND LEVEL2 DESIGN HISTORY	192
TABLE 5-14 – RUN1 RELIABILITY AND EFFECTIVENESS INDEXES, FINAL ITERATION (K=5 AND K=6)	198
TABLE 6-1 OPTIMISATION PROBLEM DEFINITION: INTERMEDIATE EXAMPLE	206
TABLE 6-2 - INTERMEDIATE EXAMPLE: LEVEL1 AND LEVEL2, MATLAB 2D PLOTS	214
TABLE 6-3 - INTERMEDIATE EXAMPLE: FUNCTION M-FILE LABELS AT LEVEL2, 3D PLOTS	214
TABLE 6-4 – BASIC EXAMPLE: COUPLING QUANTITIES STRUCTURE	237
TABLE 6-5 – INTERMEDIATE EXAMPLE: COUPLING QUANTITIES STRUCTURE	238
TABLE 6-6 – ITERATION1, LEVEL 1 AND LEVEL 2 DESIGN VARIABLES AND COUPLING QUANTITIES, INTERMEDIATE EXAMPLE	246
TABLE 6-7 – ITERATION2, LEVEL 1 AND LEVEL 2 DESIGN VARIABLES AND COUPLING QUANTITIES, INTERMEDIATE EXAMPLE	248
TABLE 6-8 – ITERATION3, LEVEL 1 AND LEVEL 2 DESIGN VARIABLES AND COUPLING QUANTITIES, INTERMEDIATE EXAMPLE	250
TABLE 6-9 – ITERATION4, LEVEL 1 AND LEVEL 2 DESIGN VARIABLES AND COUPLING QUANTITIES, INTERMEDIATE EXAMPLE	252
TABLE 6-10 – ITERATION5, LEVEL 1 AND LEVEL 2 DESIGN VARIABLES AND COUPLING QUANTITIES, INTERMEDIATE EXAMPLE	254
TABLE 6-11 – RUN2, LEVEL1 AND LEVEL2 DESIGN HISTORY	264
TABLE 6-12 – RUN2 RELIABILITY AND EFFECTIVENESS INDEXES, FINAL ITERATION (K=4)	264
TABLE 7-1 – RUN1 AND RUN2 INITIAL DATA	279
TABLE 7-2 – "NASTRAN SOL200" AND "2-LEVEL DECOMPOSITION": OPTIMUM DESIGN RUN1 AND RUN2 REFERENCE	290
TABLE 7-3 – NASTRAN SOL200 AND BASIC EXAMPLE (BE) OPTIMUM LAMINATE COMPARISON: RUN1	290
TABLE 7-4 – NASTRAN SOL200 AND INTERMEDIATE EXAMPLE (IE) OPTIMUM LAMINATE COMPARISON: RUN2	291
TABLE 7-5 – NASTRAN SOL200 AND BASIC EXAMPLE: RUN1 TOTAL OPTIMUM LAMINATE THICKNESS	291
TABLE 7-6 – NASTRAN SOL200 AND BASIC EXAMPLE: RUN2 TOTAL OPTIMUM LAMINATE THICKNESS	292
TABLE 8-1 – MULTI-LEVEL DECOMPOSITION LEVEL DESCRIPTION	304
TABLE 8-2 – ADVANCED EXAMPLE: COUPLING QUANTITIES STRUCTURE	305

List of Figures

FIGURE 1-1 - CRITICAL COMPONENTS OF THE ALCAS LATERAL WING BOX ASSEMBLY [3]	17
FIGURE 1-2 - A GENERAL MULTI-LEVEL DECOMPOSITION STRUCTURE [6]	21
FIGURE 1-3 - MULTI-LEVEL DECOMPOSITION STRUCTURE	22
FIGURE 1-4 - FLOW DIAGRAM OF THE MAIN TOOLS FOR A REQUIRED GENERIC LEVEL	28
FIGURE 2-1 - EVOLUTION IN ANALYSIS MODEL COMPLEXITY, CURRENT=2002/RECENT PAST=MID-90S/15-20 YEARS AGO= 80S - [17]	33
FIGURE 2-2 – EVOLUTION OF MODEL FROM SIMPLE MODELS TO COMPLEX 3D APPLICATIONS [5]	35
FIGURE 2-3 – EVOLUTION OF MODEL ANALYSIS TYPES [5]	36
FIGURE 2-4 - EVOLUTION IN OPTIMISATION COMPLEXITY FOR COMPOSITE PROBLEMS [5]	37
FIGURE 2-5 - THE THREE AXES OF COMPLEXITY [5]	38
FIGURE 2-6 - TRADE-OFFS IN OPTIMISATION OF COMPOSITE LAMINATE STRUCTURES [5]	39
FIGURE 2-7 - MULTI-LEVEL DECOMPOSITION: HIERARCHIC SYSTEM [26]	41
FIGURE 2-8 – MULTI-LEVEL DECOMPOSITION: NON-HIERARCHIC SYSTEM [26]	42
FIGURE 2-9 – MULTI-LEVEL DECOMPOSITION: HIERARCHIC AND NON HIERARCHIC SYSTEMS [26]	43
FIGURE 2-10 – A DSM EXAMPLE [28]	44
FIGURE 2-11 – PLATON ABSTRACT VIEW OF OPTIMISATION [36]	47
FIGURE 2-12 – THE PLATON SYSTEM [36]	48
FIGURE 2-13 – STIFFENED PANEL [37]	49
FIGURE 2-14 – STIFFENED PANEL 2D FEM [37]	49
FIGURE 2-15 – BIOENGINEERING APPLICATION: FEMORAL SCHEMATIC MODEL [4]	50
FIGURE 2-16 – FEMORAL 3D FEM [38]	50
FIGURE 2-17 – AEROSPACE APPLICATION: SIDE STAY BRACKET 3D FEM, THESIS OBJECT [3]	51
FIGURE 2-18 – HELICOPTER HINGELESS HUB SYSTEM AND PARTS BEING ANALYSED [40]	52
FIGURE 2-19 – EIGHT DIFFERENT COMPONENTS OF THE PART UNDER ANALYSIS [40]	52
FIGURE 2-20 – FINITE ELEMENT MODEL OF THE STRUCTURE: 3-D ELEMENTS [40]	53
FIGURE 5-1 – MULTILEVEL DECOMPOSITION STRUCTURE: BASIC EXAMPLE TWO-LEVEL DECOMPOSITION	118
FIGURE 5-2 MULTI-LEVEL DECOMPOSITION SINGLE LEVEL STRUCTURE	119
FIGURE 5-3 BASIC EXAMPLE: HALF LAMINATE ORIENTATION GROUP NUMBERING CONVENTION	121
FIGURE 5-4 BASIC EXAMPLE, GRAPHICAL OPTIMISER: MATLAB STRUCTURE	132
FIGURE 5-5 $\mathcal{G}_2$ INITIAL VALUE PARAMETRIC ANALYSIS, $\mathcal{G}_2 > 45^\circ$	135
FIGURE 5-6 $\mathcal{G}_2$ INITIAL VALUE PARAMETRIC ANALYSIS, $\mathcal{G}_2 < 45^\circ$	136
FIGURE 5-7 - BASIC EXAMPLE: DESIGN SPACE LEVEL 1	137
FIGURE 5-8 - BASIC EXAMPLE: ZOOMED DESIGN SPACE, "G1" CONSTRAINT FUNCTION - LEVEL 1	138
FIGURE 5-9 - BASIC EXAMPLE: "F1" OBJECTIVE FUNCTION VISUAL INSPECTION - LEVEL 1	139
FIGURE 5-10 - BASIC EXAMPLE: DESIGN SPACE LEVEL 2	140
FIGURE 5-11 - BASIC EXAMPLE: "G2" INEQUALITY CONSTRAINT DATA CURSOR "LEVELS" – L2	141
FIGURE 5-12 - BASIC EXAMPLE: DESIGN SPACE LEVEL 2 WITH NO G2 LINES	142
FIGURE 5-13 - BASIC EXAMPLE: 3D OBJECTIVE FUNCTION PLOT, PARTICULAR CASE - LEVEL 2	143
FIGURE 5-14 - BASIC EXAMPLE: 3D CONSTRAINT FUNCTION PLOT, PARTICULAR CASE - LEVEL 2	143
FIGURE 5-15- BASIC EXAMPLE: 3D CONSTRAINT FUNCTION PLOT XZ VIEW, PARTICULAR CASE LEVEL 2	144
FIGURE 5-16 - BASIC EXAMPLE: FEASIBLE DOMAIN , LEVEL 2	145
FIGURE 5-17 – BOX C: FIRST VERSION	151
FIGURE 5-18 - BOX C: CLT/FEM FLOWCHART, FINAL VERSION	152
FIGURE 5-19 - "SIDE STAY FITTING" LAMINATE THICKNESS DETAILS REQUIRED [3]	154
FIGURE 5-20 - BOX C: CLT/FEM VERY DETAILED FLOWCHART FOR BOTH LEVELS	155
FIGURE 5-21 – LEVEL 1 FEM LOADING CASE: CANTILEVER BEAM N_x	156
FIGURE 5-22 - LEVEL 2 FEM LOADING CASE: CANTILEVER BEAM N_{xy}	157

FIGURE 5-23 – LEVEL 1, N_x LOAD CASE: NORMAL STRAIN CONTOURS AND TRANSLATIONAL DISPLACEMENT	158
FIGURE 5-24 – LEVEL 2, N_{xy} LOAD CASE: SHEAR STRAIN CONTOURS AND TRANSLATIONAL DISPLACEMENT	158
FIGURE 5-25 – LEVEL 1, N_x LOAD CASE: ELEMENT 23 (MID X LENGTH)	159
FIGURE 5-26 – LEVEL 2, N_{xy} LOAD CASE: ELEMENT 25 HIGHLIGHTED BY A WHITE CIRCLE	159
FIGURE 5-27 EDGE APPLICATION NODE LABELS	162
FIGURE 5-28 – N_x LOAD CASE: LOADS AND ID NODES, LEVEL 1	163
FIGURE 5-29 – N_{xy} LOAD CASE: LOADS AND ID NODES, LEVEL 2	163
FIGURE 5-30 – PSHELL NASTRAN CARD: LAMINATE TOTAL THICKNESS	164
FIGURE 5-31 – MAT8 NASTRAN CARD: EFFECTIVE ENGINEERING PROPERTIES	164
FIGURE 5-32 - NASTRAN OUTPUT POSSIBLE REQUESTS	166
FIGURE 5-33 – LEVEL 1, OUTPUT NASTRAN FILE: ID ELEMENT 23 NORMAL STRAIN	166
FIGURE 5-34 – LEVEL 2, OUTPUT NASTRAN FILE: ID ELEMENT 25 SHEAR STRAIN	166
FIGURE 5-35 - VALIDATION TEST FLOW DIAGRAM	171
FIGURE 5-36 - VALIDATION TEST: CONSTRAINT CONDITION	172
FIGURE 5-37 - CONVERGENCE TEST FLOW DIAGRAM	176
FIGURE 5-38 - CONVERGENCE TEST CONDITION	177
FIGURE 5-39 - OPTIMUM DESIGN FLOW DIAGRAMS	179
FIGURE 5-40 I- LEVEL MATLAB CODE FLOW CHART	181
FIGURE 5-41 “BE_MAIN.M” MAIN MATLAB FILE FLOW CHART: LEVEL 1	182
FIGURE 5-42 “BE_MAIN.M” MAIN MATLAB FILE FLOW CHART: LEVEL 2	183
FIGURE 5-43 “BE_MAIN.M” MAIN MATLAB FILE FLOW CHART: LEVEL 1, LEVEL 2 AND CONVERGENCE TEST	184
FIGURE 5-44 – ITERATION1, 2D L1 AND L2 DESIGN SPACE PLOTS	185
FIGURE 5-45 – ITERATION2, 2D L1 AND L2 DESIGN SPACE PLOTS	186
FIGURE 5-46 – ITERATION3, 2D L1 AND L2 DESIGN SPACE PLOTS	187
FIGURE 5-47 – ITERATION4, 2D L1 AND L2 DESIGN SPACE PLOTS	188
FIGURE 5-48 – ITERATION5, 2D L1 AND L2 DESIGN SPACE PLOTS	189
FIGURE 5-49 – ITERATION6, 2D L1 AND L2 DESIGN SPACE PLOTS	190
FIGURE 5-50 – RUN1 ORIENTATION ANGLE DESIGN SPACE CONVERGENCE EFFECT	191
FIGURE 5-51 - RUN1 ORIENTATION ANGLE DESIGN SPACE: ACHIEVED CONVERGENCE	192
FIGURE 5-52 – RUN1, L1 AND L2 DESIGN VARIABLES HISTORY	193
FIGURE 5-53 – RUN1, L1 AND L2 OBJECTIVE FUNCTION HISTORY	194
FIGURE 5-54 – RUN1, L1 AND L2 CONSTRAINT FUNCTION AND FEM STRAIN DATA HISTORY	195
FIGURE 6-1 – MULTILEVEL DECOMPOSITION STRUCTURE: INTERMEDIATE EXAMPLE TWO-LEVEL DECOMPOSITION	202
FIGURE 6-2 INTERMEDIATE EXAMPLE MULTILEVEL DECOMPOSITION, SINGLE LEVEL-I STRUCTURE	203
FIGURE 6-3 – INTERMEDIATE EXAMPLE: 3-FIXED-ORIENTATION ANGLES (3_F_OA) HALF LAMINATE CONFIGURATION	204
FIGURE 6-4 – INTERMEDIATE EXAMPLE: 2-VARIABLE-ORIENTATION ANGLES (2_V_OA) HALF LAMINATE CONFIGURATION	205
FIGURE 6-5 HALF LAMINATE ORIENTATION GROUP NUMBERING CONVENTION	213
FIGURE 6-6 - INTERMEDIATE EXAMPLE GRAPHICAL AND NUMERICAL OPTIMISERS: MATLAB STRUCTURE	215
FIGURE 6-7 – MATLAB COMMAND WINDOW: “KUHN-TUCKER CONDITION” RESULTS	218
FIGURE 6-8 – INTERMEDIATE EXAMPLE: DESIGN SPACE LEVEL 1	220
FIGURE 6-9 – INTERMEDIATE EXAMPLE, LEVEL1: ZOOM IN STEPS TO THE OPTIMUM DESIGN	221
FIGURE 6-10 – INTERMEDIATE EXAMPLE, LEVEL 1: FEASIBLE DOMAIN	223
FIGURE 6-11 – INTERMEDIATE EXAMPLE, DESIGN SPACE LEVEL2: THE PARABOLA	224
FIGURE 6-12 – INTERMEDIATE EXAMPLE, DESIGN SPACE LEVEL2: PARABOLA AND NORMAL STRAIN OBJECTIVE FUNCTION	225

FIGURE 6-13 – INTERMEDIATE EXAMPLE, DESIGN SPACE LEVEL2: PARABOLA AND NORMAL STRAIN OBJECTIVE FUNCTION FEASIBLE DOMAIN	226
FIGURE 6-14 – INTERMEDIATE EXAMPLE, DESIGN SPACE LEVEL2: PARABOLA, NORMAL STRAIN OBJECTIVE FUNCTION, AND SHEAR STRAIN CONSTRAINT FUNCTION, FEASIBLE DOMAIN	228
FIGURE 6-15 – INTERMEDIATE EXAMPLE, DESIGN SPACE LEVEL2: THREE-FIXED ORIENTATION ANGLE CONFIGURATION, THE TRIANGLE	230
FIGURE 6-16 – INTERMEDIATE EXAMPLE, DESIGN SPACE LEVEL2: LAMINATES WITH COMBINATIONS OF THREE FIXED ORIENTATION ANGLES (0°, 90°, +/- 45°)	231
FIGURE 6-17 –DESIGN SPACE LEVEL2, OPTIMUM DESIGN CHOICE: EQUAL SPACED SEGMENT (ESS) AND DIVISION NUMBERS (DN).	232
FIGURE 6-18 – MULTI-LEVEL DECOMPOSITION: THE “G1F2” CONVERGENCE PROCESS	240
FIGURE 6-19 - IE_MAIN.M” MAIN MATLAB FILE FLOW CHART: LEVEL 1	242
FIGURE 6-20 - IE_MAIN.M” MAIN MATLAB FILE FLOW CHART: LEVEL 2	243
FIGURE 6-21 “IE_MAIN.M” MAIN MATLAB FILE FLOW CHART: LEVEL 1, LEVEL 2 AND CONVERGENCE TEST	244
FIGURE 6-22– ITERATION1, 2D L1 AND L2 DESIGN SPACE PLOTS – INTERMEDIATE EXAMPLE	245
FIGURE 6-23 – ITERATION2, 2D L1 AND L2 DESIGN SPACE PLOTS –INTERMEDIATE EXAMPLE	247
FIGURE 6-24 – ITERATION3, 2D L1 AND L2 DESIGN SPACE PLOTS –INTERMEDIATE EXAMPLE	249
FIGURE 6-25 – ITERATION4, 2D L1 AND L2 DESIGN SPACE PLOTS –INTERMEDIATE EXAMPLE	251
FIGURE 6-26 – ITERATION5, 2D L1 AND L2 DESIGN SPACE PLOTS - INTERMEDIATE EXAMPLE	253
FIGURE 6-27 – LEVEL-2 UNFEASIBLE DOMAIN: RUN2, ITERATION1	255
FIGURE 6-28 – LEVEL-2 FEASIBLE DOMAIN: RUN2, ITERATION 4	256
FIGURE 6-29 – LEVEL2 FEASIBLE DOMAIN: RUN2 ITERATION 5	257
FIGURE 6-30 – LEVEL2 FEASIBLE DOMAIN: RUN2 ITERATION 5, 2-VARIABLE ORIENTATION ANGLE	258
FIGURE 6-31 – RUN2, L1 AND L2 ORIENTATION ANGLES	259
FIGURE 6-32 – RUN2, L1 AND L2 THICKNESSES AND T2 DESIGN HISTORY	260
FIGURE 6-33 – RUN2, L1 AND L2 OBJECTIVE FUNCTIONS DESIGN HISTORY	261
FIGURE 6-34 – RUN2, L1 AND L2 CONSTRAINT FUNCTION AND FEM STRAIN DATA HISTORY	262
FIGURE 6-35 - RUN2, INTERMEDIATE EXAMPLE: HALF LAMINATE CONFIGURATION HISTORY, LEVEL2	266
FIGURE 7-1 – COUPLING ANALYSIS AND OPTIMISATION ACCORDING TO NASTRAN SOL200 [78]	271
FIGURE 7-2 – MSC NASTRAN MAIN STRUCTURE [78]	271
FIGURE 7-3 – NASTRAN SOL200: STRUCTURAL ANALYSIS AND DESIGN STUDY DATA FLOW STRUCTURE	273
FIGURE 7-4 – MULTI-LEVEL DECOMPOSITION: SINGLE LEVEL REVIEWED	274
FIGURE 7-5 – MULTI-LEVEL DECOMPOSITION: SINGLE LEVEL LABELLED USING SOL200 TERMINOLOGY	275
FIGURE 7-6 – 2-LEVEL DECOMPOSITION MAIN STRUCTURE USING SOL200 TERMINOLOGY	276
FIGURE 7-7 – RUN1 NASTRAN SOL200 OBJECTIVE AND MAXIMUM CONSTRAINT HISTORY	280
FIGURE 7-8 – RUN1 NASTRAN SOL200 DESIGN VARIABLE HISTORY	281
FIGURE 7-9 – NASTRAN SOL200 RUN1: PLY THICKNESS DESIGN VARIABLES DESIGN HISTORY	282
FIGURE 7-10 – NASTRAN SOL200 RUN1: ORIENTATION ANGLES DESIGN VARIABLES DESIGN HISTORY	282
FIGURE 7-11– NASTRAN SOL200 RUN1: MAXIMUM CONSTRAINT DESIGN HISTORY	283
FIGURE 7-12 – NASTRAN SOL200 RUN1: OBJECTIVE FUNCTION DESIGN HISTORY	283
FIGURE 7-13– RUN2 NASTRAN SOL200 OBJECTIVE AND MAXIMUM CONSTRAINT HISTORY	285
FIGURE 7-14– RUN2 NASTRAN SOL200 DESIGN VARIABLE HISTORY	285
FIGURE 7-15– NASTRAN SOL200 RUN2: PLY THICKNESS DESIGN VARIABLES DESIGN HISTORY	286
FIGURE 7-16 – NASTRAN SOL200 RUN2: MAXIMUM CONSTRAINT DESIGN HISTORY	286
FIGURE 7-17 – NASTRAN SOL200 RUN2: OBJECTIVE FUNCTION DESIGN HISTORY	287
FIGURE 7-19 – PATRAN IN-PLANE STIFFNESS ELEMENTS: MEMBRANE MATRIX	288
FIGURE 7-20 – MATLAB IN-PLANE STIFFNESS ELEMENTS: A MATRIX (EQ. 3-23)	289
FIGURE 8-1 – MULTILEVEL DECOMPOSITION STRUCTURE: ADVANCED EXAMPLE THREE-LEVEL DECOMPOSITION	301
FIGURE 8-2 – ADVANCED EXAMPLE: LEVEL STRUCTURE	302
FIGURE 8-3 – ADVANCED EXAMPLE: STRUCTURAL MODEL EVOLUTION	303

Acknowledgment

I would like to dedicate this paper to my dad Pippo and mum Pinuccia that supported me from the very beginning until the very long end throughout many adversities.

Special thanks go out to my project advisors, Dr. Kevin Hughes, who shown incredible flexibility and availability to cope with my logistic and technical issues, and Professor Rade Vignejevic, who gave me this opportunity and made all in its possibility to let me perform at my best.

Significant mention goes to G8 and Martozzo Dimba, who inspired me with long technical chats and encouragement, to Ona and Ina, they shown me the best way to cope with this and other challenges enabling me to successfully succeed in my project, but not least thanks to my very unique rare friends; Bali, Malivay, Basildon, Malice, Lucy, La Tereisa, La Louvisa, Sidg, Balulone, KKs and Il mao.

1 Introduction

1.1 General background

In a continuous effort to achieve the increasing demands of new aircraft with better performance, reliability and operability, the incorporation of advanced materials and advanced structural concepts is becoming increasingly important to manufacturers, to ensure a competitive edge in a saturated market. Composite materials are expected to be used extensively and their use is being extended to crucial aerospace components, such as Principle Structural Elements (PSE).

Laminated fibrous composite materials are finding a wide range of applications in structural design, especially for lightweight, high stiffness and excellent strength performance compared to their counterpart metallic materials. While they are attractive replacements, the analysis and design of composites are considerably far more complex. Finding an efficient composite structural design that satisfies all requirements of a specific application, can be achieved not only by sizing an isotropic structure (cross sectional areas and component thicknesses), but also by tailoring the specific orthotropic characteristic properties through selective choice of orientation, number, and stacking sequence of the layers that define the composite laminate [1].

The increased number of design variables, and the difficulty in identifying the most sensitive parameters controlling a design together with manufacturing constraints, makes mathematical optimisation techniques a natural tool for the design of laminated composite structures.

In such scenario, the European Union, in an attempt to reduce aircraft operating costs by realising the weight saving potential of composite materials, commissioned the ALCAS Project (Advanced Low-Cost Aircraft Structure), whose aim is to investigate new

design concepts and methods for expanding the use of these new materials in aerospace structures [2].

The Project is organised in four technical platforms. In one of those, “The Airliner Wing Platform”, Cranfield University will be involved with investigating the full potential of composite materials applied to the highly loaded interface between the main landing gear (MLG) and wing structures. To investigate this area of research, it will be necessary to couple optimisation techniques with numerical structural analysis codes that are capable to analyse the complex loads coming from the ground (landing, take off and ground manoeuvre conditions), which are then reacted through the aircraft main wing and ultimately the fuselage, as shown in Figure 1-1.

This requires designing further more complex components, thick laminates, which drastically increase the difficulties of structural optimisation techniques.

In fact for a thick laminated composite structure, there will always be a larger number of design variables based upon ply angle selection alone. This is further complicated by the highly nonlinear way in which related constraints associated with laminated composite structure vary with changes in fibre orientation [4].

Rapid increase in computer processing power, memory, and storage space has improved the application of optimisation techniques in this field but the computational cost and time constraints are still open issues [5].

The above boundary conditions justify the necessity to propose an efficient approach that can handle a: large number of design variables, highly non linear response and a high computational demand, that are required in the optimisation of thick composite structures.

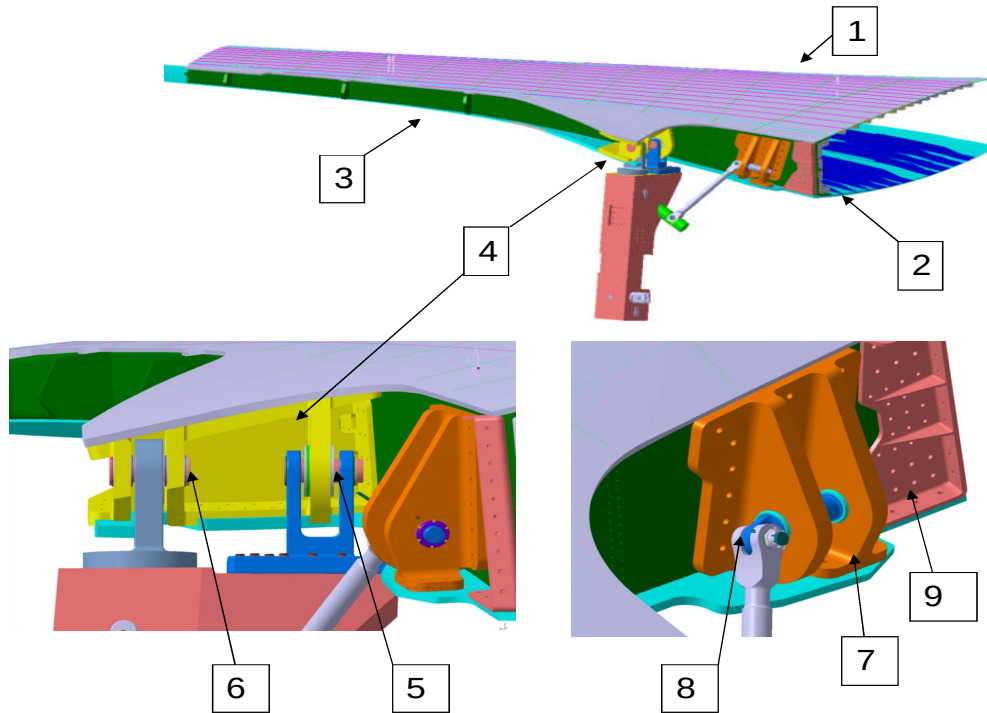


Figure 1-1 - Critical components of the ALCAS lateral wing box assembly [3]

1. Upper Cover
2. Lower Cover
3. Rear Spar
4. Landing Gear Beam
5. Forward Pintle Pin
6. Aft Pintle Pin
7. Side Stay Bracket
8. Side Stay Pin
9. Wing Root Bracket

Item 7 is focus of Cranfield involvement with ALCAS, with the ultimate aim of identifying a practical methodology for converting this bracket from metallic into a composite equivalent.

1.2 Research Objectives

The overall aim of this research is to develop a methodology to assist in the optimisation of composite laminated structures. In a general Multi-level Decomposition [4] (MD) technique, outlined schematically in Figure 1-2, the main structure at the top level is decomposed into a hierarchical tree consisting of substructures at different levels. The main advantage of the multilevel optimisation scheme lies in the fact that all substructures can be analysed and optimised independently with convenient coupling. This application makes such an algorithm attractive for parallel computing technology and this approach forms the starting point for this research [6].

The following bulleted list identifies the main research objects:

- Develop a methodology to apply in optimisation of composite laminate structures under in-plane static loads
- Historical perspective of Structural Optimisation Complexity to laminated composite materials identifying the key drivers (approaches, tools, design variables, objective functions.....)
- Learn Optimisation Techniques best suited to Composites using commercial optimiser (Nastran SOL200) and mathematical programming (Matlab)
- Understand Classical Lamination Theory (CLT), In-Plane Lamination Parameters, Failure Criteria performing all of them through Matlab Script
- Comparison of the Matlab CLT results with Pre-Post processor outcomes (Patran)
- Perform FEM (laminate modeller, homogeneous 2-D orthotropic elements) different technique analyses to composite structures and define the one to embed in the research optimisation method
- Create Multilevel Decomposition (MD) numerical application defining by Mathematical Programming: Variables Coupling, Validation Test and Convergence Test (Basic Example)
- Learn how to run and link Nastran to Matlab as part of the MD

- Increase the difficulty of the basic example using different design variables, lamination parameters, and alternative optimisation tools, Kuhn- Tucker and Lagrange Multipliers, still in a 2-level decomposition numerical application (Intermediate Example)
- Use Nastran Sol200 to compare the 2-level decomposition results
- Propose a MD example, just the framework not any numerical effort, (Advanced Example) as a ground base structure for a future optimisation of a main landing gear attachment (ALCAS project) idealised as a thick composite structure subject to “lug loading”

1.3 Multi-Level Decomposition

Decomposing the primary problem statement into a system level idealisation problem and a set of uncoupled component level problems allows the results to be obtained by iterating between the system and component level problems by a vertical data flow.

System decomposition is essentially concerned with dividing the required analyses into subsystems with minimal coupling, so as to enable a degree of concurrency in the analysis. In many multidisciplinary design problems, not just complex structural issues, it may be preferable to decompose the system under consideration along the lines of the disciplines involved. However, the idea of system decomposition is very general and can be applied to any problem where it is necessary to run (or iterate between) a sequence of modules to calculate objective and constraint functions. The aim is to reorder the analysis tasks so as to reduce the degree of feedback between them [7].

Systems can be classified into different categories, depending on the type of coupling between the subsystems. The model problem described by data transferred from discipline 1 to 2 and vice versa is a non-hierarchic system. On the other hand, decomposing a problem where there is no feedback between the disciplines is a hierarchic system. In more general terms, systems with multidirectional data flow are non-hierarchic, whereas systems with one-way data flow are hierarchic. Hybrid systems involve a combination of hierarchic and non-hierarchic subsystems [7].

A formalisation of the basic idea into a procedure suitable for computer implementation by mathematical programming will be the core of the thesis.

Mathematical models of engineering systems are usually implemented as assemblages of software modules; each module represents an aspect of the system, such as a physical phenomenon or part. To increase their usefulness in design, these modules evolved as testimonial of “design-oriented analysis”, a term introduced by Storaasli and Sobieszczanski [8]. These attributes include the ability to select the level of analysis, which ranges from inexpensive and approximate to accurate and costly; “smart” analysis is for rapid evaluation of design-change effects; and the inclusion of the data management and visualisation infrastructure necessary to handle volumes of data, which in a design process can be quite large [9].

The computational burden may simply reflect Multi-level Decomposition as the sole option to tackle it. On the contrary, it is the base to develop a specific tailored methodology, where at each level, the most suitable tools are employed to characterise the maths and physics of the problem. This led to the identification the other core part of this research;

- graphical and numerical optimisers (commercial and hand-made)
- different design lamination procedures (Classical Lamination Theories, CLT, and Lamination Parameters, LP)
- three-dimensional stress state typical of thick laminates under out of plane loading, non predictable accurately by classical lamination theory (Shear Deformation Theories, SDT)

The main driver for this thesis is to embed in a Multi-level Decomposition, all of the above different theories, and identify the right way to couple the design variables between one level and another.

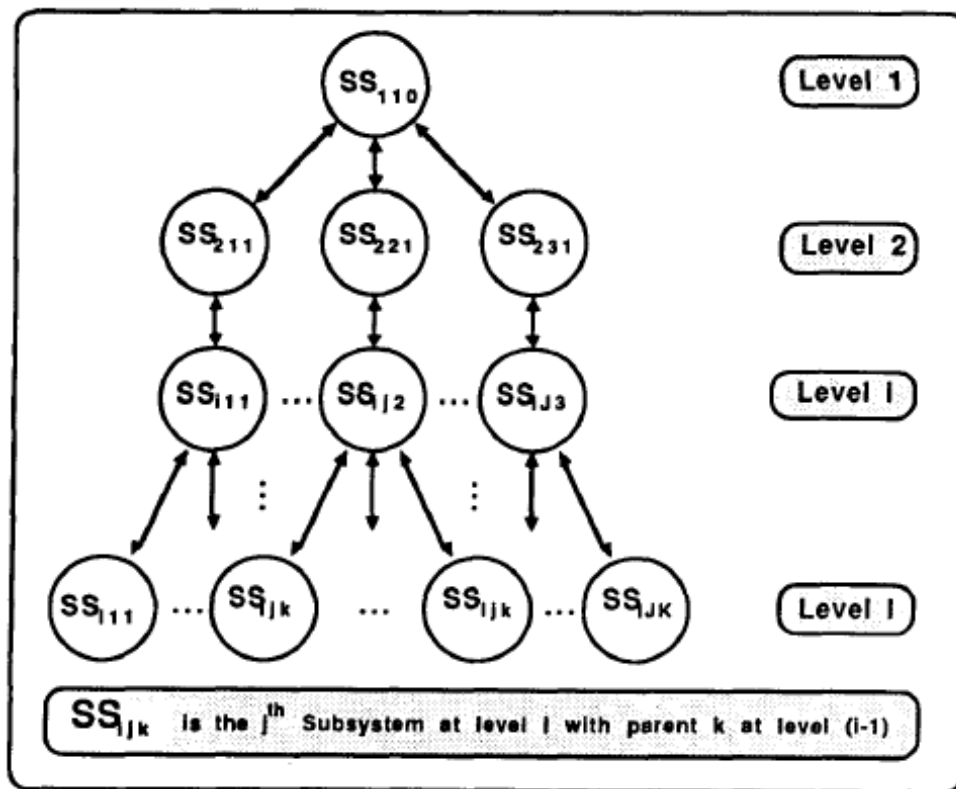


Figure 1-2 - A General Multi-Level Decomposition Structure [6]

1.4 Research Methodology

A two-level (2-LD) and three-level (3-LD) decomposition approach are chosen as the main framework on which the design and optimisation concepts of Laminated Composite Materials will be developed and validated using specific numerical applications of increasing difficulty and complexity.

The optimisation approach is basically characterised by the structure depicted in Figure 1-3, where the necessary concepts/functionality are embedded at different levels, L_i .

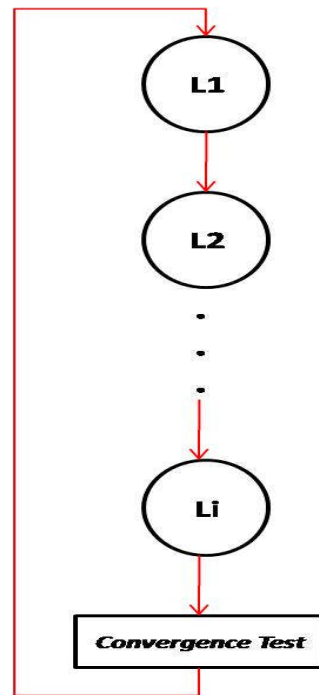


Figure 1-3 - Multi-Level Decomposition Structure

The main tasks are:

- Convergence Test
- Validation Test(s)

The Convergence Test is unique in the whole procedure and is responsible to authorise the start of the next iteration. Part of this work will be to define a suitable condition within this box, and to narrow the design space at the end of every loop until convergence is obtained.

The Validation Tests are as many as the number of levels in the decomposition. The condition chosen within this box basically confirms that the optimum design coming from the optimiser still produces data, which is consistent with the initial inequality constraint conditions. This type of check is external to the main mathematical programming, performed by a commercial FEM code, MSC Nastran.

The research strategy is based on facing different numerical applications with increasing difficulty and complexity of the laminated composite structures, in order to develop a practical and robust analysis methodology.

For this reason, it will be considered and detail in chapter 5 to 7 respectively:

- Basic Example – Chapter 5
- Intermediate Example – Chapter 6
- Advanced Example – Chapter 7

Despite the increasing complexity of the examples, the standing point is a laminate that is symmetric and balanced, subject to stress or strain based constraints under in-plane, or out-of-plane static loading. The objective function is to minimising the weight of the composite structure.

For these reasons, the Basic and Intermediate Examples will be characterised by same design parameters, inequality constraints and optimisation technique, which is below described in the following section:

- in-plane loadings (N_x and N_{xy})
- strain constraints (ε_x and γ_{xy})
- Graphical Optimisation

Whilst the main differences between the two examples are:

- Number of laminate layers
- FEM validation test
- Second Level design variables

On the other hand, the Advanced Example will consider the full package of issues, typically encountered with thick laminate structures. This will involve dealing with transverse/inter-lamina stresses and out-of-plane loadings, which will see the following

concepts being introduced, which as one can appreciate, increases the complexity over the other two examples:

- 3-level decomposition
- Failure stress constraints
- Sub-laminate idealisation
- Linear Approximation, Gradient Based and Genetic Algorithm Optimisation Techniques

For a thorough comprehension of the three examples the main steps of the design optimisation procedure will be collated in the following tables, allowing the reader to compare the main differences.

Basic Example

<p><u>Level 1</u></p> <p>Find t_1, t_2</p> <p>Minimise $f(t) \rightarrow (Weight)_{\min} \quad t_1, t_2 \in \mathcal{R}$</p> <p>Subject to</p> $\varepsilon_x(t_1, t_2) < (\bar{\varepsilon}_x)$ $0 \leq t \leq \mathcal{R}^+$	
<p><u>Level 2</u></p> <p>Find ϑ_1, ϑ_2</p> <p>Maximise $f(\vartheta) \rightarrow (\varepsilon_x)_{MAX} \quad \vartheta_1, \vartheta_2 \in \pm Deg.$</p> <p>Subject to</p> $\gamma_{xy}(\vartheta_1, \vartheta_2) < (\bar{\gamma}_{xy})$ $\forall \vartheta \in \pm Deg.$	

Table 1-1 - Basic Example 2-Level Decomposition Summary

Intermediate Example**Level 1**Find t_1, t_3 Minimise $f(t) \rightarrow (Weight)_{\min} \quad t_1, t_3, (t_2 \text{ given}) \in \Re$

Subject to

$$\varepsilon_x(t_1, t_3) < (\bar{\varepsilon}_x)$$

$$0 \leq t \leq \Re^+$$

Level 2Find V_1^*, V_3^* (lamination parameters)Maximise $f(g) \rightarrow (\varepsilon_x)_{\max} \quad V_1^*, V_3^* \in \Re$

Subject to

$$\gamma_{xy}(V_1^*, V_3^*) < (\bar{\gamma}_{xy})$$

$$-1 < V_1^*, V_3^* < 1$$

Table 1-2 - Intermediate Example 2-Level Decomposition Summary

Advanced Example**Level 1**Find t_r Minimise $f(t) \rightarrow (Weight)_{\min} \quad t_r \in \Re \quad r = 1, 2, 3, \dots, n \quad n \in \square$

Subject to

$$\varepsilon_x(t_r) < (\bar{\varepsilon}_x)_{\max}$$

$$0 \leq t_r \leq \Re^+$$

Level 2Find $(C_{ij})_{(k)}$ (sub laminate stiffness coefficients)Maximise $f((C_{ij})_{(k)}) \rightarrow (\varepsilon_x)_{\max} \quad (C_{ij})_{(k)} \in \Re \quad k = 1, 2, 3, \dots, n \quad n \in \square$

Subject to

$$(InterfacialShear)_{\max} < (Allowable)$$

$$\Re^- < (C_{ij})_{(k)} < \Re^+$$

Level 3Find $\vartheta_{(k)}$ (angles of sub-laminates)Maximise $f(\vartheta_k) \rightarrow (InterfacialShear)_{\max} \quad \vartheta_{(k)} \in \pm Deg. \quad k = 1, 2, 3, \dots, n$ $n \in \square$

Subject to

$$[(C_{ij})_{(k)}]_{InPlaneLoads} < \bar{K} \quad \& \quad [(C_{ij})_{(k)}]_{OutofPlaneLoads} < \bar{K}$$

$$\forall \vartheta \in \pm Deg.$$

Table 1-3 - Advanced Example 3-Level Decomposition Summary

It is important to note due to the constraints of this research program, the Advanced Example will be presented and described its main structure fully, as it was not possible to include any numerical validation, and this represents an area of future work.

1.5 Software Methodologies

The main hurdle of introducing different Multi-level Decompositions is mainly based on the ability of merging all the following basic concepts.

- Mathematical Programming (MA)
- Hand Made Optimiser (Graphical, Kuhn-Tucker, Gradient Based)
- Commercial Optimiser (ALTAIR HyperWorks [10], Nastran [11])
- Design Lamination Theories (CLT, LP , Shear Deformation Theories)
- Parallel Programming
- Numerical Modelling Techniques (Finite Element Method, FEM)

The Mathematical Programming was made possible by Matlab, which forms the main structure of this research [12]. Matlab has been chosen since it is a standard tool for introductory and advanced courses in mathematics, engineering, and science in many universities around the world.

In industry, it is a tool of choice for research, development, and analysis. In this Thesis, Matlab will be exploited to manipulate the main variables, between one level and the other using vectors and matrices that are natural to numerical techniques. In addition, the powerful visualisation features for graphical optimisation and the interactive features and built in support to translate algorithms into functioning code in a fraction of time needed, when compared to programming languages like C or Fortran is Matlab other strength [12]. But mainly Matlab is a very flexible code that allows to interface with external Finite Element Modelling software, such as Nastran. In fact, as part of the validation box, see later Basic and Intermediate Sections, Matlab will read and change the Nastran input text file (*.bdf), using the optimum design data coming from the Optimiser, and then it will run, with very basic command within the matlab script (*.m file), the *.bdf file through Nastran to create the output *.f06 file. This functionality

supports the decision to use Matlab as the main structure of this proposed Multi-Level Decomposition Methodology.

The main data flow is highlighted in Figure 1-4.

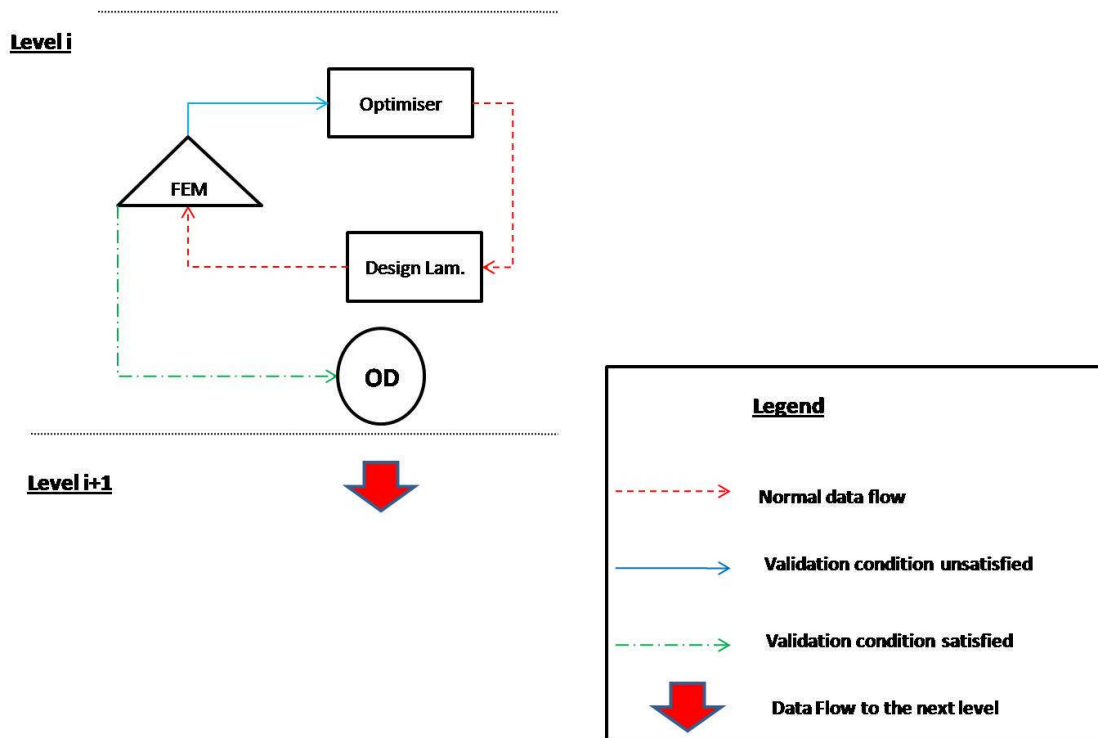


Figure 1-4 - Flow Diagram of the main Tools for a required generic level

The above flow diagram points out the link between the main tools described earlier, which have to be link via a common matlab script file. This is a typical i-level structure, where the data starts from the Optimiser, which feeds the Design Lamination Box with the optimum design variable data. Once the constitutive equations, CLT or Lamination Parameters, or Shear Deformation Theories have been applied, a new output will go into the FEM Triangle Box to verify the Validation Condition (this new data must still satisfy the initial inequality constraint condition). If the answer is positive, dash-dot line, the data will end up to the Optimum Design step, (circle box), where the data is passed to the next level. Otherwise the matlab program will revert back to its initial

state, solid line, where user intention is required to review some unreliable starting value that has resulted in an infeasible design.

It is important to precise that the Design Lamination Theory type appointed is going to define the type of Finite Element Model, required for the Validation Box. It means that dealing with Shear Deformation Theories (Interlaminar Normal and Shear Stresses [13]) one needs to introduce a FEM that takes into account those effects through the use of incorporating higher-order theories for the degree of freedom of the elements involved [14].

The same data flow described in Figure 1-4 has the potential to be performed in parallel, according to the Multi-level decomposition (MD) structure. In fact, interfacing the MLD with commercial and/or non-commercial optimisation and structural analysis codes will be another direction of this study, in order to develop a general methodology that can cope with a higher level of completeness and flexibility in tackling design optimisation of structure, for problems involving a high number of variables. Middleware is the term used to define the communication links between any pre-processor, solver and optimiser, whether this is for structural or CFD, etc, applications for example. The optimisers and simulation codes are coupled through the communication middleware tool, known as a Component Template Library (CTL) [15] , [16].

The introduction of these additional tools will be necessary when considering more Advanced Examples such as defined in Table 1-3, that deal with sub-laminate idealisation.

On the other hand, Commercial Optimisation Tools, such as ALTAIR HyperWorks [10] and Nastran SOL200 [78], which are not taking part in influencing the Flow Diagram of Figure 1-4, will be pivotal in the process of external and independent validation of the Matlab Script defining this Multi-level Decomposition approach and results generated by this novel methodology proposed within this research.

1.6 Summary

The novel part of the proposed study is based on composite laminate structure optimisation performed by the implementation of a Multi-Level Decomposition method, implemented within Matlab, which has been linked with a commercial FEM solver, Nastran, for validation. This procedure has developed within a dataflow framework,

whereby all the tasks are performed simultaneously, in an automated fashion. The flexibility of this approach will allow the development of this framework to consider increasingly complex problems through the addition of sub-blocks, set to run in parallel by coupling the flexibility of Matlab alongside a Component Template Library Utility [16].

This section has provided a brief overview of the proposed approach, by outlining the flexibility necessary to create a reliable and robust design optimisation methodology prepared to deal not only with thick composite materials, but with any problems characterised with highly non-linear equations and large number of design variables.

Unfortunately, due to the high content of the background information of the theory require, the literature review covers all the topics coupled with successful implementation and validation of Matlab, there was insufficient time available to demonstrate all examples through numerical applications.

This is the reason why the Advanced Application will be presented mainly by its theoretical framework and not through real life application problems, which were used in the chapters describing the Basic and Intermediate Examples.

Nevertheless, it is the view of this author that the material presented in this thesis will be a useful starting point for future research to develop this approach into a more general framework.

2 Literature Review

2.1 Introduction

This literature review chapter will cover the main topics related with structural optimisation of composite materials. As can be seen the aim is to examine authors and their field of research mainly involved with the following:

- Main evolutionary steps in structural optimisation
- Bioengineering Application as thick composite material applications
- Optimisation Decomposition Techniques
- Parallel computation potentialities
- In-plane and out-of-plane structural analysis loadings according to a numerical (FEM) and analytical approach

2.2 Brief Review of Structural Optimisation and Composite Materials

This literature review has been based on the effort to define and characterise the main evolutionary steps in structural optimisation of composite aerospace structures, since the early attempts during the sixties, right up to present day.

During the last forty years of work in the area of structural optimisation, (Haftka [5] as an example of one of the main researchers) computing speed and storage capacity increased one million fold at the rate of about 100-fold increase every decade [17]. In 1971, when structural optimisation was still struggling to deal with “real-life” problems,

a one million factor increase in computational speed would have seemed to solve all problems. After all, engineers were already analysing complex structures with detailed finite element models, and the ability to solve a million degrees of freedom model is sufficient for most optimisation problems.

However, several developments have reduced substantially the cost of structural analysis and optimisation. Improved finite element solvers and improved error estimates for adaptive mesh refinement, means that it is possible to use fewer degrees of freedom (DOF), for a given accuracy. Improved solvers mean it is achievable to solve the resulting system of equations faster. Based on anecdotal evidence, it is estimated that there have been improvements of one or two orders of magnitude in the cost of structural analysis between 1971 and 2002 [17]. In fact, during the 1970's, most engineering computations had to be performed on expensive (millions of dollars) mainframe, or supercomputers [5].

Today, this is not so uncommon, unless for very large-scale simulations (e.g. electronic packaging design problem of vanBloemen et al. [18] at Sandia national Labs), That as faster computers has led to engineering computations and challenging optimisation problems are now routinely performed on much cheaper desktop computers.

To deliver a historical perspective of how things have changed in the area of structural optimisation, Figure 2-1 shows that for the design of electronic packages, the numbers of degrees of freedom used to model the structure has increased by a factor of 4000 over 15-20 years [17].

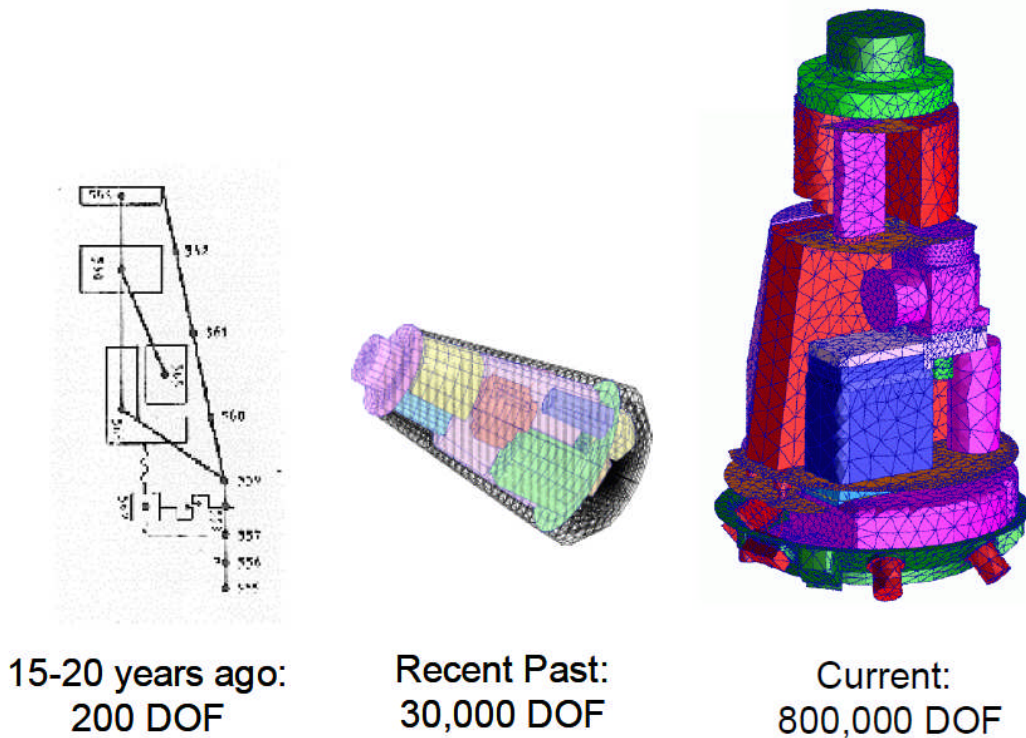


Figure 2-1 - Evolution in analysis model complexity, Current=2002/Recent Past=mid-90s/15-20 years ago= 80s - [17]

A similar example is found in the design of aircraft structures. During 1986 NATO advanced Study Institute meeting, Lecina and Petiau [19] from Avions Marcel Dassault, (France), as Wellen and Bartholomew

[20] of the Royal Aircraft Establishment, (United Kingdom), presented one of the first industry finite element analysis models used in structural optimisation of real aircraft structures. By 2001, a finite element model used to simulate the ultimate failure test of a full-scale composite wing by Jegley et al. [21] required 428,000 DOF. The model had to capture the response of the wing after it was subject to impact damage and several discrete sources of damage, in the form of a cut (7-inch crack) thorough the cover panel. In presenting the results of the verification at the 2001 SDM meeting, the authors apologised that the model prediction for the ultimate failure load was off by 3% from what was observed experimentally, which demonstrated how sophisticated numerical codes had become [5].

This brings to mind the evolution in model complexity required when analysing composite materials, with increased reliability than ever before. In fact, the use of composite materials as primary structures in the commercial aviation industry has been

gradually increasing during the last decade. This has culminated in programs such as the Airbus A350, or the Boeing 787, for which composite materials will play a major role. Using a similar analogy to that depicted in Figure 2-1, model complexity of composite materials has developed from analysing simple regular bi-dimensional (2D) plate structures to consider three dimensional (3D) structures including stiffened panels, plate with cut-outs etc with relative ease.

For a thorough understanding of the complexity of structural optimisation of composite materials, it is necessary to identify other concepts and not solely focus on modelling limitations. To answer this question, Venkataraman and Haftka [5] defined a measure of the difficulty of a structural optimisation problem using the following three complexity terms [5]:

- Model
- Analysis
- Optimisation

2.3 The Three Axes of Complexity

The terms difficulty, or complexity are not easy to quantify in terms of a structural analysis problem. However, using as a common research driver, Haftka and Venkataraman [5], studied the effect of computing power, indirectly through the computational cost. This means that geometry, design criteria, or refine of finite element model which requires many hours of effort are not be addressed, due to the difficulty of quantify the effects of those costs, in the following discussions.

In terms of computer cost, it is convenient to separate the causes of increased cost into two categories: modelling and analysis.

The number of degrees of freedom in a finite element model of a structure is often used as a first indication of the computer cost associated with modelling, but the topology of the structure has an important effect also; usually it will be much faster to analyse a slender beam type structure, rather than a compact three-dimensional structure with the same number of DOF. However, since most applications fall somewhere in between these two extremes, it will be accepted at this stage. Later, it will be shown that the differences between 2-D and 3-D composite structures may be quite substantial (such

as taking into account transverse inter-laminar shear and normal stress, requiring additional knowledge and understanding), that the common usage of reporting on the number of DOF in a discretised model is the primary measure of “modelling complexity”. Figure 2-2 provides examples of composite structures with increasing model complexity, together with a sample of technical papers that have performed optimisation of such structures [5].

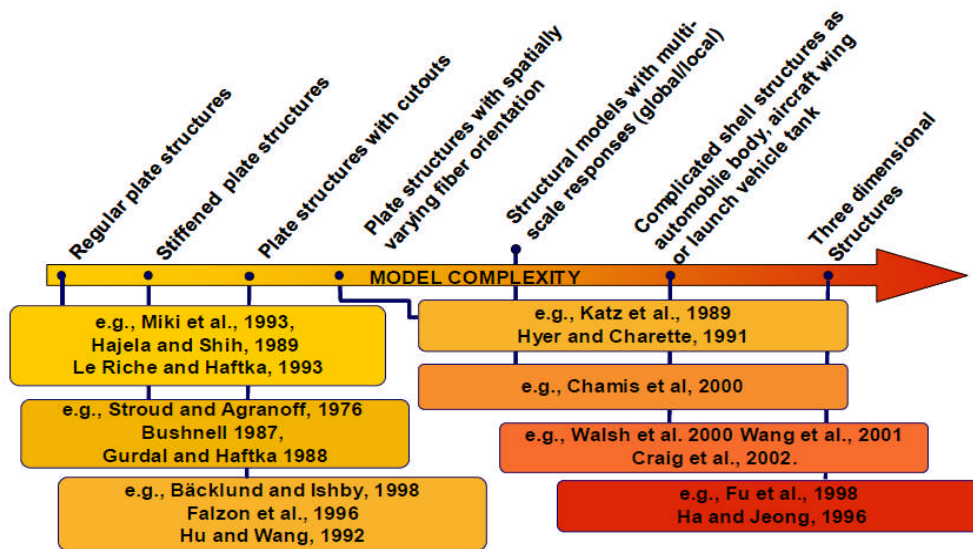


Figure 2-2 – Evolution of Model from simple models to complex 3D applications [5]

In terms of analysis complexity, linear elastic analyses are the simplest, and it is convenient to measure other types of analyses by their cost relative to a linear analysis. This is particular meaningful since more complex analyses such as nonlinear elastic analysis, linear repetitive bifurcation analysis, or dynamic response often use linearization algorithms that require the iteration of linear analyses.

History dependent nonlinear static and dynamic analyses, such as those required for metal forming and crash simulation, as well as nonlinear crack propagation problems are currently the extremes of analysis complexity. They typically require hundreds or thousands of linear analysis equivalents. Figure 2-3 describes examples of problems at various level of analysis complexity in the design of laminated structures, with references to some example papers. The simplest analysis is a linear static analysis. Eigen-value problems such as buckling and normal mode analysis require more effort for the same model size (DOF) than a static analysis. Transient and nonlinear analyses

require repeated evaluation of the incremental response and where the cost depends on the time or load step and maximum time up to which the response is desired.

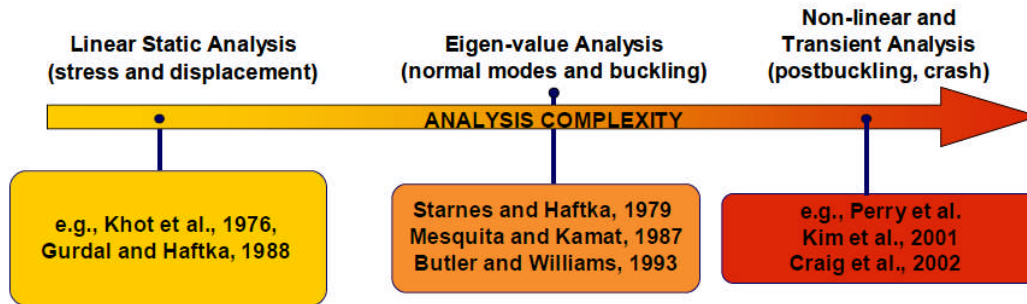


Figure 2-3 – Evolution of Model Analysis Types [5]

Finally, the cost of an optimisation is typically measured in the number of analyses required. However, this is an imperfect measure for several reasons. First, when gradient-based optimisation is used, the cost of calculating the gradients can be the dominating factor, even if the cost of a derivative may be much lower than the cost of running an analysis. Second, a large number of analyses may reflect an inefficient algorithm rather than a difficult problem. Third, when the number of design variables is comparable to the number of degrees of freedom, the cost of the optimisation algorithm itself may dominate the cost of the analyses.

Due to these shortcomings and the relationship between the number of analyses required as a measure of the optimisation cost, Haftka and Venkataraman proposed to add two other indicators that correlate with the cost [5]. One is the number of design variables and the other is the type of optimisation. The simplest type of optimisation is local, graphical and gradient based optimisation. Global optimisation with continuous variables or discrete and combinatorial optimisations are the most difficult, typically leading to difficult non-linear problems (NP) (that is the solution time may be expected to rise exponentially with the number of variables).

Another factor in optimisation complexity is the effect of uncertainty. Structural design in the presence of uncertainty typically requires the calculation of the probability of failure, which can even be formulated as an optimisation problem in its own right of identifying the most probable failure point. Structural optimisation under uncertainty therefore becomes the equivalent of a two-level optimisation process, with greatly increased cost compared to deterministic optimisation. Similarly, multi-objective

optimisation adds another order of magnitude of required analyses. Instead of finding a single optimum, it is requested to find all the designs on a Pareto Surface [7] which represents a locus of points of possible optimal designs. Multi-objective optimisation is part of the multi-disciplinary analysis process, whereby an iterative scheme is applied to the disciplinary governing equations to establish equilibrium [7], and one of the best ways to perform this is through Multi-level Decomposition Techniques. All the various levels of optimisation complexity used in the optimum design of laminated composite structures, with some sample of publications, with increasing complexity are shown in Figure 2-4.

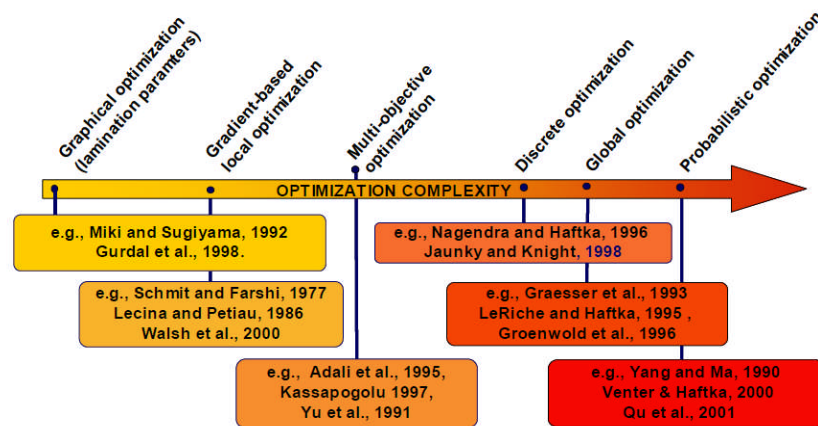


Figure 2-4 - Evolution in optimisation complexity for composite problems [5]

The “Three Axes of Complexity” is shown schematically in Figure 2-5, together with examples of differing levels of optimisation complexity identified. Figure 2-5 shows how moving from point O along the axes towards point A, or B, or C indicates increasing complexity in the analysis, model, or optimisation method used. The desire of a structural optimisation specialist is to reach point P, where all three levels of complexity are maximised. This appears a difficult, if not an impossible target, as disciplinary specialists constantly extend the limits of points A, B and C. Therefore, the state of the art optimisation problems will have to form a surface in the box shaped region, on which moving further along any one direction will require reduction in the other two directions. For example, global and/or probabilistic optimisation methods will be limited to a simple model for cases with expensive analyses.

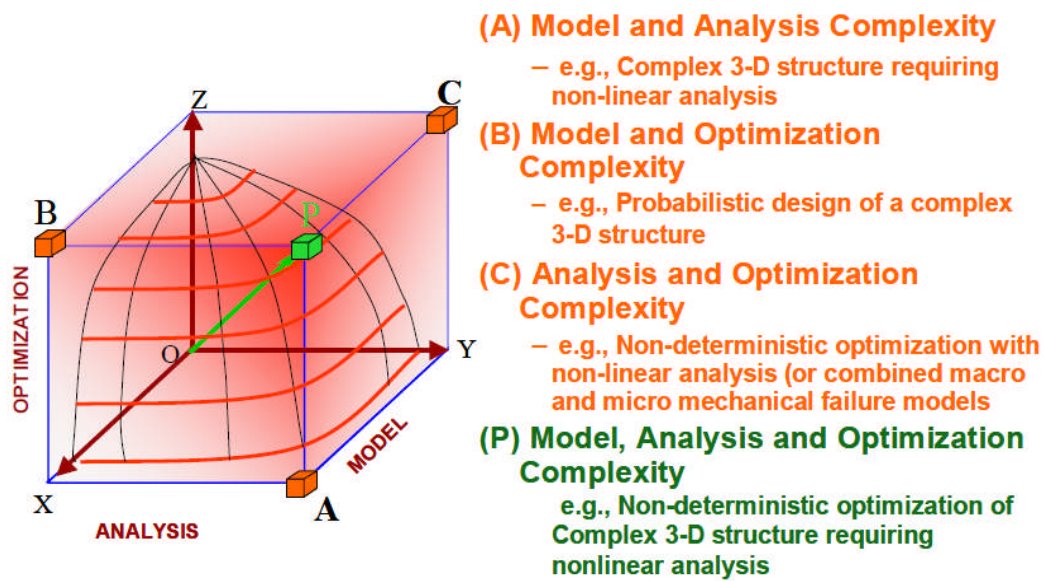


Figure 2-5 - The Three Axes of Complexity [5]

In the next section it will be discussed further investigations of thick laminates optimisation employing the information gathered by the Three Axes of Complexity described above [5].

2.4 Bioengineering Applications

A review of optimisation problems being solved at present indicate that it has been huge increase in analysis methods and optimisation algorithms, particularly in the use of approximation methods ,which have played a significant role in extending the envelope of what can be optimised [5].

Trade-offs in model, analysis and optimisation complexity has to be performed [5]. Optimisation of thick composite laminate structures falls in this category, as shown in Figure 2-6, needs to represent a compromises between analysis and optimisation complexity have to be performed. In fact, the points of the graphs represent examples of specific problems that have been previously published and the bounding curves of the points in each figure qualitatively represent the most complex problems permitted by the current state of the art [5]. The trend suggests the higher the complexity, the fewer

the publications and relating to Figure 2-2, it is well understood that analysing three dimensional structures represents a field to be pioneered.

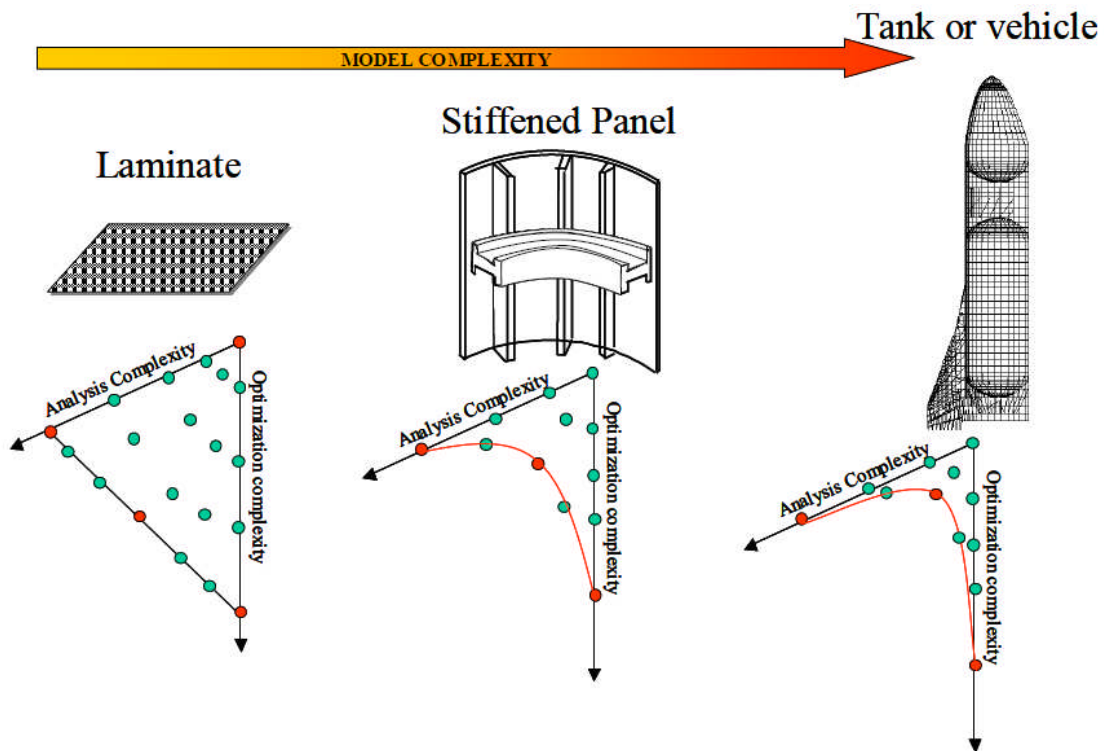


Figure 2-6 - Trade-offs in optimisation of composite laminate structures [5]

The few authors that tried to explore this increasing complexity direction are those mentioned in Figure 2-2, namely, Fu et al., Ha and Jeong, Biggers and Latour [4], who mainly work in bioengineering, and whose publications form the starting point of this review.

Total hip arthroplasty using metallic hip prostheses have been performed successfully for over three decades. However, it has been found that stress shielding and subsequent bone resorption of the proximal femur accompany such replacement procedures and may contribute to stem loosening and complicate remedial surgeries [22]. Stress shielding is caused by the non-physiological manner of joint-to-bone transfer and is proportional to the structural stiffness of the femoral component. Low-stiffness composite implants have been shown to reduce stress shielding [22]. However, clinical experience thus far has shown that the strength of such stems is a serious concern that provides a challenge to implant designers. In a laminated composite stem, the designer has the freedom to vary the component strength, and physiological performance. However, due to this increased design freedom and heterogeneous anisotropic nature of

composite stems, they are much more complex to design than their homogeneous isotropic metallic counterparts [23]. Therefore, to tackle the design of thick composite components is characterised by three main difficulties:

- Large Number of Design Variables
- Highly Non-Linear Nature
- Huge Computational Demand

In order to address these difficulties, it has been necessary to develop specific design optimisation approaches which are discussed in [4], [23], [24] and [25].

2.5 Decomposition and Multi-Level Optimisation

In order to overcome the design limitations associated with the optimisation of thick composite laminates, approaches need to be developed where the number of design variables under consideration is kept as low as possible. In addition, the highly non-linear relationships between objective, constraint, and design variables must be reduced as well.

Several studies have been devoted to the decomposition of large-scale optimisation problems. Mainly two classes of decomposition methods exist [7]:

- Formal methods
- Intuitive methods

Formal methods decompose a problem using its mathematical structure. Such decomposition may be fully automatic and be built-in within the design cycle. However, intuitive methods require an understanding of the physics of the system, as this will be the prime factor influencing the decomposition. Intuitive methods provide an alternative for decomposing problems that do not possess a structure for which a formal decomposition method can be applied [7].

Sobieski and James developed a method for decomposing an optimisation problem into a set of smaller sub-problems to reduce the design time on a large design problem [23]. In fact, Sobieski focused on the optimisation of systems whose mathematical model are an assembly of coupled modules, each transforming input to output. Analyses and optimisations usually corresponding to engineering disciplines or physical components may be executed within these modules. The term “discipline” will be used throughout to mean such a module.

Approaches to this problem can be divided into single-level optimisation approaches and multilevel optimisation approaches (the term single-level will be used later as the basic approach to employ commercial optimiser in the validation phase). In the former, only a single optimisation problem is solved for the entire system, while in the latter, optimisation problems are solved within the disciplines as well as for the system as whole. Thus, in a two-level optimisation approach, there is an optimisation at the system level, and there are optimisations at the discipline level. It may be possible to view a discipline as a system itself composed of sub-disciplines. In this case, a three-level optimisation approach may be employed, wherein optimisations occur at the system-level, the discipline-level, and the sub-discipline-level. This scheme is readily extendable beyond three levels [26] as already shown in Figure 1-3.

The first technique proposed by Sobieski, [8], has been a method for linearly decomposing a single large optimisation problem into multi-level optimisation problems. In this method, the system was decomposed into disciplines, and optimisations and analyses were performed at the system level and within each of the disciplines. The disciplines were coupled to the system but not to each other. Such a system was referred to as a “hierarchic system” as shown in Figure 2-7.

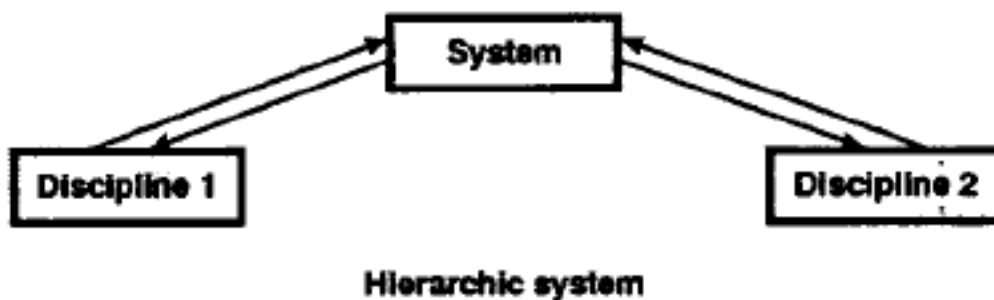


Figure 2-7 - Multi-level Decomposition: Hierarchic System [26]

The discipline-level optimisation problems sought to minimise violation in the design (or behaviour) constraints, while satisfying equality constraints on the coupling variables passed from the system to the discipline [26]. The need to satisfy equality constraints was identified as a source of numerical difficulties that occasionally arose in applications and different solutions were proposed. Schmit and Ramanathan [27], sought to minimise discrepancy in the coupling equations while satisfying the design constraints. In particular the key idea in the multi-level approach they presented is to select the component, “discipline” according to Sobieski terminology, level objective function so as to minimise disturbance of component level loading due to component level synthesis. This is accomplished by taking the change in stiffness as the component level objective function to be minimised [27]. Another approach has been employed by Biggers, Fu and Latour [23], where the constraint at one level is the objective function at the next level (see Table 1. of [23]).

Over a decade after the first Hierarchical Multi-level Decomposition approach was proposed by Sobieski [8], a more general decomposition was pioneered extending to non-hierarchic multidisciplinary systems [26]. In non-hierarchic systems, all disciplines are on the same level, and analysis and optimisation associated with the coordination of the system is implied. Figure 2-8 shows non-hierarchic systems, where each discipline may be coupled to every other discipline.

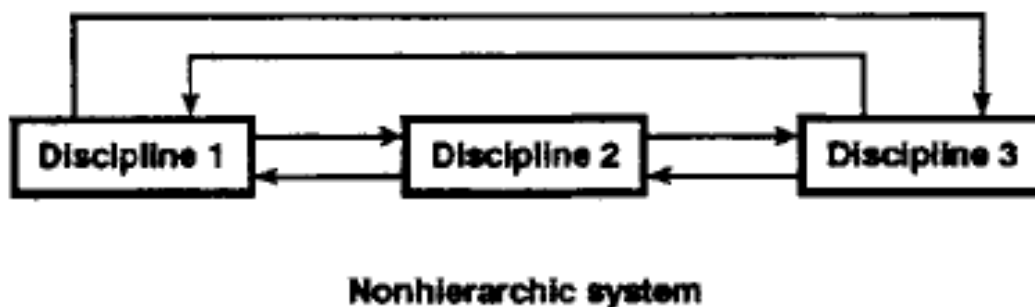
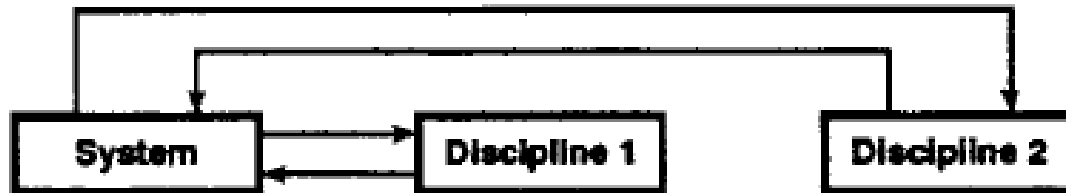


Figure 2-8 – Multi-level Decomposition: Non-Hierarchic System [26]

At the same time Sobieski proved, Figure 2-9, that the traditional hierarchic system may be viewed as a non hierarchic system by treating the “system” as a discipline on the same level as the other disciplines.



Hierarchical system viewed as a nonhierarchical system

Figure 2-9 – Multi-level Decomposition: Hierarchic and Non Hierarchic Systems [26]

In a multilevel decomposition algorithm, it is essential to estimate the sensitivity of a problem at its optimum to assumed constant parameters of the problem. This information provides for necessary coupling between various subsystems in the decomposition tree. The optimum sensitivity coefficients are essentially the Lagrange Multipliers. Numerically, they correspond to total derivatives of the objective function and the design variables with respect to the design parameters. A number of methods have been developed to compute the sensitivity coefficients directly. These methods include: Lagrange multiplier method, penalty function methods, feasible directions methods with the extension of the latter method to incorporate higher order coefficients and discontinuities [7].

Another decomposition methodology in this field is called “The Design Structure Matrix”. According to Browning [28] products, processes and organisations are each a kind of complex system. The classic approach to increasing understanding about a complex system is to model it, typically by:

- decomposing into subsystems
- noting the integration of the subsystems that give rise to the system’s behavior
- noting the external inputs and outputs and their impact on the system

With a reasonable model, it becomes possible to explore innovative approaches to system decomposition and integration. The Design Structure Matrix (DSM) is becoming a popular representation and analysis tool for system modelling, especially for purposes of decomposition and integration. A DSM displays the relationships between components of a system in a compact, visual, and analytically advantageous format. In Figure 2-10, a DSM is presented as a square matrix with identical row and column labels. In this example, elements are represented by the shaded elements along the diagonal. An off-diagonal mark signifies the dependency of one element on another. Reading across a row reveals the dependency/coupling between one element to another, whilst scanning down a column, reveals the coupling between elements.

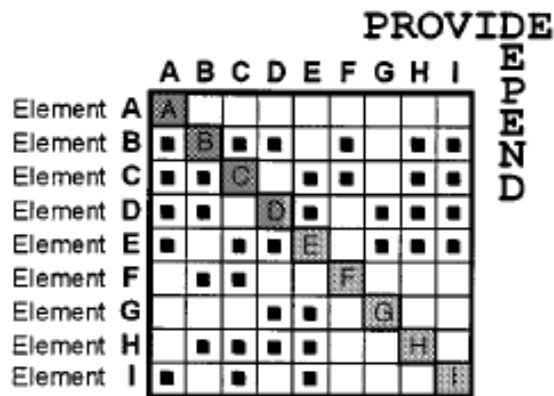


Figure 2-10 – A DSM Example [28]

That is, reading down a column reveals input sources, while reading across a row indicates output sinks. Thus, in Figure 2-10, element B provides something to elements A, C, D, F, H, and I, and it depends on something from elements C, D, F, and H.

The use of DSMs in both research and industrial practice increased greatly in the 1990s, but in the aerospace field it did not prove much technical applications but Andersson et al. [29]. In fact, DSM applications demonstrate the chief strength of matrix-based approaches with systems meant as a product, a process or an organisation.

2.6 Parallel Computation

The main advantage of the multilevel optimisation scheme lies in the fact that all substructures can be analysed and optimised independently with convenient coupling.

This application makes such an algorithm appropriate for parallel computing technology existing today [30].

In a broader sense, the terms parallel computing, or concurrent processing are used to define simultaneous execution of multiple tasks on multiple CPU's. This is achieved either by synchronising the tasks on a multiprocessor, or by effectively distributing the tasks among a network of computers. Parallel computing operations are classified in a number of ways depending upon the architecture of available computing resources and the granularity of the applications [30]. Granularity is measured by synchronisation interval which is in fact the period between synchronization events measured in number of instructions for multiple processors or processing elements [30].

