THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par **l'Institut Supérieur de l'Aéronautique et de l'Espace**
Spécialité : Signal, image, acoustique et optimisation

Présentée et soutenue par **Dimitri BETTEBGHOR**
le **9 décembre 2011**

Optimisation biniveau de structures aéronautiques composites

JURY

M. Grégoire Allaire, président
Mme Nathalie Bartoli
M. Gilles Celeux
M. Stéphane Grihon
M. Zafer Gürdal, rapporteur
M. Rodolphe Leriche, rapporteur
M. Joseph Morlier, co-directeur de thèse
M. Manuel Samuelides, directeur de thèse

École doctorale : **Aéronautique - Astronautique**

Unité de recherche : **Équipe d'accueil ISAE-ONERA MOIS**

Directeur de thèse : **M. Manuel Samuelides**
Co-directeur de thèse : **M. Joseph Morlier**

Acknowledgements

Ce travail de thèse a été rendu possible grâce à l'aide et la bienveillance de nombreuses personnes. Je tiens donc à remercier Manuel Samuelides, mon directeur de thèse, pour son soutien, sa sympathique supervision et sa constante volonté d'éléver le niveau scientifique. Son attachement à la théorie, aux mathématiques et à la rigueur a permis d'approfondir certaines thématiques et d'en donner des bases formelles plus poussées. Le corpus scientifique de ce travail lui doit beaucoup. D'autre part, c'est grâce à sa confiance et ses nombreux conseils que j'ai pu réaliser d'intéressantes opportunités académiques (enseignement en Chine, congrès, articles de journaux). Je tiens à exprimer mes vifs remerciements à Stéphane Grihon, mon encadrant industriel à Airbus et promoteur de cette thèse. Son volontarisme et son haut niveau d'exigence tant industrielle que scientifique ont aussi rendu ce travail de thèse ambitieux et passionnant. D'un point de vue professionnel, Stéphane m'a aussi permis d'acquérir certaines responsabilités techniques et scientifiques dans le cadre industriel (encadrement de stages, réprésentation et communication autour de projets européens de recherche). Je tiens de même à remercier Joseph Morlier et Nathalie Bartoli, mes encadrants à l'ISAE-Sup'Aéro et à l'Onera, pour leur soutien, leur accompagnement et leur aide précieuse tant scientifique que professionnelle. Le caractère pluridiscplinaire assez marqué de ce travail leur doit beaucoup. D'un point de vue plus personnel, Joseph et Nathalie m'ont guidé dans mon parcours de futur chercheur. Toutes ces personnes, tout au long de ces trois ans, m'ont accordé leur confiance et ont su adapter leur encadrement aux spécificités d'une thèse industrielle. Qu'ils en soient donc remerciés.

Hormis l'encadrement de nombreuses personnes m'ont aussi permis d'acquérir des connaissances techniques, scientifiques, industrielles et de conduite de

projet. Je tiens à remercier toute l'équipe transnationale d'optimisation de structures à Airbus, en particulier Lars Krog, Lucian Iorga, Peter Kaletta et Franck Seigneuret. Mes prédécesseurs Alban Carpentier de LMS-Samtech Toulouse et Antoine Merval m'ont été d'une grande aide pour prendre en main le sujet de cette thèse et pour leur aide tant scientifique que technique. Cette thèse a été l'occasion d'apprendre à collaborer dans un contexte étendu de différentes entreprises, de différents centre de recherches et nationalités. Je tiens donc à remercier Benoit Colson, Alain Remouchamps, Caroline Raick et Jean-Marie Chudyba de LMS-Samtech Liège, Patrick Sarouille et Anne Gazaix d'Airbus, Charlotte Keisser, Claire Vincent et Simona Wulf stagiaires à Airbus, François-Henri Leroy et François-Xavier Irisarri de l'Onera, Stéphane Alestra, Régis Lebrun et Fabien Mangeant d'EADS-IW, Sergei Morosov et Evgeny Burnaev de DATADVANCE, Antoine Henrot de l'Ecole des Mines de Nancy, Guillaume Alleon et Prasant Rekapalli d'EADS-IW India, Jos Vankan, Bert de Wit et Marco Nawijn du NLR.

Je tiens à exprimer mes chaleureux remerciements à Zafer Gürdal de l'Université de Delft et Rodolphe Le Riche de l'Ecole des Mines de Saint-Etienne qui ont accepté de rapporter ce travail de thèse. Leur lecture attentive du manuscript, leurs nombreuses questions précises ont permis d'améliorer sensiblement la qualité du rapport. Je tiens à remercier Grégoire Allaire de l'Ecole Polytechnique d'avoir accepté de présider le jury et Gilles Celeux de l'INRIA d'avoir apporté sa compétence de statisticien à ce jury.

Enfin je veux remercier mes parents, amis et proches qui m'ont apporté leur soutien. *Last but not least* Marianne, ma chère et tendre, Milena et Anouk, mes filles ont su m'entourer de leur amour, de leur patience. Cette thèse n'aurait tout simplement pas pu aboutir sans elles.

Contents

List of Figures xi

List of Tables xv

Résumé étendu (Extended abstract in french) 1

General introduction 33

1 Preliminaries 37

1.1 Fundamentals of structural optimization 37

1.2 From isotropic to anisotropic structures: composite structures 41

1.3 Modeling of the structure and natural hierarchy of the structural analysis 44

1.3.1 Plane stress problems 45

1.3.2 Stability constraints 46

1.4 Formal description of structural optimization problem in the engineering context 48

1.4.1 Objective function 48

1.4.2 Constraints 49

1.4.3 Formal problem 51

1.5 Objectives and motivations of this thesis 51

1.5.1 Early state of the art 52

1.5.2 Decomposition/bilevel 53

1.5.3 Adaptation to specific composite considerations 54

1.5.4 Approximations of stability constraints 54

1.6 Detailed outline 55

1.6.1 Optimization properties and early presentation of the decomposition/bilevel scheme 55

1.6.2 Adaptation to specific composite representation 55

1.6.3 Approximating discontinuous and derivative discontinuous functions . . . 56
1.6.4 Numerical investigation of monolevel and bilevel schemes . . . 56

2 Monolevel and bilevel formulation 59
2.1 Motivation for decomposition techniques . . . 59
2.1.1 Large-scale problems with natural hierarchy . . . 60
2.1.2 Towards scalable optimization . . . 61
2.2 Basics of differentiable optimization theory . . . 63
2.2.1 Vocabulary . . . 63
2.2.2 Optimality conditions . . . 65
2.2.3 Constraint qualifications . . . 67
2.2.4 Second order conditions for constrained problems . . . 68
2.2.5 Post-optimal sensitivity . . . 69
2.2.6 Original application of post-optimal sensitivities results . . . 78
2.3 Monolevel and block-angular problems . . . 81
2.3.1 Original problem . . . 81
2.3.2 Equivalence between All-In-One problem and Block-Angular problem . . . 82
2.4 Towards decomposition through inexact quadratic penalty . . . 84
2.4.1 Inexact and exact penalties . . . 84
2.4.2 Equivalence of the inexact quadratic penalty for block-angular problems . . . 86
2.5 Towards decomposition . . . 87
2.5.1 Decomposition scheme from the (AIO) initial problem . . . 87
2.5.2 Target Rigidity and Mix schemes . . . 87
2.5.3 MaxMargin and QSD schemes . . . 91
2.5.4 Towards realistic optimization problems . . . 92

3 Optimization properties for composite problems 93
3.1 Motivations and basics of composite laminate theory . . . 93
3.1.1 Introduction . . . 93
3.1.2 Composite Laminate Theory, lamination parameters and feasible stacking sequences . . . 96

3.1.2.1 Composite Laminate Theory for plates 96
3.1.2.2 Lamination parameters 101
3.1.2.3 Compatibility equations 105
3.1.2.4 Feasible stacking sequences 106
3.2 Behavior of the buckling critical factor 107
3.2.1 Generalities on buckling 109
3.2.2 Characterization of the buckling critical reserve factor and properties 114
3.2.3 Concavity of the buckling reserve factor over lamination parameters 117
3.2.4 Influence of the material 119
3.2.4.1 Behavior of λ_{cr} for orthotropic laminates under combined loadings and pure shear 121
3.2.4.2 Behavior of λ_{cr} for non-orthotropic laminates under combined loadings and pure shear 123
3.2.5 Influence of the loading conditions 123
3.3 Towards approximation strategy for discontinuous functions 124

4 Original strategy to approximate discontinuous and derivative discontinuous functions 127
4.1 Motivations 128
4.1.1 Typical behavior of stability constraints 128
4.1.2 From one expert to several experts 130
4.2 Mixtures of probability laws as a way to handle complexity 134
4.2.1 Unsupervised learning and density estimation 135
4.2.2 K-means algorithm 136
4.2.3 Mixture laws 138
4.2.4 Estimation of mixture parameters and mixture components parameters through maximum likelihood : EM algorithm 139
4.2.5 Towards supervised clustering: regression 141
4.3 Posterior probabilities, hard and soft clustering and Bayes optimal boundaries 142
4.3.1 Bayesian estimation of maximum a posteriori: probabilistic and geometric clusterings 142

4.3.2 Practical issues : choice of the number of clusters 145
4.3.2.1 Classical tools : AIC, BIC 145
4.3.2.2 Gap statistics 146
4.4 Building the overall expert for the regression case 147
4.4.1 Building local expert 149
4.4.2 Combining local experts 151
4.4.3 Simple analytical examples 155
4.5 Application of this original strategy to buckling computations approximation 155
4.6 Application to the approximation of buckling critical factor over lamination parameters and forces 160
4.6.1 Piecewise polynomial approximation in lamination space 160
4.6.2 Piecewise polynomial approximation in forces space 160
4.6.3 Piecewise rational approximation in the general case 161
4.6.4 Using directly Rayleigh-Ritz active buckling modes 161
4.7 Numerical results 162
4.7.1 Approximation models for varying stacking 165
4.7.2 Approximation models for varying loading conditions 167
4.7.3 Approximation models in the general case 169
4.7.4 Conclusion on the accuracy of the approximation 169
4.8 Applications to optimization: maximization of the buckling critical load factor 170
4.8.1 Optimization problem 170
4.8.2 Two approaches 171
4.8.3 Comparison of the optimization results 173
4.8.3.1 A combined loading example 173
4.8.3.2 A pure shear loading example 175
4.8.4 Conclusion of the optimization applications 175
4.9 EM estimation of hyperparameters for memory-based methods 176
4.9.1 Radial basis network 177
4.9.2 Weighted Least Squares 179
4.10 Towards the numerical implementation of decomposition/bilevel schemes 181

5 Monolevel optimization schemes **183**

5.1 Formal description of the test case 183
5.1.1 Analytical expression of the objective function 184
5.1.2 Stress computations and constraints 185
5.1.2.1 Stress computation 185
5.1.2.2 Optimization constraints 186
5.1.3 Existence of a minimum and feasible set 188
5.1.3.1 Existence of a minimum 188
5.1.3.2 Feasible set 189
5.2 Sensitivity analysis 191
5.2.1 AIO formulation 191
5.2.2 Direct method and numerical aspects 192
5.2.3 Sensitivity analysis for the AIO formulation 193
5.3 Mono-level methods 196
5.3.1 AIO Method 196
5.3.2 Formal description of AIO 196
5.3.3 Practical implementation 197
5.4 SAND original formulation 197
5.5 Block Angular form (SAND2) 199
5.6 Block Angular form new version (SAND3) 200
5.7 Overall comparison and finding a good initial point 201
5.7.1 AIO 201
5.7.2 SAND2 202
5.7.3 Finding a good initial design 202
5.7.4 Overall comparison 204
5.8 Towards decomposition/bilevel schemes 204

6 Bilevel formulations from bi-stages techniques to real bilevel implementation 207

6.1 Differences between bi-stages and bilevel optimizations 207
6.2 Bi-stages schemes 209
6.2.1 Decomposition based on Fully Stressed Design criterion : StiffOpt 209
6.2.1.1 Brief description of StiffOpt 209
6.2.1.2 Improvement of StiffOpt based on sensitivity 210

6.2.2 Bi-stages or alternating optimization schemes 211
6.2.2.1 Formal description 212
6.2.2.2 Target Rigidity 212
6.2.2.3 Maximum margin (a.k.a Constraint margin) 213
6.2.2.4 Local mass minimization MinMass 213
6.2.2.5 Mix scheme 214
6.2.2.6 Practical implementation as alternating optimizations . 214
6.3 Overall comparison for alternating optimizations implementation 214
6.3.1 StiffOpt 214
6.3.2 Overall comparison 216
6.4 Adaptation of QSD scheme to composite structural optimization problem 217
6.4.1 Adaptation to equality constraints 218
6.4.2 Adaptation to composite specific considerations 219
6.5 Detailed results 222
6.5.1 Practical implementation 222
6.5.1.1 Post-optimal sensitivities and chain ruling 222
6.5.1.2 Optimizers 223
6.5.2 Results for 10 bar truss 225
6.6 Towards a real aircraft structure 226

7 Application to MAAXIMUS panel 229

7.1 Surrogate models of buckling constraints 229
7.1.1 Use of IMAGE for mixed variables for surrogate over lamination parameters 229
7.1.2 Early results 230
7.1.3 Airbus stability tool: COFUS 232
7.2 Practical implementation 233
7.3 Post-identification 238
7.4 Detailed results on MAAXIMUS panel 238
7.5 Conclusion and ways forward 243

Conclusion & Perspectives 245

Collaborations & publications 251

References **257**

Appendices **261**

A More on optimization and numerical approximation **261**

A.1 Quasiconcave functions and homogeneous functions 261

A.2 Rayleigh-Ritz approximation for buckling and sensitivity analysis 262

B Detailed results for monolevel and bilevel schemes for the 10 bar truss **267**

B.1 AIO 268

B.2 SAND2 269

B.3 SAND3 270

B.4 STIFAIO 271

B.5 STISAND 272

B.6 STIFSAND2 273

B.7 StiffOpt 274

B.8 Target Rigidity 275

B.9 MinMass 275

B.10 Mix scheme 277

B.11 MaxMargin 277

Contents

List of Figures

0.1 Principales sollicitations 6
0.2 jacobienne des contraintes globales locales 12
0.3 Présentation de l'algorithme original sur un cas test 1D 23
0.4 Approximation linéaires par morceaux des fonctions discontinues et de dérivée discontinue 24
0.5 Fonction à approcher : facteur de réserve de flambage pour des séquences orthotropes 25
0.6 Exemple d'application de l'algorithme original pour l'approximation de calcul de flambage 25
0.7 Cas test : treillis 10 barres 26
0.8 Comparaison des schémas 28
0.9 Comportement erratique de MaxMarge 29
0.10 Comparaison de QSD et du problème non décomposé 30
0.11 Panneau Maaximus 31

1.1 Distribution of composite material in the A380 42
1.2 Laminates 42
1.3 Evolution of the composite proportions in Airbus aircrafts 43
1.4 Main loads and impact regions over an Airbus aircraft 46

2.1 Profile of the Jacobian of constraints for a fuselage panel optimimzation 62

3.1 Dependence of the laminate stiffness moduli over Miki's triangle 102
3.2 Natural logarithm of the number of feasible stacking sequences 108
3.3 Values of the (ξ_A^1, ξ_A^2) in-plane lamination parameters 108
3.4 Logarithm of the number of feasible stacking sequences for each value of the (ξ_A^1, ξ_A^2) in-plane lamination parameters 109
3.5 Distribution of feasible ξ_1^D and ξ_2^D 110

3.6 Distribution of feasible ξ_3^D . . . 111
3.7 Examples of shear buckling . . . 111
3.8 Approximation results: orthotropic laminates . . . 120
3.9 Approximation results: non-orthotropic laminates . . . 121
3.10 Reciprocal of the critical buckling load . . . 124

4.1 Raw data for different mixtures . . . 147
4.2 AIC, BIC, log-likelihood and gap curve . . . 148
4.3 Sketch of IMAGE over a one-dimensional test case . . . 156
4.4 Comparison of the approximations for simple piecewise linear functions 157
4.5 Optional caption for list of figures . . . 158
4.6 ANN approximation of buckling results . . . 159
4.7 Example of results for buckling approximation . . . 159
4.8 Active Rayleigh-Ritz modes over the sphere . . . 163
4.9 Automatic clustering (normalized output) . . . 163
4.10 Original function and piecewise linear approximation . . . 164
4.11 Quantile curves for the approximation of λ_{cr} in the orthotropic and non-orthotropic case . . . 166
4.12 Quantile curves for the approximation of λ_{cr} in the general case . . . 168
4.13 Objective function during the optimization process with the combined load . . . 174
4.14 Optimal solution in the lamination parameters space (the 7032 feasible ξ^D are illustrated by a black dot market) under the combined loading . 174
4.15 Objective function during the optimization process with the shear loading 175
4.16 Optimal solution in the lamination parameters space (the 7032 feasible ξ^D are illustrated by a black dot market) under the shear loading . . . 176
4.17 Weighted least squares regressions (polynomials of degree 2 and $\sigma = 0.1$) of the function $x \to \cos(4x)$ at different points and moving least squares regression . . . 180

5.1 Test-case: 10 bar truss . . . 184
5.2 Pattern of the stiffness matrix K_A . . . 186
5.3 Section S_1 of the feasible set for bar 1 . . . 189
5.4 Section of S_1 for $x_3^{(1)} = 0.1$ . . . 190

5.5 Jacobian of constraints for 10 bar truss 190
5.6 Profile of the Jacobian of the constraints for SAND and SAND2 199
5.7 Usual behavior of AIO 201
5.8 Average case for SAND2 202
5.9 Comparison of SAND2 and AIO 203
5.10 Comparison between STIFAND2 and SAND2 204

6.1 Comparison between different versions of StiffOpt 211
6.2 Usual behavior of StiffOpt in the first case 215
6.3 Optional caption for list of figures 215
6.4 Optional caption for list of figures 216
6.5 Comparison of multilevel methods : best in the second case ($\rho_2 = 100\rho_1$) 217
6.6 Erratic behavior of MaxMargin in the first case ($\rho_2 = \rho_1$) 217
6.7 Comparison of AIO optimization and proposed decomposition on load case 1 225
6.8 Comparison of AIO optimization and QSD on load case 2 225
6.9 Comparison of AIO and QSD: load case 1 227
6.10 Comparison of AIO and QSD: load case 2 227
6.11 Comparison of AIO and QSD: load case 3 228

7.1 Optional caption for list of figures 231
7.2 Comparison of the modified IMAGE Mixture of Experts predicted values and the real values for the first 500 feasible stacking sequences 231
7.3 Optional caption for list of figures 232
7.4 Original use case description 235
7.5 Upper level description 236
7.6 Lower level description 237
7.7 Lower level description 241
7.8 Lower level description 242
7.9 Integration of IMAGE into Boss Quattro 253

LIST OF FIGURES

List of Tables

3.1 Graphite/Epoxy material characteristics 119

4.1 Results for the approximation of λ_{cr} for orthotropic laminates for combined loading and pure shear loading 169
4.2 Results for the approximation of λ_{cr} for non-orthotropic laminates for combined loading and pure shear loading 170
4.3 Results for the approximation of λ_{cr} for varying loadings (biaxial loading and combined loading) 171
4.4 Results for the approximation of λ_{cr} for varying non-orthotropic laminates and for varying flux 172
4.5 Optimization result with the combined loading $N_x = -50N.mm^{-1}$, $N_y = 20N.mm^{-1}$ and $N_{xy} = 20N.mm^{-1}$. 173
4.6 Optimization result with the shear loading $N_x = N_y = 0N.mm^{-1}$ and $N_{xy} = 20N.mm^{-1}$. 175

5.1 Synthesis of the results for mono-level methods 205

B.1 AIO Load case applied at node 4 and 5 case $\rho_1 = 100\rho_2$ 268
B.2 SAND2 Load case at node 4 and 5 269
B.3 SAND3 Load case at node 4 and 5 270
B.4 STIFAIO Load case at node 4 and 5 271
B.5 STIFSAND Load case at node 4 and 5 272
B.6 STIFSAND2 Load case at node 4 and 5 273
B.7 STIFFOPT Load case at node 4 and 5 274
B.8 Target Rigidity Load case at node 4 275
B.9 MinMass Load case at node 4 276
B.10 Mix Load case at node 4 277

B.11 MaxMarge case at node 4 case $\rho_2 = 100\rho_1$ 278

Résumé étendu

Cette section propose un résumé étendu de la problématique de cette thèse et présente de manière succincte les principales innovations contenues dans ce travail de thèse. Le problème qu'on cherche à résoudre est précisément décrit, en particulier la nature hiérarchique et 'multiniveau' du dimensionnement de structures aéronautiques à Airbus. On présente ensuite les pierres d'achoppement de la résolution de tels problèmes que sont la grande taille, le calcul des sensibilités. Ce contexte établi on apporte alors de premiers éléments de réponse en adaptant un schéma issue de l'optimisation multidisciplinaire à la spécificités des matériaux composites. Les différents schémas utilisés en optimisation de structures sont présentés sur la base d'éléments théoriques d'optimisation. Dans un premier temps, le problème initial à résoudre est transformé de manière à justifier l'introduction de schémas de décomposition. Cette transforme repose essentiellement sur la nature hiérarchique et de l'analyse et du dimensionnement à Airbus. On commence donc par montrer l'équivalence des formulations d'un point de vue théorique et justifier ainsi l'adéquation de ces schémas avec les besoins industriels. Dans un premiers temps, les variables utilisées ainsi que les contraintes ne sont pas décrites et l'étude de ces décompositions est introduite de manière théorique. Dans un deuxième temps on applique ces décompositions aux spécificités des composites et des contraintes de notre problème d'optimisation. En effet une telle décomposition doit aussi se baser sur une représentation adaptées des composites. La description fine de l'orientation des couches du composite ne paraît pas adaptée aux besoins d'un dimensionnement d'une grande structure faites de nombreuses sous-structures. Une représentation plus adaptée aux besoins du dimensionnement est alors utilisée: les paramètres de stratification. Ces derniers sont en effet plus appropriés pour un processus de dimensionnement par optimisation car les contraintes de stabilité (souvent dimensionnantes) dépendent simplement de cette représentation. D'autre part, l'un des points bloquants de la résolution de tels problèmes consiste justement en les calculs

de stabilité coûteux. Le deuxième élément de réponse consiste alors en l'utilisation de modèles réduits ou modèles d'approximation non physiques. En effet, les contraintes de stabilité doivent être évaluées un grand nombre de fois au cours d'une optimisation, or ces calculs sont coûteux (de l'ordre de la minute pour une évaluation et une optimisation au niveau fuselage peut appeler ces contraintes plusieurs millions de fois). L'intérêt d'un tel modèle d'approximation est donc de remplacer ces calculs répétés par des formules analytiques issues de domaines tels que l'intelligence artificielle (réseaux de neurones), probabilités et statistiques (krigeage), théorie de l'approximation (moindres carrés mobiles) Néanmoins, l'application de ces méthodologies d'approximation n'est pas évidente et la précision obtenue avec l'une ou l'autre de ces méthodes n'est en général pas suffisante pour un processus de dimensionnement réaliste. Le manque de précision a pour origine principal le comportement fortement non linéaire et discontinue des réponses de stabilités. Le comportement typique des contraintes de stabilités et donc des codes de calcul d'Airbus est présenté. En particulier, le caractère discontinu est mis en évidence et la présence de discontinuités ruinent la qualité de l'approximation. Sur la base de ce comportement discontinu, une méthodologie originale de construction de modèles d'approximation est présentée et appliquée. Cette méthodologie construit plusieurs modèles d'approximation plutôt qu'un seul. L'avantage de cette méthode est donc de séparer les domaines de régularité des fonctions à approcher et donc d'approcher localement chaque partie régulière de la fonction sur des régions distinctes de l'approximation.

Motivations et présentation du problème

L'objectif de cette thèse est de rendre possible l'optimisation de grandes structures aéronautiques composites au sein de bureau d'études des concepteurs d'avions tels que celui d'Airbus. De fait l'orientation des grands avionneurs tels qu'Airbus est d'augmenter de manière substantielle la part des matériaux composites dans les structures primaires des futurs avions de lignes civils. Les exemples récents de l'A-350 (plus de 52% de structures primaires en matériaux composites) ou encore du Boeing 787 représentent bien cette tendance qui devrait à l'avenir même augmenter.

Les matériaux composites dans l'aéronautique

Depuis les années 70, la part des matériaux composites dans les structures travaillantes des avions Airbus ne cessent d'augmenter. De fait, leurs propriétés mécaniques et leur plus faible masse volumique par rapport aux matériaux métalliques traditionnellement utilisés (essentiellement des alliages à base d'aluminium). Les matériaux composites sont de manière générale un assemblage de matériaux distincts qui ne se mélangent pas mais qui ont entre eux de fortes propriété d'adhérence. On peut citer à titres d'exemple le béton armé, le GLARE (composite fait d'aluminium et de fibre de verre, utilisé par exemple sur l'A-380 sur les panneaux supérieurs de fuselage) ou encore les composites naturels le bois. En général, les composites consistent en l'assemblage d'un renfort ou squelette qui assure la tenue mécanique et d'une matrice qui redistribue les efforts vers le renfort et assure la cohésion du squelette. Cette matrice a en général de faibles propriétés mécaniques. Selon la nature de la matrice on distingue

- les composites à matrices organiques (CMO): carbone, silicium, verre...ce sont les matériaux composites les plus utilisés dans l'industrie
- les composites à matrices céramiques (CMC) pour des applications très spécifiques souvent liées à hautes températures (nucléaire, turbo-réacteur...)
- les composites à matrices métalliques (CMM): GLARE par exemple

Si ces trois types de composites sont utilisés sur une avion. On s'intéresse dans cette thèse uniquement aux composites à matrices organiques et plus précisément aux stratifiés. On appelle stratifié un matériaux composite fait de plusieurs couches (ou pli) de nappes unidirectionnelles de fibres avec des orientations propres à chaque pli. La séquence d'empilement est alors la donnée des angles de chacune des orientations, on la note

$$[\alpha_1/\alpha_2/.../\alpha_n] \tag{0.1}$$

pour une séquence d'empilement de n plis. Dans ce travail de thèse on considèrera uniquement des stratifiés d'orientations possible $[0, 45, -45, 90]$.

Au-delà de leur faible masse volumique, les matériaux composites et en particulier les stratifiés présentent les avantages suivants:

- Peu ou pas de sensibilité à la fatigue ce qui conduit à une baisse en fréquence des inspections et donc une diminution des coûts de maintenance
- Assez bonne tolérance aux dommages
- Peu ou pas de corrosion

Néanmoins les matériaux composites présentent encore de nombreux inconvénients

- Matériaux non-conducteurs et donc besoin de 'métalliser' les structures
- Difficultés à détecter certains endommagements, les méthodes de diagnostic des structures non destructives doivent être adaptées aux composites
- Résistance à des environnements extrêmes en particulier les températures hautes
- L'état de surface nécessite un traitement spécifique pour pouvoir être peint
- Certains assemblages (en particulier avec des structures métalliques) nécessitent des techniques spécifiques

Mais le principal avantage et inconvénient des matériaux composites consiste en leur possibilité de conception. De fait à la différence de matériaux métalliques homogène et isotrope, les stratifiés sont par définition même des matériaux inhomogènes (leurs propriétés mécaniques dépendent du point considéré, à l'échelle microscopique ou même mésoscopique ces propriétés sont différentes entre la résine et le renfort) et anisotropes (les propriétés mécaniques dépendent des directions considérées). Ces caractéristiques sont en effet intéressantes

- on peut adapter le stratifié aux chemins d'effort que la structure va rencontrer en favorisant certaines directions. On peut par exemple dans certaines parties favoriser les orientations des plis à 0 si ces parties travaillent essentiellement en sollicitation longitudinale et à l'inverse favoriser les plis à 90 si la structure travaillent en sollicitation latérale. De même le phénomène de flambage qu'on précise ensuite, pourra être mieux être prévenu avec des plis à 45 qu'avec un matériau métallique.

- on peut aussi adapter le stratifié et obtenir des couplages que les matériaux isotropes ne peuvent pas faire apparaître. L'exemple de stratifiés non balancé (plus de plis à 45 qu'à -45) est par exemple intéressant pour des applications en aéroélasticité où ce caractère favorise un couplage négatif entre la torsion et la flexion.

Néanmoins, cette plus grande liberté dans la conception ne va pas sans inconvénients

- Le caractère inhomogène et anisotrope complique considérablement l'analyse mécanique de la structure. Une hypothèse simplificatrice couramment utilisée est d'homogénéiser le stratifié en considérant des propriétés mécaniques moyennes sur toute l'épaisseur. Si cette hypothèse peut être acceptable en première approximation, elle complique déjà l'analyse de phénomènes classiques type flambage mais elle ne permet pas l'étude fine de phénomènes plus compliqués. En particulier pour des phénomènes non-linéaires ou bien liés à la structure même du stratifié (délaminage, rupture, propagation de fissure), cette approximation ne peut pas être utilisée et les modèles de prédictions n'ont pas la maturité des modèles pour les matériaux homogène isotropes.

- La conception est plus complexe et difficile du fait même des nouveaux degrés de liberté qu'elle introduit. D'un point de vue optimisation, les stratifiés font intervenir plus de variables (orientation des plis) et surtout ces nouvelles variables sont désormais discrètes. La situation est donc bien plus complexe qu'en métallique, où les principales variables d'optimisation sont les dimensions géométriques.

Structures de fuselage d'avions et comportement mécanique

Le fuselage d'un avion est un assemblage de panneaux renforcé dans le sens longitudinal par des raidisseurs et dans le sens orbital par des cadres (ou lisses). Le fuselage reprend plusieurs types d'efforts, d'une part les sollicitations dues aux efforts de voilure par l'intermédiaire du caisson central qui reprend une partie de ces efforts et les redistribue et d'autres part les efforts du au fuselage lui-même en appui sur le caisson central. Les principales sollicitations que le fuselage reprend sont donc

- la traction sur l'extrados, les panneaux supérieurs travaillent donc essentiellement en traction longitudinale.

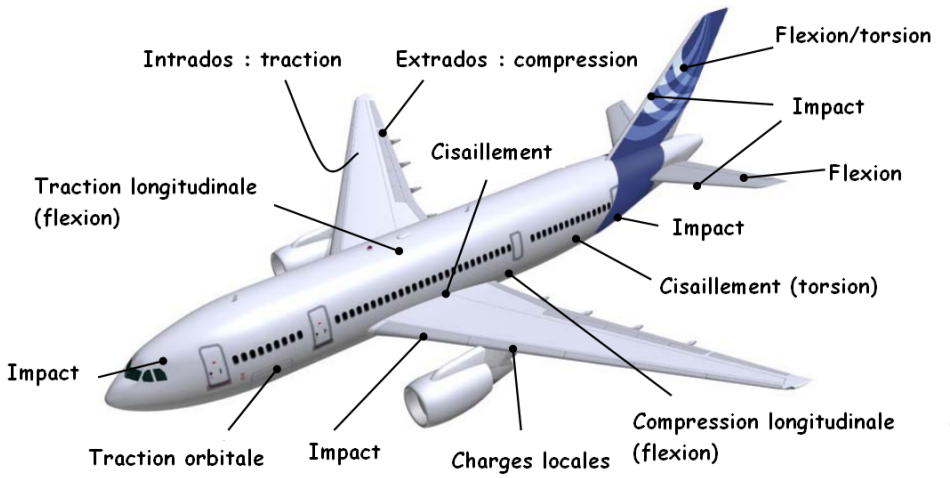


Figure 0.1: Principales sollicitations

- la compression sur l'intrados, les panneaux sous le fuselage travaillent essentiellement en compression longitudinale

- les panneaux latéraux quant à eux reprennent les efforts de cisaillement induit par la voilure ainsi que la torsion due à la flèche des ailes. Ils travaillent donc à la fois en traction/compression et e cisaillement.

On a représenté Fig. 0.1 les principales sollicitations sur l'ensemble de l'avion. D'un point de vue mécanique, la structure de fuselage est donc un ensemble de coques et de plaques minces renforcées par des poutres qui travaillent essentiellement en membrane (efforts plans). D'un point de vue modélisation, de simples modèles 2d sont suffisants pour se représenter les efforts principaux que la structure globale rencontre (par exemple hypothèse de contraintes/déformations planes). D'un point de vue optimisation, les contraintes typiques de dimensionnement à ce niveau global de la structure sont des limitations sur la valeur absolue des contraintes mécaniques, des déformations ou encore des déplacements (typiquement pour l'encadrement des portes des aéronefs). Au niveau sous-élément de la structure, les contraintes d'optimisation cherchent à ne pas faire apparaître des phénomènes d'instabilité (ou de bifurcation). Le phénomène d'instabilité le plus courant au niveau du sous-élément est le flambage,

qui se traduit par un phénomène de flexion alors que la structure est sollicitée en compression/cisaillement. Le phénomène de flambage est moins simple à prédire puisqu'il s'agit d'un phénomène non-linéaire qu'on sait au moins prédire de manière partielle en linéarisant les équations. Ce phénomène est courant dans les structures élancées telles que celle de fuselage ou de voilure.

L'optimisation de structures aéronautiques

L'optimisation de structure cherche à tirer le meilleur d'une structure. Par 'tirer le meilleur' on entend bien souvent dans le domaine aéronautique trouver la structure la plus légère qui résiste à l'ensemble des charges que l'avion va rencontrer durant son utilisation. l'optimisation de structures cherche donc à automatiser le processus de dimensionnement réalisé assez souvent sur la base de l'expertise de l'ingénieur en structures. Néanmoins, la taille de plus en plus grande des espaces de design (ensemble des variables que l'ingénieur structures a à sa disposition pour modifier la structure) ne permet pas de trouver le meilleur design sans une automatisation du processus. L'optimisation mathématique permet, une fois le problème formalisé, de trouver un meilleur design. Ceci appelle plusieurs remarques

- La formalisation du problème d'optimisation n'est pas toujours évidente. Il faut en premier lieu définir les variables d'optimisation. Selon les applications, celles-ci peuvent être des dimensions continues (dimensions géométriques de la structure), des variables discrètes (orientations des composites), des fonctions (par exemple pour l'optimisation du profil d'une aile), des variables catégorielles (type d'éléments structuraux, présence ou non d'un élément structural) et enfin la matière elle-même dans le cas de l'optimisation topologique. Sur la base de ces variables d'optimisation, on définit ensuite la fonction objectif (critère à optimiser) et les contraintes d'admissibilité. Bien souvent en optimisation de structures aéronautiques la fonction objective est la masse totale de la structure considérée et s'évalue assez simplement pour peu qu'on dispose d'une bonne paramétrisation de la structure. Les contraintes quant à elles sont en général bien plus compliquées à évaluer car elles font intervenir en général un modèle physique, ce dernier étant bien souvent dans le cadre du système de l'élasticité. Ce modèle physique doit pouvoir être évalué rapidement sur la base des variables choisies. Ce

dernier point peut s'avérer compliqué quand les variables d'optimisation ne correspondent pas nécessairement aux variables physiques du modèle (par exemple les proportions d'orientations dans un stratifié ne suffisent pas à prédire correctement le flambage). Cette évaluation rapide nécessite donc des modèles d'approximation numérique éprouvés. Ces modèles résolvent des systèmes d'équations aux dérivées partielles de l'élasticité linéaire, par exemple, pour les structures de fuselages considérées, l'approximation de contraintes planes du comportement mécanique global du fuselage (principales sollicitations sont dans le plan) est résolue à l'aide de la méthode des éléments finis. Le flambage par exemple est prédit sur la base d'équations simplifiées elles aussi dérivées de l'élasticité (non linéaire) sur la base d'hypothèses physiques raisonnables: équations de von Karman obtenues sur la base des hypothèses de Kirchhoff, ces équations sont par exemple souvent résolues numériquement à l'aide de méthodes spectrales type Rayleigh-Ritz.

- Les algorithmes d'optimisation différentiable en dimension finie permet en général de trouver un minimum local et non un minimum global. Les méthodes de gradient et de Newton et quasi-Newton, lorsqu'elles convergent et cette convergence peut au moins être assuré théoriquement lorsque le point de départ de l'optimisation est proche de l'optimum, trouvent uniquement un point satisfaisant les conditions d'optimalité d'un minimum local (stationnarité du gradient et positivité de la hessienne). C'est-à-dire que le design trouvé par ce type d'algorithme n'est a priori pas le meilleur design mais simplement un design qui localement est le meilleur. Pour remédier à ces inconvénients, il existe plusieurs stratégies comme par exemple lancer plusieurs optimisations avec des points initiaux différents (**multistart**). Néanmoins il n'y a pas d'assurance de converger vers l'optimum global. Il existe des stratégies d'optimisation globales: algorithmes génétiques, recuit simulé, essaim particulaire, colonies de fourmis... qui explorent de manière plus complète l'espace de recherche. Mais ces stratégies, souvent appelées heuristiques, n'ont pas de propriétés de convergence théorique. Un avantage considérable de ces méthodes est qu'elle ne requièrent pas les hypothèses de régularité classique de l'optimisation différentiable (continuité, différentiabilité...) et s'écrivent aisément pour des variables discrètes. Leur principal inconvénient est le nombre élevé d'appels aux fonctions objectifs et contraintes. Enfin précisons

que si le problème d'optimisation est convexe (fonction objectifs et contraintes), tout minimum local est alors minimum global. Cela fournit par exemple une piste d'amélioration pour l'optimisation de structures en aéronautique où les problèmes ne sont en général pas convexe (par exemple la masse est fonction de l'aire de section qui est un produit des dimensions locales), si l'on peut trouver une formulation du problème convexe (par exemple en utilisant des variables ou des critères plus physiques) on peut plus aisément trouver le meilleur design.