Flynn [31] has divided computer architectures from a macroscopic point of view using a stream concept. Stream in this context simply means a sequence of items (instructions or data) as executed or operated on by a processor. The four broad classifications of machine organisations are:

- The Single Instruction Stream – Single Data Stream (SISD), which represents most conventional, single-processor computing equipments [31].
- The Single Instruction Stream – Multiple Data Stream (SIMD), which includes most array processors; for example, ILLIAC IV [31].
- The Multiple Instruction Stream – Single Data Stream (MISD) type organisations; for example pipeline computers like CYBER 205 [31].
- The Multiple Instruction Stream – Multiple Data Stream (MIMD) machines, which include the multiprocessor systems and distributed computing networks. It is possible to classify the MIMD architecture further according to coupling of the multiple processors as Tightly-Coupled and Closely-Coupled systems [31].

Various parallel computing applications are also classified based on granularity ranging from infinite grain size to very fine grain size. Concurrent processing computers set new demands on data structure, data management, organisation, program coding, and adaptability considerations. These computers offer the possibility for significant gains in

computational speed for structural synthesis based on multilevel optimisation. Experience in parallel processing on NASA Langley's first multiple instruction, multiple data (MIMD) computer has shown that the greatest computational gains are obtained by writing special-purpose codes based on "rethinking" the solution method; somewhat smaller gains have resulted from "recoding" an existing algorithm, and no gain has resulted from the approach of just running an existing program on a parallel computer [32].

Significant research effort has been reported in applying linear and nonlinear finite element algorithms on concurrent processing computers using sub-structuring methods. Kowalik [33] has projected the potential for impact of parallel computers on numerical algorithms. Lootsman and Ragsdell [34] attempted a very early research of multilevel decomposition code on a network of computers focusing on works in parallel of nonlinear optimisation.

But nowadays available optimisation and simulation codes are often tailored to special tasks, and due to a minor degree of modularity are hard to apply to other application areas. This leads to a demand for a high degree of modularity, which allows to exchange components or, e.g. to facilitate the parallel execution of the optimisation and simulation-code [35].

PLATON (Parallel Adaptive Techniques for Optimisation) [36] , a component based distributed software architecture used for coupling optimisation and simulation software, is the answer to this new challenge. PLATON has an abstract view of a function depending on design parameters, where values of the design parameters have to be found in order to minimise the function. Each function value is the result of a numerical simulation (e.g. a FEM program) for a given design parameters, and this abstract entity is called the Objective Function Component. The PLATON system provides the infrastructure for a seamless coupling of these components as described in Figure 2-11.

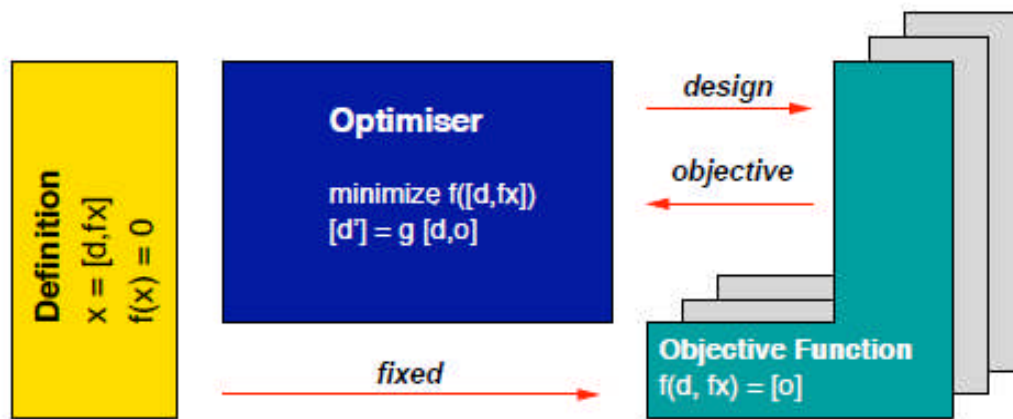


Figure 2-11 – PLATON abstract view of optimisation [36]

At the same time, PLATON is designed as a distributed component system which acts on a computational GRID [35]. Each component provides a well defined interface specifying its functionality. Within this system a component specifies a service which can be used by other software components. In this way, the resulting software system PLATON can be seen as a configuration of such reusable components [36]. In fact, PLATON is designed as an open system so that additional or new components can be integrated into the system, e.g. a visualisation tool, a database system and so forth. The following picture, Figure 2-12, shows optimisers and simulators coupled through the communication middleware CTL (Component Template Library).

The Component Template Library is an implementation of the component technology based on C++ generic template programming. Similar to CORBA (Common Object Request Broker Architecture), as a support framework of applications [35], it can be used to realise distributed component-based software systems, where a component is a piece of software which consists of a well defined interface and implementation. Interface and implementation are connected through a communication channel. In this way the usage of a component is independent of their location, meaning that the component is network transparent. Being a template library in its simplest configuration it requires nothing more than the standard libraries available on all UNIX platforms [15]. A relevant feature of this library is GALib. In fact the library includes tools for using genetic algorithms to perform optimisation in any C++ program using any

representation and any genetic operators. The flexibility of the whole communication middleware CTL makes possible to link GALib with inputs/outputs of any external package of FEM solvers.

This capability of parallel computation and all its tools might be employed in the Advanced Example at level2 and level3, Table 1-3, during the performing of sub-laminate and genetic algorithm concepts. Unfortunately, as previously mentioned, not applying any numerical application for this example will prevent any opportunity to understand strengths and weaknesses of CTL, PLATON, and GALib but it will set the basis for future work.

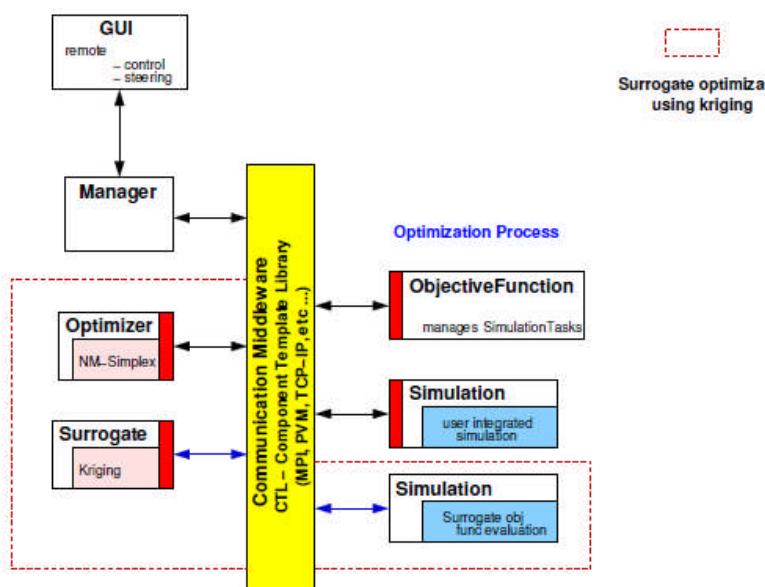


Figure 2-12 – The PLATON system [36]

2.7 Finite Element Modelling of Laminated Structures

Considering the flexibility of middleware communication software, such as PLATON, to integrate within a multi-level decomposition system, this allows the use of Finite Element Modelling Techniques for structural analysis and optimisation. The complexity of real life composite components, as aerospace primary structures, initially made use of simplified analysis models but then required a continuous necessity of increased and detailed analyses and FEM proved to be one of the main suitable tools to do it.

Primary flight structures such as wings, fuselages or empennages are mainly designed using stiffened panels, 2-D elements, under combined in-plane and out-of-plane loading as explained by Figure 2-13 and Figure 2-14.

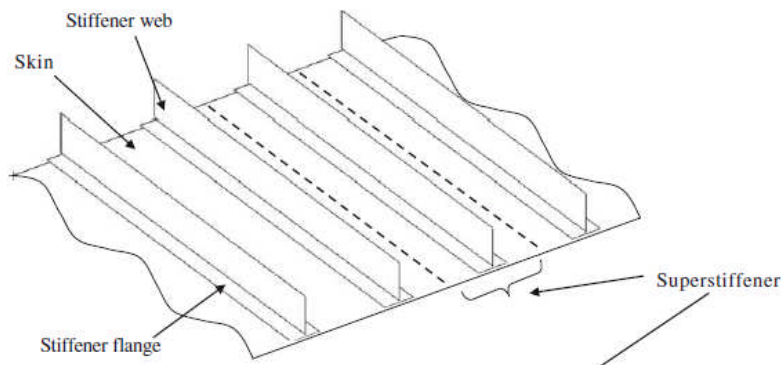


Figure 2-13 – Stiffened Panel [37]

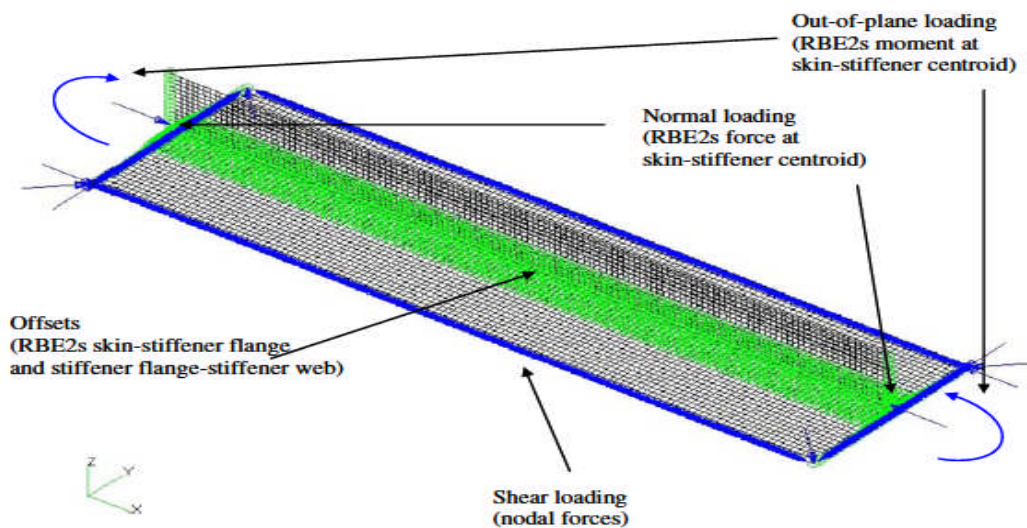


Figure 2-14 – Stiffened Panel 2D FEM [37]

But at the same time complicated three dimensional thick laminated composite structures are not just an issue in bioengineering, hip and femoral prosthesis Figure 2-15 and Figure 2-16, but it is part of aerospace fields too as showed in Figure 2-17, side stay bracket connecting MLG and Rear Wing Spar part of the ALCAS project [2], [4] and Figure 1-1.

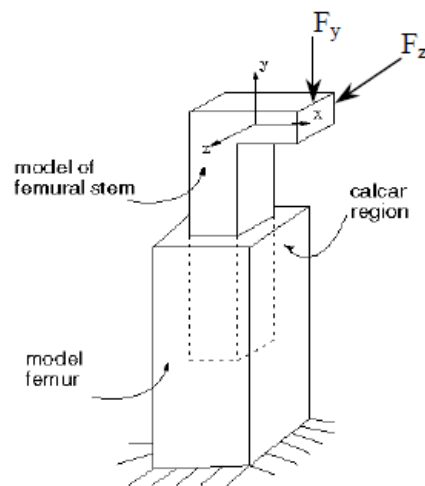


Figure 2-15 – Bioengineering Application: Femoral Schematic Model [4]

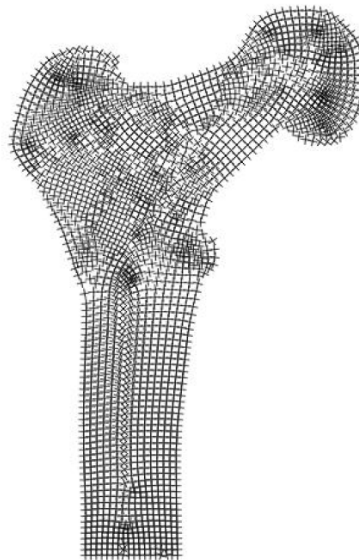


Figure 2-16 – Femoral 3D FEM [38]

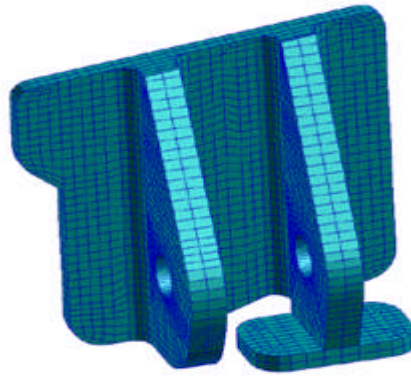


Figure 2-17 – Aerospace Application: Side Stay Bracket 3D FEM, Thesis Object [3]

Out-of-plane loading, unbalanced laminates, flexural anisotropy, instability effects (buckling, clipping and so forth), initial geometric imperfections and very thick laminates are few of the different conditions responsible to cause out-of-plane displacement in the final laminate. Classical Lamination Theory, as analytical method, and FEM ply-based modelling (MSC Laminate Modeller) do not take into account transverse shear and normal effects (more theoretical explanations will be given in section 2.8), reducing the applicability of these models to zones where these effects are not important for the modelling of the structural response [39].

So modelling thick composite structure using 2-D plane elements can cause inaccurate results since plane stress condition are not valid. Therefore, 3-D modelling should be used but the difficulties to model all the layers with different material properties and ply orientation arise. Rachmadini (et al.) [40], proposed an equivalent modeling by grouping layers with the same material and ply orientation. In such way the number of elements through the thickness is remarkably reduced and still the result is close enough to the one from a detail finite element model. This proposed modeling technique, MSC Patran used as pre/post processor and Nastran as solver, has been applied for analysis of hingeless composite rotor hub system designed by Korea Aerospace Research Institute (KARI) [40] as depicted in Figure 2-18.

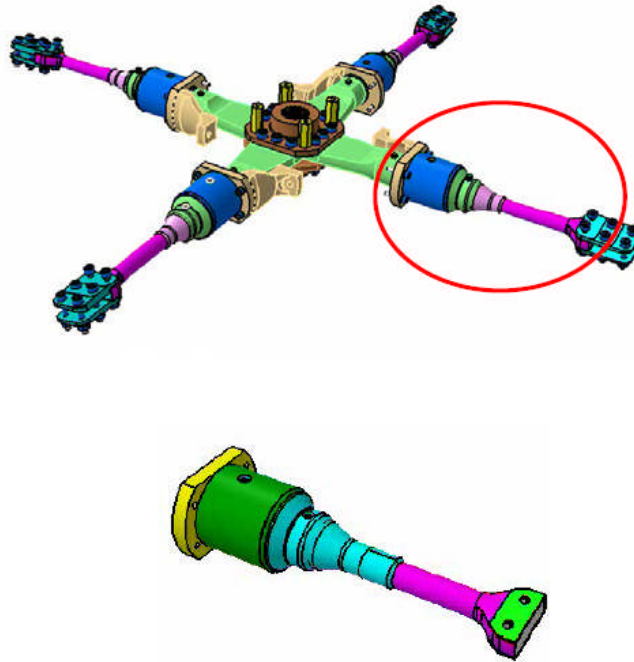


Figure 2-18 – Helicopter Hingeless Hub System and Parts being analysed [40]

The helicopter hingeless Hub Structure is made of eight individual components, Figure 2-19, that have different material properties.

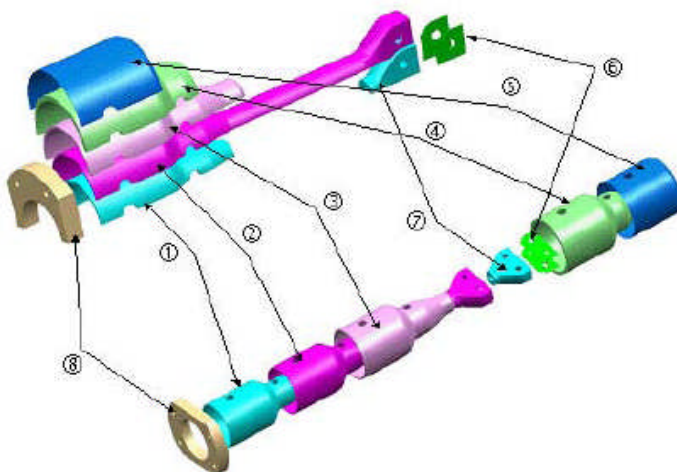


Figure 2-19 – Eight different components of the part under analysis [40]

Then the main flexure part, Figure 2-19, was modeled for finite element analysis by using solid elements, mainly HEXA8 of MSC/Nastran. Finite element modeling of the structure is shown in Figure 2-20.

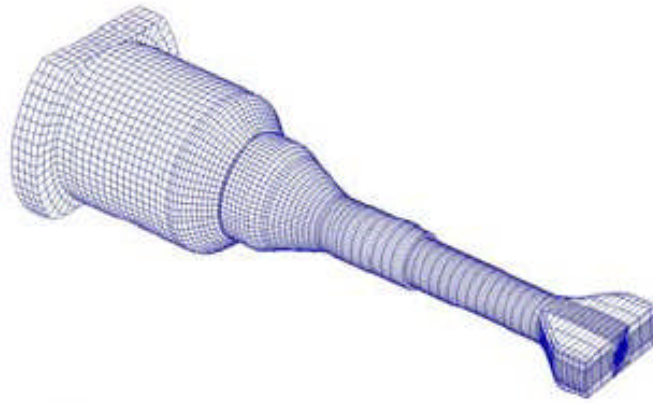


Figure 2-20 – Finite Element Model of the Structure: 3-D Elements [40]

Properties applied on the solid elements were the equivalent properties of each material or only one solid and then apply its equivalent properties. Equivalent properties can be obtained from the feature “Show Laminate Properties....” in MSC/Patran or by using classical laminate theory of composite [40].

The second option presented by Rachmadini [40] of a single homogeneous anisotropic solid, presents limitations since it is expected to have in-plane loading and then to evaluate the through-thickness elastic properties ($E_{33}, \nu_{12}, G_{13}, G_{23}$) that requires a specific approach.

One of those approaches is the essential core to represent thick laminates consisting of large numbers of a repeating sub-laminate. The typical sub-laminate is used for studying its load-deformation behaviour from which the effective elastic constants are derived from the constituent lamina properties.

Explicit expressions for effective elastic constants for general thick laminates are presented by Sun and Sijian Li [41]. In fact in their work, they started considering that in thick laminate applications, a certain periodic stacking sequence must be maintained in order to avoid warping due to the presence of curing stresses. If the characteristic length of deformation of the global composite laminate is large compared with the

periodicity, then the non-homogenous properties over each typical cell may be smeared out and effective properties used. Thus, all the cells are represented by the same anisotropic medium, and, consequently, the whole laminate can be effectively represented by homogenous anisotropic solid. The aforementioned long-wave approach was used by Postma [42], who considered a layered medium consisting of alternating layers of two different isotropic materials. The resulting effective medium was found to be a transversely isotropic solid.

Pagano [43] employed the long-wave approach to derive effective elastic moduli for anisotropic laminates. Instead of assuming the existence of periodic structuring through the laminate thickness, he considered a representative volume which contains the whole thickness of the laminate. Because of the finite thickness, moment effects were also included in the “stress-strain” relations.

In the Sun and Sijian Li [41] paper, thick laminates consist of large numbers of repeating sub-laminate, the typical cell according Postma terminology [41]. The thickness of the typical sub-laminate is assumed to be small when compared with the total laminate thickness, and the laminate is modelled as a three-dimensional homogeneous anisotropic solid whose effective moduli (in-plane and through the thickness) are derived from a representative volume, the typical sub-laminate. The sub-laminate is evaluated using constant stress and constant strain assumptions for its six independent modes of deformation. Explicit expressions for the three-dimensional effective moduli are presented alongside reduced expressions for balanced laminate of a single composite system [41].

Nowadays with a much stronger experimental support, Wisnom [44], proved the through-thickness elastic properties can be simply obtained through a transverse isotropy assumption. In fact considering “1” equals the fibre direction and “2” the transverse axis and “3” is the orthogonal through thickness direction, the isotropy assumption becomes:

$$E_{33} = E_{22} \quad , \quad \nu_{13} = \nu_{12} \quad , \quad G_{13} = G_{12} \quad , \quad G_{23} = \frac{E_{22}}{2(1 + \nu_{23})} \quad (\text{Eq. 2-1})$$

According to Wisnom [44], data is often not available for ν_{23} and it varies over a narrow range, but a value of 0.45 is usually reasonable estimate for unidirectional glass and carbon fibre-epoxy materials [45].

Wisnom [44],[45] and Sun and Sijian Li [43] homogeneity theories must be taken with extreme care, since can miss crucial aspects of behaviour as:

- Effects of non-symmetry, coupling -e.g. (0/90) laminate will bend under tension
- Laminate thermal residual stresses
- Free edge stresses
- Stresses at other discontinuities such as ply drops

2.8 Inter-laminar Stresses: Analytical and Numerical Methods

It is a well established fact that the inter-laminar stresses arise as a consequence to the above specific conditions and mainly to a mismatch in elastic properties between plies. Thus in these regions near the free edge known as boundary layers, it has been shown that the stress state is three-dimensional in nature and not predictable with classic analytical and numerical methods. In order to evaluate the 3D stress-field and the nature of stress concentration that occurs in composite laminates that have edge boundaries, a 3D elasticity boundary value problem must be solved. Unfortunately exact solutions to this problem are, as yet, unavailable. Thus numerous investigators have presented a variety of approximate methods to calculate the transverse/inter-laminar stresses at straight free-edges [46].

Reviewing the analytical methods, the first approximate solution of finite-width composite laminates was proposed by Puppo and Evensen [47], based on a laminate model containing anisotropic laminae and isotropic shear layers with inter-laminar normal stress being neglected throughout the laminate. Other approximate methods were also attempted to examine the problem such as extension of the high-order plate theory [48] by Pagano [49], the perturbation method by Hsu and Herakovic [50], a boundary layer theory by Tang [51] and Tang and Levy [52], and approximate elastic

solutions by Pipes and Pagano [53]. Later, based on assumed in-plane stresses and the use of Reissner's variational principle, Pagano developed an approximate theory [54], [55].

The most applied high-order plate theories mentioned previously [48] , [49], are first-order shear deformation plate theory (FSDT) and the third-order shear deformation plate theory (TSDT). The FSDT extends the kinematics of the classical lamination theory by relaxing the normality restriction (see previous paragraph) and allowing for arbitrary but constant rotation of transverse normals. The TSDT further relaxes the kinematic hypothesis by removing the straightness assumption; i.e., straight normal to the middle plane before deformation may become cubic curves after deformation [56].

The most significant difference between the classical and shear deformation theories is the effect of including transverse shear deformation on the predicted deflections, frequencies, and buckling loads. In fact as Reddy points out in [56], the CLT under-predicts deflections and over-predicts frequencies as well as buckling loads of plates with side-to-thickness ratios of the order of 20 or less. Lu and Liu [57], obtained the inter-laminar shear stress directly from the constitutive equations, but variationally consistent so that it can also be used for finite element formulation. This led to the Inter-laminar Shear Stress Continuity Theory (ISSCT) that due to the neglecting deformation in the thickness direction, it did not calculate the inter-laminar normal stress directly from the constitutive equation [46]. Many other investigators kept developing a variety of methods to attempt to calculate these stresses at straight free edges. Review of literature of these analytical methods is available in articles by Noor and Burton [58], [59] but the reality says that the storage requirements due to the large number of variables and computer costs make them less attractive for use by industry. In fact because of the complexities of analytical solutions for the prediction of transverse /inter-laminar stresses, more emphasis has been placed on the use of numerical methods [46].

A common feature of all analytical methods is that they can only be used for the simplistic geometric cases, since for thick realistic structural laminates; the solution to the full 3D problem is extremely complex. Thus a variety of numerical methods, e.g., finite difference and finite element have been developed to calculate these inter-laminar stresses at straight free edges. These methods not only provide the option of placing a refined mesh near regions of possible stress concentration, but can also be used with

ease for the accurate analysis of laminated composite structures having complicated geometry and/or loading [46].

The initial approach adopted to analyse composite laminates of finite size subjected to external loads was a 2D finite element method. Though the 2D elements can yield accurate results at locations away from the traction free-edges and discontinuities, it cannot predict accurately the complex stress state near any geometric or material discontinuities or with very thick laminates subject to out-of-plane loading or near a traction free-edge [60]. Once the limitation of the 2D technique became known and more powerful computers became available, 3D finite elements became increasingly used [61][62][39].

For many applications however, a full 3D analysis can be a waste of resources. In view of the above facts many investigators attempted global/local finite element analysis as Fu [23] dealt with femoral component for hip joint arthroplasty and Mao [63] Hirai [64] developed an efficient zooming method for finite element complex structure. Thomson and Griffin [65] extended the same approach and proposed a 2D-3D global/local finite element analysis. They subdivided the entire laminate into local and global regions, the local region being the traction free-edges, and the area around geometric or material discontinuities, and the global region is location far away from local region. They used a simplified 2D finite element analysis on a global region and a more detailed 3D finite element analysis on a local region. They were able to demonstrate that the global/local analysis technique yields a reasonably economical solution by achieving considerable savings in computer time and storage as compared to a complete 3D finite element analysis [46].

2.9 Summary

Identify the main evolutionary steps of the structural optimisation of laminated components helped to point out the main limiters as model, analysis and type of optimisation requested. These three main constraints are considered by Venkataraman and Haftka [5] as the complexity factors of structural optimisation of composite materials. Making the most of their experience came out that it is reasonable to deal with high complexity index in only one of the three components at the time. So, static

analysis and basic optimisation techniques are the main characteristics of this research once trying to define a framework capable to deal at the very end with very complex model such as three dimensional laminated structures. Then, from their research it is also possible to understand that the 3-D finite element analysis was mainly explored by the bioengineering instead the aerospace field. In fact, the aerospace discipline through the the rapid increases in computer processing, dealt with large number of degrees of freedom due mainly to full wing, fuselage and detailed models instead focussing on three dimensional laminated structures.

On the other hand, the bioengineering field removed this lack of interest of the aerospace, proving to be one of the first to pioneer this specific topic of thick composites, proposing the use of methods, multi-level decompositions (MD), earlier introduced by Sobieski for decomposing large optimisation problems encountered in the design of engineering systems into a number of smaller subproblems. This capability showed to be crucial to manage the difficulties of thick laminated composite structures such as large number of design variables, many thin layers orientated at various angles, highly nonlinear nature of the expressions relating structural performance, and final consequent huge computational demand.

This thesis is just meant to explore the potentialities of multi-level optimisation schemes applying it at two dedicated case studies, Basic and Intermediate examples. Furthermore, these numerical applications will show the flexibility to embed at each level the suited optimisation techniques and the needed structural calculation tools, Classical Lamination or Finite Element analyses.

Finite element techniques for thin and thick laminated components have been described to pointing out the most suited techniques to apply in the current applications. Equivalent modelling has been chosen, see Basic and Intermediate examples, to overcome the difficulties to model all the layers with different material properties and ply orientations typical of thick materials.

Then, this chapter dedicated few sections to understand and describe issues typical of thick laminated structures, (interlaminar stresses, sub-laminates, out-of-plane effects and parallel computation) despite at the end of the whole work have not been fully employed. But they were needed to define the frame work of the Advanced Example as starting point of future applications of thick laminated aerospace structures.

3 Laminated Composite Materials

3.1 General

Basically, we can consider a composite as any material that is a combination of two or more distinct constituents. This definition would encompass bricks, concrete, wood, bone, as well as modern synthetic composites such as fibre-reinforced plastics (FRP). The latter have become increasingly important over the past 50 years, and are now the first choice for fabricating structures where low weight in combination with high strength and stiffness are required. Such materials are sometimes referred to as high-performance composites, and would often be composed of carbon fibres and epoxy resin. Having said that, composite can also be made by combining fibres with a metal or a ceramic matrix, but at the moment these have a very small market share and would be considered as specialist materials [13].

3.2 Classification of Composite Materials

The general classification of composite materials is based on the nature of the constituent materials. The constituents may be either organic or inorganic. The designation of organic refers to materials originating from plants or animals or materials of hydrocarbon origin (natural or synthetic) such as carbon. Inorganic materials are those that cannot be classified as organic matter, for example, metals and minerals. Historically, perhaps the most common structural material of organic nature has been wood. Wood is a composite material because it is composed of two distinct

constituents: stiff and strong fibres surrounded by a supporting structure of softer cells. In fact, most modern composite materials imitate wood in that they consist of strong fibres embedded in softer supporting material. The supporting material in such an arrangement is commonly referred to as matrix material [1].

For most highly loaded single-material structures, the use of purely organic materials is avoided because of their generally low stiffness and strength properties. They are also sensitive to environmental effects such as temperature and moisture and have low resistance to chemicals and solvents. Despite their disadvantages, organic materials such as polymers are commonly included in composites to bring in certain characteristics that may not be achievable by using only inorganic constituents. For inorganic composites, the constituents may be all metallic, all non-metallic, or a combination of the two. The most common inorganic materials used for highly loaded structures are metals. Despite their good stiffness and strength properties, metals have high specific weight, a disadvantage in weight-critical applications, such as aircraft and spacecraft [1]. Therefore, metallic composites are mainly used when there is specific properties need, such as resistance to high temperatures.

So most frequently used composite materials, for aircraft and spacecraft structural applications where high strength –to-weight and stiffness-to-weight ratios are required, include polymeric constituents as well as inorganic glasslike substances [67].

Polymer matrix materials are made of long chain or organic molecules, generally classified as thermoset or thermoplastic. The two categories have substantially different characteristics as matrix materials. Fibres preimpregnated with thermoset matrix (thermoset prepregs) are soft and malleable. They are tacky and can be draped easily, hence allowing fabrication of complex shapes with ease. They undergo a chemical change (cross-linking of polymer chains) during curing, and the process is irreversible. Thermoplastics, on the other hand, do not require curing and they can be reshaped and reused. However, thermoplastic prepregs are hard and boardlike and lack drape and tack, making them harder to handle. They also require very high processing temperatures to shape them, although processing time can be very short. Thermoplastics also offer high resistance to moisture and can operate at higher temperatures of up to 500 ° F (260 °C) compared to thermoset operating temperatures of about 250 °F (121 °C) [1].

A more complex discussion of advantages and disadvantages of thermosets and thermoplastics is provided by Niu in his specific study of Composites in airframe structures [67].

Haftka et al. [1], present a more traditional classification than the one based on the nature of the constituents derived from their form.

- Particulate Composite
- Flake Composite
- Fibre Reinforced Composite
- Laminated Composite

Particulate composites are generally made up of randomly dispersed hard particle constituents in a softer matrix. Examples of particulate composites are metal particles in metallic, plastic, or ceramic matrices. A widely used particulate composite is concrete in which gravel is mixed in cement.

Flake composites, as the name suggests, are formed by adding thin flakes to the matrix material. Although flake dispersion in the matrix is generally random, the flakes may be made to align with one another, forming a more orderly structure compared to particulate composites. Typical examples of flake materials are glass, mica, metals and carbon.

Fibre-reinforced composites, FRC, (or fibrous composites) are the most commonly used form of the constituent combinations. The fibres of such composites are generally strong and stiff and therefore serve as the primary load-carrying constituent. The matrix holds the fibres together and serves as an agent to redistribute the loads from a broken fibre to the adjacent fibres in the material when fibres start failing under excessive loads. This property of the matrix constituent contributes to one of the most important characteristics of the fibrous composites, namely, improved strength compared to the individual constituents [1].

The last form of composite materials is thin layers of material fully bonded together to form so-called composite laminates. The layers of a laminated composite material may be different single materials, such as clad metals that are bonded together or the same material, such as wood put together with different orientations. The layers may

composites themselves, such as FRC layers placed so that different layers have different characteristics.

This type of composite is the most commonly encountered laminated composite material used in design of high-performance structures and is the primary subject of this Thesis.

3.3 Fibre-Reinforced Composite Materials

An individual layer of a laminated composite material may assume a number of different forms, depending on the arrangement of the fibre constituent. The layers may be distributed as short fibres embedded in a matrix. The short fibres may be distributed at random orientations, or they may be aligned in some manner forming oriented short-fibre composites [1].

Continuous fibre-reinforced composite layers are made up of bundles of small diameter circular fibres. Typically, the radii of these fibres are in the order of 0.0002 in (0.005 mm), such as the radius of carbon fibres. The largest diameter fibres, such as boron fibres, are of the order of 0.002 in (0.05 mm). Continuous fibre-reinforced composite materials are commercially available in the form of unidirectional type, with fibres aligned along the length of the tape. The fibres of the tape are preimpregnated with the matrix material, and for this reason the tape is sometimes referred to as a prepreg. The arrangement of aligning the fibres in a given direction provides the unique feature of the material properties of fibre-reinforced composites [1].

Another form of continuous fibre-reinforced composite layers is the woven fabric type composite, where fibre tows, which are large bundles of fibres (generally 10,000 or more fibres), are woven in two or more directions. Various fabric types with different weave features producing different material characteristics are available [68].

3.4 Fibre-Reinforced Composite Materials: Properties and Applications

Density, stiffness and strength are the properties that initially come to mind when thinking of FRP, and these would certainly be the design drivers for materials selection for transport applications such as aircraft, motor vehicles and trains. However this is a very narrow view of the potential of such materials, and they often score over metals and other conventional materials because of other mechanical, physical or chemical properties [67].

For instance, FRPs are extremely corrosion-resistant and are, in consequence, used for chemical plant, water transport and storage, and flue gas desulphurisation plant. They have interesting electromagnetic properties and because of this glass fibre-reinforced resin is used to construct mine counter measures naval vessels, the requirement being a non-magnetic material. As another illustration, carbon fibre-reinforced epoxy is used in medical application because it is transparent to X-rays; bioengineering applications mentioned in the literature review section are good examples [4][23][24]. Thermal properties can also be important; such materials have a zero coefficient of expansion, hence making possible the construction of temperature-stable components [67].

A key issue is that of cost. Although the constituents of FRP are relatively expensive, compared with conventional materials, the final component may well be less costly than one fabricated from metal. This is due, in part, to having a more integrated construction (lower “parts count”), but depends crucially on the level of automation in the manufacturing process [67].

3.5 Laminated Composite Definition

The laminate composites considered in this research are primarily laminates made up of continuous-fibre preregs. However, the level of analysis used in this thesis does not really differentiate between the woven and the unidirectional continuous composite layers, as long as material characteristics that properly define the layer are used.

Laminate composite material where the layers of unidirectional fibre-reinforced composites are stacked on top of one another are described according to a standard notation called stacking sequence, which is described in the following.

The stacking sequence lists fibred orientations measured from a reference axis of the laminate. If the orientation is counter-clockwise from the reference direction, it is

considered to be positive. The standard stacking sequence lists orientations of the different layers, starting from the top of the laminate to the bottom, in a string separated by slashes. For a laminate with N layers, each made of the same composite material and the same thickness (typical configuration adopted in this work), t , starting with the top layer with a fibre orientation ϑ_1 , the laminate is represented as:

$$[\vartheta_1 / \vartheta_2 / \dots / \vartheta_N]$$

Layers oriented at an angle from the reference axes of the laminate are called off-axis layers. When the orientation of a layer coincides with one of the reference axes of the laminate, $\vartheta = 0^\circ$ or $\vartheta = 90^\circ$, that layer is referred to as an on-axis layer [1].

When several layers with the same orientation are adjacent to one another, it is common to group them together and represent the total number of adjacent layers with the same orientation as a subscript to that particular orientation. When a group of layers is repeated, then the number of repetitions is used as a subscript to the repeating group enclosed in parenthesis. For the following laminate, $[0_2 / (0_2 / 45_2 / 90)_3 / 0_2]$, sandwiched between the two layers of 0° on the top and two layers of 0° at the bottom, the $0_2 / 45_2 / 90$ group is repeated three times [1].

A laminate is symmetric when the fibre orientations of the bottom half of the laminate are mirror images of the fibre orientations above the mid-plane of the laminate, for example, $[-45/30/0/45/45/0/30/-45]_T$ (the subscript T is used after bracket to indicate that the designation is for the total laminate). Symmetric laminates with an even number of layers are represented by the portion of the stacking sequence above the laminate mid-plane followed by a subscript s after the closing bracket, $[-45/30/0/45]_S$. If the layers within the brackets are repeated, the number of repetitions can also be placed after the bracket before the subscript s. That is, the laminate $[45/0_2/45/0_2/45/0_2/0_2/45/0_2/45/0_2/45]_T$ is represented in a compact form as $[45/0_2]_{3S}$.

In some situations it is desirable to place a negative ϑ orientation for every occurrence of the positive ϑ layer. Such laminates are referred to as a balanced laminates. Pairs of

positive and negative orientations do not have to be placed adjacent to each other, but if they are adjacent, the laminate designation may be condensed by putting a plus-minus sign in front of the orientation angle. For example, the $[30/-30/30/-30]_T$ laminate is simply designed as $[\pm 30_2]_T$. The $[-30/30/30/-30]_T$ stacking sequence, on the other hand, is designated by $[\mp 30]_S$ [1].

3.6 Laminated Composites Structural Analysis

3.6.1 Classical Analysis

The use of classical (continuum) methods of stress analysis has developed over many decades to give techniques that can be applied satisfactorily to a vast range of situations. Such analyses are based on the application of the equations of equilibrium and compatibility, together with the stress-strain relations for the material, to produce governing equations which must be solved to obtain displacements and stresses [13].

Elastic properties that form the relation between the stresses and strains are the fundamental quantities that govern the response of the material and, therefore, the structure to be designed. For this reason the objective of the next sections is to discuss how elastic properties are calculated for laminated composite materials from variables that can be changed during the design process [1].

The starting point it will be introducing the fundamental relations that control the linear elastic response of an isotropic medium in three dimensions and demonstrate how they are reduced to two-dimensional relations when two or more of these isotropic layers are laminated together. Then it will be extended the treatment to the case of layers with directional properties.

3.6.1.1 Basic Stress-Strain Relations

First it is worth introducing the strain-displacement relations. Under loading, an elastic medium undergoes displacements that vary over the domain. These displacements along

the x , y and z coordinate axes denoted as u , v and w respectively. For small deformations, the following linear form of the strain displacement relations holds:

(Eq. 3-1)

$$\begin{aligned}\varepsilon_x &= \frac{\partial u}{\partial x}, & \varepsilon_y &= \frac{\partial v}{\partial y}, & \varepsilon_z &= \frac{\partial w}{\partial z} \\ \gamma_{yz} &= \frac{\partial w}{\partial y} + \frac{\partial v}{\partial z}, & \gamma_{zx} &= \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}, & \gamma_{xy} &= \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\end{aligned}$$

Where $\varepsilon_x, \varepsilon_y, \varepsilon_z$ denote the normal strains in the x, y, z directions, respectively and $\gamma_{xy}, \gamma_{yz}, \gamma_{zx}$ denote shear strains.

In addition to the strain-displacement relations, solution of elasticity problem requires the relations between stresses and strains in the medium. The stress-strain relation for a three-dimensional anisotropic linear elastic medium, also known as Hooke's law, is expressed in the following matrix form:

(Eq. 3-2)

$$\sigma = C\varepsilon$$

(Eq. 3-3)

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{yz} \\ \tau_{zx} \\ \tau_{xy} \end{Bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} \\ C_{12} & C_{22} & C_{23} & C_{24} & C_{25} & C_{26} \\ C_{13} & C_{23} & C_{33} & C_{34} & C_{35} & C_{36} \\ C_{14} & C_{24} & C_{34} & C_{44} & C_{45} & C_{46} \\ C_{15} & C_{25} & C_{35} & C_{45} & C_{55} & C_{56} \\ C_{16} & C_{26} & C_{36} & C_{46} & C_{56} & C_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \gamma_{yz} \\ \gamma_{zx} \\ \gamma_{xy} \end{Bmatrix}$$

Where, because of symmetry, there are 21 independent material constants. The matrix C is the material stiffness matrix, and σ and ε represent vectors of stress and strain components. There are three other stress components τ_{yz}, τ_{zx} and τ_{xy} . On these three stresses it is worth highlighting that satisfy the following identities:

(Eq. 3-4)

$$\tau_{xy} = \tau_{yx}, \quad \tau_{xz} = \tau_{zx}, \quad \tau_{yz} = \tau_{zy}$$

The above equations are a direct consequence of the equilibrium equations for the moments of the stresses about each of the coordinate axes. Note that those equations apply everywhere in the material and are independent of the assumptions for the material behaviour or the strain-displacement relations.

Another particular condition related to the Hooke's law main equation is coming from the existence of a strain energy density U . In fact as diffusely explains Boresi [69], for example, upon differentiation of the strain energy density, one can shows:

$$\frac{\partial U}{\partial \varepsilon_x} = \sigma_x, \quad \frac{\partial U}{\partial \varepsilon_y} = \sigma_y \quad (\text{Eq. 3-5})$$

Further differentiation of the stresses, σ_x and σ_y , with respect to the strains, ε_y and ε_x , respectively, yields:

$$\frac{\partial \sigma_x}{\partial \varepsilon_y} = \frac{\partial \sigma_y}{\partial \varepsilon_x} \quad (\text{Eq. 3-6})$$

From the above identity it is easily understandable:

$$C_{12} = C_{21} \quad (\text{Eq. 3-7})$$

And in more general fashion:

$$C_{ij} = C_{ji} \quad (\text{Eq. 3-8})$$

The final equation proves the symmetry of the material stiffness matrix.

In the case of a three-dimensional isotropic material, which has the same properties in every direction, (Eq. 3-3) reduces to the following form:

(Eq. 3-9)

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{yz} \\ \tau_{zx} \\ \tau_{xy} \end{Bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{12} & 0 & 0 & 0 \\ C_{12} & C_{11} & C_{12} & 0 & 0 & 0 \\ C_{12} & C_{12} & C_{11} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{C_{11}-C_{12}}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{C_{11}-C_{12}}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{C_{11}-C_{12}}{2} \end{bmatrix} \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \gamma_{yz} \\ \gamma_{zx} \\ \gamma_{xy} \end{Bmatrix}$$

In equation 3-9 only two independent material constants are present. The stiffness coefficients C_{11} and C_{12} are related to modulus of elasticity (elastic modulus or Young's Modulus) E , and Poisson's ratio ν by:

(Eq. 3-10)

$$C_{11} = \frac{(1-\nu)E}{(1+\nu)(1-2\nu)}, \quad C_{12} = \frac{\nu E}{(1+\nu)(1-2\nu)}$$

The diagonal terms that relate the shear strains to the shear stresses are commonly referred to as the shear modulus G and are obtained in terms of the elastic modulus and Poisson's ratio as:

(Eq. 3-11)

$$C_{44} = C_{55} = C_{66} = \frac{C_{11}-C_{12}}{2} = \frac{E}{2(1+\nu)} = G$$

Depending on the degree of directionality of the material properties, there are other special cases of material response between the fully anisotropic case defined by (Eq. 3-3) and the simplest case of isotropic behaviour defined by (Eq. 3-9).

3.6.1.2 In-plane Response of Isotropic Layer(s)

Thin structures made up of thin layer(s) of material can carry loads primarily through two different types of mechanisms. One involves stretching or compression of the

layer(s) in their own plane. This mechanism is normally referred to as the membrane action. The other mechanism involves bending of the layer(s). In this thesis the main examples, Basic and Intermediate consider basically in-plane problem where all deformations are in the plane of the layer(s). This holds for two dimensional plate problems where all loading is limited to the plane of the plate, and the loads do not have any bending component. Following the customary notation the z axis denote the direction normal to the layers.

In this contest the stress type to describe mathematically is the “Plane Stress”.

If there are no forces acting on the z -normal surfaces of the layers, then from equilibrium equations the: σ_z, τ_{zx} and τ_{yz} stresses are zero at these surfaces. For a thin layer, it is reasonable to assume that these stress components are zero throughout the layer:

$$\sigma_z = 0, \quad \tau_{yz} = \tau_{zx} = 0 \quad (\text{Eq. 3-12})$$

The remaining stresses, σ_x, σ_y and τ_{xy} , are all in the plane of the layer(s); hence, the state of stress is called plane stress.

In this configuration the Hooke's law for an isotropic material, (Eq. 3-3), can also be simplified. The fact that no normal stress exists in z direction does not imply that the corresponding strain ε_z is zero. Substituting $\sigma_z = 0$ into (Eq. 3-3), and solving for ε_z in terms of ε_x and ε_y terms together, it leads to the stress strain relation for plane stress case as:

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \begin{bmatrix} c_{11} - \frac{\nu}{1-\nu} c_{12} & c_{12} - \frac{\nu}{1-\nu} c_{12} & 0 \\ c_{12} - \frac{\nu}{1-\nu} c_{12} & c_{11} - \frac{\nu}{1-\nu} c_{12} & 0 \\ 0 & 0 & \frac{E}{2(1+\nu)} \end{bmatrix} \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} \quad (\text{Eq. 3-13})$$

Substituting the C_{11} and C_{12} from (Eq. 3-10) the new stiffness material matrix yields:

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \begin{bmatrix} Q_{11} & Q_{12} & 0 \\ Q_{12} & Q_{11} & 0 \\ 0 & 0 & Q_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} \quad (\text{Eq. 3-14})$$

$$Q_{11} = \frac{E}{1-\nu^2}, \quad Q_{12} = \frac{\nu E}{1-\nu^2}, \quad Q_{66} = G = \frac{E}{2(1+\nu)} \quad (\text{Eq. 3-15})$$

In matrix notation (Eq. 3-14) is generally represented as:

$$\sigma = Q\varepsilon \quad (\text{Eq. 3-16})$$

The matrix Q is commonly referred to as the reduced material stiffness matrix. The same notation of (Eq. 3-16) will be adopted to describe the plane state of orthotropic layer(s) where the definitions of the Q_{ij} terms in (Eq. 3-15) are modified to take into account the directional nature of the stiffness properties.

3.6.1.3 Symmetrically Laminated Isotropic Layers

When several layers are stacked together, a set of additional assumptions is needed in order to derive the equations that govern the constitutive behaviour of the laminate. In addition to the layers being in a state of plane stress, see 3.6.1.2, each layer is assumed to be bonded perfectly to the adjacent layers, so that the laminated layers deform in unison without experiencing any discontinuity in displacements [1].

Most commonly used laminates, and mainly the laminate considered in this research, have layers with mirror image elastic properties with respect to their mid-plane. That is, the stacking sequence of the layers (the elastic properties) above the mid-plane is the mirror image of the one below it. Such laminates are referred to as symmetric laminates

or mid-plane symmetric laminates. For symmetric laminates, properly distributed in-plane loads must be applied in such a manner that their resultants are the mid-plane of the laminate. Any offset from the mid-plane would cause bending moments with respect to it and force the laminate to bend [1].

The in-plane deformation assumption together with the plane-stress assumption means that the strains and the stresses are constant throughout the thickness of each layer. The perfect bond assumption guarantees that the strains, not the stresses unless layer properties are the same, are constant through the entire thickness of the laminate and, therefore, for the laminate strains could be characterised by the strain in a single layer, say the mid-plane layer, represented by ε^0 [1].

Having a constant strain distribution through the thickness does not imply, however, a constant stress distribution. Although the stresses are constant within each layer, they will vary from one layer to the next depending on the stiffness of the individual layers, which in this section are meant to be isotropic, following the stress-strain relation of (Eq. 3-16) as :

$$\sigma_{(k)} = Q_{(k)} \varepsilon^0 \quad (\text{Eq. 3-17})$$

where k is the layer number. This variation is the main reason for the common use of symmetric stacking sequence of the layers [1]. In fact for an arbitrary stacking sequence, the stress distribution through the entire laminate thickness given by (Eq. 3-17) will have a net bending moment. Such a moment will force the laminate to bend, unless the bending of the laminate is prevented by some external means to keep the laminate in its plane. The possibility of bending is precluded if we limit ourselves to laminates that are symmetric with respect to the laminate mid-plane.

With regard to overall stiffness, the laminate behaves as a single layer with properties that are some averages of the properties of the individual layers. So in order to define a simple stress-strain relation it is needed a quantity that represents the overall effect of a stress component on the laminate response. This is achieved by through-the-thickness-integration of the stress components. That is, for a laminate of thickness h, the following is introduced:

(Eq. 3-18)

$$\begin{Bmatrix} N_x \\ N_y \\ N_{xy} \end{Bmatrix} = \int_{-h/2}^{h/2} \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} dz$$

Where N_x and N_y are in-plane normal stress resultants, and N_{xy} is the shearing stress resultant. Since the stress resultants are obtained by through-the-thickness integration of stresses, dividing them by the laminate thickness h , yields average laminate stresses,

(Eq. 3-19)

$$\begin{Bmatrix} \bar{\sigma}_x \\ \bar{\sigma}_y \\ \bar{\tau}_{xy} \end{Bmatrix} = \frac{1}{h} \begin{Bmatrix} N_x \\ N_y \\ N_{xy} \end{Bmatrix}$$

Since the stresses are constant within a layer, integration can be replaced by summation over individual layers,

(Eq. 3-20)

$$\begin{Bmatrix} N_x \\ N_y \\ N_{xy} \end{Bmatrix} = \sum_{k=1}^N \int_{z_{k-1}}^{z_k} \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix}_{(k)} dz = \sum_{k=1}^N \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix}_{(k)} (z_k - z_{k-1})$$

Where N is the number of layers, the subscript (k) denotes quantities in the k th layer, and z_k 's are the through-the-thickness locations of the interfaces between the layers. Substituting the stress-strain law, (Eq. 3-17), it is possible to obtain:

(Eq. 3-21)

$$\begin{Bmatrix} N_x \\ N_y \\ N_{xy} \end{Bmatrix} = \left(\sum_{k=1}^N \begin{bmatrix} Q_{11} & Q_{12} & 0 \\ Q_{12} & Q_{11} & 0 \\ 0 & 0 & Q_{66} \end{bmatrix}_{(k)} (z_k - z_{k-1}) \right) \begin{Bmatrix} \epsilon_x^\circ \\ \epsilon_y^\circ \\ \gamma_{xy}^\circ \end{Bmatrix}$$

where the subscript “o” is used to denote mid-plane strains (recall that strains are constant through the thickness and can be represented by the mid-plane values). The (Eq. 3-21) can be written as:

$$\begin{Bmatrix} N_x \\ N_y \\ N_{xy} \end{Bmatrix} = \begin{bmatrix} A_{11} & A_{12} & 0 \\ A_{12} & A_{11} & 0 \\ 0 & 0 & A_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_x^o \\ \varepsilon_y^o \\ \gamma_{xy}^o \end{Bmatrix} \quad (\text{Eq. 3-22})$$

Where the coefficients of the matrix A are given by

$$A_{ij} = \sum_{k=1}^N Q_{ij(k)} (z_k - z_{k-1}) \quad (\text{Eq. 3-23})$$

$$A_{11} = A_{22} = \sum_{k=1}^N \frac{E_{(k)}}{1 - \nu_{(k)}^2} t_{(k)}, \quad A_{12} = \sum_{k=1}^N \frac{\nu_{(k)} E_{(k)}}{1 - \nu_{(k)}^2} t_{(k)}, \quad A_{66} = \sum_{k=1}^N G_{(k)} t_{(k)} \quad (\text{Eq. 3-24})$$

And where $t_k = (z_k - z_{k-1})$ is the thickness of the kth layer. For isotropic laminates concerned in this section, the A_{22} term is identical to A_{11} term. The matrix A is generally referred to as the extensional material stiffness matrix or in-plane stiffness matrix and (Eq. 3-22) written in matrix notations has the following form:

$$N = A \varepsilon^o \quad (\text{Eq. 3-25})$$

Once the in-plane stiffness matrix is determined, effective engineering constants of a laminate can be obtained by relating the average stress, (Eq. 3-19), in the laminate to the strains.

(Eq. 3-26)

$$\begin{Bmatrix} \bar{\sigma}_x \\ \bar{\sigma}_y \\ \bar{\tau}_{xy} \end{Bmatrix} = \frac{1}{h} \begin{Bmatrix} N_x \\ N_y \\ N_{xy} \end{Bmatrix} = \frac{1}{h} \begin{bmatrix} A_{11} & A_{12} & 0 \\ A_{12} & A_{22} & 0 \\ 0 & 0 & A_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_x^\circ \\ \varepsilon_y^\circ \\ \gamma_{xy}^\circ \end{Bmatrix}$$

The (Eq. 3-26) is identical in form to the stress-strain relation of (Eq. 3-14): therefore, by replacing the elastic properties in (Eq. 3-15) with unknown effective properties, E_{eff} and ν_{eff} , it is possible to state the following equations in terms of the two unknowns for effective properties:

(Eq. 3-27)

$$\frac{E_{eff}}{1-\nu_{eff}^2} = \frac{A_{11}}{h}, \quad \frac{\nu_{eff} E_{eff}}{1-\nu_{eff}^2} = \frac{A_{12}}{h}$$

Which yield

(Eq. 3-28)

$$E_{eff} = \frac{1}{h} \left(\frac{A_{11}A_{22} - A_{12}^2}{A_{22}} \right), \quad \nu_{eff} = \frac{A_{12}}{A_{22}}$$

The effective shear modulus is obtained simply as

(Eq. 3-29)

$$G_{eff} = \frac{A_{66}}{h}$$

The effective properties equations, described through their main steps [1], are the core part of the Basic and Advanced Example described in Chapters 5 and 6. The final formulation will extend this approach as the previous expressions are for isotropic material laminates only. The modifications will be fully presented in section 3.6.1.5.

3.6.1.4 Stress-Strain Relations for Orthotropic Layers

Replacing isotropic layers by fibre-reinforced layers affects only the layer stress-strain relations without requiring any changes in the assumptions that form the basis for the laminate constitutive relations previously described for isotropic layers. However, individual layers then have directional properties with planes of symmetry that are not necessarily oriented along the general x-y axes of the laminated medium. For a unidirectional fibre-reinforced lamina, there are two perpendicular planes of symmetry that define two principal axes of material properties. These principal axes correspond to the direction of the fibres and a direction transverse to the fibres, denoted by subscripts 1 and 2 respectively, in the next paragraphs. Such layers are defined to be orthotropic, and axes 1 and 2 are the principal axes of orthotropy [1].

For an orthotropic lamina, the stress-strain relation in the principal material directions is given by the following set of equations with nine independent constants:

(Eq. 3-30)

$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \tau_{23} \\ \tau_{31} \\ \tau_{12} \end{Bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & 0 & 0 & 0 \\ C_{12} & C_{22} & C_{23} & 0 & 0 & 0 \\ C_{13} & C_{23} & C_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \gamma_{23} \\ \gamma_{31} \\ \gamma_{12} \end{Bmatrix}$$

Expressions for the C_{ij} 's in terms of the orthotropic engineering material constants, $E_1, E_2, E_3, G_{12}, G_{23}, G_{31}$, and ν_{ij} 's are:

(Eq. 3-31)

$$\begin{aligned} C_{11} &= \frac{E_1(1 - \nu_{23}\nu_{32})}{\Delta}, & C_{12} &= \frac{E_1(\nu_{21} + \nu_{23}\nu_{31})}{\Delta}, & C_{13} &= \frac{E_1(\nu_{31} + \nu_{21}\nu_{32})}{\Delta}, \\ C_{21} &= \frac{E_2(\nu_{12} + \nu_{13}\nu_{32})}{\Delta}, & C_{22} &= \frac{E_2(1 - \nu_{13}\nu_{31})}{\Delta}, & C_{23} &= \frac{E_2(\nu_{32} + \nu_{12}\nu_{31})}{\Delta}, \\ C_{31} &= \frac{E_3(\nu_{13} + \nu_{12}\nu_{23})}{\Delta}, & C_{32} &= \frac{E_3(\nu_{23} + \nu_{13}\nu_{21})}{\Delta}, & C_{33} &= \frac{E_3(1 - \nu_{12}\nu_{21})}{\Delta}, \end{aligned}$$

Where

(Eq. 3-32)

$$\Delta = 1 - \nu_{12}\nu_{21} - \nu_{13}\nu_{31} - \nu_{12}\nu_{23}\nu_{31} - \nu_{13}\nu_{21}\nu_{32} - \nu_{23}\nu_{32}$$

And

(Eq. 3-33)

$$C_{44} = G_{23}, \quad C_{55} = G_{13}, \quad C_{66} = G_{12}$$

In addition to the three elastic moduli and three shear moduli, there are six Poisson's ratios in the proceeding equations giving a total of twelve engineering constants. However, only nine of the constants are independent. Three reciprocity relations [1] provided below without proof may be used to express the C_{ij} 's in terms of nine independent material constants:

(Eq. 3-34)

$$\frac{\nu_{21}}{E_2} = \frac{\nu_{12}}{E_1}, \quad \frac{\nu_{31}}{E_3} = \frac{\nu_{13}}{E_1}, \quad \frac{\nu_{23}}{E_2} = \frac{\nu_{32}}{E_3}$$

For thin layers with no applied forces acting in the out-of-plane direction (main condition respected over the development of the Basic and Intermediate examples, see next sections), it is possible to invoke the plane stress assumption, already introduced in 3.6.1.2, in the 1-2 principal material plane:

(Eq. 3-35)

$$\sigma_3 = 0, \quad \tau_{31} = \tau_{23} = 0$$

This reduces the stress-strain relations to

(Eq. 3-36)

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

Where the Q_{ij} 's are reduced stiffnesses and are given in terms of four independent engineering material constants in principal material directions as:

$$\begin{aligned} Q_{11} &= \frac{E_1}{1 - \nu_{12}\nu_{21}}, & Q_{22} &= \frac{E_2}{1 - \nu_{12}\nu_{21}} \\ Q_{12} &= \frac{\nu_{12}E_2}{1 - \nu_{12}\nu_{21}} = \frac{\nu_{21}E_1}{1 - \nu_{12}\nu_{21}} \end{aligned} \quad (\text{Eq. 3-37})$$

$$Q_{66} = G_{12}$$

Two of the four independent material constants are the elastic moduli in the 1 and 2 directions, E_1 and E_2 , respectively. The other two material constants are the shear modulus in the 1-2 plane G_{12} and the major Poisson's ratio ν_{12} (minor Poisson's ratio is ν_{21}). These two Poisson's ratio are related to each other by reciprocity relations of (Eq. 3-34)

$$\nu_{21} = \frac{E_2}{E_1} \nu_{12} \quad (\text{Eq. 3-38})$$

Since orthotropic layers are generally rotated with respect to a reference coordinate system, x-y, the stress-strain relations given in the principal directions of material orthotropy, (Eq. 3-36), must be transformed to the reference axes.

The transformation of stresses and strains can be accomplished by the following equations:

$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{Bmatrix} = T \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix}, \quad \begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_{12} \end{Bmatrix} = T \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_{xy} \end{Bmatrix} \quad (\text{Eq. 3-39})$$

Where ε_{xy} and ε_{12} are the tensor shear strains which correspond to one-half of the engineering shear strains

(Eq. 3-40)

$$\varepsilon_{xy} = \frac{\gamma_{xy}}{2}, \quad \varepsilon_{12} = \frac{\gamma_{12}}{2}$$

And T is the transformation matrix given by

(Eq. 3-41)

$$T = \begin{bmatrix} m^2 & n^2 & 2mn \\ n^2 & m^2 & -2mn \\ -mn & mn & m^2 - n^2 \end{bmatrix}, \quad m = \cos \theta, \quad n = \sin \theta$$

In order to convert the strain transformation relation from tensor strain notation to engineering strain notation, the transformation matrix is pre and post-multiplied by a matrix R and the inverse of R, respectively, where

(Eq. 3-42)

$$\begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix} \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_{xy} \end{Bmatrix}, \quad R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

Replacing the transformation relations for the stresses and engineering strains into (Eq. 3-36) the resulting equation is

(Eq. 3-43)

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = T^{-1} Q R T R^{-1} \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix}$$

This transformation produces

(Eq. 3-44)

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

Where the transformed reduced stiffnesses $\bar{Q}_{ij}$ are related to the Q_{ij} by

(Eq. 3-45)

$$\begin{aligned} \bar{Q}_{11} &= Q_{11} \cos^4 \theta + 2(Q_{12} + 2Q_{66}) \sin^2 \theta \cos^2 \theta + Q_{22} \sin^4 \theta \\ \bar{Q}_{12} &= (Q_{11} + Q_{22} - 4Q_{66}) \sin^2 \theta \cos^2 \theta + Q_{12} (\sin^4 \theta + \cos^4 \theta) \\ \bar{Q}_{22} &= Q_{11} \sin^4 \theta + 2(Q_{12} + 2Q_{66}) \sin^2 \theta \cos^2 \theta + Q_{22} \cos^4 \theta \\ \bar{Q}_{16} &= (Q_{11} - Q_{12} - 2Q_{66}) \sin \theta \cos^3 \theta + (Q_{12} - Q_{22} + 2Q_{66}) \sin^3 \theta \cos \theta \\ \bar{Q}_{26} &= (Q_{11} - Q_{12} - 2Q_{66}) \sin^3 \theta \cos \theta + (Q_{12} - Q_{22} + 2Q_{66}) \sin \theta \cos^3 \theta \\ \bar{Q}_{66} &= (Q_{11} + Q_{22} - 2Q_{12} - 2Q_{66}) \sin^2 \theta \cos^2 \theta + Q_{66} (\cos^4 \theta + \sin^4 \theta) \end{aligned}$$

These equations have been put in a simpler form, through the use of trigonometric identities of the fibre orientation angle θ , by Tsai and Pagano, [70], with the main intention to express the $\bar{Q}_{ij}$ in function of terms independent of the ply orientation angles. This has been possible by introducing the following material properties:

(Eq. 3-46)

$$\begin{aligned} U_1 &= \frac{1}{8} (3Q_{11} + 3Q_{22} + 2Q_{12} + 4Q_{66}) \\ U_2 &= \frac{1}{2} (Q_{11} - Q_{22}) \\ U_3 &= \frac{1}{8} (Q_{11} + Q_{22} - 2Q_{12} - 4Q_{66}) \\ U_4 &= \frac{1}{8} (Q_{11} + Q_{22} + 6Q_{12} - 4Q_{66}) \\ U_5 &= \frac{1}{8} (Q_{11} + Q_{22} - 2Q_{12} + 4Q_{66}) \end{aligned}$$

Which yield a simpler form of the transformed reduced stiffnesses of (Eq. 3-44) as:

(Eq. 3-47)

$$\bar{Q}_{11} = U_1 + U_2 \cos 2\theta + U_3 \cos 4\theta$$

$$\bar{Q}_{12} = U_4 - U_3 \cos 4\theta$$

$$\bar{Q}_{22} = U_1 - U_2 \cos 2\theta + U_3 \cos 4\theta$$

$$\bar{Q}_{16} = \frac{1}{2} U_2 \sin 2\theta + U_3 \sin 4\theta$$

$$\bar{Q}_{26} = \frac{1}{2} U_2 \sin 2\theta - U_3 \sin 4\theta$$

$$\bar{Q}_{66} = U_5 - U_3 \cos 4\theta$$

Both the U 's and the Q 's are independent of the ply orientation, but the $\bar{Q}$'s are dependent on ply orientation for orthotropic materials. An advantage of the form of the reduced stiffnesses in (Eq. 3-47) is the constant parts of $\bar{Q}_{11}$, $\bar{Q}_{12}$, $\bar{Q}_{22}$, and $\bar{Q}_{66}$ given in terms of the U_1 , U_4 , and U_5 which do not depend on the ply orientation. Based on this feature, the U 's are sometimes called the material invariants [1], [70] and as it will be showed later, a pivotal part of the Intermediate Example.

3.6.1.5 Laminated Orthotropic Layers

The same assumptions used in deriving (Eq. 3-25) in subparagraph 3.6.1.3 are applied when plies with unidirectional properties are bonded together. That is, Classical Lamination Theory (CLT) [1] assumes:

- The structure is restricted to be a thin plate or shell consisting of an arbitrary combination of laminae cured or consolidated into a single laminated plate
- The assumptions of plate and shell theory hold; i.e. the through plate thickness direct stress, σ_z , is zero and only plane stress, σ_{xy} , exists
- The two transverse shear stresses, τ_{xz} , τ_{yz} , are neglected to meet the classic thin plate and shell theory

- The theory is a “point” analysis in an effectively infinitely large plate and shell, completely ignoring the effects of neighbouring edges, stiffeners, holes, cutouts or any other discontinuities
- The loading is assumed to be in-plane memberane stress and moments [67]

So (Eq. 3-25) will be defined by a different in-plane stiffness matrix, A , in fact substituting the stress-strain relations (Eq. 3-44) it is possible to define the constitutive relations for a laminate of orthotropic layers:

$$\begin{Bmatrix} N_x \\ N_y \\ N_{xy} \end{Bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{16} \\ A_{12} & A_{22} & A_{26} \\ A_{16} & A_{26} & A_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_x^\circ \\ \varepsilon_y^\circ \\ \gamma_{xy}^\circ \end{Bmatrix} \quad (\text{Eq. 3-48})$$

Where:

$$A_{ij} = \sum_{k=1}^N \bar{Q}_{ij(k)} (z_k - z_{k-1}) \quad (\text{Eq. 3-49})$$

The in-plane stiffness matrix elements are defined according (Eq. 3-45) and (Eq. 3-47) through the use of the material invariants (Eq. 3-46).

From the above equations is straightforward to say that the design of symmetric and balanced laminates (that is, laminates which have a -9 layer for each $+9$ layer), core of the applications developed in this thesis, is characterised by the following condition further condition to take into account:

$$A_{16} = A_{26} = 0 \quad (\text{Eq. 3-50})$$

Now it is possible to extend the effective engineering constants evaluated for an isotropic laminate (Eq. 3-28), to an orthotropic composite laminate with total thickness h :

(Eq. 3-51)

$$E_x = \frac{1}{h} \left(\frac{A_{11}A_{22} - A_{12}^2}{A_{22}} \right), \quad E_y = \frac{1}{h} \left(\frac{A_{11}A_{22} - A_{12}^2}{A_{11}} \right)$$

$$G_{xy} = \frac{1}{h} A_{66}, \quad \nu_{xy} = \frac{A_{12}}{A_{22}}$$

The bending (D) and coupling (B) matrices are purposely neglected in this description since not part of Basic and Intermediate Example numerical applications.

3.6.2 Limitations of Classical Laminate Theory

Classical Laminate Theory (CLT) applies to an element of plate over which forces and moments are assumed constant, and in which through-thickness shear strains are ignored 3.6.1.5. Thus, only in-plane direct and shear stresses are considered. CLT is equivalent to simple beam theory which only considers a pure moment loading (i.e. constant along the beam), which results in only a direct stress (acting parallel to the beam's longitudinal axis). Again shear strains (through the beam depth) are ignored [13].

For a laminated construction such shears are referred to as inter-laminar shear stresses, and as already mentioned in the literature review section 2.8[46], they can give rise to inter-laminar failure, or de-lamination. A three-point flexure test on a short beam (span-to-depth equal to 5) is a common way of determining the inter-laminar shear strength of unidirectional composites [13].

Plate equations can be altered to include varying moments and through thickness shear strains [48][49][54][55]. The plate constitutive equations would be modified to account for the through-thickness shear resultants, by including two additional terms in the “A” matrix, (in-plane stiffness matrix), as Vinson describes in [71].

The inclusion of shear strains would be appropriate for thick isotropic plates (equivalent to short beams), and also for thick composite plates. The latter, of course, will generally have a low through-thickness shear modulus (compared with in-plane extensional

moduli), thus making the inclusion of such shear strains is generally advisable for typical composites, even if of moderate thickness [13].

The limitations of CLT mentioned above (constant in-plane stresses) imply that the plate is infinitely long and wide. In other words the theory ignores edges. In many real situations, of course, laminates will have edges; for example a plate of finite width (encountered in mechanical testing), a plate containing a hole (bolts, rivets) or opening (say for access), a skin/stringer joint, a skin/spar junction, and so forth [13].

A problem in analysing these situations is that the high stresses occur within one or two plate thicknesses from the edge. Also, solutions can indicate singularities. Finite element models of this situation are a robust alternative option to the Classical Lamination Theory as in part described in section 2.7 [4][38][13].

3.6.3 Finite Element Analysis

Finite element (FE) analysis is merely an alternative approach to solving the governing equations of a structural problem. Hence, FE and classical methods will produce identical results for the same problem, provided the former method is correctly applied and the applicable conditions are correct [13].

The FE method was initially developed for isotropic materials and the majority of the available commercial software, Nastran for instance, adopted constitutive formulation of these materials where the Hooke's Law, stress directly proportional to strain, is expressed as a combination of two of the following: elastic modulus, Poisson's ratio, and shear modulus [11]. To apply the technique to composites requires different element formulations that adequately represent their anisotropic or orthotropic, stiffness and strength, as well as the laminated form of construction often used [13].

One of the key factors is the low through-thickness stiffness and strength; the former results in the need to adapt elements used for plates and the latter results in the need to model failure criteria as de-lamination.

In fact laminated materials failure occurs mainly by de-lamination. The effect of de-lamination has to be modelled on some occasions and special techniques are required if bending elements are being used. One such method is to represent a de-lamination by

modelling two half-thicknesses of the plate in the area of the de-lamination and the full plate in non-delaminated areas. This concept, to break down a laminate by identifying sub-laminates, has already been introduced in section 2.8 and described by Mao et al. [63][64] as local/global finite element analysis. The crucial part of this approach, recommended by Fu and Ha in bioengineering applications too [23][24], and also employed in the Advanced Example later on, is to join the two parts of the model and this can be done with rigid links [13], labelled in FE field as multipoint constraint (MPC).

3.6.4 Properties of Laminates Made of Sub-Laminates

Since in this research the FE analysis will be a tool to be employed alongside the CLT, the concept of global/local FE approach by introducing sub-laminates blocks is well suited in the analytical technique side too. In fact, the stiffness matrices of a laminate made of two or more sub-laminates may be derived by using the stiffness matrices of the individual sub-laminates directly, rather than using the $\bar{Q}$ matrices, (Eq. 3-44) and (Eq. 3-45), of the individual plies of the sub-laminates. Besides the convenience, the method presented by Haftka [1], provides understanding of laminate stiffness properties and can be used to construct laminates with certain desired properties based on the properties of selected sub-laminates.

Consider for example, a laminate made up of two sub-laminates denoted by M and N with n_M and n_N plies, respectively. It is possible to represent the stiffness matrices (A, B and D) of the two laminates using subscripts M and N and the combined laminate of $n_M + n_N = n_{MN}$ plies with subscript MN.

Using the definition of the in-plane stiffness matrix from (Eq. 3-45) but without going into derivation of the main steps to the final equation, for more details go to pp.83 of [1], the combined in-plane stiffness matrix of the two sub-laminates M and N is:

$$(A_{ij})_{MN} = (A_{ij})_M + (A_{ij})_N \quad (\text{Eq. 3-52})$$

That is, when two or more laminates are put together to make a new laminate, the in-plane stiffness matrix of the new laminate is simply a sum of the in-plane stiffness matrices of the sub-laminates. Note that this does not mean that the individual stiffness properties such E_x and ν_{xy} of the new laminate are the sum of the same properties of the sub-laminates [1].

3.6.5 Failure Criteria for Laminated Composites

In section 3.6.1 it has been established the stress-strain relationships for an individual lamina, or ply, and for a laminate. So it is possible to use the constitutive equations to calculate the stresses in each ply when the value of loads acting on the laminate is known. By comparing these stresses with a corresponding limiting value it is then decidable whether or not the laminate will fail when subjected to the service loads [13]. Determining whether a design adequately meets design criteria is more difficult with composites than with conventional metal materials because failure criteria are more complex in composites, which can be delamination or have either matrix or fibre failure. Therefore, most engineers use conservative approaches which needlessly limit where composites can be applied [67].

Since composite failure modes often operate concurrently, interactively, sequentially with the different modes of failure of a unidirectional fibre composite laminate. The most in the aerospace field try to:

- Establish low design strain level from testing
- No matrix failure at limit load
- No fibre failure at ultimate load [67]

The most commonly used are the maximum strain, the maximum stress, and the quadratic criterion which represent an empirical description of the failure of a composite component subjected to complex states of stress or strains [67].

Basic and Intermediate examples will employ a very basic failure criteria, maximum strain, where the non interaction between strains allows to focus more on developing the

new methodology than understanding higher level of laminate failure criteria. This is possible also by the very nature of the in-plane loading conditions adopted.

On the other hand, the Advance example will consider a much more complicate scenario since in-plane and out-of-plane loading will be introduced and then stress/strain criteria, involving differrent quantities such as laminate and sub-laminate stiffness coefficients, will be required.

Finally, it is worth highlighting that many other engineering issues typical of composites associated to the model complexity (as stiffened plate structures or plate with cutouts) and to the analysis multiplicity (not just linear and static but normal modes, buckling and post-buckling analysis,) have not been considered since the main aim of the research is to understand the multi level decomposition and all its coupling mechanisms. In fact, adding extra real life engineering complexities would have generated other areas needed to be explored leaving less time to the core of the thesis.

For instance, those further difficulties mentioned above, may produce constitutive equations that most of the time are not linear and are governed by several variables as such to complicate the use of the graphical optimisation approach, employed tool of Basic and Intermediate examples.

3.7 Summary

In this chapter, it has been investigated and described the nature of composite materials in all their different configurations and main potentials compared with the metallic counterparts, focusing specifically on the continuous fibre reinforced unidirectional composite laminates. Then the conventional stress-strain equations in three dimensions to the two-dimensions case have been explained creating enough background to enable the researcher, to develop the classical lamination theory through Matlab programming to be used in the next coming numerical applications.

The main attempt has been to highlight that the elastic properties that form the relation between the stresses and strains, the coefficients of the A matrix (Eq. 3-49), are the fundamental quantities that govern the response of the material and, therefore, the structure to be designed. Response quantities such as deformations, stresses, buckling

loads, and natural frequencies, all depend on elastic properties. In designing structures made up of monolithic materials, these properties are normally given and are not included among the set of parameters that the designer varies in order to improve the performance of the structure. In contrast, for composite materials part of the effort is to design elastic properties of the composite medium [1].

So discussing how elastic properties are calculated for laminated composite materials, identifying the variables that can be changed during the design process, it allows understanding the best optimisation techniques to employ.

In fact the material invariants, 3.6.1.4, are the starting point to deal with more efficient graphical optimisation procedure for the design of laminates with prescribed stiffness properties and mainly to overcome the limitations of using integer numbers as design variables.

Moreover, the sub-laminates concepts, 3.6.4., will help to create a suitable framework for introducing the multi-level decomposition optimisation to laminate structures as demonstrated in the bioengineering techniques on optimisation of thick composite materials[23][24].

4 Optimisation

4.1 General

Optimisation has become a necessary part of design activity in all major disciplines. The motivation to produce economically viable products or services with embedded quality is the principal reason for this inclusion. Improved production and design tools, with a synergistic thrust through inexpensive computational means, have aided the consideration of optimisation methods in new developments, particularly engineering products [12].

Structural designers seek the best possible design, be it a vehicle structure, a civil engineering structure, or a space structure, while using the least amount of resources. The measure of quality of a design depends on the application, typically related to strength or stiffness, while resources are measured in terms of weight or cost. Therefore, the best design often means either the lowest weight (or cost) with limitations on the stiffness (or strength) properties, or the maximum stiffness design with prescribed resources and strength limits [1].

Traditionally, engineers have relied on experience to achieve such designs. For a given application, first a set of essential requirements are identified and designs that satisfy these requirements are obtained. Next, structural modifications that are likely to improve the performance or reduce the weight or the cost are implemented. The tasks, stated so simply in two sentences, are often extremely tedious and require a large number of iterations by the designer [1].

Over the past two decades, mathematical optimisation deals with either maximisation or minimisation of an objective function subject to constraint functions, has emerged as a powerful tool for structural design as already presented in the section 2.2.

The use of composite materials in structural design has gained popularity over the past three decades because of several advantages, see 3.4. One of the primary reasons for their popularity is their weight advantage per unit volume; in addition some composites provide better stiffness and strength properties compared to metal as well. That is, structural members made out of composite materials may undergo smaller deformations, and carry larger static loads than their metallic counterparts. Composite materials are also known to perform better under cyclic loads than metallic materials because of their fatigue resistance.

It was these advantages that initially stimulated widespread use of composites despite their structural design have significant implications in terms of optimal design. In fact the flexibility through the use of variables that directly change the properties of the material (lamina thickness, orientation angle, stacking sequence, ply materials) in order to improve structural performance and meet the requirements of a specific design situation, increases the number of variables participating into the final design.

This increased number of design variables is both a blessing and curse for the designer. There are more controls to fine-tune the structure to meet design requirements, but the increased number of variables brings the additional burden of selecting those design variables that are important and identifying their values for the best solution of the design problem. The possibility of achieving an efficient design that is safe against multiple failure mechanisms, coupled with the difficulty in selecting the values from a large set of design variables makes mathematical optimisation a natural tool for the design of laminated composite structures [1].

This chapter attempts to address the design features associated with laminated composite materials already presented in 3.6, via the application of the most specific methods of mathematical optimisation to be employed in numerical applications later on.

4.2 Optimisation Fundamentals

It should be understood that optimisation presupposes the knowledge of the design rules for the specific problem, primarily the ability to describe the design in mathematical terms. These terms include design variables, design parameters, and design functions.

Traditional design practice, that is, design without regard to optimisation, includes all of these elements although they were not formally recognised as such. In general optimisation is a procedure for searching the best design among candidates, each of which can produce an acceptable product [12].

For making the right design choice is necessary to understand that an optimisation problem has an objective function which measures the goodness or efficiency of the design. The maximisation of the quality is generally performed within some limits that constrain the choice of the design. Such limits are called constraints. Finally, an optimisation problem has design variables, which are the parameters that are changed during the design process. Design variables can be continuous or discrete depending on whether they can take values from a continuum, or are limited to a number of discrete values. A special case of discrete variables are integer variables [1] though those will not be included in any numerical application of this thesis, but mentioned as pivotal part in the development of the advanced example.

The commonly used notation for design variable, objective function, and constraints is as follows. The vector x with n components describe the design variables to be considered. To allow for the case that some of the design variables may be discrete or limited otherwise, it is introduced X to denote the domain of these design variables. For the objective function, notation $f(x)$ is used and for constraints, the notation $g(x)$ for inequality constraints and $h(x)$ for equality constraints is also adopted [1].

The standard form of the optimisation problem is written as:

$$\text{Minimise } f(x) \quad x_n \in X$$

Subject to

$$h_i(x) = 0 \quad i = 1, \dots, n_e \quad (\text{Eq. 4-1})$$

$$g_j(x) \leq 0 \quad j = 1, \dots, n_g$$

$$x^L \leq x \leq x^U \quad (\text{Eq. 4-2})$$

Where the elements of the vectors x^L and x^U , depicted in (Eq. 4-2), are the lower and upper bounds on the values of the design variables, since they express the range for the design variables and are known as side constraints. In this standard form, it is chosen to minimise rather than maximise the objective function. To convert a maximisation problem into a minimisation problem, it is just needed to change the sign of the objective function. That is, instead of maximising $f(x)$ it is possible to minimise $-f(x)$.

Similarly it is worth mentioning that the constraints are written with a zero on the right-hand side and for a particular choice of the sense of the inequality. Obviously, it is easy to arrange for a zero on the right-hand side of the equalities or inequalities, and the sense can be changed by multiplying by -1. This particular formulation will be particularly useful in applying the optimisation standard format by Matlab.