Présentation du problème

On présente dans cette section le problème formel qu'on cherche à résoudre. Dans un premier temps, on explique en quoi la décomposition (et donc le multiniveau) se révèle intéressante pour d'une part rendre possible et accélérer la résolution de très grands problèmes d'optimisation non-linéaire. Puis on présente la processus de dimensionnement à Airbus. Pour pouvoir être appliquée, un tel schéma de décomposition doit pouvoir correspondre à ce processus. En particulier, on distinguera des niveaux d'analyse de la structure, le schéma d'optimisation multiniveau doit faire apparaître ces niveaux. Une structure hiérarchique apparaît donc et c'est sur la base de cette structure hiérarchique qu'on va cherche une décomposition appropriée. On voit donc apparaître la notion d'optimisation multiniveau. On présente donc à la fin de cette section dans un premier temps l'équivalence mathématique des formulations mathématiques puis les décompositions naturelles que ces schémas induisent.

Le processus du dimensionnement à Airbus

Le processus d'analyse et de dimensionnement à Airbus consiste en un premier temps aux calculs de redistribution des efforts sur la base d'un petit nombre de variables, dites variables de rigidités. Une fois la redistribution des efforts calculées on analyse localement la stabilité de l'élément appelé super-raidisseur, composé d'un raidisseur et des deux demi-panneau autour). Cette analyse en stabilité se base sur une description plus fine de la structure: dimensions détaillées du raidisseur et dans le cas des composites la connaissance de la séquence d'empilement est nécessaire.

Problème formel

Une structure typique de fuselage comprend plusieurs milliers de panneaux et de raidisseurs. Chacune de ces sous-structures comprend plusieurs variables d'optimisation (par exemple épaisseurs des panneaux, dimensions géométriques du profil de raidisseur) que le concepteur peut modifier. Les contraintes d'admissibilité de la structure interviennent sur chacune de ces sous-structures: déformations maximales dans chacun des panneaux, des raidisseurs, contraintes de flambages (panneaux, raidisseur, sous-ensemble du raidisseur...) tant est si bien que le nombre de contraintes est encore plus important que le nombre de variables d'optimisation. Néanmoins, on comprend déjà que toutes ces variables sont reliées entre elles par la redistribution des efforts, toute modification d'une dimension locale d'un des raidisseurs implique une modification du terme de rigidité de l'élément considéré et donc modifie (même faiblement) la redistribution des efforts au niveau global. Ainsi donc toutes les variables locales influent sur les contraintes de tous les autres éléments par le biais de la redistribution des efforts. D'autre part, les contraintes locales (type flambage) d'un élément dépendent essentiellement des dimensions locales de l'élément considéré.

Minimiser la masse d'un fuselage s'écrit ainsi comme un problème de minimisation de très grande taille avec une structure spécifique des contraintes. de fait certaines des contraintes sont essentiellement globales (liées à la redistribution des efforts) et d'autres globales-locales (liées à la fois aux dimensions locales mais aussi à toutes les autres variables par la redistribution des efforts). Plus formellement, on peut écrire le problème de la manière suivante:

- On considère que la structure est faite de N sous-structures, les dimensions locales de l'élément i sont notées $X^{(i)} = (x_i^1, \ldots, x_i^{n_i})$ où n_i est le nombre de variables locales de l'élément i.

- On note les variables de rigidités globales $Y^{(i)}$ de l'élément i. La redistribution des efforts fait intervenir donc tous les $Y^{(i)}$. Ces variables de rigidités globales dépendent directement des $X^{(i)}$ et on écrira donc $Y(X^{(i)}) = Y^{(i)}$.

- On distingue les contraintes c_{glob}^K qu'on note avec un K en exposant qui sont les contraintes globales liées à la redistribution des efforts (d'où le K pour signifier la

résolution du système linéaire $Kd = f$). On désigne le calcul de la redistribution des efforts par Φ, les efforts dans l'élément i sont donc désignés par $\Phi^{(i)}$.

- Les contraintes $c^{Ki}_{glob-loc}$ globales-locales qui font à la fois intervenir les dimensions locales et la redistribution des efforts, elles seront aussi désignées sous la forme de RF car il s'agit souvent de **Reserve Factor** ou facteurs de réserve.

- Enfin les contraintes $c^{(i)}_{loc}$ qui font intervenir uniquement les dimensions locales, ce sont des contraintes de types dimensions, boîtes...

- Le problème de minimisation s'écrit donc

$$\begin{array}{ll} \min_X & M(X) = \sum_{i=1}^{N} m^{(i)}(X^{(i)}) \\ \text{sous} & \left\{ \begin{array}{l} c^{K}_{glob}(Y^{(1)}, \ldots, Y^{(N)}) \leq 0 \\ c^{Ki}_{glob-loc}(\Phi^{(i)}(Y), X^{(i)}) \leq 0 \\ c_{glob}(X^{(1)}, \ldots, X^{(N)}) \leq 0 \\ c^{(i)}_{loc}(X^{(i)}) \leq 0 \end{array} \right. \end{array} \tag{0.2}$$

Le problème d'optimisation () s'écrit donc a priori en utilisant les variables d'optimisation X. Néanmoins, cette écriture conduit à une dépendance complexe des contraintes globales-locales puisque toutes les variables locales influent par le biais de la redistribution des efforts, on a illustré cette dépendance Fig. **??**.

L'optimisation de grande taille: vers la décomposition

Le problème (0.2) fait donc apparaître une structure particulière. De fait, comme on peut le voir Fig. (0.2), les contraintes globales-locales dépendent essentiellement des variables locales mais aussi finalement de toutes les autres variables par le biais de la redistribution des efforts. Le problème a donc une structure **quasi-bloc diagonale** et on cherche à exploiter cette structure de manière à **décomposer le problème global d'optimisation** en plusieurs sous-problèmes d'optimisation pouvant être résolues en parallèle. La décomposition naturelle de la structure en sous-éléments structuraux se reflète donc le problème d'optimisation. L'objectif est alors de décomposer le problème en sous-problèmes chacun écrit sur un sous-élément structural et de traiter correctement le couplage entre les sous-problèmes (redistribution des efforts). D'autre part, cette décomposition doit refléter aussi les différents niveaux d'analyse de la structure. En effet, on distingue

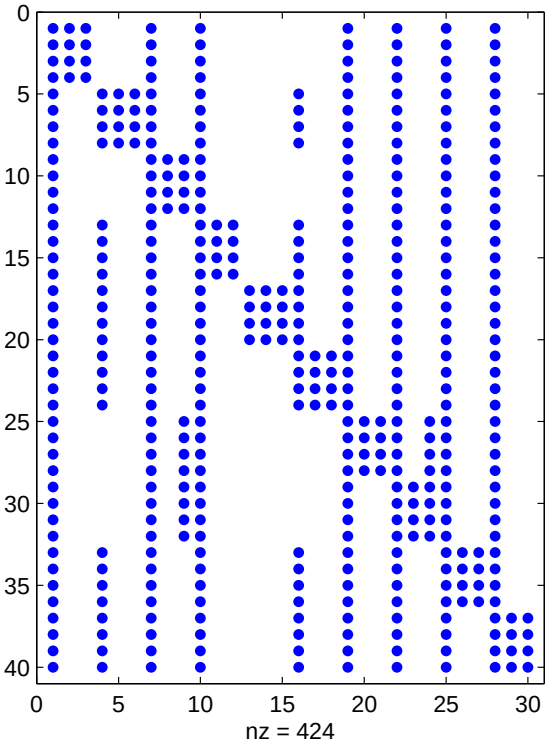


Figure 0.2: Profil de la jacobienne des contraintes globales-locales. On a représenté uniquement les termes dominants en valeur absolue. En réalité tous les termes sont non nuls et la matrice est pleine, néanmoins on voit apparaître un profil bloc-diagonal dominant

- l'analyse globale de la structure: redistribution des efforts, basée sur une représentation grossière de la structure où les seules variables sont les variables de rigidités Y (aires de section...). Ce niveau d'analyse est en général réalisé par le biais une analyse éléments finis et permet de calculer les contraintes globales (déplacements prescrits, contraintes mécaniques). Notons qu'enfin ce niveau se fonde aussi sur un certain nombre d'hypothèses mécanique : comportement plan (pas de flexion), contraintes et déformations planes, raidisseurs approchés par des barres...

- l'analyse en stabilité essentiellement locale. A ce niveau l'analyse se base sur une représentation plus fine du sous-élément structural (dimensions géométriques détaillées, séquence d'empilement). Cette analyse se fonde elle aussi des hypothèses mécaniques simplificatrices (plaques minces, théorie de Kirchhoff, pas de torsion...) et est réalisée au moyen de méthodes numériques spécifiques (spectrales type Rayleigh-Ritz) valables pour des géométries simples (un panneau trapézoïdale est par exemple approché par un panneau rectangulaire).

En définitive la décomposition doit non seulement se fonder sur la décomposition naturelle de la structure et aussi correspondre aux différents niveaux d'analyse (FEA puis outils Airbus de stabilité).

Équivalence des formulation mono-niveau

Pour ce faire, le premier travail consiste en définir clairement les variables d'optimisation. De fait, le problème initial ne fait pas explicitement apparaître les variables globales ou de rigidités Y en tant que variables d'optimisation. Ainsi formulé le problème (0.2) complique la résolution et la décomposition du problème, par exemple par le biais du calcul des sensibilités. Pour simplifier ce dernier calcul et donner au problème une forme plus adaptée à la décomposition on introduit alors les variables de rigidité Y comme variables d'optimisation a priori indépendantes des variables locales. La cohérence entre le niveau globale et le niveau locale est alors assurée par des contraintes égalités qu'on peut relaxer. Cela conduit alors au problème bloc-angulaire suivant

$$\begin{array}{ll} \min_{X,Y} & M(Y) = \sum_{i=1}^{N} m^{(i)}(Y) \\ \text{sous} & \left\{ \begin{array}{l} c^{K}_{glob}(Y^{(1)}, \ldots, Y^{(N)}) \leq 0 \\ c^{Ki}_{glob-loc}(Y, X^{(i)}) \leq 0 \\ c_{glob}(X^{(1)}, \ldots, X^{(N)}) \leq 0 \\ c^{(i)}_{loc}(X^{(i)}) \leq 0 \\ \varphi(X^{(i)}) = Y^{(i)} \end{array} \right. \end{array} \tag{0.3}$$

Notons que dans le problème (0.3), on a écrit l'objectif comme étant uniquement fonction des variables de rigidités. Cela est réaliste dans la mesure où la masse peut souvent se déduire de l'aire de section. Dans le chapitre deux de ce travail, on commence par montrer que les deux problème sont équivalents en termes de problèmes d'optimisation. On caractérise le minimum des deux problèmes par les conditions d'optimalité de Karush-Kuhn-Tucker et on montre que le minimum (local) d'un problème l'est aussi pour l'autre. L'intérêt de la réécriture du problème sous forme bloc angulaire est alors de faire apparaître une décomposition plus adaptée à nos besoins. En effet, le problème se décompose naturellement en un problème d'optimisation en Y puis des problèmes en $X^{(i)}$. En fixant les Y on obtient alors N problèmes totalement séparables en $X^{(i)}$ pouvant alors être résolus en parallèle. La difficulté réside alors dans la définition de l'objectif local qu'on donne aux problèmes d'optimisation en $X^{(i)}$. Pour faire apparaître l'une des décompositions qu'on utilise par la suite, notons qu'on peut relaxer les contraintes égalités en utilisant un pénalisation quadratique inexacte des contraintes, on obtient alors

$$\begin{array}{ll} \min_{X,Y} & M(Y,X) = \sum_{i=1}^{N} m^{(i)}(Y) + ||Y^{(i)} - \varphi(X^{(i)}||^2 \\ \text{sous} & \left\{ \begin{array}{l} c^{K}_{glob}(Y^{(1)}, \ldots, Y^{(N)}) \leq 0 \\ c^{Ki}_{glob-loc}(\Phi^{(i)}(Y), X^{(i)}) \leq 0 \\ c_{glob}(X^{(1)}, \ldots, X^{(N)}) \leq 0 \\ c^{(i)}_{loc}(X^{(i)}) \leq 0 \end{array} \right. \end{array} \tag{0.4}$$

Pour l'instant aucun des problèmes (0.2), (0.3) et (0.4) n'est décomposé. On parle donc de formulations mononiveau.

Présentation des schémas de décomposition utilisés

Sur la base de la réécriture du problème sous forme bloc angulaire, on s'intéresse alors à des formulations d'optimisation biniveau. Plus précisément on cherche à écrire le problème 0.3 sous la forme d'un problème d'optimisation en Y qui fait intervenir

dans ces contraintes la résolution de problèmes d'optimisation en $X^{(i)}$. L'écriture du problème sous forme biniveau générale est alors

- Niveau global (ou niveau système) :

$$\begin{array}{ll} \min_Y & M(Y) = \sum_{i=1}^{N} m_i(Y_i) \\ \text{s. t.} & \left\{ \begin{array}{l} c_{glob}^{K}(Y_1, \ldots, Y_N) \leq 0 \\ \gamma_i(Y) \leq 0 \quad \text{pour} \quad i = 1 \ldots N \end{array} \right. \end{array} \tag{0.5}$$

- où γ_i pour $i = 1 \ldots N$ est une fonction de couplage calculée par le niveau inférieur

$$\begin{array}{ll} \min_{X_i} & h_i(X_i, Y) \\ \text{t. q} & \left\{ \begin{array}{l} c_{glob-loc}^{(i)}(Y, X_i) \leq 0 \\ c_{loc}^{(i)}(X_i) \leq 0 \end{array} \right. \end{array} \tag{0.6}$$

- de cette manière les contraintes basées sur une analyse éléments finis sont calculées uniquement au niveau global et les N optimisations locales peuvent être calculées en parallèle

On présente maintenant les différents schémas utilisés en précisant le niveau inférieur.

Rigidité Cible

L'idée principale du schéma Rigidité Cible est donc de réaliser au premier niveau (ou niveau global) où la redistribution des efforts varient au niveau global, en considérant comme contraintes d'optimisation les contraintes mécaniques et les déformations. A ce niveau global, on considère uniquement les termes de rigidités (aires de sections,...). Au niveau local, l'optimisation cherche le design local admissible le plus proche en terme de rigidité. De cette manière, la redistribution des efforts varient peu entre les optimisations locales et l'optimisation globale.

- **Rigidité Cible** définit $\gamma_i(Y) = \frac{1}{2}|\varphi(X_i^*) - Y_i|^2$

- où X_i^* est

$$\begin{array}{ll} \text{argmin}_{X_i} & \frac{1}{2}||\varphi(X_i) - Y_i||^2 \\ \text{t. q} & \left\{ \begin{array}{l} c_{glob-loc}^{(i)}(Y, X_i) \leq 0 \\ c_{loc}^{(i)}(X_i) \leq 0 \end{array} \right. \end{array} \tag{0.7}$$

Dans le contexte de l'optimisation multidisciplinaire (MDO), le schéma Rigidité Cible est proche du schéma CO (Collaborative Optimization). Avec la contrainte $\gamma_i(Y) \leq 0$, on voit qu'à convergence (globale) le design local doit donc correspondre exactement aux variables de rigidités du niveau supérieur et ainsi être admissible car résultat d'une optimisation locale qui inclut les contraintes globales-locales (flambage) avec les efforts corrects.

Maximum Marge

Ce schéma développé en premier lieu par Haftka et Sobieski dans les années 80 et ensuite étendu dans une forme plus générale par Haftka, Liu et Watson sous le nom Quasi Separable subsystem Decomposition (QSD).

- **MaxMargin** définit $\gamma_i(Y) = \min_{X_i \in \varphi^{-1}(Y)} \{c^{(i)}_{glob-loc}(Y, X_i^*)\}$ et donc $\gamma_i(Y) = -\mu_i^\star$
- où $\mu_i^\star$ vaut

$$\begin{array}{ll} \text{argmax}_{\mu_i, X_i} & \mu_i \\ \text{t. q} & \left\{ \begin{array}{l} c^{(i)}_{glob-loc}(Y, X_i) + \mu_i \leq 0 \\ c^{(i)}_{loc}(X_i) + \mu_i \leq 0 \\ Y_i = \varphi(X_i) \end{array} \right. \end{array} \tag{0.8}$$

Le nom **Maximum Marge** vient du fait que l'optimisation locale cherche le design local qui viole le moins les contraintes globales-locales et les contraintes locales. On maximise donc au niveau local l'admissibilité et on contraint au niveau supérieur cette marge à être positive. De manière équivalente, on pourrait minimiser l'opposé de la marge et la contraindre au niveau supérieur à être négative.

Minimisation de masse locale MinMass

Ce schéma a été développé par A. Merval ((Merval 2008)) et propose de minimiser la masse en local.

- MinMass définit $\gamma_i(Y) = c^{(i)}_{glob-loc}(Y, X_i^*)$
- où X_i^* est

$$\begin{array}{ll} \text{argmin}_{X_i} & m_i(X_i) \\ \text{t. q} & \left\{ \begin{array}{l} c^{(i)}_{glob-loc}(Y, X_i) \leq 0 \\ c^{(i)}_{loc}(X_i) \leq 0 \\ Y_i \leq \varphi(X_i) \end{array} \right. \end{array} \tag{0.9}$$

schéma Mix

Au cours de cette thèse nous avons aussi développé le schéma suivant

- ce schéma est **une combinaison de MinMass et de Rigidité Cible**
- $\gamma_i(Y) = \max\{c^{(i)}_{glob-loc}(Y, X_i^*), |\varphi(X_i^*) - Y_i|\}$
- où X_i^* est

$$\begin{array}{ll} \text{argmin}_{X_i} & \varepsilon\frac{1}{2}|\varphi(X_i) - Y_i|^2 + (1-\varepsilon)m_i(X_i) \\ \text{t. q} & \left\{ \begin{array}{l} c^{(i)}_{glob-loc}(Y, X_i) \leq 0 \\ c^{(i)}_{loc}(X_i) \leq 0 \\ Y_i \leq \varphi(X_i) \end{array} \right. \end{array} \tag{0.10}$$

Approximation des contraintes coûteuses

On s'intéresse dans cette section à l'approximation des contraintes globales-locales $c^{(i)}(\Phi^{(i)}(Y), X^{(i)})$. Dans un premier temps le comportement typique de ces contraintes est étudié et plus précisément le comportement des calculs de stabilités (flambage) sur la base de la représentation du composite choisie. Ces contraintes font intervenir des fonctions irrégulières dans le sens où ce sont on voit apparaître des discontinuités de la dérivée et même dans le cas des outils métiers Airbus, des discontinuités induites par des politiques de marge.

Comportement typique des contraintes de stabilité

On s'intéresse ici au comportement du flambage. Dans le cas des stratifiés, une représentation classique des comportements plan (traction/compression, cisaillement) et hors-plan (flexion) sont les paramètres de stratifications. Dans le cas général (pas de contraintes sur les séquences d'empilement), ces derniers sont au nombre de 12 et permettent un e représentation continue des différents comportements mécaniques du stratifié. La loi constitutive (relation contraintes mécaniques/déformations) fait apparaître trois tenseurs, souvent appelés A, B et D. Chacun décrivant des couplages différents

- le tenseur de membrane A : décrit les propriétés plan du stratifié. Ce dernier est fonction des quatre paramètres de stratification de membrane $\xi^A_{1,2,3,4}$.
- le tenseur de couplage B : décrit le couplage propriétés plan/flexion. Ce dernier est nul dès que la séquence est symétrique.

- le tenseur de flexion D : décrit le comportement hors-plan du stratifié. Le phénomène de flambage dépend essentiellement des termes du tenseur D mais aussi des propriétés de membrane qui apparaissent au second membre de l'équation de flambage. De fait l'équation de flambage est grossièrement

$$D\Delta^2 w = \lambda N \Delta w \tag{0.11}$$

  où w est le déplacement transverse du stratifié et N est le vecteur des flux (N_x, N_y, N_{xy}) qui dépendent eux de la redistribution globales des efforts. Le facteur de réserve de flambage est alors λ la plus petite valeur propre positive. Notons enfin que D est fonction des paramètres de stratification de flexion $\xi^D_{1,2,3,4}$.

Dans le cas des stratifiés que nous considérons dans cette thèse, de nombreux paramètres de stratification s'annulent. Tant et si bien que seuls $\xi^A_{1,2}$ et $\xi^D_{1,2,3}$ sont non nuls. En plus d'offrir une représentation compacte et continue du flambage, de nombreux travaux ont permis d'écrire les équations dite de compatibilité qui permettent de contraindre l'espace des ξ^D sur la base de la valeur des ξ^A (qui sont équivalents pour nos stratifiés aux proportions angulaires). Dans cette thèse on s'intéresse au comportement du flambage en fonction des ξ^D et des flux (N_x, N_y, N_{xy}). s On montre par exemple des propriétés importantes pour l'optimisation, en particulier **le fait que la facteur de réserve de flambage est concave sur les** ξ^D, ce qui permet d'assurer que le problème de maximisation de ce facteur de réserve est bien posé. D'autre part, on étudie numériquement le comportement du facteur de réserve lorsque les ξ^D et les flux varient, on s'aperçoit alors que le comportement typique est de dérivée discontinue par morceaux. On cherche donc une méthode d'approximation qui tire parti de cette structure. On a représenté le comportement typique des calculs de flambage Fig.

Schéma original d'approximation des contraintes

L'un des principaux apports de cette thèse est de proposer un schéma original d'approximation applicables à des fonctions comme les contraintes de stabilités. La principale difficulté est donc d'approcher une fonction de dérivée discontinue ou même discontinue. Les 'coins' et les sauts de discontinuités empêchent en effet un modèle réduit ou d'approximation d'atteindre une grande précision. L'idée pour améliorer la précision est donc de partitionner l'espace en plusieurs régions et d'entraîner sur chacune des régions et essayant donc de contourner les discontinuités.

Applications numériques et validation du schéma original

Le schéma original développé se base donc sur la notion de mélange d'experts. Le partitionnement de la base est réalisé au moyen de l'algorithme EM appliqué sur le couple (X, Y) des entrées/sorties. On fait donc l'hypothèse que la loi conjointe (X, Y) est de la forme suivante

$$\sum_{i=1}^{K} \alpha_i \mathcal{N}(\mu_i, \Gamma_i) \tag{0.12}$$

c'est-à-dire un mélange de gaussiennes. Chacune des gaussiennes est pondérée par un facteur α_i et est de paramètres μ_i (moyenne) et de matrice de variance-covariance Γ_i. Le mélange est donc constituée de K gaussiennes ou K composantes. On estime donc grâce à l'algorithme EM les paramètres des gaussiennes dans $\mathbb{R}^{p+1}$ et on décompose ces paramètres de la manière suivante

$$\mu_k = \begin{pmatrix} \mu_k^X \\ \\ \mu_k^Y \end{pmatrix}, \tag{0.13}$$

où $\mu_k^X \in \mathbb{R}^p$ sont les coordonnées en X de la moyenne μ_i et $\mu_k^Y \in \mathbb{R}$ est la coordonnée en Y de la moyenne μ_i. On décompose la matrice de variance-covariance de la manière suivante

$$\Gamma_i = \begin{pmatrix} \Gamma_i^X & \nu_i \\ \\ \nu_i^T & \xi_i \end{pmatrix}, \tag{0.14}$$

où Γ_i^X est la matrice de variance en X pour la composante i, $\nu_i \in \mathbb{R}^p$ est $\mathrm{Cov}(X, Y)$ pour la composante i et $\xi_i = \mathrm{Var}(Y) \in \mathbb{R}$.

On peut donc dériver la loi marginale en X. On note alors la variable aléatoire κ associée au couple (X, Y) qui note l'appartenance à une composante. $\kappa_i = k$ signifie donc que l'échantillon (X_i, Y_i) appartient à la composante k. On peut donc dériver la loi de $X|\kappa = k$

$$X|\kappa = k \sim \mathcal{N}(\mu_k^X, \Gamma_k^X) \tag{0.15}$$

et donc

$$< X = x_i >= \cup_{k=1}^{K} < X = x_i \cap \kappa = k > \tag{0.16}$$

et on a finalement

$$X \sim \sum_{i=1}^{K} \alpha_k \mathcal{N}(\mu_k^X, \Gamma_k^X) \tag{0.17}$$

On peut alors calculer la probabilité postérieure d'appartenance à un cluster i en utilisant la formule de Bayes.

$$\mathbb{P}(\kappa = k_i | (X,Y) = (x,y)) = \frac{\mathbb{P}(\kappa = k_i)\mathbb{P}((X,Y) = (x,y) | \kappa = k_i)}{\sum_{k=1}^{K} \mathbb{P}(\kappa = k)\mathbb{P}((X,Y) = (x,y) \mid \kappa = k)}.$$

Dans notre cas gaussien, on a alors

$$\begin{aligned} \mathbb{P}(\kappa = k) &= \alpha_k, \\ (X,Y) \mid (\kappa = k) &\sim \mathcal{N}(\mu_k, \Gamma_k), \end{aligned}$$

ce qui conduit avec $z = (x,y)$ à

$$\mathbb{P}(\kappa = k_i | (X,Y) = (x,y)) = \frac{\det(\Gamma_{k_i})^{-\frac{1}{2}} \alpha_{k_i} e^{-\frac{1}{2}(z-\mu_{k_i})^T \Gamma_{k_i}^{-1}(z-\mu_{k_i})}}{\sum_{k=1}^{K} \det(\Gamma_k)^{-\frac{1}{2}} \alpha_k e^{-\frac{1}{2}(z-\mu_k)^T \Gamma_k^{-1}(z-\mu_k)}}. \tag{0.18}$$

De la même manière, on peut aussi appliquer la formule de Bayes uniquement sur les X_i

$$\mathbb{P}(\kappa = k_i | X = x) = \frac{\det(\Gamma_{k_i}^X)^{-\frac{1}{2}} \alpha_{k_i} e^{-\frac{1}{2}(x-\mu_{k_i}^X)^T \Gamma_{k_i}^{X\,-1}(x-\mu_{k_i}^X)}}{\sum_{k=1}^{K} \det(\Gamma_k^X)^{-\frac{1}{2}} \alpha_k e^{-\frac{1}{2}(x-\mu_k^X)^T \Gamma_k^{X\,-1}(x-\mu_k^X)}}. \tag{0.19}$$

On peut aussi réaliser un clustering en premier lieu sur les points d'apprentissage (x_i, y_i) en utilisant 4.20. Ce clustering peut être réalisé de deux manières différentes

- On peut assigner un et un seul cluster à chaque couple (x_i, y_i) en prenant le maximum des $\mathbb{P}(\kappa = k_i | (X,Y) = (x,y))$ sur les $i = 1 \dots K$ (maximum a posteriori).
- On peut assigner un clustering flou, c'est-à-dire assigner une probabilité d'appartenance à tous les cluster pour chacune des observations.

Au cours de cette thèse nous avons surtout utilisé le premier type de clustering. Une fois ce clustering réalisé on dispose alors de K bases qui forment une partition de la base d'apprentissage initiale B_A. On construit alors un modèle d'approximation ou expert local sur chacune de ces régions. Tout type de modèle d'approximation peut être construit. Au cours de cette thèse, nous nous sommes intéressé principalement aux modèles suivants

- **modèle polynomial** : construit par simple régression linéaire. L'avantage de ces modèles est la rapidité de construction. Néanmoins ce type d'expert local n'est en général pas suffisamment précis.
- **réseaux de neurones** : permet d'approcher des fonctions plus complexes que les modèles polynomiaux au prix d'un temps de construction plus élevé (choix du nombre de neurones...)
- **fonctions de bases radiales** : pour des raisons de temps de constructions, on a choisi la version interpolante où chaque fonction de base (gaussienne,...) est centré sur un point d'apprentissage. On règle alors un unique paramètre (variance de la gaussienne), le temps de construction est donc réduit par rapport à des réseaux de neurones.
- moindres carrés mobiles : moindres carrés locaux centrés en chacun des points d'apprentissage. Ce modèle est coûteux en temps de construction car implicite et devant être recalculé en chacun des points d'évaluation (résolution d'un système linéaire).

L'avantage des deux derniers modèles est qu'ils présentent, au moins théoriquement, des propriétés d'approximation intéressantes (dont la fameuse convergence spectrale en norme infinie, plus rapide que toute convergence polynomiale) pour peu que la fonction à approcher soit régulière.

Une fois le clustering réalisé et le type d'expert local choisi, on construit donc sur chacune de régions B_A^i un expert local f_i. Le modèle global où l'on recombine tous les experts locaux en un seul est donc

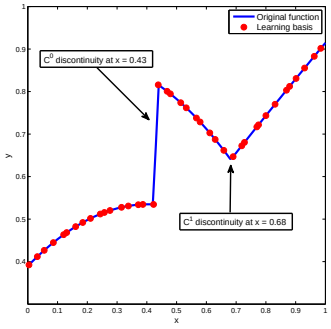
$$f(x) = \sum_{i=1}^{K} \mathbb{P}(\kappa = i | X = x) f_i(x) \tag{0.20}$$

Le terme $\mathbb{P}(\kappa = i|X = x)$ vient donc pondérer l'expert f_i. Là encore on peut choisir deux manière de recombiner suivant le modèle d'approximation qu'on cherche à construire

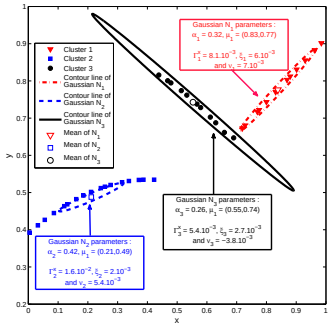
- Ou bien on calcule une probabilité d'appartenance floue pour chaque nouvel x et dans ce cas on appliquera tous les experts locaux pour un x donné. Le modèle final sera alors régulier (C^∞) et pourra par exemple être utilisé comme modèle de substitution dans une algorithme d'optimisation différentiable. On parle alors de recombinaison lisse.

- Ou bien on calcule une probabilité d'appartenance par maximum a posteriori. Un nouvel x appartiendra donc à un et un seul cluster i et le modèle global renverra donc uniquement la valeur de l'expert correspondant. On parle alors de recombinaison discontinue.

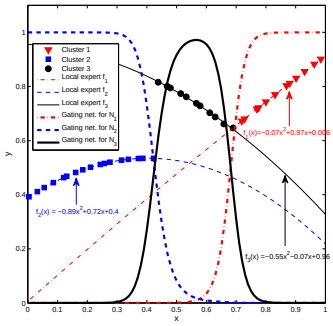
Dans cette thèse cette méthode originale de construction de modèles d'approximation est appliquée à de nombreuses reprises, en particulier pour l'approximation des contraintes de stabilité: approximation du facteur de réserve de flambage en paramètres de stratification, approximation des résultats de codes de stabilité Airbus,...mais aussi dans l'approximation de résultats d'optimisation (fonction valeur optimale) qui présentent les mêmes caractéristiques. A titre d'exemple, on a représenté Fig. 0.4 les résultats de l'approximation avec des experts locaux linéaires sur des fonctions analytiques linéaires par morceaux (dont la fonction valeur absolue) en présence de bruit (différentes variances selon les régions). On compare dans chacun des cas la reconstruction avec la fonction originale, on voit dans les deux cas que la reconstruction discontinue permet de localiser les points de discontinuités et se révèle assez précise. Le clustering permet en effet de séparer les zones et d'entraîner un expert local suffisamment bon dès lors que le clustering a séparé correctement les points d'apprentissage. Dans de nombreux cas, cette stratégie s'est révélée adaptée dans la mesure où les composantes était correctement détectée, néanmoins, la qualité du clustering EM reste assujettie à l'échantillonnage des points d'apprentissage. D'autre part la détermination du meilleur nombre de cluster reste délicate. Au cours de cette thèse, après avoir constaté l'insuffisance des critères classiques (tels que la log-vraisemblance, AIC (Akaike Information Criterion) ou BIC (Bayesian Information Criterion)), on a mis une point une stratégie basée sur outils classiques d'estimations d'erreur dans les cas de régression



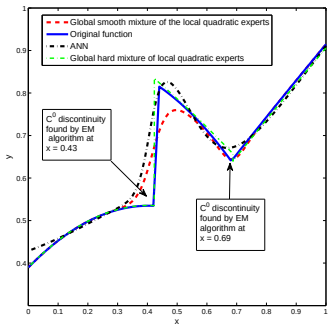
(a) Fonction à approcher et base d'apprentissage



(b) Résultats de l'algorithme EM et gaussiennes associées

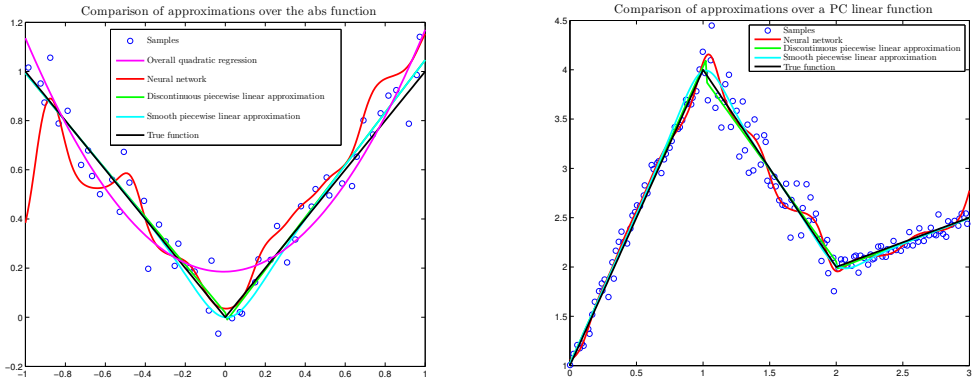


(c) Experts locaux (quadratiques) et réseaux portiers



(d) Recombinaisons

Figure 0.3: Présentation de l'algorithme sur un cas test 1D
a) Fonction à approcher et base d'apprentissage. Cet exemple fait apparaître un point de discontinuité et un point de discontinuité de la dérivée. On fixe ici le nombre de cluster à 3.
b) Clustering EM de la base d'apprentissage. On a tracé les lignes de niveau des fonctions quadratiques associées aux trois composantes gaussiennes: métrique centrée sur la moyenne et tournée et dilatée suivant la racine carrée de la matrice de variance-covariance (distance de Mahalanobis associée à chacune des gaussiennes. On note que dans cette métrique les boules sont des ellipsoïdes.
c) Apprentissage des experts locaux (en l'occurrence modèles quadratiques) et réseaux portiers associés ($\mathbb{P}(\kappa = i|X = x)$).
d) On trace les deux modèles globaux obtenus: recombinaison lisse et recombinaison discontinue. On a comparé ces deux modèles avec un réseau de neurones et la vraie fonction. Le modèle discontinu est de loin le plus précis, néanmoins il crée une discontinuité artificielle à $x = 0.69$



(a) Fonction valeur absolue avec bruit gaussien (b) Fonction linéaire par morceaux avec bruit gaussien

Figure 0.4: Résultats d'approximation avec experts locaux linéaires sur des fonctions linéaires par morceaux avec bruits

(validation simple/croisée).

On présente aussi les résultats de l'application de cette stratégie pur l'approximation de calcul de flambage Fig. 0.5 et 2.90 on a appliqué cette stratégie à l'approximation du facteur de réserve pour des séquences d'empilement orthotropes (deux variables $\xi^D_{1,2}$). On voit clairement Fig. 0.5 que la fonction à approcher fait apparaître quatre zones (minimum de quatre hyperplans bien que dans ce cas précis la linéarité ne soit pas prouvé théoriquement). Dans ce cas, notre stratégie de choix du meilleur nombre de cluster donne bien 4 et l'on voit Fig. 2.90 que le clustering avec 4 composantes semble bien diviser la base d'apprentissage, ce qui est confirmée par le modèle reconstruit (recombinaison discontinue).

Application et adaptation des schémas de décomposition

L'autre principal axe de recherche de ce travail consiste en l'adaptation et l'application de schémas de décomposition. Cela signifie par exemple la recherche d'un schéma de décomposition qui permet de retrouver le même optimum que le problème non décomposé. Différents schémas de décomposition ont été implémentés et testés notamment dans (Merval 2008), néanmoins aucun de ces schémas ne semblait trouver le

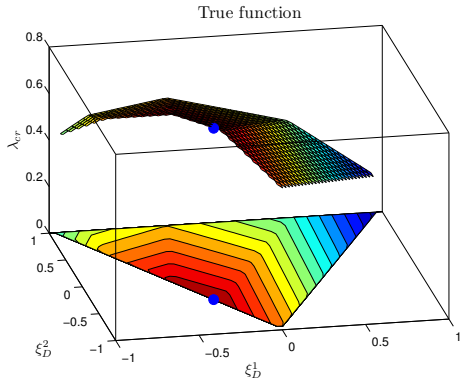
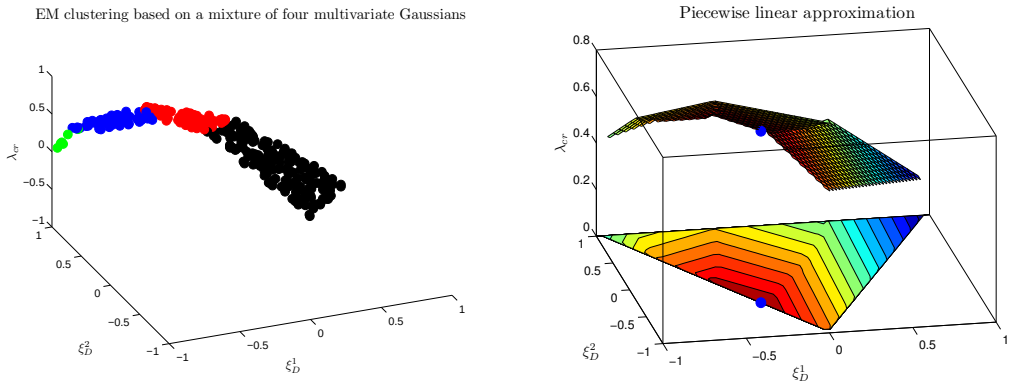


Figure 0.5: Fonction à approcher : facteur de réserve de flambage pour des séquences orthotropes



(a) Clustering de la base d'apprentissage

(b) Approximation linéaire par morceaux

Figure 0.6: Clustering et approximation

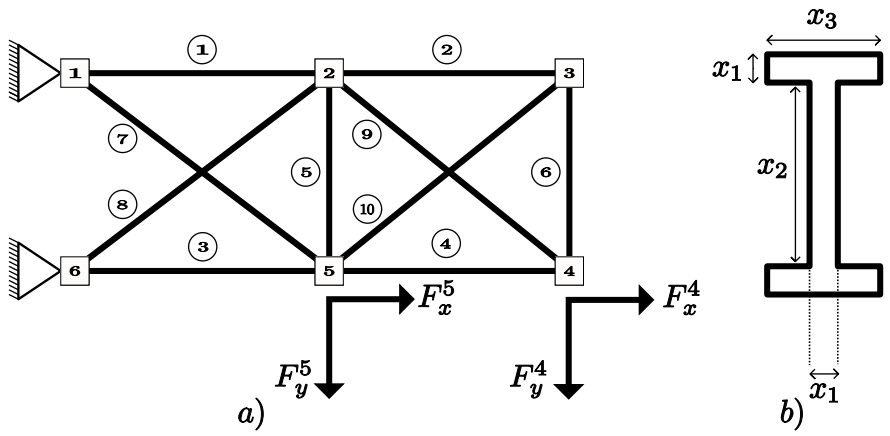


Figure 0.7: a) treillis 10 barres b) profil en I

même optimum que le problème non décomposé. D'autre part, il a aussi fallu adapter ces schémas aux spécificités du composite.

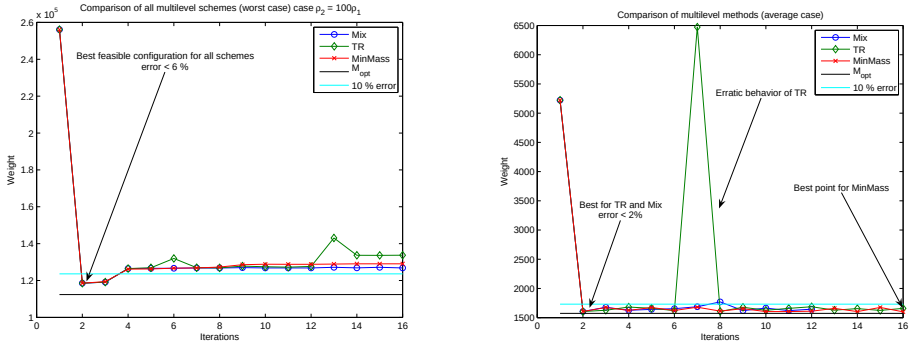
Validation sur un cas test classique de l'optimisation de structure

Dans un premier temps, on a utilisé un cas test classique de l'optimisation de structures : le treillis 10 barres pour comparer les différents mononiveau et biniveau. Il s'agit d'un treillis composé de 10 barres, chacune de ces barres étant profilées en I (de la même manière qu'un raidisseur). On cherche à minimiser la masse totale du treillis sous des contraintes de rupture ($|\sigma| \leqslant \sigma_{0.2}$) et des contraintes de flambage : flambage de colonne (toute la section) et flambage local de l'âme du raidisseur. On a représenté ce cas test Fig. 0.7. Ce cas test reflète bien les deux niveaux d'analyse mécanique de nos problèmes d'optimisation de structure. La redistribution des efforts (déplacements, déformations, contraintes, efforts et donc les contraintes d'optimisation associées) ne dépend que des aires de section des barres alors que les contraintes de flambage font intervenir la description détaillée du profil: moment d'inertie I (qui dépend des dimensions locales) pour le flambage de colonne et les dimensions de l'âme (considérée comme une plaque) pour le flambage local. En plus d'avoir une structure proche de nos problème, ce cas test est aussi connu pour montrer l'insuffisance d'un critère mécanique d'optimalité classique (**Fully Stressed Design**) qui permet de décomposer le problème d'optimisation en autant d'optimisation que de barres. On peut même artificiellement

exagérer le caractère non-FSD optimal de ce treillis en changeant considérablement les densités volumiques des barres diagonales. En effet, le critère FSD permet de trouver le vrai optimum dans le cas isomatériau alors que dans le cas où la densité volumique des barres diagonales est modifiée, le critère FSD est totalement sous-optimal. Ce dernier cas est donc été notre cas de référence pour tester et valider les schémas de décomposition. Ce cas bien qu'artificiel est plus réaliste car les grandes structures composites peuvent être considérés comme formées de matériaux différents (dans la mesure où les **modules de rigidités apparents** E_x, E_y, G_{xy} et ν_{xy} peuvent varier du simple au décuple avec les proportions, voir dans le Chapitre 3 Fig. 3.1). Les grandes structures de fuselage composites peuvent donc être considérées comme des grandes structures hyperstatiques et faites de matériaux différents, ce qui est précisément les cas où les algorithmes d'optimisation basés sur le critère FSD sont sous-optimaux.

Ce cas test a donc permis dans un premier temps de comparer les différentes formulations mononiveau, puis dans un second temps les schémas de décomposition. Les différentes formulations mononiveau se sont révélées équivalentes en termes d'optimum trouvé. Les formulation basées sur la structure bloc-angulaire se sont même avérées un peu rapides en termes de convergence et surtout plus simples en termes de calcul de sensibilités analytiques. L'introduction des variables de rigidités Y s'est donc avérée justifié même dans le cas mononiveau. Dans le cas multiniveau, on a comparé les 4 schémas déjà présentées. Tous ces schémas se sont montrés sous-optimaux pour toutes les configurations sous tous cas de charge. Bien que donnant de meilleurs résultats que les schémas basés sur le critère FSD, aucun de ces schémas n'a semblé converger vers l'optimum réel. Les résultats sur l'ensembles des configurations et des cas de charges ont été les suivants:

- **critère FSD** : erreur relative de 6% en 14 itérations
- **Rigidité Cible** : erreur relative de 3.2% en 5 itérations
- **MinMass** : erreur relative de 3.1% en 4 itérations
- **Mix** : erreur relative de 2.9% en 5 itérations
- **MaxMarge** : erreur relative de 12% en 4 itérations



(a) Comparaison des schémas de décomposition (pire cas))

(b) Comparaison des schémas de décomposition (cas moyen)

Figure 0.8: Comparaison des schémas de décomposition dans le cas non-FSD optimal : pire cas de charge et comportement typique

Les schémas Rigidité Cible, MinMass et Mix se comportent de la même manière et les différences sont assez faibles. On a représenté leur comportement typique Fig. 0.8. On a pu constater un léger avantage au schéma Mix.

Le schéma MaxMarge qui est pourtant le seul de ces schémas pour lequel il existe des propriétés théoriques d'équivalence entre le problème non décomposé et le problème décomposé a un comportement très erratique et ne converge que rarement. Une étude de l'historique des contraintes montre que la contrainte égalité n'est que très rarement satisfaite. On a représenté Fig. 0.9 ce comportement erratique ainsi que le comportement en perturbant le point initial, on observe que le comportement varie énormément malgré une faible perturbation du point initial.

Sur la base de cette sous-optimalité, nous avons donc cherché à modifier l'implémentation et à appliquer le schéma de décomposition QSD dans une version totalement biniveau : optimisation locales implicites dans la boucle d'optimisation supérieure. Le choix de la formulation QSD s'est fait sur la base des résultats théoriques, puisqu'il s'agit du seul schéma pour lequel on soit sûr que tout minimum trouvé par décomposition (QSD) est bien minimum du problème original. L'adaptation faite par rapport au schéma QSD original réside dans la relaxation de la contrainte égalité interniveau ($\varphi(x) = Y$) traitée par pénalisation quadratique inexacte plutôt que par contrainte égalité qui mène à un comportement erratique de l'algorithme. La formulation est alors

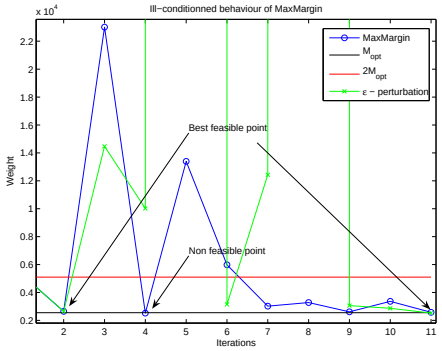


Figure 0.9: Comportement erratique de MaxMarge

- Niveau global :

$$\begin{array}{ll} \min_{Y,b} & M(Y) = \sum_{i=1}^{N} m_i(Y_i) + \sum_{i=1}^{N} b_i \\ \text{t. q} & \left\{ \begin{array}{l} c_{glob}(\Phi(Y_1, \ldots, Y_N)) \leq 0 \\ \mu_i^\star(Y_i, \Phi_i, b_i) \leq 0 \quad \text{pout} \quad i = 1 \ldots N \end{array} \right. \end{array} \tag{0.21}$$

- où $\mu_i^\star$ pour $i = 1 \ldots N$ est la fonction de couplage calculée à partir de l'optimisation de niveau inférieure :

$$\begin{array}{ll} \min_{(\mu_i, X_i)} & \mu_i \\ \text{t. q} & \left\{ \begin{array}{l} c_{glob-loc}^{(i)}(Y_i, \Phi_i, X_i) - \mu_i \leq 0 \\ c_{loc}^{(i)}(X_i) - \mu_i \leq 0 \\ \frac{1}{2}||Y_i - \varphi(X_i)||^2 - b_i - \mu_i \leq 0 \end{array} \right. \end{array} \tag{0.22}$$

L'une des difficultés de l'implémentation réside dans l'utilisation des dérivées post-optimales. De fait le problème d'optimisation au niveau supérieur fait intervenir comme contrainte une quantité calculée par des optimisations locales. Si un algorithme à base de gradient est utilisé au niveau supérieur, on devra calculer les dérivées de la fonction $\mu^\star$. Si ce calcul s'effectue par différences finies alors le problème deviendra beaucoup trop coûteux. On doit donc évaluer cette dérivée de manière analytique:

$$\frac{d\mu_i^\star}{dY_i} = \frac{\partial \mu_i^\star}{\partial Y_i} + \frac{\partial \Phi_i}{\partial Y_i} \frac{\mu_i^\star}{\partial \Phi_i} \tag{0.23}$$

et

$$\frac{d\mu_i^\star}{dY_j} = \frac{\partial \Phi_i}{\partial Y_j} \frac{\mu_i^\star}{\partial \Phi_i} \tag{0.24}$$

avec les dérivées post-optimales suivantes

$$\frac{\partial \mu_i^\star}{\partial Y_i} = \lambda^T \frac{\partial c^{(i)}}{\partial Y_i} \tag{0.25}$$

et

$$\frac{\partial \mu_i^\star}{\partial \Phi_i} = \lambda^T \frac{\partial c^{(i)}}{\partial \Phi_i} \tag{0.26}$$

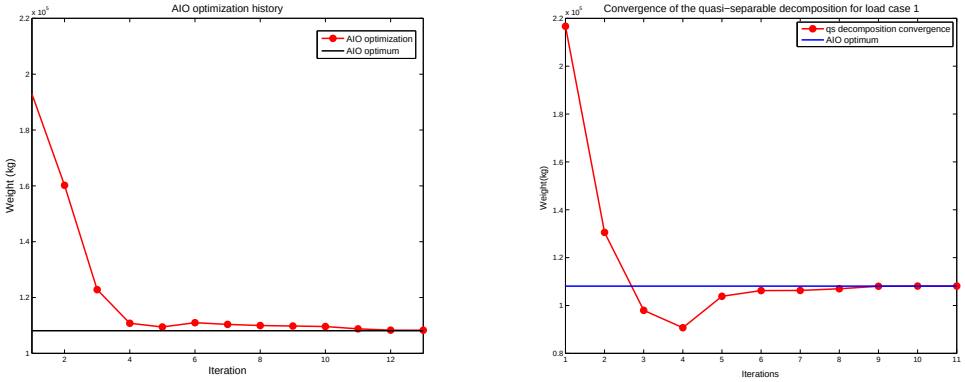


Figure 0.10: Comparaison de QSD et du problème non décomposé

Les résultats (par exemple Fig. 0.10 où l'on a représenté l'historique de la masse au cours de l'optimisation avec QSD et l'historique pour l'optimisation non décomposée) montrent que notre adaptation de QSD pour ce treillis trouvent bien le même optimum que les versions non décomposées. Au cours d'une comparaison avec de nombreuses optimisations, le schéma QSD a convergé dans la plupart des cas et en appelant moins d'analyse éléments finis et d'itérations globales de l'optimiseur que le schéma non décomposé.

Application et adaptation aux composites sur une structure réaliste de fuselage

Sur la base des bons résultats obtenus sur le treillis, nous avons aussi cherché à adapter le schéma QSD aux composites et en particuliers aux paramètres de stratification. On a donc mis en place le schéma suivant

- $M(Y) = \sum_i^N A_i + \sum_i^N b_i$ masse totale pénalisée par les budgets

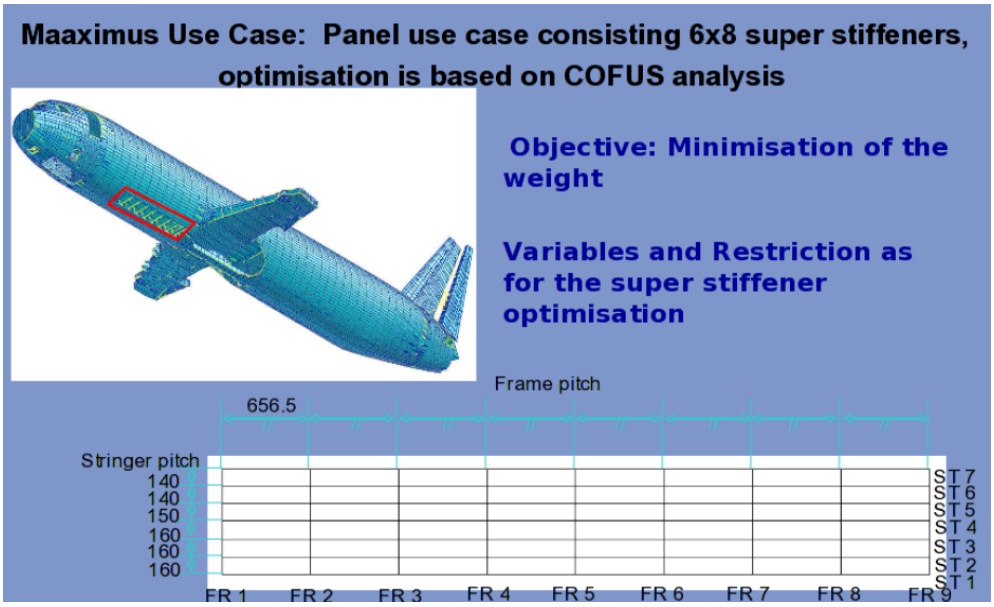


Figure 0.11: Panneau Maaximus

- L'optimisation locale est alors

$$\begin{array}{ll} \min_{(\mu_i,(e_i,\xi_D^{(i)})} & \mu_i \\ \text{t. q} & \begin{cases} RF^{(i)}_{glob-loc}(\Phi_i, e_i, \xi_D^{(i)}) - \mu_i \leq 0 \\ c^{(i)}_{comp}(\xi_A^{(i)}, \xi_D^{(i)}) \leq 0 \\ c^{(i)}_{loc}(e_i) - \mu_i \leq 0 \\ \frac{1}{2}||A_i - \varphi(X_i)||^2 - b_i - \mu_i \leq 0 \end{cases} \end{array} \tag{0.27}$$

et nous avons implémenté ce schéma sur une structure réaliste : panneau Maaximus 6×8 super-raidisseurs. Il s'agit d'un panneau sous le fuselage principalement en compression, on a représenté ce panneau Fig. 0.11. Les contraintes d'optimisation sont soit des contraintes en déformations soit des contraintes de stabilité (tolérance aux dommages, flambage, post-flambage,...) calculées à l'aide d'un outil Airbus appelé Cofus. L'implémentation a nécessité d'une part le lien vers un solveur éléments finis avec analyse de sensibilité (en l'occurrence MSC.Nastran SOL200), un optimiseur global (Boss Quattro), des optimiseurs locaux (Matlab) et enfin un modèle d'approximation de l'outil Cofus. A convergence le schéma QSD donne la même masse que le schéma non décomposé avec moins d'itérations globales.

Conclusion et perspectives

Cette thèse a cherché à rendre possible l'optimisation de grandes structures aéronautiques composites par le biais de deux axes

- la décomposition du problème d'optimisation
- le développement de modèles d'approximation adaptée aux contraintes de stabilité

Après avoir analysé la structure du problème d'optimisation ainsi que le comportement des contraintes, nous avons pu développer et adapter des schémas de décomposition aux spécificités du composite et développer une méthodologie adaptée au caractère non régulier des contraintes de stabilité. L'application de ces schémas sur ces cas test académiques et réalistes montre une bonne adéquation par rapport aux besoins industriels. En particulier, l'adaptation du schéma QSD nous permet de retrouver le même optimum que le problème non décomposé avec bien souvent moins d'appels aux analyses éléments finis et d'itérations globales. La méthodologie de construction de modèle d'approximation s'est révélée cruciale et adaptée car suffisamment précise pour remplacer les coûteuses analyses de stabilités locales (flambage).

Pour valider cette adaptation du schéma QSD, l'une des perspectives immédiates serait d'appliquer cette décomposition à un cas test de plus grande taille (couronne de fuselage ou fuselage entier). Une telle application nécessiterait une implémentation parallèle ce qui n'a pas été fait au cours de cette thèse. De la même manière, la construction du modèle réduit pourrait elle aussi s'effectuer de manière parallèle de la construction même de la base d'apprentissage à la construction des experts locaux. D'autres perspectives, plus théoriques, consisteraient en l'étude approfondie et du schéma de décomposition (convergence locale) et des propriétés d'approximation du schéma proposé (convergence spectrale par morceaux par exemple). Enfin pour se rapprocher du design final, l'une des perspectives en le traitement et l'introduction de contraintes interéléments (continuités entre les plis). De manière générale, les contributions de cette thèse pourraient être appliquées à d'autres domaines (dynamique des structures, mécanique des fluides numériques) et enfin au domaine de l'optimisation multi disciplinaire.

General introduction

A major concern in aerospace industry has always been the weight of the overall structure. Any extra weight in the air leads to excessive costs at the end of the life cycle of the aircraft. It is even more critical than ever both for aircraft designers (such as Airbus and Boeing) and also for air liners with the fuel prices skyrocketing. For the aircraft designer the objective is to maximize the ultimate payload, allowing one aircraft designer to be better able to compete the other one, possibly the other ones. Besides market and competition concerns, the lightest structure is always preferable also for security reasons. Indeed, quoting the Federal Aviation Administration official report 'Aircraft weight and balance handbook' ((Handbook 2007))

'Excessive weight reduces the efficiency of an aircraft and the safety margin available if an emergency condition should arise'

When an aircraft or any aerospace structure is designed, it is always made as light as possible. 'As possible' should be understood as 'with respect to required structural strength allowables'. Composite materials have many often equivalent stiffness properties than the traditional aluminum structures for lighter designs. They also offers a wider range of potential designs because of their anisotropic properties. Both long-range aircraft designers are now using them to design primary structures of aircrafts. However their use is, of course, not entirely problem-free: conduction issues, specific failures issues... and it is not clear whether or not composite structures are always cheaper (even though lighter) when cost considerations come into play, especially because composite material manufacturing and assembly has not reached the level of maturity of the metal manufacturing and assembly. Eventually, one of their most interesting features happen to add complexity to the design phase. Being more flexible in terms of sizing, they, of course, can be tailored to specific structural needs but we

also pay an extra cost for this wider design space. By allowing more potential designs and making the material in itself a variable of the design, situation becomes more complicated than in the case of metal structures. For metal structures, only dimensions were to be tuned, this is not an easy task...but it becomes much more complicated in the case of composite materials, mainly because of their discrete mechanical behavior in contrast with the continuous mechanical behavior of metal.

This is where structural optimization comes in the scene. Structural optimization has a long and independent history, though never far away from real life applications. It traces back to the first work in calculus of variations and has a sound mathematical basis. It aims precisely at studying and automating the whole design process and perfectly suits aircraft overall weight minimization. By writing the problem in a formal way, studying optimality conditions and criteria, it helps find new designs in an automated manner that a humanly limited designer could not reach, even the most-skilled one. Structural optimization strongly relies on mechanics, on applied mathematics and on a computer implementation. It can integrate several antagonist objectives (multi-objective optimization), uncertainties (robust optimization) and also be a part of the more general multi disciplinary optimization field.

This work fits in the structural optimization field. Our main objective is to enhance the treatment of very large structural optimization problems including the design of a specific type of composite materials: laminates. The pitfalls of such large problems lie first in the size of the problem that is hardly manageable with current computer resources and second with the number of analyses needed to run such large optimizations. Our structural optimization problems (typically fuselage structures) have quite an interesting feature that the mechanical analyses they involve are hierarchical and exhibit a structure that reflect the real structural components. In that framework, such optimizations can be thought as the optimizations of many interconnected systems. This leads to a large problem who could possibly be decomposed and then easier to solve provided we correctly treat the coupling of all these elements. For fuselage structures, this coupling remains essentially the internal loads redistribution that impact the stability analysis at sub-element level. In that context, the whole optimization problem can be thought in the spirit of multi disciplinary optimization problems where each

sub-component is considered as a discipline. Our problem though fits in the multilevel optimization framework since it involves only one discipline (mechanics) and two different levels of mechanical analysis: internal load redistribution vs. stability analysis, each of them being performed at a different level of representation of the structure. The internal load redistribution is computed on a very coarse representation while stability analysis needs a more detailed representation. This naturally advocates for a bilevel optimization strategy

There are though very few differences between multilevel optimization and multi disciplinary optimization at least in terms of resolution, practical algorithms and also in terms of the innovations required to allow the treatment of their respective problems. Quoting Jaroslaw Sobieski, one major figure of the field, in a recent talk (Sobieski), multi disciplinary optimization tools still require research and development effort in the following areas (among many others)

- **Decomposition**
- **Sensitivity analysis**
- **Surrogate modeling**

Decomposition is concerned with breaking up the large optimization problem into many smaller problems allowing an efficient treatment. As far as optimization algorithms using derivatives are involved, **sensitivity analysis** is often required to compute these derivatives in an exact and quick way. Even though a suitable decomposition is applied, the decomposed optimization problem can still be intractable because of many repetitive legacy codes evaluations, **surrogate modeling** offer strategies to approximate the results of such codes. These three points are the essential research axes in this work.

This work is intended to provide an innovative strategy to treat the optimization problem of a large composite aircraft structure. The three mentioned fields are investigated and contributions to research effort are given allowing at the very end to treat the large optimization problem. Being not really devoted to one subject over the others, this work is intended to improve the treatment of such problems by enhancing an existing bilevel

optimization strategy, the **Quasi separable Subsystem Decomposition** carried out by several researchers (Sobieski, Haftka, Liu and Watson) with innovative and original contributions ranging from the investigation of theoretical optimization properties, the investigation of the optimization constraints, the development of a new surrogate modeling capability, the numerical investigation of bilevel optimization schemes and finally on the basis of all this material treat a realistic aircraft structural optimization problem.

1 Preliminaries

This chapter describes in more details the problem we are to solve and the different difficulties that we encounter when sizing large composite structures such as fuselage structures. Basics and fundamentals of structural optimization are presented, we also give a brief description of the typical composite material that we address in this work: laminated composites. The organization of the sizing process and the natural hierarchy of the analysis of such aircraft structures are highlighted, enforcing the needs first for a decomposition scheme that allows to break the large problem into many sub-problems and second for a bilevel optimization that matches the different levels of analysis of the structures.

1.1 Fundamentals of structural optimization

Structural optimization is concerned with getting the best out of human made structures. This definition, though simple, encompasses most of the applications of structural optimization ranging from aerospace industry, automotive, buildings,... any human realization where it is expected to find a better feasible design than the one found by art or how-do of the engineers. Indeed the main objective of structural optimization is to automate the design process that has been done through the early history of industry and its ultimate objective is not only to get a better design but the best feasible design. It basically includes tools ranging from applied mathematics (study of partial differential equations, mathematical optimization, numerical analysis, sensitivity analysis, statistics), mechanics (modeling, mechanical optimality criteria) and computer sciences (effective resolution of a partial differential equation (pde), of an optimization problem, programming).

'Best' should be considered carefully for it does not necessarily refers to a physical measure rather to a measure of performance. This performance can be of course based

on a physical quantity (compliance, stiffness,...) but also connected to cost (if a design is a bit heavier but cheaper to manufacture). In the aerospace industry the main criterion to be optimized is obviously the weight of the overall aircraft, satellite,...However structural optimization is not aimed only at solving weight minimization problems, but also multi objective optimization (several objectives being often opposed), robust optimization (where some quantities are not free of errors are replaced by random variables).

'Design' also refers to many different possibilities and involves degrees-of-freedom that the engineer-designer has to modify. These are design variables and might be geometrical dimensions, shapes, presence of holes in the structures...To make it more real let's take examples: for a fuselage structure made of reinforced panels, the geometric description of the stringer or thicknesses of the panels are typical geometrical dimensions that the engineer-designer should modify to get to the best design, for aerodynamic tailoring (maximizing the lift of the wing) the typical degree-of-freedom of the engineer is the shape of the wing that has to be parameterized through some variables, however these two types of variables does not modify drastically the design. Indeed, consider geometric dimensions of a stringer profile, there is no way to change the type of profile. This geometric description does not allow to change the concept used. The same way parameterization of the shape profile only allows changes of the boundary of the structure to design but not the concept itself. The ultimate structural optimization is then the one that allows changes of the concept itself: it does not only size or find a good shape for some salmon but finds out that a salmon is needed or not and where it should be placed over the wing. Borrowing once again an example from Sobiesky, parametric or shape optimization could find out that a biplane would perform better if one of the two superimposed wings would be canceled (by making the dimensions of the wing vanish), however only material (or topology) optimization would possibly find out the other way around, namely creating another wing from a monoplane initial design.

Finally **'feasible'** refers to a structure that will withstand the loads it will endure throughout its life cycle. In particular one has to guarantee that the final design of the structure respects some set of norms and certifications. These norms or certifications involve physics: the building should resist its own weight, the aircraft should withstand

loads it endures during different flight phases, the large and thin bridge should resist against flutter effect...which means that the feasibility of the structure refers to a physical model. We then introduced the three main ingredients of a structural optimization problem which are

- **the objective function**: a criterion that is to be minimized or maximized
- **the optimization variables**: the set of degrees-of-freedom that the engineer can possibly modify. These variables might be scalar quantities (geometric dimensions), functions (shape optimization) or the domain of the structure itself
- **the physical model**: the equations (usually partial differential equations) that model a physical response that is expected to remain below a given value (e.g strain allowable) or that is to be maximized (e.g compliance). As far as mechanics are involved a typical physical model involved in structural optimization is of course linear elasticity model...Structural optimization problems that involve an interaction with flows (flutter constraints for instance) has to consider a model for the flows that can be Euler or Navier-Stokes equations.

One important characteristics of the physical model that is to be integrated in the optimization is that it should be

- **a physical model of some physical behavior that is very well known**. One has to use robust and validated numerical approximations of the pde of these models. Indeed, the optimization process will iterate and possibly affect physical parameters (material characteristics...) of a given model. If the limit of validity of the physical model is quite fuzzy or if the model can not be trusted for some values of the optimization variables, an optimization that involves this model would not be very useful and significant. For instance, linear elasticity system of pde's is well known (well-posedness,...) and the numerical approximations have been used for now more than 50 years[1]. So engineers, physicists and applied mathematician have a quite clear of this model. As opposed to this models, crack

[1] and even longer. In the literature on the history of FEM methods (see (Felippa 2000)), the aeroelastician M. J. Turner from Boeing company is often recognized as the initiator of FEM within space and aircraft design offices. His joint paper *'Stiffness and Deflection Analysis of Complex Structures'* together with R. W. Clough, H. C.Martin and L. J. Topp ((Turner *et al.* 1956)) is considered as the start of current FEM.

propagation models did not reach this maturity and would hardly be a physical model included in a structural optimization.

- **a physical model for which numerical approximation schemes are also validated and not too computationally demanding**. The optimization process will iterate over the objective functions or constraints that involve the numerical approximation of the model (and possibly derivative of some physical quantities if gradient-based optimization algorithms are involved). Even though there exist strategy to alleviate this computational burden (e.g surrogate models), a structural optimization that involves for which a sole run takes a few weeks is hardly to be performed easily. Once again, the linear elasticity models fits in this category since finite elements have been used for several decades and are not the most demanding approximation methods in comparison for instance with Direct Numerical Simulation (DNS) schemes for computational fluid dynamics or even Large Eddy Simulation (LES) that demand excessive computational resources for an optimization to be solved without any further strategy.

Let us just conclude about optimization algorithms. Structural optimization makes use of almost all practical optimization algorithms from mathematical optimization with some specific adaptations with regards to the variables and functions involved

- the classical **gradient based and quasi-Newton methods** (Sequential Quadratic Programming SQP...) are classical tools for differentiable structural optimization. Adaptation to mechanics led to specific linear approximation for constraints: such as reciprocal approximation, convex linearization (CONLIN (Fleury 1989)), method of moving asymptotes (MMA (Svanberg 1987))...They are often accompanied with theoretical convergence results often practical and fruitful (quadratic convergence to a **local minimum**) provided they are used with a good initial point. A valuable reference presenting most of these algorithms in the frame of structural optimization is (Haftka & Gurdal 1992).

- whenever variables are discrete or the functions involved are discontinuous many algorithms can be used (branch and bound, integer linear programming...) that all try to explore as much as possible the design space. Many heuristics that lie in the **global optimization** field can also be used: simulated annealing, ant

colony, particle swarm optimization...one of them is now used on a daily basis and especially for composite design: **genetic algorithms** or more generally **evolutionary algorithms**. Most of these heuristics are nature-inspired techniques that do not require, at least in their standard versions, the derivative. Another valuable reference is (Gürdal *et al.* 1999).

1.2 From isotropic to anisotropic structures: composite structures

The use of composites material in aeronautic structures is far from new. Their specific strength ratio when compared to their specific weight allow considerable weight saving with regards to traditional aluminum structures. First application for radar domes in commercial aircrafts traces back to the early 40's. Carbon Fiber Reinforced Plastics (CFRP) have been used from the early seventies: vertical tail plane of the Airbus A-300, elevators of B-727 and B-737 were made of composites. Boeing uses composites for secondary structures of the B-777 (ailerons, rudder, trailing and leading edges). Airbus reached a milestone when introducing composite material in primary structures, e.g parts of the aircraft that essentially carry loads, for the central wing box of the A-380 (see Fig. 1.1 where we depicted the composite material distribution in the A-380).

The proportions of composite material in the coming A-350 reaches more than 52% of composite primary structures. In definitive, the part of composites in civil aircraft has been continually increasing since the modern age of aerospace industry. We depicted Fig. 1.3 the evolution of the proportions of composite materials in Airbus aircrafts.

In this thesis we only address **laminates**. Laminates are made of several layers of different orientations bonded together, each layer being made of long fibers reinforced by a matrix. A laminate of N layers or plies is represented by the sequence of orientations

$$[\alpha_1/\ldots/\alpha_N] \tag{1.1}$$

We depicted such a laminate Fig. 1.2. Composite materials offer a wider range for conception because of their anisotropic properties that can be tailored to specific structural parts. Regarding specific composite constraints, composite structures are limited

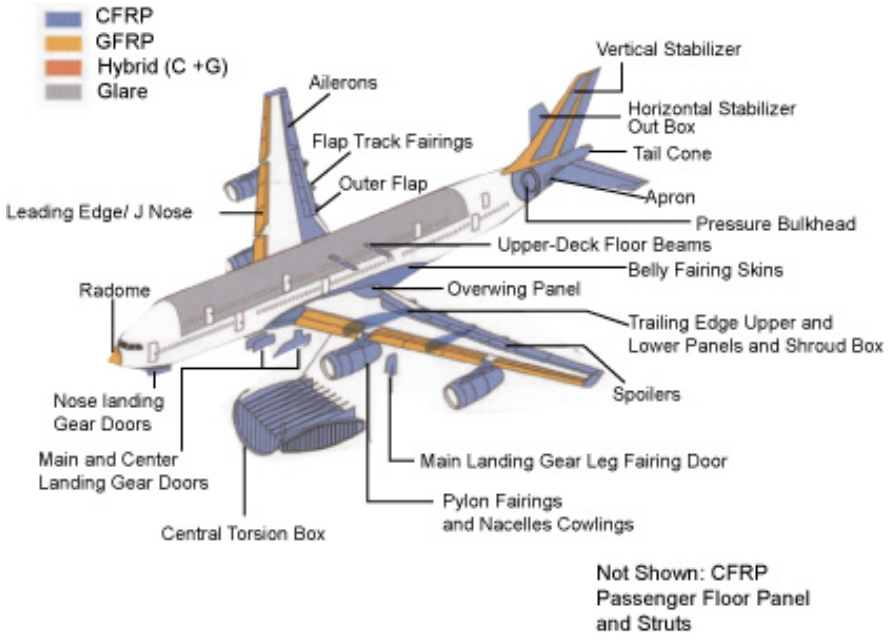


Figure 1.1: Distribution of composite material in the A380

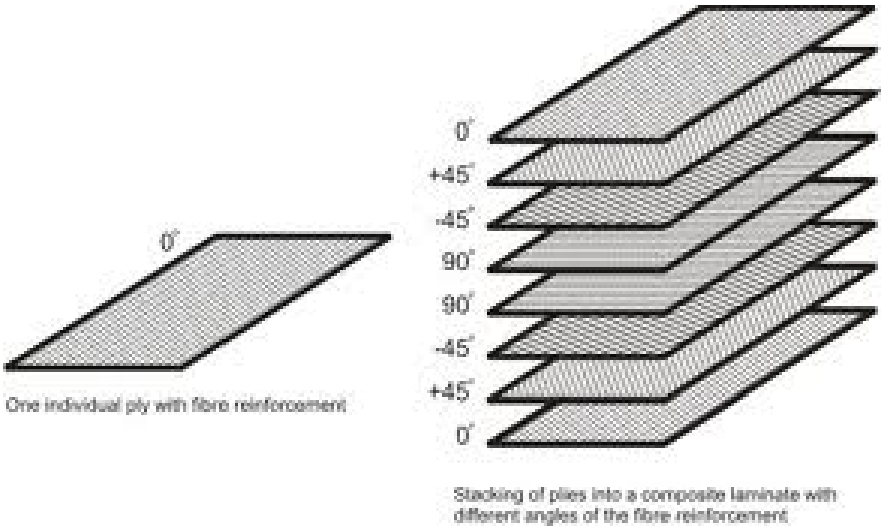


Figure 1.2: Laminates

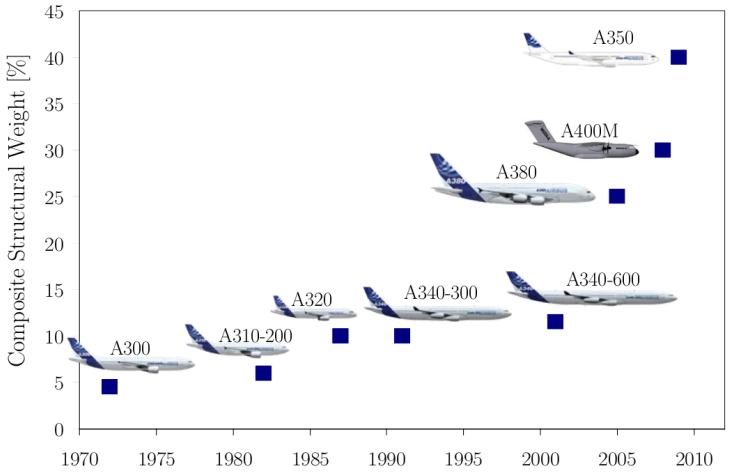


Figure 1.3: Evolution of the composite proportions in Airbus aircrafts

by strain allowables that are determined through coupon tests, while for metal structures, the mains limitations are stress allowables. Typical phenomena for composite laminates design should prevent are

- fiber failure
- matrix failure
- fiber/matrix debonding
- delamination
- compressive and shear buckling

Typical orientations used in this thesis are $[0°/45°/-45°/90°]$. Feasibility rules that come from both mechanical constraints but also manufacturing constraints are for instance

- **symmetric stacking**: the whole stacking should be symmetric. This avoids extension (in-plane)/ flexural (out-of-plane) coupling.
- **balanced stacking**: as many 45◦ plies as −45◦ plies. This is typical mechanical constraints since it cancels the stretching/shear coupling at panel level and then cancels the bending/twisting coupling at the whole structure level.