Note that there are fundamental requirements, hidden inside the above formulation (Eq. 4-1) and (Eq. 4-2), to be met. First, the set of design variables must be linearly independent. This means that it is not possible to establish the value of one of the design variables from the values of the remaining variables through basic arithmetic (scaling or addition) operations. Second, the number n of the design variables in the problem must be greater than the number of equality constraints n_e for optimisation to take place. If n is equal to n_e , then the problem will be solved without reference to the objective. In mathematical terms, the number of equations matches the number of unknowns. If n is less than n_e , then it is defined as an over-determined set of relations, which could result in an inconsistent problem definition. Then, the set of equality constraints must be linearly independent. These recommendations serve to ensure that the mathematical search for solution will not fail [12].

The set of design points in X that satisfy all the constraints is called the feasible domain. At a given design point x , a constraint may be satisfied or violated. An inequality constraint for which the equality condition holds at a particular design point x is called an active constraint. All satisfied equality constraints are active, while inequality constraints can be active or may be satisfied with a margin, in which case they are called inactive. Formulation and graphical representation of a design problem will make these concepts more understandable, as it will be shown later in the Basic and Intermediate Examples.

4.3 Optimisation Classification

Having outlined the main design processes of laminated composite materials, chapter 3, and the fundamentals of optimisation, attention is next focused trying to bring together multiple disciplines in to a single computational environment, including the subject of design search and optimisation.

These increasingly lie at heart of much computationally based design, not because designers necessarily believe that such methods can yield truly optimal designs, or even very good designs, but rather because they provide a systematic framework for considering some of the many decision designers are charged with taking. Nonetheless, it is helpful when looking at computational approaches to design to have a thorough understanding of the various classes of optimisation methods, where and how they work and, more importantly whether or not are likely to fail [7].

As already been noted in the previous section, optimisation methods may be classified according to the types of problems they are designed to deal with. Problems can be classified according to the nature of the variables the designer is able to control (nonnumeric, discrete, real valued), the number and type of objectives (one, many, deterministic, probabilistic, static, dynamic), the presence of constraints (none, variable bounds, inequalities, equalities) and the types of functional relationships involved (linear, nonlinear, discontinuous). Moreover, optimisers can be sub-classified according to their general form of approach. They can be gradient based, rule based (heuristic), or stochastic; often they will be a subtle blend of all three. They can work with a single design point at a time, or try to advance a group or “population” of designs. They can be aiming to meet a single objective or to find a set of designs that together satisfy multiple objectives [7].

As already analysed earlier in section 2.5, composite laminated materials require to be tackled by optimisation of coupled systems, that is, problems where the objective and constraint function evaluations require iterations between two or more disciplinary analysis codes. Such optimisation techniques are part of the Multidisciplinary Design Optimisation (MDO) as traditional practice in aerospace design, to employ a sequential staggered approach means to break down large corporations along disciplinary lines into different proper divisions.

4.4 System Decomposition and Optimisation

System decomposition is essentially concerned with dividing the required analyses into subsystems with minimal coupling, so as to enable a degree of concurrency in the analysis. In many multidisciplinary design problems, it may be preferable to decompose the system under consideration along the lines the disciplines involved. However, the idea of system decomposition is very general and can be applied to any problem where it is necessary to run (or iterate between) a sequence of modules to calculate objective and constraint functions [7].

It is needful to consider how tightly coupled is each module. For instance, it is very clear that when studying the aeroelasticity of flexible wing structures the interaction between structural and aerodynamic codes is very significant. On the other hand, as mentioned regarding the specific way to employ this decomposition techniques within the task of this thesis, see chapter 2.5, the crucial part is related to the ability to idealise, structurally speaking, properly the component to analyse. In fact, using the Sobieski terminology, [8],[23], see 2.5, the main structure, as an analogy of a multidisciplinary system component (e.g. the full thick laminated composite, see Advanced Example) may be treated as a system composed of the following three disciplines:

- Discipline 1: Laminate
- Discipline 2: Sub-laminates
- Discipline 3: Plies

Regarding Basic and Intermediate Examples, the “discipline approach” is still employed but reduced to a two-level process. More details on each discipline are described by Table 1-1, Table 1-2, and Table 1-3.

Each discipline will need to be coupled in two separate directions. Horizontally to validate the outcome of the optimisation block by an external solver such as Nastran, and vertically to keep the data flow going to the next discipline, making it possible to take over the next iteration step, up to the final convergence.

In this scenario, the huge flexibility of this multi-level decomposition is to enable each discipline to use the best optimisation technique required by the specific type of objective, constraint functions and design variables.

The following subparagraphs will highlight the main optimisation techniques used for the Basic, Intermediate and Advanced examples described in this thesis.

4.4.1 Graphical Solution

The design space that will be searched for optimal design, is the Euclidean or Cartesian n -dimensional space generated by the n independent design variables X . This is a generalisation of the three-dimensional physical space with which we are familiar. For ten design variables, it is ten-dimensional space. This is not easy to imagine, nor it is easy to express this information through a figure or a graph because of the limitation of the three-dimensional world. The side constraints limit the search region, implying that only solutions that lie within a certain region will be acceptable. They define an n -dimensional rectangular region, hypercube, from which the feasible and optimal solutions must be chosen [12].

The simplest determination is through a graphical representation of the design functions involved in the problem. This can be done easily for one or two variables. This graphical representation through the main numerical applications of the thesis will be made possible by employing the 2-D Matlab graphic features and sometime to enhance the overall understanding of the feasible domain, see next chapter Basic Example, the three-dimensional representation will be used too. Normally, to distinguish objective functions from constraints different colour and thickness lines will be adopted. Moreover the scaled objective function is represented through several labelled contours. Each contour is associated with a fixed value of the objective function and these values will also be shown in the figure. Normally the range of the two axes establishes the side constraints.

Recognising the nature of solutions is important in developing a search procedure and in pointing to the correct feasible domain. For example in linear programming problems, a technique employed to search for the optimal design need only search for boundaries,

particularly the intersection of the boundaries, whereas in non linear problems, the search procedure cannot ignore interior values [12].

Normally the visual inspection is the way to point out, among the infinite feasible solutions, the unique optimal solution. That is a sort of search process that starts typically by trying to guess the initial design solution. The designer can base this selection on his experience. On several occasions, the success of optimisation may hinge on his ability to choose a good initial solution. The search method, even used consistently, responds differently to the particular problem that is being solved. The degree and the type of nonlinearity may frequently cause the method to fail, assuming there are no inconsistencies in problem formulation. Often, the numbers of design variables, the number of constraints, as well as a poor initial guess play a part in unexpected behaviour under standard implementation of the method. This has led to the creation of several different techniques for optimisation. Practical experience has also suggested that a class of problems respond well to certain algorithms [12].

4.4.2 Lamination Parameters

Lamination Parameters have to be considered as one of those graphical optimisation tools that perform specifically to overcome the graphical limitations arising in a design space of orientation angles as well pointed out through the next chapter, Basic Example. In chapter 4, mainly 3.6.1.4 and 3.6.1.5, it has been discussed the use of classical lamination theory (CLT) for calculating the stiffness properties of composite laminates. Now it will be used to obtain the expressions for designing laminates with desired in-plane stiffness properties.

Normally a designer is concerned with the strength as well as the stiffness of a laminate. Unlike stiffness, the strength of composite laminates is not easy to predict, 3.6.5, since it is a result of many complex modes of failure. Then there are many applications where the design of the laminate is primarily governed by in-plane stiffness requirements. Finally, the in-plane stiffness problems are easier to manage than other laminate topics, since they do not depend on the arrangement or stacking sequence of the layers with various orientations in the laminate [1]. This is because in-plane stiffness matrix A , (Eq.

3-49), simply sums the contributions of the in-plane stiffness properties of layers with various orientations, without regard to their z through thickness location.

In this context, Miki, [66] introduced a more general graphical procedure for the design of laminates with prescribed in-plane stiffness properties. The procedure is suitable for balanced angle-ply laminates made up of multiple stacks of layers with different orientation angles.

The stacking sequences of laminates amenable for treatment by Miki's procedure may be written as $[(\pm\theta_1)_{N_1}/(\pm\theta_{l-1})_{N_{l-1}}/.../(\pm\theta_1)_{N_1}]_S$ where the total number of layers in the laminate is $N = 4 \sum_{i=1}^I N_i$, and I is the number of different $\pm\theta$ groups. In addition to the balanced angle-ply sub-laminates, stacks of unidirectional layers with principal material axes aligned with the axes of the laminate (i.e. 0° or 90°) can be included in the stacking sequence [1].

The key to Miki's procedure is that for a balanced composite laminate, engineering constants of the laminate can be described in terms of two lamination parameters that capture all the relevant information associated with the stacking sequence. These two parameters are V_1^* and V_3^* , which are obtained by normalising the in-plane stiffness components of (Eq. 3-49), expressed in terms of (Eq. 3-47), by the total laminate thickness. In fact, the elements of the A matrix divided by the laminate thickness h, can be determined by the following equations:

$$\begin{Bmatrix} A_{11}^* \\ A_{22}^* \\ A_{12}^* \\ A_{66}^* \end{Bmatrix} = \begin{bmatrix} U_1 & V_1^* & V_3^* \\ U_1 & -V_1^* & V_3^* \\ U_4 & 0 & -V_3^* \\ U_5 & 0 & -V_3^* \end{bmatrix} \begin{Bmatrix} 1 \\ U_2 \\ U_3 \end{Bmatrix} \quad (\text{Eq. 4-3})$$

Where U_i are the material invariants according (Eq. 3-46) and

$$A_{ij}^* = \frac{A_{ij}}{h} \quad (\text{Eq. 4-4})$$

The effective engineering constants can then be obtained from A_{ij}^* 's using (Eq. 3-51).

For example,

$$E_x = \left(\frac{A_{11}^* A_{22}^* - (A_{12}^*)^2}{A_{22}^*} \right) \quad (\text{Eq. 4-5})$$

The other expressions in (Eq. 3-51) can be modified in a similar manner to obtain the other effective engineering constants.

The major effort of Miki's design procedure is the construction of a lamination parameter diagram which describes the region of allowable combinations of lamination parameters. So despite the plane design space, $V_1^* V_3^*$, it is possible to consider a final design laminated structure made by n -layers and θ_k orientation angles.

For a laminate of total thickness h , in which the volume fraction of the layers with $\pm\theta_i$ orientation angles is v_i , the lamination parameters are given as:

$$V_1^* = \frac{V_{1A}}{h} = \sum_{k=1}^I v_k \cos 2\theta_k, \quad V_3^* = \frac{V_{3A}}{h} = \sum_{k=1}^I v_k \cos 4\theta_k \quad (\text{Eq. 4-6})$$

Where I is the number of different $\pm\theta$ groups, and

$$\sum_{i=1}^I v_i = 1 \quad (\text{Eq. 4-7})$$

From (Eq. 4-6) and (Eq. 4-7), the values of the lamination parameters are always bounded, $-1 \leq (V_1^*, V_3^*) \leq 1$. For a laminate with only one fibre orientation angle, $[\pm\theta_n]_s$ ($I=1$), the lamination parameters are

$$V_1^* = \cos 2\theta, \quad V_3^* = \cos 4\theta \quad (\text{Eq. 4-8})$$

Using trigonometric identity, $\cos^2 2\theta = \frac{1}{2}(\cos 4\theta + 1)$, the two parameters are related as

$$V_3^* = 2V_1^{*2} - 1 \quad (\text{Eq. 4-9})$$

Values of all possible combinations of the lamination parameters are therefore located along the boundary curve of a parabola, in a Cartesian 2-dimensional plane, where $x = V_1^*$ and $y = V_3^*$ (see chapter 7 for pictures of the lamination diagram).

If the laminate consists of two or more fibre orientations, then it can be shown that (Eq. 4-8), becomes an inequality

$$V_3^* \geq 2V_1^{*2} - 1 \quad (\text{Eq. 4-10})$$

The allowable region of the lamination parameters is the internal area bounded by the parabola of (Eq. 4-9), irrespective of the number of different layer orientation angles. Each point of this region is called a lamination point and corresponds to one or more laminates with specific properties. Any point inside the lamination parameter diagram must correspond to laminates with two or more fibre orientation angles [1].

Now the pivotal part of the development of the intermediate example is passing through the evaluation of the lamination characteristic of the final laminate once the lamination point is known. This is possible since the lamination characteristic of the points inside the lamination diagram may be related to the characteristic of the points along the edges of the diagram. In fact, the points along the edges are fully known reversing the closed form equations (Eq. 4-8). Then from (Eq. 4-6) it can be noticed that the lamination parameters $V_1^* V_3^*$ are linear functions of the volume fractions, v_i . This means that the lamination parameters along a straight line connecting two points in the diagram represent laminates with volume fractions proportional to the distance from these points. That is, let points P_s and P_q , located at a distance ℓ of each other, be characterised by vectors of volume fractions v_s and v_q respectively. The dimensionality

of the volume fraction vectors is determined by the number of distinct fibre orientation angles, $\nu = (\nu_1, \nu_2, \dots, \nu_l)$. Then the point P lying on the line $P_s P_q$ at a distance $r\ell$ from P_s and $(1-r)\ell$ from P_q has a vector ν of volume fractions

(Eq. 4-11)

$$\nu = (1-r)\nu_s + r\nu_q$$

By applying the above equations, it is possible to understand that a point in the interior of the lamination diagram may represent a very large number of laminates with include many orientation angles. However, since a point is defined by two parameters, this means that only two orientation angles, θ_1 and θ_2 , are sufficient for designing laminates for given lamination parameters and hence for prescribed stiffness requirements. It is achievable to find combinations of two angles by passing a line through the point, finding its intersection with the diagram boundary on the two opposite sides of the point, and going through the procedure illustrated to obtain (Eq. 4-11) by using the interior point and the two boundaries points [1].

Then, solving the problem algebraically for a laminate with two orientation angles, (Eq. 4-8) becomes

(Eq. 4-12)

$$V_1^* = \nu_1 \cos 2\theta_1 + \nu_2 \cos 2\theta_2, \quad V_3^* = \nu_1 \cos 4\theta_1 + \nu_2 \cos 4\theta_2$$

Where

(Eq. 4-13)

$$\nu_2 = 1 - \nu_1.$$

It is important to note that there are three parameters, ν_1 , θ_1 , and θ_2 for a particular lamination point. So, for a given volume fraction, the other is then evaluated by (Eq. 4-13). Using an appropriate range, the two angles can be obtained by solving (Eq. 4-12) as

(Eq. 4-14)

$$\vartheta_1 = 0.5 \cos^{-1} T_1, \quad \vartheta_2 = 0.5 \cos^{-1} T_2$$

Where

(Eq. 4-15)

$$T_1 = 2\nu_1 V_1^* \pm \frac{\sqrt{2\nu_1 \nu_2 (1 + V_3^* - 2V_1^{*2})}}{2\nu_1}, \quad T_2 = \frac{V_1^* - \nu_1 T_1}{\nu_2}$$

The above (Eq. 4-15) is determined by combining the following equations:

- (Eq. 4-12)
- (Eq. 4-13)
- Trigonometric identity: $\cos^2 2\vartheta = \frac{1}{2}(\cos 4\vartheta + 1)$

For balanced angle-ply laminates with more than two orientations, many combinations of orientation angles and volume fractions can produce the same lamination parameters and, therefore, the same stiffness properties (for more details see Haftka, Gurdal and Hajela [1]).

A very useful aid in the graphical solution of the laminate design problem is to draw contours of constant effective engineering stiffnesses, (Eq. 3-51), but in terms of the lamination parameters $V_1^* V_3^*$. The equations needed for such contours, inside the lamination diagram, are

(Eq. 4-16)

$$E_x \text{ Contours: } V_3^* = \frac{U_2^2 V_1^{*2} - U_2 E_y V_1^* + E_x U_1 - U_1^2 + U_4^2}{U_3 (2U_1 + 2U_4 - E_x)}$$

(Eq. 4-17)

$$E_y \text{ Contours: } V_3^* = \frac{U_2^2 V_1^{*2} + U_2 E_y V_1^* + E_y U_1 - U_1^2 + U_4^2}{U_3 (2U_1 + 2U_4 - E_y)}$$

(Eq. 4-18)

$$v_{xy} \text{ Contours: } V_3^* = \frac{v_{xy}U_2V_1^* - v_{xy}U_1 + U_4}{(1+v_{xy})U_3}$$

(Eq. 4-19)

$$G_{xy} \text{ Contours: } V_3^* = \frac{U_5 - G_{xy}}{U_3}$$

The above equations; (Eq. 4-16), (Eq. 4-17), (Eq. 4-18), and (Eq. 4-19) are the final results of the combination of the following equations:

- (Eq. 3-46)
- (Eq. 3-51)
- (Eq. 4-3)

Having the effective engineering stiffnesses expressed in terms of the lamination parameters, it enables to plot in the lamination diagram stress and strain expressed in (Eq. 3-44) as:

(Eq. 4-20)

$$\varepsilon_x = \frac{1}{E_x}(\bar{\sigma}_x - v_{xy}\bar{\sigma}_y) = \frac{1}{hE_x}(N_x - v_{xy}N_y)$$

(Eq. 4-21)

$$\varepsilon_y = \frac{1}{E_y}(\bar{\sigma}_y - v_{yx}\bar{\sigma}_x) = \frac{1}{hE_y}(N_y - v_{yx}N_x)$$

(Eq. 4-22)

$$\gamma_{xy} = \frac{\bar{\sigma}_{xy}}{G_{xy}} = \frac{N_{xy}}{hG_{xy}}$$

Where the $\bar{\sigma}_{xy}$, $\bar{\sigma}_y$ and $\bar{\sigma}_x$ are the average stresses obtained dividing the load N_{xy} , N_y and N_x by the laminate thickness h .

4.4.3 Numerical Techniques

When the graphical optimisation techniques described in 4.4.1 and 4.4.2 are not anymore applicable due to either the high number of design variables, more than two, or to the evidence that curvature and gradient of the functions involved have a significant influence on the solution, then it is worth to understand the optimality conditions which the numerical techniques are based on.

An analytical foundation is essential to understand and establish the conditions that optimal solution will have to satisfy. This is not for the sake of mathematical curiosity, but is an essential component of the numerical technique [12].

The necessary mathematical definitions will be mentioned in this section for a full detailed description is recommended to refer to for example Keane and Prasanth [7].

The optimisation problem is treated in his most general configuration as a mathematical model, see 4.2, characterised by nonlinearity and unconstrained first followed by constrained problems. In nonlinear problems it is not essential that all the functions involved be nonlinear. It is sufficient if just one of them is nonlinear. In fact there are many examples, of laminated composites, where the objective function, say the weight, is non linear (when expressed in terms of the ply thicknesses) is linear while the constraints, say the strains, are non linear (when defined by orientation angles or lamination parameters). The next Basic and Intermediate applications will show in depth these concepts.

So once identified the mathematical model of the specific problem considered the remaining task is to establish the solution. In mathematics the solution is obtained by satisfying the necessary and sufficient conditions related to the class of the problems.

The necessary conditions basically say that moving to any points in the plane from the optimum will not change the value of the function. Moving away from optimum implies that dx_1 and dx_2 are not zero. This assumption mathematically speaking is summarised by this equation

$$df = 0 \quad \text{(Eq. 4-23)}$$

Where from mathematical calculus [72] the differential change in $f(x_1, x_2)(df)$ due to the differential change in the variables $x_1(dx_1)$ and $x_2(dx_2)$ is expressed as

$$df = \frac{\partial f}{\partial x_1} dx_1 + \frac{\partial f}{\partial x_2} dx_2 \quad (\text{Eq. 4-24})$$

For convenience and simplicity, the subscript representing the point where the expression is evaluated is not indicated.

Rewriting (Eq. 4-23) in terms of x_1 and x_2

$$df = \frac{\partial f}{\partial x_1} dx_1 + \frac{\partial f}{\partial x_2} dx_2 = 0 \quad (\text{Eq. 4-25})$$

$$df = \begin{bmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} \end{bmatrix} \begin{bmatrix} dx_1 \\ dx_2 \end{bmatrix} = 0 \quad (\text{Eq. 4-26})$$

Since this should hold for all points in the plane, $dx_1 \neq 0$ and $dx_2 \neq 0$. Therefore,

$$\frac{\partial f}{\partial x_1} = 0, \quad \frac{\partial f}{\partial x_2} = 0 \quad (\text{Eq. 4-27})$$

Or the gradient of the f at the optimum must be zero.

The Gradient, according definition in [72], of a function of a single variable is associated with the slope, in two or more variables the slope is equivalent to a vector, and at any point represents the direction in which the function will increase most rapidly. The gradient is defined as

(Eq. 4-28)

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} \end{bmatrix}^T$$

Using this definition (Eq. 4-27) will be

(Eq. 4-29)

$$\nabla f(x_1^*, x_2^*) = 0, \quad \nabla f(X^*) = 0$$

Equation (Eq. 4-29) expresses the necessary condition, or first-order conditions (FOC), for unconstrained optimisation for 2 or n variables. It is well clear that FOC applies to the minimum and maximum search also.

While the necessary condition involves the first derivative the sufficient conditions employs the second derivatives of the function. In fact is often obtained through the Taylor expansion of the function to second order.

The Taylor series is a useful mechanism to approximate the value of the function $f(X)$ at a point $X_p + \Delta X$ if the function is completely known at point X_p . The Taylor expansion, stopped at the second order, in the most general configuration looks like

(Eq. 4-30)

$$f(X_p + \Delta X) = f(X_p) + \nabla f(X_p)^T \Delta X + \frac{1}{2} \Delta X^T H(X_p) \Delta X$$

Where H is the Hessian matrix, [72], known as the matrix of second derivatives of a function of several variables. For $f(X)$, where $[X] = [x_1, x_2, \dots, x_n]^T$ the Hessian is:

(Eq. 4-31)

$$[H] = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \dots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \dots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}$$

Now it is all set to describe thoroughly the sufficient conditions. If X^* is the solution, and ΔX represents the change of the variables from the optimal value which will yield a change Δf , then

(Eq. 4-32)

$$\Delta f = f(X^* + \Delta X) - f(X^*) = \nabla f(X^*)^T \Delta X + \frac{1}{2} \Delta X^T H(X^*) \Delta X$$

This is similar to (Eq. 4-30) except the expansion is about the solution. Since it has been chosen the minimum search Δf must be greater than zero. Employing the necessary conditions (Eq. 4-29), the first term on the right-hand side of (Eq. 4-32) is zero. This leaves the following inequality known as second-order conditions (SOC)

(Eq. 4-33)

$$\Delta f = \frac{1}{2} \Delta X^T H(X^*) \Delta X > 0$$

Where $H(X^*)$ is the Hessian matrix (the matrix of second derivatives of the function f at the possible optimum value X^*). For the relations in (Eq. 4-33) to hold, the matrix $H(X^*)$ must be positive definite [72]. There are three ways to establish the H is positive definite

- i. For all possible ΔX , $\Delta X^T H(X^*) \Delta X > 0$
- ii. The eigenvalues of $H(X^*)$ are all positive
- iii. The determinants of all lower orders (sub-matrices) of $H(X^*)$ that include the main diagonal are all positive

Of the three, only ii and iii can be practically applied, which is well illustrated in [7],[72], and [12]. But mainly Matlab enables to apply FOC and SOC by employing powerful features of its Optimisation tools.

Equations (Eq. 4-29) and (Eq. 4-33) gather the necessary and sufficient conditions for unconstrained optimisation problems once the objective problem is subject to equality constraints it is needed to review the mathematical model. In fact the problem is transformed by introducing an augmented function, called the Lagrangian [7], which is defined as the sum of the original objective function and a linear combination of the constraints. The coefficients of this linear combination are known as the Lagrange multipliers.

To introduce the Lagrangian the optimisation standard form highlighted in (Eq. 4-1) and (Eq. 4-2) is reviewed and the augmented problem with the Lagrangian is as follows:

$$\begin{aligned} \text{Minimise} \quad F(X, \lambda) &= f(X) + \sum_{k=1}^I \lambda_k h_k(X) \\ &= f(X) + \lambda h \end{aligned} \tag{Eq. 4-34}$$

$$\text{Subject to} \quad [h(X)]_I = 0 \tag{Eq. 4-35}$$

$$X^{low} \leq X \leq X^{up} \tag{Eq. 4-36}$$

Where F is the Lagrangian function and λ 's are the Lagrange multipliers.

Applying, to the optimisation standard form described by the equations (Eq. 4-34), (Eq. 4-35), and (Eq. 4-36), the first order conditions are as follows:

$$\text{FOC} \quad \nabla F = \nabla f + [\nabla h_1 \quad \nabla h_2 \quad . \quad . \quad . \quad \nabla h_I] \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ . \\ \lambda_I \end{bmatrix} = 0$$

In the analytical derivation of the FOC for this problem the Lagrangian F was considered unconstrained. Borrowing from the unconstrained problem of (Eq. 4-32) and (Eq. 4-33), the Second-Order conditions can be expected to satisfy the following relations:

(Eq. 4-37)

$$\begin{aligned} \text{SOC} \quad \Delta F &= F(X^* + \Delta X) - F(X^*) = \nabla F(X^*)^T \Delta X + \frac{1}{2} \Delta X^T [\nabla^2 F(X^*)] \Delta X > 0 \\ \nabla h^T \Delta X &= 0 \end{aligned}$$

In the above the equation $[\nabla^2 F(X^*)]$ is the Hessian of the Lagrangian, with respect to the design variables only evaluated at the solution. Also, FOC require that $\nabla F(X^*) = 0$. Compared to the SOC for unconstrained minimisation problem, (Eq. 4-37) is not easy to apply, especially the $\nabla h^T \Delta X = 0$. From a practical perspective the SOC is not imposed for equality constrained problems. It is left to the designer to ensure by other means that the solution is a minimum solution [12].

Finally to tackle an inequality constrained optimisation problem it will be necessary to apply a two-step transformation, from inequality to equality and then to equality constrained to un-constrained (as just showed above by using the Lagrangian).

The standard first-step transformation, from inequality to equality constrained problem, requires the introduction of the slack variables, z_j , for each inequality constrained g_j . Therefore, the square of the new variable is added to the left-hand side of the corresponding constraint. In the eventuality of an inequality of the type of “ $\leq$ ” this adds a positive value to the left-hand side to bring the constraint up to zero. Of course a zero value will be added if the constraint is already zero.

The transformation of the inequality constrained problem into an equality constrained problem is shown below.

(Eq. 4-38)

$$\text{Minimise} \quad f(x_1, x_2)$$

(Eq. 4-39)

$$\text{Subject to} \quad g_1(x_1, x_2) + z_1^2$$

(Eq. 4-40)

$$g_2(x_1, x_2) + z_2^2$$

Side Constraints

(Eq. 4-41)

$$x_1^{low} \leq x_1 \leq x_1^{up}$$

$$x_2^{low} \leq x_2 \leq x_2^{up}$$

There are four variables (x_1, x_2, z_1, z_2) and two equality constraints. It is a valid equality constrained problem (constraints < variables as requested in 4.2). Now the Lagrange multiplier method can be applied to this transformation. To distinguish the multipliers associated with inequality constraints the symbol β is used.

The unconstrained optimisation problem is formulated as follows:

Minimise

(Eq. 4-42)

$$F(x_1, x_2, z_1, z_2, \beta_1, \beta_2) = f(x_1, x_2) + \beta_1 [g_1(x_1, x_2) + z_1^2] + \beta_2 [g_2(x_1, x_2) + z_2^2]$$

The FOC, necessary conditions, are

(Eq. 4-43)

$$\frac{\partial F}{\partial x_1} = \frac{\partial f}{\partial x_1} + \beta_1 \frac{\partial g_1}{\partial x_1} + \beta_2 \frac{\partial g_2}{\partial x_1} = 0$$

(Eq. 4-44)

$$\frac{\partial F}{\partial x_2} = \frac{\partial f}{\partial x_2} + \beta_1 \frac{\partial g_1}{\partial x_2} + \beta_2 \frac{\partial g_2}{\partial x_2} = 0$$

(Eq. 4-45)

$$\frac{\partial F}{\partial z_1} = 2\beta_1 z_1 = 0$$

(Eq. 4-46)

$$\frac{\partial F}{\partial z_2} = 2\beta_2 z_2 = 0$$

(Eq. 4-47)

$$\frac{\partial F}{\partial \beta_1} = g_1 + z_1^2 = 0$$

(Eq. 4-48)

$$\frac{\partial F}{\partial \beta_2} = g_2 + z_2^2 = 0$$

Now equations from (Eq. 4-43) to (Eq. 4-48) are equality constraints.

By simple recombination [12], the above equations can be collapsed in the following four relations:

(Eq. 4-49)

$$\frac{\partial F}{\partial x_1} = \frac{\partial f}{\partial x_1} + \beta_1 \frac{\partial g_1}{\partial x_1} + \beta_2 \frac{\partial g_2}{\partial x_1} = 0$$

(Eq. 4-50)

$$\frac{\partial F}{\partial x_2} = \frac{\partial f}{\partial x_2} + \beta_1 \frac{\partial g_1}{\partial x_2} + \beta_2 \frac{\partial g_2}{\partial x_2} = 0$$

(Eq. 4-51)

$$\beta_1 g_1 = 0$$

(Eq. 4-52)

$$\beta_2 g_2 = 0$$

These four equations have to be solved for $x^*_1, x^*_2, \beta^*_1, \beta^*_2$. Equations (Eq. 4-51) and (Eq. 4-52) lay out a definite feature for a non trivial solution: either β_i is zero ($g_i \neq 0$) or g_i is zero ($\beta_i \neq 0$). This translates into the following four cases

(Eq. 4-53)

Case a: $\beta_1 = 0 [g_1 < 0]; \beta_2 = 0 [g_2 < 0]$

Case b: $\beta_1 = 0 [g_1 < 0]; \beta_2 \neq 0 [g_2 = 0]$

(Eq. 4-54)

Case c: $\beta_1 \neq 0[g_1 = 0]; \beta_2 = 0[g_2 < 0]$ Case d: $\beta_1 \neq 0[g_1 = 0]; \beta_2 \neq 0[g_2 = 0]$

Equations (Eq. 4-49), (Eq. 4-50), and (Eq. 4-53),(Eq. 4-54) provide four equations to solve for four unknowns of the problem and are the necessary conditions, FOC, termed the Kuhn-Tucker conditions [7]. All four sets must be solved for the solution. The best design is decided by scanning the several solutions and counting the sign of the multipliers. While the sign of the multiplier was ignored for the equality constrained problem, it is included as part of the FOC for the inequality constrained problem. It is important to point out that sign is not a sufficient condition. The value is not so important for optimisation but may be relevant for sensitivity analysis. Generally a positive value of the multiplier indicates that the solution is not a local maximum. Formally verifying a minimum solution requires consideration of the second derivative of the Lagrangian. In practical situations, if the problem is well defined, the positive value of the multiplier usually suggests a minimum solution [12].

4.4.4 Sequential Linear Programming

In weight minimisation problems, orientation variables influence only the strength constraints and, therefore, their convergence behaviour is slow compared to the thickness variables, requiring robust algorithms for full convergence to an optimal solution. Once the design is optimised using continuous variables, the final thicknesses can be rounded off to integer multiples of the commercially layer thickness. Schmit and Farshi [73] were among the first to present minimum weight designs of symmetric laminates subject to multiple in-plane loading conditions taking into account membrane stiffness requirements and strength limitation. The laminate was chosen symmetric balanced with fixed orientation angles.

The particularity of this study and the flexibility to employ into similar configuration, see the Advanced Example, is mainly regarding the way the non-linear mathematical programming problem has been tackled. The non-linear programming formulation is transformed into a sequence of linear programs employing an adaptation of the method

of inscribed hyperspheres in which only critical and near critical constraints are considered at each stage in the procedure [73].

With these settings the optimisation problem is formulated as

$$\text{Minimise} \quad W = \sum_{k=1}^I 2\rho_k t_k \quad (\text{Eq. 4-55})$$

$$\text{Subject to} \quad g_{kj} = (P_j^{(k)} \varepsilon_{1k} + Q_j^{(k)} \varepsilon_{2k} + R_j^{(k)} \gamma_{12k}) - 1 \leq 0 \quad (\text{Eq. 4-56})$$

$$A_{11} \geq A_{11}^{all}, \quad A_{22} \geq A_{22}^{all}, \quad A_{66} \geq A_{66}^{all} \quad (\text{Eq. 4-57})$$

And

$$-t_k \leq 0 \quad (\text{Eq. 4-58})$$

$$\text{For} \quad k = 1, \dots, I, \quad j = 1, \dots, J \quad (\text{Eq. 4-59})$$

Where ρ_k and t_k are the density and the thickness, respectively, of the k th layer, $P_j^{(k)}$, $Q_j^{(k)}$, $R_j^{(k)}$ are coefficients that define the j th boundary of a failure envelope for each layer (k) in the strain space [1], and the ε_{1k} , ε_{2k} , and γ_{12k} are the strains in the principal material direction in the k th layer. Also specified are the limits on the in-plane stiffness terms with the following allowable values: A_{11}^{all} , A_{22}^{all} , and A_{66}^{all} .

Skipping all the detailed steps of the analysis, the turning point is how Schmit and Farshi transformed the non-linear programming problem described in (Eq. 4-56) into a sequence of linear programs, a procedure commonly referred to as sequential linear programming see Haftka and Gurdal [1].

The strength constraint of (Eq. 4-56) is a non linear function of the thickness variables and therefore, is linearised as

(Eq. 4-60)

$$g_{kjl}(t) = g_{kj}(t_0) + \sum_{i=1}^l (t_i - t_{0i}) \left(P_j^{(k)} \frac{\partial \varepsilon_{1k}}{\partial t_i} + Q_j^{(k)} \frac{\partial \varepsilon_{2k}}{\partial t_i} + R_j^{(k)} \frac{\partial \gamma_{12k}}{\partial t_i} \right)$$

Where the $\frac{\partial \varepsilon_{1k}}{\partial t_i}$, $\frac{\partial \varepsilon_{2k}}{\partial t_i}$, and $\frac{\partial \gamma_{12k}}{\partial t_i}$ are the derivatives of the principle material direction

strains in the kth layer with respect to the thickness of the ith layer. The advantage of this approach lays in the fact that the point around which the approximation is expanded, t_0 , is the mid-plane of the laminate and at the same time the derivatives of (Eq. 4-60) are related to the derivatives of the laminate mid-plane strains with respect to the ith layer thickness.

The above equation (Eq. 4-60), well known as Taylor series approximation already described in (Eq. 4-30), makes the response quantities of interest to be rapidly approximated when the input variables are perturbed. The major drawback is that it has small radius of convergence, similar to the perturbation series. As a consequence, the approximations can be very poor for large perturbations of the design variables [74]. Eventually approaches for improving the accuracy of the Taylor series approximation are: intervening variables and multipoint approximations [7].

4.4.5 Genetic Algorithm

In the previous paragraphs has been described mainly technique to deal with continuous Programming but working with integer is a quite common situation for the design of laminates subject to in-plane and out-of-plane stiffness requirements. Optimisation methods for problems with continuous design variables are usually much more efficient than methods that have to handle integer variables. For this reason, it is common for designers to replace integer variables with continuous ones, whenever this does not introduce large errors, and then round up the optimum value of the variable to the nearest integer (techniques employed in the Intermediate example). Then another method to manage integers is Miki's lamination parameter diagram 4.4.2, but can be generalised to few other problems.

So over the year's researcher developed efficient linear integer programming problems but algorithms that deal with non-linear integer programming are more complex to perform in efficient fashion.

Few years ago a single class of algorithm, extensively applicable and with a good background experience in the design of composite laminates, has started to rise up in relevant way, those are the genetic algorithm.

Genetic algorithms use techniques derived from biology and rely on the application of Darwin's principle of survival of the fittest. When a population of biological creatures is allowed to evolve over generations, individual characteristics that are useful for survival tend to be passed on to future generations, because individuals carrying them get more chances to breed. In biological populations these characteristics are stored in chromosomal strings. The mechanics of natural genetics is based on operations that result in structured yet randomized exchange of genetic information between the chromosomal strings of the reproducing parents and consists of reproduction, crossover, occasional mutation, and inversion of the chromosomal strings [1].

Genetic algorithm developed by Holland [75], mimic the mechanics of natural genetics for artificial systems based on operations that are the counterparts of the natural ones. However, genetic algorithms have been experimentally proved to be robust and the reader is referred to Goldberg [76] for discussion of their theoretical properties.

The genetic representation of a minimisation problems focuses on the mechanics of three commonly used genetic operations:

- Namely
- Reproduction
- Crossover
- Mutation
- Permutation or Gene Swap

Unlike many search algorithms that move from one point to another in the design variable space, genetic algorithms work with a population of strings. This aspect of the genetic algorithms is responsible for increased chances of obtaining global or near-global optima. By keeping many solution points that may have the potential of being

close to minima (local or global) in the pool during the search process, rather than converging on a single point early in the process, rather than converging on a single point early in the process, that reduces the risk of converging to a local minimum. Working on a population of designs also suggests the possibility of implementation on parallel computers, 2.6. However, the concept of parallelism is even more basic to genetic algorithms in that evolutionary selection can improve in parallel many different characteristics of the design [1].

Applications of these algorithms to design of thick composite laminates have proved to be reliable in the bioengineering field by Biggers et al. [4]. In fact in the final example of this work, the Advance example, despite it will not be any specific numerical applications, the employ of genetic algorithms has been thought to happen at the ply level to identify, under in-plane and out-of-plane loadings, the best stacking sequence and orientation angles permutation.

4.5 Summary

Once identified the Multi-level Decomposition optimisation approach as core ground of the research and adding all the composite laminate theory of the previous chapter, consequently it has been possible to investigate the main potentialities of different tools. Each optimisation tool has been described and analysed starting from the simpler, Graphical (4.4.1), up to the more complex, Genetic Algorithm (4.4.5), passing through Lamination Parameters (4.4.2), Kuhn-Tucker (4.4.3) and Sequential Linear Programming (4.4.4). All of those will offer different options to tackle different structural laminated design applications as the Basic, Intermediate and Advanced Examples mentioned then.

At the same time the employment of commercial optimiser will be considered as part of the validation task of this work. In fact the comparison of the results obtained by the home-made Matlab code with the Nastran Sol 200 Optimisation solutions will be performed to increase the confidence of the methodology proposed and to point out the strengths and weaknesses.

This thesis from now onwards, will proceed at a brisk pace to provide experience in mixing the main tools presented in this chapter by using Matlab programming and communicating ideas in coupling optimisations through a reasonable spectrum of examples. The optimisation concepts and programming issues will be incrementally explored and implemented mainly by relying on the direct extrapolation, graphical, of the concepts and ideas developed for two variables, Basic Example, up to any number of Design Variable, Intermediate and Advanced Examples.

5 Basic Example

A Multilevel-Decomposition, as already anticipated at the very beginning of this work, will be employed for developing the whole work of this thesis. In fact the Basic Example, according to the approach methodology, 1.4, will follow the same main structure anticipated by Figure 1-3. A two-level decomposition framework will be chosen to embed design and optimisation concepts of a laminate structure properly coupled in iterative way by Matlab programming code, see Figure 5-1 .

The aims of this example will be to explore the very basic optimisation methods applicable by graphical tools and then visual inspection understanding the best way to couple between the two decomposed levels. At the same time create the structure for an extra coupling mainly due to validation box to be performed at each level. Finally define and judge the efficiency of a specific convergence test.

All these tasks will be attempted to be achieved and tested by a specific real life application. In fact seeking the lightest weight of a symmetric and balanced laminates, four layers overall ($\pm\vartheta_1/\pm\vartheta_2$), under strain limitations and in-plane loadings is the set up of the numerical example presented at the end of this chapter.

Discussion of the quality of the final results will be determinant to set improvement terms to be developed, Intermediate Example, presented on the next chapter.

5.1 Optimisation Procedure

Finding the optimal design within a two-level decomposition structure, Figure 5-1, will require handling a three-step process. First, formulate the design in terms specific to the problem at hand. Second, recast the problem as a mathematical optimisation problem.

Finally, solve the problem using the best optimisation problems at each level. Each of those steps will have to combine properly the main tools available to this specific example:

- Graphical Optimisation (4.4.1)
- Classical Lamination Theory (3.6.1.5)
- Finite Element Model (3.6.3)

According to Figure 1-4, each level data flow will go through the optimisation stage, and then after visual inspection of graphical plots it will be possible to point out the optimal design (OD) expected to be passed on the next level once the validation test is successively satisfied.

The structure of the basic example, broken down in Figure 5-1, also enables to identify the other main quantities part of the standard optimisation definition as already explained in 4.2. The natural choice of design variables for this problem is ply thicknesses, t_1, t_2 , and orientation angles ϑ_1, ϑ_2 . The other crucial information to draw attention out from the table associated to Figure 5-1 is the particular link existing between Level 1 Constraint function and Level 2 Objective function. The quantity chosen, normal strain is determinant within the whole process since will couple the two levels but working differently every time. At L1 is the constraint function defined by the laminate thicknesses on the other hand at L2 becomes the objective function expressed in terms of the orientation angles and has to be maximised. This technique allows setting an inherent convergence test of the all optimisation process.

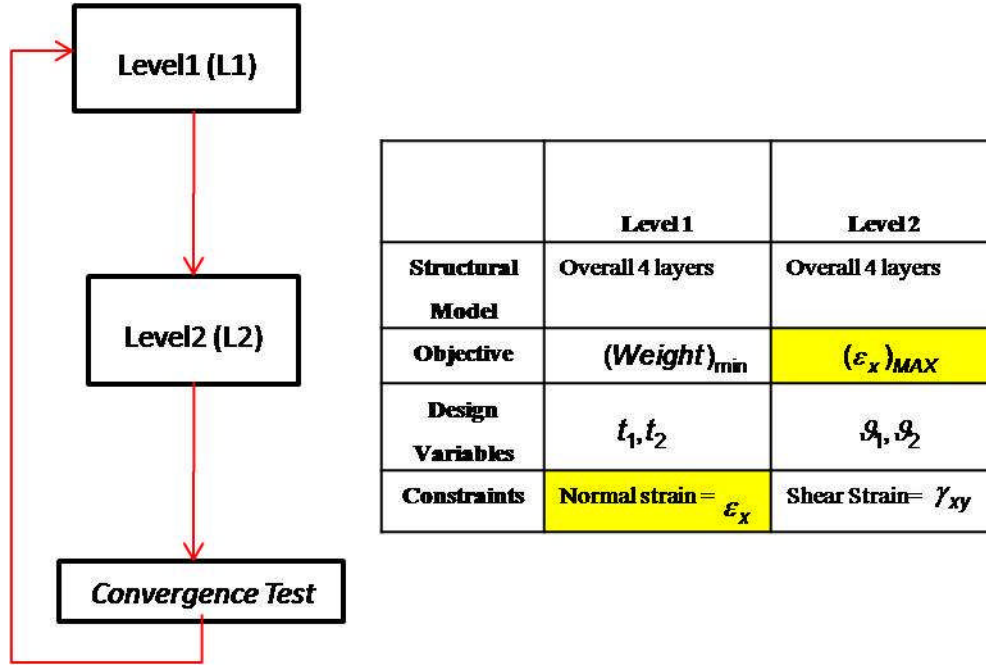


Figure 5-1 – Multilevel Decomposition structure: Basic Example two-level decomposition

The optimisation problem showed in Figure 5-1 mathematically states as:

Level 1

Find t_1, t_2

Minimise $f(t) \rightarrow (Weight)_{\min} \quad t_1, t_2 \in \mathcal{R}$

Subject to

$$\varepsilon_x(t_1, t_2) < (\bar{\varepsilon}_x)$$

$$0 \leq t \leq \mathcal{R}^+$$

Level 2

Find ϑ_1, ϑ_2

Maximise $f(\vartheta) \rightarrow (\varepsilon_x)_{MAX} \quad \vartheta_1, \vartheta_2 \in \pm Deg.$

Subject to

$$\gamma_{xy}(\vartheta_1, \vartheta_2) < (\bar{\gamma}_{xy})$$

$$\forall \vartheta \in \pm Deg.$$

A further description of the generic structure of a i-level, going into more details than Figure 1-4, is showed in Figure 5-2 where it is possible to identify the following main blocks:

- **A:** Analytical model
- **B:** Numerical and Graphical Optimiser
- **C:** Numerical Model (CLT, FEM descriptions)
- **V:** Validation Test (Constraints Check)
- **OD:** Optimum Design

These will be included in the next sections description as part of the Matlab programming code. The full details of the code, the complete Matlab lines, are attached in Appendix A.

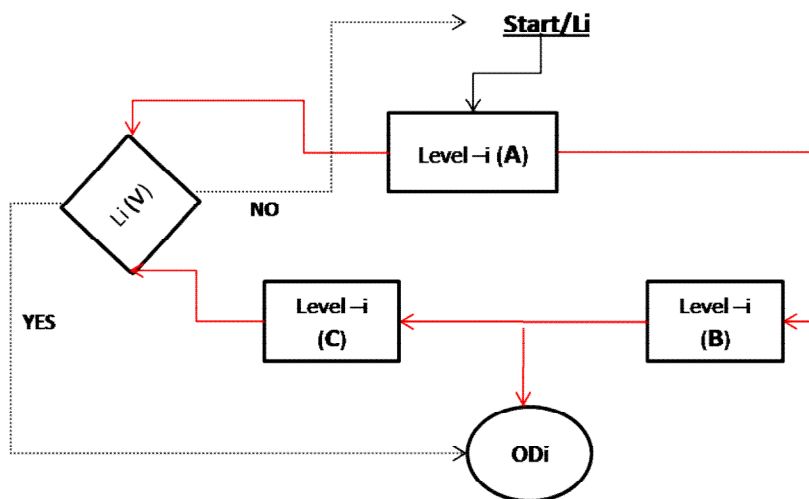


Figure 5-2 Multi-level Decomposition Single Level Structure

Since all the above boxes are applicable at both levels, inside each coming outlines will be clearly highlighted the different characteristics between L1 and L2. To afford the reader a thorough understanding of the way chosen to present these tasks a common organisation of the coming block sections is following:

- General Box Description by Flow Chart or Table
- Level1 Description
- Level2 Description
- Matlab Editor Script Description
 - ✓ Level1 and Level2 invariants
 - ✓ Level1 variables and invariants
 - ✓ Level2 variables and invariants

Next part of the present section is meant to take into account a numerical application of a Graphite/Epoxy laminate subjected to two load conditions N_x, N_{xy} . For completeness and consistency the following information is given:

Material properties [1]:

- $E_1=128$ [Gpa] (Eq. 5-1)
- $E_2=13$ [Gpa]
- $G_{12}=6.4$ [Gpa]
- $\nu_{12}=0.3$

Load Cases:

$$N_x = (175.1) \frac{N}{mm} \quad \text{(Eq. 5-2)}$$

$$N_{xy} = (52.8) \frac{N}{mm} \quad \text{(Eq. 5-3)}$$

The allowable laminate strains are 0.4% normal strain and 0.6% shear strain. Consider the following laminate structure as candidate, according to the terminology introduced in 3.5, $[(\pm\theta_1)(\pm\theta_2)]_S$. A complete representation of the potential candidate is showed in the following Figure 5-3.

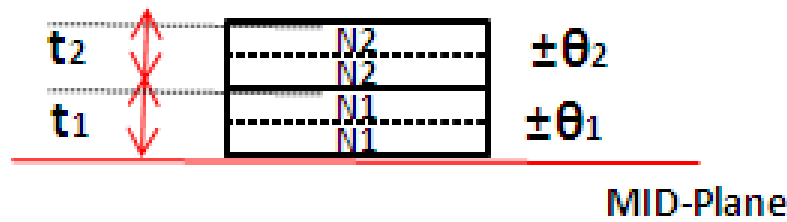


Figure 5-3 Basic Example: Half Laminate Orientation Group Numbering Convention

5.1.1 Analytical Model (A)

It should be understood that optimisation presupposes the knowledge of the design rules for the specific problem, primarily the ability to describe the design in mathematical terms. These terms include design variables, objective function(s), design parameters and constraint equation(s).

The mathematical terms of the object, unidirectional symmetric and balanced laminate, need to be expressed in terms of the design variables of the specific level while the design variables of the other level are invariants. The whole lot of equations have been already developed in section 3, Laminated Composite Materials, and 4, Optimisation.

All this information is collected in Table 5-1, see 4.2 also, referring to those quantities in the following ways:

- Objective Functions : **f(X)**
- Design Variables : **X**
- Constraint Functions : **g(X)**
- Design Parameters : **P**

All the Bold Characters relate to vector quantities.

	Level 1	Level 2
f(X)	$(\Sigma t_i)_{\min}$	$[\varepsilon_x(t_1, t_2)]_{\max}$
X	t_i	ϑ_i
g(X)	$\varepsilon_x(t_i)$	$\gamma_{xy}(\vartheta_i)$
P	L_i, G_i, MP_i	L_i, G_i, MP_i

Table 5-1 “Basic Example”: Optimisation Problem Definition

5.1.1.1 LEVEL 1

Design Variables – $\mathbf{X} = t_i$

At the first step of the Multilevel Decomposition the design variables for the balanced symmetric laminate are the stack thickness of each group of orientation angles $\pm\vartheta_i$. Since four are the overall number of stack thicknesses it will be analysed just half laminate and then possible to have just two design variables making the problem analysable in the graphical design space. Reminding that the laminate is a symmetric and balanced the “stack thickness” t_i will be designated as (t_{ply} is the ply the thickness of the material layer):

(Eq. 5-4)

$$\begin{aligned}
 & - \vartheta_i = 0^\circ \text{ or } 90^\circ \text{ then } t_i = 1 * N_i (t_{ply}) \\
 t_i \Rightarrow & \\
 & - \vartheta_i \neq 0^\circ \text{ or } 90^\circ \text{ then } t_i = 2 * N_i (t_{ply})
 \end{aligned}$$

Where N_i are the total numbers of plies of the orientation angles $+\vartheta_i$ (and $-\vartheta_i$) according to Figure 5-3 and the factor 1 and 2 is mainly part of the balanced condition that will lead to have a membrane stiffness matrix ($\mathbf{A}$) with $A_{16} = A_{26} = 0$ already discussed by (Eq. 3-50).

Design Variable Side Constraints – $\mathbf{X} = t_i$

The design variables need to satisfy basic constraints mainly related to the design model adopted and are essential to drive the search for the optimal design.

The stack thicknesses need to be positive to allow the validity of the design. Important to make clear the full awareness that this degree of freedom will lead to design a final 4 layers laminate with thickness not close to real commercial values.

$$\text{Objective Function} - \mathbf{f}(\mathbf{X}) = (\Sigma t_i)_{\min}$$

The objective of this optimisation level is to seek the lightest laminate. In fact the laminate weight is meant to be minimised. Since the material involved in the analysis will be always the same, see numerical application for more details, for simplicity the objective function to minimise will be a linear equation of the ply thicknesses as described below:

$$\mathbf{f}(\mathbf{X}) = 2(t_1 + t_2) \quad (\text{Eq. 5-5})$$

$$\text{Design Parameters} - \mathbf{P} = (L_i, G_i, MP_i)$$

Design Parameters identify constants that will not change as different designs are compared. They are significant in the discussion of modelling the optimisation problem. Examples of parameters include:

- material property
- applied loads
- geometry quantities

It is very important to emphasise that the above parameters are referred to all those quantities primarily predetermined constants in the design process. In fact, design parameters are invariants at any step of the MD iterations. For instance, the material property of the whole laminate, 3.6.1.5 effective engineering properties, is not an invariant, since it depends to the terms of membrane stiffness matrix and at each level is upgraded by using Classical Lamination Theory. On the other hand, material properties

of the single layer will be always the same. Same reason can be said to the laminate thickness (function of the design variables) and individual ply thickness (invariant).

So applied loads, N_x and N_{xy} , thickness and material property of the single ply are design parameters frozen throughout the optimisation process.

$$\underline{\text{Constraint Functions}} - \mathbf{g}(\mathbf{X}) = \varepsilon_x$$

Constraint functions are described in terms of strain criteria, so they will be characterised by inequality equations. Considering the load design parameters, N_x and N_{xy} , the normal strain and shear strain functions, ε_x and γ_{xy} , are the equations to be considered respectively at Level 1 and Level 2.

As a summary to Section 3, the normal strain equation in terms of level 1 design variables and design parameters is the following:

$$\varepsilon_x = \frac{N_x}{hE_x^{eff}} \quad (\text{Eq. 5-6})$$

Since E_x^{eff} , the Longitudinal Engineering Effective Constants of the laminate is equal to:

$$E_x^{eff} = \frac{1}{h} \left(\frac{A_{11}A_{22} - A_{12}^2}{A_{22}} \right) \quad (\text{Eq. 5-7})$$

The final normal strain equation, ε_x in terms of the design variables, t_i is:

$$\varepsilon_x = \frac{N_x A_{22}}{A_{11}A_{22} - A_{12}^2} \quad (\text{Eq. 5-8})$$

$$A_{ij} = \sum \bar{Q}_{ij(k)} t_k \quad (\text{Eq. 5-9})$$

All the above equations are a natural consequence of the Laminated Composite Materials Section, 3.6.1, and:

- k is the number of orientation angle groups here, equal to 2 see (Eq. 5-4)
- $\bar{Q}_{ij(k)}$ is the transformed reduced stiffness matrix see (Eq. 3-45)

5.1.1.2 LEVEL 2

$$\underline{\text{Design Variables}} - \mathbf{X} = \vartheta_1, \vartheta_2$$

The design variables at the second level of the Multilevel Decomposition basic example are the two orientation angles of half part of the symmetric balanced laminate , ϑ_1 and ϑ_2 .

Considering (Eq. 5-4) it is well understandable the orientation angles are inside a range of plus and minus 90-degree and for Matlab programming reasons the values will be radians.

Apart the eventuality of dealing with 0° or 90° orientation angles every orientation angles coming out as optimum design, OD, it is always associated with its negative. In fact the final configuration laminate will foresee a $-\vartheta$ layer for each $+\vartheta$ layer.

$$\underline{\text{Design Variable Side Constraints}} - \mathbf{X} = \vartheta_1, \vartheta_2$$

Considering what just mentioned above the Level 2 design variables define a design space region bounded by:

$$-90^\circ \leq \vartheta_i \leq 90^\circ \quad (\text{Eq. 5-10})$$

Where “i” is the couple of orientation angles present in half laminate. But since the use of radians the final side constraints are introduced below:

$$-1.5 \leq \vartheta_i \leq 1.5 \quad (\text{Eq. 5-11})$$

In such way the dimension of the design space are significantly shrunk making less complex the analysis of the plots and the choice of the optimum design.

$$\underline{\text{Objective Function}} - \mathbf{f}(\mathbf{X}) = [\varepsilon_x(\vartheta_1, \vartheta_2)]$$

As already mentioned before and highlighted by the yellow bar of Figure 5-1 the objective function of Level 2 has been chosen as the constraint function of Level 1. Moreover as convergence requirement this objective function is expected to be maximised at each optimisation step just to get narrow the research space iteration by iteration.

The normal strain, already defined by the stress-strain equations of an orthotropic laminate in section 3.6.1.4, has to be expressed in terms of the orientation angle design variables, ϑ_1 and ϑ_2 .

So the main equation employed next to perform the graphical optimisation in the lines of Matlab code comes from:

$$\varepsilon_x = \frac{N_x}{hE_x^{eff}} \quad (\text{Eq. 5-12})$$

So the final objective function will come out from employing the above equation (Eq. 5-12) alongside the following, previously explained, equations:

- (Eq. 5-7) Effective Engineering Quantity (E_x^{eff})
- (Eq. 5-8) Normal strain in function of (A_{ij})
- (Eq. 5-9) In-plane stiffness coefficients (A_{ij}) in terms of ($\bar{Q}_{ij}$)
- (Eq. 3-45) Transformed Reduced Stiffness Coefficients, ($\bar{Q}_{ij}$), express in terms of the ply orientation angle

Design Parameters – $\mathbf{P} = (L_i, G_i, MP_i)$

As mentioned for the Level 1 design, applied loads, geometry quantities and material property of the single layer will be invariant through the whole optimisation process.

Constraint Functions – $\mathbf{g}(\mathbf{X}) = \gamma_{xy}$

To constrain the optimisation of this level, L2, the shear strain γ_{xy} has been chosen as function.

To embed the constraint function within the Matlab code is necessary to identify and develop the equation in terms of the design variables of this level, the two orientation angles ϑ_1 and ϑ_2 .

This will be doable going through the next equations.

The shear strain is defined as:

$$\gamma_{xy} = \frac{N_{xy}}{hG_{xy}} \quad (\text{Eq. 5-13})$$

Then G_{xy}^{eff} is the in-plane shear modulus of the whole laminate, as E_x^{eff} , and is equal to:

$$G_{xy}^{eff} = \frac{A_{66}}{h} \quad (\text{Eq. 5-14})$$

as it has already explained in (Eq. 3-37).

The employing the following two main set of equations it is obtained the main constraint function:

- (Eq. 5-9) In-plane stiffness coefficients (A_{ij}) in terms of ($\bar{Q}_{ij}$)
- (Eq. 3-45) Transformed Reduced Stiffness Coefficients, ($\bar{Q}_{ij}$), express in terms of the ply orientation angle

It is peculiar to highlight how this L2 constraint function is governed by a different set of load case not any more the normal axial N_x but the shear loading N_{xy} .

5.1.1.3 MATLAB SCRIPT: BOX A

From now onwards, applicable to the next Intermediate Example also, the sections related to the Matlab Script description will be structured to enable the reader to correlate the main boxes of Figure 5-2 with the respective lines of the Matlab script pointing out the main Matlab variables and invariants for each level; L1 and L2. All these details will be possible to be found inside the lines of the program going to the Appendix where is attached the relevant Matlab script and related files.

The Program starts from the main Matlab file, named **BE_Main.m** that stands for Basic Example Main (in Appendix A: "Basic Example Matlab Scripts" are grouped all the lines of the code), with a sign, "%", followed by a note that introduces the respective **"%Box A:Analytical Model"**.

In fact after the "%" sign will follow the denomination of the box meant to initiate its first line. In Matlab all text after the "%" sign is considered comment. Matlab ignores anything to the right of the percentage sign. Comments will be used liberally in this work to identify and indicate why's and what's of command or code usage [77].

Then for making more direct the understanding when referring to Matlab variables of data in support of it will be employed the following rules:

- Matlab variables are coming with same size (10) and character (Courier New) as the Matlab Editor does
- External files loaded by Matlab will be in Bold Characters
- Each citation to Matlab Variables or External Files will be gathered and labelled as citation caption

Now it is going to be listed the *Input Starting Values*, invariant values, which will feed the Objective and Constraint Functions to be plotted at the next Graphical Optimisation **Box B:**

Level1 and Level2 Invariants:

- N_x and N_{xy} loadings, dimensionally loads over width $\left[N_x = \frac{\text{Newton}}{\text{mm}} \right]$
- Cons1 and Cons2 a-dimensional values of the inequality conditions
- Single Layer Material Properties defined by a **P** vector populated as follows:

$$P = [E_{11} \ E_{22} \ G_{12} \ \nu_{12}]^T$$

- ✓ E_{11} : In-plane longitudinal modulus
- ✓ E_{22} : In-plane transverse modulus
- ✓ G_{12} : In-plane shear modulus
- ✓ ν_{12} : Poisson Ratio

These values are saved and took back in the Matlab program from a text file. The text file is named **P_12_c_Mpa_1.dat**. The Matlab command to call back this data is where P is the vector where the values are stored.

Matlab Command:

```
load P_12_c_Mpa_1.dat
P=P_12_c_Mpa_1
```

Level1 Variables and Invariants:

- x_1, x_2 : are the design variables of L1, thickness of the two layers, and define the design space. The convention of this numbering is shown in Figure 5-3. At the end of this level the optimum design (OD1) is stored in a vector **T** and in a text file thick_OD_1.txt.

Matlab Command:

```
load thick_OD_1.txt
T=thick_OD_1
```

- e_1, e_2 : are the invariant values of this level. They are the orientation angles associated respectively to the layers of variable thickness x_1, x_2 .

At L1 those are stored in a text file, **angl1_c.dat**, and saved in a Matlab vector `ply_angle`. The Matlab command follows:

Matlab Command:

```
load angl1_c.dat  
ply_angle=angl1_c
```

Level2 Variables and Invariants:

- As already mentioned at the very beginning of this chapter and as main characteristic of this optimisation approach the design variables of one level are the invariants at the next level. Accepting that, $(x1;x2)$ and $(e1; e2)$ previous considerations are still applicable but inverted. It means at this level 2 the invariants are the thicknesses and the design variables are the orientation angles that at the very end of this level will lead to the optimum design, OD2.

As just described Ply angles, Ply Properties, and Stack Thicknesses are stored at each level as Matlab variables and text file to make easier the mechanism of recall and the need of further checking.

5.1.2 Graphical Optimiser (B)

Moving down from the Box A, as clearly showed in Figure 5-2, the data flow comes across the optimiser, Box B. As mentioned before the optimiser of the Basic Example, either at level 1 or level 2, is meant to be graphical. Moreover it will be supported by the Matlab rules and features to finally enable the user to make high quality through visual inspection and to point out the Optimum Design for that specific level.

The Graphical Solution according to Matlab will need:

- To create and run Matlab Script file

- To set Function Matlab Files

So basically two main Matlab file categories will be created, script and function files.

The “*Script file*” contains a list of Matlab commands that are executed in sequence. Within this script will be embedded the all different boxes mentioned before for first and second multilevel decomposition.

On the other hand the “*function m-file*” similarly to other programming languages as C, Java or FORTRAN represents subroutines. Mainly these m-type files are used to handle some specific type of calculations. They also provide a way for the modular development of code as well as code reuse. These code modules are used by being called or referred in other sections of the code, for instance through a script file as mentioned earlier. The code that calls the function m-file is called the calling program/code. The crucial parameters in developing the function m-file are:

- What input is necessary for the calculation
- What specific calculation must take place
- What information must be returned to the calling program

Matlab requires the structure of the function m-file to follow a prescribed format.

Before describing thoroughly the use of the Matlab Function File as a Graphical Optimiser tool, it will be convenient to describe the particular structure required.

In the first line, the first word starting from the first column is the word “function”. It is followed by the set of return parameters (*returnval*). Next, the “*name*” of the function with the parameters passed to the function within parentheses. The file must be saved as “name.m”. An example of the simplest described structure is below:

Matlab Command:

```
function retval=obj_m1(X1,X2)
retval=2*X1+2*X2
```

The above lines come from the level 1 objective function and x1,x2 are the design variables already mentioned in 5.1.1.1.

Since the 2D functions to be plotted are objective functions (L1 and L2) and constraint functions (L1 and L2) it is possible to identify the name of these external function files in the following table:

Basic Example (2D Plots)	Level1 (L1)	Level2 (L2)
-f : objective function	obj_m1.m	obj_m2.m
-g : constraint function	inecon_m1.m	inecon_m2.m

Table 5-2 Basic Example: function m-file labels at Level1 and Level2, 2D Plots

The flowchart of what just described is pictured in Figure 5-4

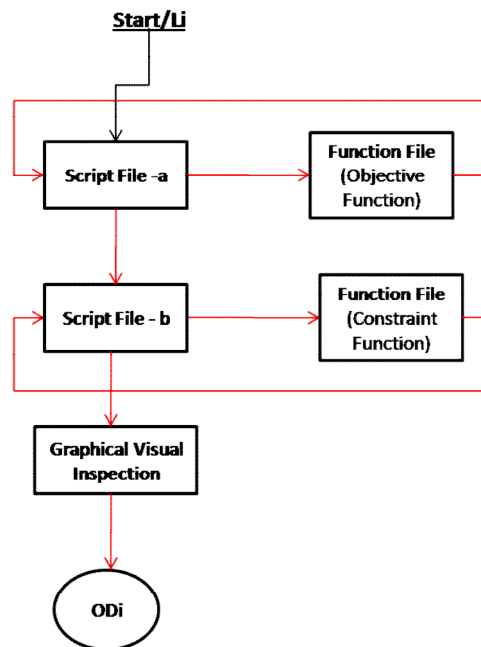


Figure 5-4 Basic Example, Graphical Optimiser: Matlab Structure

Where:

- Script file-a and b are part of the calling-program file
- OD: optimum design coming from the graphical visual inspection of the feasible domain and meant to go the next box, BOX C (CLT/FEM).

Despite will be discussed later it is still worth mentioning that

Table 5-2 describes the function m-files employed in the whole program as 2D plots but it will come later pictures of 3D plots too. These three dimensional figures will be

employed mainly at level two to enable the user an easier interpretation of the two dimensional graphics in the design space of the orientation angles that as it will be proved later it looks much more complex compared to the thicknesses design space of level 1.

The Matlab 3D function m-files of level 2 are listed below:

Basic Example (3D plots)	Level1 (L1)	Level2 (L2)
-f : objective function	Not Applicable	obj_f2.m
-g : constraint function	Not Applicable	cons_c2.m

Table 5-3 Basic Example: function m-file labels at Level2, 3D Plots

The general Graphical Matlab Optimiser structure just described will be applicable for both levels. For a thorough interpretation of the coming plots either at L1 or L2 it has the laminate under consideration will satisfy the main differences are as follows.

5.1.2.1 LEVEL 1

The graphical optimisation plot of level 1, Figure 5-7, represents the thicknesses design space during the first design cycle. The laminate configuration and boundary conditions are the same anticipated by (Eq. 5-1), (Eq. 5-2), and (Eq. 5-3). The invariants and starting values at this level are the orientation angles chosen, according to Figure 5-3 convention, as:

- $\vartheta_1 = 45^\circ$
- $\vartheta_2 = 22^\circ$

These angles were chosen considering a laminate configuration that might offer good strength either for N_x and N_{xy} load cases so that to enable an optimisation process capable to explore thoroughly both levels. So, the 45° ply orientation angle represents a common advantageous direction for both set of loading conditions. On the other hand, before selecting the 22° fibre inclination, it has been performed a parametric analysis, Figure 5-5 and Figure 5-6, driven by the search of the best optimum design, lowest

overall thickness, within the stack thickness design space. In fact, Figure 5-5 describes clearly that the more the ϑ_2 increase and consequently the higher is the slope of the dotted line $g1$, constraint function of level 1, and so offering higher thickness laminate combinations. Then, Figure 5-6 confirms the same trend just mentioned and enables to select $\vartheta_2 = 22^\circ$ as a reasonable halfway choice in particular once expected to identify thorough visual inspection the optimum design of level 1 (OD1), described later on.

This final choice is also due to the fact the constraint and load in x direction, level 1, are more significative than the respective of level 2, see (Eq. 5-2) and (Eq. 5-3), and so the attempt to start setting a candidate laminate specifically suitable to N_x loadings.

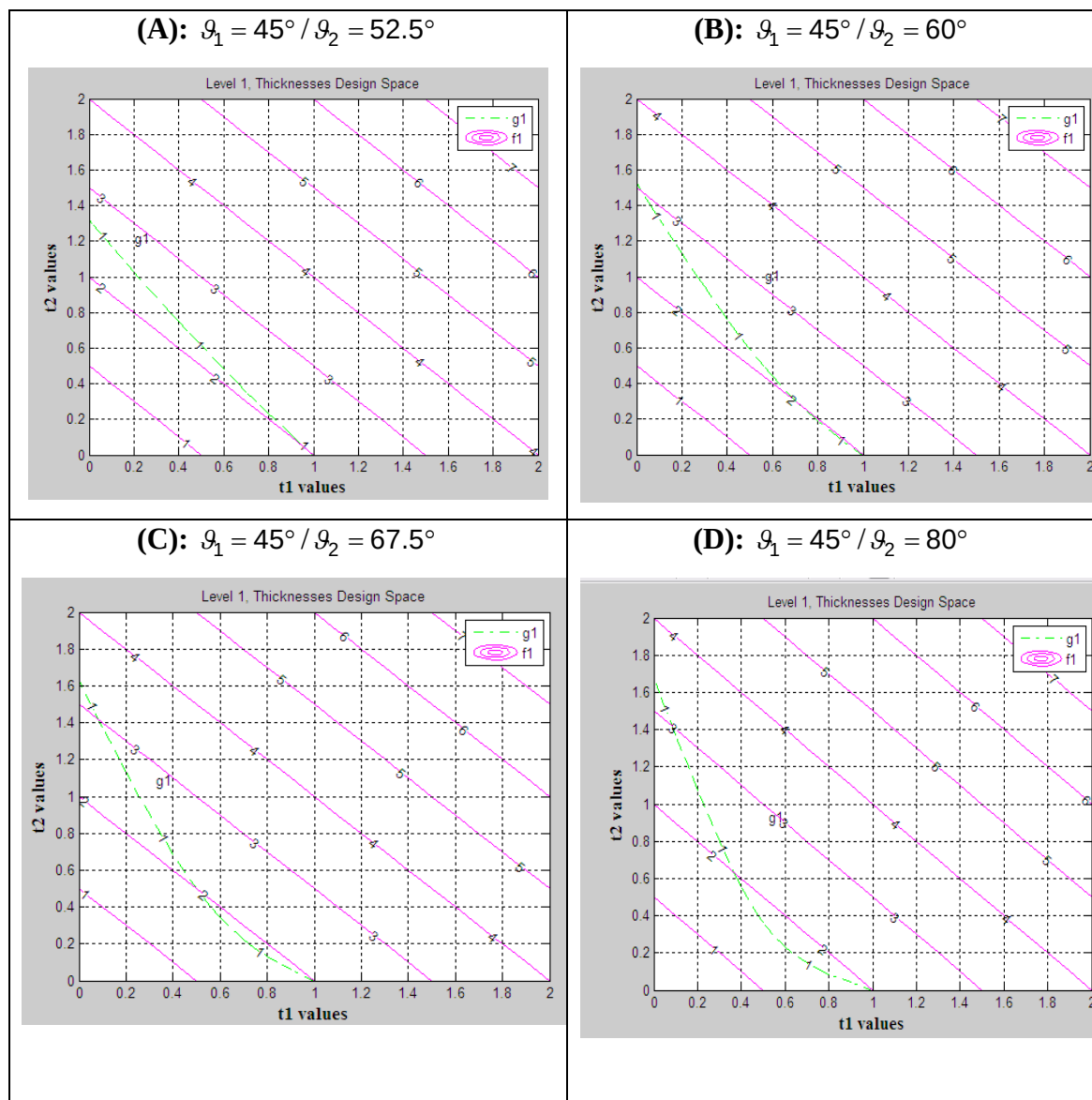


Figure 5-5 ϑ_2 initial value parametric analysis, $\vartheta_2 > 45^\circ$

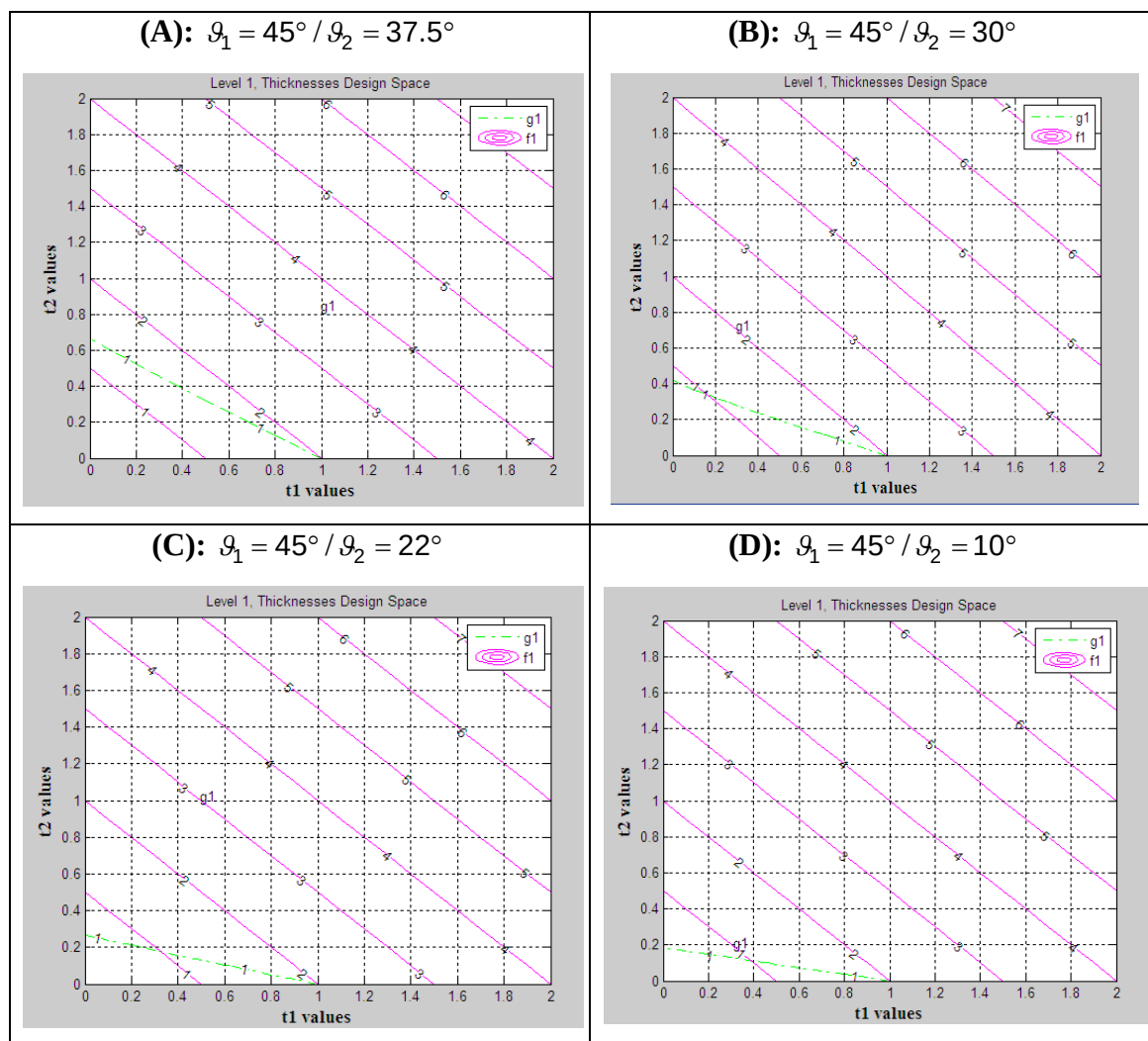


Figure 5-6 ϑ_2 initial value parametric analysis, $\vartheta_2 < 45^\circ$

Figure 5-7 shows the design variables mentioned earlier, as stack thicknesses of the symmetric laminate under consideration. The horizontal axis represents t_1 , the thickness of the first ply from the mid-plane and the vertical axis, t_2 is the ply thickness of the first ply from the top.

The range of the plot matches with the side constraints of the problem, in this specific example chosen as 2 millimetres on both axes, to enable the user to start having a clear and broad picture of the design space.

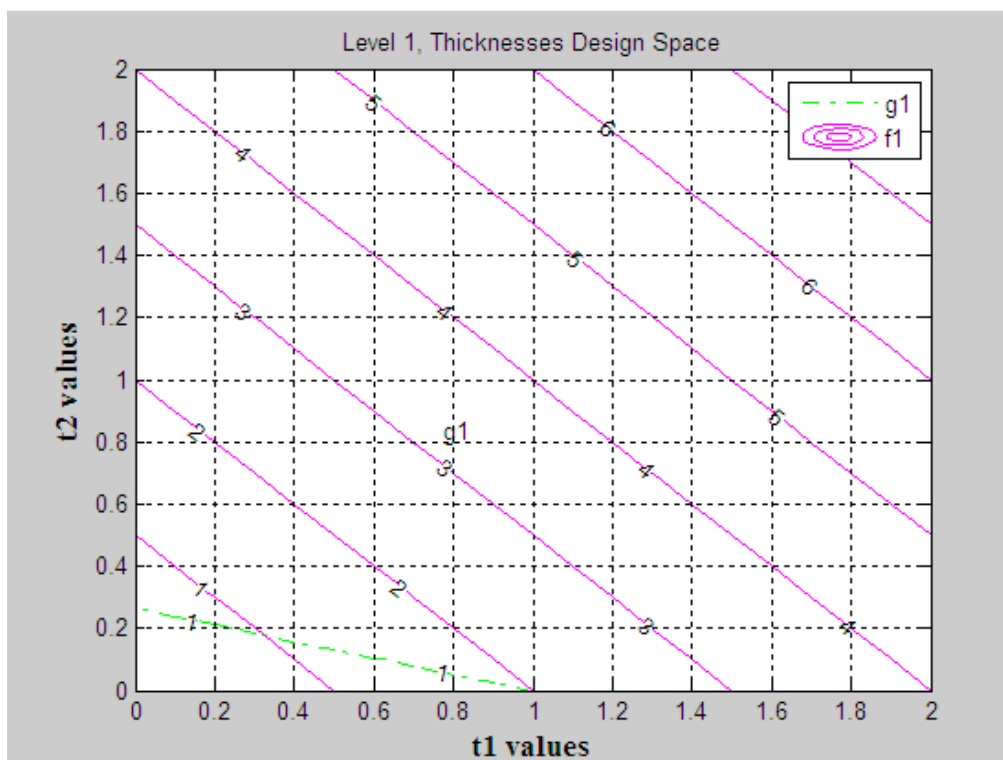


Figure 5-7 - Basic Example: Design Space Level 1

The inequality constraint of the level1 design problem, normal strain, is a dash-dot-line, marked appropriately as g1. Any pair of values on this line will give a ratio of the constraint value, 0.4%, labelled as “Cons1” within the Matlab script, and the normal strain. In the plot this ratio is equal to one as the number typed next to the curve. Since the normal strain condition, has been expressed by that ratio equation to satisfy the inequality conditions mentioned in section 5.1, the pair of thickness values has to prove to have the ratio, $\left(\frac{\text{Cons1}}{\varepsilon_x}\right)$, greater than 1.

This requirement is achievable using the Matlab cursor showed by Figure 5-8, where it is expected to be found the “Level” greater than 1.

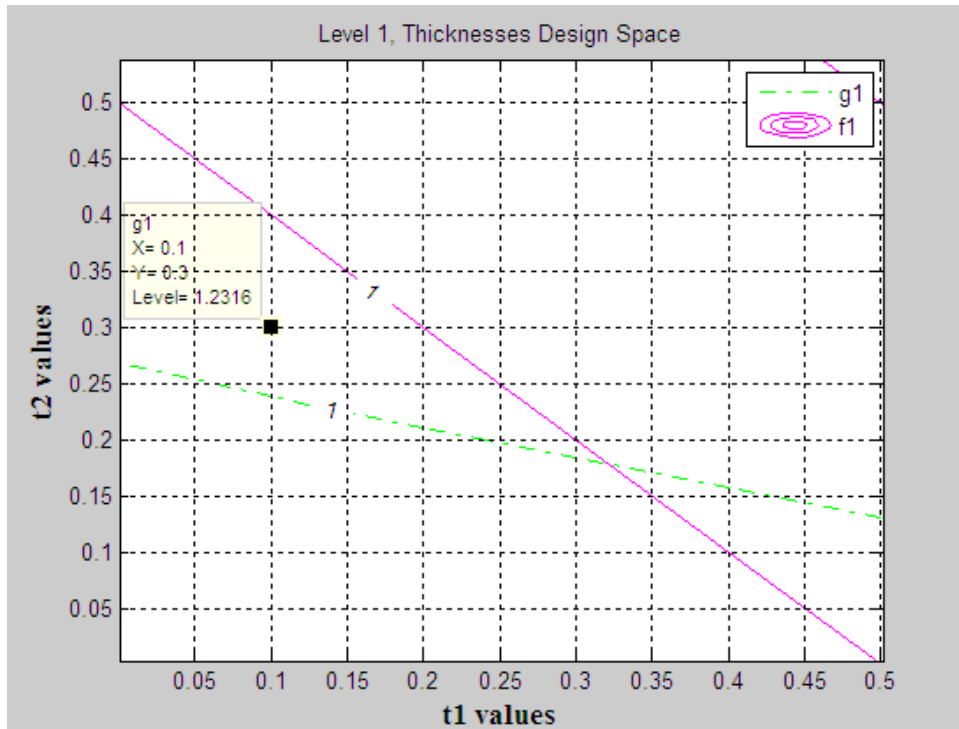


Figure 5-8 - Basic Example: Zoomed Design Space, "g1" Constraint Function - level 1

The scaled objective function of level 1, the total laminate thickness, is a solid line represented through several labelled contours spottable in Figure 5-7. Each contour is associated with a fixed value of the objective function from 7 to 1 millimetre.