- **disorientation**: no more than 45◦ of angular difference between two successive plies. This is another mechanical constraint, it is expected to prevent the inter-laminate stress concentrations.

- **adjacency**: no more than 4 successive plies of the same orientation. This prevents delamination in case of impact.

- **proportions**: we impose a minimum proportion for each orientation (typically 8 to 10%).

These rules are not easy to handle in an optimization problem. Indeed, some of them are completely discrete (disorientation, adjacency) while others can be treated in a continuous manner by means of inequality or equality constraints (proportions, balanced). This illustrates one of the difficulties when sizing composite laminate structures: such an optimization is by nature discrete, while mechanical behavior (internal load redistribution) is at global level (or macroscopic level) continuous. This is illustrated in the following section where we describe the modeling and the structural analysis of the overall structure.

1.3 Modeling of the structure and natural hierarchy of the structural analysis

The structural sizing of an aircraft at Airbus can be decomposed in three main stages

- **Conceptual sizing**: choices are made between different design alternatives. The overall dimensions are set resulting in an estimated weight.

- **Preliminary sizing**: a finite element representation is built including stringer definition, stringer and frames pitches. Stiffness's properties are checked for a set of load cases arising from aerodynamics, cabin pressure, trailing edge of the empennage....At this point, the whole structural has to withstand to

- **Detail sizing**: at this stage, details of geometry and assembly process's are set. Clips, doors, rivets, bolts are precisely described. Regarding numerical simulation, more advanced models (very fine meshes, non linear finite element methods) are used for instance to check stress concentrations.

Our analyses lies in the preliminary sizing stage. Large-scale optimization can not be performed for the moment at the detail sizing stage mainly because of the high computational cost of the mechanical analyses. Typical optimization variables at this stage are stringer pitches, frame pitches, panels thicknesses at a global level and at panel level detail of the geometry of the stringers. However, we still have different level of sizing because of the hierarchy of the preliminary sizing that first sizes the structure with respect to global stiffness properties (strains...) and also to stability constraints. These different levels of sizing implies different levels of mechanical analysis that are the backbone of our decomposition/bilevel strategy. For a typical fuselage structure, the modeling and then the internal loads redistribution are based on mechanical assumptions that are described here. At panel (or sub-element) level, the main phenomenon we want to prevent is instability and more specifically buckling.

1.3.1 Plane stress problems

A fuselage structure is an assembly of many thin panels reinforced in the longitudinal direction by stiffeners (or stringers). In the orbital direction, the panels are supported by frames. The conceptual sub-element level considered at local level is coined super-stiffener, it is the artificial element made of the stiffener together with the two-halves panels along each direction. The primary effect at the global structure level follows from in-plane loads. For instance, top covers essentially endures tension in the longitudinal direction, while bottom covers (under the fuselage) essentially endures compression in the longitudinal direction. The central wing box redistributes different loads from the wing. The lift of the wing causes a bending moment at the central wing box turning into lateral tension/compression into panels and the swept wing causes twisting that turns into shear in the panels. In definitive most of the panels are under complex in-plane loading including shear that can be very high (for instance in the rear end). Moreover, composite aircraft are not only sized through strains but also with respect to impact considerations. We represented Fig. 1.4 the different loads all over the aircraft together with the specific regions where impact has to be considered. In terms of mechanical elements, the whole fuselage is considered as a whole assembly of thin shallow shells bonded with beams. Plane stress models (with however different membrane behavior for each panel since the membrane A tensor depends on the proportions of orientations) are then sufficient to give a relative accuracy of the global behavior of

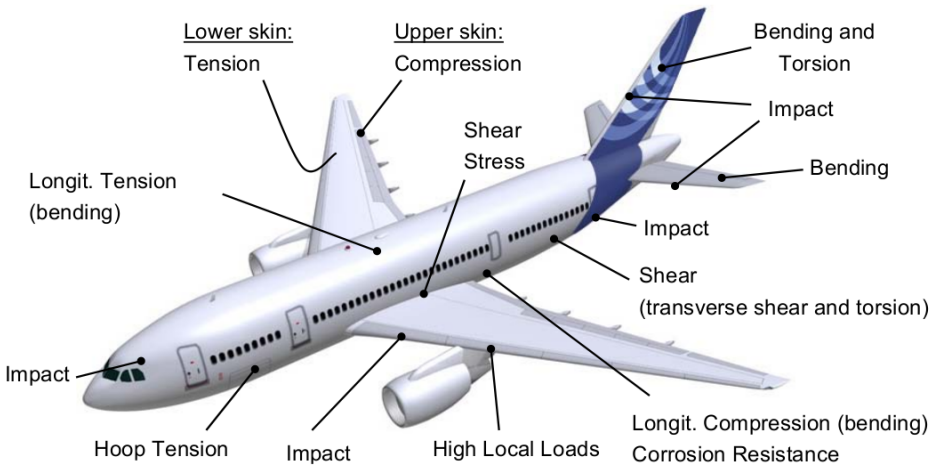


Figure 1.4: Main loads and impact regions over an Airbus aircraft

the structures. Since the flexural behavior is not considered at the full structural level, each of these sub-elements are represented as a finite mesh with few finite elements and the mechanical responses are usually very simple over each shell or beam. For instance in our applications strain and stress are considered constant over each panel or beams. The Finite Element Analysis (FEA) is often performed with classical softwares such as `MSC.Nastran`. This implies that the redistribution of internal loads is performed on the basis of only a few variables, that are denoted **stiffness variables**: these are for instance cross-sections areas, proportions of orientations for laminates. This would not be true if more advanced models were used at fuselage level. For instance, think of three-dimensional linear elasticity model, the precise description of the structure should be known to perform a FEA at global level, in particular, the detailed description of the stacking sequence should be known to predict bending behavior for instance.

1.3.2 Stability constraints

A typical instability phenomenon that appears in thin-walled structures such that fuselages is buckling. Buckling consists in a large deflection of a thin structure under compression in the transverse direction. Depending on the type of buckling, the structure might be very close to collapse (for instance Euler or global buckling of the whole superstiffener) or in the contrary it might be able to carry extra load even after buckling

(for instance for thin metallic or composite plates, even tough this capability is less pronounced for composite structures). Buckling is a typical non linear phenomenon that still fits in the category of small strains but not anymore in the category of small displacements. It is however often predicted (only the onset of buckling) through linear theory. In case of simple geometry for instance plates, buckling can be accurately predicted with the help of linearized von Karman equations and often approximated through spectral methods such as Rayleigh-Ritz methods. In case the behavior of the buckled structure has to be investigated, three-dimensional elasticity models are used leading to very expensive computations. In any case, the stability of the conceptual element super-stiffener is not only ensured through buckling computations but also through closed-form expressions, physical assumptions based on the how-do of the engineers and also experimental studies. Furthermore, the final output that the engineer checks to tell whether or not the structure will buckle is not the straightforward result of the physical computations. There are some rules-of-thumb or also certification that apply and change the output value on the basis of margin policies. This complicates a bit more the computations of stability criteria usually denoted **Reserve Factors**. These stability computations are often performed at Airbus with in-house tools (mostly based on spectral methods) with all the difficulties associated to such in-house: legacy, programming choices... that complicates the description of the output. Regarding the input of such computations, buckling behavior needs a more precise description of the geometry and the laminates (stacking sequences). Besides, the entry should feature also results of the internal loads redistribution. One could predict the buckling behavior of a thin plate with strains, stresses or displacement, however, it is often the forces (or loads per unit length) that are the entries of our stability computations. This means first that the FEA has to be performed before running the stability codes and second that the sensitivity computations is quite complicated since buckling constraints depends stiffness variables (and then on all variables) through these forces entries. To illustrate this, take the following example: say that x is the local variables of a given structures, Y the associated stiffness variable, we have $Y = f(x)$, denote Φ the in-plane mechanical response that we get from internal loads redistribution (e.g forces N_x for instance), then the buckling critical reserve factor depends on x and also on Φ

$$RF(x, \Phi) \tag{1.2}$$

to get the derivative of RF w.r.t x we need not only the partial derivative of the RF w.r.t x but also w.r.t Y

$$\frac{dRF}{dx} = \frac{\partial RF}{\partial x} + \frac{\partial \Phi}{\partial x}\frac{\partial RF}{\partial \Phi} \tag{1.3}$$

where the $\frac{\partial \Phi}{\partial x}$ terms comes from a Finite Element sensitivity analysis. Another difficulty associated to that sensitivity computation is that the FEA sensitivity only gives

$$\frac{\partial \Phi}{\partial Y} \tag{1.4}$$

hence, we need to chain sensitivities to get finally

$$\frac{dRF}{dx} = \frac{\partial RF}{\partial x} + \frac{\partial f}{\partial x}\frac{\partial \Phi}{\partial Y}\frac{\partial RF}{\partial \Phi} \tag{1.5}$$

Eq. (1.5) is easy to derive formally but as far as practical and realistic softwares are involved this chain rule is not straightforward to implement. In addition to that, the stability in-house tools do not give analytical sensitivities of RF and then finite differences are to be performed, leading to an excessive computational resources and also to numerically unstable overall derivative.

1.4 Formal description of structural optimization problem in the engineering context

When sizing a large aeronautic structure, we want to reduce the weight of structure as much as possible under hard stability constraints. This is formally described as an optimization problem with a huge number of variables (more than $10,000$) and a bigger number of constraints (more than $100,000$). Note that we will only consider parametric optimization of thin structures decomposed onto several structural elements.

1.4.1 Objective function

We consider a structure decomposed onto N elements. Each of these elements is described by several variables that can be geometric variables (skin thickness, step between frames...) but also material-dependent variables (stacking sequence for laminate composites); We denote $X^{(i)} = (x_1^{(i)}, \ldots, x_{n_i}^{(i)})$, the variables that describe the element i where n_i is the number of such variables. In all generality, there is no reason for n_i to be the same for all elements $i = 1 \ldots N$. we suppose that the weight of element i^i given

by a simple relation between dimensions X_i and we denote $M^{(i)} = m^{(i)}(x_1^{(i)}, \ldots, x_{n_i}^{(i)})$ or shorter $M^{(i)} = m^{(i)}(X^{(i)})$ the weight of element i. Our global objective is therefore denoted

$$M = \sum_{i=1}^{N} m^{(i)}(x_1^{(i)}, \ldots, x_{n_i}^{(i)}) \tag{1.6}$$

or shorter

$$M = \sum_{i=1}^{N} m^{(i)}(X^{(i)}) \tag{1.7}$$

The first important remark is that this objective function is additively separable, that is, the sum of functions independent from each other.

1.4.2 Constraints

As we said before, when sizing a large structure, we deal with a huge number of constraints. Difficulties of such problem do not come from the objective function but from the constraints. from a mechanical point of view this is quite clear : dimensions of each element change the global strength distribution and local constraints for each element (local buckling or Euler buckling) depends on global stress distribution. we distinguish different types

- global constraints on different nodes of the structure (maximal displacement or stress)
- local mechanical constraints in one element (local buckling of the stiffener's web, Euler buckling) that depends on the global stress distribution
- local dimensions constraints ($x_{min} < x < x_{max}$)
- neighborhood (or shared) constraints that connects adjacent elements. These are continuity design constraints ($x_1^{(i)} = x_1^{(i)}$).

There exist a lot of different constraints. These constraints are complicated since they connect global analyses (FEA usually based on `MSC.Nastran`) and local analyses (usually based on skill tools). For instance, for a fuselage structure made of super-stiffeners, stability constraints (local, Euler bucking, skin instability and compression or tension strains) are computed by means of complex skill tools that take as in input the local stress in the element that comes from the global analysis.

Constraints are therefore strongly dependent on the global stress distribution, this distribution is actually computed by means of an FEA. This FEA lies on a mesh. Each element of this mesh is actually a structural element and so global stiffness (displacements d, strain ε, stress σ, internal strength F) are constant on each structural element. Global stress distribution is not directly computed with detailed local dimensions but through so-called stiffness variables. Namely, each element is described by a few stiffness terms (denoted by $Y^{(i)}$), which are for instance the cross-section area, moment of inertia... They obviously depend on the detailed dimensions (we denote this $Y^{(i)} = \varphi(X^{(i)})$) but they are enough to compute the stress distribution. We can therefore write the different constraints with the following convention that if the constraints computations require the stress distribution and therefore an FEA, we make a K appear in the notation, e.g c^K (to recall that the constraints computations require the assembly and the resolution of stiffness system $Kd = f$)

- $c^K_{glob}(Y^{(1)}, \dots, Y^{(N)})$ are the maximal displacement constraints
- $c^{K(i)}_{glob-loc}(Y^{(1)}, \dots, Y^{(N)}, X^{(i)})$ are local constraints which depend on stress distribution : these are typically *Reserve Factors* computations
- $c_{glob}(X^{(1)}, \dots, X^{(N)})$ are shared constraints between elements : design continuity constraints
- $c^{(i)}_{loc}(X^{(i)})$ are purely local sizing constraints, e.g *box* constraints $x_{min} \leq x^{(i)}_j \leq x_{max}$.

This stiffness terms $Y^{(i)}$ are of great interest because they reduce the high number of variables for each element to a low number of variables which are, in definitive, the only ones we need to compute the global stress distribution. If stiffness terms are fixed, the stress distribution is fixed and therefore the structural elements are decoupled which allows to perform local optimizations concurrently. This allows also to distinguish between hard constraints (c^K) of high computational cost usually nonlinear and non convex and easy constraints usually linear (or inverse linear) and not so expensive to call and differentiate.

1.4.3 Formal problem

Before giving the formal definition of the structural optimization problem, we must remark that the weight of element i $m^{(i)}$ can be computed directly from the sole stiffness terms. For instance, the cross-section area A_i of element i is enough to compute the weight, this is why we indifferently denote it $m^{(i)}(X^{(i)})$ or $\hat{m}^{(i)}(Y^{(i)})$ and $m(X)$ or $\hat{m}(Y)$ the global weight (with $X = (X^{(i)}, \dots, X^{(N)})$ et $Y = (Y^{(i)}, \dots, Y^{(N)})$. The only important characteristic is to remain the separability of the function. On the other hand, constraints $c^{K(i)}_{glob-loc}$ do not directly imply the stiffness terms Y but rather the stress **in**the element i computed with the stiffness terms **of** of all the elements. Therefore the constraint $c^{K(i)}_{glob-loc}$ for element i depends on all the element even though it only concerns the element i. This is why we prefer to denote $\Phi^{(i)}(Y)$ the stress in element i used to compute the RF of element i and the constraints $c^{K(i)}_{glob-loc}(\Phi^{(i)}(Y), X^{(i)})$. Eventually, the minimization problem can be written as

$$\boxed{\begin{array}{ll} \min_X & M(X) = \sum_{i=1}^{N} m^{(i)}(X^{(i)}) \\ \text{s.t} & \begin{cases} c^{K}_{glob}(Y^{(1)}, \dots, Y^{(N)}) \leq 0 \\ c^{K(i)}_{glob-loc}(\Phi^{(i)}(Y), X^{(i)}) \leq 0 \\ c_{glob}(X^{(1)}, \dots, X^{(N)}) \leq 0 \\ c^{(i)}_{loc}(X^{(i)}) \leq 0 \end{cases} \end{array}} \tag{1.8}$$

1.5 Objectives and motivations of this thesis

We can now turn to the main objective of this thesis, which is to find out efficient ways to solve such problems. This resolution is to be find through different axes of research and innovation

- the very nature of Problem (1.8) advocates for **breaking up the large problem into many smaller problems** that are to be solved in parallel. Indeed once the Y's are fixed, we end up N smaller sub-problems can be solved in parallel.

- find out an **appropriate representation of the composite** that can be used in a continuous manner to be integrated in the optimization. Indeed, the very large size of the optimization problems, the continuous internal loads redistribution also advocates for gradient-based optimizations. First because heuristics such genetic algorithms or simulated annealing or all of the specific methods Mixed Integer

Non Linear Programming could hardly handle such a large number of variables and constraints

- **speed up the whole optimization by using surrogate models** that should ideally be very accurate for the global-local constraints. Indeed, these constraints are the most computationally demanding constraints because of the number of calls in a realistic optimization (several millions times) and of the sensitivity issue.

1.5.1 Early state of the art

This thesis is totally is the continuation of the work of A. Merval ((Merval 2008), (Merval *et al.*)) who highlighted the needs for decomposition and surrogate models in the frame of the sizing of metal structures. In particular, A. Merval shown that we could get more accurate results if we divided the approximation space. However this division was handmade and relied on mechanical criteria. Regarding decomposition, A. Merval initiated thorough comparisons through its original test case that we used in this thesis. However, results could not distinguish a given bilevel/decomposition among the others since none of them led to the real optimum in numerical investigations. For composite structures, this thesis is also in the continuation of the work of B. Liu. In (Liu 2001), (Liu *et al.* 2004) the author highlighted and derived the existing scheme **Quasi separable Subsystem Decomposition** for composite structures. The main optimization problem was close to the one treated here: bilevel optimization of a composite wing. In particular, the need for accurate surrogate models in high-dimensions was highlighted to speed up bilevel optimization and a decomposition based on proportions and real discrete optimizations at panel level is given. These references were first answers to the problem of the optimization of large scale composite structures.

Owing to the different subjects this thesis intends to cover (decomposition, surrogate models and composite optimization), a full state-of-the-art would hardly be useful if presented at once. This is why a more detailed state-of-the-art is presented in each related chapter.

1.5.2 Decomposition/bilevel

Decomposition or bilevel formulations are used here in a synonymous meaning, while they are not totally. As a matter of fact our aim is to derive and validate a decomposition formulation that gives rise to a bilevel optimization scheme. To that extent, **decomposition** then means breaking up the whole problem into many smaller problems. This decomposition should be done in such a way that one group of variables have major influences on the overall structure and another group should have only influence on the local design. Leading to an upper level optimization based only on a few variables that include as constraints local optimization performed at lower level. Typical choices for that group of upper level variables is of course the Y's stiffness variables, while the lower level group would obviously be the X's. Due to the nature of the problem (objective that is additively separable), the lower level optimizations over X could be performed only at sub-element i level, leading then to N sub-problems that are solved concurrently.

At this point, we should mention that the typical decomposition/bilevel scheme that we address in this thesis are very similar in spirit to **Multi Disciplinary Optimization** (MDO) schemes. Indeed, our aim is to break up the large problem into into many problems that matches the different levels of analysis. That is to say, an upper level where internal loads redistribution is considered and lower level where only stability constraints are considered. This lower level of course should also use the natural and structural decomposition of the overall structure and should be done at structural sub-element level (typically super-stiffener level). One could imagine other decompositions (stiffener level in one hand and panels in the other hand). However, no matter what is the decomposition used, structural sub-elements can be thought as systems connected between them and this interaction should be considered during the design process. This is precisely the definition of MDO problems, for instance in (Sobieszczanski-Sobieski & Haftka 1997). The main difference here is that we do not consider different disciplines with different objectives rather different elements with the same local objective (weight), but the overall goal remain the same that is decomposing the whole optimization problem on the basis on its natural definition in terms of interconnected systems (or sub-elements). We refer to the specific term of multilevel

because our analysis of the whole structure implies different levels or representation: in-plane properties based on a rough description of the structure (in MDO jargon, stiffness variables are public variables, internal loads are coupling variables), out-of-plane properties based on a more realistic description of the structure. We could think of deeper level again more realistic (ply drop-offs for instance). Even though multilevel seems a more appropriate term to our problem, our challenging tasks remain essentially the same as in MDO: computational and organizational issues.

The current sizing process for very large structures relies essentially on classical mechanical optimality criteria that are known in some cases (hyperstatic, redundant structures or structures made of different material) that are known to be sub-optimal (though the optimum is often near the real optimum, see (Haftka & Gurdal 1992)). Our objective is then to find out a decomposition scheme that improves the optimum and possibly finds the real optimum.

1.5.3 Adaptation to specific composite considerations

Such a decomposition should be adapted to composite laminate structures. In particular, we have a find a way to treat the different levels of analysis for composites. Proportions of fiber orientations are a convenient way to treat the internal loads redistribution since under plane stress assumptions and for symmetric laminates, the internal load redistribution depend only on these variables. However, proportions are not enough to describe the bending behavior (buckling...) and usually discrete optimization algorithms are used to find out realistic stacking sequences (genetic algorithms...). In our case for efficiency reasons, we want to use gradient-based optimizations algorithms and then need a continuous relaxed representation of the bending behavior. This representation should also ideally be adapted to our decomposition scheme and well-suited for local optimizations problems (convexity properties for instance) and finally this representation should also be convenient to design approximation models of buckling constraints.

1.5.4 Approximations of stability constraints

Besides decomposition, buckling constraints are called too many times to get an efficient resolution of large scale optimization problems. The other way of speeding up the whole

optimization is by using **approximation models or surrogate models**. the very nature of the buckling constraints (non linear, piecewise discontinuous) and Airbus stability tools make this approximation no very good when using classical surrogate models (polynomial regression, neural networks,...). The high dimension is also quite challenging. Our objective is then **to improve the accuracy of the approximation by taking into account specific knowledge of the functions** to be approximated. This obviously implies the investigation of the nature of such functions.

1.6 Detailed outline

We can now give a detailed outline of the thesis.

1.6.1 Optimization properties and early presentation of the decomposition/bilevel scheme

We start with the main core of our thesis which is essentially concerned with large-scale optimization. In Chapter 1, a few fundamentals facts on theoretical optimization are recalled ranging from basics optimality conditions to the less classical subject of sensitivity of optimization with respect to the parameters. Roughly speaking, in the multilevel or MDO framework we often consider optimization within optimization, and whenever gradient-based methods are used, we need to assess the derivation of the inner loop optimization with respect to its parameters. This post-optimal sensitivities are to be used when implementing the decomposition/bilevel schemes. Moreover, eigen-problem sensitivities (such as buckling) can be thought as post-optimal sensitivities, we then give this useful and original interpretation. The original monolevel problem is then shown to be equivalent to an optimization that uses directly stiffness variables as optimization variables. This equivalence is the backbone of our decomposition/bilevel schemes since they can be derived from decomposing the different equivalent monolevel versions of our original problem.

1.6.2 Adaptation to specific composite representation

Before testing and validating we need first to find out a representation of the composite and more precisely of the bending behavior that can be used within a decomposition scheme and also that can be used to build approximation models. These two aspects

are investigated in Chapter 3 by introducing a very classical and well known representation of laminates: the lamination parameters. We review some of their properties and derive a quite rigorous proof of the concavity of the buckling critical load factor over lamination parameters. Buckling critical load factor or **Reserve Factor** is a typical eigenproblem parameterized by lamination parameters, loading... Based on the eigenproblem sensitivities presented in the former chapter and on the Rayleigh-Ritz method, we investigate the behavior of the Reserve Factor and show that it behaves essentially as a piecewise differentiable function and relates this behavior to theoretical references. Lamination parameters also offers a compact representation, few variables can be used as predictive variables. An ideal approximation strategy for the buckling constraints would be first to use lamination parameters and take into account the numerically observed behavior. Furthermore, when used in the more complex and realistic stability tools, buckling computations can also lead to real discontinuities.

1.6.3 Approximating discontinuous and derivative discontinuous functions

A whole new approximation strategy is developed in Chapter 4. This is aimed to handle the specific behavior of buckling constraints. The main and quite simple idea is to decompose the input space into many regions over each the function to approximate is simple (at least continuous). This is achieved by using tools from unsupervised statistical learning such as clustering and density estimation. Our original strategy relies on the Expectation-Maximization algorithm for Gaussian mixtures. We present our algorithm on the basis of probability/statistics framework. Many test cases are presented to evaluate our strategy that was coined by the author IMAGE (**Improved Metamodeling Approximation through Gaussian mixture of Experts**). This strategy lies in the field of mixture of experts. Several types of experts used throughout this thesis are then briefly reviewed an an original interpretation of a standard approximation model (weighted least squares) is given in terms of mixture of experts giving more insight to improvement methods.

1.6.4 Numerical investigation of monolevel and bilevel schemes

The main tools of our bilevel schemes are at this point already presented, we then turn to the investigation and comparison of the different schemes. However, before comparing bilevel schemes, we need to set up first a test case, inherited from (Merval 2008).

We also recall a few facts on sensitivity analysis from problems involving FEA's. This material is presented and applied in Chapter 5 where we also investigate the numerical behavior of the different monolevel schemes.

Chapter 6 is concerned with the numerical comparison of the different bilevel schemes already presented on the basis of our realistic test case. The important difference in terms of implementation are highlighted. Indeed, such bilevel formulation can be implemented in very various manners: either we chain optimizations, first a top level optimization that finishes and then we send the results to the lower level optimizations and then iterate back and forth, leading to a lot of optimizations, or we implement it as a real bilevel optimization, only one upper level optimization is performed, where some constraints involves solving at each iteration many local optimizations problems. The alternating optimizations version is shown to be sub-optimal and we then turn to a real bilevel implementation of the **Quasi-separable Subsystem Decomposition** scheme. This QSD is shown to be optimal in our test case and is also shown to require slightly less FEA's and global iterations than the AIO version.

On the basis of this results, we turn to a realistic composite structure in Chapter 7, where we implement and apply the QSD scheme on the basis of the innovative material we presented in the former chapters. First lamination parameters are used at both levels, we then adapt the QSD scheme to lamination parameters and the stability constraints computed through an Airbus in-house tool are approximated through the IMAGE strategy extended to mixed variables. Finally post-optimal sensitivities are chained with internal loads sensitivities. Detail of the implementation and results are given in that final chapter.

2 Monolevel and bilevel formulation

This chapter is concerned with the theoretical properties of the different formulations that we apply in the sequel. Indeed, our large-scale optimization problem exhibits a specific structure that is used to derive several classical decomposition schemes. This chapter is also essentially concerned with the theoretical properties of optimization: the basics of differentiable optimization are recalled, an important insight is given into post-optimal sensitivities, that is the sensitivity of the optimum value function with respect to problem parameters. Indeed, this derivative is a cornerstone of our bilevel implementation. An original application of these classical results is given when deriving the sensitivity of an eigenvalue problem. The original monolevel formulation is then shown to be mathematically equivalent to the block-angular form. This block-angular form is then transformed through inexact quadratic penalty, resulting in three different monolevel instances of the same optimization problem, each of them leading to different bilevel decomposition schemes. The different implementations will be presented and validated on Chapter 6.

2.1 Motivation for decomposition techniques

The aim of **decomposition techniques** (a.k.a **multilevel optimization**) is twofold. First it allows to treat problem of size out-of-scope due to memory limitations by **breaking up a large problem into many smaller sub-problems**. As outlined in the first chapter, the size of typical structural optimization for fuselage structures is about several thousands optimization variables and much more optimization constraints, making the practical optimization hard to handle in one shot. Then the original problem is to be decomposed into many smaller problems that can be solved in one-shot due to their reduced size. The second important practical aspect that the decomposition schemes want to reach is **scalable optimization**, the smaller sub-problems should be solvable in parallel hence they should not depend on each other at sub-problem level. Their

connection is to be solved at upper level by means of constraints. In this short section we give more details on these two aspects to show and understand the objective of the decomposition techniques before presenting them at the end of the chapter.

2.1.1 Large-scale problems with natural hierarchy

Large-scale problems arising in structural optimization often exhibits a **specific structure**. This structure is derived from the very nature of the structure to be optimized. The global structure is made of many sub-elements all connected between them by internal loads redistribution. This implies also differences of level of analysis, the internal loads redistribution is made at global level on a rough and coarse mesh of the fuselage, e.g panels are represented by a few finite elements and typical responses such as strains ε, stresses σ and forces (loads per unit length) $N_x, ...$ are considered constant at sub-element level. Furthermore only a few variables are needed to compute this redistribution through a Finite Element Analysis. In general, internal loads redistribution computations is based on simplified elasticity models and do not consider 3 dimensional linear elasticity equations. Hence, a typical FEA over the fuselage involves only stiffness variables such cross-section areas, proportions of layers orientations for laminates. At sub-element level the mechanical analysis is much more detailed and is based on more advanced modeling such as large deflections of thin plates (linear von Karman equations for instance) for buckling, post-buckling behavior for thin plates,...The number of design variables is larger (detailed dimensions of the stiffener, stacking sequence for laminates) and the analysis can be performed on different approximation methods (Rayleigh-Ritz for buckling, see appendix), classical closed-form expressions (for damage tolerance, post-buckling), engineer formulations based on the art and experience...and in case more details are to be known a non linear finite elements analysis can be performed. One of the most important feature of these analyses is that they usually take as inputs responses computed from the internal loads redistribution, making these constraints (typically buckling) not only depending of local sub-elements variables but also on all coarse variables (or stiffness variables), these constraints are then referred to global-local constraints and induce a natural hierarchy, analysis start with FEA, based on its results, stability analysis at local level is performed. These different levels of mechanical analysis were already outlined in the first chapter. They induce a natural hierarchy on the analysis that translates into the optimization problem. Furthermore,

we understand from a mechanical point of view that such global-local constraints essentially depends on the detailed dimensions but also on the coarse stiffness variables, making the constraints Jacobian block diagonal dominating. Typical such block diagonal dominating form is depicted Fig. 2.1. This structure is not the most appropriate to be decomposed, indeed, roughly speaking 'everyone depends on everyone', it does not make appear any specific group of variables that could be used as upper level optimization variables. This initial formulation will be however our reference problem and will be referred to as **All-in-One Optimization**. This is why the first step that we achieve in this chapter is to find equivalent formulations that exhibit a more valuable structure for decomposition. The form that we will derive in the next sections is known in the literature as **block-angular structure** and will introduce and use stiffness variables as design variables. Besides decomposition, the block-angular structure has benefits in terms of sensitivity computations. Indeed the profile of the constraint Jacobian is much more simple to handle and has many more zero elements. See for instance Chapter 5 where we derive analytical sensitivities for the AIO problem. Other types of optimization problem structures can be found in (Haftka & Gurdal 1992), (De Wit 2009) and (de Wit & van Keulen 2011). Last but not least, we will see that block-angular decomposition allows to fit the different levels of mechanical analysis, since in our approach only the stiffness variables appear at the upper level and thus finite element analysis is to be performed only at this level while only detailed variables appears at sub-element level and only local stability analysis is performed while keeping constant responses from the FEA performed at upper level.

2.1.2 Towards scalable optimization

The other immediate benefit of decomposition is that the smaller sub-problems can be solved in parallel, allowing to speed up design. Indeed, we will see that, in the block-angular transform, once the stiffness variables are fixes, the sub-problems are no more connected among them and can be solved in parallel. Optimization at upper level uses quantities computed through the resolution of the lower level optimizations as constraints, making then one iteration of the upper level computationally demanding, however, these constraints can be computed in parallel, as well as their sensitivities. Hence, provided there are only a few iterations at global level and that sub-element optimization are fast (for instance a well-posed convex problem solved by quasi-Newton

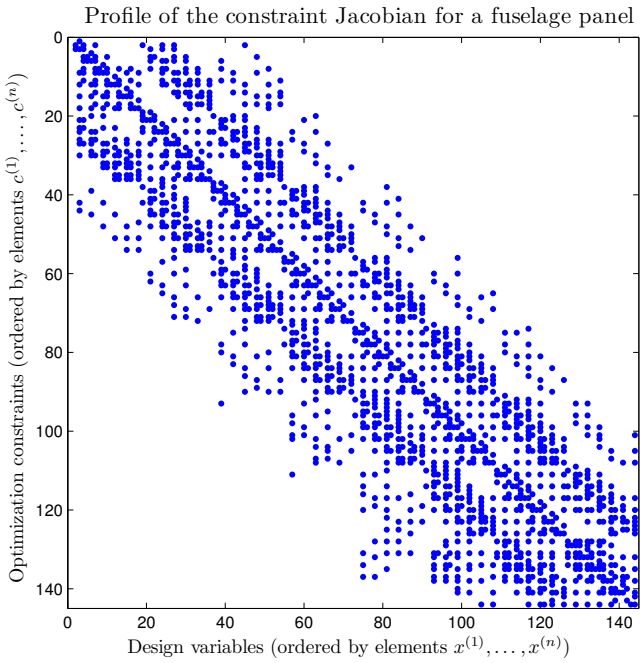


Figure 2.1: Profile of the dominating elements in absolute value of the Jacobian of constraints. Note that this profile is obtained after ordering variables and constraints, we sort the variables sub-elements per sub-elements and constraints the same way. In reality all the elements of this matrix are non zero, we only depicted the largest elements

methods (SQP, GCM,...) using a smart initial point), the global process might be faster than an all-in-one optimization.

2.2 Basics of differentiable optimization theory

We briefly recall in this section a few general facts on differentiable optimization theory that will be used throughout this chapter and in next chapter as well. The optimality conditions for constrained problems are recalled together with a few remarks on constraints qualification that are a pitfall in bilevel programming. Indeed, bilevel programming involves 'optimization within optimization', creating then implicit functions that are computed through the resolution of an optimization problem. Such functions are usually not differentiable whenever the implicit optimization problem has many active constraints at the optimum and the local solution is at a point where any modification of the optimization parameters will make the set of active constraints change. In case, such a function is differentiable, we recall here main results on its derivatives that are often called **post-optimal sensitivities**. The first part is very classical and does not present any innovation, emphasis is put on post-optimal sensitivity results. An original application of this results is presented: derivative of an eigenvalue problem.

2.2.1 Vocabulary

Let f be a function defined over some subset K of $\mathbb{R}^n$. K will be the feasible set of f, for the moment we do not consider explicit constraints. We just think of K as the set where some constraints are satisfied. We are interested in the following problem

$$\min_{x \in K} f(x) \tag{2.1}$$

$x^\star \in K$ is a global solution of Problem (2.1) whenever

$$\forall x \in K \quad f(x^\star) \leqslant f(x) \tag{2.2}$$

$x^\star \in K$ is a local solution of Problem (2.1) whenever $x^\star$ minimizes f in a neighborhood of $x^\star$. More formally, there exists $\varepsilon > 0$ s.t

$$\forall x \in B_\varepsilon(x^\star) \quad f(x^\star) \leqslant f(x) \tag{2.3}$$

where $B_\varepsilon(x^\star) = \{x \in K \ ||x - x^\star|| < \varepsilon\}$ is the open ball of radius ε. A global solution (resp. a local solution) is called a strict global solution (resp. a strict local solution) provided the inequality in (2.2) (resp. (2.3)) is strict. Problem (2.1) has a global solution whenever f is continuous and K is compact. In case K is closed but not compact (hence not bounded), the existence of a global solution is ensured whenever f tends to infinity at infinity on X, i.e

$$\lim_{x \in X, ||x|| \to +\infty} f(x) = +\infty \tag{2.4}$$

Remark. Note that in the context of parametric structural optimization, we often show that a minimum exists by using this result. See the chapter on parametric optimization in (Allaire 2007). See also the appendix where we show with that basic argument that a global minimum exists for our test-case.

A set K is convex provided for any x and y in K, $\theta x + (1-\theta)y$ also belongs to K for all $t \in (0,1)$. A function f defined over a convex K is convex provided for any x and y in K we have

$$\forall \theta \in (0,1) \quad f(\theta x + (1-\theta)y) \leqslant \theta f(x) + (1-\theta) f(y) \tag{2.5}$$

Convexity is of primary importance in optimization since it provides important features about an optimization problem. For instance, for Problem (2.1) with K convex, in case f is also convex, we have

- If $x^\star$ **is a local solution of (2.1) then it is a global solution.**
- The set of all global solutions of (2.1) is convex.
- In case, f is strictly convex, (2.1) has at most one solution.

It should be noted that convexity is not by itself sufficient to ensure the existence of a solution (think of $x \mapsto \frac{1}{x}$ over $\mathbb{R}_+^\star$) with no further assumption on K. In practical algorithms, convexity plays also a very important role, since the following optimality conditions that are necessary become sufficient in case of convexity.

2.2.2 Optimality conditions

We will assume that f is $\mathscr{C}^2$ over K. We first derive the optimality condition for an unconstrained problem $K = \mathbb{R}^n$.

$$\min_{x \in \mathbb{R}^n} f(x) \tag{2.6}$$

We summarize here the necessary and sufficient conditions for a point $x^\star$ to be a local solution of (2.6)

- Necessary conditions
  - **First order necessary condition**: the gradient of f at $x^\star$ is zero

$$\nabla f(x^\star) = 0 \tag{2.7}$$

  - **Second order necessary condition**: the Hessian of f at $x^\star$ is positive semi-definite

$$\forall d \in \mathbb{R}^n \quad d^T \nabla^2 f(x^\star) d \geqslant 0 \tag{2.8}$$

- Sufficient conditions: if $x^\star$ is s.t $\nabla f(x^\star) = 0$ and $\nabla^2 f(x^\star)$ is positive definite, then $x^\star$ is a local minimum of (2.6).

Practically, it means that the first-order condition is not enough to say whether or not we are at a local minimum of f. Indeed, a local maximum also satisfies the first-order necessary condition (with the Hessian negative semi-definite) and there may be points satisfying the first-order necessary condition (such points are often called **critical** or **stationary**) that are neither local minimum nor local maximum (**saddle points**, at these points the Hessian has both positive and negative eigenvalues). The two necessary conditions are often denoted **zero-slope** (vanishing gradient) and **non-negative curvature** (all the directions are ascent directions) conditions. Let f be a convex function, then its Hessian is non-negative at all $x \in \mathbb{R}^n$. We then see that the first-order necessary condition is also a sufficient condition for global minimum.