Now to pick the optimum design through visual inspection is necessary to characterise the feasible space. Then it is not allowed to stay below the dash-dot-line but at the same time it is expected to choose pair values that give the smallest possible objective function f_1 . This logic allows defining the feasible space of the optimisation within this design space.

Figure 5-9 enables to point out a potential candidate as optimum design (OD), highlighting the thickness magnitudes, expressed as X and Y , and the value of the objective function, f_1 , through the application of the cursor data described just earlier.

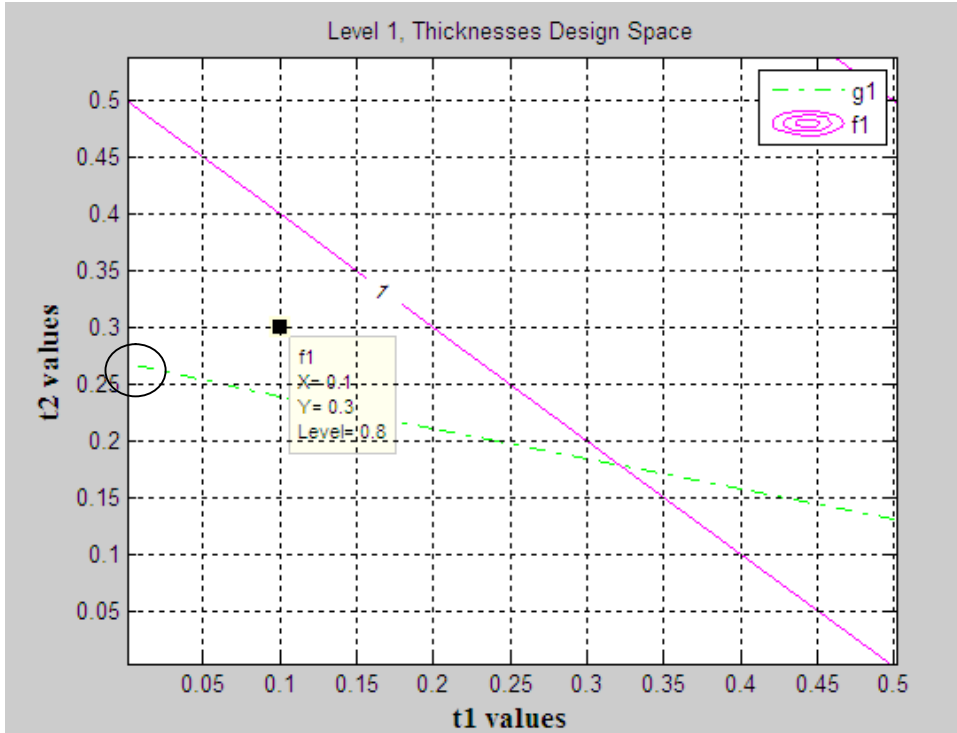


Figure 5-9 - Basic Example: “f1” Objective Function Visual Inspection - level 1

These judgements lead to consider a candidate optimum design as closer as possible to the minimum objective function line affordable to the inequality constraint condition but still satisfying the initial laminate configuration. For such reason a point identified by a circle of the Figure 5-9 y axis is not acceptable since, though it will minimise best the objective function, it will create a single ply laminate configuration ($t_1 = 0$).

The following thicknesses have been chosen as Optimum Design of L1:

(Eq. 5-15)

- $t_{1OD} = 0.1\text{mm}$
- $t_{2OD} = 0.3\text{mm}$

This choice is driven by the same criteria adopted to select $\vartheta_2 = 22^\circ$, try to identify the lightest laminate, but at the same time defining a configuration that satisfies the constraint, above the dotted line of Figure 5-9, and still gives enough margin of the design space to be explored at the next iteration.

The t_{1OD}, t_{2OD} will be passed on the next level, L2, to perform another graphical optimisation but in a different design space.

5.1.2.2 LEVEL 2

The graphical problem of level 1 presented a quite smooth research of the feasible domain and consequent identification of the optimum design. On the other hand level 2 is going to be more complex since, as already described by the table of Figure 5-1, the design variables this time are meant to be the orientation angles. In fact objective function, normal strain, and inequality constrain, shear strain, this time are both non linear equations expressed in terms of the two orientation angles ϑ_1, ϑ_2 .

In contrast to level 1 plots the level 2 design space showed in Figure 5-10 proves to be characterised by not straight lines, either to objective or inequality constraint functions.

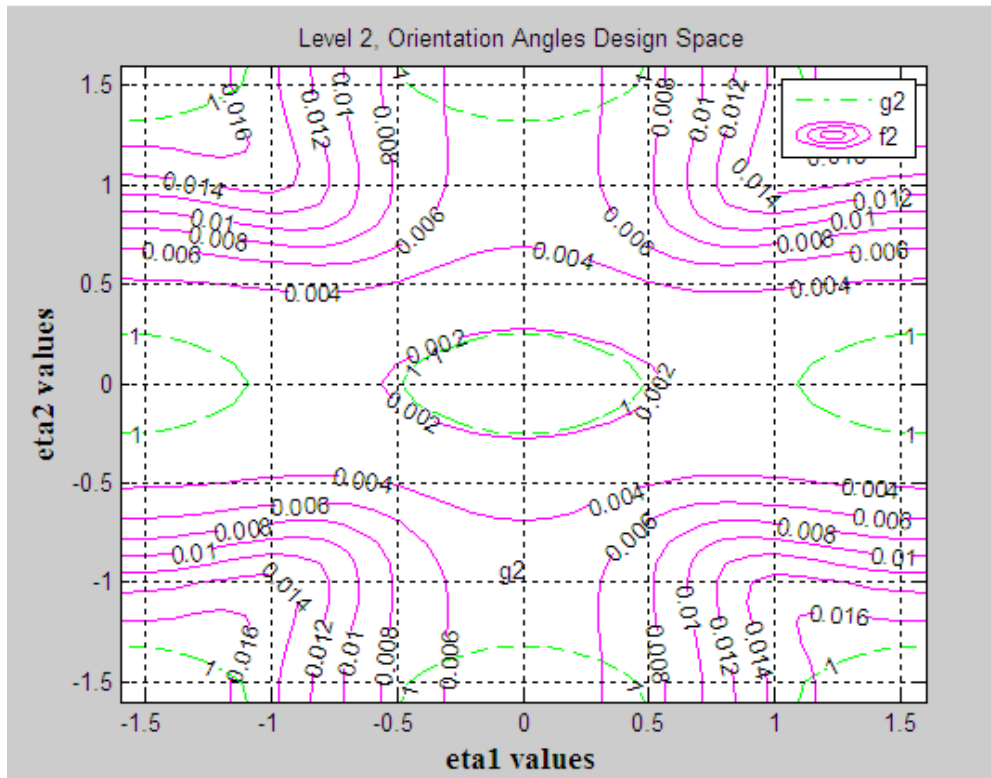


Figure 5-10 - Basic Example: Design Space Level 2

The horizontal and vertical axis are the orientation angles, ϑ_1, ϑ_2 , expressed in radians and the side constraints allow values inside the range of -90° and $+90^\circ$. This range expressed in radians is equivalent to: $-1.571 < \vartheta_1, \vartheta_2 < +1.571$.

Again, several solid line contours of the objective function f_2 are drawn to aid the recognition of the solution. The range, from 0.002 up to 0.016 micro-strains, of these contours is not a casual choice but is meant to count in the limitation, as highlighted in

section 5.1 and in the previous level, of 0.4% of the normal strain within the overall design optimisation process.

All of the inequalities, g_2 , are in dash-dot-line indicating a specific ratio equal to one. This time the ratio is a sort of the inverse of the level 1 expression explained earlier. In fact it is given by the shear strain divided by the shear strain limitation value anticipated in section 5.1, and labelled “Cons2” within the Matlab program. The final expression is

$$\left(\frac{\gamma_{xy}}{\text{Cons2}} \right) \text{ and to satisfy this inequality condition has to be lower than 1.}$$

The feasible region is expected to be enclosed by values, using the same data cursor brought up in Figure 5-8 and Figure 5-9, presenting respectively the “Level” lower 1, Figure 5-11, for inequality function g_2 and as much as possible closer and lower to 0.004 micro-strains for the objective line contours f_2 .

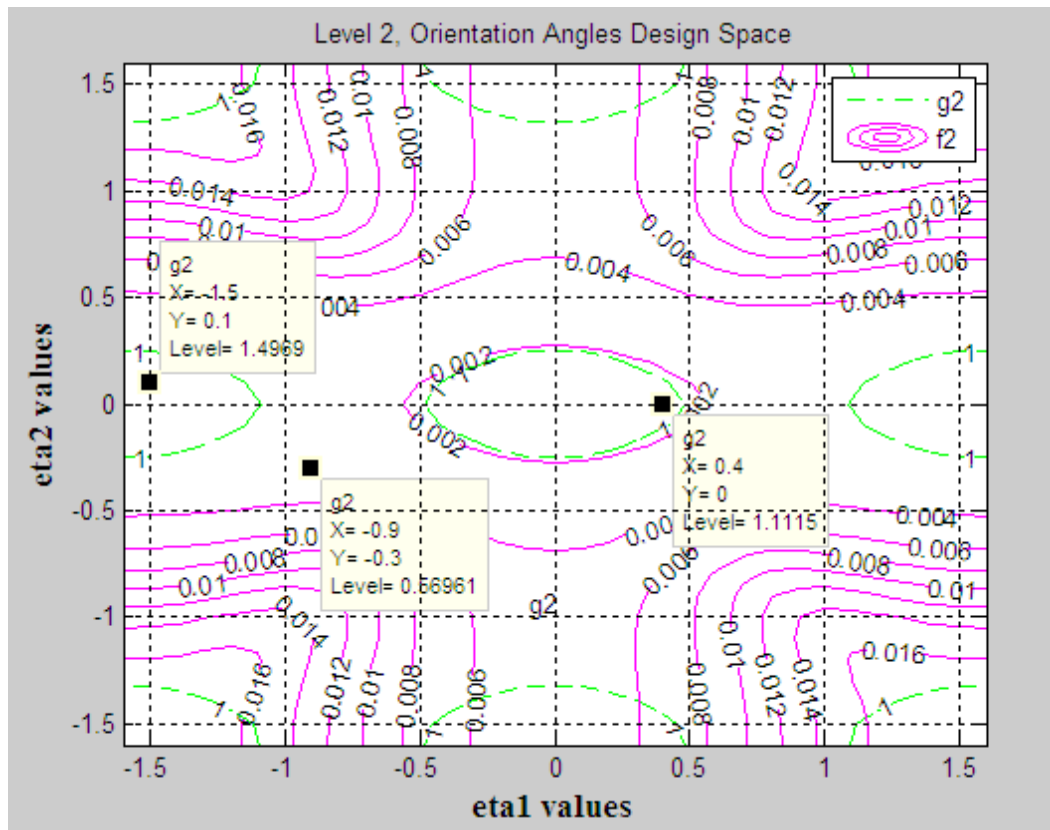


Figure 5-11 - Basic Example: “ g_2 ” Inequality Constraint Data Cursor “Levels” – L2

But trying to define the feasible region using only the data cursor window showed in Figure 5-11 it is not an easy operation since moving from left to right the g_2 Level changes from 1.4969 to 1.1115 passing through 0.56961. The same issue is expected moving from the top to the bottom of the graph and not just for g_2 but f_2 also.

That dedicated three-dimensional plot tool shows to be useful in situation where the 2D level-2 plot does not show any dash-dot-line as inequality constraint equations, g2, is proved below in Figure 5-12.

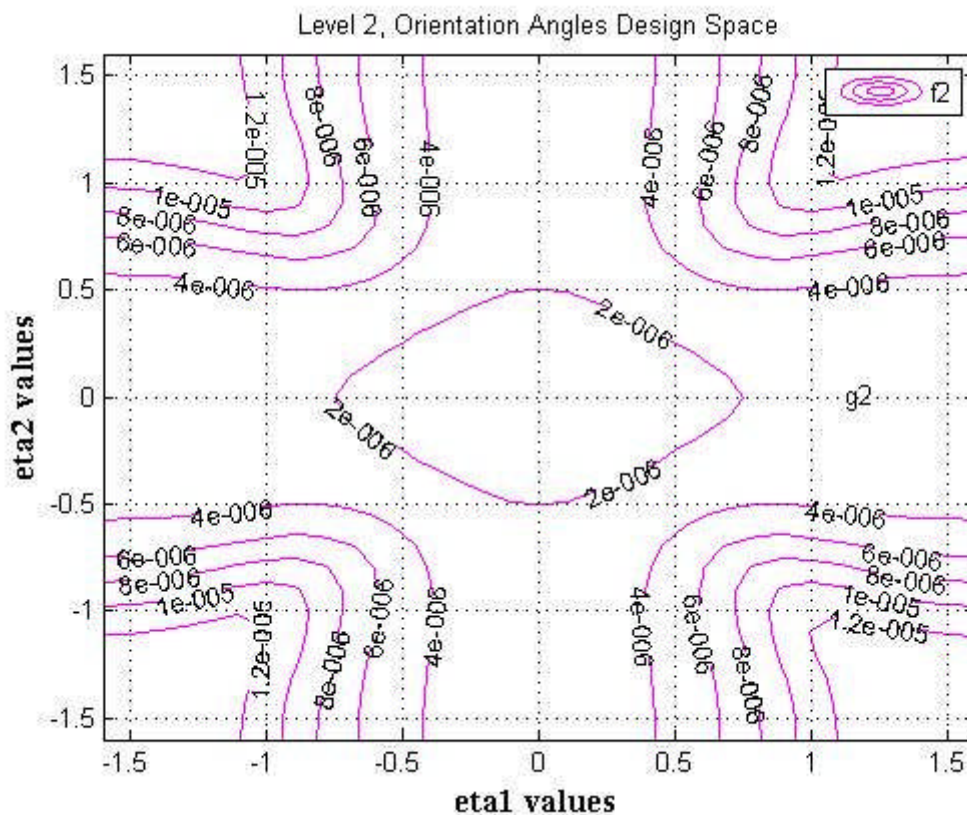


Figure 5-12 - Basic Example: Design Space Level 2 with no g2 lines

The above figure is the 2D outcome of the 3D objective function plot of Figure 5-13 intersected by several z planes.

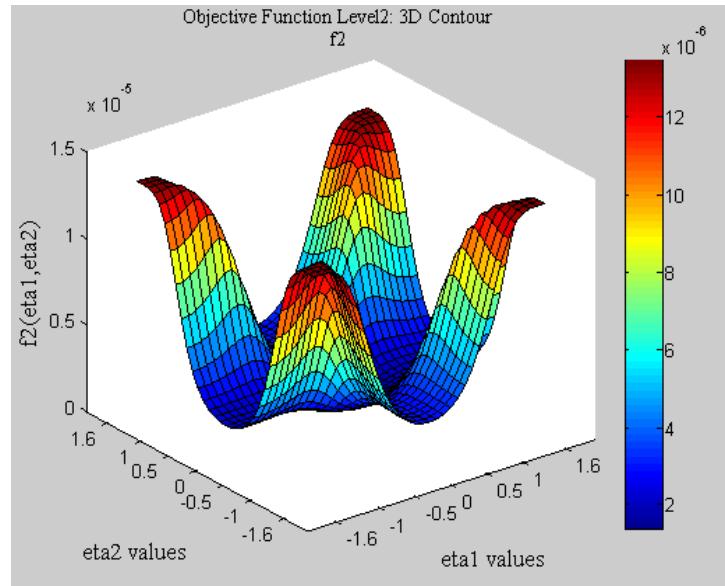


Figure 5-13 - Basic Example: 3D Objective Function Plot, Particular Case - Level 2

Figure 5-13 represents a particular case where the thicknesses have been chosen specifically very high, $t_{1,OD} = 200$ mm and $t_{2,OD} = 300$ mm, to cause none intersections between the 3D inequality constraint plot, Figure 5-14, with the plane $z=0.006$ micro-strains as further described by the following plots:

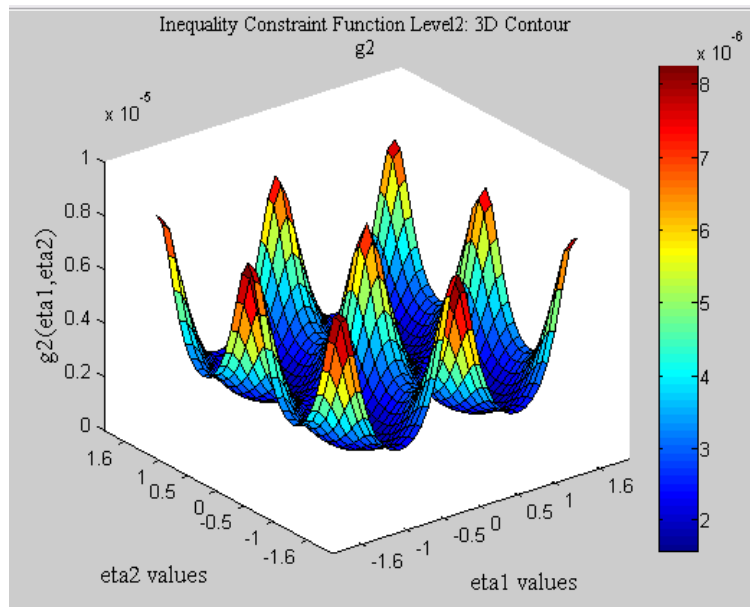


Figure 5-14 - Basic Example: 3D Constraint Function Plot, Particular Case - Level 2

In particular from Figure 5-15 is clear to figure out that when the vertical axis z is equal to 0.006 does not meet any point of the surface.

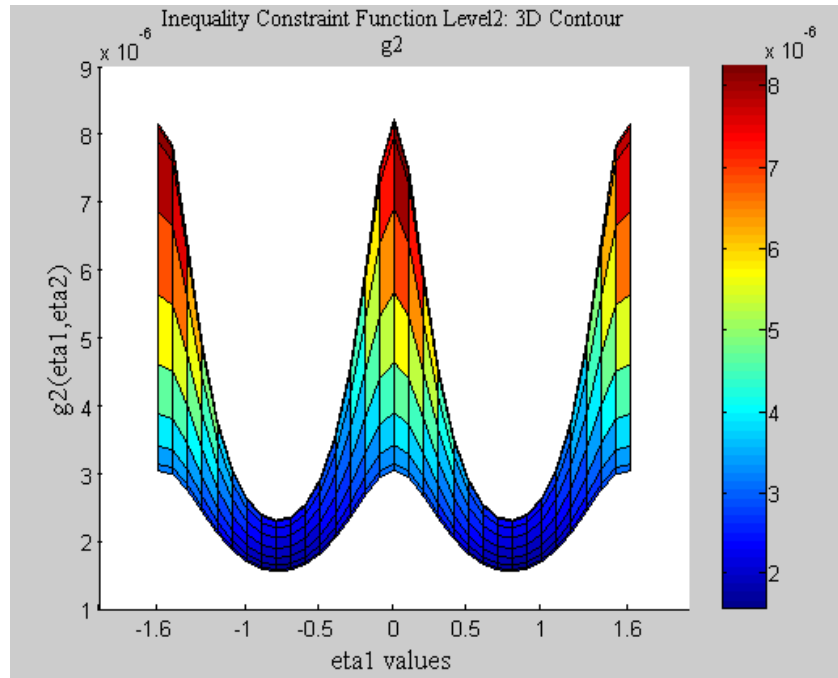


Figure 5-15- Basic Example: 3D Constraint Function Plot XZ View, Particular Case Level 2

Finally at this stage there is all the necessary information to refine the feasible domain. Adding the constraint lines the final orientation angles feasible domain necessary to obtain the optimum design at this level 2 is showed by Figure 5-16. The hashed side of this figure indicates the unfeasible domain within the orientation angles design space. The precise values of η_1 and η_2 will come out from the Figure 5-16 further analysis by positioning the cursor data window within the un-dashed region but as much as possible closer to the objective function solid line of 0.004 since it is meant to be maximised. So the below figure will be counted as an equation providing specific value of the angles, **Error! Not a valid bookmark self-reference.**

The two orientation angles coming out from (Eq.5-16) will be labelled as optimum design of level 2 as (Eq. 5-15) are optimum design of level 1 in the design space of the thicknesses.

(Eq. 5-16)

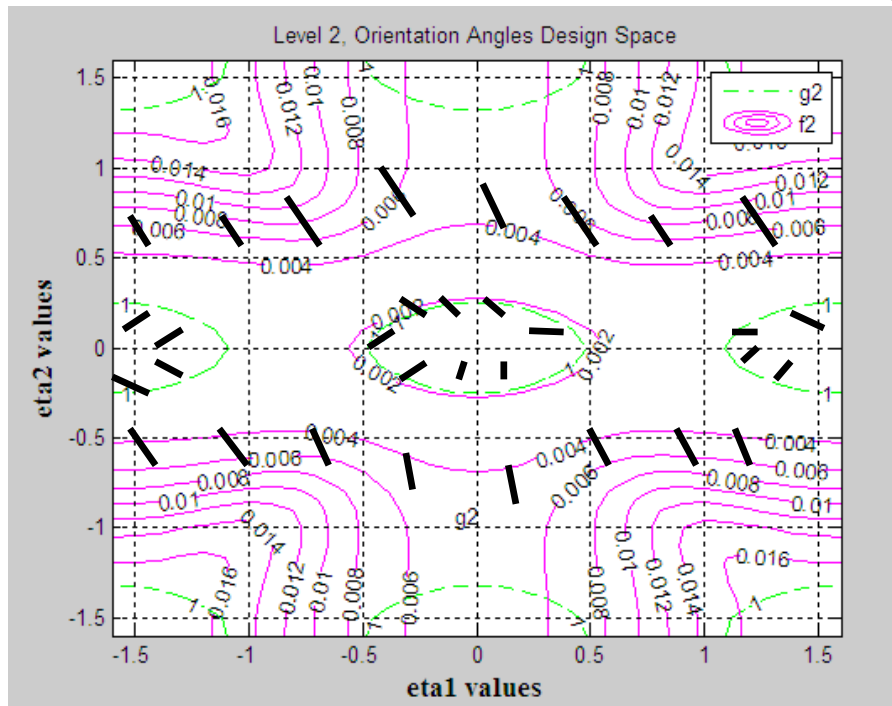
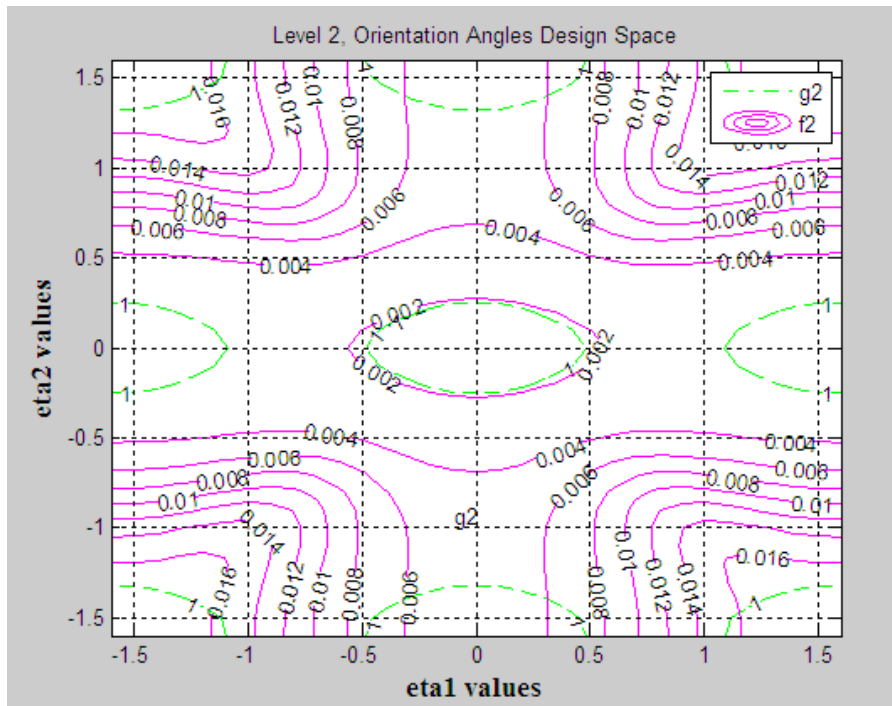


Figure 5-16 - Basic Example: Feasible Domain , Level 2

The above plot defines the feasible domain but cannot provide just a unique solution. For this reason according a correct optimisation terminology equations (Eq. 5-15) and Finally at this stage there is all the necessary information to refine the feasible domain. Adding the constraint lines the final orientation angles feasible domain necessary to obtain the optimum design at this level 2 is showed by Figure 5-16. The hashed side of this figure indicates the unfeasible domain within the orientation angles design space. The precise values of eta1 and eta2 will come out from the Figure 5-16 further analysis by positioning the cursor data window within the un-dashed region but as much as possible closer to the objective function solid line of 0.004 since it is meant to be maximised. So the below figure will be counted as an equation providing specific value of the angles, **Error! Not a valid bookmark self-reference.**

The two orientation angles coming out from (Eq.5-16) will be labelled as optimum design of level 2 as (Eq. 5-15) are optimum design of level 1 in the design space of the thicknesses.

(Eq. 5-16)



should not be

labelled as that but for commodity within the whole process that will be still used.

Finally at this stage there is all the necessary information to refine the feasible domain. Adding the constraint lines the final orientation angles feasible domain necessary to obtain the optimum design at this level 2 is showed by Figure 5-16. The hashed side of this figure indicates the unfeasible domain within the orientation angles design space. The precise values of eta1 and eta2 will come out from the Figure 5-16 further analysis by positioning the cursor data window within the un-dashed region but as much as possible closer to the objective function solid line of 0.004 since it is meant to be maximised. So the below figure will be counted as an equation providing specific value of the angles, (Eq.5-16).

The two orientation angles coming out from (Eq.5-16) will be labelled as optimum design of level 2 as (Eq. 5-15) are optimum design of level 1 in the design space of the thicknesses.

The level 1 and level 2 optimum designs at the end of this first design cycle will identify the laminate configuration, thicknesses and orientation angles, capable to withstand the load cases N_x, N_{xy} without exceeding the normal and shear strain Cons1, 0.004 micro

strains, and Cons2 of 0.006 micro-strains. For this reason the data will be passed on the level 1 to start the second design cycle, using the optimum design of level 2, first design cycle, as invariants. Then the process will move on as just described in the previous pages. The all design cycles of this specific numerical application will be described in the final section of this chapter.

5.1.2.3 Matlab script: BOX B

The main Matlab script, **BE_Main.m**, continues taking over the values of Box A to feed the Graphical Optimiser. In the Matlab editor this Box is anticipated by the comment “% Box B: Optimiser “(See Appendix A: “Basic Example Matlab Script and Related files”).

Input Starting Values that will feed the Objective and Constraint Functions to be plotted at the next Graphical Optimisation (Box B):

Level1 and Level2 invariants:

- $[C] = \begin{pmatrix} c_{11} & c_{12} & 0 \\ c_{12} & c_{22} & 0 \\ 0 & 0 & c_{66} \end{pmatrix}$: In-plane stiffness matrix coming from the single

layer matrix property $P = [E_{11} \ E_{22} \ G_{12} \ \nu_{12}]^T$ already imported in BOX A. The C matrix is a key quantity since according to (Eq. 3-30) links stress and strain values but mainly allows evaluating the transformed reduced stiffnesses, (Eq. 3-45), and then the in-plane stiffness matrix elements, (Eq. 3-49) and finally makes possible to plot the normal, (Eq. 5-6) and (Eq. 5-7), and shear strains (Eq. 5-13) and (Eq. 5-14).

Level1 Variables and Invariants:

- The Transform Matrix, $[T]_{(k)}$, at this level is employed through (Eq. 3-41). Since the orientation angles are invariant, $[T]_{(k)}$ is constant too and it is fed by orientation angles stored from the previous level 2 or as starting value (at first design cycle). $[T]_{(k)}$ matrixes are as many as the orientation angle groups.
- Using the thickness design variables the 2D plot of the objective function, $f1$, will be make possible through the following :

Matlab Command:

```
f1=obj_m1(X1,X2);
[C,han]=contour(x1,x2,f1,'m+');
set(han,'linewidth',1)
set(han,'LineStyle','-')
```

Where the first line command is the links to the function m-file mentioned in Table 5-2 .

- Again, using the thickness design variables previously described the 2D plot of the constraint function, $ineq1$, will be make possible through the following :

Matlab Command:

```
ineq1=inecon_m1(X1,X2);
[C1,han1]=contour(x1,x2,ineq1,[1,1],'g');
set(han1,'LineStyle','-')
set(han1,'linewidth',1)
```

Where the first line command is the links to the function m-file mentioned in Table 5-2 .

The previous two “Matlab Commands” gather all the information necessary to make up the 2D plot of Figure 5-7. The final equations to define the setting of the objective and constraint function lines are burned respectively in the function m-file **obj_m1.m** and **inecon_m1.m** already introduced by Table 5-2.

- That will come after the visual inspection of the plot of Figure 5-7 and through a specific request of the main Matlab script asking to type in the Matlab Command Window the optimum design and consequently save in the text file **thick_OD_1.txt**. The Matlab details of what just described follow:

Matlab Command:

%type in the optimum design (OD1)

```
opt1= input(['\type in your optimum design, t1, first layer next to  
the mid-plane\n'])  
t1=opt1  
opt2= input(['\type in your optimum design, t2, second layer from the  
mid-plane\n'])  
t2=opt2
```

%save the Optimum Design, OD1, of Level 1 (System Level)in a txt file

```
fid_Thickness = fopen('thick_OD_1.txt','wt');  
fprintf(fid_Thickness, '%5.3f\n', t2);  
fprintf(fid_Thickness, '%5.3f\n', t1);  
fprintf(fid_Thickness, '%5.3f\n', -t1);  
fprintf(fid_Thickness, '%5.3f', -t2);  
fclose(fid_Thickness);
```

The quantity to transport to the next level, Level2, will be the optimum design of Level 1 as thickness t_1, t_2 .

Level2 Variables and Invariants:

Once typed in the optimum design of level1 and after performing successfully the validation box (see coming section) the script is set to switch to a different feasible search domain this time within the orientation angles design space.

- Since at this level the invariants are the thicknesses, and the design variables are the orientation angles, the transform matrix, is now a variable and all its terms will be part of the final objective and constraint equations to be drawn in bi and three dimensional plots.

- The main Matlab script will ask whether or not plot objective and constraint functions using 3D graphic tools.

Matlab Command:

%Choice to use 3D Plots of f2 and g2, expressed in terms of orientation angles, before dealing with 2D Optimisation

```
choice_g2= input(['\if you want the 3D plots of "f2,g2" type in yes  
(1) if you do not (-1)\n'])
```

- Using the orientation angles, the 3D orientation plot of the constraint function, C2, and objective function, of2, will be make possible through the following:

Matlab Command:

```
C2 = cons_c2(G1,G2)
colormap(jet)
surf(g3,g2,C2)
colorbar
```

```
of2 = obj_f2(F1,F2)
colormap(jet)
surf(f1,f2,of2)
colorbar
```

The first lines, C2 = cons_c2(G1,G2)and of2 = obj_f2(F1,F2), are the links to the function m-file mentioned in Table 5-3. In fact the two Matlab files with all the main equations applied to obtain the objective and constraint function plots are **obj_f2.m** and **cons_c2.m**.

- Using the orientation angles, the 2D orientation plot of the constraint function, inecon_m2, and objective function, obj_m2, will be make possible through the following:

Matlab Command:

```
ineq2=inecon_m2(E1,E2);
[C1,han1]=contour(e1,e2,ineq2,[1,1],'g-');
set(han1,'LineStyle','-.')
set(han1,'linewidth',1)
```

```
f2=obj_m2(E1,E2);
[C1,han1]=contour(e1,e2,ineq2,[1,1],'g-');
set(han1,'LineStyle','-.')
set(han1,'linewidth',1)
```

The first lines, `ineq2=inecon_m2(E1,E2)` and `f2=obj_m2(E1,E2)`, are the links to the function m-file mentioned in Table 5-2. In fact the two Matlab files with all the main equations applied to obtain the Figure 5-10 are **obj_m2.m** and **inecon_m2.m**.

- As at level1 at this stage of the whole 2-level decomposition process after visual inspection of the optimum design the main Matlab script will allow to type in the two orientation angles chosen and then will save them all in the **angl1_c.dat**. The Matlab details of what just described follow:

Matlab Command:

```
%type in optimum design in radians, OD2

opt1= input(['\type in your optimum design in radians, eta1, first
layer next to the mid-plane\n'])
E1=opt1

E1_deg=convang(E1, 'rad', 'deg')

opt2= input(['\type in your optimum design in radians, eta2, second
layer from the mid-plane\n'])
E2=opt2
E2_deg=convang(E2, 'rad', 'deg')
```

It is quite relevant to notice that the Matlab command requires the angles coherently to the design variables ranges adopted in the plots and then convert them all in degrees since straight after will save these data in **angl1_c.dat** as showed below.

```
%save the Optimum Design, OD2, of Level 2 (Component Level) in a txt
file
fid_Orientation_Ang = fopen('angl1_c.dat','wt');

fprintf(fid_Orientation_Ang, '%5.3f\n', E2_deg);
fprintf(fid_Orientation_Ang, '%5.3f\n', E1_deg);
fprintf(fid_Orientation_Ang, '%5.3f\n', E1_deg);
fprintf(fid_Orientation_Ang, '%5.3f', E2_deg);

fclose(fid_Orientation_Ang)
```

The quantity to transport to the next level, Level1, will be the optimum design of Level 2 as thickness ϑ_1, ϑ_2 and a new design cycle will start again as long as either convergence box or validation boxes will flag up any reason to break up the iteration.

5.1.3 Numerical Model: CLT & FEM (C)

The Graphical Optimiser output becomes input of the Numerical Model stage labelled: Box C according to Figure 5-2. This box can present two different structures mainly related to the type of FEM chosen.

The former is presented by Figure 5-17 where the data coming out the Box B is going into the FEM and then goes to the Validation Box. So the “numerical model”, Box C, is just made by the FEM and nothing more. That was the first Version of the Basic Example since the Finite Element of the laminate configuration described in (Eq. 5-1), (Eq. 5-2), (Eq. 5-3), and Figure 5-3 was realised according to a ply-based modelling. So once identified the optimum design solution for thicknesses and orientation angles thorough visual inspection, those four data values were updated inside the input file of the solver.

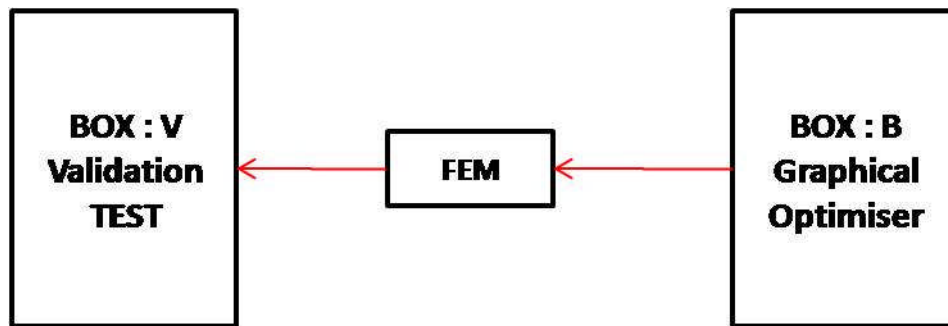


Figure 5-17 – Box C: First Version

The latter is described clearly by Figure 5-18 and it sounds soon more articulated than the previous. As anticipated earlier, this time the reason is still hidden within the Finite Element Model chosen. In fact, instead of considering all n orthotropic-layers of the laminate, it has been decided to idealise the structure as a single 2D orthotropic material. This approach requires to use the Nastran MAT8 card, see Table 5-4 and Figure 5-31, that will be populated by the effective elastic constants (Eq. 3-51), and

applying the homogeneity material assumption of Wisnom, [44][45]. This Wisnom's theory takes into account that the through-thickness elastic properties can be simply obtained through a transverse isotropy assumption. That concept is mathematically described by (Eq.2-1). The direct advantage of this structural idealisation is, no matter the number of layers the laminate is made of, the 2-level decomposition Matlab code will have to deal just with a single laminate thickness. So, every iteration the Nastran card to be reviewed will be just one as shown in Figure 5-30, simplifying the final code in particular when the number of ply is increasing.

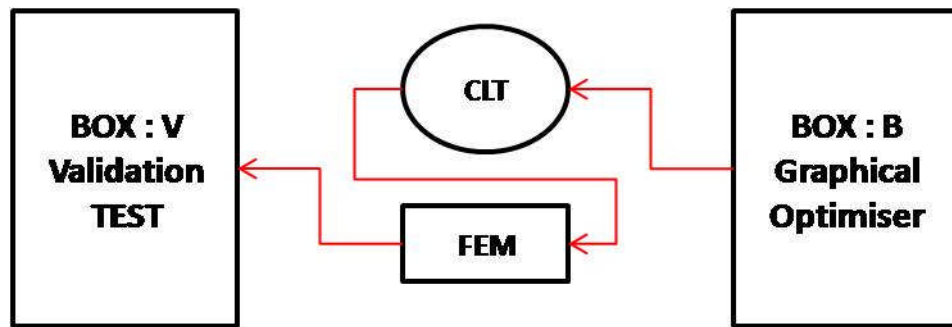


Figure 5-18 - Box C: CLT/FEM flowchart, Final Version

Despite the Matlab script has been set to embed both two structures of Figure 5-17 and Figure 5-18 the latter has been preferred since it is the best solution dealing later on with high number of layers, typical configuration of thick laminated materials, since no matter how many plies there are but the values to be updated in the input Nastran file will be always the same.

The main differences between the FEM structural idealisation employed by the Box C First Version, Figure 5-17, and by the Final Version of Figure 5-18 are broken down in the following Table 5-4, comparing the Nastran Solver Cards described better later on.

Box C Version	Structural Idealisation	Element Properties Nastran Card	Material Description Nastran Card
First Box C	Layers	PCOMP (all layers:	MAT8 single

		thick. and angle plies)	ply elastic constants
Final Box C	2D-orthotropic equivalent material	PSHELL (global thickness)	MAT8 laminate effective elastic constants

Table 5-4 - Different Structural Idealisation between First and Final Box C

The choice to idealise the laminate with a single thickness equivalent orthotropic material as anticipated earlier is due to the real life laminate aerospace fitting application thickness. In fact the ALCAS [2] side stay fitting showed in Figure 5-19 could reach up to few centimetres of thickness requiring hundreds of plies. Moreover considering that every layer to be defined needs thickness and orientation angle, as employed in the basic example, it is clear the increased efficiency adopting this advanced example structural idealisation.

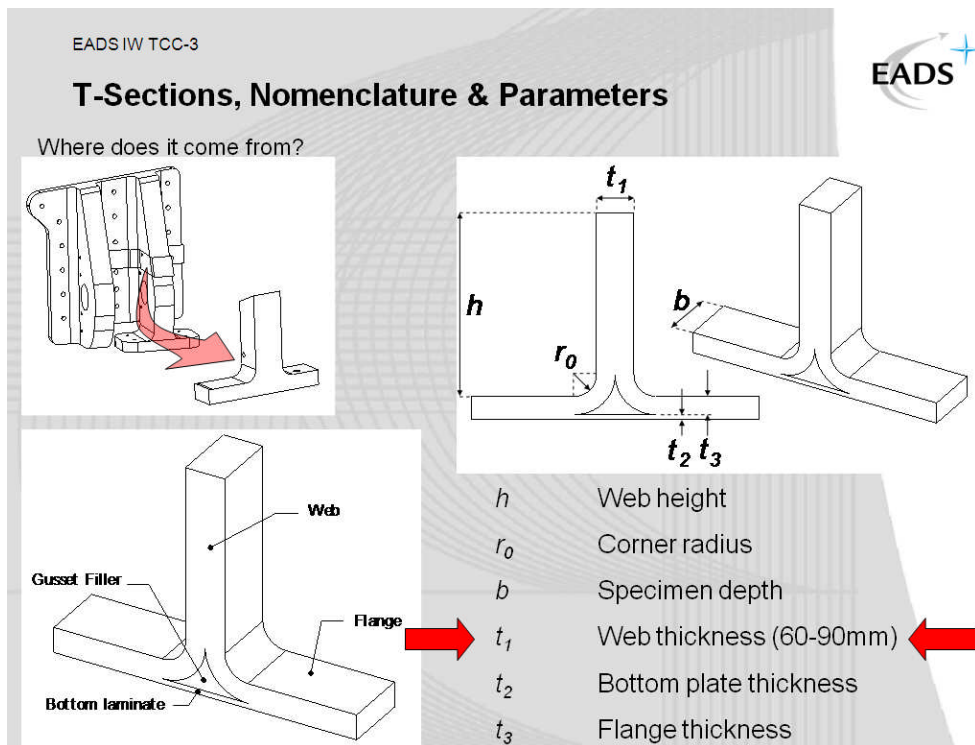


Figure 5-19 - “Side Stay Fitting” laminate thickness details required [3]

Once delivered the main differences and then the reasons of adopting the Box C Final Version now it is necessary to outline the data flow of Figure 5-18, the output of the Graphical Optimiser becomes input of the Classical Lamination Theory, CLT, script then the data goes to the finite element model, FEM, of the current laminate at that specific stage of the iteration.

The overall process of the last BOX C configuration is depicted by Figure 5-20 where the sub-tasks are highlighted.

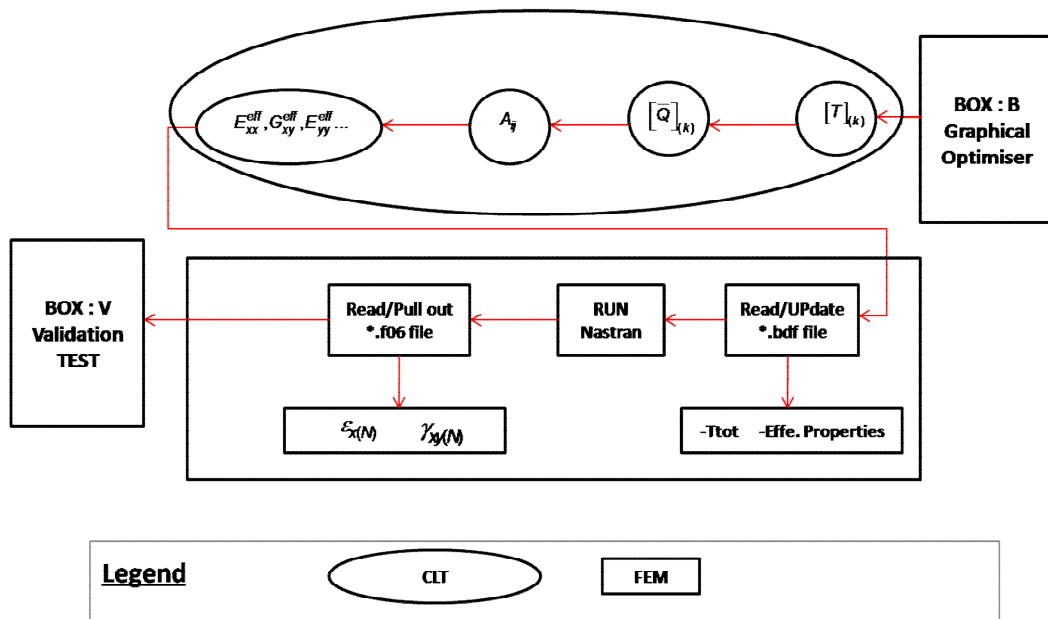


Figure 5-20 - Box C: CLT/FEM very detailed flowchart for both levels

5.1.3.1 Level1 and Level2

Classical Lamination Theory (CLT)

CLT is in the circle high level box of Figure 5-20. Part of the above in sequence quantities have already been mentioned as input data for the Graphical Optimiser but those where expressed in terms of the design variables and so still variants. Once the optimum design, output of Box B, either stack thicknesses of level1 or orientation angles of level2 has been chosen from visual graphical inspection of the design space, provides all the necessary information to apply the same main steps,

$[T]_k \rightarrow [\bar{Q}]_k \rightarrow (A_{ij})_k$, to get finally to evaluate the effective engineering elastic properties of the candidate laminate (Eq. 3-51).

Finite Element Model (FEM)

The FEM process showed in the square low level box of Figure 5-20 will start its routine making use of the “Homogeneity Assumption” [44],[45]. This assumption allows idealising the n-ply laminate as a material with equivalent homogeneous properties. The following are the *2D Orthotropic Material Characteristics* requested by the FEM solver Nastran to introduce an equivalent homogeneous material:

- E_{xx}^{eff} , E_{yy}^{eff} , G_{xy}^{eff} , ν_{xy}^{eff} consequence of CLT calculation (Eq. 3-51)
- G_{xz}^{eff} , G_{yz}^{eff} , ν_{yz}^{eff} evaluated through the Transverse Isotropy Assumption (Eq. 2-1)

The transverse isotropy assumptions of section 2.7, for the laminate configuration in (Eq. 5-1), (Eq. 5-2), and (Eq. 5-3) is applicable since dealing with unidirectional carbon fibre- epoxy material, $\nu_{23}^{eff} = 0.45$, and mainly with in-plane loading cases. In fact the structural FEM idealisation focuses on a cantilever beam loaded, according to the level analysed, by N_x at level 1 and N_{xy} at level 2 as showed by the below Figure 5-21 and Figure 5-22.

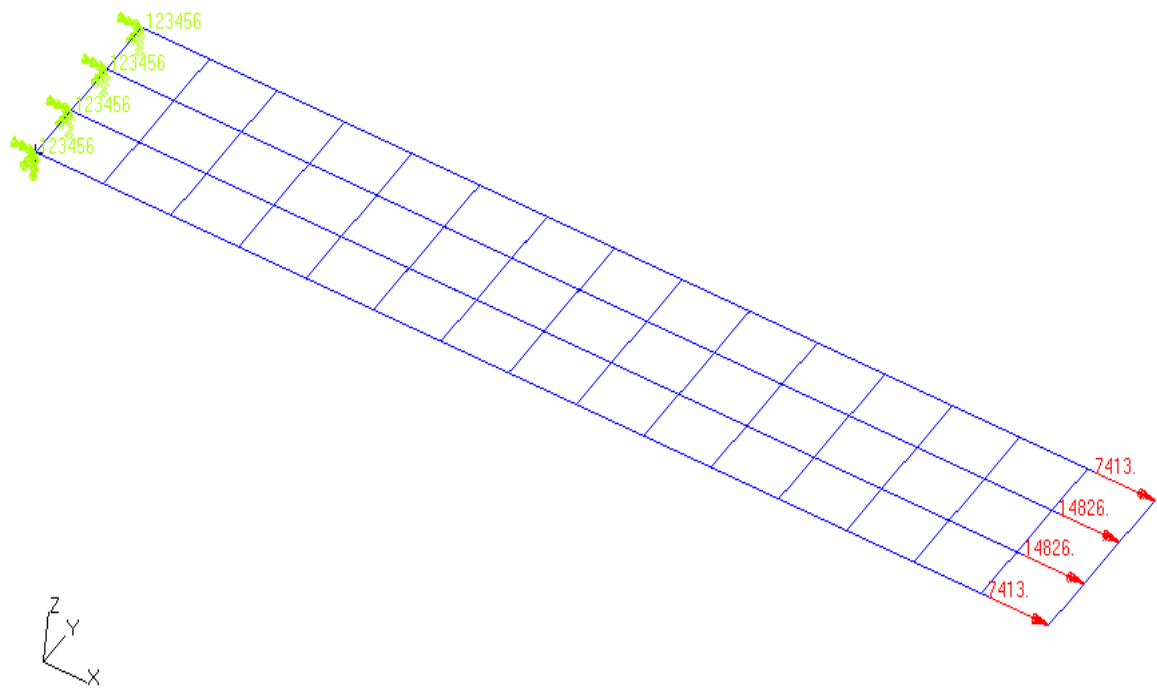


Figure 5-21 – Level 1 FEM Loading Case: Cantilever Beam N_x

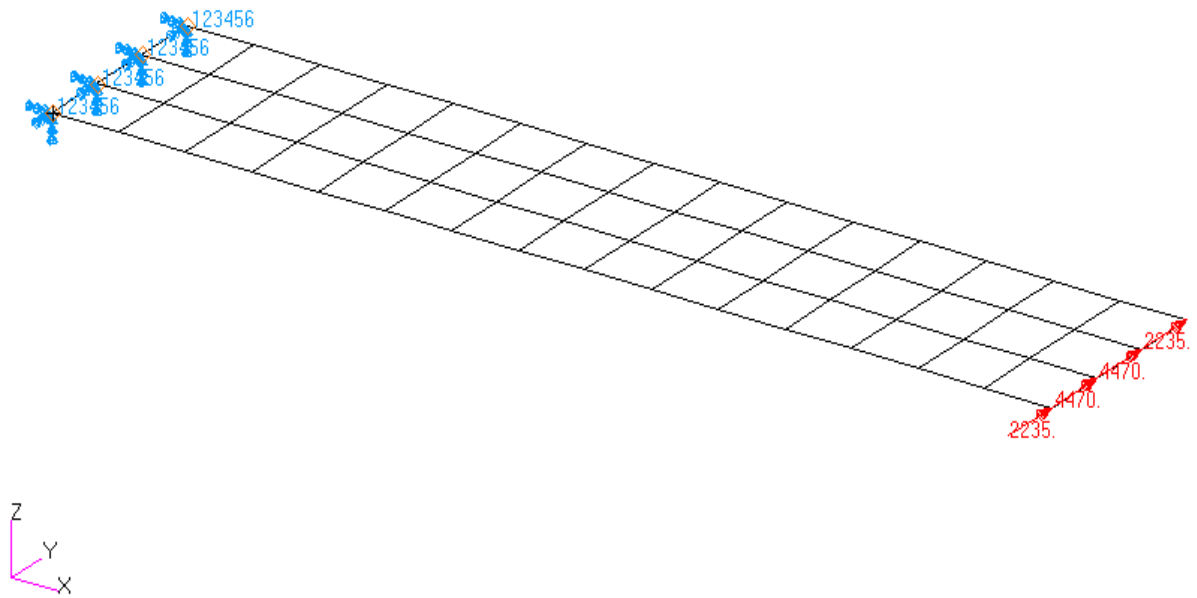


Figure 5-22 - Level 2 FEM Loading Case: Cantilever Beam N_{xy}

The common scheme of the above figures is a cantilever beam fixed at one edge, all six degrees of freedom of the edge nodes, and free at the opposite side where point loads are applied.

The main dimensions of the FE laminate employed coherently to the reference axes of Figure 5-21 and Figure 5-22 are as follows:

- Width, $W= 254$ mm (y dimension)
- Length, $L=1270$ mm (x dimension)
- Thickness, $H=$ variable quantity (z dimension)

The width magnitude justifies the intention to mesh the beam with a reasonable number of elements, three, in the y direction such that the load applied might be spread on four nodes along the width and eventually reduce and control the local load effects that it will be discussed and shown later.

The x length of the cantilever beam has been defined to control the localised structural behaviour induced by poisson-effect, next to the constrained area, and load-effect of the tip edge. In fact in Figure 5-23 and Figure 5-24 it is possible to control these unwanted effects (for the two different load cases N_x and N_{xy}) and then to identify an area where the strain is smooth and uniform as expected by the CLT approach. Inside this even area, it is recommended to select the ID element which will be compared later on with a Classical Lamination Theory output within the Validation Box.

JSC.Patran 2005 14-Dec-10 19:13:04

Fringe: MAO_NX.SC1, A1:Static Subcase, Strain Tensor, , X Component (NON-LAYERED)

Deform: MAO_NX.SC1, A1:Static Subcase, Displacements, Translational,

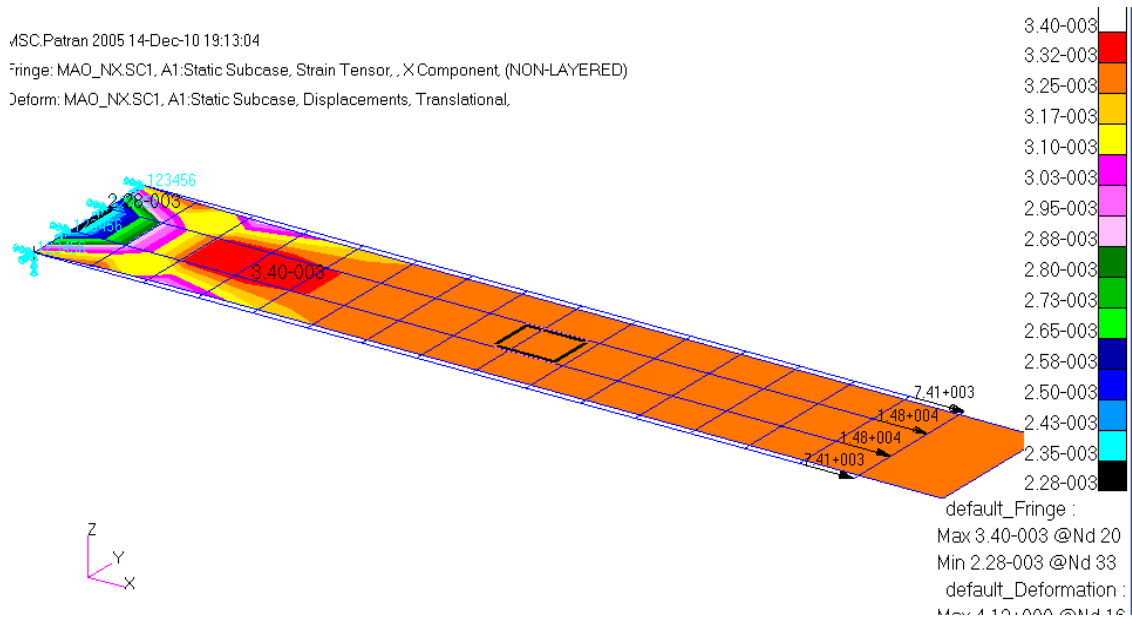


Figure 5-23 – Level 1, N_x Load Case: Normal Strain Contours and Translational Displacement

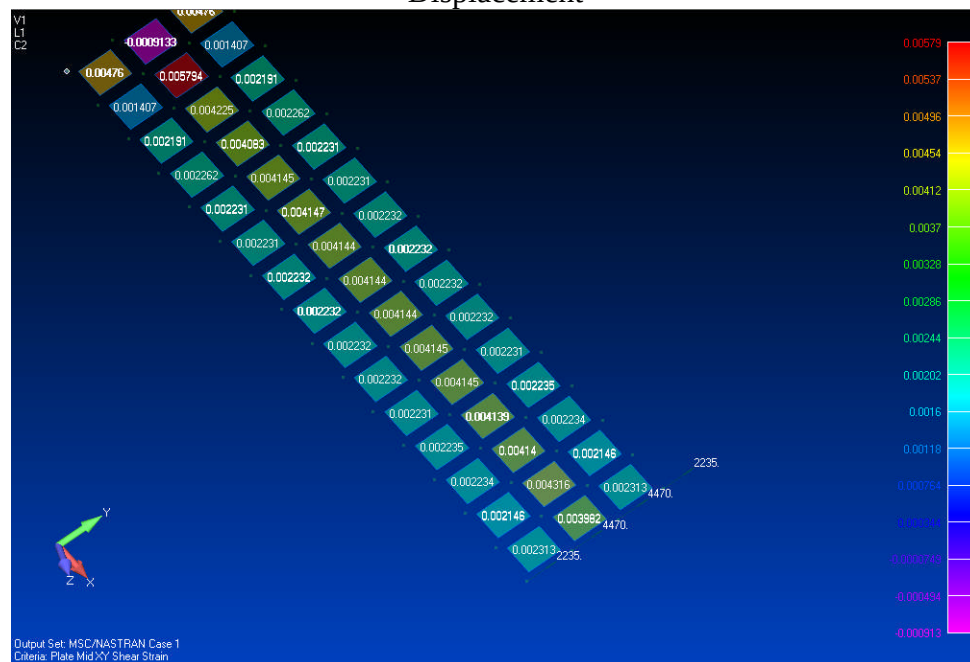


Figure 5-24 – Level 2, N_{xy} Load Case: Shear Strain Contours and Translational Displacement

Following these considerations the elements chosen to be used as comparison data with the classical lamination they are highlighted in the coming pictures:

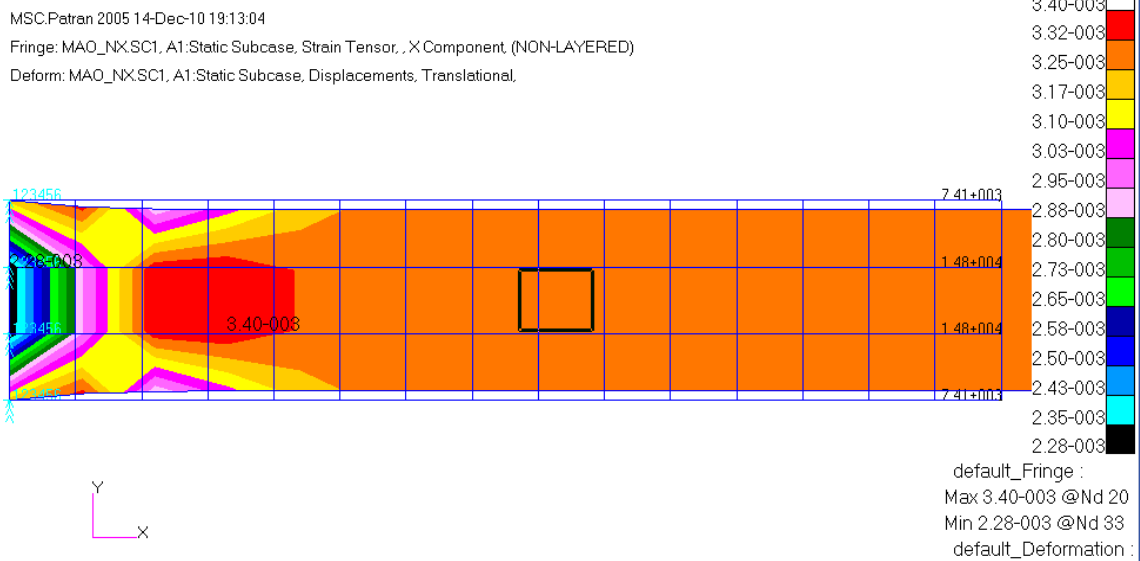


Figure 5-25 – Level 1, N_x Load Case: Element 23 (mid x length)

Figure 5-25 identifies the ID element 23, highlighted by a lighter frame, as the one capable to provide reliable data since in a very uniform strain contour region.



Figure 5-26 – Level 2, N_{xy} Load Case: Element 25 highlighted by a white circle

Figure 5-26 reveals the uniform shear strain distribution area where the element has been singled out. This element location lays on a uniform longitudinal strip characterised by the same shear strain values counting out tip and constrained element

areas. The ID element chose as representative of this strip is element 30 (see light blue circle in Figure 5-26).

The reliability of the magnitude of the above normal and shear element strains, has been tested against the Classical Lamination Theory and compared later on during the run of the Basic Example numerical application. On this regard is pivotal to emphasize that the shear strain plotted by Figure 5-24 and Figure 5-26 are engineering strain and to be compared to CLT need to be divided by a factor of two to get true shear strain [1]. In fact, Nastran, employed as solver, to perform the FE analysis chooses as its own set up to output the Engineering strain value instead the True ones [11].

The sole laminate dimension not counted in the structural idealisation as invariant is the thickness. In fact the thickness quantity in the above FE models has not linked anymore to the number of layers as been employed in the basic example first version of Figure 5-17. Rather it has been associated with a unique equivalent global thickness, Z dimension, by employing the concepts of homogenous properties already mentioned in section 3.6.3. Note that strain contours shown so far are related to the first iteration introduced in section 5.1.2 and identified with (Eq. 5-15) all the main values for the two separate load cases are summarised in the following table:

Load Case	Total thickn. [mm]	Eta1 [degrees]	Eta2 [degrees]
Level 1 : N_x	0.8	45	-45.840
Level 2: N_{xy}	0.8	22	22.918

Table 5-5 – Laminate Configurations at Level 1 and Level 2, 1st iteration

For thoroughly interpret the above table it is necessary to remember the orientation angles are not directly used in the FEM, since the ply based modelling has not been used, but are employed to develop the effective engineering properties of (Eq. 3-51) and (Eq. 2-1). Total thickness and effective engineering properties will be updated iteration after iteration within the input Nastran file as described later analysing the “bdf”, Nastran input, and “f06”, output, files.

For a complete description of the FEM figures shown so far it is needed to explain why the loads of Figure 5-21 and Figure 5-22 apparently do not match with (Eq. 5-2) and (Eq. 5-3).

The in-plane load cases described in (Eq. 5-2) and (Eq. 5-3) are expressed as a load over a width dimensionally speaking. So to apply those load cases to the cantilever beam under investigation it is required to multiply (Eq. 5-2) and (Eq. 5-3) by the “W” quantity introduced few pages earlier.

That will lead to a proper dimensional load. Then the coming considerations will be necessary to show the approach followed to apply the loads uniformly taking into account the number of nodes used and their locations.

In fact, each of the two edge nodes, whose label numbers are 16 and 64, as shown in Figure 5-27, satisfies (Eq. 5-17) since they share the applied vector load with one element. On the other hand, each of the two mid-nodes, label numbers 32 and 48, as described in Figure 5-27, comply with (Eq. 5-18) since they share the vector load to two different elements. These two mentioned equations are applicable to loads directed either as N_x or N_{xy} , and their resultant will be given by (Eq. 5-19).

These explanations have been applied to obtain the numbers of the Nastran cards of Figure 5-28 and Figure 5-29.

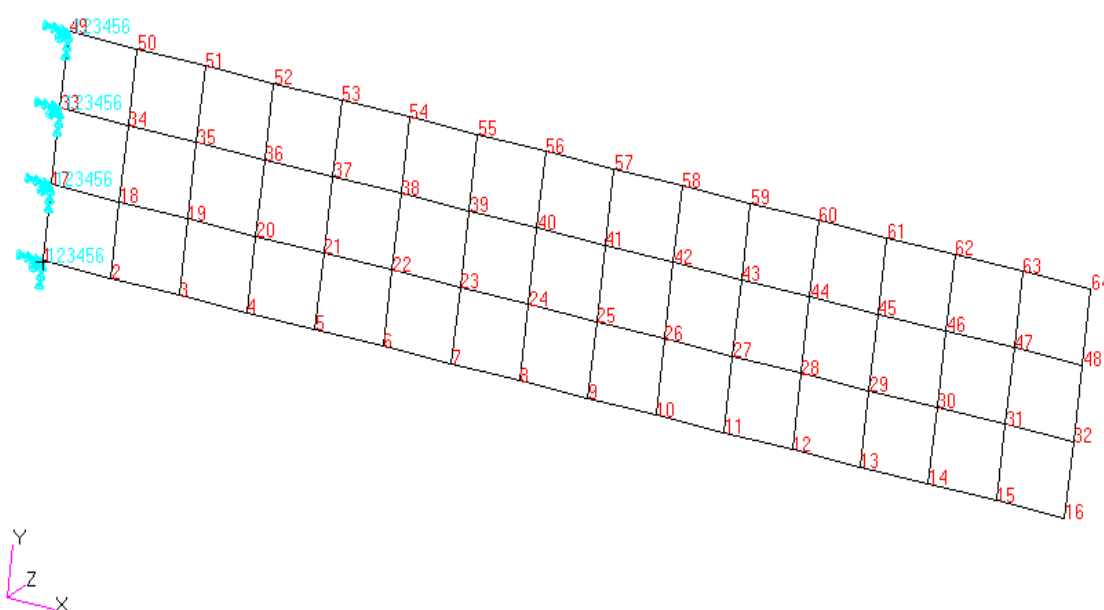


Figure 5-27 Edge Application Node Labels

As mentioned before:

W = beam width dimension = 254 mm

N° = number of nodes applied the loads = 4

(Eq. 5-17)

$$N_{x_EdgeNodeNastran} = \frac{N_x W}{6} = \left[\frac{\text{Newton}}{\text{Node}} \right]$$

(Eq. 5-18)

$$N_{x_MidNodeNastran} = 2 \left(\frac{N_x W}{6} \right) = \left[\frac{\text{Newton}}{\text{Node}} \right]$$

(Eq. 5-19)

$$R = (N_{x_EdgeNodeNastran})_{16} + (N_{x_EdgeNodeNastran})_{64} + (N_{x_MidNodeNastran})_{48} + (N_{x_MidNodeNastran})_{32} = N_x W$$

In the following figures it is shown the Nastran Force Cards for N_x (Level 1) and N_{xy} (Level 2) input files. The first column from the left within the rectangle gives the list of the ID load application nodes whilst the other rectangle highlights the magnitude of the loads.

\$ Nodal Forces	of Load	Set :	force2			
FORCE	2	32	0	14826.	1.	0.
FORCE	2	48	0	14826.	1.	0.
\$ Nodal Forces	of Load	Set :	force3			
FORCE	4	16	0	7413.	1.	0.
FORCE	4	64	0	7413.	1.	0.

Figure 5-28 – N_x Load case: Loads and ID nodes, Level 1

```

      1.      6
$ Displacement Constraints of Load Set : spc1.1
SPC1      1      123456  1      17      33      49
$ Nodal Forces of Load Set : force 2
FORCE     2      16      0      2235.      0.      1.      0.
FORCE     2      64      0      2235.      0.      1.      0.
$ Nodal Forces of Load Set : force 3
FORCE     5      32      0      4470.      0.      1.      0.
FORCE     5      48      0      4470.      0.      1.      0.
$ Nodal Forces of Load Set : force 3

```

Figure 5-29 – Nxy Load case: Loads and ID nodes, Level 2

Now it is possible to appreciate Figure 5-28 and Figure 5-29 loads are the same of Figure 5-21 and Figure 5-22.

Following the main sub-tasks of Figure 5-20 the FEM, static analysis, embedded in the Multilevel Decomposition Frame, either Basic or Intermediate Example, is made of:

- Nastran Input file “*.bdf”
- Nastran Run
- Nastran Output file “*.f06”

Nastran Input file “*.bdf”

The input file is a text file with a file name and a .bdf , bulk data file, extension. It can be created with a text editor (WordPad), or a finite element pre-processor (Patran). The MSC.Nastran input file contains a complete description of the finite element model including:

- The type of the analysis to be performed
- The model’s geometry
- The model’s material properties
- A collection of finite elements
- Loads
- Constraints (boundary conditions)
- Requests for the type of output quantities to be calculated

Since the “*.bdf” is a text file, Matlab can have access into it and change the requested variables, whilst maintaining the Nastran format.

As highlighted by the flow-data diagram, the only variables to be changed in are:

- Tot: total thickness of the equivalent homogeneous material
- Effective Engineering Properties

The below image is an extract of the bdf file, where the Nastran PSHELL card is highlighted to show the position quantity to be reviewed at each iteration.

```
PARAM      PRTMAXIM YES
$ Elements and Element Properties for region : Homogeneous_lam
PSHELL     1      1      0.800000 1      1
```

Figure 5-30 – PSHELL Nastran Card: laminate total thickness

In particular is highlighted inside the rectangle the total laminate thickness at the first iteration (Eq. 5-15).

On the other hand the effective engineering properties, evaluated by CLT, are described below by the Nastran MAT8 card.

```
$ Description of Material : Date: 03-May-09      Time: 16:09
MAT8       1      6.7e+0041.8e+0048.8e-0012.3e+0042.3e+0046.3e+003
$ Nodes of the Entire Model
```

Figure 5-31 – MAT8 Nastran Card: Effective Engineering Properties

The Figure 5-31 rectangle underlines the quantities to update at each level, with the following meaning from left to right

E_{xx}^{eff}	E_{yy}^{eff}	ν_{xy}^{eff}	G_{xy}^{eff}	G_{xz}^{eff}	G_{yz}^{eff}
----------------	----------------	------------------	----------------	----------------	----------------

Table 5-6 – Laminate Effective Material Properties

All the details of the “Basic Example bdf Nastran File” are gathered within Appendix A.

Nastran Run

To execute MSC.Nastran, the user will be asked by the Matlab Script to type in the name of the bdf file in the Matlab Command Window. Iteration by iteration, the name of the Nastran input file will not change since it will be always upgraded, overwriting the previous version.

The main requirement for the success of this task is that the following files:

- Nastran executable file : nastran.exe
- The Matlab Script
- The bdf file

are stored in the same folder.

Nastran Output file “*.f06”

As the bdf Nastran input file, the “*.f06” output is a text file and will be accessible by Matlab in the same way. The output data of Nastran are gathered in this file according the requests stored in the bdf file. Since the quantities to be monitored are the normal and shear strain, the Element Strains are the Output Request among all the other possible choices shown in Figure 5-32:

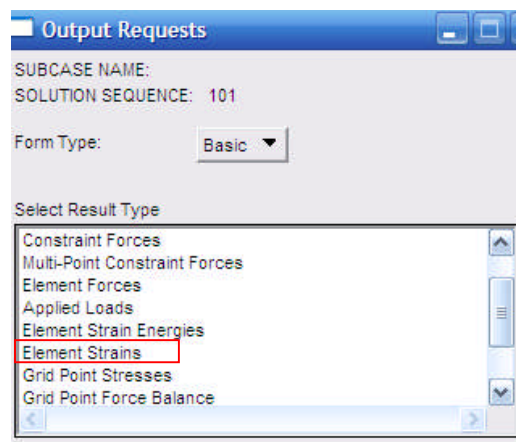


Figure 5-32 - Nastran Output possible Requests

After each iteration, the results are read by the Matlab Routine Script such that $\varepsilon_{x(N)}$ and $\gamma_{xy(N)}$ are saved by the program and then passed on the Validation Test Box. The two strain measures and their locations in the f06 file are shown in the two following figures respectively to level 1, normal strain of ID element 23, and level 2, shear strain of ID element 25.

ELEMENT		STRAIN		STRAINS IN ELEMENT COORD		
SYSTEM	ID	PRINCIPAL GRID-ID	STRAINS (ZERO CURVATURE	SHEAR) NORMAL-X	NORMAL-Y	
SHEAR-XY		ANGLE	MAJOR	MINOR	VON MISES	
0	23	CEN/4	0.0	3.264679E-03	-2.869284E-03	
1.225416E-11		0.0000	3.264679E-03	-2.869284E-03	3.543897E-03	
			-1.000000E+00	0.0	0.0	0.0
0.0	0.0		0.0	0.0		

Figure 5-33 – Level 1, Output Nastran File: ID Element 23 normal strain

STRAINS IN QUADRILATERAL ELEMENT						
ELEMENT	GRID-ID	STRAIN	STRAINS IN ELEMENT COORD		COORD-SYSTEM	PRINC:
ID		CURVATURE	NORMAL-X	NORMAL-Y	SHEAR-XY	ANGLE
25	CEN/4	0.0	1.283122E-08	-1.128328E-08	4.144614E-03	44.9998
		-1.000000E+00	0.0	0.0	0.0	0.0

Figure 5-34 – Level 2, Output Nastran File: ID Element 25 shear strain

The above normal and shear strains are numerical data associated to the Table 5-5 initial, first iteration, laminate configuration subject respectively to the load cases N_x and N_{xy} . In fact these values are the same of the data shows by the contours of Figure 5-23 and Figure 5-24.

5.1.3.2 Matlab script: BOX C

The data flows down to the Validation Box from the Graphical Optimiser passing through the numerical model block; the Box C final version as anticipated by Figure 5-18. In fact the optimiser optimum design will supply the CLT and then the FEM. In

the Matlab editor this Box is announced by the comment “% Box C: CLT and FEM” (See the Appendices 5: “Basic Example Matlab Script”).

Level1 and Level2 invariants:

- To perform the Classical Lamination Theory analysis both levels need to take in their routine the following step by step matrix process:

$$[T]_k \rightarrow [\bar{Q}]_k \rightarrow (A_{ij})_k$$
 basically identified by the following.
- Applying the above steps the effective engineering properties, later on updated in the input Nastran file, as follows:

Matlab Command:

```
%Effective Engineering Elastic Constants
E11=(1/Ttot)*(Atotal(1,1)*Atotal(2,2)-Atotal(1,2)^2)*(1/Atotal(2,2))
E22=(1/Ttot)*(Atotal(1,1)*Atotal(2,2)-Atotal(1,2)^2)*(1/Atotal(1,1))
V12=Atotal(1,2)/Atotal(2,2)
G12=Atotal(3,3)/Ttot
%Wisnom Assumptions applicable to Carbon, Graphite and Glass Fibers for
%Equivalent Homogenous Properties idealization
G13=G12
V23=0.45
G23=E22/(2*(1+V23))
Eff=[E11 E22 V12 G12 G13 G23]
```

- Main Matlab script, **BE_Main.m**, asks for the input of the bdf file and then a routine updates the values already highlighted in Figure 5-30 and Figure 5-31.

Matlab Command:

```
%Update *.bdf file with new Effective Properties and Total Thickness
%To add as many as those below Effective Elastic Properties I used
Equivalent Homogeneous Properties

filename1= input(['\nenter the file listing the data readings
(.bdf)\n'], 's');
```

- It is requested to type in the output Nastran file, f06, and then a routine pulls out the strain data respectively of the ID element of level 1 and level 2 as described by Figure 5-33 and Figure 5-34.

Matlab Command:

```
%Pull out the normal strain from *.f06 output Nastran file

filename2= input(['\nenter the file listing the data readings
(.f06)\n'], 's');
fid2=fopen(filename2, 'r')
```

The main differences of the two routines just described are that the former opens, reads bdf file then searches the quantities to change through the Matlab command “fprintf(fid1, '%3.1e', E11)” and finally saves the file. On the other hand the latter opens, reads f06 file then searches for a single quantity to pull out the value without changing the file through the different Matlab command “tline=fgets(fid2)”.

Level1 Variables and Invariants:

- In between the Matlab routines dealing with input and output Nastran files it is needed to run the software Nastran to make it possible to be created the updated output f06 file. At this level 1 the run will expect to use the input bdf based on the N_x load case of Figure 5-21.

Matlab Command:

```
%Run Nastran using the *.bdf as input file and then I ll have a *.f06
as output

! C:\MSC.Software\MSC.Nastran\bin\nastranw.exe
```

```
C:\MSC.Software\MSC.Nastran\bin\Thesis\Basic_Example_Package_June\NX_m  
ao_Lminus.bdf
```

The above command is broken down in two separate terms. The first identify the location of the Nastran executable and with a specific syntax runs it. The second identifies the specific setting of the input Nastran file associated with the level considered.

- Specifically to this level the aim to read the f06 file is meant to point out the normal strain of the ID element 23, Figure 5-33, and stores it in a specific value will be used then in the box, V.

Matlab Command:

```
Z = textscan(tline, '%s %s %s %s %s',1)  
eps_X = str2num(Z{5}{1})
```

Level2 Variables and Invariants:

- The equivalent task of the previous level 1, to run Nastran software, is expected to be performed at level 2. This time the bdf Nastran input file to be run will be based on a different load case, N_{xy} , as Figure 5-22 says.

Matlab Command:

```
%Run Nastran using the *.bdf as input file and then I ll have a *.f06  
as output  
! C:\MSC.Software\MSC.Nastran\bin\nastranw.exe  
  
C:\MSC.Software\MSC.Nastran\bin\Thesis\Basic_Example_Package_June\Nxy_  
miomao_Lminus.bdf
```

This time the bdf file name is: “Nxy_miomao_Lminus.bdf”.

- Opening and reading the f06 file reveals the shear strain of the ID element 25, Figure 5-29, and stores it in a specific value will feed the next box, Validation Test (V).

Matlab Command:

```
Z = textscan(tline, '%s %s %s %s %s %s %s',1)
eps_XY = str2num(Z{7}{1})
```

The following two quantities are to be transferred to the validation test box:

- eps_X
- eps_XY

5.1.4 Validation Test: Constraint Checks (V)

This is the only task, either at level1 or level2 that has two separate inputs coming from two different boxes. In Fig.6.18, is very clear the double input data flow coming from:

- BOX C: CLT/FEM
- BOX A: Analytical Model

The output also is double; it means it has to possible different directions depending on a successful result of the internal “Constraint Condition”.

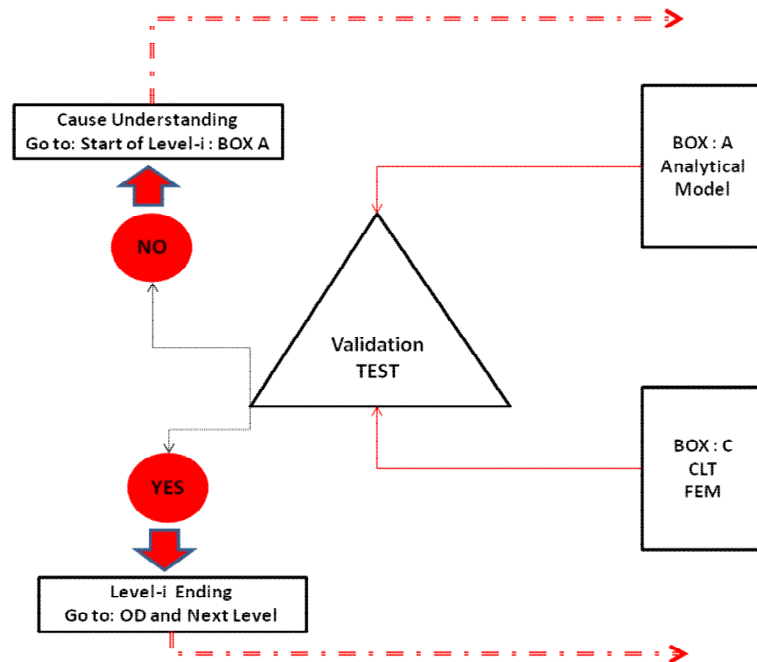


Figure 5-35 - Validation Test Flow Diagram

The “Constraint Condition” part of the Validation Box (V), is explained inside the square dot line of Figure 5-36 that shows the normal or shear strain extracted from the f06 file with the same quantity coming from the analytical model, identified as an inequality constraint condition than can be compared.

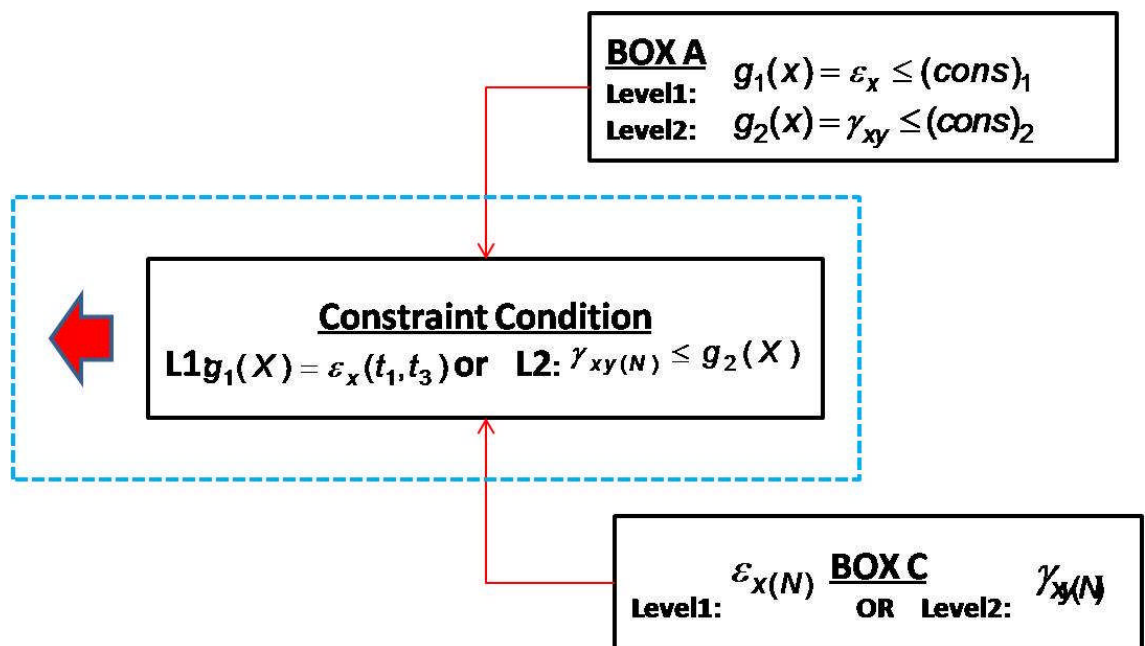


Figure 5-36 - Validation Test: Constraint Condition

In the eventuality the above “Constraint Condition” is not satisfied, the program stops and displays a Warning Note, highlighting the need to:

- Understand the causes
- Restart the HMD from the Analytical Model

Differently to the Convergence Test, this non-compliance of the Validation Test will not allow to take into account any results of the Multilevel Decomposition Process, since the values returned are unreliable.

Moreover not having a similarity from FEM and Box A inequality constraints, may suggest a difference between Classical Lamination Theory and Finite Element Model. In fact, the Optimum Design coming out the Optimiser has been already chosen to satisfy the constraint function, $g(X)$, and the same quantities have been used in the Nastran input file. So, if the f06 gives strain higher than the starting constraint quantities, the cause could be attributed to any of the following:

1. Homogeneous Property Idealisation
2. Effective Engineering Properties
3. Format Typing Issues compiling the bdf file or reading the f06
4. FE model local effect

The general understanding of the above four points applied to the Basic Example say that:

First and Fourth cause types are already discounted, since the current cantilever beam is not part of the “Homogeneity Assumption” recommendations anticipated in section 2.7. In fact the elements chosen, to read the strain, are placed as emphasised by Figure 5-25 and Figure 5-26 far away from undesired local effect.

On the other hand, the second potential motive is likely could be the effective engineering poisson ratio evaluated through CLT could have a value too high (greater than 0.6). Such high and nearly unrealistic value in composite laminated material real

life applications says a different material idealisation, rather than the 2D-orthotropic approach is necessary.

Finally regarding to the third factor it is useful to stress that Matlab syntax to type into a Nastran card maintaining the previous requested format has been tested in many different digit configurations but the full range of permutations, mainly with very high or low number, is still a possible source of error.

5.1.4.1 LEVEL 1

Since the constraint function at this level is the normal strain, $g_1(X) = \varepsilon_x(t_1, t_2)$, the “*Constraint Condition*” will be:

(Eq. 5-20)

- $\varepsilon_{x(N)} \leq g_1(X)$

Then the positive outcome of the Test will give the green light to pass the Optimum Design to the next Level, L2, to begin a new optimisation cycle (see Figure 5-2).

5.1.4.2 LEVEL 2

At this level, the normal strain, $f_2(X) = \varepsilon_x(\vartheta_1, \vartheta_2)$, is the objective function and the constraint is the shear strain, $g_2(X) = \gamma_{xy}(\vartheta_1, \vartheta_2)$, so the “*Constraint Condition*” will be:

(Eq. 5-21)

- $\gamma_{xy(N)} \leq g_2(X)$

Since level2 is the last step of multilevel decomposition, the successive task is the Convergence Test. In fact, once the above inequality condition is satisfied, the optimum design of this level is passed to the Convergence Box.

5.1.4.3 Matlab script: BOX V

The Nastran normal/shear strain extracted from the f06 file, output of Box C, is going to the Validation Box as reported, within the Matlab editor, by the comment “% Box V: Validation Test “(See the Appendices 5: “Basic Example Matlab Script”).

Level1 and Level2 invariants:

It is not applicable to this Validation Box

Level1 Variables and Invariants:

- The normal strain data coming from the reading of the numerical f06 model file, is becoming the core part of the Matlab “if condition” that controls the program whether continue or stop and give a warning message.

Matlab Command:

```
% Box V: Validation Test

if eps_X> Cons1
    warning('your "Optimum Design1" is not satisfying anymore the
Constraint Conditions Cons1')
    break
end
```

Once the Validation Condition is successful the Optimum Design of level1, (OD1), is passed to the next level, L2, ready to start another design cycle as pictured by Figure 5-1 and Figure 5-2.

Level2 Variables and Invariants:

- In very similar way to Level 1, at this level the shear strain from the f06 file, the engineering value divided by two by a Matlab routine to convert to true strain, is passed to validation test as follows:

Matlab Command:

```
% Box V: Validation Test

if eps_XY> Cons2
    warning('your "Optimum Design2" is not satisfying anymore the  
Constraint Conditions Cons2')
    break
end
```

At this level, L2, the success of the Validation Test enables the Optimum Design data, (OD2), to go to the Convergence Test before moving on to another iteration step.

5.1.5 Convergence Test (CT)

Without the Convergence Test, the two-level decomposition will keep running without any confidence whether diverging or converging towards the final optimum design. So, its correct position is at the very end of each iteration cycle to determine whether the combined optimum design of each level is still within the feasible overall space design. This box, as shown in Figure 5-37, is the only one of the whole framework to be unique, regardless of the two different levels considered.

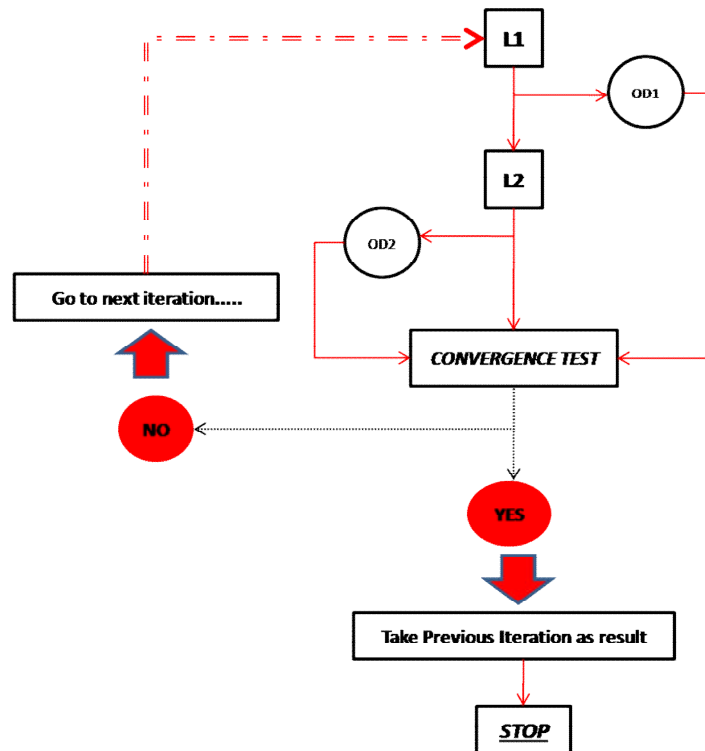


Figure 5-37 - Convergence Test Flow Diagram

The above Figure 5-37 highlights that the Optimum Design of both levels participates to define this Box condition. Once the internal test is performed, two alternative flow data direction are drawn in the diagram. The former goes to the next iteration; the latter stops the MD program and saves the final result as the previous iteration data. The Figure 5-38 below explains in detail the type of condition involved in such a process.

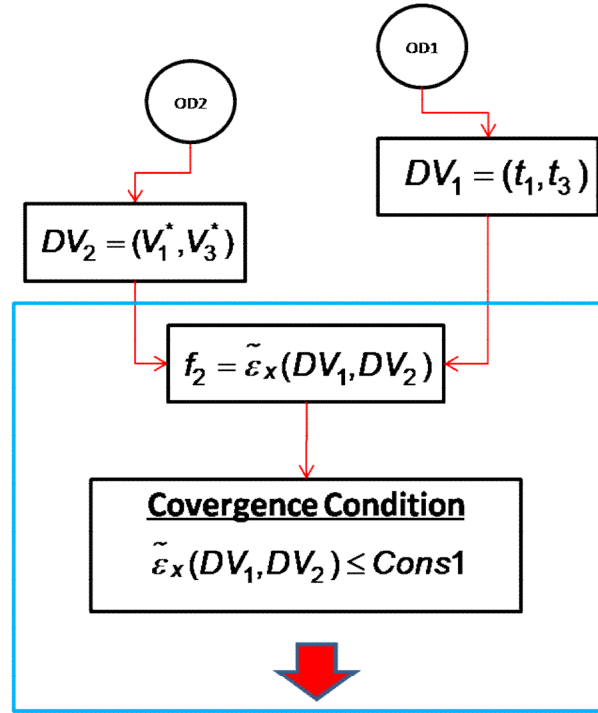


Figure 5-38 - Convergence Test Condition

The optimum design variables of both levels, OD1 and OD2, coming after Visual Inspection of the Graphical Optimiser, are employed to evaluate the normal strain in those specific values labelled as $\tilde{\varepsilon}_x(DV_1, DV_2)$. If this value $\tilde{\varepsilon}_x$ is still smaller than the starting constraint quantity, Cons1 of level1, the convergence has not reached yet and the program will set itself to begin another iteration loop. On the other hand, when the following condition is satisfied, $\tilde{\varepsilon}_x(DV_1, DV_2) > Cons1$, the program terminates after storing as the best optimum design, the data of the previous iteration.

5.1.5.1 Matlab Script: BOX CT

The first script line inside the Matlab Editor of the program highlights, as already done many times before, the start of this box by the comment line : “% Convergence Test : BOX CT” .

As final condition to decide whether to enable the next iteration, this is an unique condition and depending on the two levels simultaneously.

- Matlab defines the normal strain quantity, constraint function at level1 and objective function at level 2, evaluated through the design variables that came out of the respective graphical optimiser, and so stored that value in a Matlab variable function `f2_it(k)`.
- `f2_it(k)` is expressed in terms of a `k` Matlab variable that allows to create the iteration based on a sequence of “if elseif for” loops.
- the if elseif loop is structured as follows:

The very beginning of the Matlab Script, just before the comment line of BOX A there is:

Matlab Command:

```
Cons1=0.004
f2_it(1)=0
for k=2:1000
f2_it(k)= f2_it(k-1)+f2_it(1)
```

Straight after follows the first “if” condition

```
if f2_it(k)< Cons1
```

Finally, the “elseif” last loop condition is at the very end of the program just about the end of the validation test, BOX V, at level 2 and beginning of BOX CT

```
% Convergence Test : BOX CT
elseif f2_it(k)> Cons1

warning('Reached Convergence. Take Optimum Designs, OD1 and OD2, of
the Previous Iteration\n')
    Result_T=OD(k-1).L2
    Result_eta=OD(k-1).eta
    break
%these two "end" are linked to the if and for at the very top of the
program
end
end
```

A thorough explanation of the above Matlab Script Lines requires to say that when the normal strain, “`f2_it(k)`” satisfies the `Cons1` constraint condition (Constraint function of level 1), the “for” loop starts another `k` iteration and the Multilevel Decomposition will start again at level 1. On the other hand, once the condition “`f2_it(k)> Cons1`” is

true a warning is displayed in the Matlab Comand Window. The Program then stops and the candidate laminate to be stored as final optimum design is the one of the previous iteration as “**Result_T**” or “**Result_eta**” according to the type of laminate chosen in the Graphical Optimiser of level 2.

The convergence test has been chosen to be performed on “**f2_it(k)**” objective function of the level 2, since this function is meant to be maximised but simultaneously at level 1 it is expected to be constrained to fulfil inequality conditions. This peculiarity makes the normal strain as the main coupling quantity inside the multilevel decomposition but it needs to be controlled to keep satisfying the inequality condition of level 1.

5.1.6 Optimum Design (OD)

The Optimum Design is the final product of the whole Hierarchical Multilevel Decomposition optimisation process. It can be identified at the end of each level and once the validation test is successful.

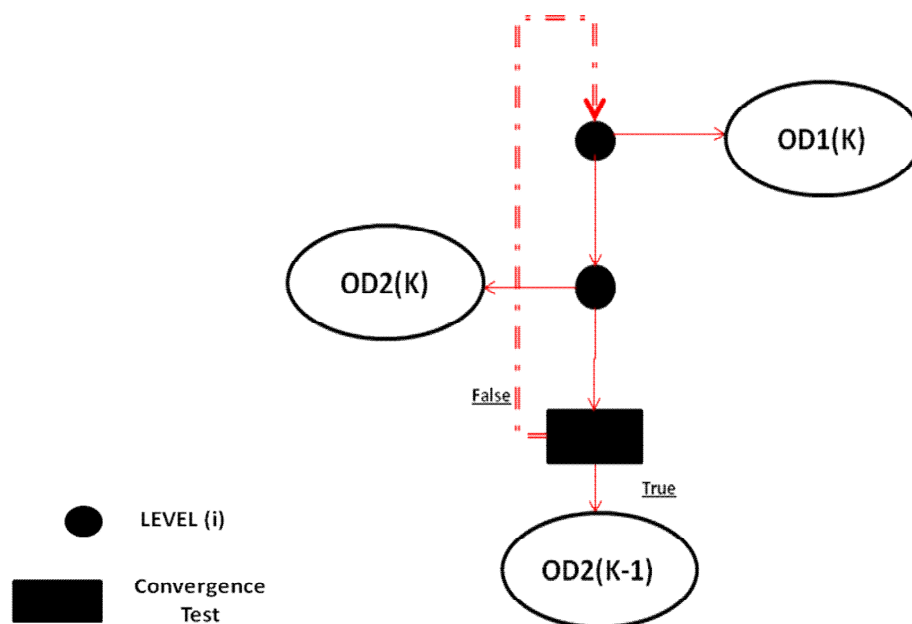


Figure 5-39 - Optimum Design flow diagrams

The above Figure 5-39 highlights the output of each level, dark spots, as the Optimum Design and once the Validation Test Condition is *true*, Dark Square, the program finishes with the optimum design of the previous iteration ($k-1$).

5.1.6.1 Matlab Script: BOX OD

There is no need to describe this box within the Matlab script lines since has been already mentioned as pivotal part of the output of the Graphical Optimiser , BOX C, and then as input data of the Analytical Model, BOX A.

5.1.6.2 Matlab Code Structure

Before moving on to the numerical application, it is worth summarising the previous sections providing a dedicated flow chart for each level of the multi-level decomposition Matlab code. In particular, the main Matlab file (running file of the whole procedure) lines will be linked with:

- External re-called Matlab/Text/Nastran files
- User input data requests
- Main Routines and their relevant actions
- Routine inter-dependance

Figure 5-40 describes the layout structure of the following level 1 and level 2. Then, Figure 5-41 and Figure 5-42, highlight each level flowchart in more details.

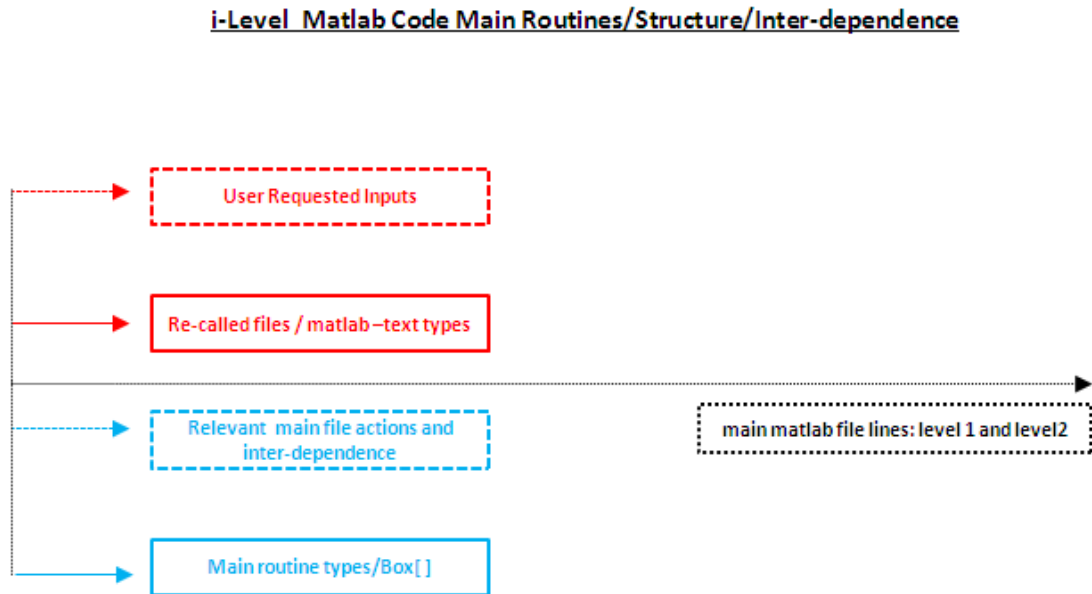


Figure 5-40 i- Level Matlab Code Flow Chart

The lower part of Figure 5-40 shows the main Matlab routine functions with their relevant characteristic and inter-dependence using the same terminology introduced in the previous sub-sections. On the other hand, on the higher side, the re-called files are listed and the points of the main Matlab code where the user is expected to insert the design data are shown.

The aim of the coming flowcharts is to enable the user not just to fully understand the structure of the whole code, to take control of the main routines and to run the program independently but to eventually have the possibility to improve the code itself.

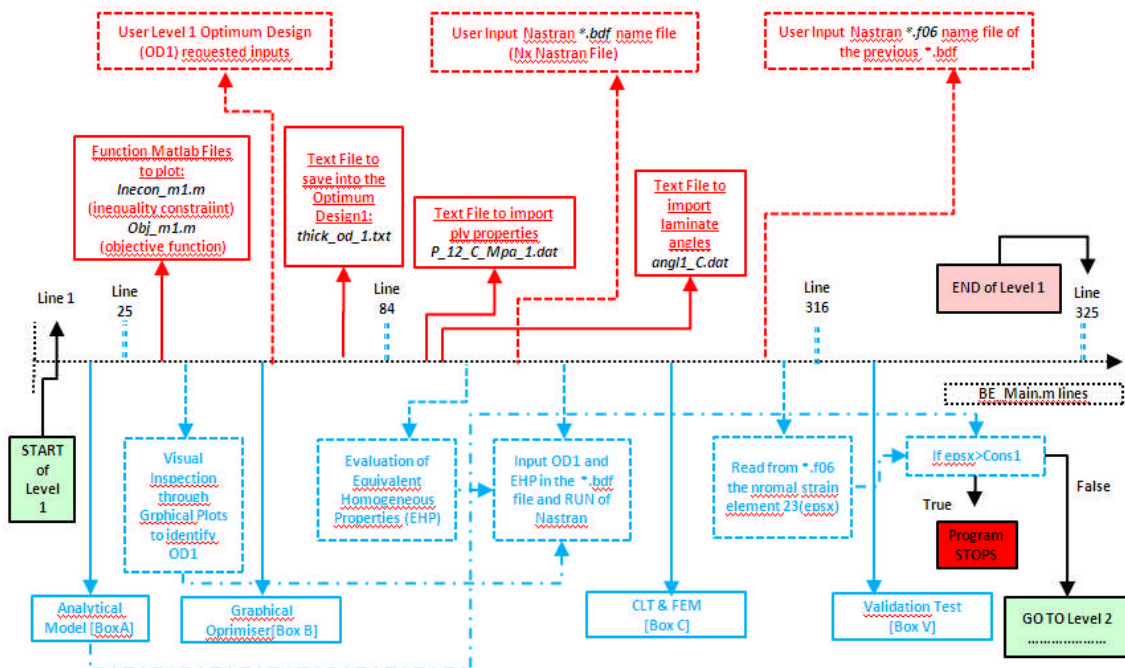


Figure 5-41 “BE_Main.m” Main Matlab File Flow Chart: Level 1

Figure 5-41 and Figure 5-42 reveal the same structure, as already anticipated in the previous sections and summarised in Figure 5-2. The main differences are the different design space, ply thicknesses in level1 and orientation angles with level2, different constraint functions, normal strain and shear strain, and then different Nastran input files. From level 1 to move on level 2 it is expected to be satisfied the Validation Test condition.

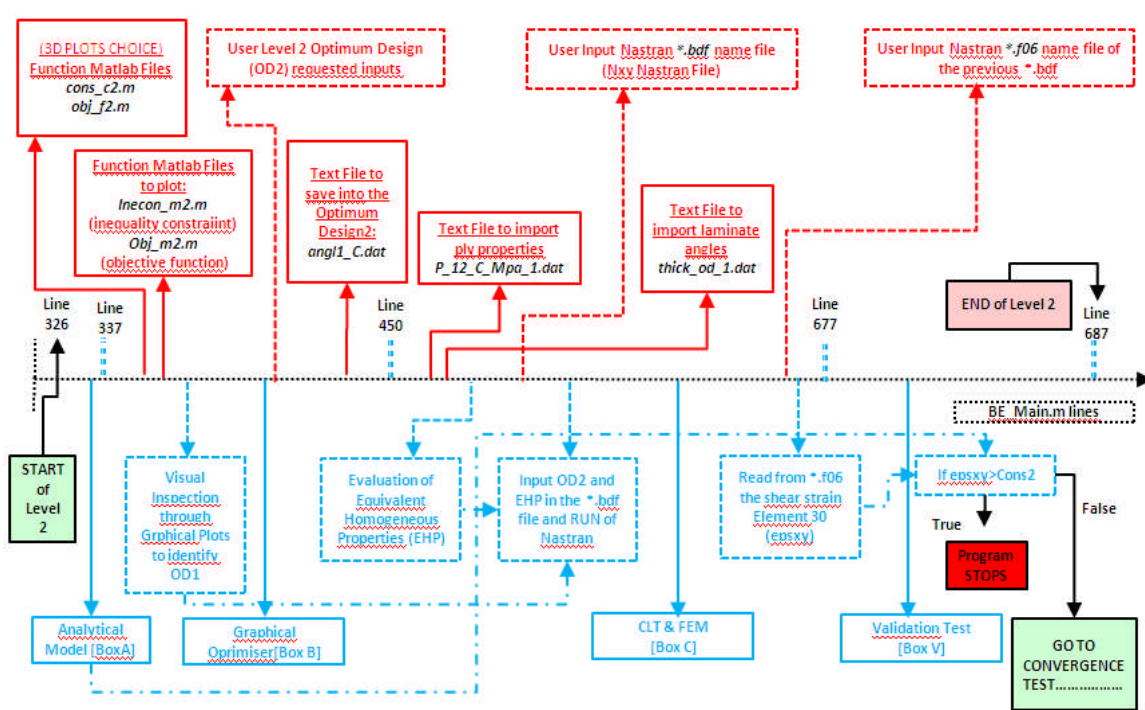
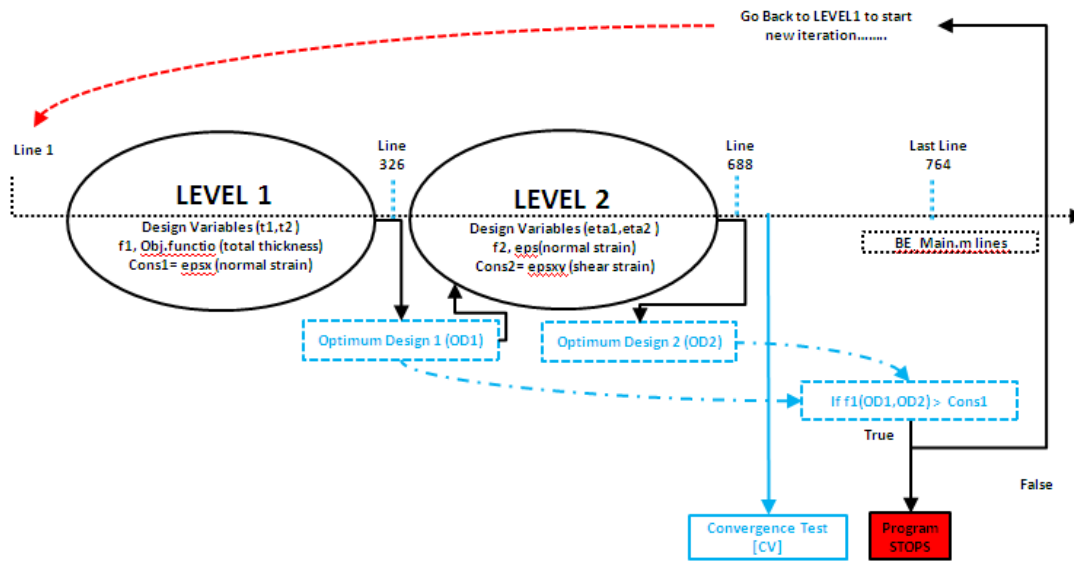


Figure 5-42 “BE_Main.m” Main Matlab File Flow Chart: Level 2

The link between level-2 and level-1, the start of another iteration, is possible through the “Convergence Test” placed at the end of the level-2 as shown in Figure 5-42. In fact, once successful this test the iteration happens and the level-1 starts again to refine further the design space and get closer to select the best Optimum Design passing through level2 again as long as validation test and convergence test permit according to the summarised final flowchart of Figure 5-43.

Basic Example Level 1 & Level 2 & Convegence Test Inter-Dependence Flow Chart**Figure 5-43 “BE_Main.m” Main Matlab File Flow Chart: Level 1, Level 2 and Convergence Test**

To obtain more details or to see the “to do list” for running a Basic Example Numerical application refer to Appendix A.

5.2 Numerical Application Results and Discussion

The intention of this section, as anticipated at the end of 5.1.2.2, is to gather the design cycle history of the numerical application introduced and described diffusely through the previous pages of this chapter, see 5.1.2.

The laminate description is the same anticipated A by:

- (Eq. 5-1): Laminate Material
- (Eq. 5-2): Level 1 Load Case, N_x
- (Eq. 5-3): Level 2 Load Case, N_{xy}
- Figure 5-3: Laminate ply configuration

The next pages will be representative of the Design Optimisation History of a specific numerical application, **Run1**. Each page will point out in a single figure with two 2D-plots showing the design variables at both levels, L1 and L2, for that specific iteration. The data highlighted in the table represent the invariants at that specific level coherently to what mentioned in section 5.1.1.3.

At the end of each Run the summary of the overall optimisation are presented through determined plots and indexes.

RUN1 - Iteration1

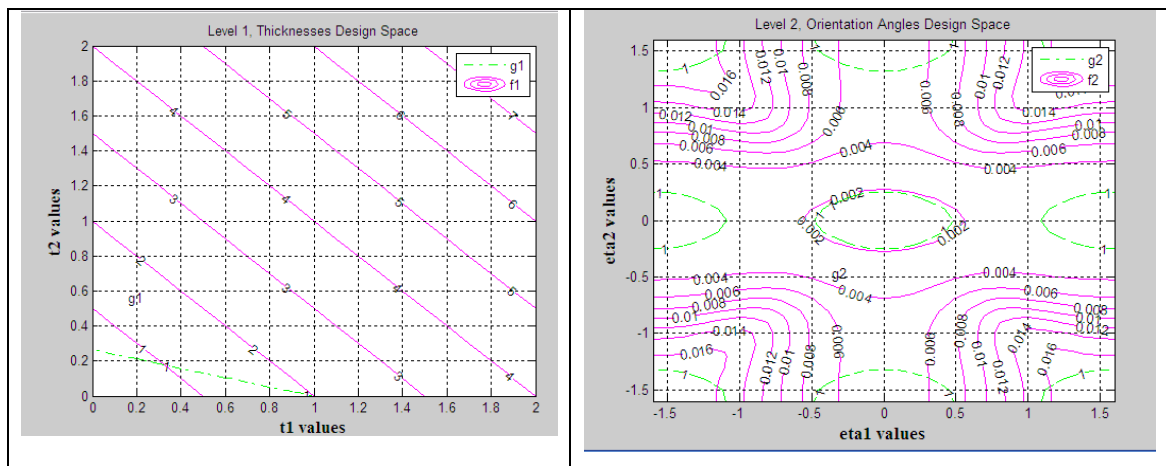
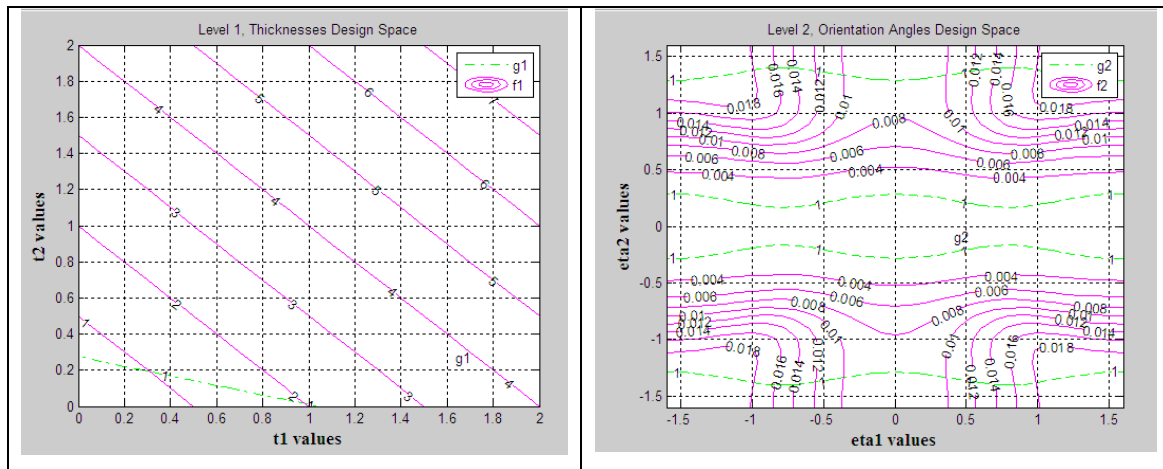


Figure 5-44 – Iteration1, 2D L1 and L2 design space plots

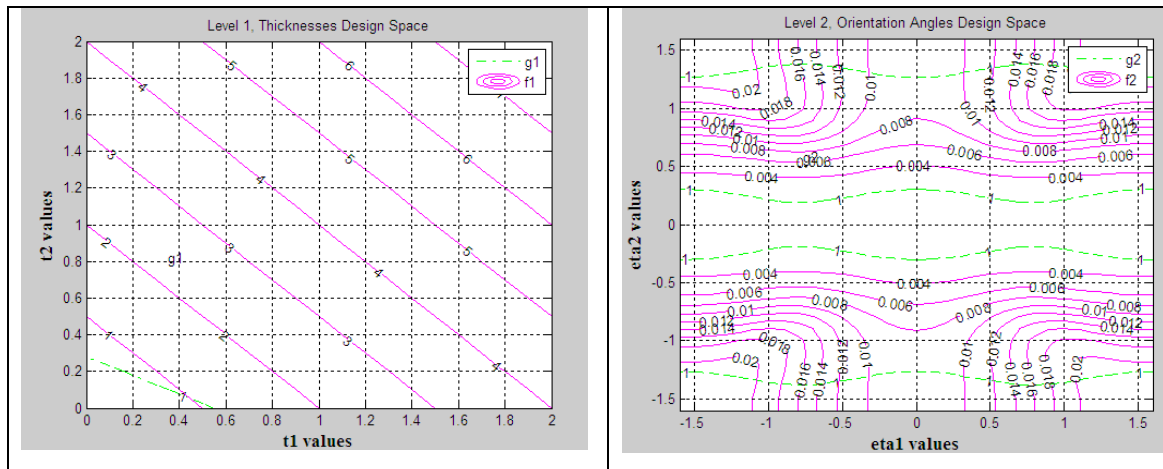
Iteration1 Design Variables	Level 1	Level 2
Eta1 [radians]	0.79	-0.8
Eta2 [radians]	0.38	0.4
t1 [mm]	0.1	0.1
t2 [mm]	0.3	0.3

Table 5-7 – Iteration1, Level 1 and Level 2 Design Variables

RUN1 – Iteration2**Figure 5-45 – Iteration2, 2D L1 and L2 design space plots**

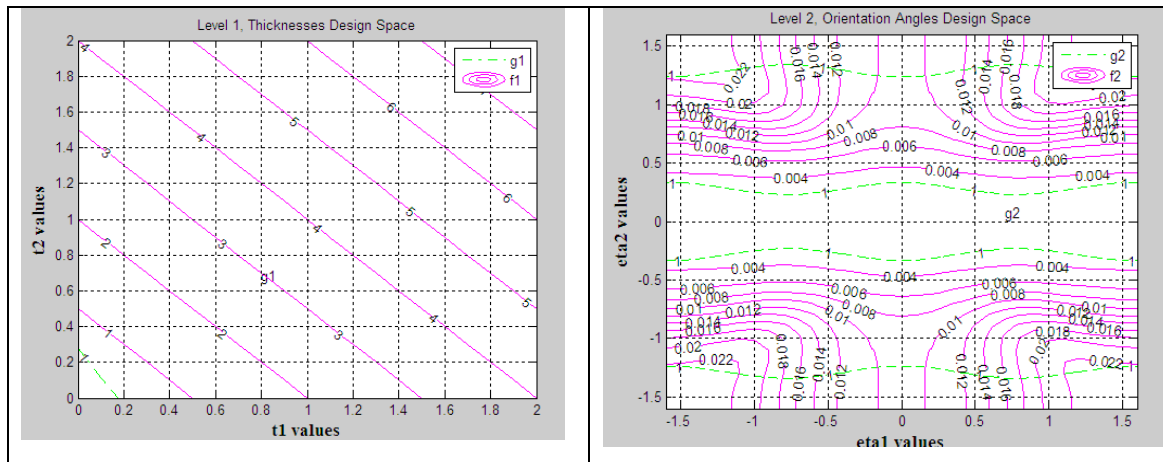
Iteration2 Design Variables	Level 1	Level 2
Eta1 [radians]	-0.8	-0.6
Eta2 [radians]	0.4	0.4
t1 [mm]	0.05	0.05
t2 [mm]	0.29	0.29

Table 5-8 – Iteration2, Level 1 and Level 2 Design Variables

RUN1 – Iteration3**Figure 5-46** – Iteration3, 2D L1 and L2 design space plots

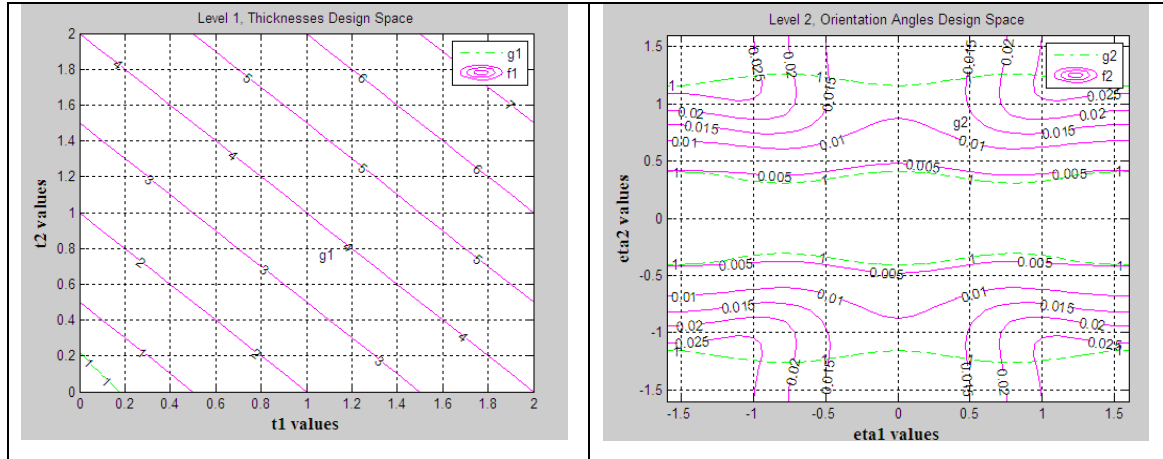
Iteration3 Design Variables	Level 1	Level 2
Eta1 [radians]	-0.6	-0.1
Eta2 [radians]	0.4	0.4
t1 [mm]	0.05	0.05
t2 [mm]	0.27	0.27

Table 5-9 – Iteration3, Level 1 and Level 2 Design Variables

RUN1 – Iteration4**Figure 5-47 – Iteration4, 2D L1 and L2 design space plots**

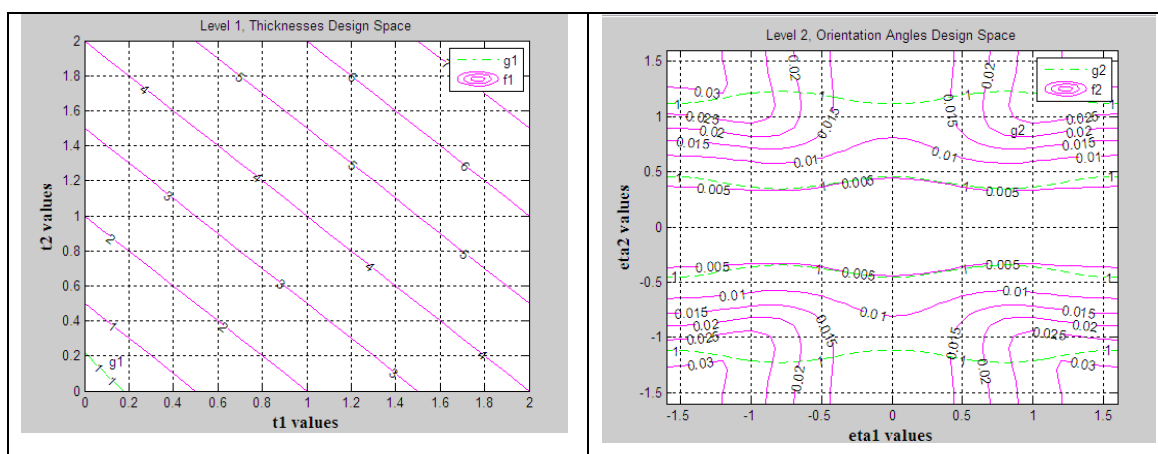
Iteration4 Design Variables	Level 1	Level 2
Eta1 [radians]	-0.1	-0.1
Eta2 [radians]	0.4	0.4
t1 [mm]	0.045	0.045
t2 [mm]	0.25	0.25

Table 5-10 – Iteration4, Level 1 and Level 2 Design Variables

RUN1 – Iteration5**Figure 5-48** – Iteration5, 2D L1 and L2 design space plots

Iteration5 Design Variables	Level 1	Level 2
Eta1 [radians]	-0.1	-0.1
Eta2 [radians]	0.4	0.4
t1 [mm]	0.04	0.04
t2 [mm]	0.24	0.24

Table 5-11 – Iteration5, Level 1 and Level 2 Design Variables

RUN1 – Iteration6**Figure 5-49** – Iteration6, 2D L1 and L2 design space plots

Iteration1 Design Variables	Level 1	Level 2
Eta1 [radians]	-0.1	Not Applicable
Eta2 [radians]	0.4	Not Applicable
t1 [mm]	0.04	Not Applicable
t2 [mm]	0.19	Not Applicable

Table 5-12 – Iteration6, Level 1 and Level 2 Design Variables

Intention of Run1 was to test the Validation Test, 5.1.4, and Convergence Test, 5.1.5, at the beginning of this Optimisation Cycle. Then, it was meant to explore the two design spaces, Level 1 and Level 2, trying to obtain at the very end of the whole optimisation process the best possible laminate according to the objective and constraint functions previously defined.

At the same time the iterations proved to move from one level to the other narrowing down the respective design space and consequently enabling the user to refine the laminate configuration in terms of stack thicknesses and orientation angles. That is very

clear, not just going through the plots of Figure 5-44 Figure 5-45 Figure 5-46 Figure 5-47 Figure 5-48 and Figure 5-49, but looking at the final laminate thickness of iteration 6 compared to the one of iteration 1. The orientation angle design space shrinking effect is visible mainly looking at the plots instead at the numbers. In fact, Figure 5-50 describes clearly the zone defined by the arrows, orientation angle allowable area, is closing down.

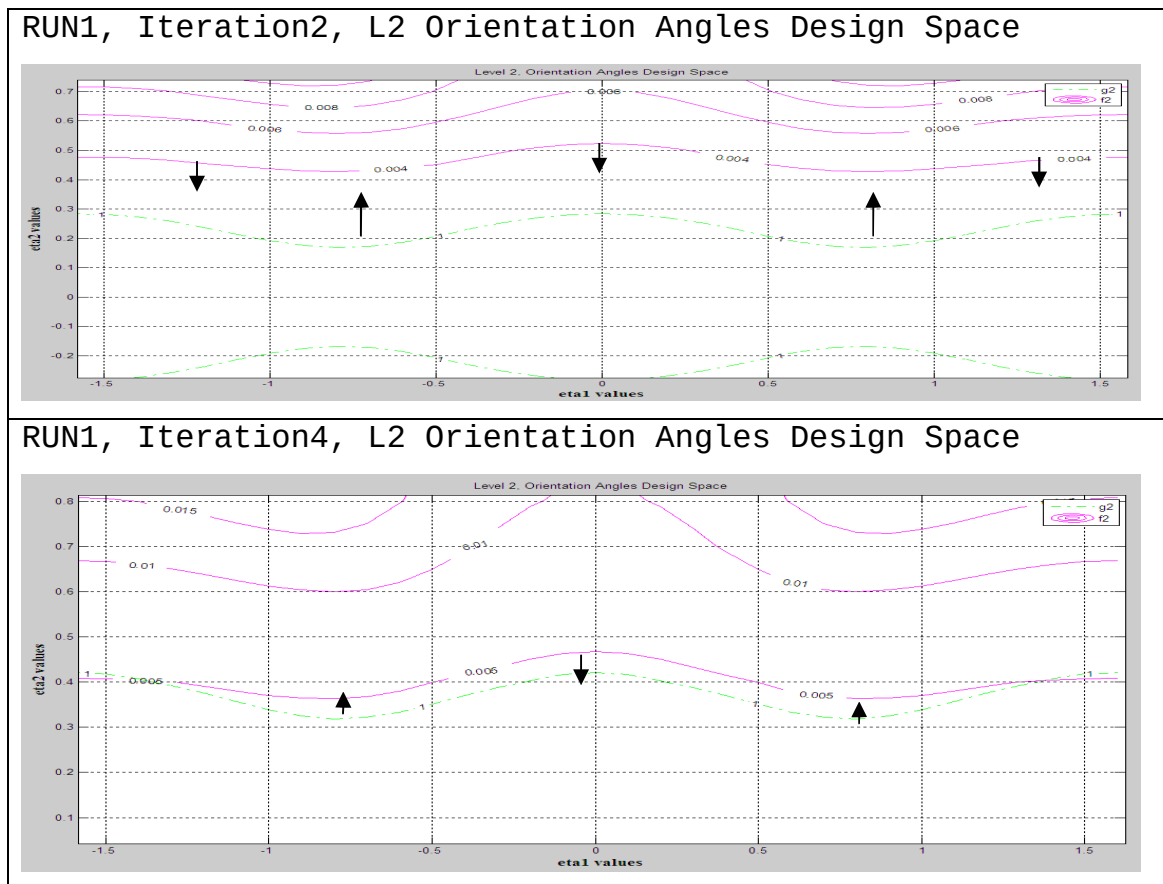


Figure 5-50 – RUN1 Orientation Angle Design Space Convergence Effect

Finally, Figure 5-51 shows the orientation angle design space at the last iteration; when the allowable region has been definitely closed down and the whole process stops due to achieved convergence.

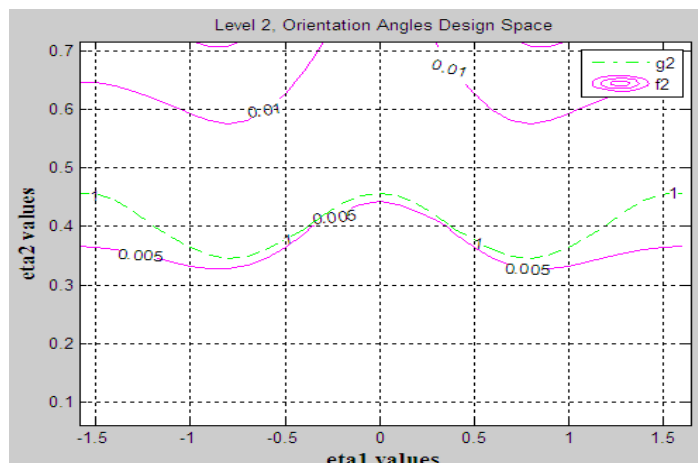


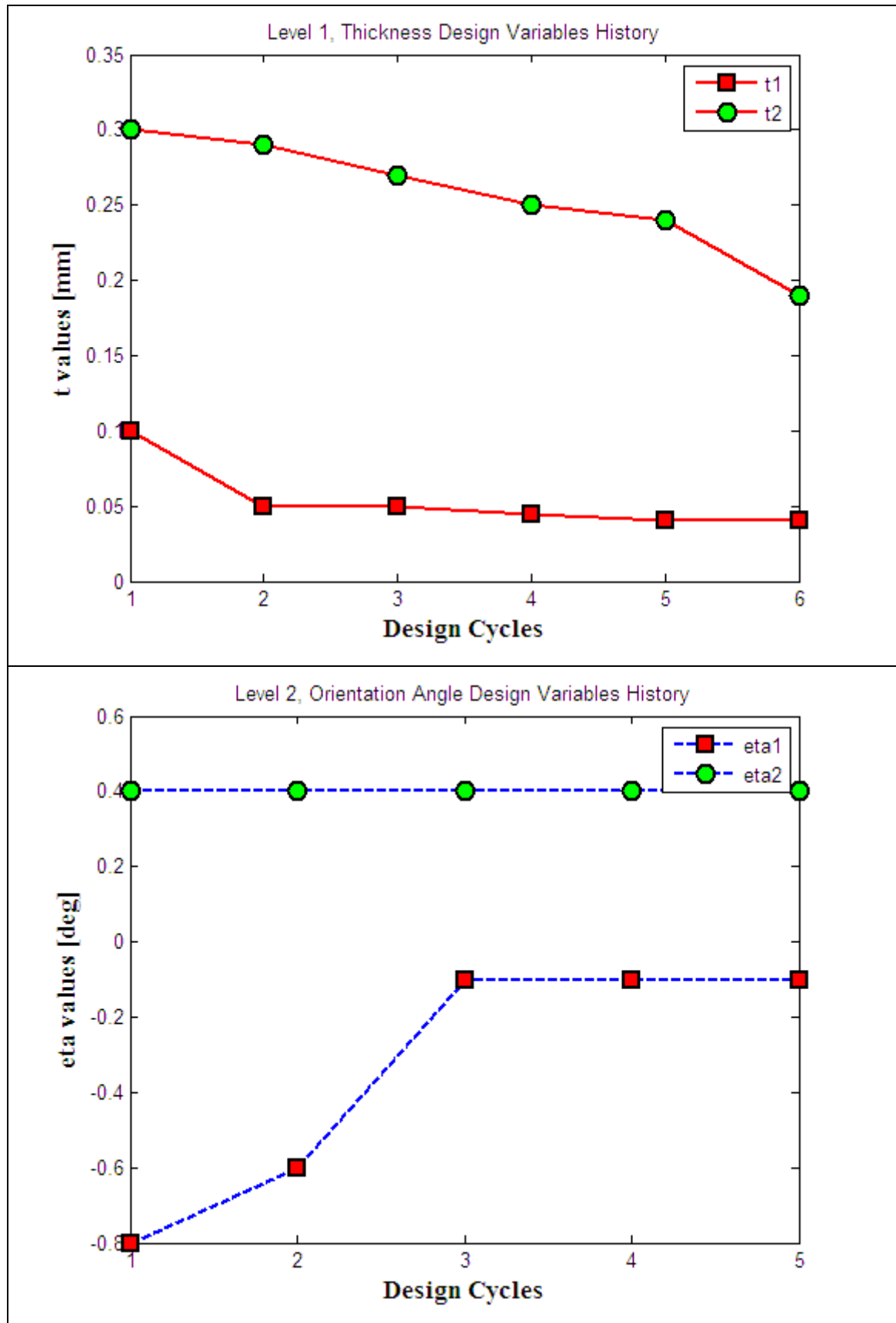
Figure 5-51 - RUN1 Orientation Angle Design Space: Achieved Convergence

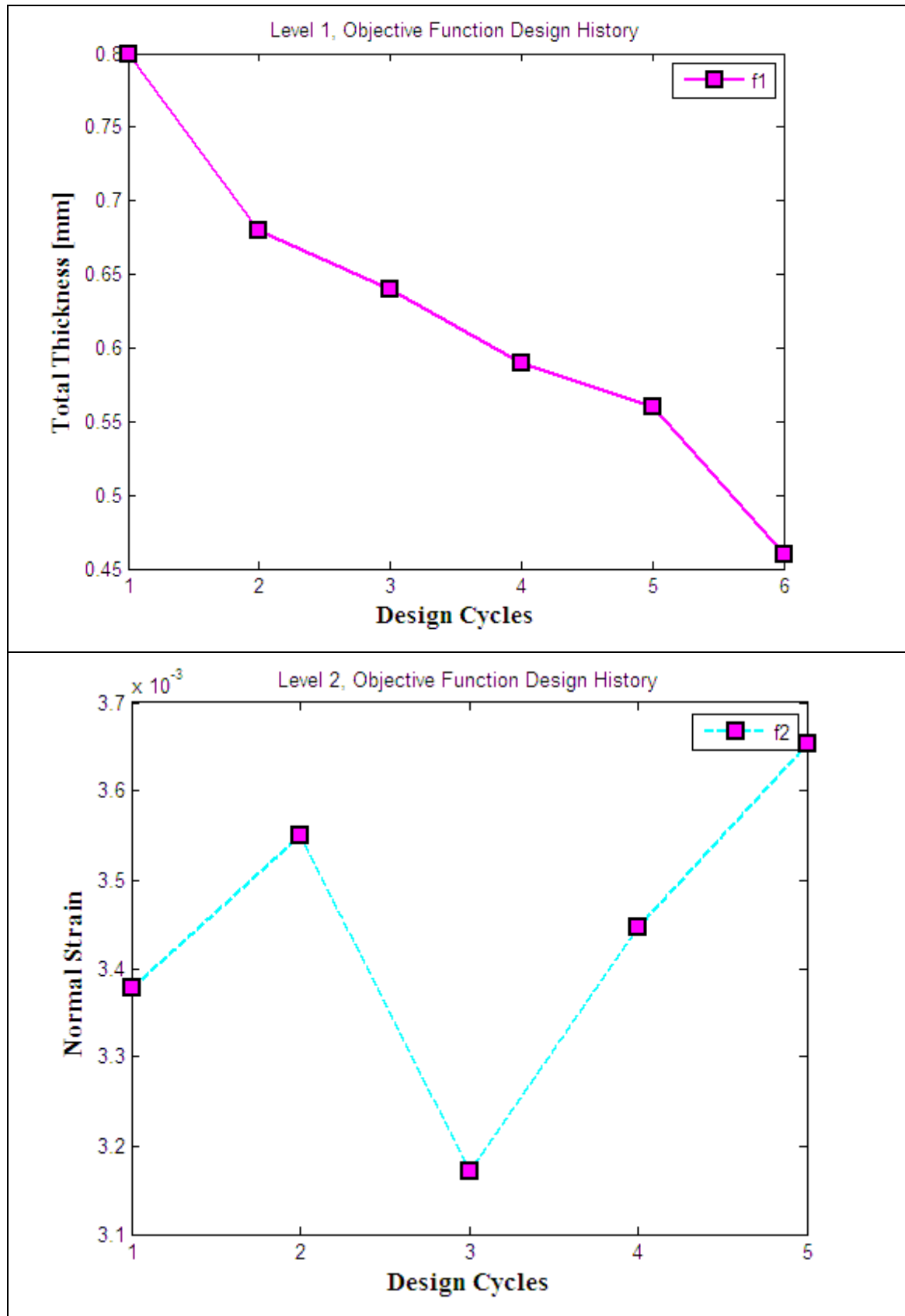
A complete summary of RUN1 Optimisation is grouped in the following plots presenting at level1 and level2 the Design Variable, Objective Function, and Constraint Function quantities against the design cycles.

The Table 5-13 below gathers all data used to plot: Figure 5-44, Figure 5-45, Figure 5-47, Figure 5-52, Figure 5-53, and Figure 5-54.

Cycle Number	Level1					Level2				
	$t1$	$t2$	$f1$	$g1$	FEM_x	$eta1$	$eta2$	$f2$	$G2$	FEM_{xy}
1	0.1	0.3	0.8	0.003248	0.003265	-0.8	0.4	0.003378	0.002793	0.001986
2	0.05	0.29	0.68	0.003655	0.003676	-0.6	0.4	0.003550	0.003568	0.002549
3	0.05	0.27	0.64	0.003791	0.003795	-0.1	0.4	0.003171	0.004498	0.003310
4	0.045	0.25	0.59	0.003447	0.003449	-0.1	0.4	0.003447	0.004867	0.003590
5	0.04	0.24	0.56	0.003653	0.003633	-0.1	0.3	0.003653	0.005093	0.003584
6	0.04	0.19	0.46	0.003659	0.003806					

Table 5-13 – RUN1, Level1 and Level2 Design History

**Figure 5-52** – RUN1, L1 and L2 Design Variables History

**Figure 5-53** – RUN1, L1 and L2 Objective Function History

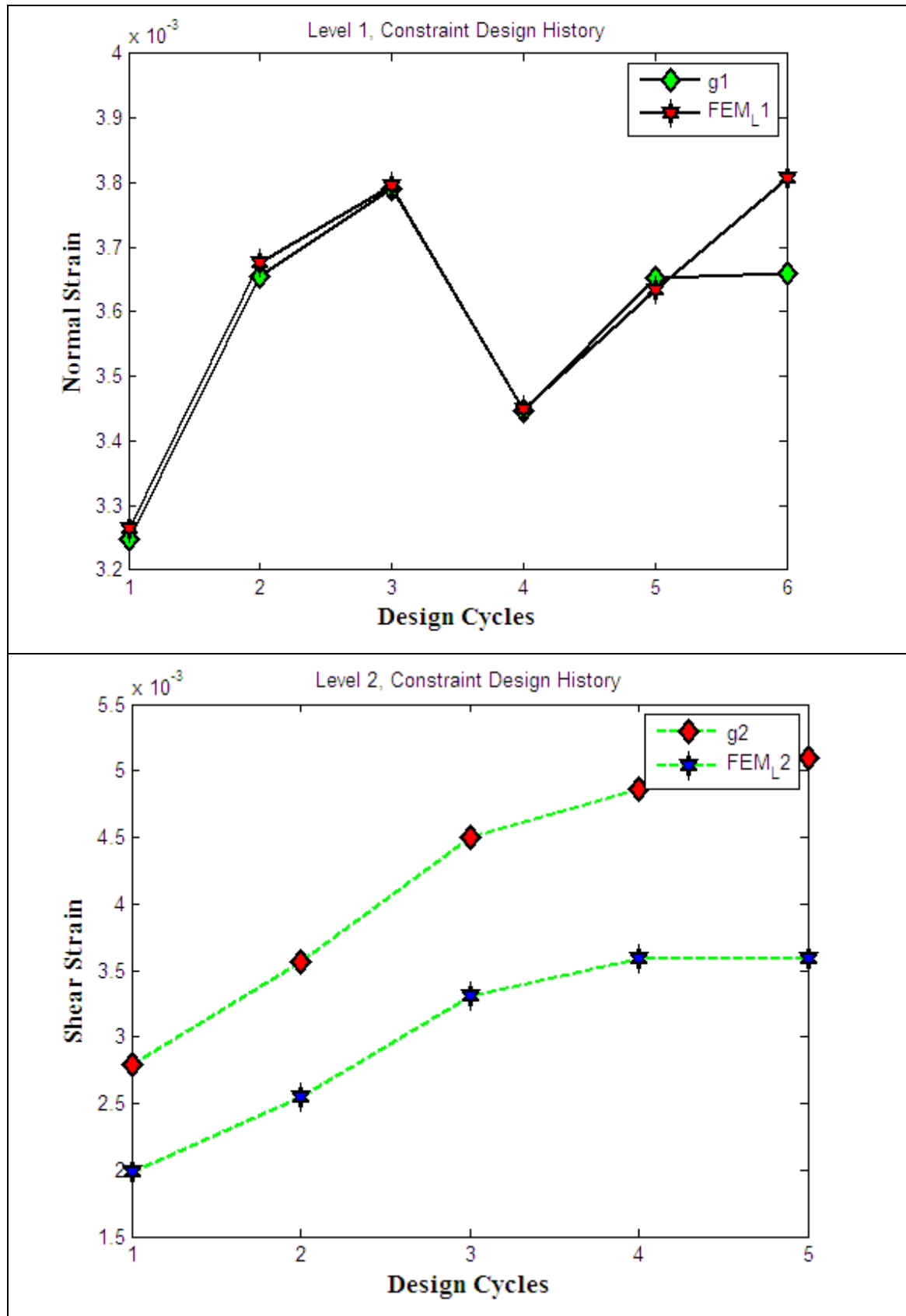


Figure 5-54 – RUN1, L1 and L2 Constraint Function and FEM strain data History

The top plots of Figure 5-52, Figure 5-53, and Figure 5-54 show level1 solid line design cycle quantities and the bottom dashed line plots are related to level2. The level1 plots present a greater number of iteration data than level2 since the main Matlab script stopped at the sixth iteration of the first level when the thicknesses were chosen, Figure 5-47, outside the feasible area and then not enabling the program to reach level 2 having an unsuccessful validation test, Figure 5-2.

The Level 2 design variable plots of Figure 5-52 show that the orientation angle design variables are not self explicative to indicate whether the user is leading the design cycles to a convergence solution. Whilst the thickness design variables of level1 have this capability since the mathematical relation to the objective function f_1 is linear. In fact, when the top plot of Figure 5-52 has negative slope the same does the Figure 5-52 level1 objective function.

Indicative of the correctness of the whole main program is showed by the Figure 5-52 objective function slopes. The top plot of this figure is the objective function of level1, laminate total thickness, meant to be minimised and in fact the slope is negative and the bottom plot is the objective function, normal strain, expected to be maximised and with positive slope apart the *Design Cycle 3*. This is explained to the choice of η_1 , orientation angles of the first ply group, smaller compared to the respective one of the previous iteration, from -0.6 to -0.1 radians. In fact, that gives more strength to the laminate configuration, and smaller normal strain, in the x direction (the fiber direction is closer to the x load direction). Then the magnitude of the slope might provide more information relating the effectiveness of each design cycle to get to the optimum design and that is clearly related to amount of decrease of the ply thicknesses. In fact, the steepness of f_1 , Figure 5-52 top plot, is much greater going from iteration 1 to 2 and 5 to iteration 6 than any other iteration, and that understandable looking at the same design variable plots of Figure 5-52. The t_1 and t_2 stack thicknesses show consistent decrease from iteration 1 to 2 (t_2) and 5 to 6 (t_1).

Finally Figure 5-52 describes a correct motion of the search closer to the allowable constraint values accepted at the beginning of the optimisation process when (Eq. 5-1) it has been highlighted the normal and shear strain not to be overcome are respectively 0.004 and 0.006. In particular, the discontinuity of the top plot design cycle 4 of Figure 5-54 is a consequence of the advantageous change of the orientation angle design

variable at the previous iteration, from -0.6 to -0.1. Regarding the bottom plot of Figure 5-54 the trend is coherently controlled moving to the upper limit constraint. Moreover the constraint function plots enables to compare the FEM data with the CLT and the outcome says how close are the normal strain values despite the poor similarity of the shear strain ones. The final constraint design history value is an important effectiveness index that describes how close the whole optimisation reached the allowable values, Cons1 and Cons2.

At the end of this investigation it is worth introducing two index coefficients useful to understand:

- Reliability of the model configuration
- Effectiveness of the whole Design Cycles

The former mainly is coming from the comparison of the normal and shear strain, g_1 and g_2 , with the FEM output. It is defined by the following two:

$$(RI_N)_k = \frac{(g_1)_k}{(FEM_x)_k} \quad (\text{Eq. 5-22})$$

$$(RI_S)_k = \frac{(g_2)_k}{(FEM_xy)_k} \quad (\text{Eq. 5-23})$$

These two equations represent respectively the normal, (Eq. 5-22), and shear, (Eq. 5-23), reliability indexes.

The latter coefficient is based on the ratio of the allowable constraint values and the constraint functions. The coming two equations, first related to level1 constraint values and second to level2, describe this index of effectiveness:

$$(EI_1)_k = \frac{Cons_1}{(g_1)_k} \quad (\text{Eq. 5-24})$$

(Eq. 5-25)

$$(El_2)_k = \frac{Cons_2}{(g_2)_k}$$

These four equations, (Eq. 5-22), (Eq. 5-23), (Eq. 5-24), and (Eq. 5-25) are expressed in terms of k that indicates the iteration involved in the analysis of the reliability and effectiveness indexes. In fact, these coefficients can be evaluated at any design cycle step of the analysis and mainly might representative just at the final. The following Table 5-14 is illustrative of the reliability and effectiveness indexes evaluated for the RUN1 at the last iteration available, level1 (normal strain) at $k=6$ and level 2 (shear strain) at $k=5$.

	$(Rl_N)_k$	$(Rl_S)_k$	$(El_1)_k$	$(El_2)_k$
RUN1, (L2: $k=5$) (L1: $k=6$)	0.9614	1.421	1.0932	1.1781

Table 5-14 – RUN1 Reliability and Effectiveness Indexes, final iteration ($k=5$ and $k=6$)

The above data indexes have been evaluated using the values coming from Table 5-13 and the constraint allowable numbers of (Eq. 5-1) where $Cons_1$ is equal to 0.004 and $Cons_2$ to 0.006. Then, the first two indexes from left basically confirmed what already mentioned before about the very good matching of the normal strain of the Classical Lamination Theory and FEM, number very close to one. On the other hand the shear strain is worst being just 1.421. To understand completely what this index says it is needed to compare the trend of the strain, normal and shear, through the iterations. In fact, Figure 5-54 describes properly that the reliability index of the normal strain level 1, top plot, all the way through is the same just at the last iteration shows a slight difference, 0.9614. That is due to the decreasing of the thickness of the laminate that offers to believe the Classical Lamination Theory tends to differ to the FEM idealisation. On the other hand, the normal strain starts from the very beginning to a very poor reliability index, in Figure 5-54, the “square” and “star” marker never match. That seems to be an effect explainable to different causes. First the cantilever beam

constraint might take into the final shear strain an effect of the lateral bending. Though, looking at the strain contour of Figure 5-24 and Figure 5-26, it seems the element 25 does not have much more values than the element, of the same central strip, at the tip. Then, it seems to be possible that the Wisnom, [44] [45], assumption stated by (Eq. 2-1), regarding the homogeneous properties mentioned in section 5.1.3, might need to be further investigated.

The last two indexes, RI, give a feedback of the amount of feasible space used in the whole optimisation process. In fact, 1.0932 says how good the final optimum design of level1 was since g_1 is 0.003659 very close to Cons1 (0.004). Differently the RI of level2 describes the amount of shear strain feasible space still left to be potentially used. Moreover the reliability index introduces one of the main difficulties of employing a multi-level decomposition optimisation process. To obtain a close to one index for both levels is the very first goal of the whole search optimum design activity. The more the levels, part of the decomposition, and the higher the difficulty to control them all through the iterations. A recommended way to pay attention is refining the design space with very small decrement and understanding thoroughly the physical problem.

The all four indexes show the highest quality of the optimisation is when they are very close to the unity.

It is very likely that all these considerations may be optimised in a RUN where the overall number of iterations is much greater than the just three design cycles faced in RUN1.

5.3 Summary

This chapter has been crucial for a full understanding of the whole work presented in this thesis since introduces the main structure of the multilevel decomposition process employed. Many of these paragraphs will be employed by the Intermediate Example too.

Specific two-level decomposition has been outlined describing the main steps within each level. Alongside the main characterisation boxes of the process it has been decided to illustrate a numerical application example. This choice enables a better

comprehension of the core box of the technique under investigation; the Graphical Optimiser. In fact, the main features of Matlab have been used to describe the different visual inspections to be performed at level1 and level2 meant to identify the optimum design to pass then to the next level. Bi-dimensional and three-dimensional plots have been discussed in detail showing importance to be considered in particular in the orientation angles design space where objective and constraint function present non linear behaviour. Furthermore it has explained the Matlab program flexibility to link with external software as Nastran and to use it as internal validation test at each level. Finally the pivotal box of the convergence test has been taken in a detailed consideration as responsible to identify when the convergence has been reached. All these main concepts have been described showing the programming side and with very specific citation to the main Matlab script.

The other main section of this chapter is a continuation of the numerical application introduced earlier to explain the 2D and 3D plots of the graphical optimiser box and of the Finite Element Modelling. The main iteration steps have been broken down to a list of the main quantity data trend and then gathered in final plots as summary of the design optimisation history. Reliability and effectiveness indexes have been defined as tool to judge the quality of the overall optimisation process. Despite not part of this work it is a strong belief that these indexes will be an important starting point to perform any future sensitivity analysis just studying and understanding their distribution alongside the design variable of each level.

The whole chapter is to be considered the main platform on which to build an evolved 2-level decomposition structure that will be supposed to overcome space limitation due to design variables of an infinite ply laminates. The intermediate example is meant to explore in that direction employing at level2 different design variables; the lamination parameters (4.4.2).

6 Intermediate Example

The intention of this section is to remove the limitations left by the Basic Example, and whilst at the same time, still aiming to develop such tools and concepts necessary to deal with optimisation of thick composite laminated materials.

The first goal is to apply the Multilevel Decomposition to a real life engineering application, not just to a laminate consisting of only four layers. The numerical methods used are still iterative, maintaining the same objective and constraint functions that allow a smooth search of the optimum design orientation angles, with the intention of improving the reliability and quality of the final optimum design.

The feasibility of these tasks will be possible by introducing different design variables at each level, (with respect to the Basic Example), but mainly working with a much reduced and controlled design space by introducing the “Lamination Diagram” in the 2-D domain of the lamination parameters V_1^*, V_3^* (see 4.4.2).

Identifying the quantities to couple within the MD to allow iterations to investigate successfully will be the crucial step of this intermediate example.

With respect to all above, enhancements will be required within the Matlab Script, by modifying these tasks:

- Classical Lamination Theory (CLT)
- Graphical Optimiser
- Convergence Test

The following method is presented for minimum weight design of a symmetric and balanced fibre-composite laminates subject to multiple in-plane loading conditions

which takes into account fixed ($0^\circ/90^\circ/\pm 45^\circ$), variable orientation angles ($\pm \vartheta_1/\pm \vartheta_2$) and strain limitations.

The measure of unit will be the SI with main quantities as Newton (N), millimetre (mm)

and Mega-Pascal ($MPa = \frac{N}{mm^2}$) to match coherently the Nastran input file too.

6.1 Optimisation Procedure

The optimisation procedure employed herein is an enhanced adaptation of the Multilevel Decomposition approach already used within the Basic Example.

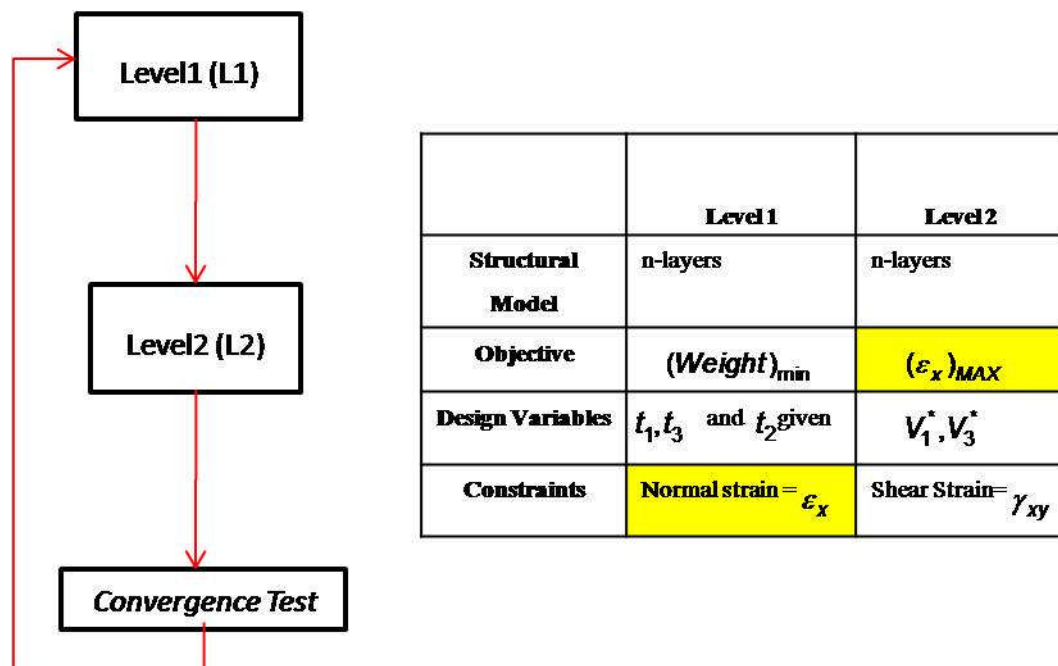


Figure 6-1 – Multilevel Decomposition structure: Intermediate Example two-level decomposition

Figure 6-1 points out the main difference to the Basic Example and its analogue picture, Figure 5-1, says that the followings have been changed:

- Structural model
- Design Variables

In fact the laminate is not anymore constrained to a specific number of layers and consequently the design variables are reviewed. In particular a new optimisation tool has been introduced at level 2 by adding a different set of design variables, the lamination parameters.

From a more detailed analysis of the structure of each level, Figure 6-2 reveals further differences to the Basic Example. The dashed-line rectangle frame of the below figure, equivalent to the (B) boxes of Figure 5-2, defines the main area of distinction.

However the main boxes that will be described in the next sections are listed below:

- **A:** Analytical model
- **B:** B bis1 (Numerical Optimiser) and B bis2 (Graphical Optimiser)
- **C:** Numerical Model (CLT, FEM descriptions)
- **V:** Validation Test (Constraints Check), Li(V) and B(V).
- **OD:** Optimum Design

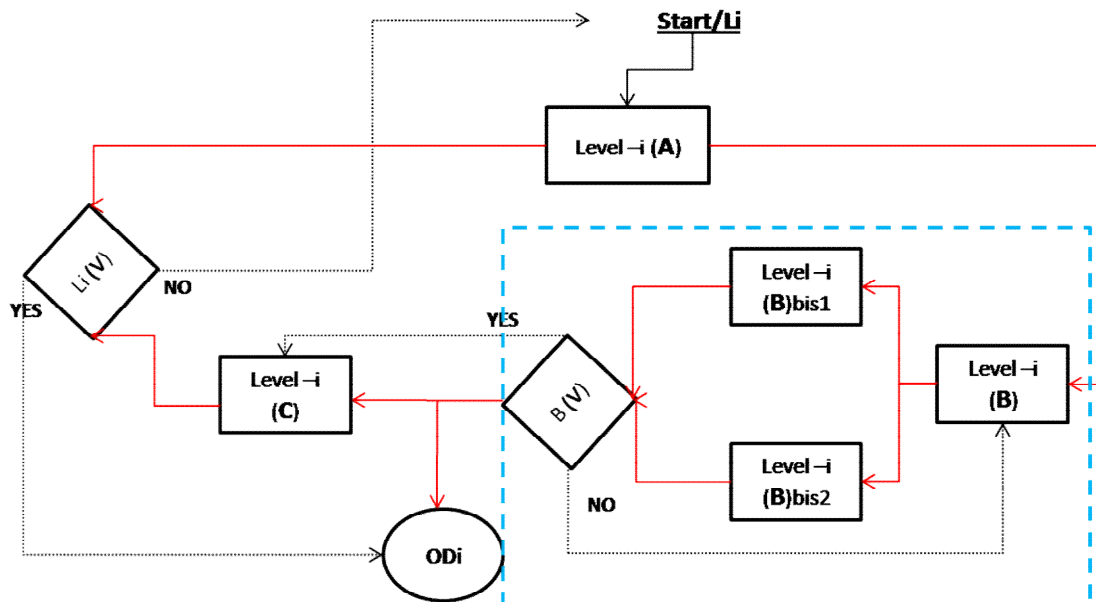


Figure 6-2 Intermediate Example Multilevel Decomposition, single Level-i structure

The main structure of the next Boxes Description will follow mainly the same, already presented in the Basic Example in section 5.1.

In fact, again it will be adopted alongside the “blocks” that will require, a numerical application of a laminate made of Graphite/Epoxy, material properties (Eq. 5-1), and with the same load cases and constraints (shear and normal strain) of the Basic Example listed by (Eq. 5-2), and (Eq. 5-3) seeking the lightest weight configuration. Then the main distinction introduced by the Intermediate Example is about the final structure of the optimum candidate laminate. The two laminate arrangements considered are:

- 3 Fixed-Orientation Angles (3_F_OA)
- 2 Variable-Orientation Angles (2_V_OA)

The following two figures, Figure 6-3 and Figure 6-4, describe the two different laminate configurations above.

$$\left[(\pm 45)_{N3} (90)_{N2} (0)_{N1} \right]_{nS}$$

3	N3	$\Theta 3 = +/- 45^\circ$
2	N2	$\Theta 2 = 90^\circ$
1	N1	$\Theta 1 = 0^\circ$

MID-Plane

Figure 6-3 – Intermediate Example: 3-Fixed-Orientation Angles (3_F_OA) half laminate configuration

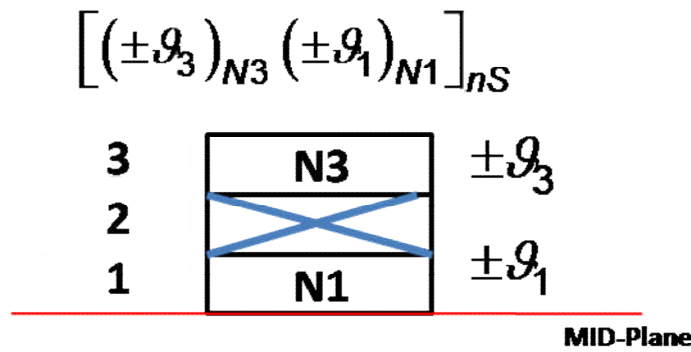


Figure 6-4 – Intermediate Example: 2-Variable-Orientation Angles (2_V_OA) half laminate configuration

Here, again the study focuses on symmetrical and balanced laminate as anticipated by the above figures. Then both arrangements are counted to be used in the same design cycle searching mechanism for the optimum laminate candidate. That means, as later on will be showed, the user during the iteration phase can afford to swap from one configuration to the other. This flexibility explains the reason the convention used, Figure 6-3 and Figure 6-4, is always the same.

Then, the enhancements of the current example are due to the attempt to define a real life laminate. For such reason the final laminate configuration will be characterised by:

- Orientation Angles
- Volume Fraction (proportion of plies among the different angles)
- Number of layers per Orientation Angles (these are not limited)

Optimisation methods for problems with continuous design variables are usually much more efficient than methods that have to handle integer variables. For this reason it is common for designers to replace integer variables with continuous ones, whenever this does not introduce large errors [1]. To adopt this approach, the thickness of a single layer has been counted as a fixed value, 0.127 mm, and the design variables employed are the stack thickness of each orientation angles. These variables will be later replaced by integer ply numbers rounding up/down to the optimum value of the variable of the nearest integer value.

6.1.1 Analytical Model (A)

At this stage of the Intermediate Example the main difference compared to the Basic Example, is defining the mathematical terms of the objective and constraint functions in terms of the new design variables, the lamination parameters, 4.4.2, of the second level of the decomposition. Note that the main optimisation structure of this example is showed in

Table 6-1, where as for Table 5-1 of the Basic Example, the main quantities involved in the whole optimisation process are broken down at each level.

The bold are characters have the same chapter meaning as follows:

- Objective Functions : **f(X)**
- Design Variables : **X**
- Constraint Functions : **g(X)**
- Design Parameters : **P**

	Level 1	Level 2
f(X)	$(\Sigma t_i)_{\min}$	$[\varepsilon_x(V_1^*, V_3^*)]_{\max}$
X	t_1, t_3	V_1^*, V_3^*
g(X)	$\varepsilon_x(t_i)$	$\gamma_{xy}(V_1^*, V_3^*)$
P	L_i, G_i, MP_i	L_i, G_i, MP_i

Table 6-1 Optimisation Problem Definition: Intermediate Example

6.1.1.1 LEVEL 1

Design Variables – **X** = t_i

The design variables for the laminate at the first step of the Multilevel Decomposition are the stack thickness of each group of orientation angles $\pm\theta_i$. In this way, the laminate

is broken up in sub-laminates, 3.6.4, characterised by the same type of orientation angles, material properties and ply thickness (t_{ply}).

Reminding that the laminate is a symmetric and balanced, the “stack thickness” t_i will be designated as:

$$\begin{aligned} & \text{(Eq. 6-1)} \\ & - \vartheta_i = 0^\circ \text{ or } 90^\circ \text{ then } t_i = 2 N_i (t_{ply}) \\ t_i \Rightarrow & \\ & - \vartheta_i \neq 0^\circ \text{ or } 90^\circ \text{ then } t_i = 4 N_i (t_{ply}) \end{aligned}$$

Where N_i are the total numbers of plies of the orientation angles $\pm\vartheta_i$ and the factor 2 and 4 is mainly part of the balanced condition that will lead to have a membrane stiffness matrix ($\mathbf{A}$) with $A_{16} = A_{26} = 0$ already discussed by (Eq. 3-50).