We now turn to optimality conditions for constrained problems of the sort

$$\begin{array}{ll} \min_{x \in \mathbb{R}^n} & f(x) \\ \text{s.t} & \left\{ \begin{array}{l} g(x) \leqslant 0 \\ h(x) = 0 \end{array} \right. \end{array} \tag{2.9}$$

where the inequality constraints $g : \mathbb{R}^n \mapsto \mathbb{R}^{m_I}$ and the equality constraints $h : \mathbb{R}^n \mapsto \mathbb{R}^{m_E}$ will be assumed to be $\mathscr{C}^2$. Recall that the Jacobian $Dg(x)$ of g at x is the $m_I \times n$ matrix

$$Dg(x) = \begin{pmatrix} \frac{\partial g_1}{\partial x_1} & \cdots & \frac{\partial g_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial g_{m_I}}{\partial x_1} & \cdots & \frac{\partial g_{m_I}}{\partial x_n} \end{pmatrix} \tag{2.10}$$

hence in gradient notation (vector of partial derivatives)

$$Dg(x) = \begin{pmatrix} \nabla g_1(x)^T \\ \vdots \\ \nabla g_{m_I}(x)^T \end{pmatrix} \tag{2.11}$$

This allows us to define the gradient of g at x with $\nabla g(x) = Dg(x)^T$.

Note $m = m_E + m_I$, $E = 1 \dots m_E$, and $I = m_E + 1 \dots m$ (indexes of constraints) and consider $c : \mathbb{R}^n \mapsto \mathbb{R}^m$ defined as $c(x) = (h(x), g(x))$, we will make use of the Jacobian of c that we will denote $A(x) = \nabla c(x)^T$ of size (m, n). So $A_{ij} = \frac{\partial c_i}{\partial x_j}$.

The necessary and sufficient conditions for a constrained problem of type (2.9) make appear natural quantities associated to the constraints called the **Lagrange multipliers** (or **Karush-Kuhn-Tucker multipliers**) traditionally noted λ. They also make naturally appear the **Lagrangian** function $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^m$ defined as

$$\mathcal{L}(x, \lambda) = f(x) + \lambda^T c(x) \tag{2.12}$$

with $\lambda = (\lambda_I, \lambda_E) \in \mathbb{R}^{m_I + m_E}$. We can now state the first-order necessary conditions for a point $x^\star$ to be a local minimum of Problem (2.9) provided $x^\star$ satisfies some regularity condition for the constraints (**constraint qualification**) that we will develop briefly in the next section.

There exists $\lambda^\star \in \mathbb{R}^m$ s.t

- **Optimality condition** $(x^\star, \lambda^\star)$ is a stationary point of the Lagrangian w.r.t x.

$$\nabla f(x^\star) + A(x^\star)^T \lambda^\star = 0 \tag{2.13}$$

- **Feasibility** Constraints are satisfied at $x^\star$

$$h(x^\star) = 0, g(x^\star) \leqslant 0 \tag{2.14}$$

- **Complementarity slackness conditions** The Lagrange multipliers $\lambda_I^\star$ associated to inequality constraints satisfy the following relations

$$\lambda_I^\star \geqslant 0 \tag{2.15}$$

and

$$\lambda_I^{\star T} g(x^\star) = 0 \tag{2.16}$$

The set of Equations (2.13, 2.14, 2.15, 2.16) are the **Karush-Kuhn-Tucker conditions**. Note that (2.13) is indeed a stationary point of $\mathcal{L}$ w.r.t to x

$$\nabla_x \mathcal{L}(x, \lambda) = \nabla f(x) + \nabla c \lambda \tag{2.17}$$

and (2.16) means that

- the multipliers corresponding to inactive constraints are zero

$$g_i(x) < 0 \implies \lambda_i^\star = 0 \tag{2.18}$$

- The multipliers associated to inequality constraints are always nonnegative, this will appear more clear when we will derive the interpretation of Lagrange multipliers as the derivative of the objective function at the optimum w.r.t optimization parameters.

We now turn to the regularity conditions of constraints that need to be satisfied at a point $x^\star$ to be a local minimum.

2.2.3 Constraint qualifications

There exist many different constraint qualification, some of them being quite sophisticated. The purpose of that short section is to give only the most classical constraint qualification. First denote $I_0(x^\star) = \{i \in I \quad \text{s.t} \quad c_i(x^\star) = 0\}$, the set of active (inequality) constraints at $x^\star$. Note that, $\lambda_i^\star = 0$ does not necessarily implies that $c_i(x^\star) < 0$. However, in case we have $\lambda_i^\star = 0 \implies g_i(x) < 0$, this is referred to as

- **Strict Complementary Slackness Condition** (**SCSC**) whenever we have

$$\lambda_i^\star = 0 \iff g_i(x) < 0 \quad \forall i \in I_0(x^\star) \tag{2.19}$$

Define the Linear Independence Constraint Qualification (LICQ) at x with

- (**LICQ**) The gradients of the active constraints $\{\nabla c_i(x) \quad \text{s.t} \quad i \in E \cup I^0(x)\}$ are linearly independent

$$\sum_{i \in E \cup I^0(x)} \alpha_i \nabla c_i(x) = 0 \implies \alpha_i = 0 \quad \forall i \in E \cup I^0(x) \tag{2.20}$$

LICQ ensures that there exists at most one Lagrange multiplier $\lambda^\star$ satisfying the first-order optimality conditions. If we have $\lambda^\star$ and $\mu^\star$ satisfying (2.13) at the same $x^\star$, LICQ implies that $\lambda^\star = \mu^\star$.

The following constraint qualification known as Mangasarian-Fromovitz, (MFCQ) is weaker than LICQ and hence is more often satisfied. We say that the constraints c_i at $x^\star$ are MFCQ qualified provided

- (**MFCQ**) The gradients of the active constraints $\{\nabla c_i(x) \quad \text{s.t} \quad i \in E \cup I^0(x)\}$ are positively linearly independent

$$\text{if} \sum_{i \in E \cup I^0(x)} \alpha_i \nabla c_i(x) = 0 \quad \text{with} \quad \alpha_i \geqslant 0 \quad \text{for} \quad i \in I^0(x) \tag{2.21}$$

then

$$\alpha_i = 0 \quad \forall i \in E \cup I^0(x) \tag{2.22}$$

2.2.4 Second order conditions for constrained problems

In this section, we assume that f, g and h are twice continuously differentiable w.r.t x. In that case, we have the following necessary conditions. As in the unconstrained case, there exist sufficient conditions, quite close, that ensure practically that we did find a local minimum

- (**Necessary condition of second order for constrained problems (NC2CO)**) Let $x^\star$ be a local minimum of Problem (2.9) then
  (a) Conditions (2.13), (2.14), (2.15) and (2.16) hold.

(b) The Hessian of the Lagrangian is positive semi-definite

$$z^T \nabla^2_{xx} \mathcal{L}(x^\star, \lambda^\star) z \geqslant 0 \tag{2.23}$$

for all $z \neq 0$ in $\mathbb{R}^n$ s.t

$$\nabla g_i(x^\star) z \leqslant 0 \quad \forall i \in I^0(x^\star) \tag{2.24}$$

$$\nabla g_i(x^\star) z = 0 \quad \forall i \in I \quad \text{s.t} \quad (\lambda^\star_I)_i > 0 \tag{2.25}$$

$$\nabla h_i(x^\star) z = 0 \quad \forall i \in E \tag{2.26}$$

- (**Sufficient condition of second order for constrained problems (SC2CO)**)
  Let $x^\star$ be a point of $\mathbb{R}^n$, if

  (i) There exist $\lambda^\star_I$ and $\lambda^\star_E$ s.t conditions (2.13), (2.14), (2.15) and (2.16) hold.

  (ii) We have

$$z^T \nabla^2_{xx} \mathcal{L}(x^\star, \lambda^\star) z > 0 \tag{2.27}$$

for all $z \neq 0$ in $\mathbb{R}^n$ s.t

$$\nabla g_i(x^\star) z \leqslant 0 \quad \forall i \in I^0(x^\star) \tag{2.28}$$

$$\nabla g_i(x^\star) z = 0 \quad \forall i \in I \quad \text{s.t} \quad (\lambda^\star_I)_i > 0 \tag{2.29}$$

$$\nabla h_i(x^\star) z = 0 \quad \forall i \in E \tag{2.30}$$

Then $x^\star$ is a **strict local minimum of Problem (2.9)**.

2.2.5 Post-optimal sensitivity

Post-optimal sensitivity refers to the area of estimating the sensitivity of a problem of optimization w.r.t problem parameters. The typical formulation of such a parameterized problem is

$$\begin{array}{ll} \min_{x \in \mathbb{R}^n} & f(x;p) \\ \text{s.t} & \begin{cases} g(x;p) \leqslant 0 \\ h(x;p) = 0 \end{cases} \end{array} \tag{2.31}$$

where p is some fixed parameter that remains unchanged during the optimization. We formally define the **optimal value function** $\mu^\star : \mathbb{R} \mapsto \mathbb{R}$ as

$$\begin{array}{lll} \mu^\star(p) := & \min_{x \in \mathbb{R}^n} & f(x;p) \\ & \text{s.t} & \begin{cases} g(x;p) \leqslant 0 \\ h(x;p) = 0 \end{cases} \end{array} \tag{2.32}$$

The same way we can define not a function but 'multivalued' function that associates to p the set of minimizers of Problem (2.33). This is often called in the literature (see (Fiacco 1983)) the **solution set mapping** $x^\star : \mathbb{R} \mapsto 2^{\mathbb{R}^n}$ with

$$x^\star(p) := \begin{array}{ll} \text{argmin}_{x\in\mathbb{R}^n} & f(x;p) \\ \text{s.t} & \begin{cases} g(x;p) \leqslant 0 \\ h(x;p) = 0 \end{cases} \end{array} \tag{2.33}$$

where $2^{\mathbb{R}^n}$ is the power set of $\mathbb{R}^n$ the set of all subsets of $\mathbb{R}^n$. In case Problem (2.33) has a unique solution, for instance in case of strict convexity of f, ellipticity of f and g convex, $x^\star$ turns out to be a **vector-valued function** and no more a point-to-set mapping. In case Problem (2.33) is a convex parametric optimization problem for all p, $x^\star(p)$ is a convex and closed set of $\mathbb{R}^n$ but possibly empty (recall that convexity does not ensure the existence of a minimum). Similarly to the **solution set mapping** we can define **Lagrange multipliers set mapping** $\lambda^\star$ that associate the Lagrange multipliers found at optimums. In the sequel, we will be interested in deriving properties for the optimal value function $\mu^\star$.

This is related to the estimation of the behavior of the optimal solution w.r.t an optimization parameter. To make it more clear, we can think of a structural optimization problem, for instance the weight of a structure made of an isotropic homogeneous material of material characteristics (E, ν) is minimized under stability constraint, suppose an optimum is found. A question that naturally arises is what would happen if we do not know very precisely E and ν or if we want to replace the material by a 'close' material E' and ν'. **What would be the new optimum? Would that be close to the original optimum?** To make it more precise, suppose we do not know E at $\delta E > 0$ and we could derive the derivative of $\mu^\star(E)$ w.r.t E, $\mu^\star$ being some optimal value function (e.g weight...) then the real optimum lies in (in case $\frac{\partial \mu^\star}{\partial E} > 0$)

$$[\mu^\star - \delta E \frac{\partial \mu^\star}{\partial E}, \mu^\star + \delta E \frac{\partial \mu^\star}{\partial E}] \tag{2.34}$$

Post-optimal sensitivity tries to answer these questions by giving **Lagrange multipliers a fundamental role that allows to get the derivative of the optimal value function**. Note that it is an old subject for it was studied in the early 40's (Hotelling, Roy, Samuelson...see (Löfgren 2011)) and the main results were already know from the economists (**shadow price** or **marginal utility**, **envelope theorem**) and inspired

the theoretical work of Bergé (Berge & Patterson 1963) (**maximum theorem**)[1]. However, no general rigorous results for nonlinear programming were derived until the 70's. It traces back to the work of Fiacco, Mc Cormick (Armacost & Fiacco 1974), (Fiacco 1976)... one of the first articles that applies it in an engineering context was from Sobieski and Barthelemy (Sobieszczanski-Sobieski *et al.* 1981). The main results stated here comes from the book of Fiacco (Fiacco 1983). However before giving these rather technical results, we will just start with a quite classical example of the interpretation of Lagrange multiplier for a simplified parametric optimization problem with only one equality constraint and where the parameter only appears in the right hand side.

$$\begin{array}{rlll} \mu^\star(p) := & \min_{x\in\mathbb{R}^n} & f(x) \\ & \text{s.t} & \left\{ \; h(x) = p \right. \end{array} \tag{2.35}$$

We assume that is Problem (2.35) is well-posed and has a unique solution for all p denoted $x^\star$ (we will see that in reality solution should be locally unique) and the associated Lagrange multiplier $\lambda^\star$. We introduce the Lagrangian of Problem (2.70) which also depends on p.

$$\begin{array}{rccl} \mathcal{L}: & \mathbb{R}^n \times \mathbb{R} \times \mathbb{R} & \mapsto & \mathbb{R} \\ & (x, \lambda, p) & \mapsto & \mathcal{L}(x, \lambda, p) = f(x) + \lambda.(h(x) - p) \end{array} \tag{2.36}$$

Note that

$$\mu^\star(p) = f(x^\star) = \mathcal{L}(x^\star, \lambda^\star, p) \tag{2.37}$$

indeed at the optimum, the equality constraint is satisfied and we have

$$h(x^\star) - p = 0 \tag{2.38}$$

However, $x^\star$ and $\lambda^\star$ depend on reality on p, so we make this dependence appear by writing $x^\star(p)$ and $\lambda^\star(p)$. Now assuming that these functions do exist and that $\mu^\star(.)$, $x^\star(.)$ and $\lambda^\star(.)$ are differentiable. We can now write

$$\frac{\partial \mu^\star(p)}{\partial p} = \frac{\partial \mathcal{L}(x^\star(p), \lambda^\star(p), p)}{\partial p} \tag{2.39}$$

[1]In reality, most of the following results were already known from mathematicians of the 19^{th} century: Zermelo, Darboux,...Hamilton himself derived the formulas in the context of infinite-dimensional optimization problems, more precisely in what we call today **optimal control theory**

and by applying chain rule, we get

$$\frac{d\mathcal{L}(x^\star(p),\lambda^\star(p),p)}{dp} = \frac{\partial x^\star(p)}{\partial p}\nabla_x\mathcal{L}(x^\star(p),\lambda^\star(p),p) + \frac{\partial \lambda^\star(p)}{\partial p}\nabla_\lambda\mathcal{L}(x^\star(p),\lambda^\star(p),p) \tag{2.40}$$

$$+ \frac{\partial \mathcal{L}(x^\star(p),\lambda^\star(p),p)}{\partial p} \tag{2.41}$$

Now using KKT conditions, we know that at the optimum the partial gradients of the Lagrangian vanish. More precisely

$$\nabla_x\mathcal{L}(x^\star(p),\lambda^\star(p),p) = 0 \tag{2.42}$$

and

$$\nabla_\lambda\mathcal{L}(x^\star(p),\lambda^\star(p),p) = 0 \tag{2.43}$$

Now inserting Eq. (2.39), Eq. (2.42) and Eq. (2.43) in Eq. (2.40), we get

$$\boxed{\frac{d\mu^\star(p)}{dp} = -\lambda^\star} \tag{2.44}$$

We see that, in that simple case, the Lagrange multiplier $\lambda^\star$ can be interpreted simply as the **derivative of the optimal value function with respect to** p. In economics, this interpretation is often referred to as the **marginal price**, that is the price we have to pay by making the constraints harder to satisfy in the case of inequality constraints. Indeed we could have derived exactly the same expression when changing the equality constraint into an inequality constraint using the same equations. However, we will see in an informal way that the treatment of inequality constraint is not as straightforward as the equality constraint. Indeed, what will appear in the following to derive post-optimal sensitivities results is that we need to assume that **the set of active inequality constraints remains unchanged in a neighborhood of** p. Consider the inequality parametric optimization problem

$$\begin{array}{rlll} \mu^\star(p) := & \min_{x\in\mathbb{R}^n} & f(x) \\ & \text{s.t} & \left\{\ g(x) \leqslant p \right. \end{array} \tag{2.45}$$

in that case the Lagrangian is

$$\begin{array}{rccl} \mathcal{L}: & \mathbb{R}^n\times\mathbb{R}\times\mathbb{R} & \mapsto & \mathbb{R} \\ & (x,\lambda,p) & \mapsto & \mathcal{L}(x,\lambda,p) = f(x) + \lambda.(g(x)-p) \end{array} \tag{2.46}$$

Using complementarity slackness condition (2.16), we still have

$$\mu^\star(p) = f(x^\star) = \mathcal{L}(x^\star(p), \lambda^\star(p), p) \tag{2.47}$$

since

$$\lambda^\star(g(x^\star) - p) = 0 \tag{2.48}$$

Now using KKT conditions we still have

$$\nabla_x \mathcal{L}(x^\star(p), \lambda^\star(p), p) = 0 \tag{2.49}$$

but we do not have anymore

$$\nabla_\lambda \mathcal{L}(x^\star(p), \lambda^\star(p), p) = 0 \tag{2.50}$$

Indeed in case the constraint is active, the former equality is true, but in case it is not this equality need not be satisfied. Just take the following trivial example

$$\begin{array}{ll} \min_{x\in\mathbb{R}^2} & x^2 + y^2 \\ \text{s.t} & \left\{ \; x^2 + y^2 \leqslant p(=1) \right. \end{array} \tag{2.51}$$

at the KKT point $(x^\star(1), y^\star(1)) = (0, 0)$ and $\lambda^\star(1) = 0$, we have

$$\nabla_\lambda \mathcal{L}((x^\star(1), y^\star(1)), \lambda^\star(1), 1) = -1 \neq 0 \tag{2.52}$$

To get to a tractable expression, we will need a further quite strong assumption that in a neighborhood of p, the inequality constraint does not become active, this way $\lambda^\star$ is constant in a neighborhood and then

$$\frac{d\lambda^\star(p)}{dp} = 0 \tag{2.53}$$

and this way we get to the following expression of the post-optimal sensitivities similar to Eq. (2.44).

$$\frac{d\mu^\star(p)}{dp} = -\lambda^\star \tag{2.54}$$

We can now use this formula to explain the sign of the Lagrange multiplier associated to an active inequality constraint. Suppose we now want to predict the new optimal value for a close parameter $p + h$ (where h is sufficiently small for the inequality constraint to remain active) and take $h > 0$ s.t the feasible space is greater (in the sense

of inclusion) than the one of the original problem, we can write the first-order Taylor development of $\mu^\star$

$$\mu^\star(p+h) = \mu^\star(p) + h\frac{d\mu^\star(p)}{dp} + o(||h||) \tag{2.55}$$

then

$$\mu^\star(p+h) = \mu^\star(p) - h\lambda^\star + o(||h||) \tag{2.56}$$

in terms of optimal value of f

$$f(x^\star(p+h)) = f(x^\star(h)) - h\lambda^\star + o(||h||) \tag{2.57}$$

since the feasible space for the perturbed problem is greater than the original one, we necessarily have

$$f(x^\star(p+h)) \leqslant f(x^\star(h)) \tag{2.58}$$

and then $\lambda^\star$ **has to be nonnegative**.

The former derivations were not at all rigorous. Indeed, we did not fuss over about the hypotheses to ensure for instance, existence and regularity properties (continuity and differentiability) of the optimal value function $\mu^\star$, of the optimal solution function (provided it is a function) $x^\star$ and of the optimal Lagrange multipliers function (provided it is a function) $\lambda^\star$. We also saw that in case of inequality constraints there were additional local assumptions such as there exists a neighborhood in the parameter space in which the perturbed problem (for $p+h$) remains essentially the same as the original unperturbed problem (for p). This hypothesis is essential or even critical since it leads to a non-differentiable optimal value function whenever we switch from one active constraint to another or when the active constraint ceases to be active, examples of such non-differentiability can be found in (Haftka & Gurdal 1992) for instance. The theory needed to derive such sensitivity results lies in classical technical tools described below.

- The main ingredient to show existence and regularity properties of the optimal functions is one of the most valuable theorem in applied mathematics, namely the **implicit function theorem**. This theorem (and its multiple versions) indeed guarantees existence and regularity of implicit functions defined through an

implicit relationship. More precisely, they treat the general problem of solving an equation of the form

$$F(x, y) = 0 \tag{2.59}$$

for x in terms of y, where F can handle a lot of different situations. In our case, this is of course the KKT conditions with $x = (x^\star, \lambda^\star)$ and $y = p$, but in other fields that we mention in next chapters such as the sensitivity of the solution of partial differential equation w.r.t parameters, the implicit relationship is either the variational formulation (infinite dimension version) or directly the linear system to solve whenever a finite-dimensional space of approximation has been set (finite dimension). In the different but related area of bifurcation and branching solutions of nonlinear problems (e.g buckling), the implicit function theorem is an essential tool to exhibit the link between the eigenvalues of the linearized problem (linear buckling) and bifurcation points of the nonlinear system (nonlinear buckling). Roughly speaking, the implicit function theorem will ensure the existence and the differentiability of functions of the sort of $x^\star(.)$.

- The **chain rule** is the practical tool to get the expressions of the derivatives of the implicit functions. Provided we can apply the implicit function theorem, namely to get the existence, continuity and differentiability of the implicit function φ that describes x in terms of y. The expression of the derivative follows easily from the chain rule applied to

$$F(\varphi(y), y) = 0 \tag{2.60}$$

we then get for instance for a one-dimensional parameter y

$$\frac{d\varphi}{dy}\nabla_x F + \frac{\partial F}{\partial y} = 0 \tag{2.61}$$

and then

$$\frac{d\varphi}{dy} = -(\nabla_x F)^{-1}\frac{\partial F}{\partial y} \tag{2.62}$$

meaning that usually we get analytical sensitivities of implicit functions by solving linear systems involving the Jacobian of F w.r.t x.

- As said below, the implicit function may not be always differentiable. To get general expression, one needs to go **non smooth optimization** using tools that generalize the notion of gradient such as sub-differential and Clarke's sub-gradient.

The following reference (Dempe 2002) is a valuable reference that introduces all this material together with a lot of theoretical results on the optimal value function.

We now give without proof the two main rigorous results for sensitivity analysis, proofs can be found in (Fiacco 1983). The first result deals with the hypotheses needed to apply the implicit function. The second is a corollary that gives explicit expressions to compute the post-optimal sensitivities of the optimal value functions. This theorem basically extends the results that we informally derived in case when the parameter only appears in the right hand member of constraints to the more general case when there are several constraints and both objective functions and constraints depend on p. Note that all these derivations are made for only one parameter p, but in practical situations we have many optimization parameters and the derivative $\frac{d\mu^\star(p)}{dp}$ should be seen as $\frac{\partial\mu^\star(p)}{\partial p}$.

Let's recall the prototypical problem

$$(\mathcal{P}(p)) \quad \begin{array}{ll} \min_{x\in\mathbb{R}^n} & f(x;p) \\ \text{s.t} & \left\{ \begin{array}{l} g(x;p) \leqslant 0 \\ h(x;p) = 0 \end{array} \right. \end{array} \tag{2.63}$$

To assess the derivative of the optimal functions w.r.t, we need also to consider f, g and h as function depending on both x and p in $(\mathcal{P}(p))$.

Theorem 1. *(Basic sensitivity theorem (Fiacco, 1976)) Consider $x^\star$ a local minimizer of $(\mathcal{P}(p))$, if*

(i) f, g and h are twice continuously differentiable w.r.t x, and $\nabla_x f$, $\nabla_x g$ and $\nabla_x h$ are once continuously differentiable w.r.t to p in a neighborhood of $(x^\star, p)$,

*(ii) Second order sufficient conditions (**SC2CO**) hold at $x^\star$ and the associated Lagrange multiplier are $\lambda_E^\star$ and $\lambda_I^\star$,*

*(iii) Linear independence constraint qualification (**LICQ**) holds at $x^\star$,*

*(iv) Strict complementarity slackness condition (**SCSC**) holds at $x^\star$,*

Then

*(a) $x^\star$ is a **local isolated minimizer**, i.e locally unique and the associated Lagrange multipliers are unique*

(b) For h sufficiently small, i.e in a neighborhood of p, note $p' = p + h$, there exist ***uniquely defined continuously differentiable functions*** *$x^\star(p')$, $\lambda_E^\star(p')$ and $\lambda_I^\star(p')$ that satisfy the second-order sufficient conditions for Problem $\mathcal{P}(p')$ s.t $x^\star(p) = x^\star$, $\lambda_E^\star(p) = \lambda_E^\star$, and $\lambda_I^\star(p) = \lambda_I^\star$. $x^\star(p')$ is also* ***locally unique*** *and the associated Lagrange multipliers are $\lambda_E^\star(p')$ and $\lambda_I^\star(p')$. Denote also $\lambda^\star(p') = (\lambda_E^\star(p'), \lambda_I^\star(p'))$*

(c) For h sufficiently small, the set of active inequalities in Problem $\mathcal{P}(p')$ remains unchanged w.r.t the ones in Problem $\mathcal{P}(p)$, strict complementarity slackness conditions also hold and satisfy LICQ.

This theorem is the first step to get to a tractable expression of the derivative of the optimal value function. Indeed, once the existence and regularity of the implicit functions is guaranteed, we can then apply the chain rule as a corollary. We now state useful results with that expression. Based on the former theorem, we can define formally the following function

- Define the **(local) optimal value function** $\mu^\star$

$$\mu^\star(p) = f(x^\star(p), p) \tag{2.64}$$

  note that, such a function $\mu^\star$ is a **nested function** in the sense it implicitly considers the resolution of an optimization problem. The same way, when describing structural optimization problem we define also nested constraints that implicitly consider the resolution of the elasticity system of equations (or in discretized version an FEA).

- Define the **(local) optimal value Lagrangian** $\mathcal{L}^\star$

$$\mathcal{L}^\star(p) = \mathcal{L}(x^\star(p), \lambda^\star(p), p) \tag{2.65}$$

We can now state the following theorem.

Theorem 2. *(First-order derivative of the optimal value function (Fiacco, 1976)). Whenever conditions of theorem 1 hold. Then there exists a neighborhood of p, i.e there exists $h > 0$ s.t for any $p' \in [p - h, p + h]$*

(a) The optimal value function and the optimal value Lagrangian are equal

$$\mu^\star(p') = \mathcal{L}^\star(p') \tag{2.66}$$

(b) The derivative of $\mu^\star$ w.r.t p is

$$\boxed{\frac{d\mu^\star(p')}{dp} = \frac{\partial f(x^\star, p')}{\partial p} + \sum_{i\in I} \lambda_i^\star(p') \frac{\partial g_i(x^\star, p')}{\partial p} + \sum_{j\in E} \lambda_j^\star \frac{\partial h_j(x^\star, p')}{\partial p}} \tag{2.67}$$

Remark Note that the $\frac{\partial f(x^\star,p')}{\partial p}$ term correspond to the **partial derivative of** $f(x,p)$ **w.r.t** p **with no implicit dependence of** $x^\star(p)$. This is different from the **total derivative of** $f(x^\star(p),p)$ **w.r.t** p which is nothing else than $\frac{d\mu^\star(p)}{dp}$. The same goes for the $\frac{\partial g_i(x^\star,p')}{\partial p}$ and $\frac{\partial h_j(x^\star,p')}{\partial p}$.

2.2.6 Original application of post-optimal sensitivities results

The following example illustrates the formulas seen below. Consider the following classical result from linear algebra. Consider A a $n \times n$ symmetric matrix. The lowest eigenvalue λ of A can be shown to be the (global) **minimum value of the Rayleigh quotient** (this result is known as the **Courant-Fisher theorem**, see for instance (Lascaux & Théodor 1987), and can be extended to self-adjoint linear operators, see next chapter)

$$R(x) = \frac{x^T A x}{x^T x} \tag{2.68}$$

over $\mathbb{R}^n$. Equivalently, we have

$$\begin{array}{rll} \lambda_1 := & \min_{x\in\mathbb{R}^n} & x^T A x \\ & \text{s.t} & \left\{ \; x^T x = 1 \right. \end{array} \tag{2.69}$$

Now consider a parametric eigenvalue problem, such as the ones that structural optimization problems make appear: buckling constraints (computation of the critical buckling load) or vibrations constraints (computation of the first fundamental frequency, see (Gürdal *et al.* 1999)), flutter constraints (positive real part of the eigenvalue),... Namely the matrix A depends on a parameter p and we note

$$\begin{array}{rll} \lambda_{cr}(p) := & \min_{x\in\mathbb{R}^n} & x^T A(p) x \\ & \text{s.t} & \left\{ \; x^T x = 1 \right. \end{array} \tag{2.70}$$

We know from linear algebra that $\lambda_{cr}(p) = \lambda_1$, where λ_1 is the smallest eigenvalue (that might be repeated) and this minimum is attained for v_1 an eigenvector of norm 1 s.t

$$A(p)v_1 = \lambda_1 v_1 \tag{2.71}$$

In this equality, we skipped the dependency but it is clear that both λ_1 and v_1 depend implicitly on p. We are interested here in computing $\frac{d\lambda_{cr}(p)}{dp}$ to get the sensitivity of the eigenvalue w.r.t p. This is precisely the optimal value function considered in the former sections. To apply the former theorems we need to ensure the hypothesis. Note

- $f(x,p) = x^T A(p)x$, where we assume that $A : \mathbb{R} \mapsto \mathcal{M}_n(\mathbb{R})$ is sufficiently regular, for instance once continuously differentiable, and $h(x,p) = x^T x$. Then we have

$$\nabla_x f(x,p) = 2A(p)x \tag{2.72}$$

and

$$\nabla^2_{xx} f(x,p) = 2A(p) \tag{2.73}$$

and

$$\nabla^2_{xp} f(x,p) = 2A'(p)x \tag{2.74}$$

and $g(x,p) = x^T x$ which does not depend on p. Then

$$\nabla_x h(x,p) = 2x \tag{2.75}$$

and

$$\nabla^2_{xp} h(x,p) = 0 \tag{2.76}$$

and

$$\nabla^2_{xx} h(x,p) = 2Id \tag{2.77}$$

- Note that the gradient of the Lagrangian w.r.t x is then

$$\nabla_x \mathcal{L}(x,\lambda,p) = 2A(p)x + 2\lambda x \tag{2.78}$$

which is precisely the eigenvalue system, we then have at the optimum v_1

$$2(A(p)v_1 + \lambda^\star)v_1 = 0 \tag{2.79}$$

and so we have

$$\lambda^\star = -\lambda_1 \tag{2.80}$$

and that the Hessian of the Lagrangian w.r.t x is

$$\nabla^2_{xx} \mathcal{L}(x,\lambda) = 2(A(p) + \lambda Id) \tag{2.81}$$

hence all the hypothesis of regularity are satisfied (i). Hypotheses (iii) (v_1 is non zero, by definition of an eigenvector) and (iv) are trivially satisfied (in case A is such that $\lambda_1 = 0$, we can still artificially make λ_1 non zero by adding a constant, this is known in numerical approximation of eigenproblems as the spectral offset). Our main concern is about hypothesis (ii) which is that second order sufficient conditions hold at $x^\star$. Indeed, second order sufficient conditions in that case take the following form (with no inequality constraints)

$$z^T \nabla^2_{xx} \mathcal{L}(v_1, -\lambda_1, p) z > 0 \tag{2.82}$$

for any z s.t $v_1^T z = 0$, i.e in the orthogonal space $(v_1)^\perp$. Eq (2.82) takes the following form

$$\nabla^2_x \mathcal{L}(v_1, -\lambda_1, p) = 2(A(p) - \lambda_1 Id) \tag{2.83}$$

and we already now that this matrix is positive semi-definite from necessary conditions. We then need to ensure that this matrix is definite over $(v_1)^\perp$, stated in other words, there is not $v \in (v_1)^\perp$ s.t v is an eigenvector for λ_1. Clearly this is ensured whenever λ_1 is simple. In that case $\mathrm{Ker}(A - \lambda_1 Id)$ is the 1-dimensional space generated with v_1, in that case, we see that no vector from $(v_1)^\perp$ can be an eigenvector for λ_1. We then assume that $\lambda_{cr} = \lambda_1$ is a **simple eigenvalue** and sufficient second order conditions are then ensured.

We now apply Theorem (2) to get

$$\frac{\partial \lambda_{cr}}{\partial p} = \frac{\partial f(v_1, p)}{\partial p} \tag{2.84}$$

and then

$$\frac{\partial \lambda_{cr}}{\partial p} = v_1^T A'(p) v_1 \tag{2.85}$$

which corresponds to the formula found for instance in (Haftka & Gurdal 1992). This derivation was not only to illustrate the post-optimal sensitivity since we will make use of that formula to compute analytical sensitivities of buckling critical factor in Chapter 3. Note that in case where p also appears in the constraints, for instance for generalized eigenvalues problems with constraints such as $x^T B(p) x = 1$, the same can be applied to derive the following formula

$$\frac{\partial \lambda_{cr}}{\partial p} = v_1^T (\frac{\partial A}{\partial p} + \lambda_1 \frac{\partial B}{\partial p}) v_1 \tag{2.86}$$

with v_1 the associated eigenvector whenever λ_1 is simple.

2.3 Monolevel and block-angular problems

2.3.1 Original problem

We are concerned here with large scale optimization problems arising in engineering: structural optimization, multidisciplinary optimization...which have the following form

$$\boxed{\begin{array}{lll} & \min_x & \hat{M}(x) \\ (AIO) & \text{s.t} & \begin{cases} \hat{c}(x) \leqslant 0 & \\ \hat{c}_i(x) \leqslant 0 & \forall i = 1 \dots N \end{cases} \end{array}} \tag{2.87}$$

We will assume in the sequel that, both objective and constraints are computed on the basis of a few responses systems Z that are function of all x through coarser variables Y. To illustrate this kind of problem, let's think of a structural optimization problem where most constraints are based on the analysis of the whole structure and hence depend on the loads redistribution that is function of stiffness terms Y (cross section area, quadratic momentum,...), when x are the precise geometric dimensions, or material variables they usually not appear directly on the structural analysis. The coarser variables Y are enough to analyze the structure. However, this coarser variables are often a function of the local design variables x, $Y = \varphi(x)$. This dependence naturally a hierarchical optimization problem, where the structural responses are computed on the basis of the Y's and theses Y's are expected to match the x. This is not uncommon in engineering optimization, most of constraints are computed on the basis of systems responses (that depend on all variables) but only a few local design variables are concerned by each constraint. We denote this situation by using the following relationships

$$\hat{m}(x) = m(\varphi(x)) \tag{2.88}$$

and

$$\hat{c}(x) = c(\varphi(x)) \tag{2.89}$$

and

$$\hat{c}_i(x) = c_i(x^{(i)}, Y) \tag{2.90}$$

where $x^{(i)}$ are the few local design variables involved in the computation of c_i. Again, in structural optimization, m can be the weight (that can often be computed through local design variables or Y's), c can be the structural constraint that do not involve

local geometry of substructures (linear maximum deformation for instance), and c_i can be local stability constraints (e.g buckling) that are based both on the structural responses and local geometry. We do not consider here inter-element constraints (e.g continuity of the design) over $x^{(i)}$ for different i's. However, some coarse continuity constraints can be ensured through the $c(Y)$. Problem 2.87 will be referred to as the *All-In-One* (AIO) formulation (or *Nested*). Using the Y variables in the formulation of the optimization problem leads to the classical block angular form

$$\boxed{\begin{array}{lll} & \min_{x,Y} & \hat{M}(Y) \\ (BAO) & \text{s.t} & \left\{\begin{array}{l} c(Y) \leqslant 0 \\ c_i(x^{(i)}, Y) \leqslant 0 \quad \forall i = 1 \dots N \\ Y_i = \varphi(x^{(i)}) \quad \forall i = 1 \dots N \end{array}\right. \end{array}} \tag{2.91}$$

2.3.2 Equivalence between All-In-One problem and Block-Angular problem

In the sequel we will show that first Problems (2.87) and (2.91) are equivalent in terms of optimization, which means that any KKT point of one problem is actually a KKT point of the other.