Design Variable Side Constraints – $\mathbf{X} = t_i$

The design variables need to satisfy basic constraints mainly related to the design model adopted and are essential to drive the search for the optimal design.

The stack thicknesses need to be positive to allow the validity of the design and to enhance the final results; respect what came out from the Basic Example, the thickness will be a multiple of a commercial standard thickness layer t_{ply} . In fact it is common to replace integer variables, as would be expected in this example, with continuous ones.

$$\text{Objective Function} - \mathbf{f}(\mathbf{X}) = (\sum t_i)_{\min}$$

The objective function is to minimise the mass of the laminate. The mass at this level is just a function of the design variables as a simple sum. This makes the final equation very much linear and easy to deal with in the next optimisation task. In fact, the final equation is given by:

$$\mathbf{f}(\mathbf{X}) = t_1 + t_2 + t_3 \quad \text{(Eq. 6-2)}$$

In this problem, the final laminate is assumed to be the result of three orientation angle groups.

$$\underline{\text{Design Parameters}} - \mathbf{P} = (L_i, G_i, MP_i)$$

It is very much applicable what mentioned by the Basic Example at this stage. In fact the design parameter quantities behave at the same way and are:

- material property
- applied loads
- geometry quantities

$$\underline{\text{Constraint Functions}} - \mathbf{g}(\mathbf{X}) = \varepsilon_x$$

The main constraint equations are identical to the previous basic example; it means they are still led by inequality constraints as it will be shown by the end of this section. The main difference is related to the setting of the laminate under consideration; not anymore a 4 layers but a n-layers. However the following (Eq. 5-6), (Eq. 5-7), (Eq. 5-8), and (Eq. 5-9) are still applicable.

6.1.1.2 LEVEL 2

$$\underline{\text{Design Variables}} - \mathbf{X} = V_1^*, V_3^*$$

The design variables at the second level of the MD advanced example are the lamination parameters, V_1^* and V_3^* . Lamination parameters allow laminate stiffness, Engineering Effective Constants, to be written as a linear combination of ply stiffness's and linear trigonometric terms. In fact for a laminate with:

- n orientation groups

- v_i Volume fraction of the layers with $\pm\theta_i$ orientation angles

The lamination parameters V_1^*, V_3^* are as follows:

$$V_1^* = \sum_{k=1}^n v_k \cos 2\theta_k \quad (\text{Eq. 6-3})$$

$$V_3^* = \sum_{k=1}^n v_k \cos 4\theta_k \quad (\text{Eq. 6-4})$$

As, already described in section 4.4.2, by (Eq. 4-6) throughout (Eq. 4-3), (Eq. 4-4), (Eq. 4-7), (Eq. 4-8), and trigonometric identity it is then possible to define the equation of the feasible domain of this level as parabola:

$$V_3^* = 2V_1^{*2} - 1 \quad (\text{Eq. 6-5})$$

Design Variable Side Constraints – $\mathbf{X} = V_1^*, V_3^*$

Considering the equations employed above to describe the Level 2 design variables the design space region will be coherently satisfied when:

$$-1 \leq V_1^*, V_3^* \leq 1 \quad (\text{Eq. 6-6})$$

It means the feasible domain is determined by the points on the edge or inside the curve described by $V_3^* = 2V_1^{*2} - 1$.

Such a simple condition shows immediately how reduced is the search of the best laminate candidate, when compared to the Level 1 configuration.

Objective Function – $\mathbf{f}(\mathbf{X}) = [\varepsilon_x(V_1^*, V_3^*)]$

Level 2 objective function is, according the principles of the HMD (see Section 4), the constraint function of the higher Level. Moreover, consistently to the idea to narrowing the design space, this function, ε_x , will be maximised with the intention that subsequent iterations after iterations will converge to the optimal design.

So the normal strain expressed in terms of the lamination parameters, (Eq. 6-3) and (Eq. 6-4), design variables, will be presented as:

$$\varepsilon_x = \frac{N_x}{hE_x^{eff}} = \frac{N_x}{h} \left(\frac{U_1 - V_1^* U_2 + V_3^* U_3}{U_1^2 - U_4^2 + 2U_1 U_3 V_3^* + 2U_4 U_3 V_3^* - U_2^2 V_1^{*2}} \right) \quad (\text{Eq. 6-7})$$

The above analytical normal strain expression is a result of the combinations of the following equations:

- (Eq. 4-3)
- (Eq. 4-4)
- (Eq. 4-7)
- (Eq. 4-20)

Design Parameters – $\mathbf{P} = (L_i, G_i, MP_i)$

Like Level 1 of Basic and Intermediate Example, the design parameters, applied loads, plus

- material property
- geometry quantities

That are related to the single ply will not change at any time during the multilevel decomposition optimisation.

$$\text{Constraint Functions} - \mathbf{g}(\mathbf{X}) = \gamma_{xy}$$

Constraint functions of Level 2 is an inequality equation of the last term of the strain criteria; the shear strain “ γ_{xy} ”.

“ γ_{xy} ” defined in terms of the lamination parameters, employing the same main mathematical steps showed to describe the normal strain [$\varepsilon_x(V_1^*, V_3^*)$], and is given by:

(Eq. 6-8)

$$\gamma_{xy} = \frac{N_{xy}}{hG_{xy}} = \frac{N_{xy}}{h} \left(\frac{1}{U_5 - U_3 V_3^*} \right)$$

6.1.1.3 MATLAB SCRIPT: BOX A

The Program starts, within the **IE_Main.m** (in Appendix B :”Intermediate Example Matlab Script and related files” are gathered all the lines of the code) Matlab man script, with the same format of the Basic Example . Then the variables and invariants of each level are presented in the coming sections.

Since the Matlab command lines to refer to are most of the time similar to the Basic Example, it has been chosen to avoid repeating the “**Matlab Command**” all the time that happens. So from now onwards throughout the intermediate example description it may happen to find references to the basic example Matlab section still applicable.

Input Starting Values, which will feed the Objective and Constraint Functions to be plotted at the next Graphical Optimisation (Box B) follow.

Level1and Level2 Invariants:

- N_x and N_{xy} loads over width $\left[N_x = \frac{\text{Newton}}{\text{mm}} \right]$
- Single Layer Material Properties defined by a **P** vector
- $th1 = 0.127$ mm as single ply thickness

Level1 Variables and Invariants:

- $P_x = \begin{pmatrix} N_3 \\ N_2 \\ N_1 \end{pmatrix}^T$: N_i are the Number of Layers of half laminate (the laminate is symmetric and balanced) for the three different orientation angle groups assumed by the program. The convention of this numbering is shown in Figure 6-5.
- $ply_angle = [\vartheta_3 \ \vartheta_2 \ \vartheta_1]^T$ Where the program has been designed so the user has the option to describe a candidate laminate with three fixed orientation angles ($0^\circ/90^\circ/\pm 45^\circ$), Figure 6-3, or two variable angle groups ($\pm\vartheta_1/\pm\vartheta_2$), Figure 6-4.

Level2 Variables and Invariants:

- $Th = [t3_L1 \ t2_L2 \ t2_L1]$: Each element is the stack thickness, design variable of the previous level (a part $t2_L2$), of the whole laminate (the laminate is symmetric and balanced) for the different orientation angle groups assumed by the program. If the chosen laminate leads to a two variable angle groups automatically the $t2_L2$ becomes zero. The convention of this numbering is shown in Figure 6-5.
- N_{tot} : Total number of layers of the candidate laminate. Quantity coming as direct output of Level1. Considered as “Couple Quantity”, since part of the main equations of the optimisation problem at level2.

Ply angles, Ply Properties, and Stack Thicknesses are stored at each level into text files at the same way already described by the Basic Example.

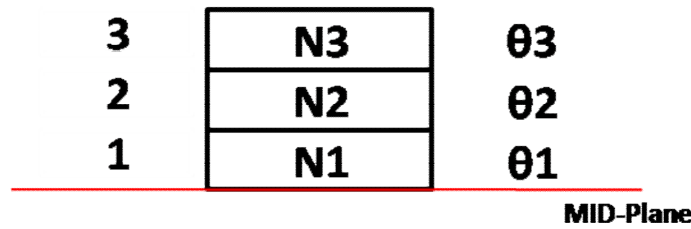


Figure 6-5 Half Laminate Orientation Group Numbering Convention

6.1.2 Numerical and Graphical Optimiser (B)

In a continuous attempt to enhance the Intermediate Example approach for tackling the laminate design optimisation, it has decided to add further tools compared to the Basic Example. In fact both levels, L1 and L2, are based on the same common ground, Matlab script, but employed differently than already used in the previous chapter, 5.1, optimisation approaches.

Graphical Optimisation is still the main mean to obtain the optimum design, but this time is supported by a numerical solution, Kuhn-Tucker (see 4.4.3), and an alternative graphical technique, Lamination Diagram 4.4.2.

The Matlab features adopted in the Basic Example to perform graphical optimisation through the function m-file technique, introduced in Figure 5-4, are not the only command of the programming sequence. In fact it has adopted an alternative Matlab tool much faster and shorter that allows to program inside the main Matlab script without requiring extra files of support. An example of that command, named in Matlab `contour`, is showed below.

Matlab Command:

```
[c2,h2]= contour(x1,x3,g1val,[0 0], 'g-- ');
set(h2, 'LineWidth',2);
```

The above “`contour`” command plots the points of the function `g1val` (previously defined in the code) inside the design space of `x1` and `x3` without the need of recall an external m-file.

On the other hand the function m-file techniques is still utilised for the three – dimensional plots of level 2 necessary to facilitate the interpretation of the objective and constraint functions, expressed in terms of the lamination parameters.

A summary of the different Matlab graphic techniques employed at level1 and level2 is showed in the coming tables, respectively Table 6-2 for 2D plots and Table 6-3 related to 3D plots.

Intermediate Example (2D Plots)	Level1 (L1)	Level2 (L2)
-f : objective function	"contour" command	"contour" command
-g : constraint function	"contour" command	"contour" command
-Lamination Diagram: Parabola	Not Applicable	"contour" command

Table 6-2 - Intermediate Example: Level1 and Level2, Matlab 2D Plots

Intermediate Example (3D plots)	Level1 (L1)	Level2 (L2)
-f : objective function	Not Applicable	obj_Ex_strain.m
-g : constraint function	Not Applicable	inecon2_strain.m
-Lamination Diagram : parabola	Not Applicable	obj_G2.m

Table 6-3 - Intermediate Example: function m-file labels at Level2, 3D Plots

The above tables shows an extra row that was not considered in the Basic Example; the lamination diagram. This is a straight consequence of the level2 lamination parameters design space coming from (Eq. 6-5).

The flowchart viewed in Figure 6-6 is a further outline of the two different optimisation tools, already anticipated in Figure 6-2, named as "B bis1" and "B bis2". Figure 6-6 highlights inside a dash-dotted rectangle line the main difference to the Basic Example.

In fact the left boxes says it has been introduced a different optimisation tool, numerical (Kuhn-Tucker 4.4.3), and then shows how is structured in the Matlab program.

It is important to emphasise this optimisation tool, part of the main Matlab script, applies the FOC, from (Eq. 4-43) up to (Eq. 4-48), and more specifically verifies the four conditions, (Eq. 4-53) and (Eq. 4-54) coming from the Kuhn-Tucker conditions identifying which of those is applicable coherently to the type of equations involved. Then the final results, taking into account the sign of the Lagrange Multiplier according what discussed in section 4.4.3, goes to be compared with an expected similar outcome from the Graphical side. Figure 6-6 shows clearly the possible options after the comparison of the numerical and graphical different optimisation approaches.

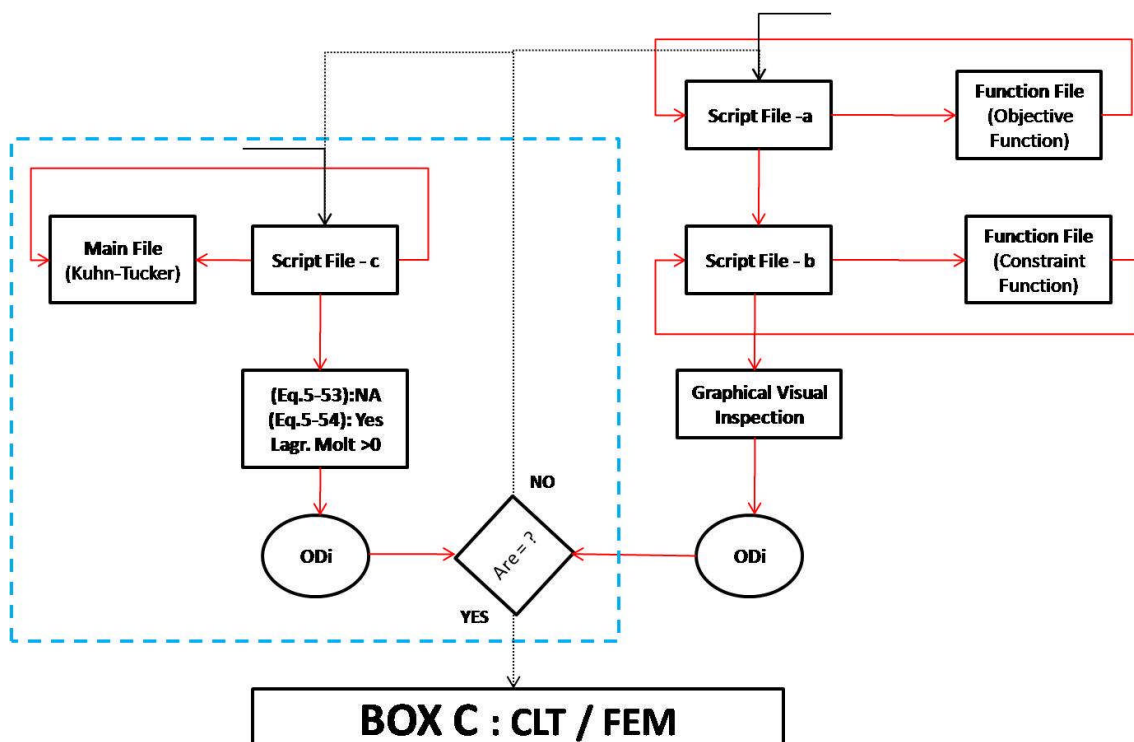


Figure 6-6 - Intermediate Example Graphical and Numerical Optimisers: Matlab Structure

The lack of similarity, not identity considering the lack of precision coming from the visual inspection, of the results of the above figure will require further investigations blocking the data flow to move on the next Box C.

Finally despite the apparent parallel structure of the two components of Box B, chosen just for sake of simplicity, the data flow within the Matlab code is sequential.

The differences of applying numerical and graphical optimisations are described in the following sections.

6.1.2.1 LEVEL 1

The optimisation at this level, as already displayed by Figure 6-6, is obtained by employing two separate tools; numerical and graphical, and performed throughout Matlab script. Then it is expected to compare the two outcomes, the optimum designs, and finally judging their reliability for enabling the data flow to move on the level 2.

That it will be explained through specific laminate configurations and boundary conditions according to:

- (Eq. 5-1)
- (Eq. 5-2)
- (Eq. 5-3)

Since the optimisation technique employed is the same of the Basic Example, to start the two-level decomposition it will be requested to identify starting values. Those are the invariants of the current level, and while in the basic example they were just the two orientation angles of Figure 5-3, here there is an option to select between:

- Three-fixed orientation angle laminate arrangement (3_F_OA): Figure 6-3
- Two- variable orientation angle laminate arrangement (2_V_OA): Figure 6-4

Since the second selection might define a pattern similar to the one already followed in the basic example, it has been decided to go for the three-fixed orientation angle configuration.

This choice, as described by Table 6-1, will define as design variables the total thickness of the laminate plies, t_1 and t_3 according to the convention of Figure 6-3 and as invariants and starting values:

- $\vartheta_3 = 45$
- $\vartheta_2 = 90$
- $\vartheta_1 = 0$
- $t_2 = (0.127) * 4 = 0.508$

The last data t_2 shows one of the developments of this intermediate example; to deal with plies and not just any thickness that in real life applications are not applicable. For such reason the use of 0.127 as realistic thickness unit of a single layer will be a pivotal criteria choice, named “realistic criteria” for identifying the optimum design of the thickness design space.

To make more effective the comparison of numerical and graphical optimisation it has been introduced a further inequality constraint; the shear strain function of level 2. This will make straightforward the visual inspection and easier the selection of the graphical optimum design to compare with the Kuhn-Tucker, 4.4.3, condition of the numerical technique.

The running of the numerical optimiser through Matlab is made possible by the Symbolic Math Tool, which permits symbolic computation integrated in the numerical environment of Matlab [12]. This allows the user to explore problems in calculus, linear algebra, solutions of system of equations, and other areas. The symbolic computation in Matlab performed using a symbolic *object* or *sym*. This is another data type like the number and string data types used in earlier programming. The symbolic Matlab toolbox uses sym objects to represent symbolic variables, expressions, and matrices [12]. For instance the Matlab lines of the numerical optimiser are:

Matlab Command:

```
% symbolic procedure
%-----
% define symbolic variables
format compact
syms A11 A12 A22 A66
syms x1 x3 b1 b2 g1 g2 f F
```

```
A11=x1*Qt1(1,1)+t2_L2*Qt2(1,1)+x3*Qt3(1,1)
```

```

A12=x1*Qt1(1,2)+t2_L2*Qt2(1,2)+x3*Qt3(1,2)
A22=x1*Qt1(2,2)+t2_L2*Qt2(2,2)+x3*Qt3(2,2)
A66=x1*Qt1(3,3)+t2_L2*Qt2(3,3)+x3*Qt3(3,3)

```

```

factorx=Nx/Cons1
factorxy=Nxy/Cons2

```

```

f = x1+x3+t2_L2;
g1 = -A11*A22+A12*A12+factorx*A22
g2 =(factorxy-A66)/A66
F = x1+x3+t2_L2+b1*g1+b2*g2;

```

These show the definition of the multiple objects, not just the design variables, x_1 x_3 , but also all the functions defined in function of them.

Additional symbolic computations have been introduced through code as appropriate to finally solve Kuhn-Tucker conditions given by:

- (Eq. 4-49) and (Eq. 4-50)
- (Eq. 4-53) and (Eq. 4-54)

Mainly the four cases of (Eq. 4-53) and (Eq. 4-54) will be reduced just to “case d”, 4.4.3, as potential reliable solution as showed in Figure 6-7.

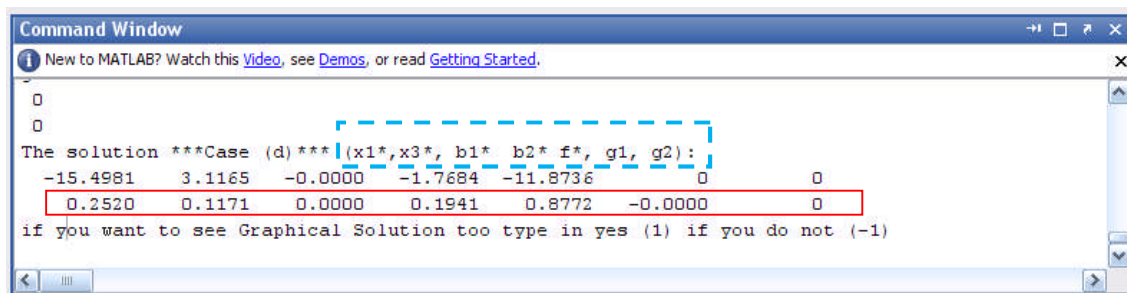


Figure 6-7 – Matlab Command Window: “Kuhn-Tucker Condition” Results

Figure 6-7 expresses the Matlab Command Window results of the optimum design of the unconstrained problem defined by Basic Example equations; (Eq. 5-1), (Eq. 5-2), (Eq. 5-3), and by the similar equations of the Intermediate Example adding the shear strain inequality function g_2 as further constraint equation. In fact, the figure shows as part of the results the normal strain, g_1 , and shear strain, g_2 , outcome. The interpretation of the data result of Figure 6-7 is highlighted by the solid rectangle line. These numbers

follow the order of the labels inside the dashed rectangle line. Therefore, the optimum design according to the numerical optimiser is:

- $t_1 = 0.2520 \text{ mm}$
- $t_3 = 0.1171 \text{ mm}$

The above optimum result is expected to be reliable as the Lagrange Multipliers, b_1 and b_2 in Figure 6-7, of the unconstrained problem are non negative in a search process of a minimum [12]. Moreover the constraint function equations, g_1 and g_2 , written as follows:

- $g_1 = -A_{11} \cdot A_{22} + A_{12} \cdot A_{12} + \text{factorx} \cdot A_{22}$; $g_2 = (\text{factorxy} - A_{66}) / A_{66}$
- $\text{factorx} = N_x / \text{Cons1}$; $\text{factorxy} = N_{xy} / \text{Cons2}$

(Loads, N_x and N_{xy} , and inequality constraint values, Cons1 and Cons2 , are the same of the Basic Example already described.)

are minor or equal to zero satisfying the inequality conditions. Finally the design variables are both positive numbers.

The next step is to prove the Graphical Optimiser at this level will provide a very similar optimum design to the one of t_1 and t_3 just stated above.

Figure 6-8 describes graphically the level 1 optimisation problem. The labels “ t_3 ” and “ t_1 ” refer to the design variables mentioned earlier, as thicknesses of the whole, not like the BE half laminate, symmetric and balanced laminate under consideration.

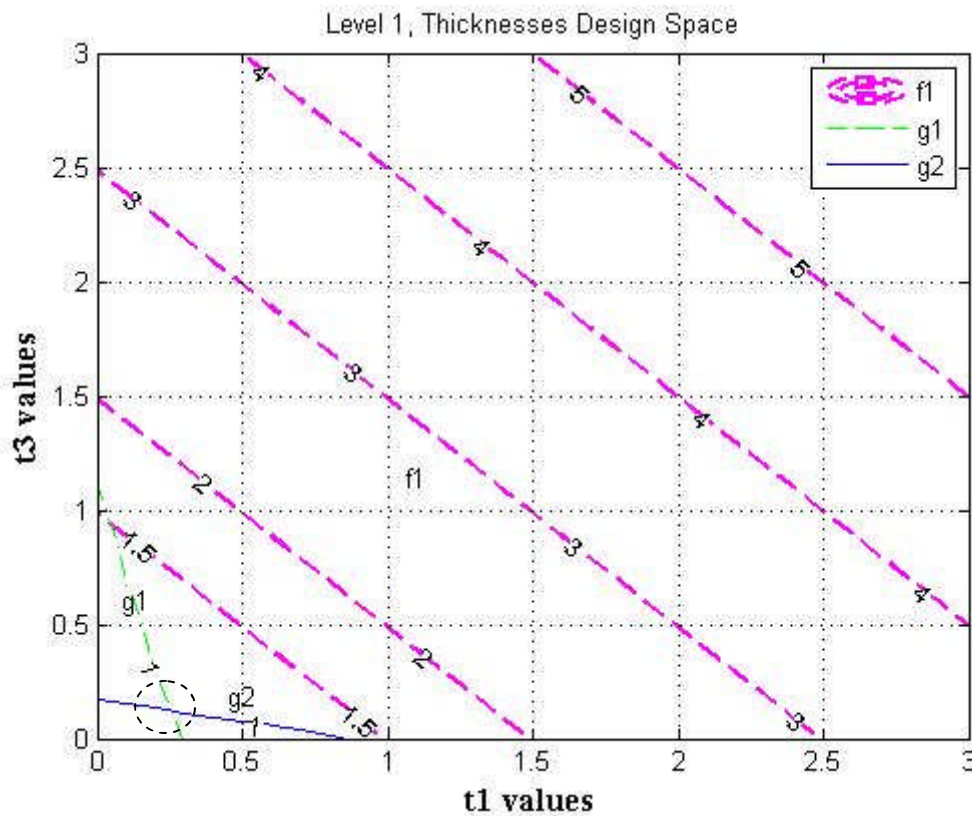


Figure 6-8 – Intermediate Example: Design Space Level 1

The x axis represents t_1 , the thickness of the first layer group from the mid-plane and the vertical axis, t_3 is the thickness of the first ply group from the top and bottom, all these quantities are expressed in millimetres.

As mentioned before in this plot, there are two inequality constraints; the normal strain, g_1 , is the dash-dotted line and the shear strain, g_2 , is the solid darker line. The former has to be lower than zero to produce a normal strain not greater to 0.4%. On the other hand the latter is defined to have shear strain smaller than 0.6%. The pareto solution will come out from the intersection of these two constraint lines and the minimum objective function f_1 passing through that point will characterise the smallest thickness laminate line among the several labelled displayed contours (Figure 6-8).

Zooming in the area of interest of Figure 6-8, highlighted by the circle, will bring a better understanding of the optimum design searched.

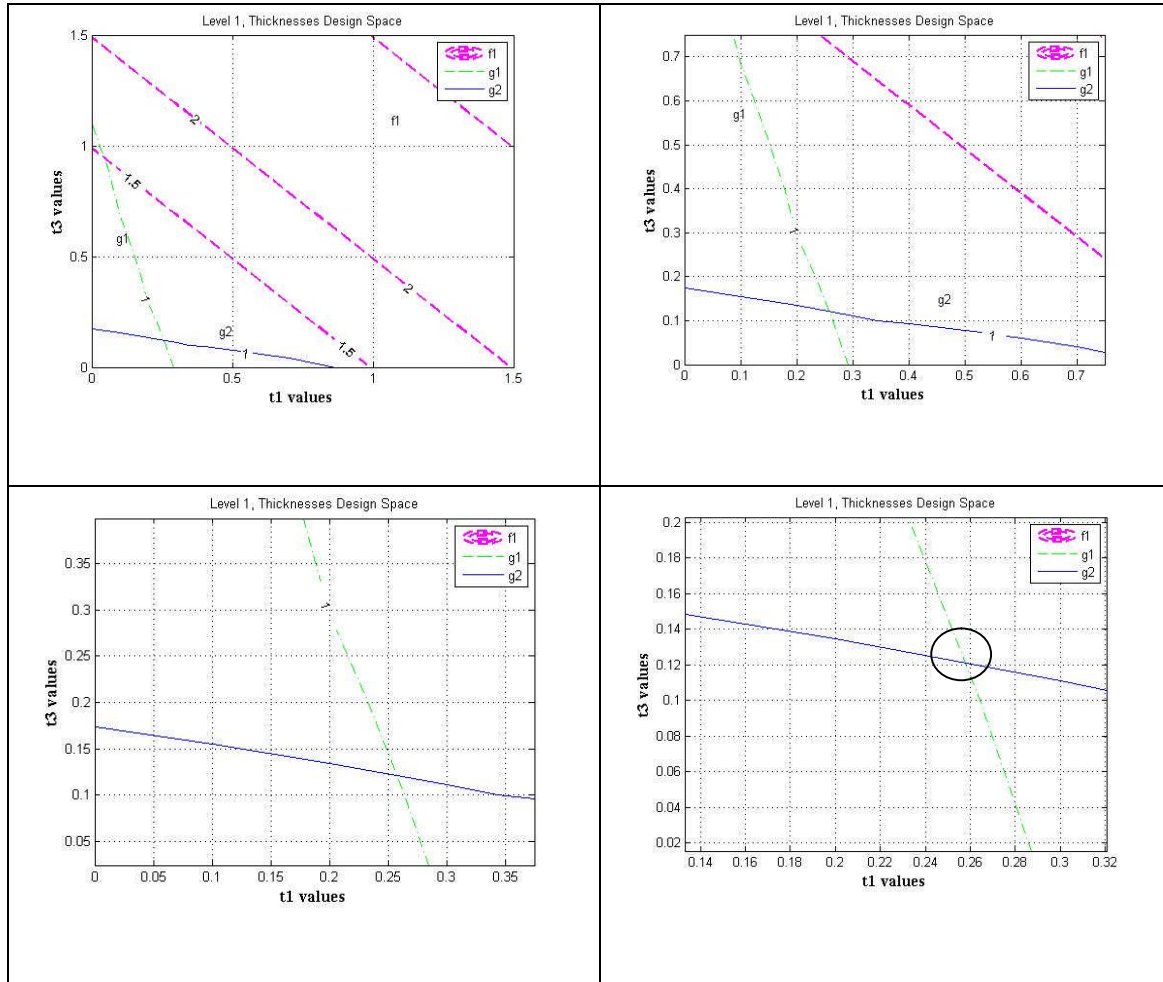


Figure 6-9 – Intermediate Example, Level1: Zoom in steps to the Optimum Design

Figure 6-9 shows four main steps of the magnifying process about the intersection of the constraint curves. The bottom right plot proves the numerical optimum design of $t_1 = 0.2520\text{mm}$ and $t_3 = 0.1171\text{mm}$ is matching well with the graphical optimisation visual inspection outcome.

Once proved the reliability coming from numerical and graphical optimisers and learned a further tool, Kuhn-Tucker condition performed through Matlab Symbolic Algebra, in the search of the optimum design it is needed to make additional considerations on the coherence of the optimum design of the thickness design space. In fact, the definition of the design variables t_1 and t_3 correspond respectively to $N_1 = 1.984$ and $N_3 = 0.922$, overall number of plies for each orientation angle groups. The obvious integer solution is obtained by rounding up both numbers to $N_1 = 2$ and $N_3 = 1$ which corresponds to $[(45)_{0.5}/(90)_2/(0)]_S$, using the terminology introduced in section 3.5. That is not a

real symmetric laminate arrangement and mainly establishes that the numerical technique is not as flexible as the graphical optimiser for this specific application. In fact, coherently to the search of an optimum design which still maintains the symmetric characteristics, using the graphical optimiser it is possible to choose inside the feasible domain, the hashed side of Figure 6-10 indicates the infeasible domain, as optimum design a pair of values that will correspond to a symmetric configuration. This point of the feasible domain of Figure 6-10 is characterised by the following numbers:

- $t_{1OD} = (0.127) * 4 = 0.508$
- $t_{3OD} = (0.127) * 8 = 1.016$

These two optimum design (OD) values will lead finally to a symmetric and balanced laminate configuration as the one below:

- $[(45)_4 / (90)_2 / (0)_2]_S$

So, from now onwards the shear strain function, g_2 , drawn in Figure 6-8 and Figure 6-9 will not be anymore taken into account in the next Intermediate Example steps since was just useful mean to explain numerical and graphical results.

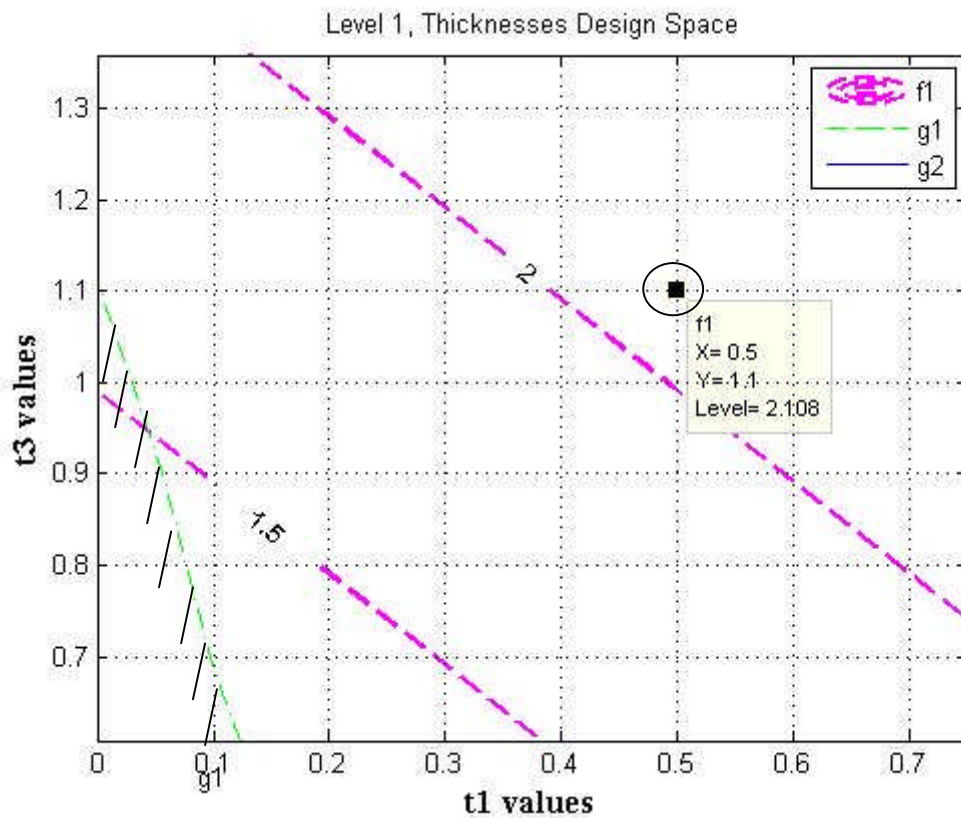


Figure 6-10 – Intermediate Example, Level 1: Feasible Domain

Now the optimum design highlighted in Figure 6-8 will be passed on the next level 2, to be employed inside the design space of the lamination parameters.

6.1.2.2 LEVEL 2

Unlike the Basic Example, the Intermediate Example at level 2 will deal with a graphical optimiser working within a design space that is indirectly linked to the orientation angles of the laminate under investigation. In fact, as mentioned earlier in Table 6-1, the idea now is to introduce the potentialities of the lamination parameters already described in detail in section 4.4.2.

The description of the feasible space of the lamination diagram described by the two design variables, $V_1^*V_3^*$, according to Miki's procedure [66], requires a step by step process to introduce and clarify each main curve involved later on in the visual inspection.

First, the main region outlined by the lamination parameters is the parabola described by (Eq. 4-9), and displayed through the dark thick solid line of Figure 6-11.

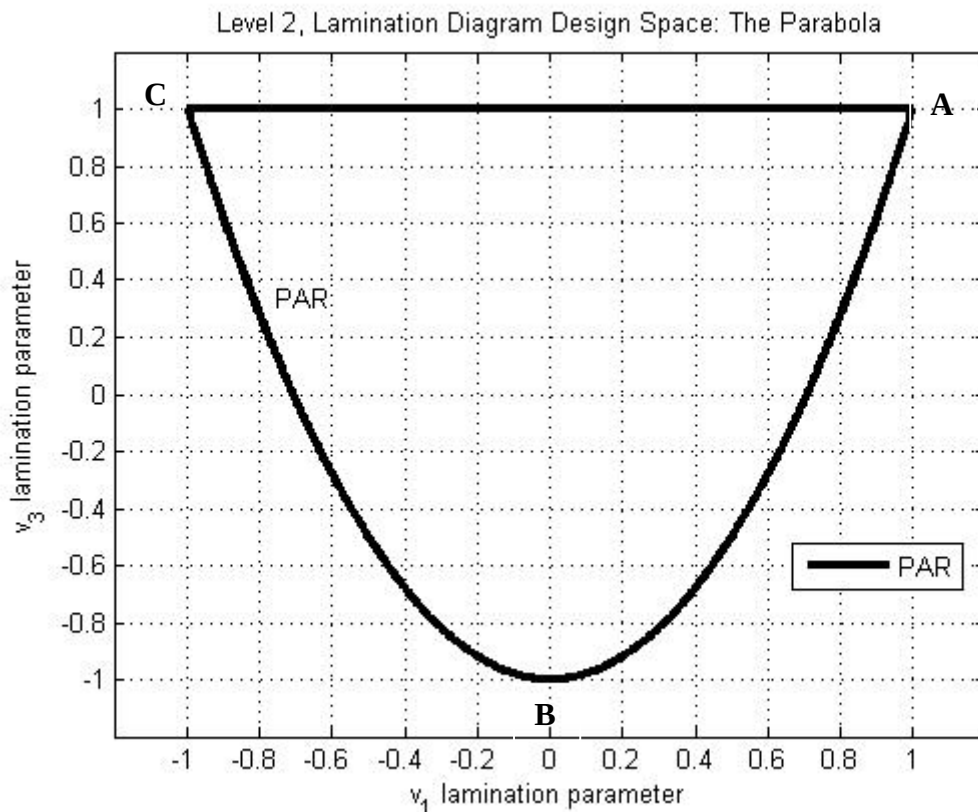


Figure 6-11 – Intermediate Example, Design Space Level2: The Parabola

Values of all possible combinations of the lamination parameters are, therefore, located along the boundary curve ABC in Figure 6-11. Note that points A, B, and C correspond to laminates with $[0]$, $[\pm 45]_S$, and $[90]$ orientation angles, respectively [1]. If the laminate consists of two or more fibre orientations, then it can be referred to the inequality of (Eq. 4-10), that describes no more a curve but a region. In fact, (Eq. 4-10) marks out the allowable region of the lamination parameters as the area bounded by the curve ABC, irrespective of the number of different layer orientation angles. Any point

inside the lamination parameter diagram must correspond to laminates with two or more fibre orientations. As already adopted in the Basic Example, a thorough understanding of the 2D plot of Figure 6-11 might be done employing 3D plots as specific request of the user.

Next curve to be introduced within the design space of Figure 6-11 is the normal strain objective function. This curve, unlike the parabola plot that is an invariant, showed in Figure 6-12 by dashed thin line contours is evaluated using the optimum design coming from level 1.

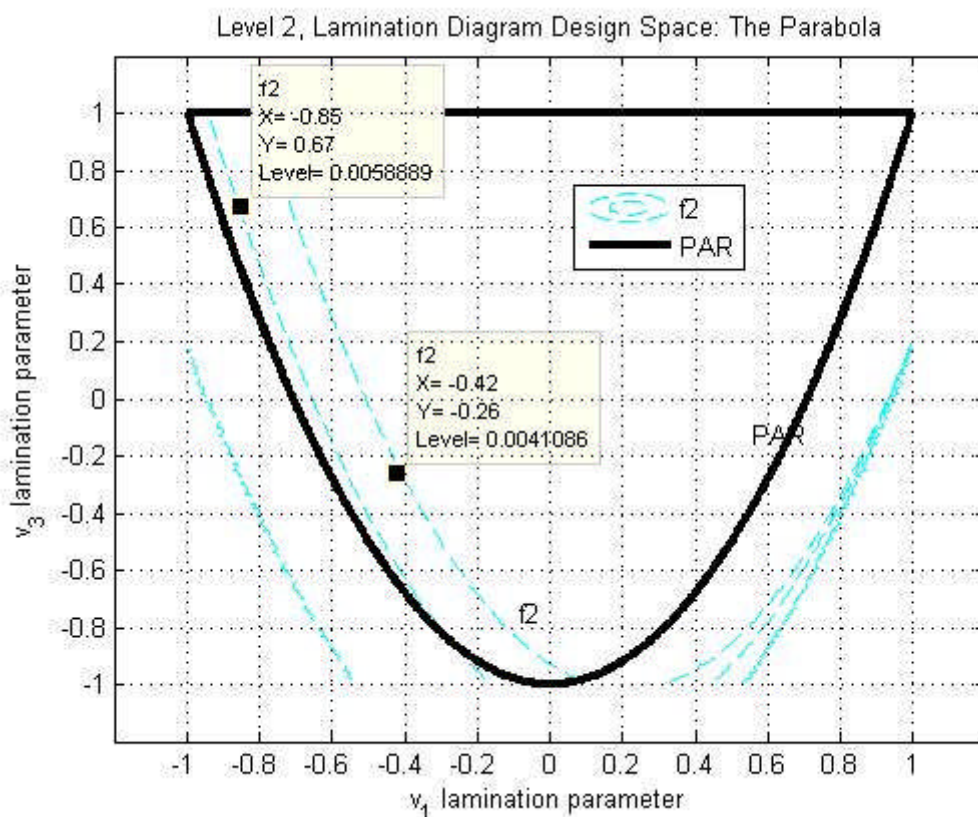


Figure 6-12 – Intermediate Example, Design Space Level2: Parabola and Normal Strain Objective Function

The normal strain contours plotted in Figure 6-12 with the Cursor Matlab aid feature, already employed in the Basic Example, describe a feasible space considering that the normal strain is meant not to be greater than 0.004 micro-strains. According this information the allowable area of the lamination parameters will be outside the hash marks drawn in Figure 6-13, bounded by the superior and right side of the Parabola and

the maximum objective function, f_2 , at the left. In fact it is import to repeat that for convergence reason the objective function of level 2, Table 6-1, is expected to be maximised.

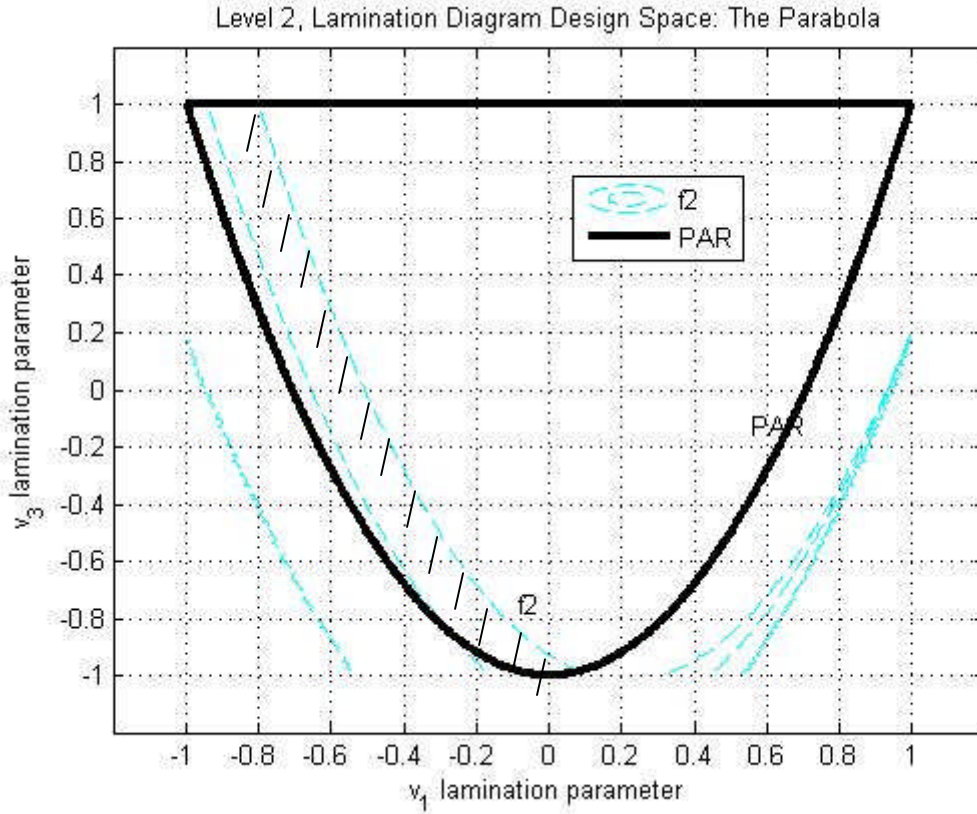


Figure 6-13 – Intermediate Example, Design Space Level2: Parabola and Normal Strain Objective Function feasible domain

The X and Y axes of Figure 6-12 are the lamination parameters as described by the labels whilst the Z axis is the normal strain f_2 equation, expressed in terms of the lamination parameters by the following:

$$\varepsilon_x = \frac{N_x}{hE_x^{eff}} = \frac{N_x}{h} \left(\frac{U_1 - V_1^* U_2 + V_3^* U_3}{U_1^2 - U_4^2 + 2U_1 U_3 V_3^* + 2U_4 U_3 V_3^* - U_2^2 V_1^{*2}} \right) \quad (\text{Eq. 6-9})$$

This equation is obtained combining the basic normal strain definition of (Eq. 4-20), with the Effective Elastic Modulus of (Eq. 4-16) and taking into account the material invariants of (Eq. 3-46).

For completing the lamination diagram design space of level 2 it is necessary to introduce the curves of the constraint function, g_2 .

The lamination diagram design space in Figure 6-14 is perfected by the dash-dot line of the shear strain constraint function contours. These horizontal lines are drawn applying the following shear strain equation, expressed in terms of the lamination parameters $V_1^* V_3^*$:

$$\gamma_{xy} = \frac{N_{xy}}{hG_{xy}} = \frac{N_{xy}}{h} \left(\frac{1}{U_5 - U_3 V_3^*} \right) \quad (\text{Eq. 6-10})$$

That equation (Eq. 6-10) is obtained putting together the Effective Shear Modulus, (Eq. 4-19), with the shear strain basic definition of (Eq. 4-22) and applying material invariant equations (Eq. 3-46).

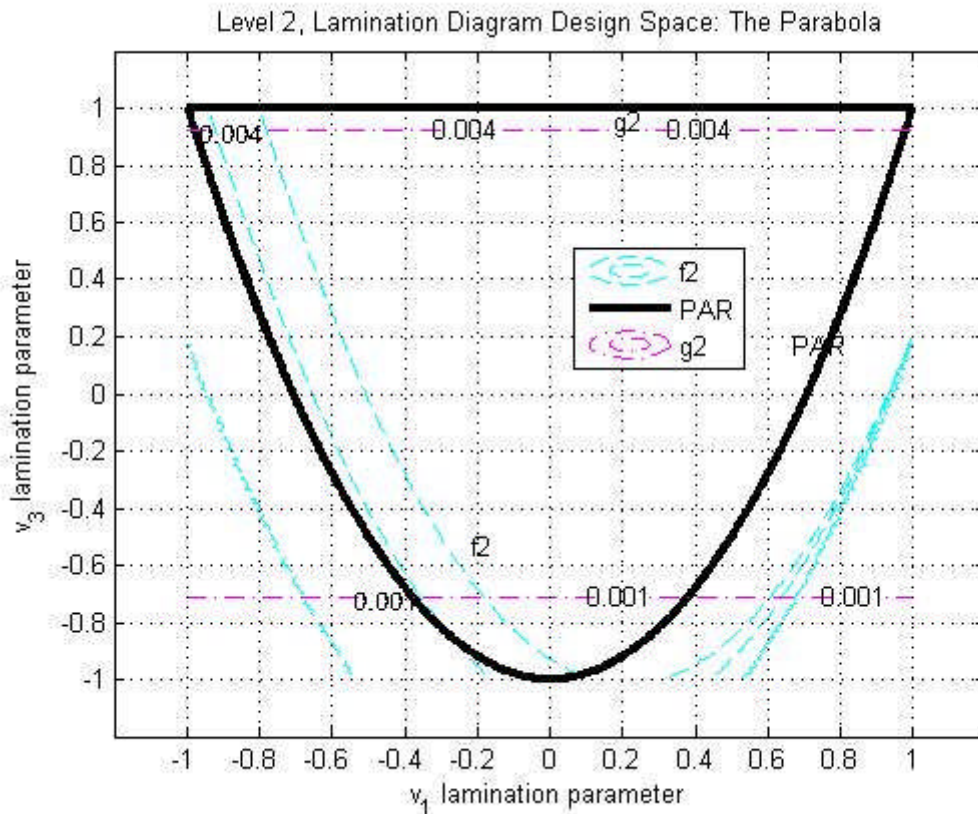


Figure 6-14 – Intermediate Example, Design Space Level2: Parabola, Normal Strain Objective Function, and Shear Strain Constraint Function, feasible domain

Figure 6-14 describes two specific strain contours, the first from the top labelled by 0.004 micro-strains, and then at the very bottom of the diagram a 0.001 micro-strains. These values help to define a shear strain value trend inside the main design space of the parabola. In fact all constraint functions with micro-strains greater than 0.004 tend to go out the parabola domain from its top edge. This is relevant to this analysis since the shear strain inequality condition, Cons2, says not to overcome the 0.006 micro-strains. Therefore the feasible design space at this first iteration of the laminate design cycle is still confirmed to be the one Figure 6-14.

Considering that the Intermediate Example offers at this stage the choice of two different laminate configurations; 2-variable orientation angles (2_V_OA), Figure 6-4, and 3 fixed-orientation angles (3_F_OA), Figure 6-3, the feasible domain of Figure 6-4 will be managed differently.

With the former configuration, 2_V_OA, the search of the optimum design will lead to identify any lamination parameter points inside the feasible region of Figure 6-4 and

then applying (Eq. 4-14) and (Eq. 4-15) identify ϑ_1 and ϑ_3 to be passed to the level 1 next iteration. Where a new design cycle will begin with ϑ_1 and ϑ_3 new invariants and t_2 equals to zero since dealing with two variable orientation angles lay out.

The latter laminate arrangement, with predetermined orientation angles, allows introducing further useful features of the laminate parameters design space.

Miki and Sugiyama [66] showed that the laminate diagram can be used for designing laminates with fixed orientation angles. They showed that the feasible region for such laminate is a polygon with vertices located on the envelope of the lamination parameter diagram. If the optimum design is on the perimeter of the diagram, the laminate is an angle-ply laminate with one fibre orientation, as the point A, B and C showed in Figure 6-11. Therefore, given a set of permissible integer orientation angles, vertices of the polygons are placed at those locations on the perimeter of the diagram that correspond to the selected angles. For instance, all angle-ply laminates with orientations limited to integer multiples of 45° are showed by Figure 6-15.

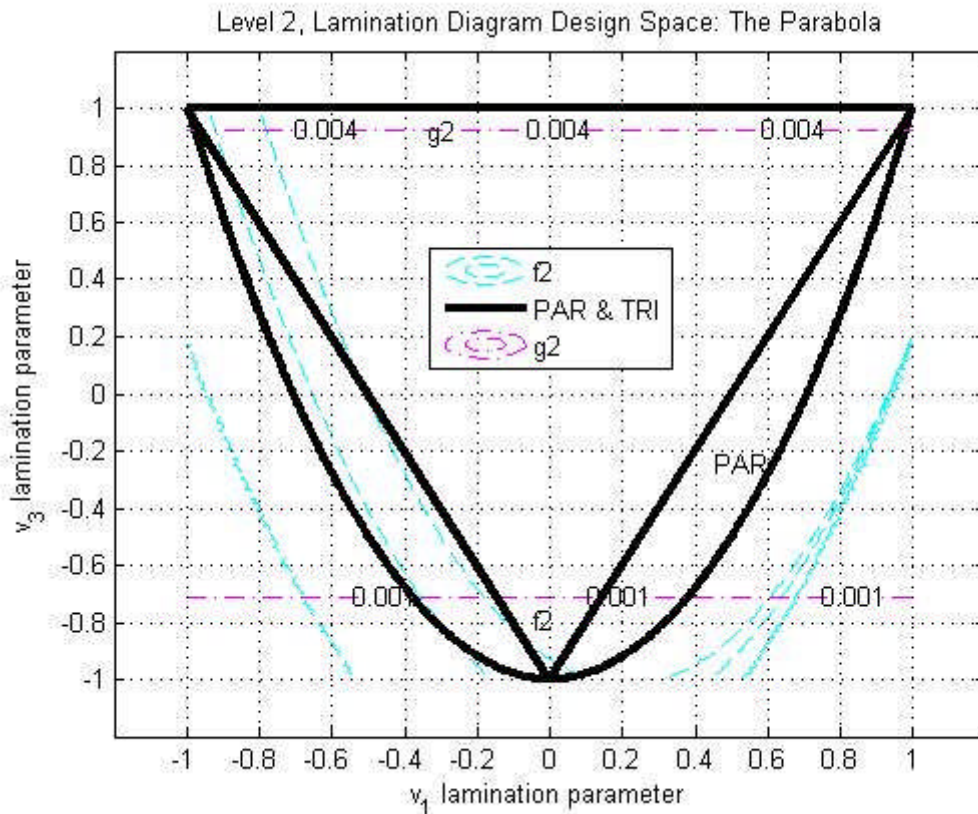


Figure 6-15 – Intermediate Example, Design Space Level2: Three-fixed orientation angle configuration, The Triangle

The triangle displayed in the above Figure 6-15 by solid thick lines, represents the edge of the new feasible space of a laminate made of only $\vartheta_1 = 0^\circ, \vartheta_2 = 90^\circ$, and $\vartheta_3 = \pm 45^\circ$ layers according to Figure 6-3 convention scheme. For points inside the triangle, a procedure similar to the one introduced by Miki, [1] [66], and already described by (Eq. 4-11) can be used to determine the possible laminates of Figure 6-3. So the number of lamination points along the triangle edges and interior points of it corresponding to laminates with orientation angles of $0^\circ, 90^\circ$ and 45° are displayed in Figure 6-16 by the intersection points of the thin solid lines inside the triangle. Considering Figure 6-16 has been built up with the data of $t_{1,OD} = (0.127) * 4 = 0.508$, $t_{3,OD} = (0.127) * 8 = 1.016$ and $[(45)_4 / (90)_2 / (0)_2]_S$ from level 1, it corresponds to a laminate made of total number of layers as N equals to 16. Then the nodes and vertices on and inside the triangle describe the half laminate configurations available. The equally spaced segments, ESS, on the edge of the triangle are N/2, eight in this specific case, and the horizontal parallel lines

moving from the top side of the parable to the very bottom of it are the division number, DN, of the laminate diagram, defined by $N/2$ too.

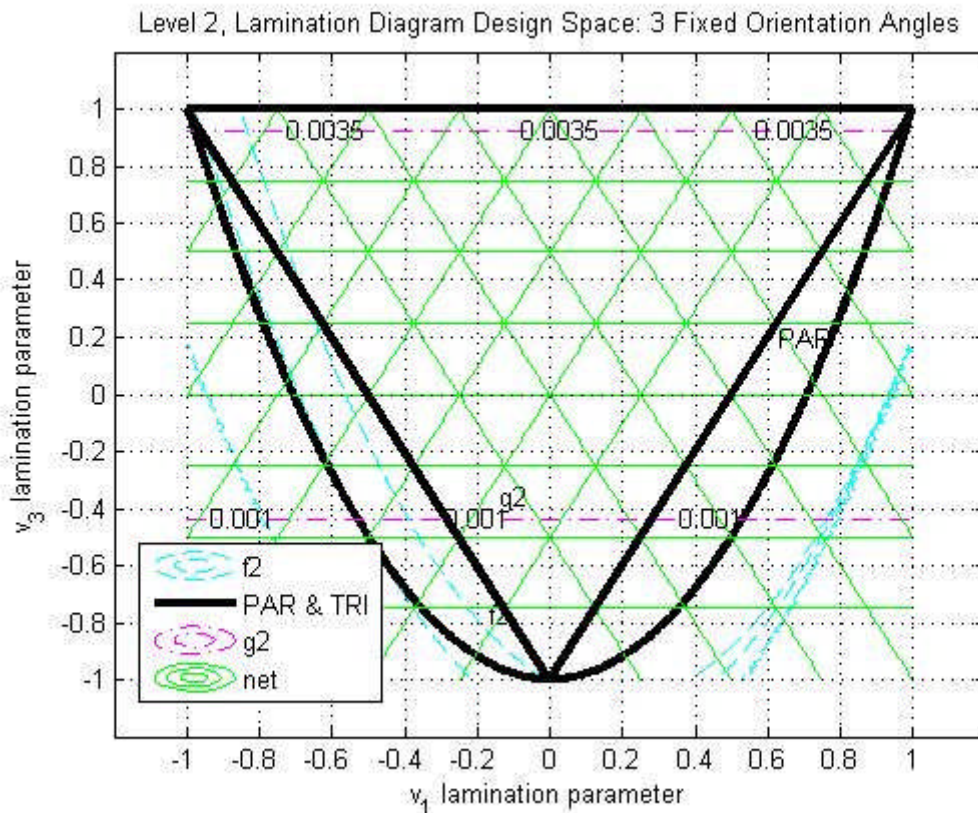


Figure 6-16 – Intermediate Example, Design Space Level2: Laminates with combinations of three fixed orientation angles (0° , 90° , $\pm 45^\circ$)

These two quantities have been used as two input values requested by the main Matlab script to characterise the laminate configuration of an optimum design.

In fact ESS and DN showed clearly by Figure 6-17, for a laminate of 16 plies according to the level 1 optimum design are the only values needed to define uniquely a node of the Figure 6-15 feasible domain. Figure 6-17 focused on making comprehensible the evaluation of the optimum design for a predetermined orientation angle laminate. Among all the internal nodes of the thick solid line triangle it has been pointed out one, identified by a black dot next to a label OD, as optimum design. The characterisation of this OD point requires the vertical position inside the design space, given by the DN number. Then a horizontal measure is necessary to complete the location of the node. That is provided by a ESS_L, left equivalent spaced segment along x axis, that counts the number of ESS on the left of the OD point.

In such contest the OD point of Figure 6-17 is recognised by the following quantities:

(Eq. 6-11)

- $ESS_L=1$
- $DN=3$

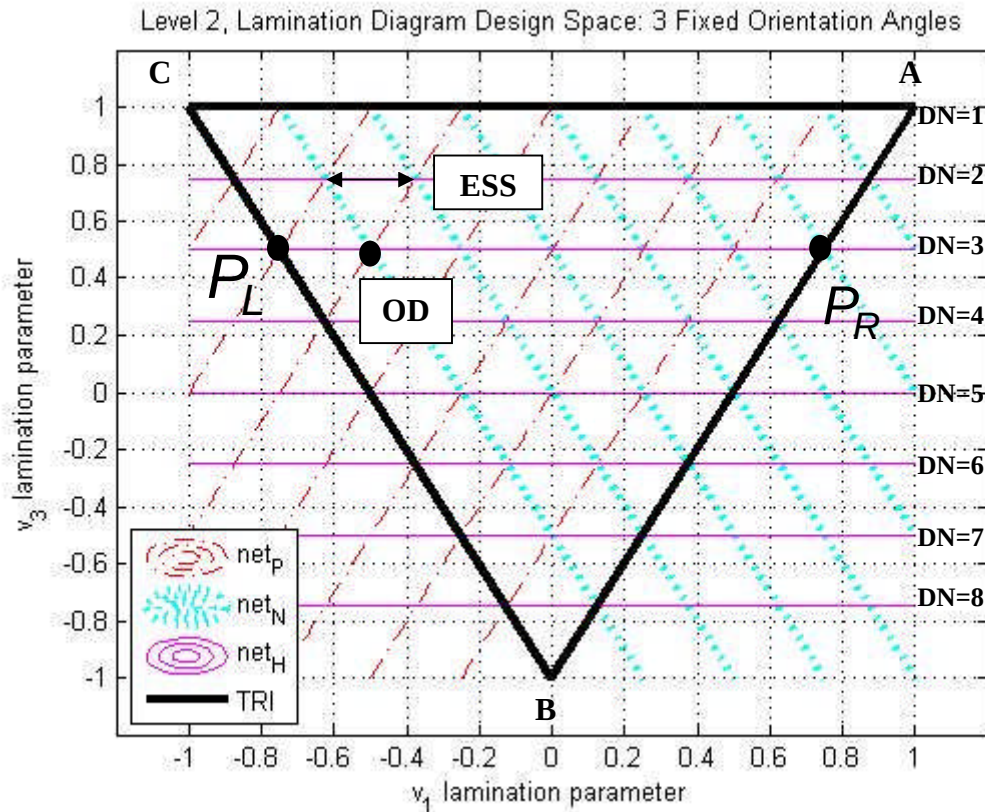


Figure 6-17 –Design Space Level2, Optimum Design Choice: Equal Spaced Segment (ESS) and Division Numbers (DN).

The reason of choosing this specific dot as Optimum Design is coming from the attempt to select a laminate configuration, inside the feasible domain, as such to have as much as possible a higher normal strain (Figure 6-15), to satisfy the level 2 objective function of maximising the normal strain (this is the reason the OD is on the left instead being picked on the right of the feasible space), and to still offer a sufficient laminate thickness to be reduced at the next iterations (that explains why the OD was not chosen at the bottom of design space).

Finally, Figure 6-17 enables the user to point out the OD and so to type in the Matlab Command Window the two quantities of (Eq. 6-11), then a specific algorithm based on the rules of (Eq. 4-11) will provide the final optimum design laminate configuration of this first design cycle.

Before moving on it is worth describing the main numerical steps that will lead from the (Eq. 6-11) to the final ply arrangement to be passed on the second iteration at level 1.

The goal to characterise the ply arrangements of the optimum design, OD, in Figure 6-17, through a volume fraction vector P_{OD} , is achievable solving the next equation based on the Miki's principles of (Eq. 4-11) [1],[66]:

(Eq. 6-12)

$$P_{OD} = ESS_L * P_R + ESS_R * P_L$$

Where:

- ESS_L and ESS_R are left and right distance from the OD to the horizontal points on the edge of the triangle
- P_R and P_L are volume fractions of two discrete points laying on the same line of the optimum design (horizontally)

The equivalent spaced segments either left or right, are known values from visual inspection. On the other hand the two left and right volume fractions need following two equations to be computed.

(Eq. 6-13)

- $P_L = ESS_B * P_C + ESS_T * P_B$
- $P_R = ESS_B * P_A + ESS_B * P_B$

Where:

- $ESS_T = \overline{CP_L}$ and $ESS_B = \overline{BP_L}$ coherently to the labels of Figure 6-17
- $P_A, P_B, and P_C$ are volume fractions of three discrete points laying on the edge of the parabola and triangle feasible domain.

The equivalent spaced segments just defined are given by visual inspection again. They are defined in the specific case of Figure 6-17 as:

$$\begin{aligned} & \bullet \text{ ESS}_T=2 \\ & \bullet \text{ ESS}_B=6 \end{aligned} \quad (\text{Eq. 6-14})$$

Regarding the three volume fractions $P_A, P_B, \text{ and } P_C$, they are already known volume fraction since introduced Figure 6-11. In fact, they correspond respectively to the following volume fraction vectors:

$$\begin{aligned} & \bullet P_A = [0]_S \\ & \bullet P_B = [45]_S \\ & \bullet P_C = [90]_S \end{aligned} \quad (\text{Eq. 6-15})$$

Applying (Eq. 6-14) and (Eq. 6-15) to (Eq. 6-15) :

$$\begin{aligned} & \bullet P_L = 6 * [90^\circ]_S + 2 * [45]_S \\ & \bullet P_R = 6 * [0^\circ]_S + 2 * [45^\circ]_S \end{aligned} \quad (\text{Eq. 6-16})$$

These two known equations, plus (Eq. 6-15), and $\text{ESS}_R=5$ applied to the main equation (Eq. 6-15) give:

$$\bullet P_{OD} = 30 * [90^\circ]_S + 10 * [45^\circ]_S + 2 * [45^\circ]_S + 6 * [0^\circ]_S \quad (\text{Eq. 6-17})$$

After further simplification steps it is obtained:

(Eq. 6-18)

$$\bullet P_{OD} = 6(2 * [45^\circ]_S + 5 * [90^\circ]_S + [0^\circ]_S)$$

The above equation provides finally the optimum design at first “level 2” design cycle of this intermediate example corresponding to the below laminate configuration:

(Eq. 6-19)

$$\bullet [(45)_2 / (90)_5 / (0)]_S$$

The particularity of the Matlab algorithm designed to reach (Eq. 6-15) just requiring the two input values, ESS_L and DN, of (Eq. 6-15) is established on the following step by step considerations for a general laminate made by N_{tot} as total number of layers.

(Eq. 6-20)

1. The program evaluates the total number of division numbers:

$$DN_{tot} = \frac{N_{tot}}{2}.$$

2. The user chooses an integer number from one to DN_{tot} . This value identifies one of the horizontal lines in Figure 6-17 on which will lay the discrete point, the black spot, as potential optimum design. This value, labelled DN, has to be an odd number to satisfy the symmetric and balanced initial requirements.
3. The program evaluates the total number of equal segment spaced laying on the division number chosen, $ESS_{(DN)}$, through the following:

$$ESS_{(DN)} = \frac{N_{tot}}{2} - (DN - 1)$$

4. The user types in the left equal spaced segment, ESS_L.
5. The program calculates the right equal spaced segment applying:

$$ESS_R = ESS_{(DN)} - ESS_L$$

6. The program is ready to compute the top and bottom equal spaced segments:

$$ESS_T = DN - 1 ; ESS_B = \frac{N_{tot}}{2} - ESS_T$$

7. Taking into account all the previous steps the program is set to determine (Eq. 6-13) and at the end the optimum design of (Eq. 6-12).

The described algorithm shows that for any symmetrical laminate configuration, N_{tot} is expected to be an even number, once defined the division number and a left equal spaced segment, quantities needed to locate the discrete point of the optimum design, and then automatically evaluates all the left above equations up to the last pivotal (Eq. 6-12). Useful consideration and double check applicable at the end of each level2 iteration plot is to count the ESS at DN=1. In fact, applying the equation at the above 3 bullet point, the equal spaced segment of the first division number, Figure 6-16, are expected to be equal to half ply laminated number.

6.2 Basic Example and Intermediate Example: Similarities and Differences

Over the last paragraphs, describing the main steps of the intermediate example, it has already been adopted the tendency to point out main differences, most of the time enhancements, between the two examples.

But now it has been tasked to highlight main characteristics of diversity.

Despite the same multilevel decomposition structure might lead to believe no enhancement has been taken into the optimisation technique itself. That has been one of the main directions of improvement and it is going to be proved bringing up the main coupling quantities moving to one level to the other. Next tables will make easier such comparison between Basic Example and Intermediate Example.

2-Level Decomposition , Basic Example		
Level 1	Invariants	Optimum Design
	(g_1, g_2)	(t_1, t_2)
Coupling Quantities	(t_1, t_2)	
Level 2	Invariants	Optimum Design
	(t_1, t_2)	$(g_{1,new}, g_{2,new})$
Coupling Quantities	$(g_{1,new}, g_{2,new})$	
<i>!End of i-iteration, data goes to Level1 to start next iteration</i>		

Table 6-4 – Basic Example: Coupling Quantities Structure

2-Level Decomposition, Intermediate Example				
Level 1	Invariants		Optimum Design	
	3-Fixed OA	2-Variables OA	(t_1, t_3)	
	$\begin{Bmatrix} t_2 \\ \vartheta_1 = 0^\circ \\ \vartheta_2 = 90^\circ \\ \vartheta_3 = \pm 45^\circ \end{Bmatrix}$	$\begin{Bmatrix} t_2 = 0 \\ \pm \overline{\vartheta}_1 \\ \pm \overline{\vartheta}_3 \end{Bmatrix}$		
Coupling Quantities	$\left\{ N_{tot} = \frac{1}{0.127} (t_1 + t_2 + t_3) \right\}$			
Level 2	Invariants		Optimum Design	
	3-Fixed OA	2-Variables OA	3-Fixed OA	2-Variables OA
	$\{N_{tot}\}$	$\begin{Bmatrix} N_{tot} \\ t_2 = 0 \end{Bmatrix}$	$\begin{Bmatrix} DN \\ ESS_L \end{Bmatrix}$	(V_1, V_3)
Coupling Quantities	3_F_OA	$\{t_{2,new} \quad \vartheta_1 = 0^\circ \quad \vartheta_2 = 90^\circ \quad \vartheta_3 = \pm 45^\circ\}$		
	2_V_ OA	$\{t_{2,new} = 0 \quad \pm \overline{\vartheta}_{1,new} \quad \pm \overline{\vartheta}_{3,new}\}$		
<u>!Once chosen the laminate configuration, (3_F_OA) or(2_F_OA), data goes to level 1 for next iteration.</u>				

Table 6-5 – Intermediate Example: Coupling Quantities Structure

Basically these tables break down the data flow through a generic iteration outlining at each step the main characteristics of a multilevel decomposition technique. The necessity to link the first level to the second and eventually vice versa is made possible

choosing specific quantities that can be coupled. This ability enables the process to continue iteration after iteration its main task to narrow a search area going as much as possible closer to a unique optimum design solution. In such scenario it is well understood the importance of the coupling quantities as responsible of the flexibility of the whole process in converging to the searched solution faster and with high precision. The Basic Example, summarised by Table 6-4, is a very straightforward coupling process. In fact, the coupling quantities are no more than the design variables, or optimum design results, of the earlier level. The whole iteration terminates at level 2 with a specific couple of design variables, orientation angles, identified as the optimum design. The label new has been introduced to avoid any misunderstanding with the invariants of level 1 still orientation angles. This process does not offer any real control, or any real quantities to be monitored, that might suit better a real life application. Quantities as total number of laminate layers, predefined orientation angles are definitely improvement required in a natural laminate design optimisation development. Then the option, at anytime of the design cycle, to swap from one configuration, 3-fixed orientation angles, to another, 2 variable orientation angles, is something that make the Intermediate Example potentially more exhaustive and flexible than the previous. These characteristics are mixed up properly according the information provided in Table 6-4. According to this table the first level couple quantity is not anymore just a thickness expressed in millimetres but a more indicative data as the total number of plies in the laminate under investigation. $\{N_{tot}\}$, obtained earlier from **Error! Reference source not found.** and **Error! Reference source not found.**, is then passed to the lamination diagram where will be used draw the “net” contour lines of Figure 6-16 and Figure 6-17 required to identify the discrete points as potential laminate configuration. At that point comes out the user choice to deal with two-variable or three-predefined orientation angle arrangement as showed clearly by the several choice columns in the table. Then, as depth description of the previous paragraph, the optimum design is a result of the two quantities of (Eq. 6-11), ESS_L and DN, for a 3-fixed orientation angle option. On the other hand the 2-variable orientation angle requires the identification of a point within the feasible domain of Figure 6-17 defined by the two lamination parameters (V_1, V_3) .

Looking at the coupling quantities of level 2 is easy to see how the next iteration at level 1 will need to know the configuration chosen earlier, here the configuration to use is not

anymore an option but a constraint brought back from the previous level and so defined as invariants, and with this information defines the new design space as the one in Figure 6-17. Important to highlight the sole quantity invariant in a whole iteration is the coupling data of level 1 $\{N_{tot}\}$, whilst all the other quantities, as in the basic example table, not to get confused are labelled by the subscript “new”.

These two tables, Table 6-4 and Table 6-5, describe well alongside the previous sections, 5.1.2 and 6.1.2, the way the whole multi-level decomposition flows but do not make clear enough how it stops.

This stopping process or better identified as “g1f2” convergence process has already anticipated by the highlighted boxes in the tables of Figure 5-1 and Figure 6-1.

In fact the main characteristic of the multi-level decomposition is to break up a complicate optimisation problem in several simpler serialised problems as showed by Figure 6-18.

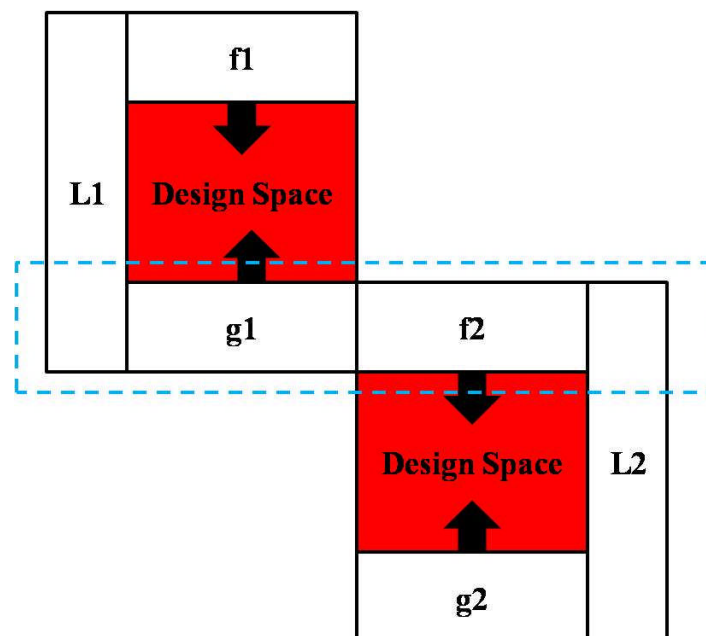


Figure 6-18 – Multi-level Decomposition: The “g1f2” Convergence Process

The above Figure 6-18 says that the flow data will go to an end when the design space at each level will shrink down. This will be possible when the objective functions, $f1$ and $f2$, and the constraint inequality functions of each level, $g1$ and $g2$, will act together to define a feasible region narrowed through the increasing numbers of the iterations. The main difficulty is to harmonise this process making smaller the design space of

each level simultaneously. This harmonisation process is made possible defining correct design variables but mainly making the constraint function of the first level the objective function of the second level and then choosing to maximise it. In such way the design space at level 2 has both constraints of the whole process. That will make this design space the one that controls most the entire optimisation process. This is highlighted in Figure 6-18 by the dashed rectangle line. In fact the choice of putting g_1 equals to f_2 has been a maintained characteristic of both examples.

Finally, instead describing again CLT/FEM (5.1.3), Validation Test (5.1.4), and Optimum Design (5.1.6) boxes for the Intermediate Example, since they are very similar to the Basic Example ones as the just discussed Convergence Process structure; it will be more convenient focusing on the numerical application left with the level 2 optimum design of equation (Eq. 6-19).

For completeness, at this stage it is worth showing the Intermediate Example Matlab Code Structure, as already done with the Basic Example in section 5.1.6.2. Using the same terminology and approach highlighted in Figure 5-40 the following flowcharts will describe thoroughly the Level 1 and Level 2 Intermediate Example pointing out the external re-called files (Matlab, Text, and Nastran), user input data requests, main routines, relevant actions and inter-dependences.

Figure 6-19, describes the Intermediate Example Level 1 structure and highlights the main differences comparing it to the same level 1 Basic Example Matlab structure flowchart of Figure 5-41. In fact, the relevant differences happen inside the Graphical Optimiser, Box [B], between lines 24 and 348. Applying the same driver stated at the very beginning of this Chapter, the intention to bring in new features and tools capable to enhance the 2-level decomposition optimisation process. So, it has been added a routine to evaluate Lagrange Multipliers, 4.4.3, and eventually the possibility to employ the classical visual inspection, based no more on external Matlab files to plot the relevant graphics but embedded in the main routine. Moreover, the choice of having 3D plots at this level has not been anymore offered since already proved of being not so relevant in the Basic Example. The other important difference coming out from Figure 6-19 is the need to read not just the text anlge file, *angle1_c_RL_3.dat*, but the analogue

thickness text file, *thick_OD_2OG.txt*, since this time level 1 has as invariants not just the angles but the *t2* thickness too, as already anticipated in Table 6-5.

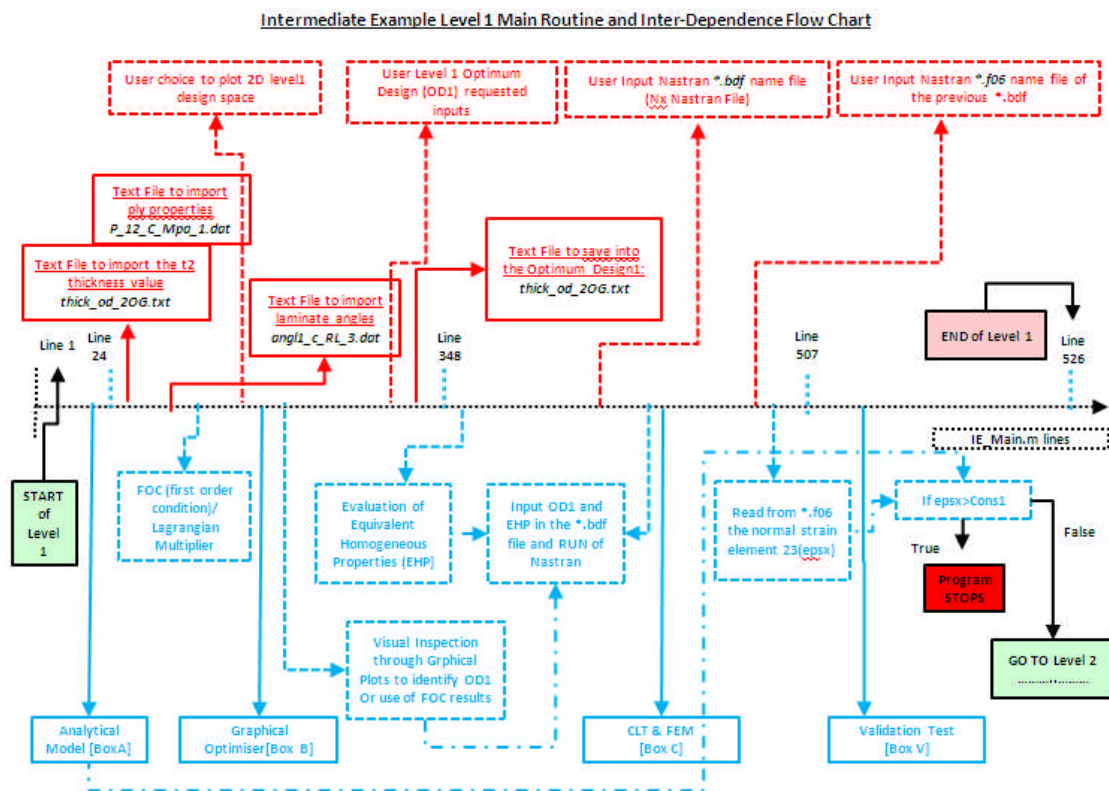


Figure 6-19 - IE_Main.m” Main Matlab File Flow Chart: Level 1

Figure 6-20 confirms that the Intermediate Example level 2 Matlab structure differs to the Basic Example in the Graphical Optimiser Box, between line 551 and 1044. That is coming from the different design space employed, lamination diagram, and the relevant routine created to define a feasible region according to Miki’s design procedure [1] [66]. Then, it has been decided to offer to the user the option to choose between two different laminate configurations, two variable or three fixed orientation angles, and eventually to switch back this choice according to the optimum design level 2 achieved. Here again the end of level 2 takes to the convergence test as the Basic Example did.