Theorem 3. *$x^\star$ is a minimizer of Problem (2.87) satisfying linear independence constraint qualification if and only if $(x^\star, y^\star)$ is a minimizer of Problem (2.91) satisfying linear independence constraint qualification.*

Proof. For sake of simplicity, this proof is given for two sub-problems, adaptation to N sub-problems is easy and only complicates notations. Our reference problem is then

$$\begin{array}{lll} & \min_{x} & \hat{M}(x) \\ (\mathbf{AIO}) & \text{s.t} & \left\{\begin{array}{l} \hat{c}(x) \leqslant 0 \\ \hat{c}_1(x) \leqslant 0 \\ \hat{c}_2(x) \leqslant 0 \end{array}\right. \end{array} \tag{2.92}$$

and then BAO version is

$$\begin{array}{lll} & \min_{x,Y} & \hat{M}(Y) \\ (\mathbf{BAO}) & \text{s.t} & \left\{\begin{array}{l} c(Y) \leqslant 0 \\ c_1(x^{(1)}, Y) \leqslant 0 \\ c_2(x^{(2)}, Y) \leqslant 0 \\ Y_1 = \varphi(x^{(1)}) \\ Y_2 = \varphi(x^{(2)}) \end{array}\right. \end{array} \tag{2.93}$$

Let $x^\star$ be a minimizer of Problem (2.92) satisfying LICQ. In particular, there exists λ_1, λ_2 and λ_3 in $\mathbb{R}^-$ such that

$$\begin{pmatrix} \nabla_{x^{(1)}}\hat{m}(x^\star) \\ \nabla_{x^{(2)}}\hat{m}(x^\star) \end{pmatrix} = \begin{pmatrix} \nabla_{x^{(1)}}\hat{c}(x^\star) & \nabla_{x^{(1)}}\hat{c}_1(x^\star) & 0 \\ \nabla_{x^{(2)}}\hat{c}(x^\star) & 0 & \nabla_{x^{(2)}}\hat{c}_2(x^\star) \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{pmatrix} \tag{2.94}$$

Consider $Y^\star = \varphi(x^\star)$ and using chain rule, we get

$$\begin{pmatrix} \nabla_{x^{(1)}}\varphi\nabla_Y m \\ \nabla_{x^{(2)}}\varphi\nabla_Y m \end{pmatrix} = \begin{pmatrix} \nabla_{x^{(1)}}\varphi\nabla_Y c & \nabla_{x^{(1)}}c_1 + \nabla_{x^{(1)}}\varphi\nabla_Y c_1 & 0 \\ \nabla_{x^{(2)}}\varphi\nabla_Y c & 0 & \nabla_{x^{(2)}}c_2 + \nabla_{x^{(2)}}\varphi\nabla_Y c_2 \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{pmatrix} \tag{2.95}$$

where we skip the dependence of function w.r.t x and Y. We clearly have

$$\begin{pmatrix} 0 \\ 0 \\ \nabla_Y m \end{pmatrix} = \begin{pmatrix} 0 & \nabla_{x^{(1)}}c_1 & 0 & \nabla_{x^{(1)}}\varphi \\ 0 & 0 & \nabla_{x^{(2)}}c_2 & \nabla_{x^{(2)}}\varphi \\ \varphi\nabla_Y c & \varphi\nabla_Y c_1 & \varphi\nabla_Y c_2 & -I \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ -\nabla_Y m + \lambda_1^T\nabla_Y c + \lambda_2^T\nabla_Y c_1 + \lambda_3^T\nabla_Y c_2 \end{pmatrix} \tag{2.96}$$

from which it follows that $(x^\star, y^\star)$ is a KKT point of Problem (2.91) with $\mu = -\nabla_Y m + \lambda_1^T\nabla_Y c + \lambda_2^T\nabla_Y c_1 + \lambda_3^T\nabla_Y c_2$ being the KKT multipliers associated to the equality constraints $\varphi(x) - Y$.

Now suppose that $(x^\star, y^\star)$ does not satisfy linear independence constraints qualification, there exist $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ not all zeros such that

$$\begin{pmatrix} 0 & \nabla_x c_1 & \nabla_x c_2 & \nabla_x\varphi \\ \nabla_Y c & \nabla_Y c_1 & \nabla_Y c_2 & -I \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{pmatrix} = 0 \tag{2.97}$$

then from last row, we get

$$\alpha_4 = \nabla_Y c\alpha_1 + \nabla_Y c_1\alpha_2 + \nabla_Y c_2\alpha_3 \tag{2.98}$$

now inserting Eq.(2.98) into the first row, we get

$$\nabla_x\varphi\nabla_Y c\alpha_1 + (\nabla_x c_1 + \nabla_x\varphi\nabla_y c_1)\alpha_2 + (\nabla_x c_2 + \nabla_x\varphi\nabla_y c_2)\alpha_3 = 0 \tag{2.99}$$

and $x^\star$ does not satisfy linear independence qualification constraints. Hence linear constraints qualification of AIO imply linear constraint qualification of BAO. The last part of the proof consists in taking one KKT point of BAO and shows it is also a KKT point of AIO by using exactly the same computations, the other way around. □

2.4 Towards decomposition through inexact quadratic penalty

In this section we show by applying standard results of optimization theory that the Block-angular problem optimization problem is equivalent to the inexact quadratic penalty problem. We start with very briefly reviewing inexact and exact penalties and states a classical result that shows in what sense these two problems are equivalent. Most of the material described is derived from (Fletcher 1981).

2.4.1 Inexact and exact penalties

The main idea of penalty is simply to insert a given constraint into the objective function. This way we get rid of the constraint and the optimization becomes unconstrained. Illustrate this in the following problem

$$\begin{array}{ll} \min_x & f(x) \\ \text{s.t} & \left\{ \ c(x) = 0 \right. \end{array} \tag{2.100}$$

We want to replace this problem by an unconstrained problem where a certain function is to be minimized

$$\varphi(x, \gamma) \tag{2.101}$$

where γ is a penalty parameter that weights the constraint w.r.t to the objective function. Problem (2.100) is then replaced by

$$\min_x \quad \varphi(x, \gamma) \tag{2.102}$$

γ is Problem (2.102) is then a fixed parameter (exactly as the optimization parameter p in the former sections). We will denote then $x_\gamma = x^\star(\gamma)$, i.e the solution of Problem (2.102) for the corresponding value of γ, provided there exists a unique (global) solution. The earliest penalty function that was introduced in (Courant 1994) is the **quadratic penalty**

$$\varphi(x, \gamma) = f(x) + \frac{1}{2}\gamma c(x)^2 \tag{2.103}$$

and in case when c is a vector-valued constraint in $\mathbb{R}^p$ we have

$$\varphi(x, \gamma) = f(x) + \frac{1}{2}\gamma \sum_{i=1}^{p} (c_i(x))^2 \tag{2.104}$$

$$= f(x) + \frac{1}{2}\gamma c(x)^T c(x) \tag{2.105}$$

in case of inequality constraints, this turns out to be

$$\varphi(x,\gamma) = f(x) + \frac{1}{2}\gamma \sum_{i=1}^{p} (\max(0, c_i(x))^2 \tag{2.106}$$

This penalty is the sum of squares of the constraints violations. We see that the solution of the unconstrained problem for a given value of γ can be different from the solution of the initial problem. This is why this type of penalty is called **inexact penalty**. There exists however **exact penalty** which guarantees that the minimum of the unconstrained problem is the same as the solution of the initial problem. For instance, the L_1 exact penalty

$$\varphi(x) = f(x) + \sum_{i=1}^{p} |c_i(x)| \tag{2.107}$$

this exact penalty does not use any parameter, since a (local) solution of the minimization problem of φ is given by $x^\star$ the solution of the initial problem. The main drawback of this exact penalty approach is that the resulting problem is no more smooth ($|.|$ being not differentiable at 0) all the gradient-based methods or Newton-like and quasi-Newton methods can not be applied anymore and one has to go nonsmooth optimization tools. In the sequel, we only focus on inexact quadratic penalty. One important feature of inexact penalty methods is they do not directly give an approximate solution of the initial problem. In practice, inexact penalty leads to the following resolution

1. Choose a sequence of γ_k that goes to ∞. For instance, a typical such sequence is given in (Fletcher 1981) as $1, 10, 10^2, ...$
2. For γ_k, find a (local) minimizer $x_k^\star = x^\star(\gamma_k)$ of Problem (2.102).
3. Termination criterion is attained when $c(x_k^\star)$ is sufficiently small.

We then see that inexact penalty leads to a **sequence** of unconstrained minimization problems, whose sequential resolution can be eased by using the solution of for γ_k as a starting point for solving problem for γ_{k+1}, the same goes for the approximation of the inverse of the Hessian of the objective function whenever quasi Newton (Broyden, DFP, BFGS) methods are involved. This inexact penalty transforms leads then to **Sequential Minimization Optimization**. However, one doe not solve many different optimization problems in practice. In some situations when the parameter γ is well set

(from experience for instance), one might end up with a reasonably good approximation of the initial solution.

2.4.2 Equivalence of the inexact quadratic penalty for block-angular problems

We now state the main theoretical result taken from (Fletcher 1981).

Theorem 4. *(Penalty function convergence) If γ_k goes to ∞ then*

(i) $(\varphi(x_k^\star, \gamma_k))_k$ *is non decreasing*

(ii) $(c(x_k^\star)^T c(x_k^\star)_k$ *is non decreasing*

(iii) $(f(x_k^\star)_k$ *is non decreasing and* $c(x_k^\star) \to 0$ *and any accumulation point* $x^\star$ *of* $(f(x_k^\star))_k$ *solves initial Problem 2.100.*

In our case, this leads to the following problem, referred to as the **Inexact quadratic penalty block angular form** (IQP-BAF)

$$\boxed{(IQP-BAF) \quad \begin{array}{ll} \min_{y,x} & f(y) + \frac{1}{2}\gamma^k \sum_{i=1}^n ||\varphi(x_i) - y_i||^2 \\ \text{s.t} & \begin{cases} c(y) \leqslant 0 \\ c_i(y, x_i) \leqslant 0 \end{cases} \end{array}} \tag{2.108}$$

Now using this theorem in combination with Theorem 3, we can state the following theorem

Theorem 5. *Problem AIO (2.87) and Problem IQP-BAF 2.108 are equivalent optimization problems in the following sense:*
Consider a sequence of parameters $(\gamma^k)_k$ that goes to ∞ and the associated sequence of optimization problems and its associated local minimizers $(x_k^\star)$. Then any accumulation point $x^\star$ of $(x_k^\star)_k$ is a (local) minimizer of the AIO problem.

This results is not only important for theoretical but also for practical reasons. Indeed, we saw that we had three different monolevel instances of our optimization problem: AIO (2.87), BAO (2.91) and IQP-BAO (2.108), each of them will be decomposed leading to different multilevel or decomposition-aggregation schemes. The fact these three instances are equivalent is a first step to guarantee that we do not change the optimum before decomposition. However, in practice and as noted before, we do not solve a sequence of optimization problems but rather fix a reasonably good value for the penalty parameter γ. We turn now to the brief description of the decomposition schemes.

2.5 Towards decomposition

We end this chapter with a presentation of the decomposition schemes that are to be applied in Chapter 6. These schemes are only presented from a theoretical point of view since we will make them more precise in the context of a real structural optimization problem in Chapter 6 and practical resolution will be considered in Chapter 6 and not in this chapter. For example, the decomposition approach of the initial problem (AIO) is a practical resolution and does not rely on a decomposition or bilevel formulation. The emphasis here is made on formulations independently from the resolution. Indeed, the practical resolution of decomposition schemes is a full subject in the sense that several strategies might be applied (alternating optimizations implementation or real bilevel implementation). This is why we do not describe here the practical algorithm.

2.5.1 Decomposition scheme from the (AIO) initial problem

Before decomposing the block-angular versions of the optimization problem, we can try to decompose the original problem (AIO). This leads to a classical algorithm of practical resolution in structural optimization based on the Fully Stressed Criterion (StiffOpt algorithm described in Chapter 6).

2.5.2 Target Rigidity and Mix schemes

We then mainly focus on formulations of decomposition techniques or bilevel optimization scheme. The typical formulation of a decomposition scheme or bilevel optimization scheme is the following

- Global-level (or system-level) :

$$\begin{array}{ll} \min_Y & M(Y) \\ \text{s. t.} & \left\{ \begin{array}{l} c(Y) \leq 0 \\ \mu_i^\star(Y) \leq 0 \quad \text{for} \quad i = 1 \dots N \end{array} \right. \end{array} \tag{2.109}$$

- where $\mu_i^\star$ for $i = 1 \dots N$ is the coupling function computed from the lower level, we keep the notation for the optimal value function for it is often computed on the basis of the value of the objective function at the optimum.

$$\begin{array}{ll} \min_{x_i} & h_i(x_i, Y) \\ \text{s. t.} & \left\{ \; c^{(i)}(Y, x_i) \leq 0 \right. \end{array} \tag{2.110}$$

Now depending on which monolevel problem is decomposed we get different schemes.

Target Rigidity is the straightforward decomposition from problem 2.108, which leads to

- Global-level (or system-level) :

$$\begin{array}{ll} \min_Y & M(Y) \\ \text{s. t.} & \left\{ \begin{array}{l} c(Y) \leq 0 \\ \mu_i^\star(Y) \leq 0 \quad \text{for} \quad i = 1 \dots N \end{array} \right. \end{array} \tag{2.111}$$

- where $\mu_i^\star$ for $i = 1 \dots N$ is the function computed from the lower level :

$$\begin{array}{ll} \min_{X_i} & \frac{1}{2}||\varphi(x_i) - Y_i||^2 \\ \text{s. t.} & \left\{ c_i(Y, x_i) \leq 0 \right. \end{array} \tag{2.112}$$

  we define $\mu_i^\star(Y) = \frac{1}{2}||\varphi(x_i^\star - Y_i)||^2$, i.e the value of the objective function at the optimum $x_i^\star$, it is the optimal value function defined in the former sections.

we see that the constraint $\mu_i^\star(Y) \leqslant 0$ is in reality $\mu_i^\star(Y) = 0$ or equivalently by inserting the $\mu_i^\star$ constraint into the objective of the upper level by inexact quadratic penalty, we happen to have

- Global-level (or system-level) :

$$\begin{array}{ll} \min_Y & M(Y) + \frac{1}{2}\sum_{i=1}^{N} \gamma^k ||\varphi(x_i^\star) - Y_i||^2 \\ \text{s. t.} & \left\{ c(Y) \leq 0 \right. \end{array} \tag{2.113}$$

This scheme, which was originally derived in (Ramanathan 1976) in the framework of structural optimization is similar to the Collaborative Optimization scheme in the Multi Disciplinary Optimization (MDO) framework derived in (Braun & Kroo 1997) in the sense that they both intent to minimize the distance between coupling variables. However, the main difference is that in CO the Y's variables are replicated into each sub-problem and appear as optimization variables within the lower level problem and the upper level optimization tries to minimize the difference between all replicates of Y's by using intervening variables. In our framework, this approach could not lead to something attractive in terms of computations since we want to keep the finite element analysis at upper level while applying CO directly would lead to finite element analysis performed at lower level.

Remark One immediate pitfall of this scheme comes from constraint qualifications. Indeed, at an optimum of the original problem $x^\star, Y^\star$, we have

$$\varphi(x_i^\star) = Y_i^\star \tag{2.114}$$

Denote $f_i(x_i, Y) = 1/2||\varphi(x_i) - Y_i||^2$ and then the (local) gradient of the objective function (by local, we mean w.r.t local variables) of sub-problem i is

$$\nabla_{x_i} f_i(x_i, Y) = 1/2\nabla_x \varphi(x_i^\star)||\varphi(x_i^\star) - Y_i^\star|| \tag{2.115}$$

which is obviously zero, hence sub-problem i **does not satisfy LICQ**. Now suppose that the initial problem AIO satisfies LICQ at $x^\star$, then BAO satisfies also LICQ, in particular $\nabla_{x_i} c_i(x_i^\star, Y^\star)$ are linearly independent hence nonzero. Since $\nabla_{x_i} f_i(x_i^\star, Y^\star) = 0$ this implies that **the Lagrange multipliers (of sub-problem i) associated to c_i are zero**, which means precisely that the **Strict Complementarity Slackness Condition does not hold at** $x^\star$. The sub-problem is called in that case **degenerate**. Furthermore, we clearly see that we do not fit the conditions for the optimal value function to be differentiated. Indeed, the (global) gradient at $Y^\star$ is

$$\nabla_Y f_i(x_i^\star, Y^\star) = \begin{pmatrix} 0 \\ \vdots \\ -1/2||\varphi(x_i^\star) - Y_i^\star|| \\ \vdots \\ 0 \end{pmatrix} \tag{2.116}$$

which is also zero. This leads to a **non-qualified problem and to the optimal value function $\mu_i^\star$ which is not smooth**. As outlined in (DeMiguel & Murray 2006), smoothness and qualification are usual assumptions to show local convergence properties. Moreover, as reported in (Alexandrov *et al.* 2000) and (Alexandrov & Lewis 1999), this leads to non convergent algorithm in the case of CO, even when using a starting point very close to the optimum the algorithm may not converge to the real optimum. In our case, we also observed that **Target Rigidity leads to sub-optimal design even when starting at the real AIO optimum**.

We now present a quite close original scheme that can be derived in our case based on the following observation that the weight of element i can be computed both on the

basis of the Y_i and the design variables x_i's. This way we can write the local weight as a convex combination of both objectives

$$\varepsilon\hat{M}(x_i) + (1-\varepsilon)M(Y_i) \tag{2.117}$$

leading the **Mix scheme**

- Global-level (or system-level) :

$$\begin{array}{ll} \min_Y & M(Y) \\ \text{s. t.} & \left\{ \begin{array}{l} c(Y) \leq 0 \\ \mu_i^\star(Y) \leq 0 \quad \text{for} \quad i = 1 \dots N \end{array} \right. \end{array} \tag{2.118}$$

- where $\mu_i^\star$ for $i = 1 \dots N$ is the function computed from the lower level :

$$\begin{array}{ll} \min_{x_i} & \varepsilon\hat{M}(x_i) + (1-\varepsilon)\frac{1}{2}||\varphi(x_i) - Y_i||^2 \\ \text{s. t.} & \left\{ \; c_i(Y, x_i) \leq 0 \right. \end{array} \tag{2.119}$$

  where $\mu_i^\star(Y) = \frac{1}{2}||\varphi(x_i^\star - Y_i)||^2$

This way the degeneracy of the sub-problem can be avoided. However there is no evidence neither theoretical nor computational that this scheme performs better than Target Rigidity. In particular, the ε term should set carefully. These two schemes are compared in Chapter 6 and the Mix scheme shows very slight improvement when compared to Target Rigidity, it is still sub-optimal.

We finally present the **MinMass** scheme derived in (Merval 2008). This scheme is close to Target Rigidity or Mix schemes in the sense that the constraint $\varphi(x_i) = Y_i$ is relaxed to allow more freedom in the local search.

- Global-level (or system-level) :

$$\begin{array}{ll} \min_Y & M(Y) \\ \text{s. t.} & \left\{ \begin{array}{l} c(Y) \leq 0 \\ \mu_i^\star(Y) \leq 0 \quad \text{for} \quad i = 1 \dots N \end{array} \right. \end{array} \tag{2.120}$$

- where $\mu_i^\star$ for $i = 1 \dots N$ is the function computed from the lower level :

$$\begin{array}{ll} \min_{x_i} & \hat{M}(x_i) \\ \text{s. t.} & \left\{ \begin{array}{l} c_i(Y, x_i) \leq 0 \\ \varphi(x_i) \geqslant Y_i \end{array} \right. \end{array} \tag{2.121}$$

  we define $\mu_i^\star(Y) = \frac{1}{2}||\varphi(x_i^\star - Y_i)$, i.e the value of the objective function at the optimum $x_i^\star$, it is the optimal value function defined in the former sections.

2.5.3 MaxMargin and QSD schemes

The scheme named **MaxMargin** (from **Maximum Margin** a.k.a **Constraint Margin**) is different from the latter schemes since is does not use the local objective of minimizing the weight but instead maximize the feasibility of the design at lower level. From a decomposition perspective, its early version decomposes the initial block-angular form with the equality constraint. This scheme was carried out first by Sobiesky in the 80's in (Sobieszczanski-Sobieski *et al.* 1985), latter complemented in (BARTHELEMY 1988), approximation models were inserted in this scheme in (Liu *et al.* 2004). The scheme then extended in a more general form called **Quasi Separable subsystem Decomposition** (QSD) by Haftka and Watson who gave theoretical results in ((Haftka & Watson 2005)), one of them being that **the decomposition does not introduce any artificial minimum**, any **minimum of the decomposition/bilevel formulation is indeed a minimum of the original problem**. In our case, the original problem is (IQP-BAF). Based on the equivalence results we derived in the former sections, we can then conclude that **any minimum of QSD is indeed a minimum of AIO**. It was also extended to mixed problems ((Haftka & Watson 2006)) and a recent realistic structural optimization example is given in ((Schutte & Haftka 2010)).

- **MaxMargin** defines $\gamma_i(Y) = \min_{x_i \in \varphi^{-1}(Y)}\{c_i(Y, x_i^*)\}$ and therefore $\gamma_i(Y) = -\mu_i^*$
- where $\mu_i^\star$ is equal to

$$\begin{array}{ll} \text{argmax}_{\mu_i, x_i} & \mu_i \\ \text{s. t.} & \left\{ \begin{array}{l} c_i(Y, x_i) + \mu_i \leq 0 \\ c_i(x_i) + \mu_i \leq 0 \\ Y_i = \varphi(x_i) \end{array} \right. \end{array} \tag{2.122}$$

and the **QSD** form in its general formulation introduces budget variables b_i that allows for non feasibility at local design level.

- Global-level (or system-level) :

$$\begin{array}{ll} \min_{Y,b} & M(Y) + \sum_{i=1}^{N} b_i \\ \text{s. t.} & \left\{ \begin{array}{l} c(Y) \leq 0 \\ \mu_i^\star(Y, b_i) \leq 0 \quad \text{for} \quad i = 1 \ldots N \end{array} \right. \end{array} \tag{2.123}$$

- where $\mu_i^\star$ for $i = 1 \dots N$ is the coupling function computed from the lower-level (or sub-element level) :

$$\begin{array}{ll} \min_{(\mu_i, x_i)} & \mu_i \\ \text{s. t.} & \left\{ \begin{array}{l} c_i(Y, x_i) - \mu_i \leq 0 \\ \frac{1}{2}||Y_i - \varphi(X_i)||^2 - b_i - \mu_i \leq 0 \end{array} \right. \end{array} \tag{2.124}$$

2.5.4 Towards realistic optimization problems

This chapter was essentially theoretical and was aimed at giving essential tools that will be used throughout this thesis. However, we only insisted here on the structure of our large problem to solve and only describe the theoretical aspects of decompositions. Before applying these schemes that we introduced at the end of the chapter, we need first to assess the types of variables that we are to use for composite representation and also to investigate the behavior of the buckling constraints. This is done in the next chapter

3 Optimization properties for composite problems

This chapter is concerned with the definition and the investigation of the buckling critical load factor for thin composite plates. The main objective of that chapter is first to introduce the classical laminate representation that we use in this work: the well-known lamination parameters and then to investigate the behavior of the buckling critical load factor over this representation in order to be able to derive an efficient approximation strategy. Basics of composite laminate theory are recalled, we do not delve into the very precise derivation of the constitutive equation (A, B, D tensors) but simply indicate how these tensors are computed on the basis of lamination parameters. The buckling critical load factor for thin laminate plates is then defined formally and investigated, this behavior is of utter importance since amongst all the stability constraints computed at super-stiffener level by Airbus skill tools, the most complicated ones are derived from this first computation that is skin buckling computation.

3.1 Motivations and basics of composite laminate theory

We briefly recall what are the pitfalls associated to laminate composite optimization. Subject has been quite well investigated (see for instance (Gürdal *et al.* 1999)) and heuristics to tackle with the discreteness of the variables have been developed. This following section briefly introduces this subject.

3.1.1 Introduction

Composite structural optimization for thin-walled structures often exhibits a large computational time due to repetitive stability analysis. For composite structure made of thin plates or thin shallow shells (such as aircraft fuselage), buckling computation is of primary importance since it is one of the critical sizing constraints when minimizing

the weight. This computational burden becomes even critical when we address laminated composites and when the stacking sequence is to be optimized. In such cases, one often goes to population-based methods (such as genetic algorithms) to address the discreteness of the design variables. Such methods require many evaluations to ensure that a reasonably good optimum was found. To circumvent such issues, one classical strategy uses a continuous representation of the in-plane and out-of-plane behaviors of the stacking sequence by means of either lamination parameters, polar invariant representation or directly A and D tensors terms. This offers a way to treat the optimization (for instance maximizing the buckling critical load of a laminate plate under a given loading) as a continuous one, reducing the number of evaluations needed to converge whenever the optimization problem is well posed (e.g convex). A good property of lamination parameters is that most classical objective functions have interesting optimization properties when evaluated over the lamination space which happens to be convex. However, this continuous relaxation has two majors drawbacks. First when a continuous optimum is found, it usually does not match a discrete solution, there is no practical stacking sequence that gives these optimal lamination parameters. A new discrete optimization step is to be performed to identify a correct discrete solution and there is no evidence that the discrete optimum found by this step is the real discrete optimum. The other important drawback is that not all sizing constraints can be computed on the basis of lamination parameters. Some optimization constraints do need the real stacking sequence to be computed: mechanical constraints (first-ply failure...), feasibility constraints (contiguity of same orientations to avoid matrix cracking for instance). However, lamination parameters do provide a practical continuous representation of in-plane and out-of-plane behavior making possible to build approximation models of the expensive optimization constraints, such as buckling computations. Even though continuous optimization over lamination is not performed, such an approximation model allows to perform discrete optimizations in a much more efficient way.

Since their first application to the design of composite in the seminal work of Miki ((Miki 1983), (Miki & Sugiyama 1991)), where the author derive a graphical procedure to optimize stiffness properties, lamination parameters have been extensively studied and covered as a practical and convenient tool to optimize laminated composites. Early work by Miki allowed to bound the lamination space. Much work

was also done in describing the dependence of the buckling critical load in the lamination space: Grenested ((Grenestedt 1989), (Grenestedt 1991), (Grenestedt 1990)) or Walker (Walker *et al.* 1996) investigated the influence of lamination parameters for the buckling critical load in different cases (pure shear loading, non-orthotropic laminates,...) through different approximation methods: orthotropic closed-form expression, finite strip method, Rayleigh-Ritz method... Grenested found out that non-orthotropy decreases the buckling critical load for uniaxial compression loading and in many situations the optimum in the lamination space belongs to the boundary previously described by Miki, corresponding to angle-ply laminate $[+\theta/-\theta]$ with continuous angle θ. However practical stacking sequences that satisfy manufacturability constraints are not orthotropic nor angle-ply stacking sequences. A lot of work was also done in applying directly the optimization problem in the lamination space and then get from the continuous optimum a discrete solution usually by means of genetic algorithms (Fukunaga *et al.* 1995), (Harrison *et al.* 1995). Lamination parameters associated to real practical stacking sequence of orientations $[0°/45°/-45°/90°]$ exhibit a fractal structure that was exploited in (Terada *et al.* 2001) to derive a fractal branch-and-bound method. The precise description of the feasible set of lamination parameters and more specifically the coupling between in-plane lamination parameters and out-of-plane lamination parameters was solved in (Diaconu *et al.* 2002), (Bloomfield *et al.* 2009), resulting, amongst many things, into so-called compatibility equations that constrain the feasible space for out-of-plane lamination parameters based on the values of in-plane lamination parameters. These compatibility equations describe a convex feasible set and can be directly integrated into an optimization (Herencia *et al.* 2008b). Regarding the area of building approximations of the buckling critical load over lamination parameter, much work was done with the help of response surfaces (polynomial regression of degree 2) (Todoroki *et al.* 2003) (Abu-Odeh & Jones 1998) (Harrison *et al.* 1995), Taylor-based first-order linear approximation (Herencia *et al.* 2008b), adaptive response surface (Todoroki & Sasai 2002). The non-differentiability of the buckling critical load with respect to lamination parameters, which was already noticed in lot of work (see (Rousselet & Chenais 1990) for general results on buckling critical load differentiability not specifically devoted to composite) does not seem to have been considered when building approximation models. Regarding potential property of the buckling critical load well suited for optimization,

concavity was cited in several references (Autio 2001), but to the authors' knowledge, it has never been proved on a sound mathematical basis for the general case (arbitrary geometry, non-orthotropic laminate).

This chapter aims at studying numerically some properties of the buckling critical load that will ease the construction of accurate approximations to be used within an optimization process. The approximation of the buckling critical load over lamination parameters is not restricted to continuous optimization over lamination parameters, since such an approximation can be used when the stacking is directly addressed as optimization variables. In particular, we show that the buckling critical load is concave over lamination parameters. We also give numerical evidence of the piecewise behavior of the buckling critical load depending on the region of the input space. Based on this knowledge we will develop in the next chapter an original strategy to build piecewise approximation models. The behavior of the buckling critical load both over lamination parameters and varying loading conditions is covered.

3.1.2 Composite Laminate Theory, lamination parameters and feasible stacking sequences

We present in this section the well-known lamination parameters. We start with their definition in the case of a general laminate and we also briefly recall Composite Laminate Theory. We describe the structure of the lamination space for feasible stacking sequences for conventional laminates of orientation $[0/45/-45/90]$ and describe the compatibility equations.

3.1.2.1 Composite Laminate Theory for plates

The purpose of this section is not to recall the whole theory of laminates that can be found for instance in (Reddy 2004) or (Daniel 1989) but simply to recall essential hypothesis and definitions in the context of laminated thin plate theory that is the basis of the original work presented in this chapter. We first start with the description of a composite plate.

The composite plates considered in this chapter are rectangular domains of dimensions $a \times b \times h$ and are made of a number of perfectly bonded layers. These layers have

different orientations and each layer is considered as a homogeneous anisotropic material. Moreover the thickness of the laminate is assumed to be constant. There exist several mathematical models used to determine stress and strain in plates (leading of course to very different Finite Element Methods, see for instance the valuable reference on FEM (Felippa 2000))

- **Kirchhoff-Love theory** for thin plate, which leads to Classical Plate Theory
- **Mindlin-Reissner theory** for thicker plates (but still moderately thick)
- **Three-dimensional elasticity** for very thick plates

One important feature of the first two models is that they provide a two-dimensional modeling of the plate. They get rid of the thickness by simple assumption over the behavior of mechanical responses through the thickness. This leads to less computations.

The classification between these different thicknesses is a bit fuzzy and also strongly depends on the phenomena we want to predict. In (Ventsel & Krauthammer 2001), the authors sort plates with the ratio $\frac{L}{h}$ where L is a typical characteristic length of the plate with

- ratios s.t $L/h > 100$, the plate is very thin and called **membrane**. Such a plate mostly carries lateral loads by axial tensile and shear forces that remain in-plane and act on the middle surface. These forces are then referred to as **membrane forces**.
- ratios s.t $10 < L/h < 100$, the plate is called **thin plate**. Unlike membranes, thin plates are not devoid of flexural rigidity. Depending on the relative part of membrane forces w.r.t bending forces, the plate is referred to either **stiff plate** (mostly carrying loads by bending and twisting moments) or **flexible plate** (mostly carrying loads by a combination of bending, twisting moments with membrane forces). In (Ventsel & Krauthammer 2001) the distinction between the two is made over the ratio w_{max}/h where w_{max} is the maximum transverse displacement.
- ratios s.t $L/h < 10$, plate is called **thick plate**. Such thicknesses usually imply three dimensional elasticity modeling and no specific mechanical assumption is made over displacement, strain or stress.

It is worth noting that the classification is not straight and highly depends on boundary conditions, type of loading, material and also the accuracy we want to reach in the prediction, what stage of the sizing we are in... For composite fuselage structures for instance, we can take L as b the typical stringer pitch for it is smaller than a the frame pitch. Typical dimensions for these quantities are $b = 150$mm to 200mm and $h = 1.6$mm to 5mm leading to ratios in between 37 and 125. We see that plates are then considered as **thin plates** but can also be **membranes**. Typical plates used in aerospace are **flexible plates** because of their attractive weight-to-load ratio, regarding mathematical modeling, thin plates are often analyzed though classical plate theory based on Kirchhoff-Love theory at a preliminary sizing level. For more advanced sizing stages, one often goes to three-dimensional elasticity to accurately predict specific phenomena. In the sequel, we only focus on Kirchhoff-Love theory.

Kirchhoff-Love theory was first carried out by Love in the late 19^{th} century and latter extended by Kirchhoff based on the classical **Kirchhoff's hypotheses** which are

(i) straight lines normal to the mid-surface remain straight after deformation

(ii) straight lines normal to the mid-surface remain normal after deformation

It is supplemented by the following kinematic assumption

(iii) thickness does not change during deformation

Note that Mindlin-Reissner theory generalizes the former theory by assuming hypotheses (i) and (iii) but no more (ii). Denote u, v and w the displacement components of the plates, under Kirchhoff hypotheses, u and v can be expressed

$$u(x, y, z) = u_0(x, y) - z\frac{\partial w}{\partial x} \tag{3.1}$$

and

$$v(x, y, z) = v_0(x, y) - z\frac{\partial w}{\partial y} \tag{3.2}$$

where u_0 and v_0 are mid-plane displacements. Using the strain-displacement equations of classical plane elasticity theory,

$$\varepsilon_x = \frac{\partial u}{\partial x}, \quad \varepsilon_y = \frac{\partial v}{\partial y}, \quad , \gamma_{xy} = \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \tag{3.3}$$

where ε_x, ε_y are in-plane normal strains and γ_{xy} is the in-plane shear strain. Now we can define the mid-plane curvature during deformation

$$\kappa_x = \frac{\partial^2 w}{\partial x^2}, \quad \kappa_y = \frac{\partial^2 w}{\partial y^2}, \quad \kappa_{xy} = 2\frac{\partial^2 w}{\partial x \partial y} \tag{3.4}$$

Using inserting this in Eq. (3.1) and Eq. (3.2)

$$\varepsilon_x = \varepsilon_x^0 - z\kappa_x, \quad \varepsilon_y = \varepsilon_y^0 - z\kappa_y, \quad \gamma_{xy} = \gamma_{xy}^0 - z\kappa_{xy} \tag{3.5}$$

where ε_x^0, ε_y^0 and γ_{xy}^0 are mid-plane normal and shear strains. In-plane forces resultants N_x, N_y and N_{xy} are obtained by integrating stress components over the thickness, with σ_x, σ_y are in-plane normal stress components and τ_{xy} is the in-plane shear stress

$$N_x = \int_{-h/2}^{h/2} \sigma_x dz \tag{3.6}$$

$$N_y = \int_{-h/2}^{h/2} \sigma_y dz \tag{3.7}$$

and

$$N_{xy} = \int_{-h/2}^{h/2} \tau_{xy} dz \tag{3.8}$$

Moment resultants that are bending moments M_x and M_y and twisting moment M_{xy} are obtained by

$$M_x = \int_{-h/2}^{h/2} \sigma_x z dz \tag{3.9}$$

$$M_y = \int_{-h/2}^{h/2} \sigma_y z dz \tag{3.10}$$

and

$$M_{xy} = \int_{-h/2}^{h/2} \tau_{xy} z dz \tag{3.11}$$

Now we use the constitutive equation (strain-stress relation) of each layer (i) of orientation angle $\theta^{(i)}$ with the principal axis. In the $\theta^{(i)}$ reference axis with the assumption of plane stress, this constitutive equation is

$$\begin{pmatrix} \sigma_x^{(i)} \\ \sigma_y^{(i)} \\ \tau_{xy}^{(i)} \end{pmatrix} = \begin{pmatrix} Q_{11} & Q_{12} & 0 \\ Q_{12} & Q_{22} & 0 \\ 0 & 0 & Q_{66} \end{pmatrix} \begin{pmatrix} \varepsilon_x^{(i)} \\ \varepsilon_y^{(i)} \\ \gamma_{xy}^{(i)} \end{pmatrix} \tag{3.12}$$

where the Q_{ij} terms are the **reduced stiffness's**, that are obtained with E_1, E_2, ν_{12} and G_{12} the characteristic moduli of the layer (i) through

$$Q_{11} = \frac{E_1}{1-\nu_{12}\nu_{21}}, \quad Q_{22} = \frac{E_2}{1-\nu_{12}\nu_{21}}, \quad Q_{12} = \frac{\nu_{12}E_2}{1-\nu_{12}\nu_{21}} \tag{3.13}$$

and

$$Q_{66} = G_{12}. \tag{3.14}$$

Finally using this constitutive equation, we integrate the stress-strain relation piecewise, layer by layer, and changing the references from $\theta^{(i)}$ references axis to the principal axis through transformation matrices, we get to the **constitutive equation of the laminate** (details of the computations can be found in any references book on laminates such as (Ashton & Whitney 1970), (Jones 1999), (Daniel 1989) or (Lekhnitskii 1968))

$$\begin{pmatrix} N_x \\ N_y \\ N_{xy} \\ M_x \\ M_y \\ M_{xy} \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} & A_{16} & B_{11} & B_{12} & B_{16} \\ A_{12} & A_{22} & A_{26} & B_{12} & B_{22} & B_{26} \\ A_{16} & A_{26} & A_{66} & B_{16} & B_{26} & B_{66} \\ B_{11} & B_{12} & B_{16} & D_{11} & D_{12} & D_{16} \\ B_{12} & B_{22} & B_{26} & D_{12} & D_{22} & D_{26} \\ B_{16} & B_{26} & B_{66} & D_{16} & D_{26} & D_{66} \end{pmatrix} \begin{pmatrix} \varepsilon_x^0 \\ \varepsilon_y^0 \\ \gamma_{xy}^0 \\ -\kappa_x \\ -\kappa_y \\ -\kappa_{xy} \end{pmatrix} \tag{3.15}$$

where the A, B and D sub-matrices are positive definite matrices, in the sequel we will denote them A, B and D **tensors**. From the very equations, we clearly see that this relation highlight the complicated mechanical behavior of the laminate, indeed

- the A tensor corresponds to **in-plane properties**. It is then often called the **membrane tensor**. The A_{16} and A_{26} are extension-shear coupling terms. At the whole structure level (assembly of several plates), this coupling corresponds to the coupling between bending and twisting. One often want to design laminate such that this coupling vanishes, this is ensured whenever the laminate is balanced. However, for **aeroelastic tailoring**, this coupling can be useful to design structures against flutter. Indeed, flutter corresponds to structural bending and twisting modes that are too close. Natural frequencies of bending and twisting can be modified by the flows and become close, leading to a collapse of the aircraft. In our application however, we will impose that these terms are zero.

- the B tensor causes **coupling between bending and stretching** in the plate during transverse displacements. For a symmetric laminate, this B tensor vanishes. Since we only work with symmetric laminates, we will not consider this

tensor anymore. One important consequence is that in-plane behavior does not depend on the bending behavior. It is then independent and a membrane analysis of the whole structure (by FEA for instance) can be performed without the knowledge of the D tensor. It is worth noting that, as shown in (Gürdal *et al.* 1999), symmetry is not a necessary condition for the B tensor to be zero, there exist non-symmetric laminate for which the B tensor is zero.

- the D tensor corresponds to **out-of-plane properties**. It is the tensor-form laminate generalization of the bending stiffness for homogeneous isotropic plate defined as

$$D = \frac{Eh^3}{12(1-\nu^2)} \tag{3.16}$$

The main difference is that it weights some orientations over others (D_{11} is the bending stiffness in the x-axis for instance) and more important the D_{16} and D_{26} term cause coupling between bending and twisting at plate level. These terms are then referred to as **bending-twisting coefficients**, we will see in the sequel that they usually complicate buckling analysis and often degrade the structure against buckling (not in case of shear loading). Laminates with zero bending-twisting terms are called **orthotropic** and are usually easier to handle (closed-form expressions for buckling for instance for simple domains). Unlike A_{16} and A_{26} terms, making bending-twisting terms zero is not obvious. Cross-ply laminates for instance $[+\theta/-\theta]$ are orthotropic, see other examples in (Gürdal *et al.* 1999).

We now turn to the definition of lamination parameters that allows the computations of the A and D tensors.

3.1.2.2 Lamination parameters

As outlined in the introduction, lamination parameters were introduced in the early seventies by Miki and since have been quite used as practical design variables since lamination parameters allows to decompose the A, B and D tensor terms into two distinct types of variables: **material-dependent variables** which are the **Tsai-Pagano parameters** and **stacking-sequence dependent parameters** which are the **lamination parameters**.

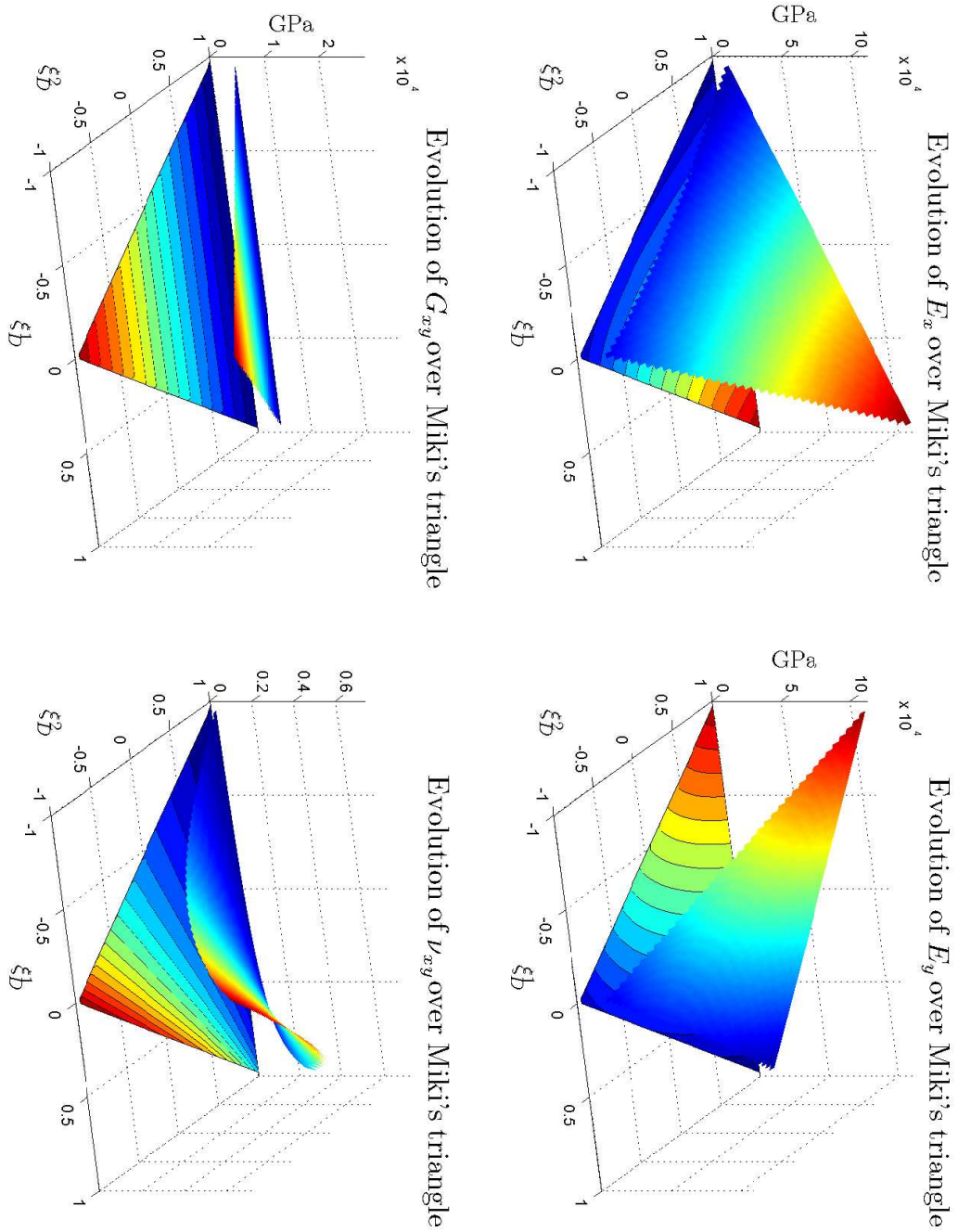


Figure 3.1: Dependence of the laminate stiffness moduli over Miki's triangle

Consider a composite of given material characteristics E_1, E_2, G_{12} and ν_{12}. Note Q_{11}, Q_{22}, Q_{12} and Q_{66} the reduced stiffness's. The Tsai-Pagano parameters $(U_i)_{i=1...6}$ (or material invariants) are defined as

$$\begin{pmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ U_5 \end{pmatrix} = \begin{pmatrix} 3/8 & 3/8 & 1/4 & 1/2 \\ 1/2 & -1/2 & 0 & 0 \\ 1/8 & 1/8 & -1/4 & -1/2 \\ 1/8 & 1/8 & 3/4 & -1/2 \\ 1/8 & 1/8 & -1/4 & 1/2 \end{pmatrix} \begin{pmatrix} Q_{11} \\ Q_{22} \\ Q_{12} \\ Q_{66} \end{pmatrix} \tag{3.17}$$

The **in-plane (or stretching) lamination parameters** are defined as

$$\xi^A_{\{1,2,3,4\}} = \frac{1}{h}\int_{-h/2}^{h/2} \{\cos(2\theta(z)), \cos(4\theta(z)), \sin(2\theta(z)), \sin(4\theta(z))\} dz \tag{3.18}$$

Now the A tensor is expressed as an affine function of the in-plane lamination parameters as follows

$$\begin{aligned} A_{11} &= h(U_1 + U_2\xi^A_1 + U_3\xi^A_2) && (3.19)\\ A_{22} &= h(U_1 - U_2\xi^A_1 + U_3\xi^A_2) && (3.20)\\ A_{12} &= h(U_4 - U_3\xi^A_2) && (3.21)\\ A_{66} &= h(U_5 - U_3\xi^A_2) && (3.22)\\ A_{16} &= h(U_2\frac{\xi^A_3}{2} + U_3\xi^A_4) && (3.23)\\ A_{26} &= h(U_2\frac{\xi^A_3}{2} + U_3\xi^A_4) && (3.24)\\ & && (3.25) \end{aligned}$$

Note that $\xi^A_4 = 0$ whenever the angles are 0, 45 and 90.

The **out-of-plane (or bending) lamination parameters** are defined as

$$\xi^{\{1,2,3,4\}}_D = \frac{12}{h^3}\int_{-h/2}^{h/2} \{\cos(2\theta(z)), \cos(4\theta(z)), \sin(2\theta(z)), \sin(4\theta(z))\} z^2 dz \tag{3.26}$$

The same way the D tensor is expressed as an affine function of the out-of-plane lami-

nation parameters as follows

$$D_{11} = \frac{h^3}{12}(U_1 + U_2\xi_1^D + U_3\xi_2^D) \tag{3.27}$$

$$D_{22} = \frac{h^3}{12}(U_1 - U_2\xi_1^D + U_3\xi_2^D) \tag{3.28}$$

$$D_{12} = \frac{h^3}{12}(U_4 - U_3\xi_2^D) \tag{3.29}$$

$$D_{66} = \frac{h^3}{12}(U_5 - U_3\xi_2^D) \tag{3.30}$$

$$D_{16} = \frac{h^3}{12}(U_2\frac{\xi_3^D}{2} + U_3\xi_4^D) \tag{3.31}$$

$$D_{26} = \frac{h^3}{12}(U_2\frac{\xi_3^D}{2} + U_3\xi_4^D) \tag{3.32}$$

$$\tag{3.33}$$

Note that $\xi_4^D = 0$ whenever the angles are 0, 45 and 90.

Note that we do not give the definition (similar except that there is not constant term in the definition of B) for the ξ^B and the B for we assumed that all the stacking sequences we consider are symmetric, hence the ξ^B's are zero. It can be found in many articles, such as (Bloomfield *et al.* 2009), (Herencia *et al.* 2008b), (Setoodeh *et al.* 2006).

One valuable feature of lamination parameters is that first they allow to tailor the stacking sequence dependence when used within an optimization process. They are a compact representation of the quantities of interest, for instance regarding in-plane properties depending only on A terms, provided we work only with balanced stacking sequences, the A tensor only depends on two variables $\xi_{1,2}^A$ and therefore the quantities of interest can be represented as a surface over the design space of lamination parameters. For instance, one of the interesting optimization features of lamination parameters are the simple dependence of **laminate stiffness moduli (a.k.a laminate engineer constants**) defined as compliance matrix terms (inverse of in-plane stiffness tensor A)

$$S = \begin{pmatrix} A_{11} & A_{12} & 0 \\ A_{12} & A_{22} & 0 \\ 0 & 0 & A_{66} \end{pmatrix}^{-1} \tag{3.34}$$

The laminate engineer constants are simply defined as

$$E_x = \frac{1}{S_{11}} = \frac{A_{11}A_{22} - A_{12}^2}{A_{22}} \tag{3.35}$$

and

$$E_y = \frac{1}{S_{22}} = \frac{A_{11}A_{22} - A_{12}^2}{A_{11}} \tag{3.36}$$

and

$$G_{xy} = \frac{1}{S_{66}} = A_{66} \tag{3.37}$$

and

$$E_y = \frac{S_{12}}{S_{11}} = -\frac{A_{12}}{A_{11}} \tag{3.38}$$

These quantities depend simply on lamination parameters, as it is illustrated in Fig. 3.1. This simple dependency is the basis of the graphical procedure described in the work of Miki for designing laminate under stiffness constraint (Miki 1983).

3.1.2.3 Compatibility equations

The feasible space (w.r.t positivity of A and D tensor) has been investigated from the early work of Miki (Miki & Sugiyama 1991). Over the past ten years, much work has been done in this area by Diaconu, Weaver et al. see for instance (Diaconu *et al.* 2002), (Bloomfield *et al.* 2009), resulting on a convex subset that can be described by means of **compatibility equations**. We will not describe here the quite technical way to obtain these compatibility equations and we simply give the equation over lamination parameters that are non zero in our study $\xi^A_{1,2}$ and $\xi^D_{1,2,3}$, since $\xi^A_3 = 0$ because of the

balanced property.

$$2|\xi_1^{A,D}| + \xi_2^{A,D} - 1 \leqslant 0 \tag{3.39}$$

$$2|\xi_3^{A,D}| + \xi_2^{A,D} - 1 \leqslant 0 \tag{3.40}$$

$$(\xi_i^A - 1)^4 - 4(\xi_i^D - 1)(\xi_i^A - 1) \leqslant 0 \quad i = 1, 2 \tag{3.41}$$

$$(\xi_i^A + 1)^4 - 4(\xi_i^D + 1)(\xi_i^A + 1) \leqslant 0 \quad i = 1, 2 \tag{3.42}$$

$$(2\xi_1^A - \xi_2^A - 1)^4 - 16(2\xi_1^D - \xi_2^D - 1)(2\xi_A - \xi_2^A - 1) \leqslant 0 \tag{3.43}$$

$$(2\xi_1^A + \xi_2^A + 1)^4 - 16(2\xi_1^D + \xi_2^D + 1)(2\xi_1^A + \xi_2^A + 1) \leqslant 0 \tag{3.44}$$

$$(2\xi_1^A - \xi_2^A + 3)^4 - 16(2\xi_1^D - \xi_2^D + 3)(2\xi_1^A - \xi_2^A + 3) \leqslant 0 \tag{3.45}$$

$$(2\xi_1^A + \xi_2^A - 3)^4 - 16(2\xi_1^D + \xi_2^D - 3)(2\xi_A + \xi_2^A - 3) \leqslant 0 \tag{3.46}$$

$$(2\xi_1^A - \xi_2^A + 1)^4 - 16(2\xi_1^D - \xi_2^D + 1)(2\xi_1^A - \xi_2^A + 1) \leqslant 0 \tag{3.47}$$

$$(2\xi_1^A + \xi_2^A - 1)^4 - 16(2\xi_1^D + \xi_2^D - 1)(2\xi_1^A + \xi_2^A - 1) \leqslant 0 \tag{3.48}$$

$$(2\xi_1^A - \xi_2^A - 3)^4 - 16(2\xi_1^D - \xi_2^D - 3)(2\xi_1^A - \xi_2^A - 3) \leqslant 0 \tag{3.49}$$

$$(2\xi_1^A + \xi_2^A + 3)^4 - 16(2\xi_1^D + \xi_2^D + 3)(2\xi_1^A + \xi_2^A + 3) \leqslant 0 \tag{3.50}$$

$$(\xi_1^A - \xi_2^A - 1)^4 - 4(\xi_1^D - \xi_2^D - 1)(\xi_1^A - \xi_2^A - 1) \leqslant 0 \tag{3.51}$$

$$(\xi_1^A + \xi_2^A + 1)^4 - 4(\xi_1^D + \xi_2^D + 1)(\xi_1^A + \xi_2^A + 1) \leqslant 0 \tag{3.52}$$

$$(\xi_1^A - \xi_2^A + 1)^4 - 4(\xi_1^D - \xi_2^D + 1)(\xi_1^A - \xi_2^A + 1) \leqslant 0 \tag{3.53}$$

$$(\xi_1^A + \xi_2^A - 1)^4 - 4(\xi_1^D + \xi_2^D - 1)(\xi_1^A + \xi_2^A - 1) \leqslant 0 \tag{3.54}$$

3.1.2.4 Feasible stacking sequences

We are interested here in conventional laminates of fiber orientations $[0°/45°/-45°/90°]$. We used the following common feasibility rules that both come from mechanical properties but also manufacturability properties

- **Symmetric laminates**. This ensures that the B tensor vanishes. Note that there exist laminates with $B = 0$ that are not symmetric.
- **Balanced laminates**. The number of 45 layers is equal to the number of -45 layers. This makes the A_{16} and A_{26} terms of the A tensor vanish.
- **Contiguity rule**. No more than 4 successive layers of the same orientation.

- **Disorientation rule**. No more than 45 of angle difference between 2 successive layers. In particular $[45/-45]$ is not allowed although many reference in the literature on composite design allow it.

- **Percentages rule**. Each orientation should be represented in the laminate. We took 8% as a minimum percentage of each orientation.

Based on these rules, the number of feasible stacking sequences remain relatively for thin laminates. The number of feasible stacking sequences based on these rules for a 2×10 layers symmetric laminate is for instance 7032. We found numerically that we could very easily exhaust all feasible stacking sequences based on vector-oriented programming language (such as Matlab) up to laminates of 34 layers. Our concern here is not in finding such feasible stackings but it should be noted that for thicker laminates the exhaustive search might take excessive memory resources unless proportions are given and a discrete optimization is preferable. However, for relatively thin laminates (such as the ones used for aircraft fuselage or wing) we consider that we have all feasible stacking sequences and that an exact optimization can be performed by listing all the feasible laminates. We depicted Fig. 3.2 the logarithm of the number of feasible stacking sequences from 4 to 34 plies. This number $N_{feas}(n)$ where n stands for the number of plies is approximated by

$$N_{feas} = K1.72^n \tag{3.55}$$

where $K = 0.12$.

We computed the 5 nonzero lamination parameters for all feasible stacking sequences from 8 to 32 plies. We obtained only 159 different values of the couple (ξ_1^A, ξ_2^A). We also depicted the number of different stacking sequences associated with each value of the couple (ξ_A^1, ξ_A^2) in Fig. 3.4. We also depicted the distribution of feasible ξ_1^D, ξ_2^D and ξ_3^D Fig. 3.5 and 3.6.

3.2 Behavior of the buckling critical factor

In this section, we first recall some basic facts on buckling, including the definition of the buckling critical factor for thin composite plates through variational formulation. Based on this definition we show that this buckling critical factor is concave over the

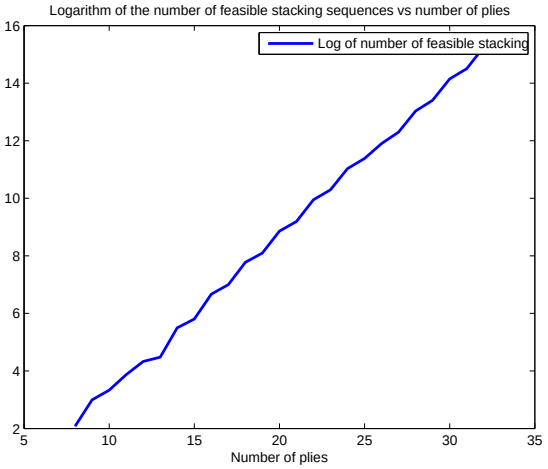


Figure 3.2: Natural logarithm of the number of feasible stacking sequences

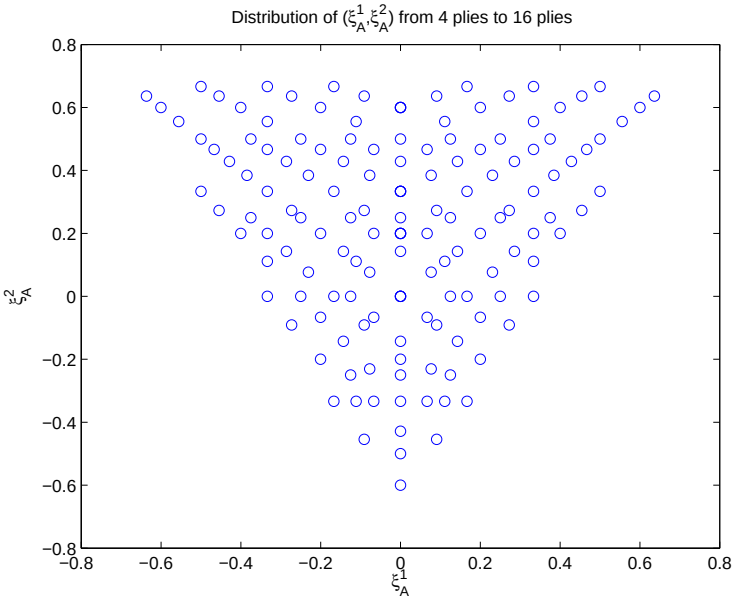


Figure 3.3: Values of the (ξ_A^1, ξ_A^2) in-plane lamination parameters

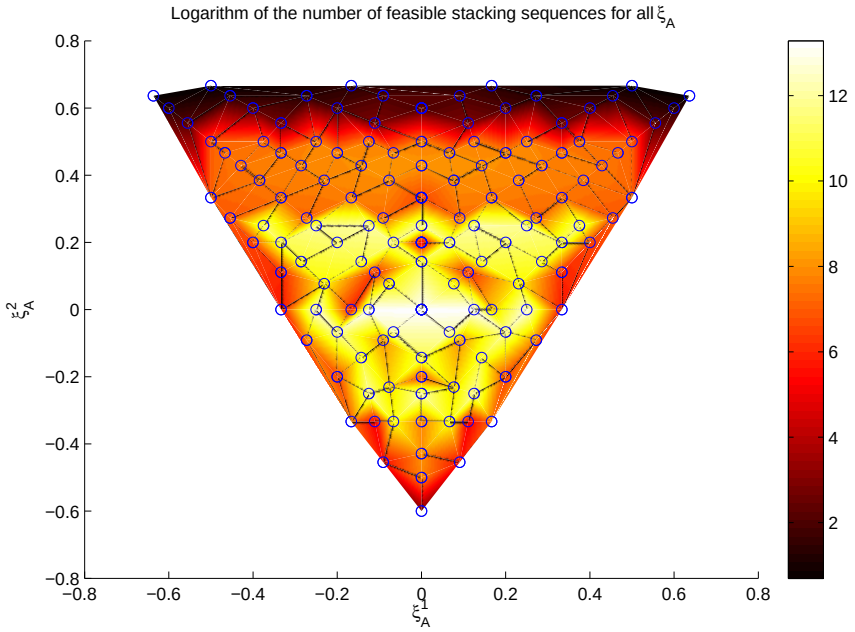


Figure 3.4: Logarithm of the number of feasible stacking sequences for each value of the (ξ_A^1, ξ_A^2) in-plane lamination parameters

well-known lamination parameters. We then depict two interesting aspects: firstly, the typical behavior of the buckling reserve factor when the stacking sequence varies and secondly the typical behavior of its reciprocal when the loading conditions vary.

3.2.1 Generalities on buckling

The buckling phenomenon for thin plates and shells has been extensively studied since the beginning of the 20^{th} century. It consists in a large transverse deflection phenomenon for thin-walled structures under compressive and shear in-plane loading. It can not be described in the frame of small perturbations. We depicted an example of shear buckling Fig. 3.7.

The first rigorous equations for large deflections of thin plates were written down by von Karman in 1910. In their first derivation for isotropic homogeneous structures, these equations were a system of two nonlinear partial differential equations of fourth order. As outlined in (Antman 2006), the mathematical analysis used to solve these equations involved development of bifurcation theory, as it can be found in

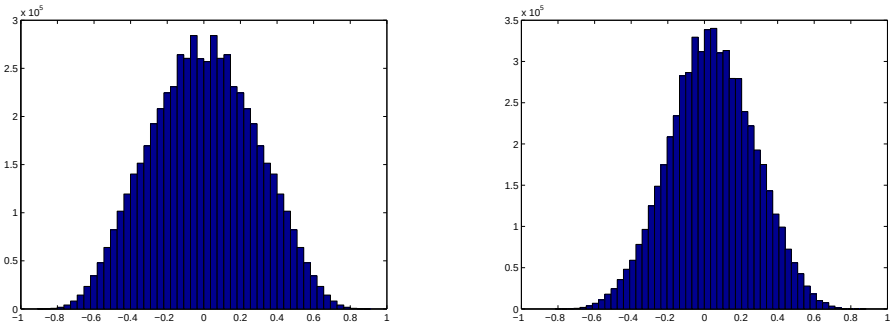


Figure 3.5: Distribution of feasible ξ_1^D and ξ_2^D

(Bloom & Coffin 2000), (Lebedev & Vorovich 2003) and (Antman & Renardy 2005). A very brief introduction to bifurcation theory not specifically devoted to buckling can be found in (Stakgold 1971). The major idea is that bifurcation points of the nonlinear pde's (or stability points where the trivial solution $w = 0$ ceases to be unique) are to be found amongst the points where the linearized operator is not invertible thanks to implicit functions theorem. The topological Leray-Schauder degree theory offers the other way around. Namely, the points where the linearized operator is not invertible (precisely the eigenelements of the linearized operator) are actually bifurcation points of the original nonlinear pde system. We will not delve into the mathematical theory but this brief explanation means that buckling is a nonlinear phenomenon that is often partially solved through a linear eigenvalue problem. It is worth noting that the linearized buckling can not describe the behavior of the material after buckling, it only predicts the onset of buckling. Being an eigenfunction, the buckled transverse displacement is known up to a multiplicative constant, which means that no maximal transverse displacement can be computed through linear buckling. To study the behavior of the plate or the shells after buckling one should go to nonlinear analysis, the first work on post-buckling can be found in the seminal paper (Koiter 1962). Our concern here is on the prediction of buckling, we will therefore study buckling of laminated composite plates into the framework of linear buckling. Under Kirchhoff hypotheses, one can show (see (Turvey 1995)) that the equation for buckling of a simply supported composite plate leads to the following eigenvalue problem. The **buckling critical factor** is defined as

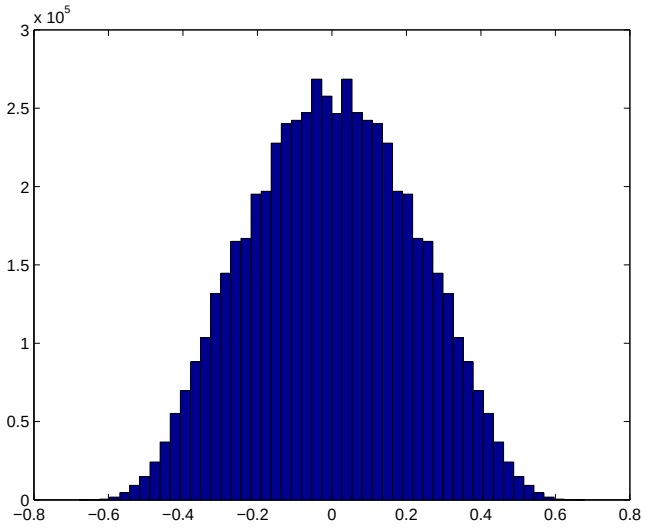


Figure 3.6: Distribution of feasible ξ_3^D

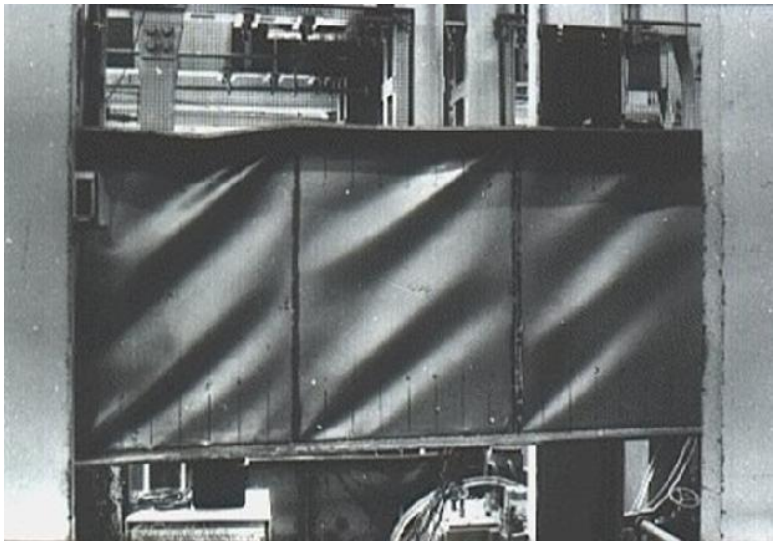


Figure 3.7: Examples of shear buckling

the smallest positive eigenvalue λ_1 of

$$D_{11}\frac{\partial^4 w}{\partial x^4} + D_{22}\frac{\partial^4 w}{\partial y^4} + 2(D_{12}+2D_{66})\frac{\partial^4 w}{\partial x^2 \partial y^2} + 4D_{16}\frac{\partial^4 w}{\partial x^3 \partial y} + 4D_{26}\frac{\partial^4 w}{\partial y^3 \partial x} = \lambda_1(N_x\frac{\partial^2 w}{\partial x^2} + N_y\frac{\partial^2 w}{\partial y^2} + N_{xy}\frac{\partial^2 w}{\partial x \partial y}) \quad (3.56)$$

where w is the transverse displacement, with $w = 0$ on $\partial\Omega$ and $\Delta w = 0$ on $\partial\Omega$. N_x, N_y and N_{xy} are the loads per unit length respectively for longitudinal, lateral and shear loading, and equivalently the **buckling critical load** N_{cr} is defined as $N_{cr} = (\lambda_{cr}(N_x, N_y, N_{xy})$. Note that equation (3.56) is valid for any geometry Ω. For the numerical experiments in this chapter, we only focus on rectangular plates but, in general, we can already see that buckling essentially depends on three different types of parameters

- Ω: the domain where the buckling equation 3.56 is defined. Dependence of eigenvalue problems to the geometry is quite complicated. For rectangular plates $\Omega = [0, a] \times [0, b]$, the eigenvalues depend only on the aspect ratio $\frac{a}{b}$.
- D tensor: the idealized buckling problem studied in shape optimization (in (Ashbaugh 1999) for instance) only defines bi-Laplacian and Laplacian operators. This corresponds to buckling for isotropic homogeneous material under biaxial compression of the same magnitude ($N_x = N_y < 0$). For composite, the D tensor makes the buckling equation somewhat more complex since it weights the different derivatives and we clearly see that the D_{16} and D_{26} (bending-twisting terms) are not part of the bilaplacian.
- $N = (N_x, N_y, N_{xy})$: the loading conditions. When $N_{xy} = 0$ and $N_x = N_y < 0$, note that the right-hand differential operator boils down to a laplacian.

Buckling problems of this type can be solved exactly in a few cases with simple geometry (rectangle, disk,...) and simple operators (bi-Laplacian which corresponds to orthotropy, laplacian that corresponds to uniform biaxal compression with no shear), many closed-form expressions can be found in (Lekhnitskii 1968). In the orthotropic case, where $D_{16} = D_{26} = 0$, buckling of a rectangular composite plate of dimension $[0, a] \times [0, b]$, under biaxial loading (no shear), the buckling critical factor is

$$\lambda_1 = \min_{m,n\in\mathbb{N}^2} \frac{\pi^2(D_{11}m^4 + 2(D_{12}+2D_{66})m^2n^2R^2 + D_{22}n^4R^4)}{a^2(N_xm^2 + N_yn^2R^2)} \quad (3.57)$$

where $R = a/b$ is the aspect ratio and n (resp. m) is the number of half-waves of the transverse displacement along x direction (resp. y direction). When no closed-form expression can be derived, more complex geometry, non-orthotropic laminates, shear loading, approximation methods are to be used. Amongst the classical methods, we have

- Rayleigh-Ritz method: historically, this was the first method develop to tackle with eigenvalue problem arising in vibration and buckling. This yields to solve an eigenvalue problem over a finite-dimensional space which is the span of natural eigenfunctions for a simple operator, e.g $\sin(\frac{i\pi x}{a})\sin(\frac{j\pi y}{b})$ for a rectangular plate. The idea is therefore to write the variational characterization of eigenvalue as an optimization problem solved in this finite-dimensional space. The Rayleigh-Ritz can be thought as a spectral method (see appendix). Unlike finite element methods, matrices are usually dense. General introduction can be found in (Courant 1994), (Akulenko & Nesterov 2005) and (Turvey 1995) for composite buckling

- Finite strip method (see (Turvey 1995))

- Finite elements method

In general, the approximation finite-dimensional space $\mathcal{V}_h$ is included in the $\mathcal{V}$ Hilbert space where the solution w belongs to (conformal approximation). Under this usual property, the exact eigenvalues can be shown to be approximated by above through these different approximation methods. In this chapter, most of the numerical computations were done with a Rayleigh-Ritz method.

A few basic facts can be noted in (3.57) regarding the dependence of λ_1 with respect to stacking sequence and loading conditions

- For fixed loading conditions, buckling critical factor is concave over the D tensor variables, being the minimum of concave functions (linear). We will see in the sequel that this can be established rigorously in the general non-orthotropic case, for any loading and geometry Ω.

- For fixed stacking sequence, hence fixed D tensor, the reciprocal of the buckling critical factor is piecewise linear. We will see that numerically it seems to be a good approximation for non-orthotropic laminates under biaxial loading.

- For fixed loading conditions, the buckling critical reserve factor is piecewise linear on D hence on lamination parameters. In the general case (combined loading conditions) this does not seem to be still valid, however, simple piecewise quadratic models over lamination parameters are shown to be very accurate to predict the buckling critical reserve factor.

- Define formally a buckling critical factor function

$$\begin{array}{rll} \lambda_{cr}: & S_3(\mathbb{R})^{++} \times \mathbb{R}^3 & \mapsto \mathbb{R}^+ \\ & (D, N_x, N_y, N_{xy}) & \mapsto \lambda_1 = \lambda_{cr}(D, N_x, N_y, N_{xy}) \end{array} \tag{3.58}$$

  where D the out-of-plane (symmetric positive definite) tensor in $S_3(\mathbb{R})^{++}$ the convex cone of symmetric definite positive tensors and $N = (N_x, N_y, N_{xy})$ the loading conditions. We see that in the orthotropic case, λ_{cr} does have some partial homogeneity properties (e.g $\lambda_{cr}(\kappa N) = \kappa^{-1}\lambda_c r(N)$). Most of these properties are still valid in the general case (non-orthotropic and combined loading), this is shown in the sequel.

3.2.2 Characterization of the buckling critical reserve factor and properties

In most cases, complex geometry, shear loading, non-orthotropic laminate, no closed-form expression can be derived. One way to derive some properties of the buckling critical reserve factor λ_1 is through variational formulation. Let us define some notations for sake of clarity

- $\mathcal{V} = \tilde{H}_0^2(\Omega) = \{v \in H_0^2(\Omega) \quad \text{s.t} \quad \frac{\partial v}{\partial n} = 0\}$, the Hilbert space of feasible displacement satisfying the boundary conditions (clamped[1]).

- $\mathcal{H} = H_0^1(\Omega)$. Note that a key property is that $\mathcal{V} \hookrightarrow \mathcal{H}$ and this embedding is continuous and compact. This ensures that the spectrum is countable.

[1] we do not derive it for the simply supported boundary conditions for we can not define straightforwardly the natural Hilbert space associated to $\Delta u_n = 0$ (trace operator is not defined for Δu since it is in $L^2(\Omega)$), however, a posteriori, solution u is in $H^4(\Omega)$ and the trace of the Laplacian can be defined

- $a_D : \mathcal{V} \times \mathcal{V} \mapsto \mathbb{R}^+$ a continuous symmetric bilinear coercive form where D is an element of $S_3(\mathbb{R})^{++}$
- $b_N : \mathcal{H} \times \mathcal{H} \mapsto \mathbb{R}$ a continuous symmetric bilinear form that need not be positive with $N = (N_x, N_y, N_{xy})$. A basic assumption to ensure existence of (realistic) positive buckling eigenvalues is that there exists $w \in \mathcal{H}$ such that $b_N(w, w) \geq 0$.

In the case of buckling of composite laminate these bilinear forms are

$$a_D(u,v) = \iint_\Omega D_{11}\frac{\partial^2 u}{\partial x^2}\frac{\partial^2 v}{\partial x^2} + D_{22}\frac{\partial^2 v}{\partial y^2}\frac{\partial^2 u}{\partial y^2} + 2(D_{12}+D_{66})\frac{\partial^2 u}{\partial x \partial y}\frac{\partial^2 v}{\partial x \partial y} + 4D_{16}\frac{\partial^2 u}{\partial x^2}\frac{\partial^2 v}{\partial x \partial y} + 4D_{26}\frac{\partial^2 u}{\partial y^2}\frac{\partial^2 v}{\partial x \partial y} \quad (3.59)$$

and

$$b_N(u,v) = \iint_\Omega N_x\frac{\partial u}{\partial x}\frac{\partial v}{\partial x} + N_y\frac{\partial u}{\partial y}\frac{\partial v}{\partial y} + \frac{N_{xy}}{2}(\frac{\partial u}{\partial x}\frac{\partial v}{\partial y} + \frac{\partial u}{\partial y}\frac{\partial v}{\partial x}) \quad (3.60)$$

The coercitivity of a_D simply follows from Poincaré's inequality applied to the gradient of u and from the positivity of the D tensor. We now have the following characterization of λ_1, known as the Courant-Fisher characterization

$$\lambda_{cr}(D, N_x, N_y, N_{xy}) = \lambda_1 = \min_{w \in \mathcal{V}\setminus 0} \frac{a_D(w,w)}{|b_N(w,w)|} \quad (3.61)$$

Let's first recall a result from (Rousselet & Chenais 1990) that states that λ_{cr} is continuous over D, N_x, N_y, N_{xy}, differentiability can not be established as well, indeed, what is shown in (Rousselet & Chenais 1990) is that λ_{cr} is differentiable whenever it is a simple eigenvalue, in case it is a multiple eigenvalue only directional differentiability can be established.

We can use (3.61) to derive some homogeneity properties. Let $\kappa > 0$ and $D \in S_3(\mathbb{R})^{++}$, $N = (N_x, N_y, N_{xy})$, for any $w \in \mathcal{V}$

$$a_{\kappa D}(w,w) = \kappa a_D(w,w) \quad (3.62)$$

and

$$|b_{\kappa N}| = \kappa |b_N(w,w)| \quad (3.63)$$

Taking the infimum over $\mathcal{V}$, we get

$$\lambda_{cr}(\kappa D, N) = \kappa \lambda_{cr}(D, N) \tag{3.64}$$

and

$$\lambda_{cr}(D, \kappa N) = \frac{1}{\kappa} \lambda_{cr}(D, N) \tag{3.65}$$

(3.64) and (3.65) mean that λ_{cr} is *homogeneous of degree* 1 over D and *homogeneous of degree* -1 over N. Applying Euler identity theorem (see appendix) to (3.64), (3.65) whenever λ_{cr} is differentiable gives

$$\sum_{i=1}^{6} D_i \frac{\partial \lambda_{cr}}{\partial D_i} = \lambda_{cr} \tag{3.66}$$

$$\sum_{i=1}^{3} N_i \frac{\partial \lambda_{cr}}{\partial N_i} = -\lambda_{cr} \tag{3.67}$$

These equations can be used within an optimization process to save sensitivity analysis computations. Note that by simply rewriting (3.65) in terms of the reciprocal of the buckling critical reserve factor

$$\frac{1}{\lambda_{cr}(D, \kappa N)} = \frac{\kappa}{\lambda_{cr}(D, N)} \tag{3.68}$$

and we see that the reciprocal of λ_{cr} is homogeneous of degree 1, which yields

$$N_x \frac{\partial \frac{1}{\lambda_{cr}}}{\partial N_x} + N_y \frac{\partial \frac{1}{\lambda_{cr}}}{\partial N_y} + N_{xy} \frac{\partial \frac{1}{\lambda_{cr}}}{\partial N_{xy}} = \frac{1}{\lambda_{cr}}. \tag{3.69}$$

Note any of equations (3.66), (3.67) and (3.69) can be used to directly derive an exact expression of the reciprocal of the buckling critical load factor whenever the partial derivatives are simple (e.g piecewise constant). It should emphasized however that these equations are not globally valid in the whole domain since the derivatives of λ_{cr} do not exist everywhere. We will see in the sequel that these equations are valid piecewise, since λ_{cr} is differentiable over connected regions that can be easily describe.