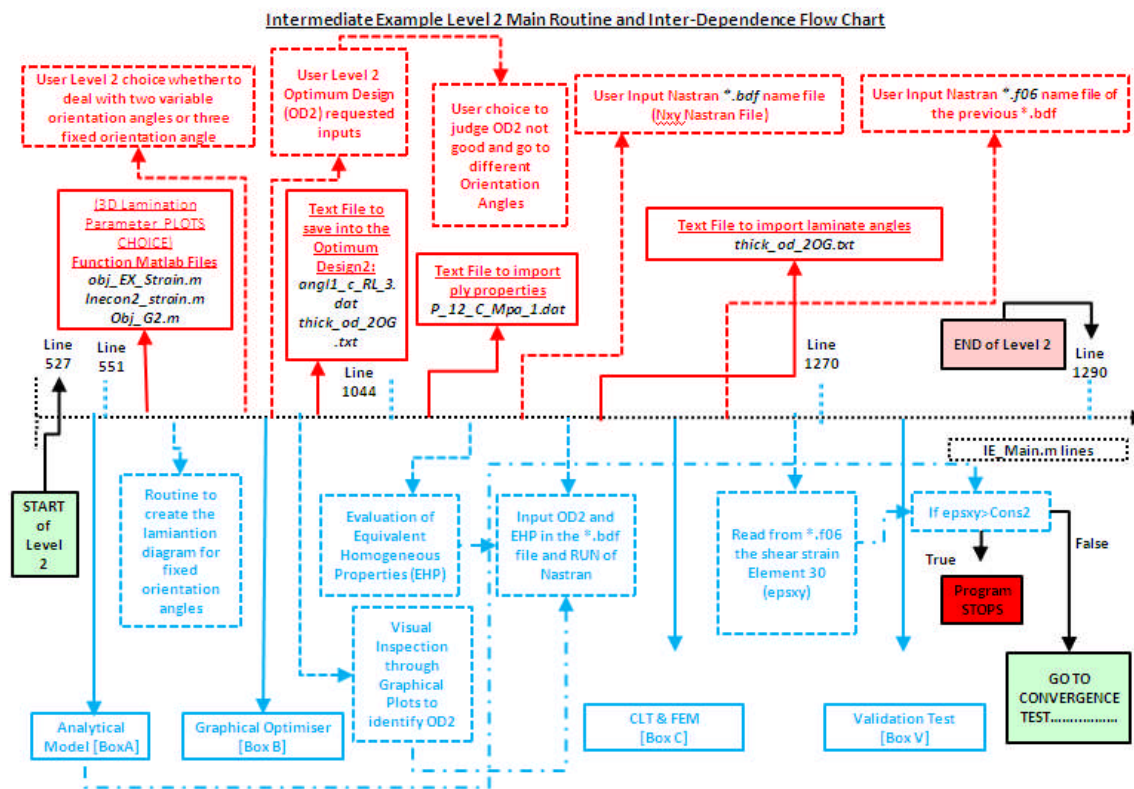
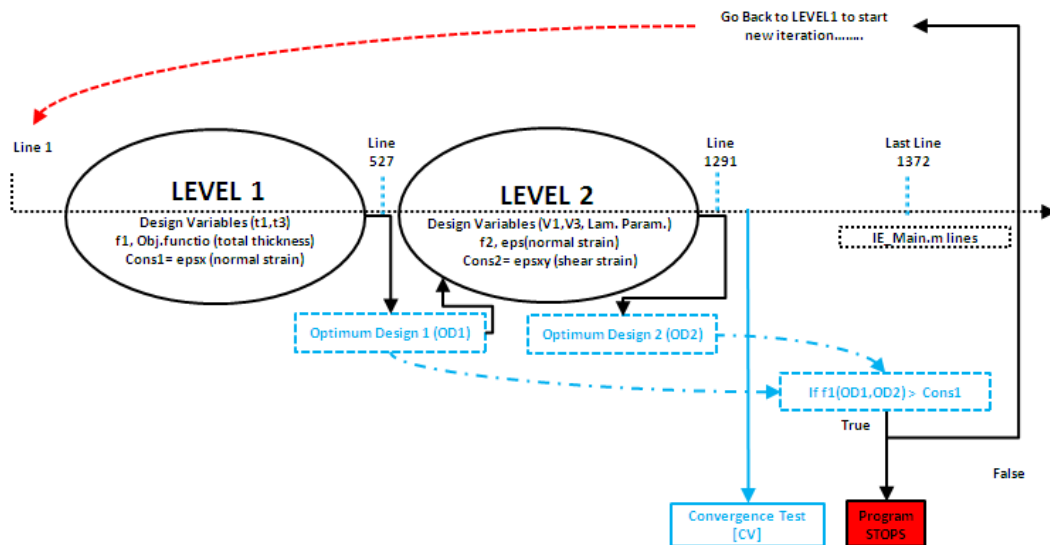


Figure 6-20 - IE_Main.m” Main Matlab File Flow Chart: Level 2

The summary of the whole process including the final Convergence Test are pictured in Figure 6-21. The Optimum Design of Level 1 and Level 2 moves in the same Basic Example way from one level to the other and the possibility to trigger another iteration is ruled by the convergence test based on the strain output of the FEM model.

Intermediate Example Level 1 & Level 2 & Convergence Test Inter-Dependence Flow Chart**Figure 6-21 “IE_Main.m” Main Matlab File Flow Chart: Level 1, Level 2 and Convergence Test**

To obtain more details or to see the “to do list” for running an Intermediate Example Numerical application refer to Appendix B.

6.3 Numerical Application Results and Discussion

Like for the Basic Example, the goal of this sub-section is to group the results of each design cycle at level 1 and level 2 for a laminate configuration built by:

- (Eq. 5-1): Laminate Material
- (Eq. 5-2): Level 1 Load Case, N_x
- (Eq. 5-3): Level 2 Load Case, N_{xy}
- Either Figure 6-3 or Figure 6-4 laminate configuration

At the end of the coming design optimisation result collection, gathered using relevant plots and tables, will be employed specific indexes to enable a more critic and thorough discussion.

This next run, **RUN2**, will be based on the same optimum designs of level1 and level2 just described respectively by:

- 6.1.1.1, OD at level1
- (Eq. 6-19), OD at level2

RUN2 - Iteration1

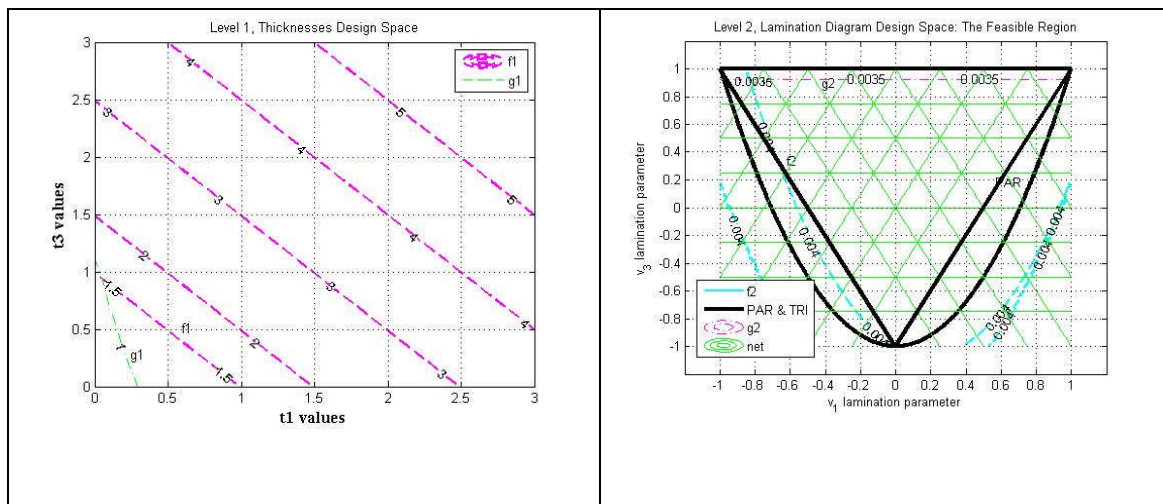
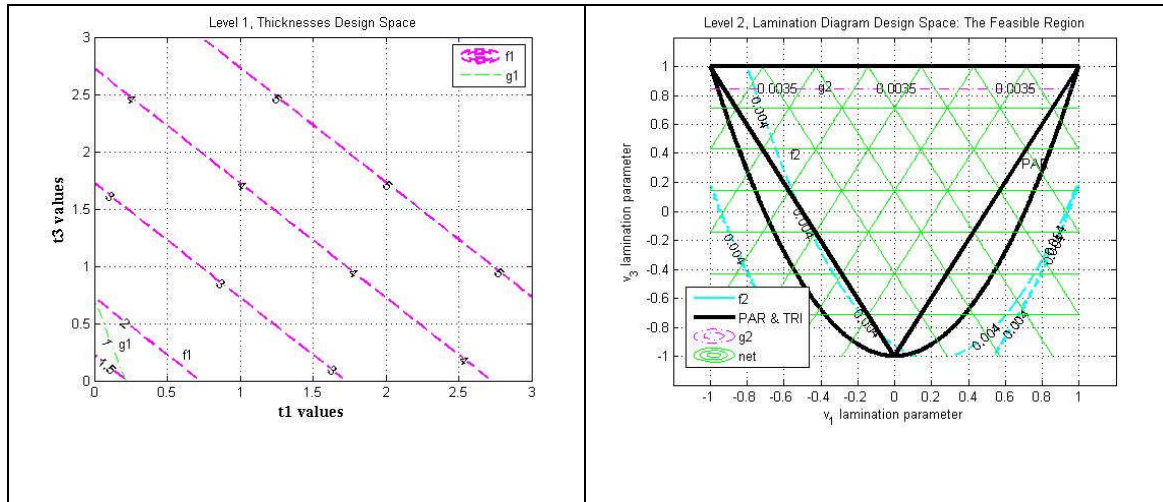


Figure 6-22– Iteration1, 2D L1 and L2 design space plots – Intermediate Example

RUN2 – Iteration1

2-Level Decomposition, Intermediate Example				
Level 1	Invariants		Optimum Design	
	3-Fixed OA	2-Variables OA	$(t_1 = 0.508, t_3 = 1.016)$	
	$\left\{ \begin{array}{l} t_2 = 0.508 \\ \vartheta_1 = 0^\circ \\ \vartheta_2 = 90^\circ \\ \vartheta_3 = \pm 45^\circ \end{array} \right\}$	<u>Not Applicable</u>		
Coupling Quantities	$\{N_{tot} = 16\}$			
Level 2	Invariants		Optimum Design	
	3-Fixed OA	2-Variables OA	3-Fixed OA	2-Variables OA
	$\{N_{tot} = 16\}$	<u>Not Applicable</u>	$\left\{ \begin{array}{l} DN = 3 \\ ESS_L = 1 \end{array} \right\}$	<u>Not Applicable</u>
Coupling Quantities	3_F_OA	$\{t_{2,new} = 1.270 \quad \vartheta_1 = 0^\circ \quad \vartheta_2 = 90^\circ \quad \vartheta_3 = \pm 45^\circ\}$		
	2_V_OA	<u>Not Applicable</u>		

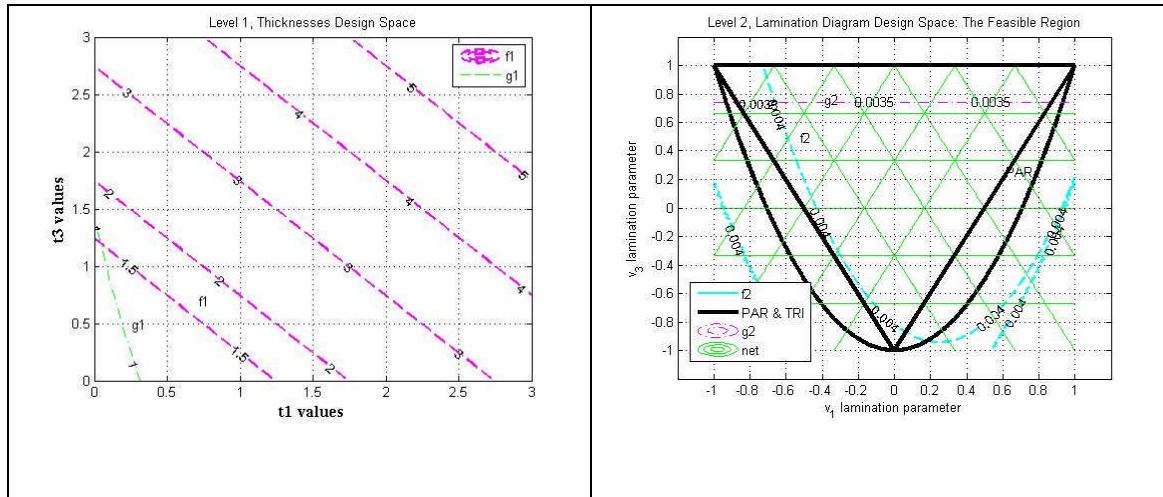
Table 6-6 – Iteration1, Level 1 and Level 2 Design Variables and Coupling Quantities, Intermediate Example

RUN2 – Iteration2**Figure 6-23 – Iteration2, 2D L1 and L2 design space plots –Intermediate Example**

RUN2 – Iteration2

2-Level Decomposition, Intermediate Example				
Level 1	Invariants		Optimum Design	
	3-Fixed OA	2-Variables OA	$(t_1 = 0.254, t_3 = 0.254)$	
	$\left\{ \begin{array}{l} t_2 = 1.270 \\ \vartheta_1 = 0^\circ \\ \vartheta_2 = 90^\circ \\ \vartheta_3 = \pm 45^\circ \end{array} \right\}$	<u>Not Applicable</u>		
Coupling Quantities	$\{N_{tot} = 14\}$			
Level 2	Invariants		Optimum Design	
	3-Fixed OA	2-Variables OA	3-Fixed OA	2-Variables OA
	$\{N_{tot} = 14\}$	<u>Not Applicable</u>	$\left\{ \begin{array}{l} DN = 5 \\ ESS_L = 2 \end{array} \right\}$	<u>Not Applicable</u>
Coupling Quantities	3_F_OA	$\{t_{2,new} = 0.254 \quad \vartheta_1 = 0^\circ \quad \vartheta_2 = 90^\circ \quad \vartheta_3 = \pm 45^\circ\}$		
	2_V_OA	<u>Not Applicable</u>		

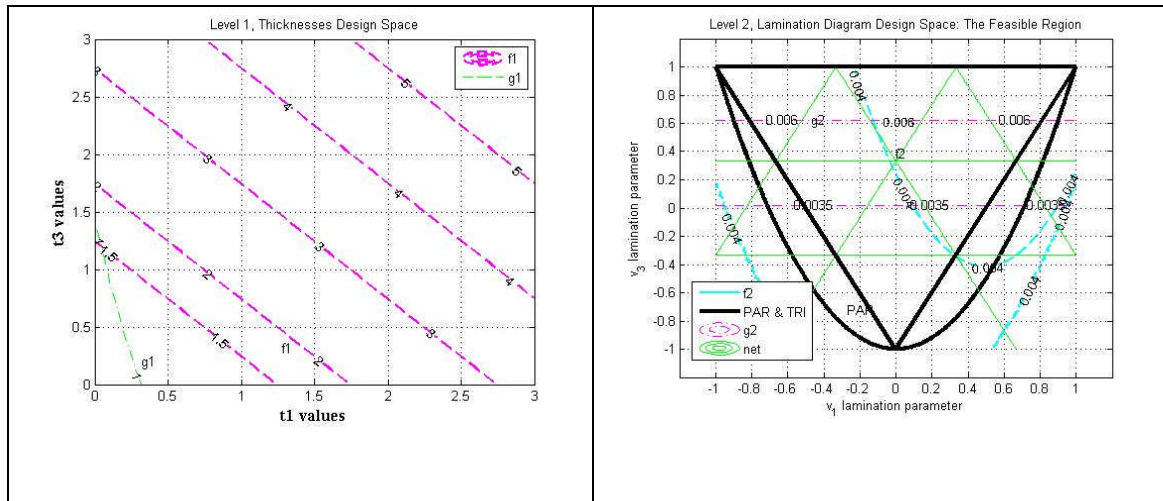
Table 6-7 – Iteration2, Level 1 and Level 2 Design Variables and Coupling Quantities, Intermediate Example

RUN2 – Iteration3**Figure 6-24** – Iteration3, 2D L1 and L2 design space plots –Intermediate Example

RUN2 – Iteration3

2-Level Decomposition, Intermediate Example				
Level 1	Invariants		Optimum Design	
	3-Fixed OA	2-Variables OA	$(t_1 = 0.254, t_3 = 1.016)$	
	$\left\{ \begin{array}{l} t_2 = 0.254 \\ \vartheta_1 = 0^\circ \\ \vartheta_2 = 90^\circ \\ \vartheta_3 = \pm 45^\circ \end{array} \right\}$	<u>Not Applicable</u>		
Coupling Quantities	$\{N_{tot} = 12\}$			
Level 2	Invariants		Optimum Design	
	3-Fixed OA	2-Variables OA	3-Fixed OA	2-Variables OA
	$\{N_{tot} = 12\}$	<u>Not Applicable</u>	$\left\{ \begin{array}{l} DN = 5 \\ ESS_L = 1 \end{array} \right\}$	<u>Not Applicable</u>
Coupling Quantities	3_F_OA	$\{t_{2,new} = 0.254 \quad \vartheta_1 = 0^\circ \quad \vartheta_2 = 90^\circ \quad \vartheta_3 = \pm 45^\circ\}$		
	2_V_OA	<u>Not Applicable</u>		

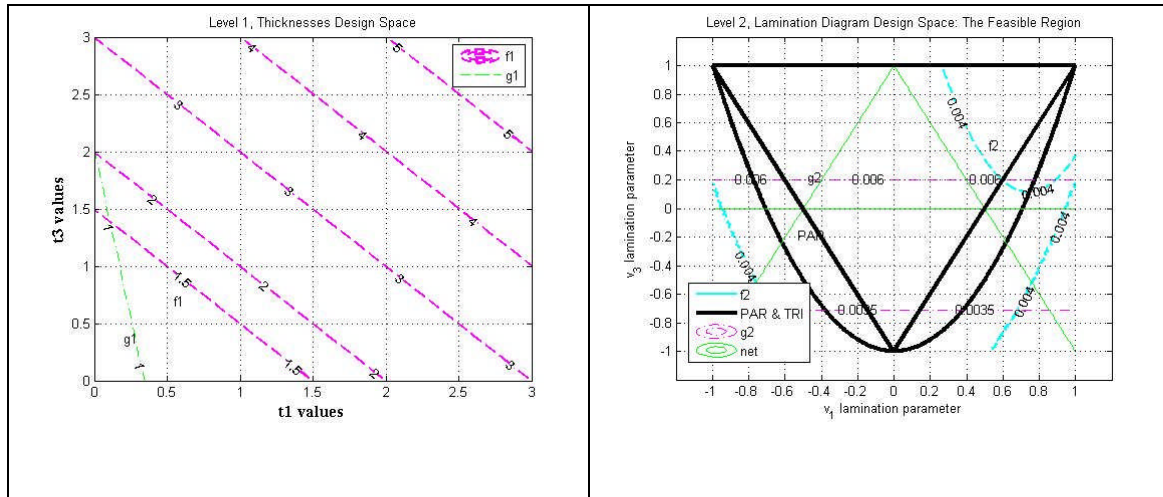
Table 6-8 – Iteration3, Level 1 and Level 2 Design Variables and Coupling Quantities, Intermediate Example

RUN2 – Iteration4**Figure 6-25 – Iteration4, 2D L1 and L2 design space plots –Intermediate Example**

RUN2 – Iteration4

2-Level Decomposition, Intermediate Example				
Level 1	Invariants		Optimum Design	
	3-Fixed OA	2-Variables OA	$(t_1 = 0.254, t_3 = 0.254)$	
	$\left\{ \begin{array}{l} t_2 = 0.254 \\ \vartheta_1 = 0^\circ \\ \vartheta_2 = 90^\circ \\ \vartheta_3 = \pm 45^\circ \end{array} \right\}$	<u>Not Applicable</u>		
Coupling Quantities	$\{N_{tot} = 6\}$			
Level 2	Invariants		Optimum Design	
	3-Fixed OA	2-Variables OA	3-Fixed OA	2-Variables OA
	$\{N_{tot} = 6\}$	<u>Not Applicable</u>	$\left\{ \begin{array}{l} DN = 3 \\ ESS_L = 1 \end{array} \right\}$	<u>Not Applicable</u>
Coupling Quantities	3_F_OA	<u>Not Applicable</u>		
	2_V_OA	$\{t_{2,new} = 0 \quad \vartheta_1 = 0^\circ \quad \vartheta_3 = \pm 45^\circ\}$		

Table 6-9 – Iteration4, Level 1 and Level 2 Design Variables and Coupling Quantities, Intermediate Example

RUN2 – Iteration5**Figure 6-26** – Iteration5, 2D L1 and L2 design space plots - Intermediate Example

RUN2 – Iteration5

2-Level Decomposition, Intermediate Example				
Level 1	Invariants		Optimum Design	
	3-Fixed OA	2-Variables OA	$(t_1 = 0.254, t_3 = 0.254)$	
	<u>Not Applicable</u>	$\left\{ \begin{array}{l} t_2 = 0 \\ g_1 = 0 \\ g_3 = 45 \end{array} \right\}$		
Coupling Quantities	$\{N_{tot} = 4\}$			
Level 2	Invariants		Optimum Design	
	3-Fixed OA	2-Variables OA	3-Fixed OA	2-Variables OA
	$\{N_{tot} = 4\}$	<u>Not Applicable</u>	$\left\{ \begin{array}{l} DN = 1 \\ ESS_L = 1 \end{array} \right\}$	<u>Not Applicable</u>
Coupling Quantities	3_F_OA	<u>End of the iteration since overcome the allowable normal strain</u>		
	2_V_OA	<u>Not Applicable</u>		

Table 6-10 – Iteration5, Level 1 and Level 2 Design Variables and Coupling Quantities, Intermediate Example

The last pages represent the collection of all RUN2 iterations taken to get to the final best laminate configuration. Plots and tables are grouped according to the same approach already introduced by the Basic Example in section 5. Each figure has a top

row of 2D plots, level 1 and level 2 design spaces, and a bottom of 3D graphs just of level 2 objective and inequality constraint functions. The parabola mentioned earlier, Figure 6-11, has not counted anymore as invariant through the whole optimisation process. On the other hand tables summarise the main level 1 and level 2 data coming from the previous plots.

The first iteration data, Figure 6-22 and Table 6-6, has already been described and discussed earlier at the beginning of the section 6.1.2. They evidence the optimum design anticipated by 6.1.1.1 and (Eq. 6-19). In fact, the 3-fixed orientation angle option is chosen and the total number of laminate plies at this stage is, no matter which level, sixteen. Level-2 feasible domain is quite open since the 0.6% shear strain constraint function does not appear yet. The only bounds are due to the thick solid triangle lines, Figure 6-22, and the objective dash-dot function line, which is inherently limited, as constraint function of level 1, to a 0.4% normal strain. So the feasible iteration-1 domain is obtained counting out from the big thick solid line triangle area the tiny curve-line CMH triangle zoomed in Figure 6-22.

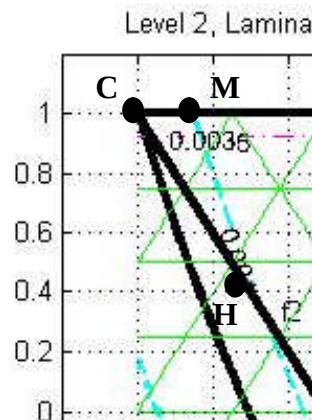


Figure 6-27 – Level-2 Unfeasible Domain: Run2, Iteration1

Finally the optimum design chosen at this level, (Eq. 6-19), identified then as coupling quantity to bring to the next iteration, gives a ply 2 thickness, $t_{2,new}$, of 10 plies. This value is much higher than the four plies laminate at iteration1 level1, and this is the reason why the level1 design space at iteration2, Figure 6-22 top left plot, widened. To compensate this undesired effect the choice of the level1 optimum design is so to make an overall iteration2 laminate smaller than in the previous design cycle. This consideration is pivotal for a proper convergence step toward the correct final solution.

In fact, iteration2, Figure 6-22 and Table 6-6, moves one step closer to the sought lightest laminate configuration defining at both levels a laminate of 14 plies. So, the total number of plies is dropped down compared the previous sixteen and that is confirmed too by the number of equal spaced segment, ESS, at the first division number of the top right Figure 6-22 graph. These are seven; half the overall ply laminate numbers at this iteration according to the description in the third bullet point equation of (Eq. 6-20). Unfortunately this iteration at level 2 is still not relevant to identify a visible narrowed feasible domain since the 0.6% shear strain constraint function is a value not yet reachable as proved by the 3D bottom right plot of Figure 6-22 . The reason is found in the size configuration of the laminate, so far too thick to strain in sensitive way to N_{xy} load case. The $t_{2,new}$ value to be passed on the next iteration, iteration 3, now is quite small, equal to 2 according to the very bottom of Table 6-6, or as described by the level1 invariant values of iteration3 Table 6-6. This quite relevant reduction of t_2 , from 1.270 mm (iteration2 level 2) to 0.254 mm (iteration3 level 1), makes the feasible domain of level1 2D Figure 6-22 plot shrunk as expected. But the first considerable feasible level 2 space shrinking is the one in Figure 6-22 at iteration 4. In fact, the 2D top right plot shows a visible smaller allowable region for the selection of the best laminate as Figure 6-28 displays.

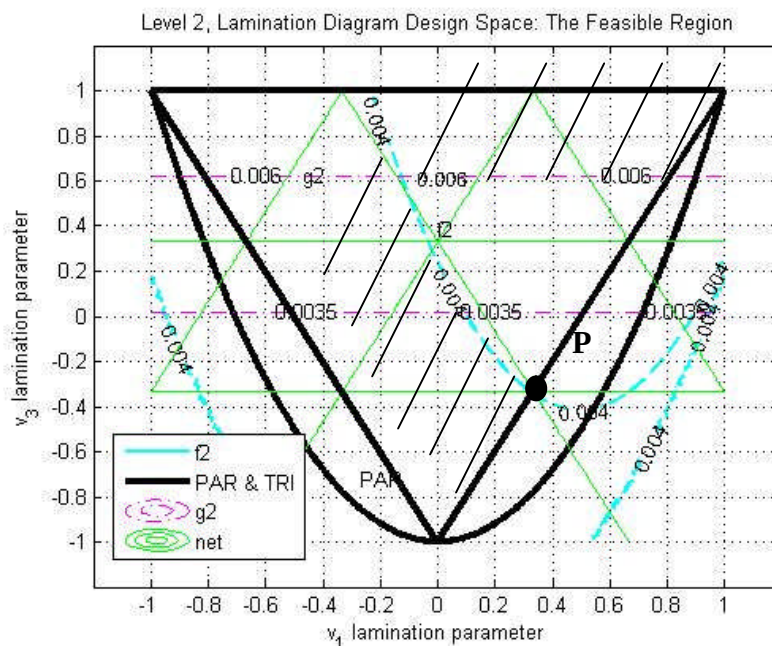


Figure 6-28 – Level-2 Feasible Domain: Run2, Iteration 4

This region is defined in the below Figure 6-28 , the hashed side indicates the infeasible region, by the region bounded by 0.6% shear strain horizontal line, by the oblique edge of the thick solid triangle and finally by the 0.4% normal strain dashed line. The main reason of such smaller feasible domain size is in the overall number of layers at this iteration, Table 6-9, just six. That N_{tot} value is confirmed by the ESS of the first division number Figure 6-28, these are three correctly the half of the total plies. Since still dealing with predefined orientation angles, the only optimum design option so far is identified by the black P spot in Figure 6-28. This level 2 optimum design is identified by a division number equal to 3 and ESS_L to 1 as highlighted in Table 6-9. This choice corresponds to a laminate with two orientation angles, since $t_{2,new}$ becomes equal to zero, and enables to prove the other capability of this program to cope with two variable orientation angles. This new particular condition, $t_2 = 0$ as confirmed in Table 6-9 gives to the level 1 design space Figure 6-22 the smallest feasible space as confirmed by the g_1 x and y intersections, much higher than any earlier iteration. Now, to confirm the trend to decrease at each iteration the number of plies and at the same time to describe a symmetric laminate, the user is taken to select a level 1 optimum design as such the laminate is a 4 layers arranged. But this decision in the level 2 design space gives no more margin of selection for a 3-fixed orientation angle configuration as better described by Figure 6-29.

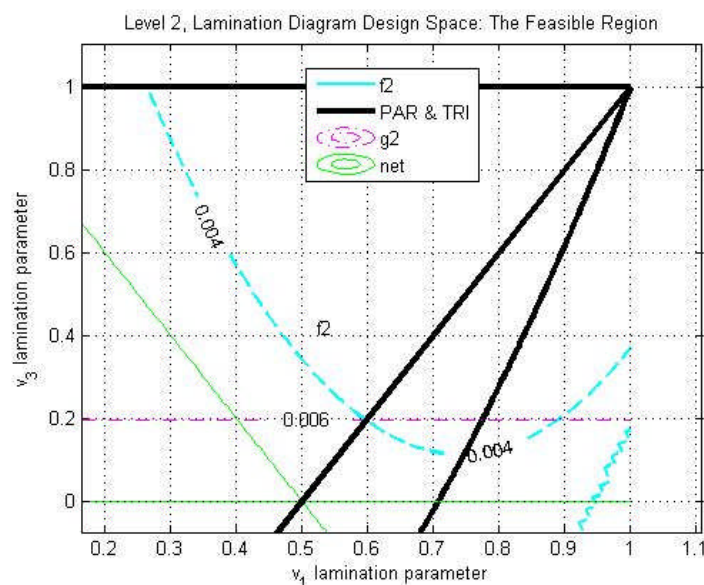


Figure 6-29 – Level2 Feasible Domain: RUN2 Iteration 5

Then as outlined by Table 6-10 the choice of DN and ESS_L equal to 1 is just a conscious decision to quit the process since selecting a discrete point inside the infeasible region. So as consequently expected the Matlab script flags out the message:

Matlab Command:

```
%Covergence Test Condition

elseif f2_it(k)> Cons1

    warning('Reached Convergence. Take Optimum Designs, OD1 and
    OD2, of the Previuos Iteration\n')
    Result_T=OD(k-1).L2
    Result_eta=OD(k-1).eta

    Break

end
```

Before moving on to the RUN2 time history main plots, it is worth adding a comment on the Figure 6-29 meaning. Once reached this stage, the program puts the user to a further option that would have enabled the main process to continue. Swapping to the 2-variable orientation angle configuration changes the feasible region domain. In fact, as described by Figure 6-11 the “triangle” space domain is applicable to the predefined orientation angle configuration only. The decision to work with any orientation angle domain brings back to the “parabola” feasible region of Figure 6-11. According to these considerations the new feasible domain at RUN2 iteration 5 will be the one described below in Figure 6-11.

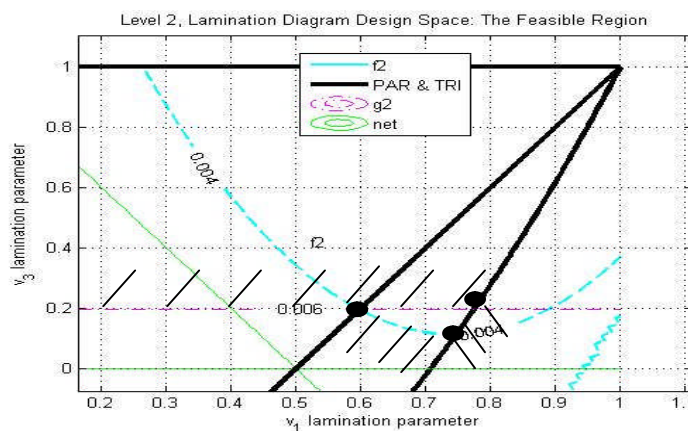


Figure 6-30 – Level2 Feasible Domain: RUN2 Iteration 5, 2-variable orientation angle

The above narrowing design space process is one of the main point of differences between the Basic and Intermediate Examples. The former, behaves coherently according to Figure 6-11 diagram; at any iteration step both level feasible domain tend to shrink themselves. The latter, differently as just showed, continues to shrink just the lamination parameters design space whilst the level 1 thickness domain is not. That functioning is mainly due to the “t2” intermediate example coupling quantity that in the basic example does not either exist.

After having documented the RUN2 convergence process of the Intermediate Example through the five above iteration design cycles it is time to focus on the trend of the whole optimisation process. The following figures grouped the design variables, objective and constraint functions at each level for four design cycles, and not five since the Table 6-10 iteration 5, is out of the normal and shear strain bounds.

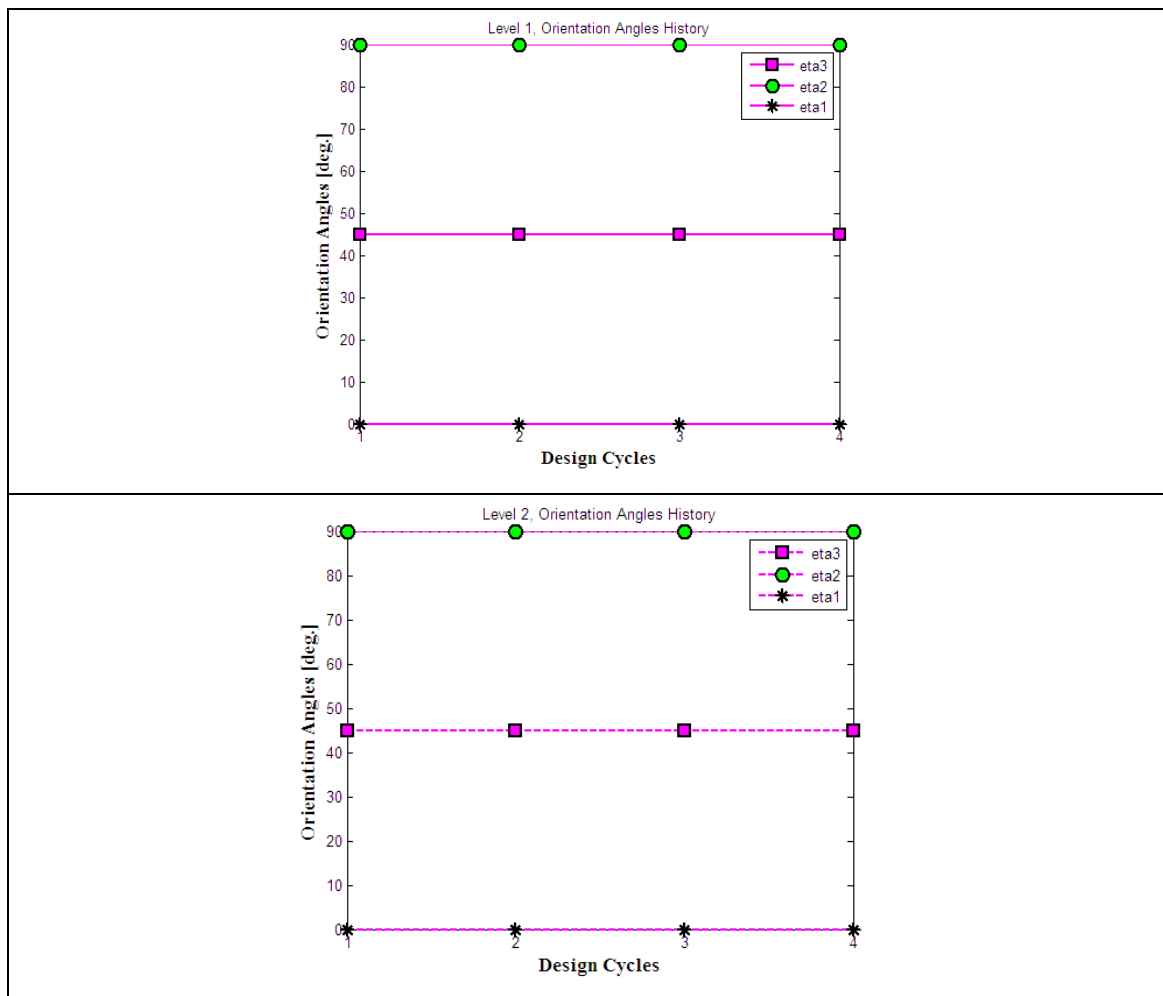


Figure 6-31 – RUN2, L1 and L2 Orientation Angles

Figure 6-31 for the specific RUN2 typology is not very indicative since the orientation angles history does not change through the optimisation. But it is still a necessary plot to be available whenever the user decided to swap laminate configuration and record the change.

Next series of graphs of Figure 6-31 are grouped in a particular way to enable reciprocal comparison in both vertical and horizontal directions. The first plot row describes t_1 and t_3 history at level 1 and level 2. These two graphs basically do not provide the same negative slope distribution as the Basic Example showed. The reason is in the different design variables of the intermediate example. The thicknesses, t_1 and t_2 , are design variables just at level1 and as mentioned before and described by the feasible region design cycle trend of Figure 6-27, Figure 6-28, Figure 6-29, and Figure 6-30 the level 2 domain is the one most significant in the whole convergence process. Unfortunately the level 2 lamination parameters, in a three-fixed orientation angle configuration, have not been selected through the iterations to point the optimum design. This weird situation, plotting DN and ESS_L would not be of any real support, recommended to consider the t_2 thickness of the fixed orientation group angle, see Figure 6-3.

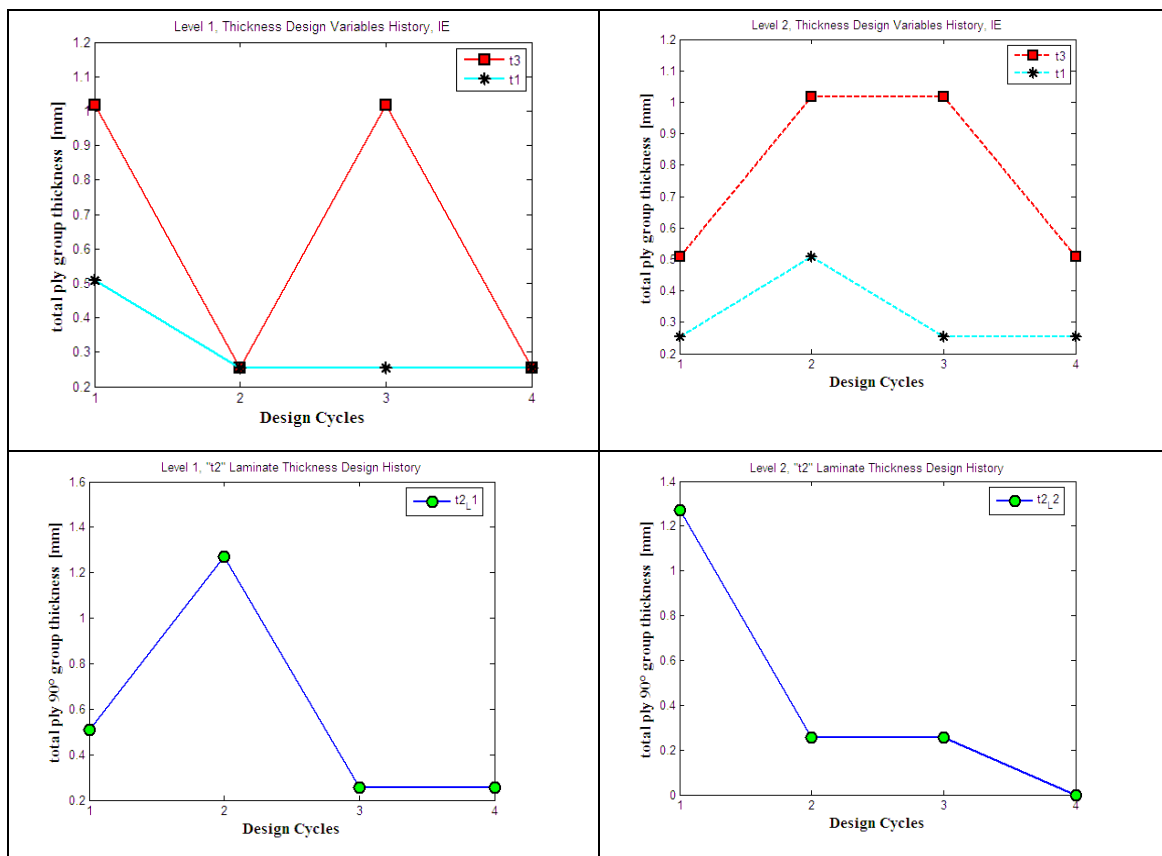


Figure 6-32 – RUN2, L1 and L2 thicknesses and t_2 design history

This decision to plot t_2 design history in Figure 6-31 proves to be useful. In fact, either at level 1 or level 2, when t_1 and/or t_3 decrease t_2 increases and vice versa. This behaviour had been already described as part of the user attempt to select at any iteration a laminate lighter. That is mainly due to the lack of control on the single ply thickness of t_1 and t_3 during the second level optimum design selection. Then, Figure 6-31, says that t_2 despite is not a design variables at any level, behaves as a reliable trend index in the overall optimisation process. It proves to have a negative slope as the objective function of level 1 in Figure 6-31.

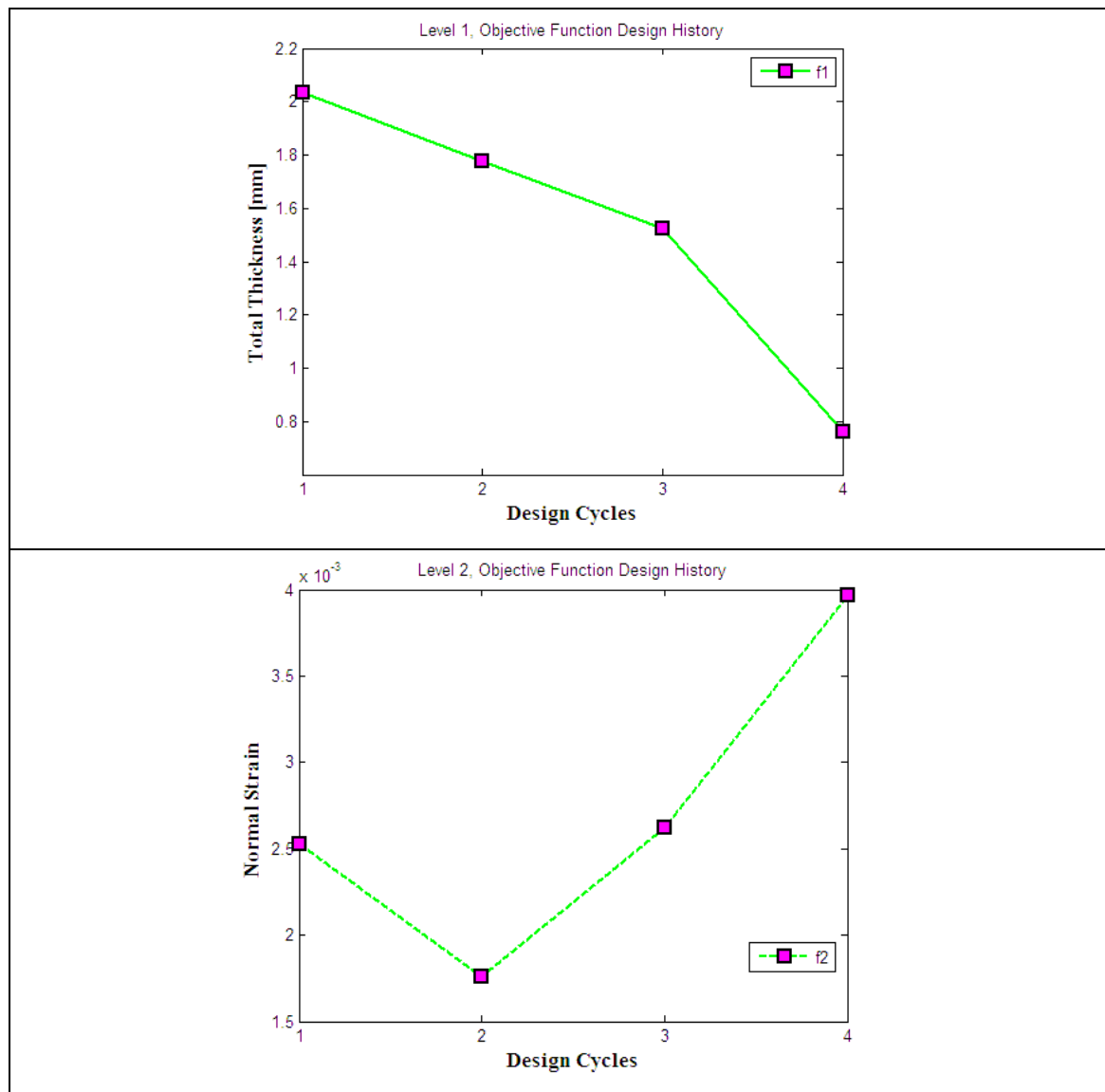


Figure 6-33 – RUN2, L1 and L2 Objective Functions Design History

In fact, the top Figure 6-31 plot tends to decrease its magnitude over the optimisation process since the total laminate thickness, f_1 , is expected to be minimised. On the other hand the level 2 objective function, bottom graph, for the convergence requirements mentioned earlier, has to be maximised and a part the second iteration data, it does so. In fact, the plot is increasing his y magnitude over the design cycles.

Finally the constraint function and the respective FEM strain of level 1 and level 2 are put together Figure 6-34.

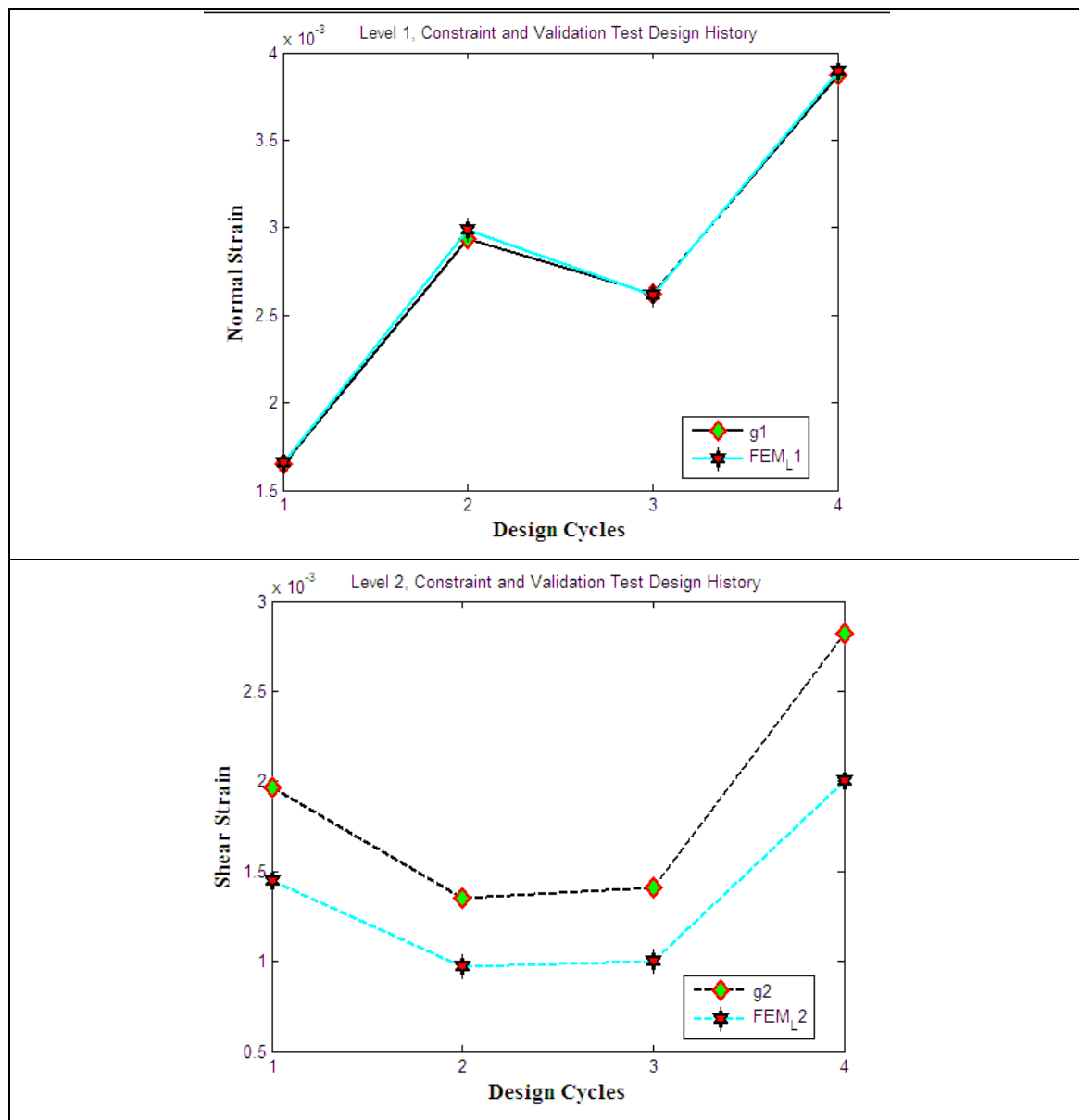


Figure 6-34 – RUN2, L1 and L2 Constraint Function and FEM strain data History

The level 1 and level 2 Figure 6-34 plots highlight a very similar pattern between classical lamination theory and finite element modelling data but low level of matching in relation of the singular data regarding the level 2. The shear strain distribution, level2 plot, had already come out in the basic example, Figure 5-54, confirming the Nastran outcome is lower. Surprisingly the normal strain of this RUN2 Intermediate Example seems to be very close when RUN1 Basic Example proved differently at the very last iteration 6. A possible justification is in the overall thickness of the laminate configuration at these stages. In fact, Basic Example iteration-6 provided an overall thickness, level 1 objective function, of 0.46 mm, Table 5-13, on the other hand the Intermediate Example iteration-4 objective function is equal to 0.76 mm, Table 6-11. That confirms an impression had in the basic example results discussion that the higher the laminate thickness and nearly perfect is the normal strain matching between CLT and FEM. Here, still having a higher overall thickness, 076 mm, no divergence effect is present contrary to the pattern of the Basic Example Constraint design history in Figure 5-54. This time the RUN2 is made of a higher number of orientation angles than the RUN1 basic example had been doing. Despite this simple evidence so far the shear strain is still lower than the classical lamination theory, and eventually will require a deeper investigation supported by a sensitivity analysis of the different load cases, (Eq. 5-2) and (Eq. 5-3), considered. Moreover Figure 6-34 level1 constraint function proves to move closer after any iteration to the 0.4% normal strain threshold than the level2 does to the 0.6% shear strain. This might explained by the less flexible level 2 design space. In fact, dealing with fixed orientation angles the selection of the optimum laminate candidate is restricted to a number of discrete points as showed earlier by top and bottom plots in Figure 6-34.

The following Table 6-11 summarises all the main data of the previous plots and then enables to evaluate reliability and effectiveness information on the RUN2.

Cycle Number	Level 1			Level 2		
	f1	g1	FEM_epsx	f2	g2	FEM_epsxy
1	2.032	0.001651	0.001657	0.002527	0.001968	0.001443
2	1.778	0.002940	0.002984	0.001763	0.001353	0.000975
3	1.524	0.002622	0.002612	0.002622	0.001412	0.001001
4	0.762	0.003871	0.003895	0.003970	0.002824	0.002001

Table 6-11 – RUN2, Level1 and Level2 Design History

In fact, employing (Eq. 5-22), (Eq. 5-23), (Eq. 5-24), and (Eq. 5-25) considering the shear and normal strain threshold, Cons1 and Cons2, are still the same of the Basic Example, then it is possible to populate the following table:

	$(RI_N)_k$	$(RI_S)_k$	$(EI_1)_k$	$(EI_2)_k$
RUN2, (k=4)	0.9938	1.413	1.034	2.125

Table 6-12 – RUN2 Reliability and Effectiveness Indexes, final iteration (k=4)

The values of the above Table 6-12 confirm the considerations came out from the last plots. In fact, the reliability indexes of normal, $(RI_N)_k$, and shear, $(RI_S)_k$, strains present opposite behaviour as Figure 6-34 describes and Table 5-14 of Basic Example anticipated. The fact the reliability index of level-2, shear strain, is greater than unit says the Nastran structural idealisation, alongside the Wisnom's assumption, [44] [45], stated in (Eq. 2-1), need to be further understood when a side load, N_{xy} , is applied. It is important to underline this effect does not seem to be related to the thickness of the laminate under investigation, as the normal strain of the basic example at the last design cycle, but it is an effect present from the very beginning. In fact, it might be useful, when the interpretation of the data is more complex or when the decomposition levels are higher than two, to plot the reliability index through the whole optimisation process. In this case both reliability indexes would prove to be pretty much consistent. Then, it is worth noting that Table 6-12 has been populated either for the level-1 or the level-2 at

the same k-iteration, while it did not happen to the basic example Table 5-13. This is explainable to the different level where the constraint condition has been overtaken. The basic example at level-2 interrupted, Figure 5-51 shows the convergence achieved, on the other hand the intermediate example at level 1 passed the normal strain allowable. These considerations might be confirmed looking at the effectiveness indexes in Table 6-12. The level-1 effectiveness value close to one at the last iteration says the goodness of the refining optimum design search process of using nearly all the design space available. Differently it happened for the level-2 effectiveness index, greater than two. That denotes a poor capacity to explore the lamination diagram feasible space. Of course the constraint of dealing with fixed orientation angles it might be a further limitation to reduce the flexibility of the whole refining process of this level. At the end these two indexes teaches something else that for a significative multi-level optimisation it is expected the capability to take both the effectiveness indexes close to one otherwise a poor final optimum design will be obtained. In fact, in this specific case the laminate in Figure 6-35 describes a configuration properly designed in relation to axial load N_x but overdesigned regarding the side load N_{xy} .

The conclusion of this section is made by the laminate configuration history at the end of any level 2 iteration. The y axis of Figure 6-35 bar graph describes the number of half laminate plies for each orientation group.

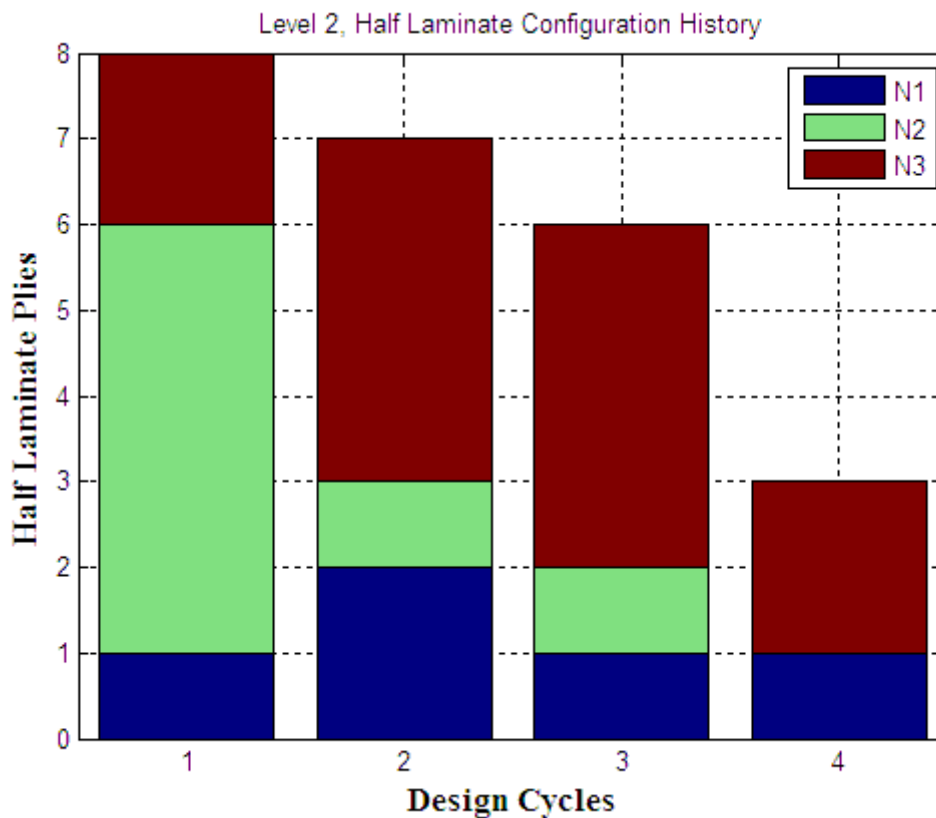


Figure 6-35 - RUN2, Intermediate Example: Half Laminate Configuration History, Level2

The above bar plot calls together a number of information already discussed through the past diagrams of Figure 6-31, Figure 6-32, Figure 6-33, and Figure 6-34 and enables to make a check of the reliability of the previous data.

Figure 6-35 describes the ply group thicknesses of half laminate already showed by the top and bottom right plots of Figure 6-33. They match considering the single ply thickness is always 0.127 mm. Then is confirmed the decreasing effect on the number of overall laminate layers, it starts from 8 and ends up with 3 passing through 7 and 6 plies. It is quite interesting that in such specific 2-level decomposition process despite each single ply group thickness goes up and down the overall laminate thickness keeps decreasing as the entire multistep iteration optimisation expects.

The final optimisation laminate configuration according to Figure 6-33 and Figure 6-3 convention is as follows:

(Eq. 6-21)

- $[(\pm 45^\circ) (0^\circ)]_S$

The above final RUN2 optimum laminate configuration of (Eq. 6-21) confirms the symmetric characteristics but goes against the two laminate arrangements stated at the very beginning of this chapter and described by Figure 6-3 and Figure 6-4. In fact, the former expects to deal with three fixed orientation angles (0° , 90° , and $\pm 45^\circ$) while the latter with two fixed orientation angles (any except to 0° and 90°).

6.4 Summary

A 2-level decomposition optimisation process has been enhanced in this chapter by performing a numerical application, RUN2. Despite the optimisation technique employed is the same of the Basic Example; the Intermediate Example developed a series of tools and concepts needed in the continuous attempt to define a solid frame work to deal with thick composite laminate. First, the introduction of a numerical optimiser, 6.1.2, helped to understand the way to tackle optimisation problem not manageable just with graphical means. Moreover expanding the knowledge of Matlab applying symbolic techniques to deal with Kuhn-Tucker, 4.4.3, conditions. The next step to compare the numerical optimiser outcome and matching it with the graphical visual inspection gave more confidence in either the different level1 optimisation search tools. But the main novelty of the current example has been dealing and looking for real life laminate optimum design configuration. To achieve that has been introduced a flexibility within the program to deal either with 3-fixed or 2-variable orientation angle arrangement, Figure 6-3 and Figure 6-4. Such new boundary conditions have been supported by a different level2 design space. In fact, the introduction of the lamination parameters, gave a different response of the coupling quantities of the whole multi-level decomposition process, as highlighted through the comparison of Table 6-4 and Table 6-5. All that overcame the lack of clarity faced with the basic example orientation angle domain, the convergence inside the lamination diagram has proved to be more effective,

as through Figure 6-27, Figure 6-28, Figure 6-29, and Figure 6-30. Then the quantities that identify the predefined orientation angle laminate possible configurations are not anymore radiant numbers but easier integer numbers as DN and ESS_L (Eq. 6-11).

Those innovations caused few undesired setback effects, as those described by the effectiveness and reliability indexes, Table 6-12. The former is mainly related to the predefined orientation angles choice that as already discussed and showed, Figure 6-29, makes the optimum design selection more rigid. The latter alerted on the poor matching from classical lamination theory and FEM strain outcome, despite the design cycle pattern has proved to be very the same, Figure 6-34. This limitation needs to be explored further but there are all the signals that might depend on the increased number of orientation angle groups inside the current example. In fact, the basic example proved to have better matching, Table 6-12, in particular regarding the normal strain value very close to one, 0.9979.

That weakness is the next direction of improvement to think about. Therefore it has been decided to perform a FEM idealisation as the one described in the basic example chapter, 5.1.3, but just applied on a single orientation angle group. This new approach will be based on the concepts of sub-laminate blocks already described in 3.6.4 and it will be one of the new challenges of the coming section; the Advanced Example.

7 Commercial Optimiser Comparison

This chapter is meant to try to understand the weaknesses, strengths and finally the future direction of improvements of the 2-level decomposition optimisation processes presented by Basic and Intermediate Examples. To succeed in that task it will be necessary to introduce a different optimiser to execute the same numerical application analyses earlier described by RUN1 and RUN2 and then compare the results.

Nastran Optimiser, SOL200 [78] has been chosen as alternative mean to evaluate the optimum design of the real-life examples outlined before. A thorough description of its main structure will be made then to facilitate the interpretation and comparison of the final results. In fact, a main data flow characterisation, identifying the main boxes of the whole process, will be performed to point out similarities with the 2-level decomposition structure.

The main steps to develop the SOL200 input file have been performed through PATRAN as external pre-post processor. Even if the main results post process will be fulfilled with Matlab.

Finally a comparison of the optimum design data of the basic and intermediate example has pulled out the main characteristics of the different optimisers adopted. Then the Classical Lamination Theory, used in the present work as core of the structural analysis, has been validated comparing its outcome with the one coming from a specific laminate properties evaluator embedded in the Patran software.

7.1 Optimiser Description

“MSC Nastran Design Sensitivity and Optimization” commercial solver is intended to enhance the capability of the standard MSC Nastran to predict the effect of changes in the structural model, the same model described within the basic and intermediate example as part of the validation test (section 5.1.3), on structural responses (sensitivity) and to resize the structure so as to satisfy imposed conditions while taking a specific response to minimum or maximum value [78].

This particularity of linking directly the optimisation methods with the structural analysis in general all inside a single process made Nastran as first optimiser choice. In fact, coupling the concepts of design properties and design responses familiar to the analysis to the design variables and constraints that are familiar to the optimisation, makes the whole process very similar to the one already described by the basic and intermediate examples.

MSC Nastran links these two models, analysis and optimisation, by introducing two intermediate concepts. The first is the designed property and provides a link between the design variable used by the optimiser and the finite element property used in the finite element analysis. The second concept is the design response. A design response is a physically based result that can be used as the design objective or it can be used, with user imposed limits, as a design constraint. The presence of this intermediate layer between the design quantities and the analysis quantities is displayed in Figure 7-1 by the novel idea of the Approximate Design Model gathering these two concepts.

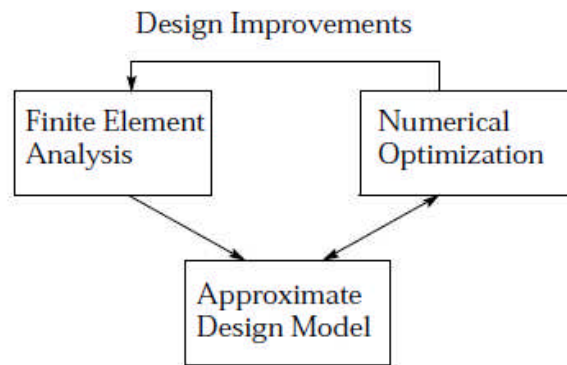


Figure 7-1 – Coupling Analysis and Optimisation according to Nastran SOL200 [78]

Figure 7-1 shows the finite element analysis forms the basis for creation of the approximate model which is subsequently used by the optimiser. The approximate model includes the effects of design variable linking, constraint deletion, and formal approximations. Design variable linking is established by the user (as the ply thickness of the laminate), while constraint deletion and formal approximations are performed automatically in MSC Nastran [78]. The constraint deletion alongside a specific sensitivity design tasks are the main boxes that feed the data to the approximate model as described clearly by Figure 7-2.

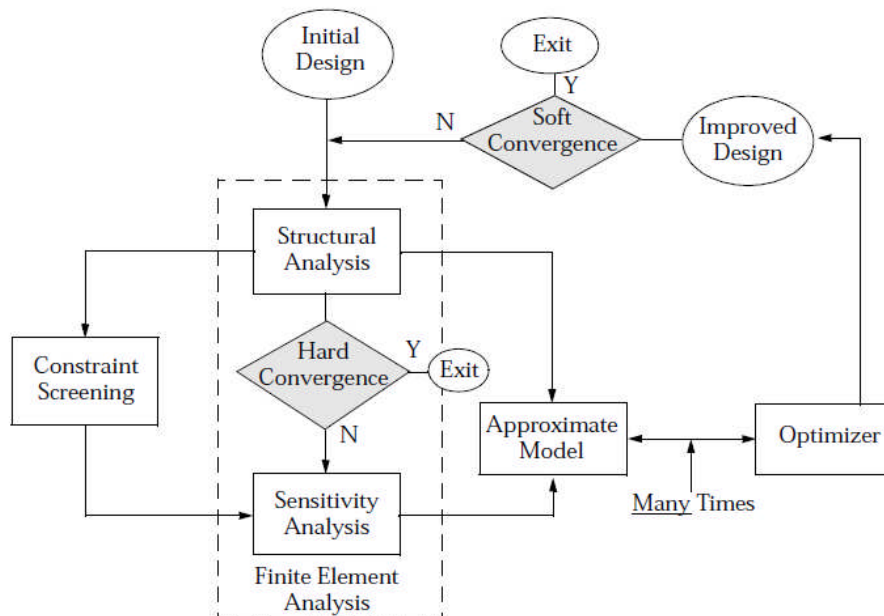


Figure 7-2 – MSC Nastran main structure [78]

In particular the first introduces the idea to limit the constraints that are considered in the redesign to those that are likely to play an active role in the redesign. The second, design sensitivity, computes the rates of change of structural response quantities with respect to the changes in design variables. Those quantities, evaluated with partial derivatives, provide the essential gradient information to the optimiser passing through the approximate model.

Since structural optimisation is an iterative process, numerical criterion must be established to determine when the overall process has converged, same need of the multi-level decomposition technique of this work of research 5.1.5. In Nastran SOL200 there are two levels at which convergence is tested: the first and lower level is at the optimiser level, part of the numerical optimisation technique employed, and the second and higher level is with respect to the overall design cycles. The last convergence type is further broken up as soft convergence and hard convergence as shown in Figure 7-2 which highlights the convergence aspects of the design cycle loop. Soft convergence is based on the results of the approximate optimisation, while hard convergence is based on finite element analysis results.

Once a new design has been proposed by the optimiser, based on the information supplied by the approximate model, 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 7-2 If a subsequent approximate optimisation is deemed necessary, the finite element analysis serves as the new baseline from which to construct another approximate sub-problem [78]. This cycle may be repeated as necessary until convergence is achieved.

All these concepts can be grouped in the simpler and more essential following Figure 7-3 adopting a display similar to the one of the basic and intermediate example in Figure 5-1, and Figure 6-1.

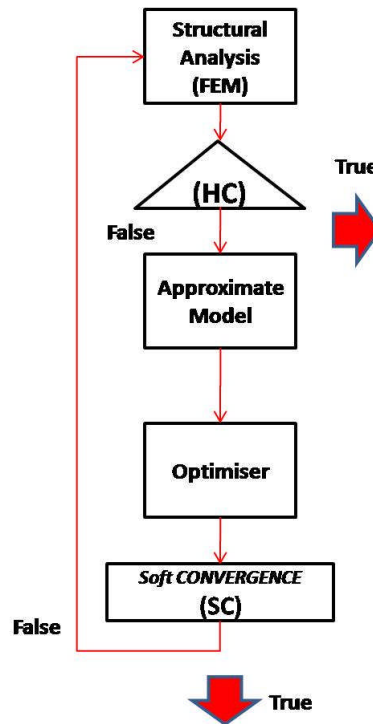


Figure 7-3 – Nastran SOL200: Structural Analysis and Design Study Data Flow Structure

Figure 7-3 highlights the main order tasks within a Nastran SOL200 optimisation structure, putting in evidence that the whole process terminates when the two convergence tests, Hard Convergence (HC) and Soft Convergence are satisfied. The structure of Figure 7-2 and Figure 7-2 basically is the same the only advantage is making easier to pull out similarities with the 2-level decomposition process.

7.2 “Commercial Optimiser” VS “2-Level Decomposition Technique”

The essence of the comparison is passing through a flow data description of the 2-level decomposition pulling the structural analysis and design study tasks out.

Reviewing in such fashion the single level structure of the whole process of the basic and intermediate examples, described by Figure 5-2 and Figure 6-2, it is then coming out Figure 7-4.

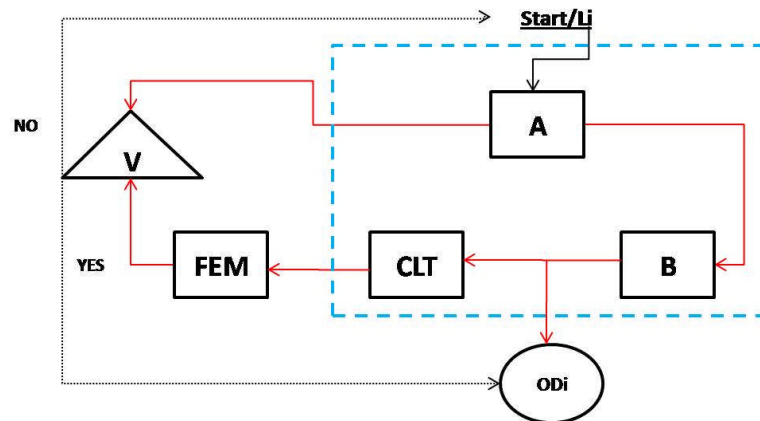


Figure 7-4 – Multi-level Decomposition: Single Level Reviewed

The above Figure 7-4 displays the main blocks, using the same labels, introduced in the basic, intermediate and advanced examples, breaking up the C box. In this way it is possible to group in a single dash rectangle line, A (the Analytical Model), B (the Optimiser), and CLT (Classical Lamination Theory). These are meant to be further defined, using similar terminology of Figure 7-4, as shows Figure 7-4 below.

This picture highlights mainly two groups; the dash rectangle line composed of structural analysis and design study main blocks and the dash circle line with the convergence test, HC, and the approximate model. HC is nothing but the validation test of the 2-level decomposition if not satisfied the program terminates or goes to the starting point, and it works as the hard convergence of Nastran since it is based on the FEM analysis outcome. Then the 2-level decomposition FEM task, part of the C box, is very similar to the Approximate Model concept just described in the previous section. In fact, as described by section 5.1.3, the finite element properties are part of an approximation process according to the homogeneous equivalent idealisation theory. Following what mentioned in 2.7, the numerical model is not defined by all the laminate plies but by a single thickness orthotropic laminate whose effective engineering properties are computed through Wisnom equations (Eq. 2-1).

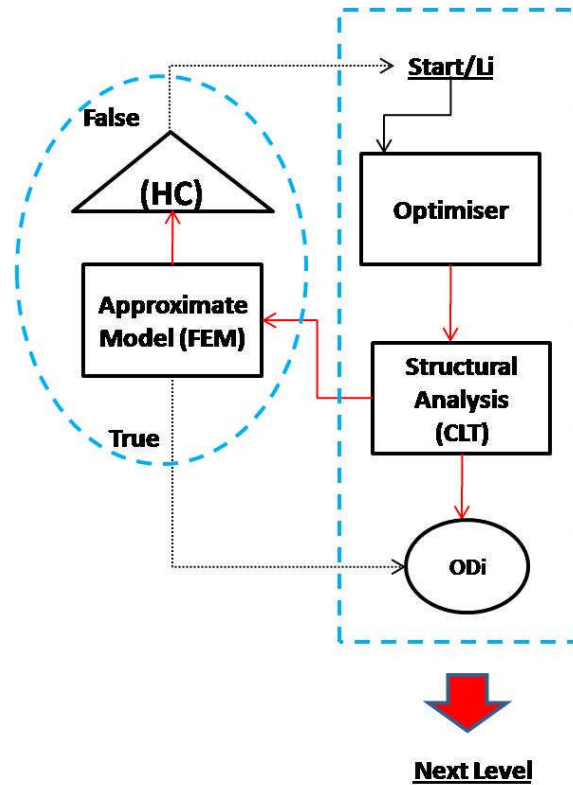


Figure 7-5 – Multi-level Decomposition: Single Level labelled using SOL200 terminology

Finally to compare thoroughly the two main structure methodologies, Nastran SOL200 and 2-level decomposition, Figure 7-4 will make that goal simpler. Using same terminology and labels employed in Figure 7-4 it is now possible to understand their similarities and main differences.

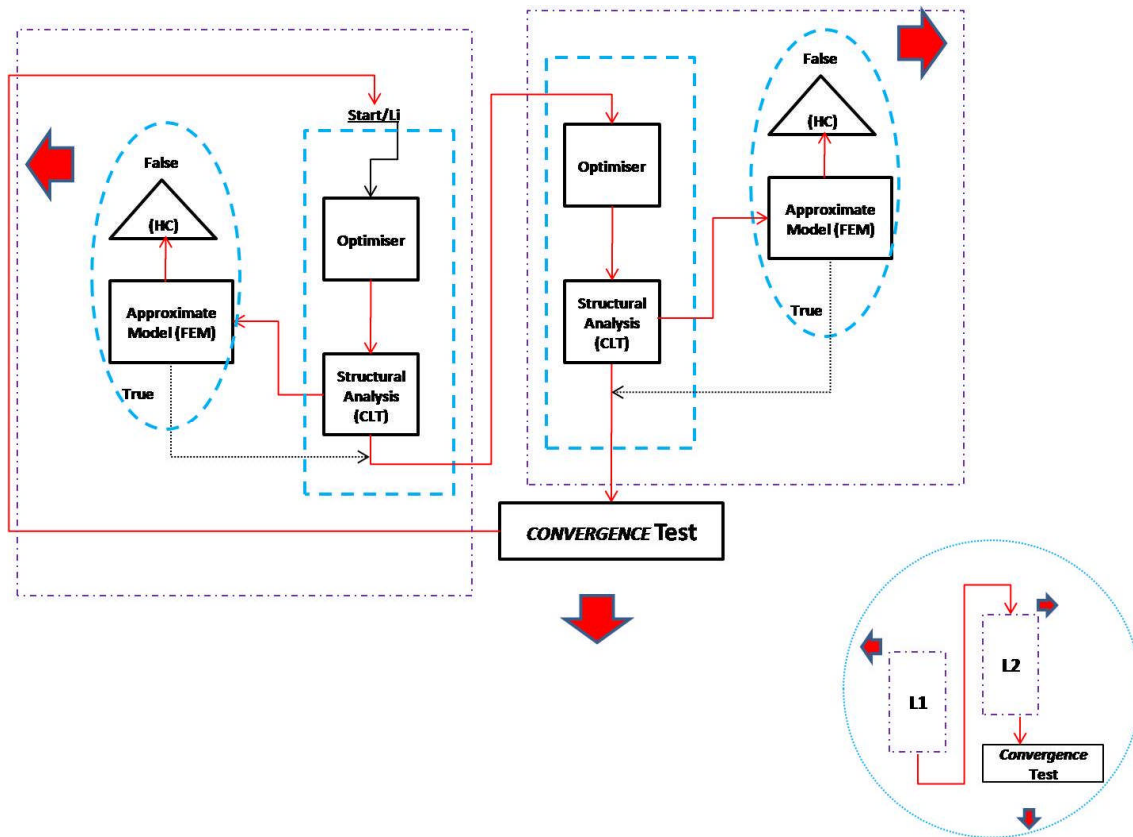


Figure 7-6 – 2-Level Decomposition Main Structure using SOL200 terminology

Figure 7-4, which summary is described by the right bottom detail, outlines:

- Convergence test
- Position where the program terminates (red arrows)
- Structural analysis and Design Study (optimiser) boxes
- Approximate Model

These bullet points, already part of the Nastran SOL200 Figure 7-4 description, show , apart the presence of two levels, L1 and L2, that the two methodologies stop for convergence reasons at intermediate and conclusive levels. Then the SOL200 bases all the optimiser analysis on the FEM of the Approximate Model output while the 2-level decomposition uses the FEM Approximate Model just as part of an extra and external validation box of the entire program that it might be considered a sort of convergence test. The conclusive convergence test of Figure 7-4 is still the last check on the data

before moving to the next iteration but here is based on the results coming from the classical lamination theory instead on the approximate optimiser data as Nastran SOL200 does.

Basically Figure 7-4 and Figure 7-4 say that the main difference is the role of the FEM task. Nastran makes it the core of the optimisation and for this reasons requires to have an approximation of it since it could slow down the overall process when dealing with very big finite element models. On the other hand Basic and Intermediate Examples focus all the analysis on an analytical theory, CLT, but deal with the optimisation task in much more flexible fashion since most of the time the optimum design is coming after visual inspection. This lack of flexibility in Nastran SOL200 will be proved by the next paragraph while searching for a reliable final configuration. Nastran will compensate this “visual analysis capability” by the introduction of a very powerful tool , essential with more complicated applications, as the Sensitivity Analysis task of Figure 7-4 completely missing in the 2-level decomposition method.

7.3 RUN1 and RUN2: Results and Discussion

RUN1 and RUN2 as part of the numerical applications of basic, chapter 5, and intermediate, chapter 6, examples are meant to be solved by Nastran SOL200 and then compared with the final optimum laminate configuration showed by Table 5-13 and Table 6-9, not (Eq. 6-21) since not anymore satisfying the initial condition to be a balanced laminate. This choice, to take the optimum design of the intermediate example of the iteration-4 of Table 6-9, is essential to compare with Nastran SOL 200 same laminate configurations.

Nastran, as described in detail in the last paragraph, to compute an optimum configuration needs to define first a structural model analysis and then to set properly the design study including design variables, objective function and constraints, coupling each other properly. That will be performed using Patran 2005 r3 as pre-processor.

The structural analysis is the same cantilever beam of the 2-level decomposition applications. Size, ply material, and loading case are perfectly the same defined by (Eq. 5-1), (Eq. 5-2), and (Eq. 5-3) only difference is the choice of describing the laminate made up by layers instead of an equivalent orthotropic single ply laminate thickness.

This decision is meant to employ as part of the model analysis a PCOMP card instead the PSHELL as outlined by Table 5-4, so requiring to include all the same initial data optimum data of RUN1 and RUN2, including ply thickness and orientation angles.

The following bullet points list the main Nastran Card used to characterise the structural model:

- PCOMP : ply laminate
- MAT8: single ply material properties
- GRID: wideness and length measures of the cantilever beam
- FORCE: N_x and N_{xy} loading cases

These cards complete the input Nastran bdf file to perform a static analysis, SOL101, and produce stresses and strains of the model. Now Nastran needs to be told which response of the static analysis will be part of the optimisation problem to be run through a second bdf file made by the previous cards and adding few new cards.

In such way the design study will be linked to the model analysis employing the coming Nastran cards:

- DESVAR1: to define design variables
- DVPREL1: defines the design variable to the analysis model parameters: links thicknesses and orientation angles to the PCOMP cards
- DRESP, DCONADD: identify SOL101 responses employed in the design study
- DCONSTR: to characterise the constraints and to link to the same elements of the basic and intermediate example (element 23 and 25)
- DLINK: to work through the design cycle with balanced laminates, each ply group has the same thickness and orientation angles as the BE and IE
- DESMAX: defines the maximum number of design cycles before stopping the SOL200

More details on the above Nastran Cards can be found in any Nastran Quick Reference Guide as [79].

Once defined model analysis (structural analysis FEM model) and design study (optimisation set up) Nastran SOL200 can be run. RUN1 and RUN2 results can be obtained through the different initial data as:

- RUN1 initial data: (Eq. 5-15) for the Basic Example
- RUN2 initial data: Table 6-6 for the Intermediate Example

	RUN1 (Eq. 5-15)		RUN2 Table 6-6	
	Thickness[mm]	O.A. [degree]	Thickness[mm]	O.A. [degree]
Initial Data	$t_1 = 0.1$	$\vartheta_1 = \pm 45^\circ$	$t_1 = 0.508$	$\vartheta_1 = 0$
	$t_2 = 0.3$	$\vartheta_2 = \pm 22^\circ$	$t_2 = 0.508$	$\vartheta_2 = 90$
			$t_3 = 1.016$	$\vartheta_3 = \pm 45$

Table 7-1 – RUN1 and RUN2 Initial Data

The values in Table 7-1, whose conventions are the same of Figure 5-3 and Figure 6-5, are as important not just as data necessary to define the model analysis input file but also as starting value of the optimisation search process of the best laminate configuration.

Taking into account all this information the RUN1 and RUN2 optimisation analyses have been conducted with Nastran SOL200 looking for the optimum symmetric laminate arrangement, ply thicknesses and/or orientation angles, subject to normal and shear strain constraint conditions respectively of 0.4% and 0.6%, the same of basic and intermediate example, under in-plane, N_x and N_{xy} , loading conditions.

According to Table 7-1 the design variables of RUN1 are the thickness and orientation angle of each ply. Dealing with a symmetric laminate of eight overall layers the total number of design variables will be eight, two for each half ply laminate. That is showed by Figure 5-3 where the f06 output SOL200 file data is gathered with the design history.

Figure 5-3 does not outline clearly the trend of the main quantities involved in the entire optimisation process, for that reason it has been thought valuable to use support plots of the same f06 output data but displayed through Matlab. These plots will be employed from now onwards for RUN1 and RUN2 to understand and discuss the Nastran SOL200 computations better.

The categories of the plots will be mainly made of objective function, design variables and Nastran SOL200 constraint value that it is expected to be greater than 0 and with a negative slope over the design cycle [78].

OBJECTIVE AND MAXIMUM CONSTRAINT HISTORY				
CYCLE NUMBER	OBJECTIVE FROM APPROXIMATE OPTIMIZATION	OBJECTIVE FROM EXACT ANALYSIS	FRACTIONAL ERROR OF APPROXIMATION	MAXIMUM VALUE OF CONSTRAINT
INITIAL		2.580640E+05		2.684457E+00
1	7.592144E+03	7.741920E+03	-1.934609E-02	6.334725E+04
2	7.745015E+03	7.745026E+03	-1.513068E-06	6.794705E+04
3	7.745026E+03	7.745026E+03	0.000000E+00	6.324129E+04
4	9.236515E+03	9.236259E+03	2.770162E-05	3.744316E+04
5	2.279902E+04	2.279816E+04	3.735224E-05	2.674657E+03
6	2.866970E+04	2.867451E+04	-1.678320E-04	1.279294E+03
7	3.781062E+04	3.781151E+04	-2.355433E-05	5.227051E+02
8	4.931394E+04	4.930965E+04	8.706144E-05	2.174718E+02
9	6.408396E+04	6.408600E+04	-3.181760E-05	8.774601E+01
10	7.335343E+04	7.328148E+04	9.818733E-04	5.801481E+01
11	9.171845E+04	9.170106E+04	1.896448E-04	2.377836E+01
12	1.072604E+05	1.072171E+05	4.034600E-04	9.976557E+00
13	1.320184E+05	1.320205E+05	-1.621433E-05	4.030804E+00
14	1.320205E+05	1.320205E+05	0.000000E+00	4.030804E+00

Figure 7-7 – RUN1 Nastran SOL200 Objective and Maximum Constraint History

DESIGN VARIABLE HISTORY														
INTERNAL DV. ID.	EXTERNAL DV. ID.	LABEL	INITIAL	:	1	:	2	:	3	:	4	:	5	:
1	1	DV_TH:1	5.0000E-02	:	5.0000E-03	:	5.0004E-03	:	5.0004E-03	:	6.1561E-03	:	1.6667E-02	:
2	3	PCOMP_3	1.5000E-01	:	1.0000E-03	:	1.0020E-03	:	1.0020E-03	:	1.0020E-03	:	1.0020E-03	:
3	10	PCOMP:10	2.2000E+01	:	1.0000E+01	:	1.2206E+01	:	1.0000E+01	:	1.0000E+01	:	1.3564E+01	:
4	12	PCOMP:12	4.5000E+01	:	0.0000E+00	:	1.2162E-08	:	1.2162E-08	:	1.2162E-08	:	1.2162E-08	:
5	2	PCOMP_2	5.0000E-02	:	5.0000E-03	:	5.0004E-03	:	5.0004E-03	:	6.1561E-03	:	1.6667E-02	:
6	4	PCOMP_4	1.5000E-01	:	1.0000E-03	:	1.0020E-03	:	1.0020E-03	:	1.0020E-03	:	1.0020E-03	:
7	9	PCOMP_9	-2.2000E+01	:	-1.0000E+01	:	-1.2206E+01	:	-1.0000E+01	:	-1.0000E+01	:	-1.3564E+01	:
8	11	PCOMP:11	-4.5000E+01	:	0.0000E+00	:	-1.2162E-08	:	-1.2162E-08	:	-1.2162E-08	:	-1.2162E-08	:
INTERNAL DV. ID.	EXTERNAL DV. ID.	LABEL		:	6	:	7	:	8	:	9	:	10	:
1	1	DV_TH:1	2.0906E-02	:	2.7938E-02	:	3.7215E-02	:	4.8667E-02	:	4.6039E-02	:	5.2331E-02	:
2	3	PCOMP_3	1.3171E-03	:	1.3659E-03	:	1.0000E-03	:	1.0000E-03	:	1.0755E-02	:	1.8737E-02	:
3	10	PCOMP:10	1.3565E+01	:	1.3564E+01	:	1.3564E+01	:	1.3561E+01	:	1.9306E+01	:	1.9304E+01	:
4	12	PCOMP:12	1.2162E-08	:	1.2162E-08	:	1.2162E-08	:	1.2162E-08	:	1.2162E-08	:	1.2162E-08	:
5	2	PCOMP_2	2.0906E-02	:	2.7938E-02	:	3.7215E-02	:	4.8667E-02	:	4.6039E-02	:	5.2331E-02	:
6	4	PCOMP_4	1.3171E-03	:	1.3659E-03	:	1.0000E-03	:	1.0000E-03	:	1.0755E-02	:	1.8737E-02	:
7	9	PCOMP_9	-1.3565E+01	:	-1.3564E+01	:	-1.3564E+01	:	-1.3561E+01	:	-1.9306E+01	:	-1.9304E+01	:
8	11	PCOMP:11	-1.2162E-08	:	-1.2162E-08	:	-1.2162E-08	:	-1.2162E-08	:	-1.2162E-08	:	-1.2162E-08	:
INTERNAL DV. ID.	EXTERNAL DV. ID.	LABEL		:	12	:	13	:	14	:	15	:	16	:
1	1	DV_TH:1	3.9356E-02	:	3.4403E-02	:	3.4403E-02	:	3.4403E-02	:	3.4403E-02	:	3.4403E-02	:
2	3	PCOMP_3	4.3737E-02	:	6.7913E-02	:	6.7913E-02	:	6.7913E-02	:	6.7913E-02	:	6.7913E-02	:
3	10	PCOMP:10	1.6324E+01	:	1.6305E+01	:	1.6305E+01	:	1.6305E+01	:	1.6305E+01	:	1.6305E+01	:
4	12	PCOMP:12	1.2162E-08	:	1.2162E-08	:	1.2162E-08	:	1.2162E-08	:	1.2162E-08	:	1.2162E-08	:
5	2	PCOMP_2	3.9356E-02	:	3.4403E-02	:	3.4403E-02	:	3.4403E-02	:	3.4403E-02	:	3.4403E-02	:
6	4	PCOMP_4	4.3737E-02	:	6.7913E-02	:	6.7913E-02	:	6.7913E-02	:	6.7913E-02	:	6.7913E-02	:
7	9	PCOMP_9	-1.6324E+01	:	-1.6305E+01	:	-1.6305E+01	:	-1.6305E+01	:	-1.6305E+01	:	-1.6305E+01	:
8	11	PCOMP:11	-1.2162E-08	:	-1.2162E-08	:	-1.2162E-08	:	-1.2162E-08	:	-1.2162E-08	:	-1.2162E-08	:

Figure 7-8 – RUN1 Nastran SOL200 Design Variable History

Figure 7-7 and Figure 7-8 SOL200 data plots are grouped in the following pictures;
Figure 7-9, Figure 7-10, Figure 7-11, and Figure 7-12.

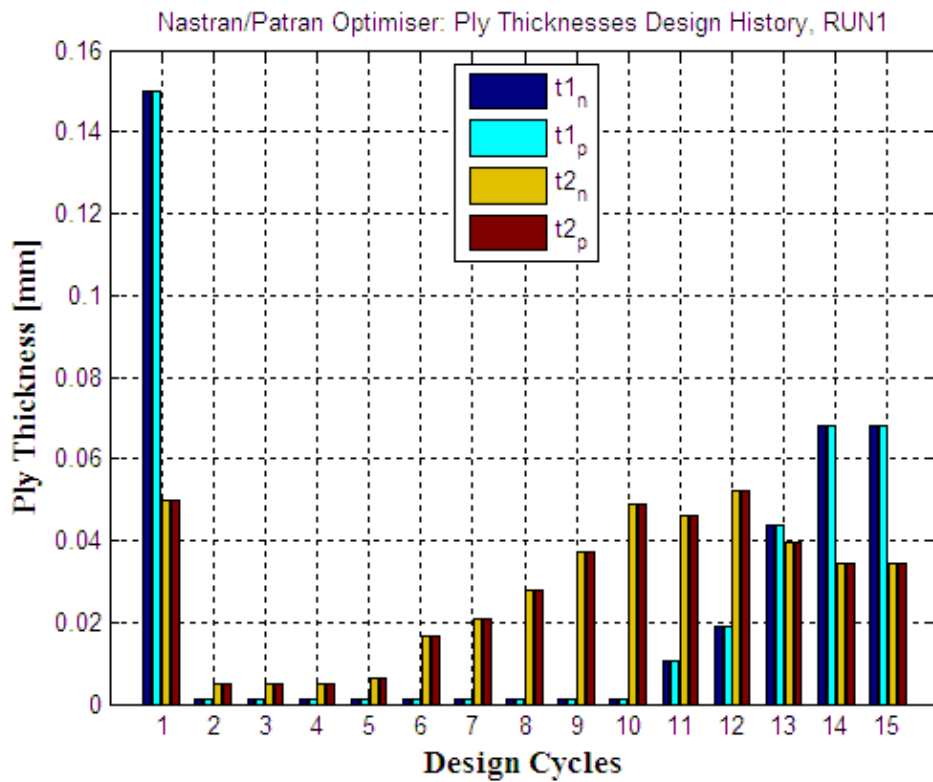


Figure 7-9 – Nastran SOL200 RUN1: Ply Thickness design variables Design History

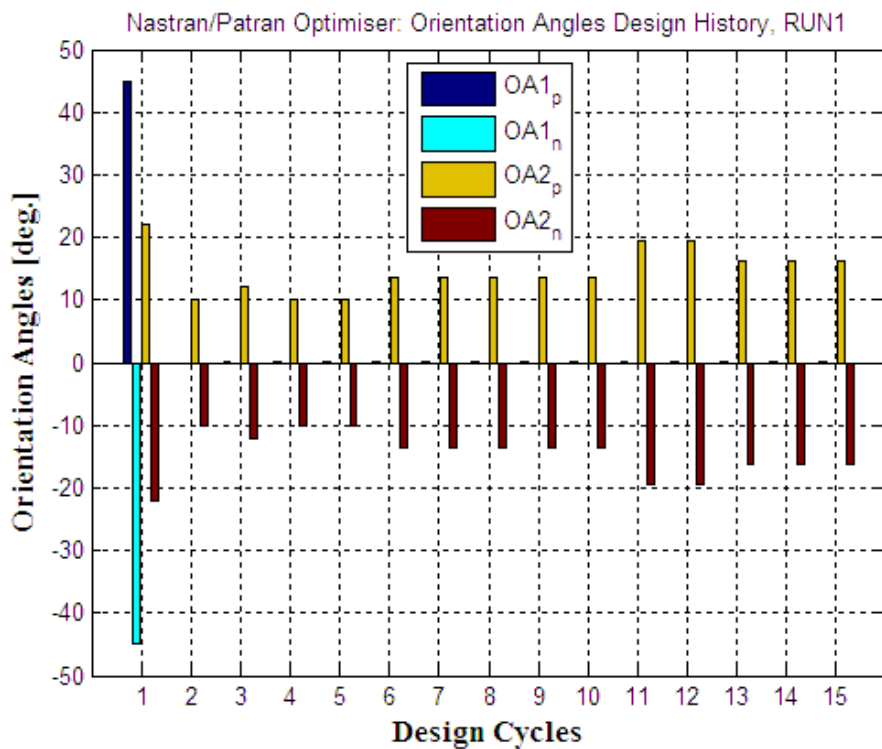


Figure 7-10 – Nastran SOL200 RUN1: Orientation Angles design variables Design History

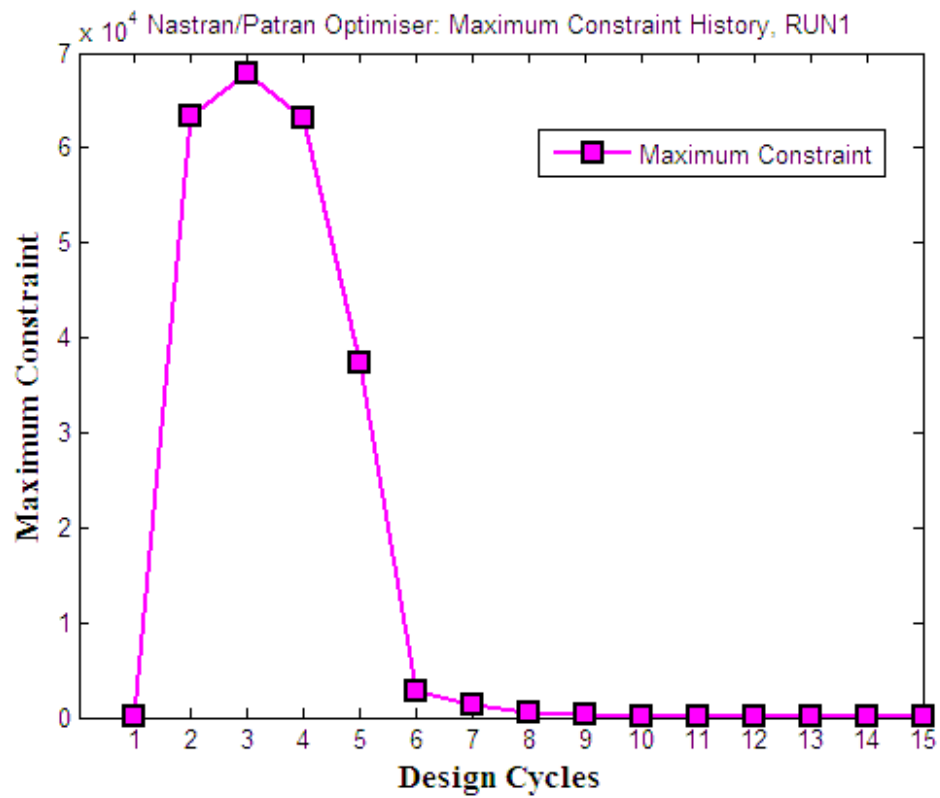


Figure 7-11– Nastran SOL200 RUN1: Maximum Constraint Design History

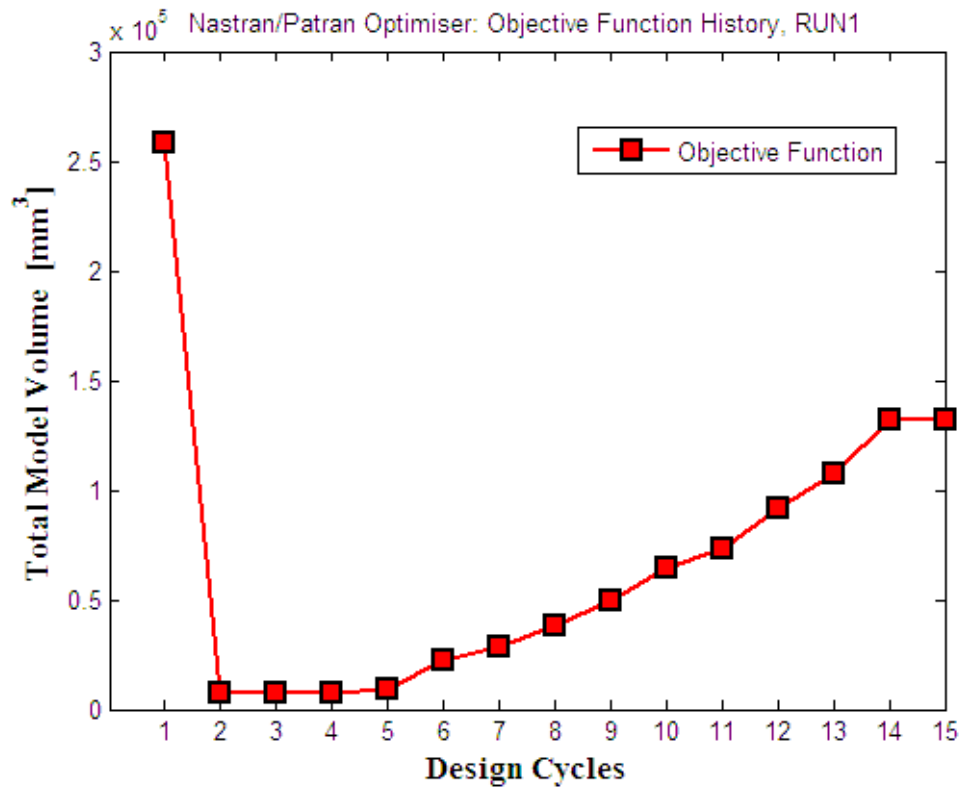


Figure 7-12 – Nastran SOL200 RUN1: Objective Function Design History

The first two bar-plots, Figure 7-9 and Figure 7-10 show half laminate design variables (where the underscore p label associates with positive orientation angle ply and vice-versa the underscore n is linked to the negative ply angle), thickness and orientation angles at any iteration. The total design cycle counts the initial value too; in fact Figure 7-7 and **Error! Reference source not found.** cycle numbers are fourteen and not fifteen. Coherently to what just stated, Figure 7-9 and Figure 7-10 first design cycle identifies the RUN1 initial value of Table 7-1. The trend of these two plots is really surprising, and that will be confirmed later in the respective Objective and Constraint function plots too, in particular the one of the thicknesses since the overall laminate thickness, meant to decrease to obtain a minimum laminate volume, seems to increase around the design cycle 5 up to the end. That is due to the fact the Nastran SOL200, already at the second iteration, drops to thickness values very small, too small to go further coherently to the constraints, and so then it has just to rise up to the optimum design of the design cycle 15. This unexpected plot slope is balanced by the fact that from the initial value, Figure 7-9 design cycle one, to the last iteration the overall and single ply thickness has been decreased anyway. Moreover, that impression is confirmed by the fact that the normal and shear strain constraints, summarised by the maximum constraint plot distribution of Figure 7-11, merged in a unique index by Nastran, are fully greater than zero. That sounds correct since the increase of thickness is meant to produce lower strains. The orientation angle bar plot in Figure 7-10 gives a quite consistent pattern showing that one ply group is meant to be closer to zero while the other stays stable around 16.3° degrees. A similar optimum laminate configuration design was offered from the 2-level decomposition basic example too. In fact, the final laminate orientation angle configuration described in Table 5-13 is 0.4 radians, 22.9° deg., and 0.1 radians, 5.72° deg. Then the objective function plot of Figure 7-7 substantiated a completely undesired positive global slope trend but mainly the volume of the last design cycle is much greater than the first initial one. Basically the objective function instead being minimised and decrease over the iterations does the opposite, but behaves correctly overall according to what already mentioned about the bar thickness plot effect in Figure 7-9.

Finally, SOL200 RUN1 optimisation results just described show the same assumption made in the basic example 2-level decomposition approach, since through the design

cycles the laminate configuration is consistent to the initial assumption of a symmetric balanced laminate. In fact, the design variables, thickness and orientation angle, change coupled thanks to the use of DLINK and DVPREL1 Nastran cards, pivotal to perform the RUN2 optimisation too. Then, the initial data of SOL200 RUN1, the same adopted in the Basic Example multi-level decomposition Table 5-7, , allowed to achieve soft and hard convergence as the Nastran output “f06” file in Appendix C can confirm.

Applying what just learned from RUN1 investigation to RUN2 will allow interpreting Nastran SOL200 result data gathered in Figure 7-13 and Figure 7-14.

OBJECTIVE AND MAXIMUM CONSTRAINT HISTORY				
CYCLE NUMBER	OBJECTIVE FROM APPROXIMATE OPTIMIZATION	OBJECTIVE FROM EXACT ANALYSIS	FRACTIONAL ERROR OF APPROXIMATION	MAXIMUM VALUE OF CONSTRAINT
INITIAL		6.554826E+05		8.432800E-01
1	7.535408E+05	7.535925E+05	-6.858813E-05	7.188942E-01
2	6.272929E+05	6.275654E+05	-4.342177E-04	5.303313E-01
3	5.403359E+05	5.405122E+05	-3.261952E-04	3.867411E-01
4	4.542359E+05	4.542213E+05	3.212916E-05	2.638380E-01
5	4.542213E+05	4.542213E+05	0.000000E+00	2.638380E-01

Figure 7-13– RUN2 Nastran SOL200 Objective and Maximum Constraint History

DESIGN VARIABLE HISTORY										
INTERNAL DV. ID.	EXTERNAL DV. ID.	LABEL	INITIAL	1	2	3	4	5		
1	1	DV_TH:1	2.5400E-01	5.0800E-01	4.7633E-01	4.1790E-01	3.5102E-01	3.5102E-01	:	:
2	3	PCOMP_3	2.5400E-01	7.6035E-02	1.0032E-02	1.0000E-03	1.0000E-03	1.0000E-03	:	:
3	2	PCOMP_2	2.5400E-01	5.0800E-01	4.7633E-01	4.1790E-01	3.5102E-01	3.5102E-01	:	:
4	4	PCOMP_4	2.5400E-01	7.6035E-02	1.0032E-02	1.0000E-03	1.0000E-03	1.0000E-03	:	:

Figure 7-14– RUN2 Nastran SOL200 Design Variable History

The SOL200 RUN2 data plots are collected in the coming up Figure 7-15, Figure 7-16, and Figure 7-17.

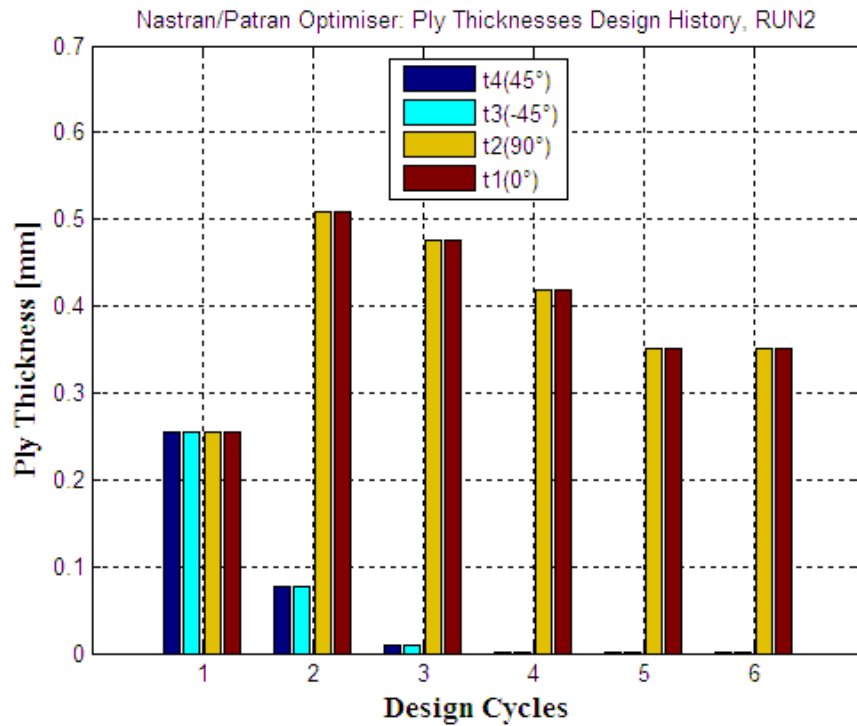


Figure 7-15– Nastran SOL200 RUN2: Ply Thickness design variables Design History

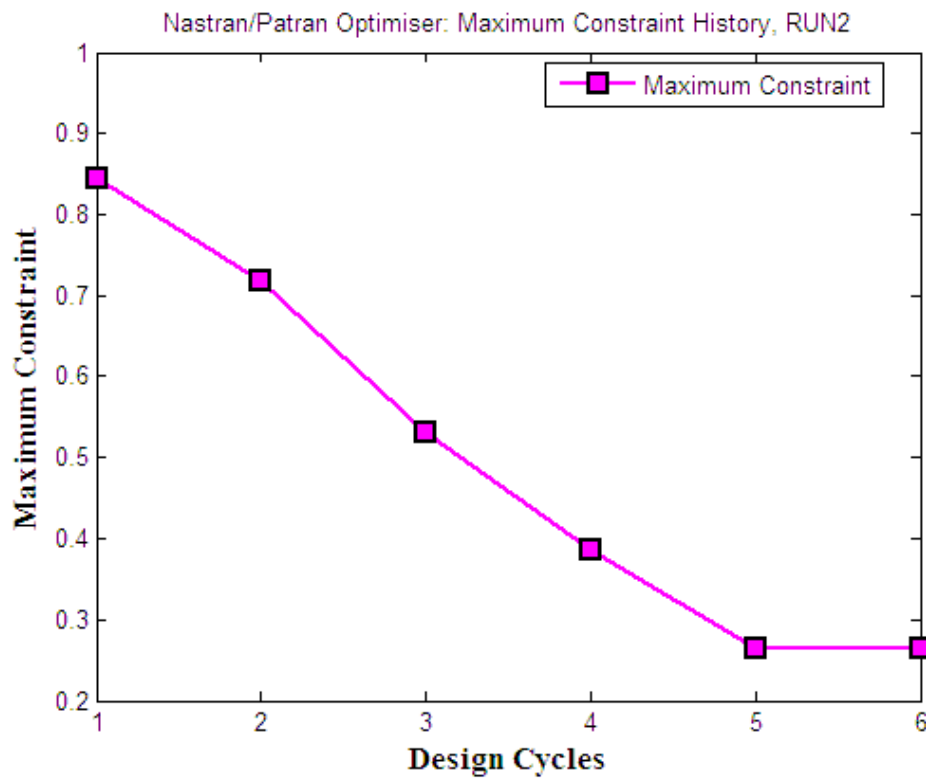


Figure 7-16 – Nastran SOL200 RUN2: Maximum Constraint Design History

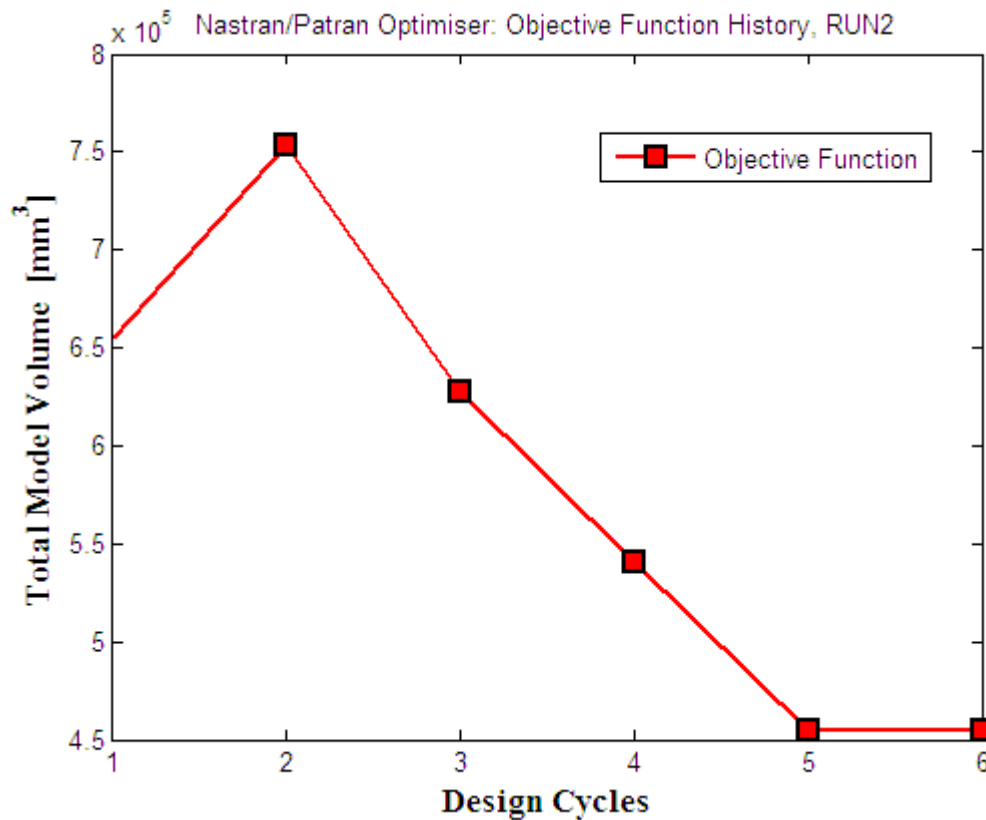


Figure 7-17 – Nastran SOL200 RUN2: Objective Function Design History

Moving through Figure 7-15, Figure 7-16, and Figure 7-17 it comes out that RUN2 has just one set of design variables compared to the RUN1 diagrams described by Figure 7-9, Figure 7-10, Figure 7-11, and Figure 7-12. This particularity is coherent to the optimisation history of the intermediate example already showed in Table 6-10, in fact it had been chosen to maintain through all the iteration steps the same fixed orientation angle configuration. So, to make consistent the successive data comparison, the same laminate configuration has been kept, where according to Figure 6-3 half laminate convention the angles selected are $\vartheta_1 = 0^\circ$, $\vartheta_2 = 90^\circ$, and $\vartheta_3 = \pm 45^\circ$. Then, adopting the same Nastran cards of RUN1, DLINK and DVPREL1, the thickness design variables have been properly coupled to maintain initial assumption of symmetric and balanced laminate. In fact the thickness of the orientation ply angle $+45^\circ$ and -45° varies through the design cycles coupled, as the same do the 90° and 0° . This time the optimisation process over the iterations shows a very reasonable trend, differently to the SOL200 RUN1. In fact, the objective function, Figure 7-17, slope plot is negative since the

overall laminate volume is meant to be minimised. At the same way, Figure 7-16 constraint function behaviour is decreasing through the design cycles proving to use as much as possible design space available. The impression, also proved by the overall number of design cycles much lower than SOL200 RUN1, is the RUN2 is a very much simpler optimisation than RUN1 since the soft and hard convergence has been achieved very soon. Then, taking the Nastran RUN2 optimum design to compare with the multi-level decomposition Intermediate Example result the ply stack group, $\pm 45^\circ$, this time has been taken to zero offering a final laminate configuration of 0° and 90° .

Before comparing the optimum design obtained by the commercial optimiser to the Matlab 2-level decomposition technique results, it is worth validating the Classical Lamination Theory CLT, core of the basic and intermediate structural analysis examples, outcome. Patran, pre-processor of Nastran SOL200, contributes to evaluate the main quantities necessary to perform any CLT analysis: the in-plane stiffness matrix elements see 3.6.1.

Membrane, Bending, and Coupling Matrices						
		Membrane			Bending	
Membrane		6.84E+004	1.64E+004	0.00E+000	-1.9E-003	-4.8E-004
		1.64E+004	1.66E+004	0.00E+000	-4.8E-004	-3.0E-004
		0.00E+000	0.00E+000	1.84E+004	1.22E-004	0.00E+000
Bending		-1.9E-003	-4.8E-004	1.22E-004	4.21E+003	7.35E+002
		-4.8E-004	-3.0E-004	0.00E+000	7.35E+002	7.14E+002
		1.22E-004	0.00E+000	-6.1E-004	-7.7E+002	-1.6E+002

Figure 7-18 – Patran In-plane Stiffness Elements: Membrane Matrix

Figure 7-18 shows the in-plane stiffness matrix, named by Patran as Membrane matrix, of a symmetric laminate characterised by the following half laminate set up, the convention is the same of Figure 5-3:

- $\vartheta_1 = 45^\circ, \vartheta_2 = 22^\circ$
- $t_1 = 0.05, t_2 = 0.15$

These membrane matrix elements of Figure 7-18 are compared to the values coming from the CLT, C box of section 5.1.3, Matlab run displayed in Figure 7-18. The $[A]$ matrix to be matched might be described by the below matrix equation:

$$[A] = \begin{bmatrix} A_{11} & A_{12} & A_{16} \\ A_{12} & A_{11} & A_{26} \\ A_{16} & A_{26} & A_{66} \end{bmatrix} \quad (\text{Eq. 7-1})$$

Comparing Figure 7-18 to Figure 7-18 is confirmed the very good matching of all the elements A_{11} , A_{12} and A_{66} of (Eq. 7-1) obtained through (Eq. 3-44) and (Eq. 3-23).

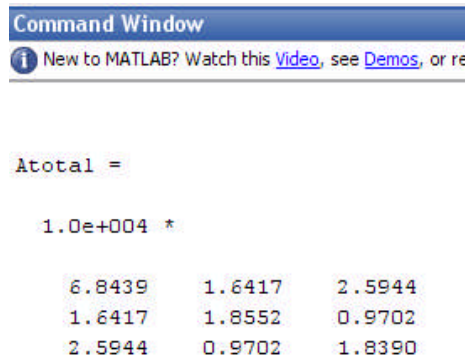


Figure 7-19 – Matlab In-Plane Stiffness Elements: A matrix (Eq. 3-23)

The elements that do not match are A_{16} and A_{26} but that fact is well known and expected since the Matlab program to evaluate the in-plane stiffnesses employs always the positive orientation angles as for in-plane loading case the reduced stiffnesses to be used are only the first three equations of (Eq. 3-45) that rely on positive power of the orientation angles. Basically dealing with in-plane loading conditions, as RUN1 and RUN2 do, all the effective engineering constants needed to evaluate normal and shear strains, see (Eq. 3-45), rely just on A_{11} , A_{12} and A_{66} . Then, the reason of having A_{16} and A_{26} terms not zero is due to the assumption used to calculate these values and to develop the whole 2-level decomposition optimisation process. In fact, the data used in the calculation, thicknesses and orientation angles, are referred to a single stack ply group, for instance, see 5.1.2.1 the orientation angles are not diversified by being

positive or negative and the thickness includes both the positive and the negative ply angle as described in Figure 5-3, consequently that assumption for the terms, A_{11} A_{12} and A_{66} , to the even power produces correct data not for those, as A_{16} and A_{26} whose equations, (Eq. 3-45) (Eq. 3-47) and (Eq. 3-49), include uneven powers.

Once validated the CLT outcome of basic and intermediate examples it is reasonable to correlate the optimum design outcomes of RUN1 and RUN2 obtained with SOL200 (see Appendix C for all input and output files) and 2-level decomposition basic and intermediate examples. To facilitate the back track through the whole work thesis the following Table 5-13, has created.

	Nastran SOL200	2-Level Decomposition
RUN1	Figure 7-18 15 th Design Cycle Column	Basic Example Table 5-13
RUN2	Figure 7-18 6 th Design Cycle Column	Intermediate Example Table 6-9

Table 7-2 – “Nastran SOL200” and “2-Level Decomposition”: Optimum Design RUN1 and RUN2 reference

Following the references reported in Table 7-2 is possible to identify the half laminate design variable quantities obtained at the end of these analyses and compare them all with the next final tables.

RUN1	Thicknesses [mm]				Orientation Angles [degree]			
	t_{1_N}	t_{1_P}	t_{2_N}	t_{2_P}	θ_{1_N}	θ_{1_P}	θ_{2_N}	θ_{2_P}
SOL200	0.0344	0.0344	0.0679	0.0679	-16.3	16.3	-1.2E-8	1.2E-8
BE	0.02	0.02	0.095	0.095	-22.9	22.9	-5.729	5.729

Table 7-3 – Nastran SOL200 and Basic Example (BE) optimum laminate comparison: RUN1

RUN2	Thicknesses [mm]				Orientation Angles[deg]
	$t_{1_{0^\circ}}$	$t_{2_{90^\circ}}$	$t_{3_{(-45^\circ)}}$	$t_{3_{(45^\circ)}}$	<i>Invariants</i>
SOL200	0.351	0.351	0.003	0.003	$\vartheta_1 = 0^\circ, \vartheta_2 = 90^\circ$
IE	0.254	0.254	0.127	0.127	$\vartheta_3 = \pm 45^\circ$

Table 7-4 – Nastran SOL200 and Intermediate Example (IE) optimum laminate comparison: RUN2

Since the convention of the thicknesses between Basic and Intermediate Examples is not the same, it is crucial to highlight soon that Table 7-3 and Table 7-4 consider each thickness as the value of the single ply of the half laminate. Moreover the subscript next to the ply number is just to relate the thickness of the layer with either the positive or the negative orientation angle.

The thorough understanding of these data needs to focus on the overall weight of the final proposed laminate to define which optimisation tool provides the lightest configuration.

In fact, Table 7-5 and Table 7-6 group the results of the optimum laminate thickness obtained for RUN1 and RUN2 using commercial optimiser and the 2-level decomposition technique.

RUN1	Total Laminate THICKNESS [mm]
Nastran SOL200	0.4092
Basic Example	0.46

Table 7-5 – Nastran SOL200 and Basic Example: RUN1 total optimum laminate thickness

RUN2	Total Laminate THICKNESS [mm]
Nastran SOL200	1.404
Intermediate Example	1.524

Table 7-6 – Nastran SOL200 and Basic Example: RUN2 total optimum laminate thickness

These results show that the commercial optimiser can provide lighter laminate configurations. The configuration with variable orientation angles, RUN1 has showed that SOL200 can drop down the weight of the 2-level decomposition methods but taking one couple of orientation angles, Table 7-3, to nearly zero. The same tendency has been displayed by the multi-level decomposition, where the optimum design configuration is a laminate with an orientation angle, much smaller than the other (22.9° against to 5.7°). Then, it is quite important to highlight the different convergence process employed by the two different approaches but offering at the end quite similar output. In fact, SOL200 starts soon to drop the design variables close to the optimum laminate configuration, Figure 7-12 after two iterations has a very steep slope, then slowly climbs back to the best configuration. On the other hand, the 2-level decomposition starts consistently, Figure 5-53 top plot, to move down to the optimum design. Finally the two different approaches reached the optimum laminate through a different number of iterations, 15 the commercial optimiser and just 6 the multi-level decomposition.

RUN2 SOL200, fixed orientation angle arrangement, offers again better results than the 2-level decomposition approach. That was quite expected since the effectiveness index of level 2, Table 6-12, was already warning the final result might have been better using more feasible space of the level 2. Both approaches, this time comes to the optimum design through similar Objective Functions plot pattern, Figure 6-33 and Figure 7-13 and similar number of overall iterations also. Furthermore the Intermediate Example has been based on a much more real life numerical application, in fact, the ply thickness has been chosen being a multiple, and not smaller, of commercial graphite layer whose thickness matches the 0.127 millimetres. These considerations can explain why the intermediate example, RUN2, at the end of all gives a heavier laminate. Then to complete this comparison it is fair to remind the intermediate example might be used at anytime of the design cycles, as 2-variable orientation angle arrangement.

After all for a much clear state of the art of the two different methodologies would have been convenient employ all the 2-level decomposition potentialities requiring doing more RUN. The same recommendation must be done for Nastran SOL200; in fact all its huge potentialities have not taken out mainly due to such trivial example as the cantilever beam under in-plane loadings as described in 5.1.3.

For a thorough appreciation of this outcome it is important to note that the results are meant to be very comparable since run are based on the same assumptions, not just constraints, loads, geometry and material employed, but the same initial laminate configuration that might be crucial. In fact, it might be interesting investigating the SOL200 run with different initial data and eventually lower bound design variable input too. The only relevant difference is linked to the constraint. SOL200 takes into account the strain between each single layer whilst the multi-level decomposition employs the Classical Lamination Theory and the Homogeneous material structural idealisation.

The final impression is that the 2-level decomposition method proved to be more flexible in the way to the optimum design solution, mainly due to graphical visual inspections and mainly to its strength to decompose the number of design variables and constraints using two specific levels. This sort of superimposition technique made a huge difference compared to Nastran SOL200 run.

For a more reliable and satisfactory use of Nastran SOL200 is suggested a numerical application where the use of the approximate model of the structural model has real advantage. In fact, the advance example, not performed by any real life example numbers, it might be a very well posed basis of comparison since the Classical Lamination Theory needs to be reviewed and the Finite Element Theory must come in and eventually not just as part of an external validation tool.

7.4 Summary

This conclusive chapter is essential to wrap up all the concepts introduced in this thesis. In fact, comparing the outcomes of the Basic and Intermediate Example numerical applications to a proper commercial optimiser, Nastran SOL200, enables to understand the reliability of the whole work proposed. Instead running straight Nastran SOL200, it has decided to focus on its main structure adopting specific flow diagrams to compare

with the 2-level decomposition whole process. A specific similarity terminology has been proposed to finally describe the two different methods in the flow charts of Figure 7-3 and Figure 7-6. The considerations that came out proved to have very similar structure based on completely different structural analysis theory. The Finite Element Method for the Nastran SOL200 and the much more limited Classical Lamination Theory for the 2-level decomposition Matlab program.

Then RUN1 and RUN2 cantilever beam applications, of two different laminate configurations, have been run by the commercial optimiser proving the necessity of a proper external sensitivity investigation to identify an optimum set up capable to produce correct outcomes not affected by gradient based numerical convergence issues. To overcome this undesired situation initial thickness ply data and its lower bound value have been recognised as quantities to work on. After a series of Nastran SOL200 runs a final optimum design for RUN1 and RUN2 has been selected to then being compared with the basic and intermediate examples.

The result correlation proved an overall better behaviour for the commercial optimiser despite a series of specific circumstances related to the 2-level decomposition optimisation process.

8 Conclusions

This conclusive chapter comprises two sections. The former, “Objectives and Outcomes”, focuses on the main achievements of the present research comparing to the original objectives. The latter, “Future Work”, provides recommendations and directions of improvement in the next future applications and describes the main framework of a more complex example: The Advanced Example.

8.1 Objectives and Outcomes

The thesis addressed the problem of structural optimisation of laminated components under static in-plane loadings. The methodology employed, Multi-level Decomposition, is specific to deal with thick composites. In fact, the literature review demonstrated that tackling the design of such composite structures requires a specific optimisation methodology capable to deal with high number of design variables, non linear equations, and at the same time flexible to embed in its main structure analytical and numerical structural analysis tools.

Within these boundaries the following objectives have been met:

- Development of a multi-level decomposition (MD) process through Matlab script linkable with external numerical structural solver (Nastran) and supported by internal analytical structural code (Classical Lamination Theory).
- Investigation of different Optimisation techniques (Graphical, Lamination Parameters, Kuhn-Kutta conditions) to be performed by Matlab.

- Understanding of different finite element modelling (FEM) structural idealisation to cope with the thick laminated structures (laminate modeller and homogeneous 2D orthotropic elements).
- Characterised numerically two increasing difficulty examples (Basic, and Intermediate Example) to be performed by the MD methodology. Set and run through numerical applications (RUN1 and RUN2).
- Assessment of the Basic and Intermediate Example results introducing specific, Effectiveness and Reliability, indexes.
- Validation of the internal analytical structural code, CLT, through the Pre-Post Processor, PATRAN.
- Demonstration of the reliability of Multi-Level Decomposition Methodology by comparing the numerical application results of Basic and Intermediate Examples to similar runs performed by Commercial Optimiser (Nastran SOL200) after specific understanding of the code.
- Definition of future work activities proposing the framework of an Advanced Example eventually capable to deal with specific thick composite material issues.

The development of the Multi-Level Decomposition methodology allowed a specific optimisation methods employed at each level of the decomposition, modulated according to the design variables chosen and difficulty of the application. Two numerical applications, Basic and Intermediate Example, have been dedicated to find the optimum design of a symmetric laminate cantilever beam subject just to in-plane loading.

The Basic Example used the Graphical Optimisation tools of Matlab applying the typical design variables such as thicknesses and orientation angles. Then the Intermediate Example considered numerical optimisation conditions with Kuhn- Tucker and Lagrange Multipliers and again Graphical Methods but, using lamination parameters. Lamination parameters are extensional and flexural stiffness terms that are integral quantities of a lamina, or laminate properties. Their use mainly overcame the difficulty to define unique solution in design space of the orientation angles. Finally, in the Advanced Example other optimisation techniques have been described as Sequential

Linear Programming, Gradient Based and Genetic Algorithm. The design variables will be different too, in fact they will be laminate and sub-laminate stiffness coefficients as already pioneered by arthroplasty applications. Unfortunately, it has not been possible to perform any real-life application for this example missing the opportunity to verify the potential of parallel computation, the complexity of out-of-plane loadings and a critic validation of the whole process novelties comparing the outcome to commercial optimisation software.

From Basic through Advanced Example, an increasing difficulty has been applied. So the first two easier applications did not consider any inter-laminar effect, whereas out-of-plane loading has been included in the last more complex numerical example, which justifies higher-order plate theories and 3D global/local finite element approaches.

Nastran SOL200, as commercial optimiser, has participated in the validation process of Basic and Intermediate Example results. After a structural comparison of the two methodologies involved, the numerical outcome proved that SOL200 has been capable to point out a lighter laminate configuration. In fact, the multi-level decomposition techniques proved to be far more flexible and friendly to use than Nastran SOL200. Moreover interacting at any iteration through visual inspection optimisation choice made the whole process a very much understandable process instead the black box Nastran SOL200 run. However, it is reasonable to highlight the limitations present in the multi-level decomposition approach. A clear example is coming from the effectiveness indexes of the Basic, Table 5-14, and Intermediate, Table 6-12, examples. These indexes proved to be an interesting tool to judge the final optimum design of a multi-level decomposition. The ability to have those indexes close to one at each level is the core of the goodness of the whole optimisation. In fact, the poor result of the intermediate numerical application is confirmed by a very high index at level 2. This specific effect is meant to be more relevant and difficult to control once the levels of the decomposition tend to increase, as the Advanced Example meant to have three levels. At the same time this difficulty is directly linked to the correct quantities chose to link one level to the other. In fact, the coupling quantities are a very sensitive task inside the development of a multi-level decomposition process and it requires several iterations before being frozen.

The final impression is the two examples are probably too trivial to allow the commercial software to perform at its best. So, a complete final judgement would have needed first a wider spectrum of runs for Basic and Intermediate Examples, not just RUN1 and RUN2, second, a numerical outcome from the Advanced Example.

In fact, just the Advanced Example is the base framework to finally describe a multi-level decomposition structure capable to deal with the issues of a thick laminated component as the “Side Stay Fitting” attachment requires.

8.2 Future Work

This work has set the basis for dealing with thick composite laminated components subject to in-plane and out-of plane loadings. In fact, the MD methodology presented might be considered the frame work on which improve the following tasks, grouped by main areas;

Transverse/Inter-laminar Effects

- Composite Joining/Lug failure mode (bearing, Shear-out, Net-tension...)
- Abandon Classical Lamination Theory (CLT) for analytical and numerical methods that take into account the Through-Thickness Stress effects (shear and normal)
- Bending and Out-of-Plane loads
- Extend the Failure Criteria adopted so far (Maximum Strain and Stress). Quadratic failure Criteria (Tsai-Hill, Hoffman, Tsai-Wu).
- Consider free-edge effects (de-lamination)
- Adopt specific 3D Finite Element count the effect of inter-laminar shear and normal stresses

Optimisation Procedure

- Setting a new Hierarchical Multilevel Decomposition identifying:

- o The quantities to couple between higher and lower level
 - o The analytical Equations of those quantities to perform optimisation
 - o The specific optimisation search procedure at each level
 - o A robust iteration technique
-
- Stack Sequence optimisation: Genetic Algorithm Search Procedure (integer programming)
 - Consider Run in Parallel to improve efficiency of the calculation (CTL/Platon)

Many of these tasks, partially already explored in this research, are meant to be performed in specific applications as the Advanced Example is expected to be completed and enhanced to a further complex level.

Next part of this section will go in more details to explain the Advanced Example structure adopting the same terminology and flow charts already employed in the Basic and Intermediate examples.

8.2.1 Advanced Example

The last example presented in this research work is expected to move through the boundaries faced in the Basic and Intermediate Examples bringing in new concepts. In fact, the Advanced Example will gather in a single application all the main challenges associated with thick laminate composite materials. Sequence of linear programming, sub-laminates, out-of-plane loadings, interfacial shear stresses, genetic algorithm are the new main concepts launched in the novel multi-level decomposition process proposed. The optimisation technique employed, to cope with all these new tasks, is a three-level decomposition method explicitly developed for the weight optimisation of a thick

laminated to possess desired levels of structural stiffness and maximum strength. Then the convergence technique will be the same of the last examples based on the “g1f2” process described in the previous chapter.

Novelty issue, coming from the out-of-plane loads, is the optimum stacking and orientation of each ply. That it will be achievable introducing, at last level3, a specific optimisation search procedure based on genetic algorithm. But the main innovation will regard the adaptation of coupling concepts not anymore from the higher to lower level but horizontally inside the same level. This apparent awkward technique will sound more natural later on describing the functionality of the sub-laminates level2. Finally the SLP (sequence of linear programming) specific optimiser technique, adopted by the first two levels, will be a very handy approximation used for optimisation instead of the original non linear functions.

Unfortunately will not be described any real life application to support this approach as the Basic and Intermediate Examples did through plots, effectiveness and reliability indexes. In particular, working for the first time with three-dimensional finite element models would require specific numeric bench mark to verify the accuracy of the numerical outcome and its correspondence with the analytical solution. Basically the same of what performed in the Validation Box earlier on.

But, despite the no numerical application offered at this time, it is believed that additional efficiency is achieved in the optimum search process by using at each level design variables and optimisation search procedures improved compared to the previous. Moreover the potential enhanced methodology introduced in the advanced example is confirmed by the higher level of experience accumulated so far through the earlier applications.

The next pages will explain the whole 3-level decomposition pointing out the coupling quantities between one level and the others as main criteria to prove data flow harmonisation of the entire process.

8.2.1.1 Optimisation Procedure

The 3-level decomposition structure employed in the Advanced Example is specifically suited to enable at each level different structural idealisation and optimisation search procedures by applying vertical and horizontal sensible coupling quantities .

The data flow, as Figure 8-1 displays, follows the same principles of the previous 2-level decomposition methods.

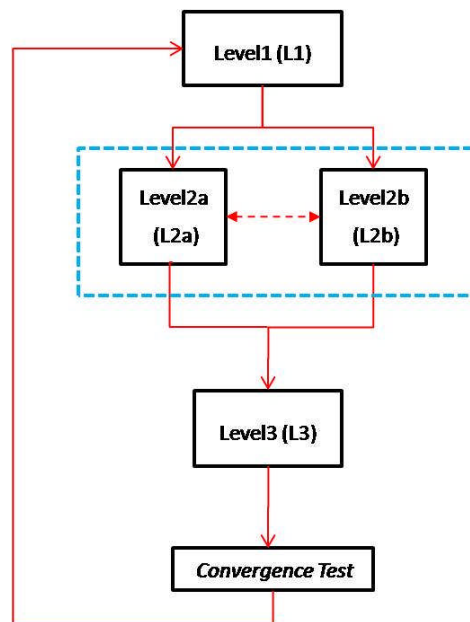


Figure 8-1 – Multilevel Decomposition structure: Advanced Example three-level decomposition

It starts at the very top, in level1, and then flows down to the lower levels up to the final convergence test set, once satisfied the usual required conditions, to goes up again to start new design cycle iteration. Level2 is broken down into two sub-levels according to the principles of the sub-laminates already described, though for in-plane loadings, in section 3.6.4. Each level structure is basically the same of the previous examples as Figure 8-2 says.

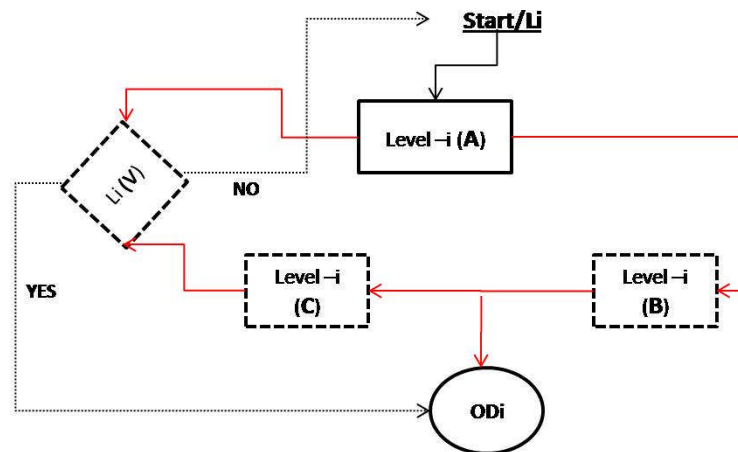


Figure 8-2 – Advanced Example: Level Structure

The main differences are just inside those boxes highlighted with dash solid lines. In fact according to the following former meaning labels:

- **A:** Analytical model
- **B:** Optimiser
- **C:** Numerical Model (reviewed CLT and FEM)
- **V:** Validation Test (Constraints Check)
- **OD:** Optimum Design

Boxes B, C and V will be different.

First, the optimisation techniques will rely less on graphical optimiser since the number of design variables will be overall greater than two, minimum amount to be dealt just by visual inspection. So new methods, still friendly to be developed by Matlab, as sequential of linear programming, gradient based and genetic algorithm will be brought in to deal efficiently with out-of-plane loading effects and design variables that might be different as sub-laminate stiffness coefficients $(C_{ij})_{(k)}$.

Second, still the out-of-plane issue will lead to review the classical lamination theory principles and consequently the finite element modelling techniques, as anticipated in section 2.7 and 2.8, taking into account the inter-laminar consequences.

Finally, associated to the B and C box changes the validation test, box V, at the end of the level data-flow is meant to be enhanced. In fact, will count in not just a comparison

of the Nastran outcome to the inequality threshold constraint value; see Figure 5-36, but the reliability and effectiveness indexes introduced earlier by equations (Eq. 5-22), (Eq. 5-23), (Eq. 5-24), and (Eq. 5-25). That will enable the user to monitor in real time, not just at the end of the entire process, but at a level stage, the quality of the whole optimisation going on.

Next paragraph will explode these developments at each level of the whole three-level decomposition optimisation method highlighting objective functions, inequality constraints, design variables and optimisation techniques.

8.2.1.2 Level Descriptions and Coupling Quantities

The design optimisation analysis to start requires formulating the objective and constraint functions in terms of the respective level design variables. The structural model idealisation will be chosen in connection to the type of loading cases. Then this choice will determine the finite element model and analytical equation (reviewed CLT and FEM) employed inside the box C of Figure 8-2. All those data are grouped and explained by Table 8-1.

The presented three-level optimisation procedure of Table 8-1 has the important characteristic that moving from left, level1, to right, level3, describes quantities that tend to be more particular and less general, a Macro-Micro process. This specific attribute is clearly proved by the below the structural model evolution in Figure 8-3.

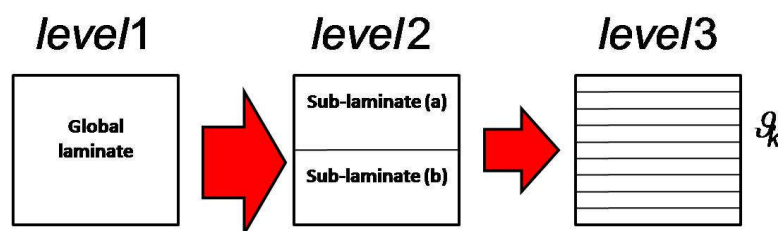


Figure 8-3 – Advanced Example: Structural Model Evolution

Figure 8-3 describes well the convergence process of the advanced example optimisation structure. In fact, it is not just a peculiarity of the “structural model” but, looking down Table 8-1, it is similar for the “design variables” also.

Therefore it is clear that Level1 is based on the whole laminate characteristics then level2 frame swaps to a sub-laminate basis. Finally, level3, the most micro step into the laminate concept, is focused on the single ply.

ADVANCED Ex. Optimisation Level	Level 1	Level 2	Level 3
Structural Model	Homogeneous Orthotropic idealisation	Sub-Laminates Homogenous Orthotropic	Individual Plies
Objective	$(weight)_{min}$	$(Strain\ Criterion)_{MAX}$	$(InterfacialShear)_{MAX}$
Design Variables	t_r = ply thickck.	$(C_{ij})_{(k)}$ = sub-laminates stiff.	$\theta_{(k)}$ = angle of sub-laminates
Constraints	Maximum Strain Criterion	Interfacial Shear stress between sub-laminates	In-plane and out-of-plane stiffness
Loadings	In-plane	Out-of-plane	In-plane and Out-of-plane
FEM	Classic FEM	3D sub-laminate FEM	3Ddetailed FEM
Optimisation Search Procedure	Sequential Linear Program. (SLP) Matlab	Gradient Based/SLP Matlab	Genetic Algorithm. Matlab

Table 8-1 – Multi-level Decomposition Level Description

Describing in detail the main data flow process it starts with the orthotropic symmetric and balanced laminate thicknesses taken as design variables of the weight objective function to be minimised under in-plane loading. Since the loading is in plane as the previous examples, the validation test box will be supported by a classic finite element

model theory. In the second level, sub-laminate level, the laminate is divided into sub-laminates and the stiffness coefficients of the orthotropic sub-laminates are taken as design variables to optimise the laminate out-of-plane stiffness under out-of-plane loading while keeping, invariants, the overall in-plane performance, (Eq. 3-52), as optimised in level1. At this level a reviewed CLT and FEM will need to take into account the inter-laminar effects.

In the third level the optimum orientation design coming from the previous level2 is the input data to determine the optimum stack sequence to maximise the laminate strength according to both intra-lamina and inter-lamina theories, section 2.8, while maintaining the overall sub-laminate stiffness coefficient values as determined during the previous level2.

As described the whole procedure applies the same level concepts of variables and invariants already faced during basic and intermediate examples. The optimum design of the previous level optimisation search becomes invariant at the next level and that is the data flow theory between one level and the other. That data flow characteristic is broken down in Table 8-2, highlighting the coupling quantities through an entire design iteration cycle as made in Table 6-4 and Table 6-5 for the earlier 2-level decomposition analysis.

	Level1	Level2	Level3
Invariants	$\mathcal{G}_k, \mathcal{G}_{k,SS}^*$	$(C_{ij})_L$	$\mathcal{G}_k$
Optimum Design	t_r	$(C_{ij})_{(K)}$	$\mathcal{G}_{k,SS}^*$
Coupling Quantities	$(C_{ij})_L$	$\mathcal{G}_k$	$\mathcal{G}_k, \mathcal{G}_{k,SS}^*$

Table 8-2 – Advanced Example: Coupling Quantities Structure

The above Table 8-2 is the platform to developing the whole 3-level decomposition process through Matlab programming. As described by the basic example, Table 6-4, the optimum design at one level, expressed in terms of the design variables, are the coupling quantities to be passed on the next level becoming invariants at that level. When the application structure is becoming more elaborated the coupling quantities become a more evolution quantity, not necessarily the optimum design. In fact at level1 of Table 6-4 , once selected as optimum design the thicknesses of the laminate, t_r , using the orientation angle invariants, ϑ_k , it is possible to evaluate the laminate stiffness coefficients, $(C_{ij})_L$. These laminate stiffness coefficients become automatically invariants at level2 and are necessary information to be used alongside the laminate thicknesses, to evaluate the sub-laminate stiffness coefficients. That will be possible applying the sub-laminate equation (Eq. 3-45) and other similar equations counting the out-of-plane effects, and so transferring data horizontally from one sub-laminate to the other. Once successfully performed that horizontal transfer of data, based on laminate and sub-laminate stiffness coefficients, the orientation angles of each sub-laminates can be evaluated and then those continue the vertical data flow to the last level3. Finally the laminate optimum design configuration is nearly defined, since thicknesses and orientation angles of the plies laminate are known, but the stacking sequence, $\vartheta_{k,SS}^*$, is the last missing value. Then, orientation angles and stacking sequence as last coupling quantities of the whole iteration will move back at that the very top of whole process, level1, set to start a new design cycle. This will happen once the convergence test proves that the inequality constraints of the all previous levels are still successfully satisfied.

At the end of the Advanced Example description is worth to highlight the use of modern mathematical optimisation, as genetic algorithm, and more old fashion techniques as sequential linear programming, combined with composite structural analysis methods enhanced by out-of-plane effects will provide a means to successfully handle such whole three-level optimisation.

8.2.1.3 Advanced Example Final Considerations

Three-level decomposition optimisation process has been introduced in this section. The advanced example, though not including any numerical application, has been described in its all main potentialities and differences compare to the previous basic and intermediate examples. The description has been pointed out on a general and specific basis. Then the main technical novelties of the laminate under investigation have been illustrated through the out-of-plane loading, inter and intra-lamina shear stress concepts. The main framework of this example has been supported by a time history outline of the main quantities involved in the entire run. In particular optimum design, invariants and coupling quantities at each level have been expressed as the platform on which build later on the Matlab script needed to develop a real life numeric example.

Fresh particularities contextualised to each level have been introduced. The horizontal coupling quantities of the sub-laminates level, the reviewed CLT and FEM theories, and more useful critic tools, effectiveness and reliability indexes, employed in the validation test are just part of the new means used to get closer to the solution of a thick laminate structure under in and out-of-plane loading cases.

Unfortunately the last word on the applicability and reliability of this advanced example will be left to a numerical example to be validated by commercial optimiser software as the next chapter will try to achieve for the basic and intermediate examples.

References

- [1]
Gurdal Z , Haftka R T and Hajela P, *Design and Optimization of Laminated Composite Materials*. Canada: John Wiley & Sons, 1999.
- [2]
website, ALCAS. <http://www.alcas.eu.com/>. (accessed 2007).
- [3]
Mahmood A. «Design and Analysis of Landing Gear for Composite Aircraft.» Msc Thesis, Cranfield University , 2007.
- [4]
Chen W, Fu W, Biggers S B, Latour R A,. «An Affordable Approach for Robust Design of Thick Laminated Composite Strutcure.», *Optimisation and Engineering* Volume 1, Number 3, September 2000, pp. 305-322(18).
- [5]
Venkataraman S, Haftka R T. *Structural Optimisation: What Has Moore's Law Done for Us?* Gainesville Florida: American Institute of Aeronautics and Astronautics AIAA-2002-1342, 2002.
- [6]
Bhatt V, Ragsdell K M. *Multilevel Decomposition of Complete Vehicle Configuration in a Parallel Computing Environment*. April 1, 1989. Design Productivity Center - University of Missouri Columbia, MO.
- [7]
Keane A J, Nair P B. *Computational Approaches for Aerospace Design*. University of Southampton: John Wiley & Sons, Ltd, 2005.
- [8]
Sobieszczanski-Sobieski J -. «A Linear Decomposition Method for Large Optimization Problems - Blueprint for Development.» NASA Technical Memorandum 83248, Hampton Virginia, February 1982.
- [9]
Sobieszczanski-Sobieski J, Haftka R T. *Multidisciplinary Aerospace Design Optimization: Survey of Recent Developments*. Structural Optimization 14, 1-23, Springer-Verlag, 1997.
- [10]
V9, ALTAIR Hyperworks. *Help Manual*. 2007.

[11]

MSC Software, MD Nastran 2006. *MSC.Nastran 2005 r3 : Design Sensitivity and Optimization User's Guide*. 2006.

[12]

Venkataraman P. *Applied Optimization with MATLAB Programming*. Canada: John Wiley & Sons, 2002.

[13]

Matthews, F L, Davies G.A.O, Hitchings D. and Soutis C. *Finite Element Modelling of Composite Materials and Structures*. Cambridge - England: Woodhead Publishing Limited, 2000.

[14]

Kant T, Swaminathan K. «Estimation of Transverse/Interlaminar Stresses in Laminated Composites - a Seelctive Review and Survey of Current Developments.» *Composite Structures*, 2000: 65-75.

[15]

Klosche, Martin. *Calling a C++ CTL Component from Python using SWIG underLinux/Unix*. Brunswick, Germany: Institute of Scientific Computing, Technical University Braunschweig, 2005.

[16]

Bugling, Boris. *Coupling CTL and Matlab*. Brunswick, Germany: Institute of Scinetific Computing, Technical University Braunschweig, 2007.

[17]

Haftka R T. <http://www.aero.ufl.edu/~haftka>. List of Pubblications (accessed on April 15, 2002).

[18]

vanBloemenWaanders B, Eldered M, Guinta A, Reese G, Bhardwaj M, and Fulcher C., *Proceedings of the 43rd Conference AIAA/ASME/ASCE/AHS/ASC Structures,Structural Dynamics and Material Conference*. Seattle, Washington, April 16-19, 2001. AIAA-2001-1625.

[19]

Lamberti L., Venkataraman S, Haftka R T, and Johnson T F. «Preliminary Design Optimization of Stiffned Panels Using Approximate Analysis Models.» *Internal Journal for Numerical Methods in Engineering*, 2002.

[20]

Wellen H, and Bartholomew P. «Structural Optimization in Aircraft Construction.» *NATO ASI Series, Vol F27, Computer Aided Optimal Design: Structural and*

Mechanical Systems, edited by Mota Soares, C. A., Springer-Verlag, Berlin Heidelberg, 1987: 955-970.

[21]

Jegley D C, Bush H. G, and Lovejoy A E,. «Structural Response and Failure of a Full Scale Sticked Graphite Epoxy Wing.» *Proceedings of the 42nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference* , April 16-19, 2001, Seattle, Washington: AIAA Paper No.2001-1334.

[22]

Poss R. «Natural Factors That Affect the Shape and Strength of the Aging Human Femur.» *Clinical Orthopaedics and Related Rsearch*, 274, 1992: 194-201.

[23]

Fu W, Biggers S B, and Latour R A. «Design Optimization of a Laminated Composite Femoral Component for Hip Joint Arthroplasty.» *Journal of Thermoplastic Composite Materials*, Vol 11, March 1998: 99-112.

[24]

Ha S K, and Jeong J Y. «Design Optimisation of Hip Prosthesis of Thick Laminated Composites by Developing Finite Element Method and Sensitivity Analysis.» *KSME Journal*, Vol. 10, No. 1, 1996: 1-11.

[25]

Conti P, and Cella A. «An Optimal Design of Multilayered Laminates Based on Finite Element Stress Analysis.» *Composite Material Technology*, ASME 1992: PD-Vol. 45, 205-212.

[26]

Sobieszczanski-Sobieski J, and Balling R J. *An Algorythm for Solving the System-Level Problem in Multilevel Optimisation*. ICASE 26 p, HAMPTON, VA: NASA, December 1994.

[27]

Schmit Jr L A ,and Ramanathan R K. «Multilevel Approach to Minimum Weight Design including Buckling Constraints.» *AIAA Journal*, February 1978: Vol. 16,No. 2, pp 97-104.

[28]

Browning Tyson R. «Applying the Design Structure Matrix to System Decomposition and Integration Problems:A Review and New Direction.» *IEEE Transactions on Engineering Management*, August, 2001: Vol 48, No. 3, pp 292-306.

[29]

Andersson J. *On Engineering Systems Design: A Simulation and Optimisation Approach*. Thesis, Linköping, Sweden: Linköpings Universitet, 1999.

[30]

Bhatt V, and Ragsdell K M. *Multilevel Decomposition of Complete Vehicle Configuration in a Parallel Computing Environment*. N89-25205, Columbia, MO: Design Productivity Center, University of Missouri.

[31]

Flynn M J, «Some Computer Organizations and their Effectiveness.» *IEEE Transactions on Computers*, Sept. 1972: Vol. C-21, No. 9.

[32]

Storaasli O, and Bergan P. «Nonlinear Substructuring Method for Concurrent Processing Computers.» *AIAA Journal*, June 1987: Vol.25 , No.6.

[33]

Kowalik J S. *Parallel Computers and their Impact on Numerical Algorithms and Related Software*. Nonlinear Optimisation 1981, NATO conference series, Academic Press, 1982.

[34]

Lootsman F A, and Ragsdell K M. *State-of-the-Art in Parallel Nonlinear Optimization*. #86DPC027, University of Missouri, Columbia: Design Productivity Center Report , Nov. 1986.

[35]

Fahimuddin A K M, Krosche M, Niekamp R, Axmann J K, and Mathies H G. *PLATON - An Environment for Coupling Optimisation and Simulation Codes*. Poster Presentation, Berlin: 15th International Conference on Domain Decomposition Methods, Freie University of Berlin, 2003.

[36]

Krosche M, Niekamp R, and Matthies H. G. *PLATON: A Problem Solving Environment for Computational Steering of Evolutionary Optimisation on the Grid*. Barcelona: EUROGEN , 2003.

[37]

Herencia J E, Weaver P M, and Friswell M I. *Optimization of anisotropic composite panels with T-shaped stiffeners including transverse shear effects and out-of-plane loading*. DOI 10.1007/s00158-008-0227-6, Bristol: Struct Multidisc Optim, Nov. 2007.

[38]

Impelluso J T. *Locally Orthotropic Femur Remodelling*. American Society of Biomechanics. La Jolla, California. August 10-11, 2001

[39]

Johansen L, and Lund E. *Optimisation of laminated composite structures using delamination criteria and hierarchical models*. DOI 10.1007/s00158-008-0280-1, Aalborg, Denmark: Struct Multidisc Optim, Springer -Verlag 2008, April 2008.

[40]

Rachmadini Y, Lim S M, Lee S, Park H C, Lee M K, Kim D K, Hong D B and Joo G. *Equivalent Finite Element Modeling of Thick Composite Structures for Analysis of Composite Hingeless Rotor Hub System*. Key Engineering Materials Volume 261-263, Advances in Fracture and Failure Prevention, pp. 585-590, April 2004.

[41]

Sun C T, and Li S. «Three-Dimensional Effective Elastic Constants for Thick Laminates.» *Journal of Composite Materials*, Vol. 22, July 1988: pp 629-639.

[42]

Postma G W. «Wave Propagation in a Stratified Medium.» *Geophysics*, 20, 1955: pp 780-806.

[43]

Pagano G. W. «Exact Moduli of Anisotropic Laminates.» *Composite Materials*, Broutman, L. J., and Krock, R. H., eds. Vol. 2, *Mechanics of Composite Materials*, Seneckyj, G. P., ed. Academic Press, 1974: pp 23-45.

[44]

Wisnom M. «Important Issues in Modelling the Response and Failure of Fibre-reinforced Composites.» *Material Properties for Finite Element Simulations - Getting it right (Part III - Composite Materials)*. Woburn House, London: Institution of MECHANICAL ENGINEERS, 28 April 2009 .

[45]

Wisnom M R, Khan B, and Hallett S R. «Size Effects in Unnotched Tensile Strength of Unidirectional and Quasi-isotropic Carbon/Epoxy Composites.» *Composite Structures* 84, 2008: pp 21-28.

[46]

Kant T, and Swaminathan K. «Estimation of transverse/interlaminar stresses in laminated compsoites - a selective review and survey of current developments.» *Composite Structures*, Vol. 49, 2000: pp 65-75.

[47]

Puppo A H, and Evensen H A. «Interlaminar Shear in Laminated Composite Under Generalized Plane Stress .» *Composite Materials*, Vol. 4, 1970: pp 204-220.

[48]

Whitney J M, and Sun C T. «A High-Order Theory for Extensional Motion of Laminated Composites.» *Sound and Vibration*, Vol. 30(1), 1973: pp 85-97.

[49]

Pagano N J. «On the Calculation of Interlaminar Normal Stress in Composite Laminate.» *Composite Materials*, Vol. 8, 1974: pp 65-81.

[50]

Hsu P W, and Herakovic C T. «Edge Effects in Angle-Ply Composite Laminates.» *Composite Materials*, Vol. 11, 1977: pp 422-428.

[51]

Tang S. «A Boundary-Layer Theory Part 1: Laminated Composites in Plane Stress .» *Composite Materials*, Vol. 9, 1975: pp 33-41.

[52]

Tang S, and Levy A. «A Boundary-Layer Theory Part 2: Extension of Laminated Finite Strip.» *Composite Materials*, Vol. 9, 1975: pp 42-52.

[53]

Pipes R B, and Pagano N J. «Interlaminar Stresses in Composite Laminates - An Approximate Elastic Solution.» *ASME, Appl. Mech.*, Vol. 41, 1974: pp 668-672.

[54]

Pagano N J. «Stress Fields in Composite Laminates.» *Solids and Struct.*, Vol. 14(4), 1978a: pp 385-400.

[55]

Pagano N J. «Free-Edge Stress Fields in Composite Laminates. .» *Solids and Struct.*, Vol.14, 1978b: pp 401-406.

[56]

Reddy J N. *Theory and Analysis of Elastic Plates and Shells*. College Station, Texas: CRC Press, Taylor & Francis Group, 2007.

[57]

Lu X, and Liu D. «An Interlaminar Shear Stress Continuity Theory.» *Proceedings of the Fifth Technical Conference of the American Society for Composites*. Lancaster, PA: Technomic, 1990. pp 479-483.

[58]

Noor A K, and Burton W S. «Assessment of Computational Model for Multi-layered Composite Shells.» *Appl. Mech. Rev.*, 43(4), 1990: pp 67-97.

[59]

Noor A K, and Burton W S. «Assessment of Shear Deformation Theories for Multi-layered Composite Plates.» *Appl. Mech. Rev.*, Vol. 42(1), 1989: pp 1-13.

[60]

Salamon N J. «An Assessment of the Interlaminar Stress Problem in laminated composites.» *Composite Materials, Supplement 14*, 1980: 177-194.

[61]

Kim J Y, and Hong C S. «Three-Dimensional Finite Element Analysis of Inter-laminar stresses in thick composite laminates.» *Comput. Struct.*, Vol. 40(6), 1991: 1395-1404.

[62]

Putcha N S, and Reddy J N. «Three-Dimensional finite element analysis of layered composite plates.» *Advances in Aerospace Structures and Materials, ASME, Winter Annual Meeting*, 1982: pp 29-34.

[63]

Mao K M, and Sun C T. «A refined global-local finite element analysis method.» *Numer. Meth. Engrg*, Vol. 32, 1991: pp 29-43.

[64]

Hirai I, Wang B P, and Pilkey W D. «An efficient zooming method for finite element analysis.» *Numer. Meth. Engrg.*, Vol. 20, 1984: pp 1671-1683.

[65]

Thomson D M, and Griffin Jr O H «2D to 3D global/local finite element analysis of cross-ply composite laminates.» *Reinforced Plastics and Comp.*, Vol. 9(5), 1990: pp 492-502.

[66]

Miki M, and Sugiyama Y. «Optimum Design of Laminated Composite Plates Using Lamination Parameters.» *AIAA*, Vol. 31, No.5, May 1993: pp 921-922.

[67]

Niu Micheal C Y. *COMPOSITE Airframe Structures - Practical design information data*. Hong Kong: Hong Kong Conmilit Press LTD., 1992.

[68]

Middleton D H. *Composite Materials in Aircraft Structures*. New York: Longman Scientific and Technical , 1990.

[69]

Boresi A P, and Chong K P. *Elasticity in Engineering Mechanics* . New York: Elsevier, 1987.

[70]

Tsai S W and Pagano N J. «Invariant Properties of Composite Materials.» *Composite Materials Workshop*. Technomic Publishing Co., 1968. pp. 233-253.

[71]

Vinson J R and Sierakowski R L. *The Behaviour of Structures Composed of Composite Materials*. Dordrecht: Martinus Nijhoff, 1986.

[72]

Stein S K. *Calculus and Analytical Geometry*. New York: McGraw-Hill, 1987.

[73]

Schmit L A, and Farshi B. «Optimum Laminate Design for Strength and Stiffness.» *International Journal for Numerical Methods in Engineering*, 1973: pp. 519-536. VOL. 7.

[74]

Storaasli O O, and Sobieszczanski-Sobieski J. «On the Accuracy of the Taylor approximation for Structural resizing.» *AIAA Journal* , 12, 1974: pp. 231-233.

[75]

Holland J H. *Adaptation of Natural and Artificial Systems*. Ann Arbor, MI: University of Michigan Press, 1975.

[76]

Goldberg D E, and Samtani M P. «Engineering Optimization via Genetic Algorithm.» ASCE, , 1986 February: pp.471-482.

[77]

Biran A, and Breiner M. *Matlab 6 for Engineers*. Dorchester, Dorset: Prentice Hall, 2002.

[78]

Software, MSC. *MSC.NASTRAN 2005 r3, Design Sensitivity and Optimization User's Guide*. MSC Software, 2005.

[79]

MSC/NASTRAN Version 70.5 : Quick Reference Guide. The MacNeal-Schwendler Corporation, 1992.

Appendix A

The Basic Example, chapter 5, files needed to run the Matlab program are:

Matlab Files

Master file: **BE_Main.m**

Level 1: **obj_m1.m** , **inecon_m1.m**

Level 2: **obj_m2.m** , **inecon_m2.m** / **cons_c2.m** , **obj_f2.m**

Text Files

P_12_c_Mpa_1.dat

thick_OD_1.txt

angl1_c.dat

Nastran input file

Level 1: **NX_mao_Lminus.bdf**

Level 2: **Nxy_miomao_Lminus.bdf**

To run the Basic Example the following software packages need to be installed:

- MSC NASTRAN
- MATLAB

To run a Basic Example Numerical application the below “*to do list*” has to be followed:

- Group all twelve files in the same directory
- Go to 288 and 649 lines in the Matlab script of the master file **BE_Main.m** and update the location of the bdf files and the location of the Nastran executable, **nastranw.exe**, they all need to be in the same folder
- Save in the **angl1_c.dat** file the orientation angle chosen to start the iteration

- Agree the constraint values defined by the variable *Cons1* and *Cons2* inside the BoxA of the level1 matlab code of the main file **BE_Main.m**
- Agree the loads values defined by the variable *Nx* and *Nxy* inside the BoxA of the level1 matlab code of the main file **BE_Main.m** and eventually, if changed, update coherently the two Nastran input file cards
- Open from Matlab editor the master file **BE_Main.m** and Run it

NB: Twelve (12) are the overall number of files of Appendix A.

Appendix B

The Intermediate Example, chapter 6, files needed to run the Matlab program are:

Matlab Files

Master file: **IE_Main.m**

Level 2: **obj_Ex_strain.m** , **inecon2_strain.m** , **obj_G2.m**

Text Files

P_12_c_Mpa_2.dat

thick_OD_2OG.txt

angl1_c_RL_3.dat

Nastran input file

Level 1: **NX_mao_Lminus.bdf**

Level 2: **Nxy_miomao_Lminus.bdf**

To run the Intermediate Example the following software packages need to be installed:

- MSC NASTRAN
- MATLAB

To run an Intermediate Example Numerical application the below “*to do list*” has to be followed:

- Group all nine files in the same directory
- Go to 484 and 1245 lines in the Matlab script of the master file **IE_Main.m** and update the location of the bdf files and the location of the Nastran executable, **nastranw.exe**, they all need to be in the same folder

- Save in the **angl1_c_RL_3.dat** file the orientation angle chosen to start the iteration and do the same in **thick_OD_2OG.dat** to decide the starting value of the invariant t_2
- Agree the constraint values defined by the variable *Cons1* and *Cons2* inside the BoxA of the level1 matlab code of the main file **IE_Main.m**
- Agree the loads values defined by the variable N_x and N_{xy} inside the BoxA of the level1 matlab code of the main file **IE_Main.m** and eventually, if changed, update coherently the two Nastran input file cards
- Open from Matlab editor the master file **IE_Main.m** and Run it

NB: Nine (9) are the overall Number of Files of Appendix B.

Appendix C

Nastran SOL200 files are grouped in two main folders named, according to the same terminology of chapter 7, such as Design Study RUN1 and Design Study RUN2. Each folder contains the specific Nastran input and output files according to section 7.3.

The list of input, *.bdf, and output, *.f06, files follow:

Design Study

RUN1

Input Files: **BE_RUN1.bdf**

Output Files: **BE_RUN1.f06**

RUN2

Input Files: **IE_RUN2.bdf**

Output Files: **IE_RUN2.f06**

NB: Four (4) are the overall Number of Files of Appendix C.