The homogeneity of the buckling critical reserve factor, which was already noticed, for instance with respect to the loading conditions in (Grenestedt 1991), can of course be used to reduce the size of the regression problem. One way to use it would be to build an approximation model over the sphere of radius $\{||N|| = K\}$ and use for

instance the spherical coordinates (θ, ϕ) as approximation variables. However doing so we would definitely reduce the dimension but we are likely to miss some simple dependency of λ_{cr} over the natural variables N. We can also observe that if we consider the buckling critical reserve factor defined over the panels thickness h and the lamination parameters ξ_D instead of being defined over the D tensor, we get

$$\lambda_{cr}(\kappa h) = \kappa^3 \lambda_{cr}(h) \tag{3.70}$$

This equation means that we can get rid of panel thickness h and only focus on lamination parameters. Finally, we can also observe that for the specific case of rectangular plates, we have by applying the change of coordinates $x' = \kappa x$ and $y' = \kappa y$

$$\lambda_{cr}(\kappa a, \kappa b) = \frac{1}{\kappa^2} \lambda_{cr}(a, b) \tag{3.71}$$

and the corresponding Euler identity. Note that the formula (3.71) is very similar to the formula for the first eigenvalue of the Laplace operator that can be found in (Henrot 2006) for instance. In this reference, it is shown that for an homothety $\kappa\Omega$, the first eigenvalue of the Laplace operator μ_{cr} satisfies

$$\mu_{cr}(\kappa\Omega) = \frac{1}{\kappa^2} \mu_{cr}(\Omega) \tag{3.72}$$

for any domain Ω.

3.2.3 Concavity of the buckling reserve factor over lamination parameters

We now turn on the concavity of λ_{cr}. To show that it is concave over the ξ_D, we first show that it is concave over D, concavity being preserved by pre-composition by a affine function (with h fixed), concavity over ξ_D will be ensured.

In this paragraph, the loading conditions remain fixed, we therefore skip them when writing the dependency of λ_{cr}. We want to show that λ_{cr} is concave over $S_3(\mathbb{R})^{++}$. Let $\alpha > 0$, consider the upper level set $D_\alpha = \{D \in S_3(\mathbb{R})^{++} \quad \text{s.t} \quad \lambda_{cr} \geqslant \alpha\}$. Let $D^{(1)}$ and $D^{(2)}$ in D_α and $\theta \in (0, 1)$, let denote $D^{(\theta)} = \theta D^{(1)} + (1-\theta) D^{(2)}$. $S_3(\mathbb{R})^{++}$ being convex, $D^{(\theta)}$ is clearly a symmetric positive definite tensor and consider now the bilinear form $a_{D^{(\theta)}}$. It is clearly continuous, symmetric and coercive. Let $w \in \mathcal{V}$ we then have

$$a_{D^{(\theta)}}(w, w) = \theta a_{D^{(1)}}(w, w) + (1-\theta) a_{D^{(2)}}(w, w) \tag{3.73}$$

hence

$$\frac{a_{D^{(\theta)}}(w,w)}{b_N(w,w)} = \theta\frac{a_{D^{(1)}}(w,w)}{b_N(w,w)} + (1-\theta)\frac{a_{D^{(2)}}(w,w)}{b_N(w,w)} \tag{3.74}$$

Since $D^{(1)}$ and $D^{(2)}$ are in D_α, for all $w \in \mathcal{V}$

$$\frac{a_{D^{(1)}}(w,w)}{b_N(w,w)} \geqslant \alpha \tag{3.75}$$

and

$$\frac{a_{D^{(2)}}(w,w)}{b_N(w,w)} \geqslant \alpha \tag{3.76}$$

hence for all $w \in \mathcal{V}$

$$\frac{a_{D^{(\theta)}}(w,w)}{b_N(w,w)} \geqslant \alpha \tag{3.77}$$

and $D^{(\theta)}$ is in D_α. This implies that λ_{cr} is quasiconcave, since its upper level sets D_α are convex. Being also homogeneous of degree 1, it is concave (see appendix). Note that strict concavity cannot be ensured, this follows obviously from equation (3.57) where we see that in the vicinity of $(\xi_D^1, \xi_D^2, 0)$ is a minimum of linear functions, hence at a point where λ_{cr} is differentiable, there exists a neighborhood where λ_{cr} is linear over (ξ_D^1, ξ_D^2) and cannot be strictly concave. Another quick way to see that is to note that, in case λ_{cr} is twice continuously differentiable over D, Euler identity for the homogeneity of order 0 of each partial derivative of λ_{cr} w.r.t to D_j yields

$$\forall j = 1 \dots 6 \quad \sum_{i=1}^{6} D_i \frac{\partial^2 \lambda_{cr}}{\partial D_j \partial D_i} = 0 \tag{3.78}$$

Equation (3.78) simply means that D belongs to the kernel of the Hessian $H\lambda_{cr}(D)$ of λ_{cr} at D and hence $H\lambda_{cr}(D)$ is not definite.

Using the following definition of the reciprocal $1/\lambda_{cr}(N)$

$$1/\lambda_{cr}(N) = \max_{w \in \mathcal{V}} \frac{|b_N(w,w)|}{a_D(w,w)} \tag{3.79}$$

together with the following *lower level sets* $N_\alpha = \{N \in \mathbb{R}^3 \quad \text{s.t} \quad 1/\lambda_{cr}(N) \leqslant \alpha\}$ and the triangular inequality leads to the the quasi-convexity of $1/\lambda_{cr}(N)$ over N. Being homogeneous or degree 1 it is convex. Strict convexity of $1/\lambda_{cr}(N)$ cannot be ensured the same way as the strict concavity of $\lambda_{cr}(\xi_D)$.

Characteristics	Graphite/Epoxy
E_1 Longitudinal modulus	112 GPa
E_2 Transverse modulus	8.2 GPa
G_{12} Shear modulus	4.5 GPa
t Elementary ply thickness	0.26 mm

Table 3.1: Graphite/Epoxy material characteristics

3.2.4 Influence of the material

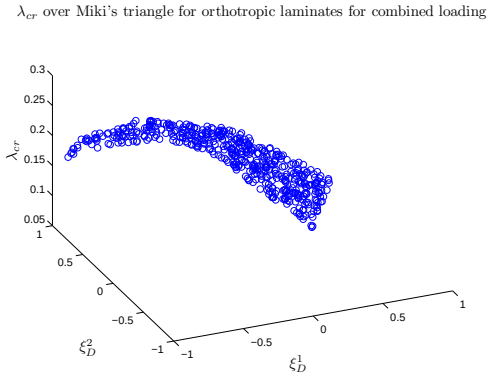
We focus in this section on the behavior of the buckling critical reserve factor λ_{cr} when the loading N is fixed and when the stacking sequence of the plate laminate varies. We use lamination parameters ξ_D to assess this behavior. We focused here on lamination parameters of conventional laminates $[0°/45°/-45°/90°]$, in that case $\xi_D^4 = 0$. We then restrict our study to the influence of $(\xi_D^1, \xi_D^2, \xi_D^3)$ over the feasible domain described in (Bloomfield *et al.* 2009), (Herencia *et al.* 2008a) and (Herencia *et al.* 2008b)

$$2|\xi_D^1| - \xi_D^2 - 1 \leqslant 0 \tag{3.80}$$

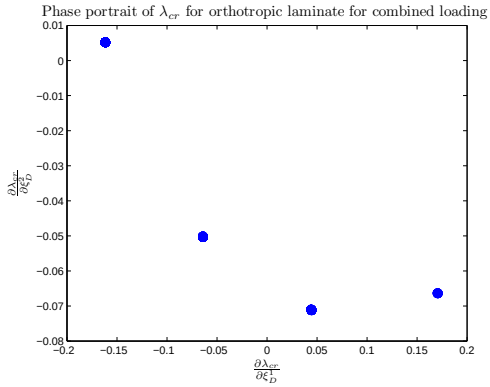
and

$$2|\xi_D^3| + \xi_D^2 - 1 \leqslant 0 \tag{3.81}$$

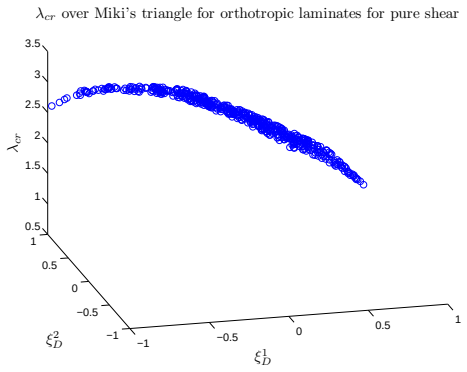
Note that it is a convex domain.The triangle in $[-1,1]^2$ that Eq. (3.80) will be referred to as *Miki's triangle* (with $\xi_D^3 = 0$) and the polyhedron in $[-1,1]^3$ defined both by Eq. (3.80) and Eq. (3.81) will be referred to as *Miki's tetrahedron*. The main interest of the compatibility equations described in (Herencia *et al.* 2008a) is that they provide a convex feasible set for $\xi_D \in [-1,1]^3$ for given in-plane lamination parameters ξ_A (or equivalently proportions of fiber orientation). In an optimization process, this offers a way to ensure compatibility in a continuous way between in-plane properties (A tensor) and out-of-plane properties (D tensor). In the sequel we study the behavior of λ_{cr} all over its definition domain with no regards to proportions, but we should keep in mind that in practice, the out-of-plane are related to proportions (or equivalently ξ_A). Note that most of the computations done in this paper were based on a Rayleigh-Ritz method (see appendix) with $N = 20 \times 20$ basis functions over a rectangular plates of dimensions $a = 650$mm and $b = 250$mm, we chose a classical Graphite/Epoxy whose characteristics are given Tab. 3.2.4.



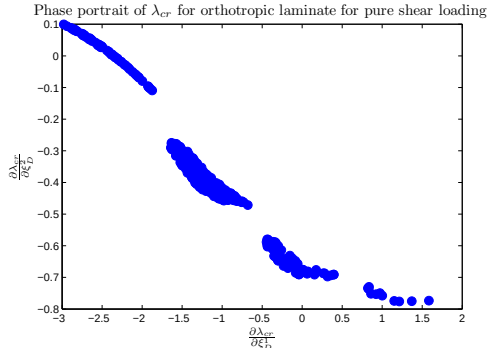
(a) Piecewise dependence of λ_{cr} over ξ_D^1 and ξ_D^2 on Miki's triangle for combined loading $N_x = -50N.mm^{-1}$, $N_y = -20N.mm^{-1}$ and $N_{xy} = 20N.mm^{-1}$



(b) Phase portrait associated to the couple ξ_D^1, ξ_D^2 for combined loading. Each point is the associated partial derivatives

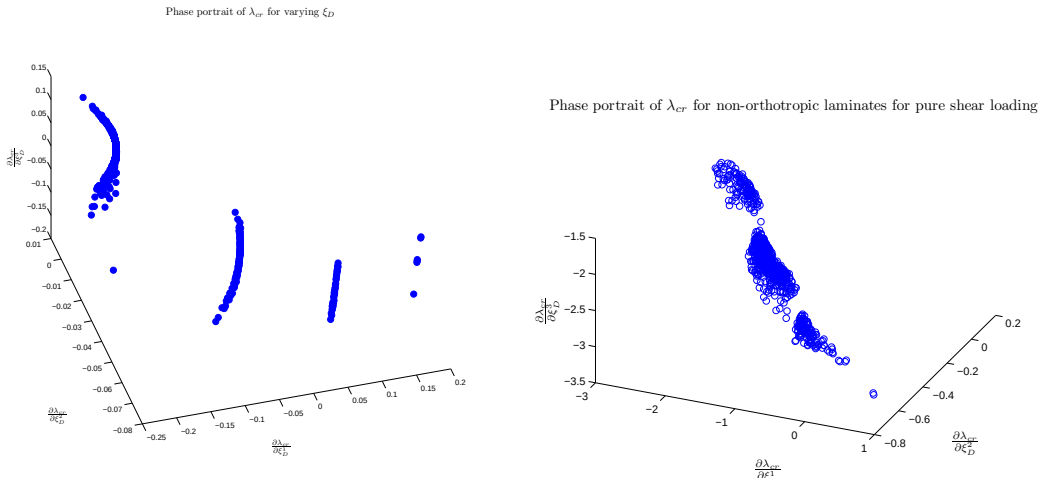


(c) Piecewise dependence of λ_{cr} over ξ_D^1 and ξ_D^2 on Miki's triangle for pure shear loading $N_x = 0N.mm^{-1}$, $N_y = 0N.mm^{-1}$ and $N_{xy} = 20N.mm^{-1}$



(d) Phase portrait associated to the couple ξ_D^1, ξ_D^2 for pure shear loading. Each point is the associated partial derivatives

Figure 3.8: Results for orthotropic laminates under combined loading and pure shear loading



(a) Phase portrait associated to the couple ξ_D^1, ξ_D^2 for combined loading. Each point is the associated partial derivatives

(b) Phase portrait associated to the couple ξ_D^1, ξ_D^2 for pure shear loading. Each point is the associated partial derivatives

Figure 3.9: Results for non-orthotropic laminates under combined loading and pure shear loading

3.2.4.1 Behavior of λ_{cr} for orthotropic laminates under combined loadings and pure shear

For uniaxial and biaxial compressive loadings, a closed-form equation is given by (3.57). For fixed loading, it follows that λ_{cr} is piecewise linear over tensor D terms and therefore piecewise affine over ξ_D. For combined loadings (including shear) or pure shear loading, no closed form expression can be derived. The main reason for this is that the $\frac{\partial^2 w}{\partial x \partial y}$ term prevents from using the classical techniques as a double sine series (see (Reddy 2004)). Correction formulas were given (see for instance (Whitney 1969)), but all of them were approximation expressions based on simplified assumption (infinite length plate for instance) and there is no evidence at the time being that

- for arbitrary loading (combined loading, pure shear), λ_{cr} is still piecewise affine over ξ_D for orthotropic laminate
- for arbitrary loading, λ_{cr} can be defined in a piecewise manner for non-orthotropic laminates

In the sequel, we show numerically that λ_{cr} seems to be piecewise affine over ξ_D for combined loading where shear is not very high. However, for pure shear loading, we observed that λ_{cr} can still be defined in a piecewise manner but each part of the function seems to be more complex than affine functions. The most important fact shown here is that λ_{cr} behaves in a piecewise manner and a good approximation technique should be able to detect these different regions or so-called modes and build an approximation for each region.

To assess the evolution of λ_{cr}, we sample uniformly Miki's triangle for ξ_D^1, ξ_D^2 and compute for each couple of lamination parameters its associated λ_{cr}. The loading is taken as a combined loading with shear and we also compute the associated sensitivities $\frac{\partial \lambda_{cr}}{\partial \xi_D^1}$ and $\frac{\partial \lambda_{cr}}{\partial \xi_D^2}$ and plot the phase portrait: for each couple we associate a point of the gradient space.

The different plots depicted Fig. 3.8 were obtained for two different loading conditions

- Combined loading: $N_x = -50$ N.mm^{-1}, $N_y = -20$ N.mm^{-1}, $N_{xy} = 20$ N.mm^{-1}
- Pure shear loading: $N_x = 0$ N.mm^{-1}, $N_y = 0$ N.mm^{-1}, $N_{xy} = 20$ N.mm^{-1}

For each loading case, we generated 800 couples (ξ_D^1, ξ_D^2) uniformly distributed over Miki's triangle. We depicted the behavior of λ_{cr} Fig. 3.8 a) and **??** c). We can observe that for the first loading case, we clearly see the piecewise dependence of λ_{cr} over (ξ_D^1, ξ_D^2), as in the biaxial compression case of Eq. (3.57), λ_{cr} seems still defined as the minimum of hyperplanes, making the overall function simply defined over a sole hyperplane but not differentiable at the boundaries between hyperplanes. It is worth noting that directional differentiability seems still valid as it is proved in (Rousselet & Chenais 1990). For this loading case, λ_{cr} seems to be still defined as piecewise linear, this is numerically ensured by the phase portrait Fig. 3.8 b), where we see all the 800 points shrink down to only 4 different points, which means over each region, λ_{cr} is linear or very close to be linear. For the pure shear loading case, this dependence does not seem to hold, indeed, the regions are not as clearly defined as in the combined loading. The phase portrait shows 4 different regions disconnected,

which again means that λ_{cr} can be defined in a piecewise manner, though its behavior over each region seems more complicated than linear.

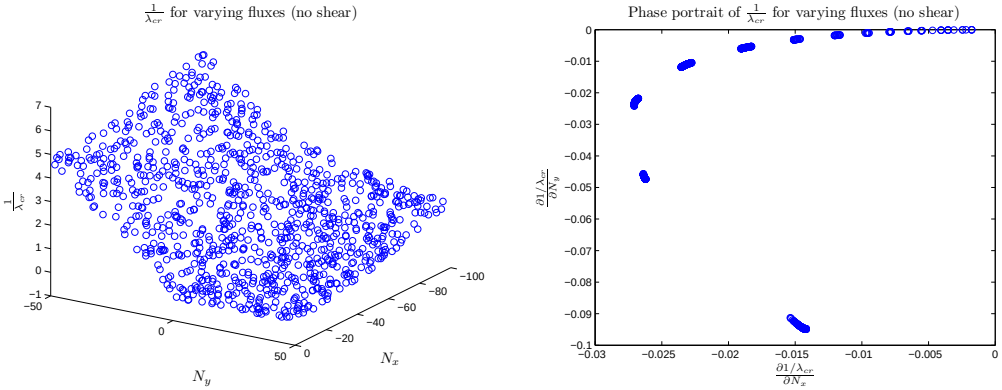
3.2.4.2 Behavior of λ_{cr} for non-orthotropic laminates under combined loadings and pure shear

We also depicted Fig. 3.9 the phase portrait for non-orthotropic laminates by making the ξ_D varying in the Miki's tetrahedron for the same loadings. We observe in both cases that the phase portrait has the same structure that is to say 4 disconnected components. However, the dispersion of these clusters are quite different for the two loadings. In the first case, for combined loading, we observe a low variance over the two first dimensions ($\frac{\partial \lambda_{cr}}{\partial \xi_D^1}$, $\frac{\partial \lambda_{cr}}{\partial \xi_D^2}$) and bigger variations for the last dimension ($\frac{\partial \lambda_{cr}}{\partial \xi_D^3}$),this would naturally indicate that λ_{cr} is close to depend linearly over the two first dimensions and that it depends in a more complex manner over the last dimension. In case of pure shear loading, the dependence is much more complex, it does not seem that there is a dimension that has a bigger influence than the others. However, the main observation that we can make is that in all cases, λ_{cr} seems to depend in a piecewise manner and that the function is likely to be well approximated when taking advantage of this structure.

3.2.5 Influence of the loading conditions

In this section, we are interested in the behavior of the λ_{cr} for fixed stacking sequence when the loading conditions vary. In most cases, when loading conditions vary, the reciprocal of λ_{cr} was found to be more easier to depict and interpret. This can be easily seen again in the general formula (3.61) where all the stacking sequence dependence appear in the numerator and all the loading conditions appear in the denominator. Furthermore, it also follows from both theoretical and computational considerations. Indeed, for instance in (Rousselet & Chenais 1990), the authors do not study directly λ_{cr} but first use its reciprocal to show it is continuous and differentiable whenever it is a simple eigenvalue. Numerically, it is preferable to compute the reciprocal of λ_{cr} that is to find w and the largest κ such that

$$Lw = \kappa K w \tag{3.82}$$



(a) $\frac{1}{\lambda_{cr}}$ over (N_x, N_y) for non-orthotropic laminate

(b) Phase portrait associated to the couple (N_x, N_y) for non-orthotropic laminate. Each point is the associated partial derivatives

Figure 3.10: Evolution of the reciprocal of λ_{cr} for a fixed non-orthotropic stacking sequence w.r.t loading conditions

where K is the matrix associated to the a_D bilinear form (stiffness material matrix) and L matrix associated to the b_N bilinear form (geometry stiffness matrix) and eventually $\lambda_{cr} = 1/\kappa$. Next, as noted before, in case of the orthotropic formula, we can observe that $1/\lambda_{cr}$ is piecewise linear over $N = (N_x, N_y)$. Regarding computational issues when making the loading conditions vary, it can be noted that for general loading (including shear) the feasible set (i.e the fluxes that gives rise to buckling, this is related to the assumption of positivity of b_N) are not easy to describe. When the loading conditions approaches near the boundary, the eigensolver might not converge or exhibit numerical ill-conditioning behavior and λ_{cr} increases very quickly and behaves like an hyperbola.

What we observe Fig. 3.10 is that the reciprocal of λ_{cr} behaves quite simply. At first glance, we could think it is piecewise linear, however the phase portrait show many different regions where the gradient do not vary very much. We could imagine then that a piecewise linear approximation would perform quite well.

3.3 Towards approximation strategy for discontinuous functions

We investigated the behavior of the buckling criticial load factor over lamination parameters. This behavior numerically seems to feature regions of continuity (or even

more regularity) well separated at least in the gradient space. Owing to this feature, one appropriate approximation strategy should consider dividing the input space (or learning space in the parlance of statistical learning) and build a local approximation in each region. This is the innovative strategy that we present in the next chapter.

4 Original strategy to approximate discontinuous and derivative discontinuous functions

In a lot of complex data-based approximation problems, the use of one sole predictor (Artificial Neural Network, Support Vector Machines,...) often exhibits a lack of accuracy especially in high-dimensional problems. This is due to the growing complexity of the underlying functions that present different behaviors depending on the region. If we think of computer experiment approximations for instance, the output of the true function can be noisy, and/or computed according to different algorithms with respect to the entries or feature discontinuities based on the underlying physics but also on the programming methods. Despite its versatility, one simple predictor might not be enough to correctly predict the output on the whole entries space. Ongoing research suggests to replace one predictor by many of them, making the prediction more accurate by taking into account the specific structure of the data. In this chapter, we present a general framework to enhance the learning of several experts based on probabilistic clustering techniques. The learning data is used to find an appropriate probabilistic description of the data by means of mixtures laws. The input-output relationship is first modeled as a specific distribution whose parameters are to be estimated. This estimation usually proceeds from maximum likelihood estimation. We first find the mixture law of the input-output space and then design a clustering technique to separate the learning data. This split of the data may be "hard" or "soft", leading either to strictly defined clusters or to probabilistic assignment of each learning point to the whole set of clusters. Once this assignment is done, a local or fuzzy expert is built, making the overall predictor as a hard or soft split following the model of one level tree whose leaves are simple predictors : artificial neural network, quadratic regression...Generalization is done through Bayesian approach of maximum a posteriori (MAP) estimation in the projection of the

mixture probability law in the entries space. This technique can be used to improve accuracy in regression problems but also misclassification rate in classification problems. Furthermore, the clustering part can be used to help parameter estimation in radial basis network or weighted least square model. It can also be used for local polynomial regression or kernel smoothing techniques.

4.1 Motivations

Our main objective in that chapter is to derive a new strategy to allow accurate approximations of the typical stability constraints to be used in an optimization process. These functions typically features derivative discontinuities and also real discontinuities.

4.1.1 Typical behavior of stability constraints

As shown in the former chapter, stability constraints and more specifically buckling computations exhibit derivative discontinuities with respect to the buckling analysis parameters (lamination parameters w.r.t changes of the stacking sequence, forces w.r.t changes in the loading conditions, geometric dimensions w.r.t a fixed profile). These discontinuities are physical-based and relies on the very computation of the critical buckling load factor (or **Reserve Factor**) formally defined as the smallest positive eigenvalue of a pde. Moreover, if we think of a more general optimization where for instance the shape of the stringer can change and takes different categorical variables (T-shaped, I-shaped,...) the switch from one type of domain to another one also leads to discontinuities. The physics behind lead to derivative discontinuities but they are also practical reasons that can make the computation not of the critical buckling load factor but of quantities derived from them really discontinuous. Think for instance of margin policies, rule-of-thumbs that are based on the art of the engineer that favor one buckling critical factor over another one for some prescribed values of the parameters. These discontinuities can also derive from programming reasons, a common example in in-house tools to reduce the number of outputs to check is to consider minimum of several different buckling modes. To make this more clear, the Airbus skill tool named COFUS (see also Chapter 7) which analyses the stability of a super-stiffener (two halves thin shallows shells of possibly different curvature radii, thicknesses and stacking sequences reinforced by a stiffener and simply supported on the frames) as

an output for stiffener buckling that generally analyses the stability of the different parts of the stiffeners (web, flange, foot) and gives the minimum buckling reserve factor. These types of buckling essentially depends on the geometric dimensions of the stiffener, the stacking sequence of the stiffener but also on the in-plane properties of the panels (proportions of orientations and forces (loads per unit length) that the two half-panels endures). However, for some specific shapes or stringer, e.g Omega-shaped stringers) this stringer buckling reserve factor also features a reserve factor associated to the buckling o the small panel between the two webs of the stringer: this is typical skin buckling and it essentially depends on the stacking sequences of the panels. This example shows two essential characteristics of the functions we want to approximate, first a given output when switching from one buckling type to another shows discontinuities but also the variables that explains the behavior of the function can change. So a clever strategy to accurately approximate these outputs should not only detect discontinuities but also locally weights the predictive variables by allowing some variables to be more important in some areas and switching to other predictive variables in other regions of the input space. There exists another type of functions that we consider in this work, this the typical optimal value function (presented in Chapter 2). The functions that associates the value of the function at a minimum of a certain optimization problems to parameters that govern this optimization problem. We saw Chapter 2 that the derivative of that function w.r.t optimization parameters exists, provided the minimum satisfies some conditions. However, as a global function this optimal value function (or **design curves** in aerospace jargon) often exhibits derivative discontinuities from changes in the set of active constraints. On the bilevel context, one often tries to build accurate approximations of this optimal value function. As outlined in (Haftka & Watson 2005), constructions of such accurate approximations is challenging due to the high number of input variables and also discontinuities. A last but not least reason making these approximation hard to build is that the learning points come from an optimization algorithm that may not always converge, or that can stop because of a fixed number of iterations, or even worse the algorithm might give no useful results (crash or get stuck in a non feasible point) making the quality of the learning points hard to assess. An optimal strategy for approximating such results should also consider the quality (or at least an 'approximation' of the quality) as an important feature associated to the construction of the approximation.

4.1.2 From one expert to several experts

We should emphasize and recall here that these approximation are mainly used to speed up design. When used within a design process approximation models were developed to tackle the slowness of repetitive code evaluations. When used within a design process these approximation methods are often called surrogate models. Surrogate models arise from statistics and probability theory and are now widespread tools to approximate complicated functions. They are used inside the optimization process to approximate the objective function or the constraints or they can directly approximate the results of the optimization process as a function of the optimization problem parameters (materials characteristics, load case in structural optimization for instance). They are also widely used in the multilevel and multidisciplinary optimization framework (Liu *et al.* 2004), (Merval 2008) and (Merval *et al.*). Surrogate modeling offers many ways to approximate function from sample data : artificial neural networks (ANN) (Dreyfus 2005), moving least squares (MLS) (Nealen & Darmstadt 2004), radial basis functions (RBF) (Buhmann 2001), kriging (van Beers & Kleijnen 2004), support vector machines (Smola & Schölkopf 2004), multivariate adaptive regressive splines (MARS) (Friedman *et al.* 2001). A good overview of the existing surrogate models can be found in (Friedman *et al.* 2001), (Kleijnen *et al.* 2005), (Wang & Shan 2007) and (Simpson *et al.* 2008). In (Forrester & Keane 2009), Forrester and Keane provide an intensive review of most of the surrogate models and compare them when used within an optimization process. Nonetheless, one simple surrogate model might not be enough to approximate a complicated function, especially when this function features different behaviors depending on the region of the input space. This situation happens quite often in mechanics when computing critical buckling modes. For instance in (Merval 2008) the optimization constraints to be approximated (reserve factors for skin buckling and local web stringer buckling for a composite stiffened panel) happen to be discontinuous and derivative-discontinuous which precludes the training of an accurate surrogate model. Indeed surrogate models usually assume that the function to approximate is smooth and are themselves smooth, resulting in a high variance around the discontinuities and therefore on a overfitting that make the generalization power of the surrogate model poorer. One way to prevent this overfitting would be to get rid of this discontinuities first by building several surrogate models but also by dividing the

input space into regions that do not feature discontinuities.

To improve the accuracy of a surrogate model, a common practice is to build several surrogate models on the same common learning basis. One is more likely to find a more accurate surrogate when building many surrogate models. As explained in (Viana *et al.* 2009) several surrogate models prevents from building poorly fitted models. On the basis of the different surrogate models, one can the best one based on the classical statistical techniques of cross-validation and bootstrap (see for instance (Kohavi 1995) and (Picard & Cook 1984)) to obtain estimates of the generalization error of the different surrogate models. As pointed out in (Acar & Rais-Rohani 2009), one of the drawbacks of choosing the best predictor is that we do not make use of all the resources used in the construction of the discarded surrogate models. The chosen predictor may be globally precise enough but may lack accuracy in some crucial areas of the input space (boundaries for instance), while one of several of the discarded surrogate models may perform better in these very areas. One could overcome this drawback by combining all the surrogate models by means of weights. This practice of combination relies on the same basis as committees of machines in artificial intelligence. A committee of machines is a collection of intelligent agents that vote and decide all together, hoping that errors would cancel as there are several experts. In the area of machine learning, this practice of combination appears in the *bagging* and *boosting* techniques. In the case of ensemble of surrogate models, the different surrogate models may be simply averaged or weighted. Note that the weighting may be done globally (constant weights over the input space) as it is done in (Viana *et al.* 2009) and in (Acar & Rais-Rohani 2009) or locally (depending on the input space) as it is done in (Zerpa *et al.* 2005) and in (Sanchez *et al.* 2008). Even though the idea of combining several surrogate models seems appropriate to approximate functions, there is no evidence that combining is always better than selecting the best surrogate, as it is pointed out in (Yang 2003).

Our approach is in the framework of ensemble of locally weighted surrogate models except it is also based on a partitioning of the learning basis, while in most of the described techniques of ensemble, all the surrogate models are built on the same common learning basis. Indeed, our concern is about the approximation of functions featuring

discontinuities, heterogeneous behaviors and very different landscapes depending on the region of the input space. This is why, in our approach, a surrogate model is built over a specific region of the input space. From an optimization point of view, the global surrogate model needs to be continuous and even smooth (for gradient-based optimization techniques), this is why we combine them in a way that their errors are canceled, especially in the vicinity of discontinuities. As pointed out, this approach slightly differs from the ensemble of surrogate models, since each surrogate model, though applied over the whole input space at the very end, is only built over a specific region of the input space. Our approach is based both on the same idea as the committee of machines but also on the 'Divide and Conquer' principle. In the literature, this kind of approach is referred to as mixture of experts (MoE's). A general introduction of mixture of experts can be found in (Friedman *et al.* 2001). One classical application is known as hierarchical mixture of experts (HME) and is described in (Jordan & Jacobs 1994). In this study, Jordan and Jacobs present a general architecture of mixture of experts for supervised learning (regression and classification). This architecture is a tree structure where each nonterminal produces a soft split of the input value coming from the upper level. This soft split consists in giving different weights to the lower sub levels by means of a generalized linear model and the authors call it *gating network*. Each gating network produces soft splits until the terminal leaves which produce output (real value for regression and binary for classification) by means also of a generalized linear model. These terminal leaves are called by the authors *expert network*. The parameters of the different generalized linear models are estimated using a classical algorithm in statistics : the Expectation-Maximization algorithm (EM algorithm) on the input-output space, which means that in their study, partitioning and learning are based on the same algorithm. We propose a different method where clustering is separated from learning. The gating networks are not generalized linear models (GLIM) but Gaussian mixture models (GMM) still estimated through EM algorithm. Based on the Gaussian mixture models estimates, the input-output space (or conjoint space) is clustered with respect to the maximum a posteriori (MAP), once this clustering is done, we have a certain number of sub-bases and a surrogate model is trained over each sub-basis, this surrogate model can be quadratic regression, artificial neural networks and moving least squares regression, while in Jordan and Jacobs use a generalized linear model. All the surrogate models are combined on the basis of the Gaussian parameters found by EM

algorithm. Roughly speaking, our technique relies on a certain number of hard tasks that try to answer the following questions

a) How do we cluster the learning basis ? Clustering (or *automatic classification*) is one of most important areas of unsupervised learning. It aims at finding groups whose individuals are close in some sense into a collection of points. There are many different techniques to cluster data (K-means, K-medoids, quality threshold clustering (QT clustering), density clustering see (Berkhin 2002) for instance where most of the classical algorithms for clustering are reviewed). Some of them are hard clustering : a point of the design space belongs to one and only one cluster, some others are fuzzy clustering the points belong to several clusters, each point is associated to a random vector defining the probabilities to lie within each cluster.

b) Which local experts do we build and how do we combine them? As pointed out in (Yang 2003), there is no reason that combining several surrogate will perform better than only one, which means that the combination has to be done carefully based on a reasonable way to cancel errors.

c) How to choose the number K of clusters? This is a rather difficult question since there might not be perfect number of clusters. Indeed, we just want to find a good number of clusters such that each expert would do well enough to be combined. There might be different choices for K. The question of the number of clusters is central in clustering theory and involves different tools from statistics, probability theory and information theory. In (Burnham & Anderson 2004) some of the common criteria to find the best number of clusters (Akaike Information Criterion and the Bayesian Information Criterion (AIC, BIC) are thoroughly investigated and compared).

As said earlier, our concern is mostly about the approximation of discontinuous functions to the end of optimization. The clustering should be done such that the boundary between clusters would be as close as possible to the real discontinuities. We assume that these discontinuities may be distributed all over the domain and would naturally divide the input-output space into bunches of points that are connected but that may

be anisotropic. Clustering should also be done in such a way that the we have a center and a parameterization of each cluster to combine the local surrogate models. To that end, we assume that a good representation of the conjoint data for discontinuous functions would be Gaussian mixture models. The EM algorithm would allow us to find good estimates of the Gaussian parameters. EM clustering for Gaussian mixture models gives such an anisotropic representation of the data and therefore a parameterization of the clusters that makes possible to combine the different surrogate models built on each cluster. In (Bradley *et al.* 1998) some of the benefits that can be taken from using EM clustering are developed. Next, we describe our method by answering the former questions. We first present the theoretical background of our proposed technique : Gaussian mixtures models, clustering based on the GMM estimates in section 2 and answer question a). We then focus on the combination of the local experts trained on the basis of the clustering, this will allow us to answer question b) and derive the original algorithm presented in section 3 to improve the accuracy of surrogate modeling for discontinuous functions. In section 4, we validate our proposed algorithm on test cases obtained from a discontinuous functions samples generator called Samgen and we also give a practical answer to question c). In section 5, we finally test our proposed algorithm on a structural optimization problem from aeronautics. We also give an original interpretation of a standard surrogate model the weighted least squares (WLS) in terms of mixture of experts based on a soft clustering and local quadratic experts in section . The tool called Samgen we implemented to provide highly discontinuous functions in arbitrary dimension is outlined in appendix C.

4.2 Mixtures of probability laws as a way to handle complexity

This section is devoted to the description of the formal mixture probability model. Apart from the end of the section where we restrict ourselves to the specific case of categorical variables and multivariate regression, we do not impose a specific probability density function. After a brief introduction to unsupervised learning, we first describe mixture laws and the way the mixture parameters and laws components can be estimated through classical maximum likelihood estimation.

4.2.1 Unsupervised learning and density estimation

Unsupervised learning is concerned with the exploration of data. In this case, we do not look for prediction based on learning data for new input data, but solely on a as much as possible adequate description of the learning data base. More formally, suppose we are given a set of N observations $(x_1, x_2, \ldots, x_N)$ of a random vector X in $\mathbb{R}^p$ having probability density g, the goal is to estimate properties of the density to get a better understanding of the data. Hence, the objective of unsupervised learning is to give an insight of the data and not to predict new outcome. Indeed, in supervised learning the overall goal is to be able to predict outcome or response for independent input based on a learning database and the accuracy of the prediction can be measured directly. In unsupervised learning, the lack of outcome makes this accuracy hard to assess. We will see later that our technique combines unsupervised learning techniques with supervised learning objective and hence such techniques have been called semi-supervised learning techniques.

One particular area of unsupervised learning is related to partition a collection of observations into subsets or 'clusters'. The observations lying in the same cluster share properties that make them similar in some way. More precisely, observations within a same cluster are similar amongst them and dissimilar with regard to the observations in the other clusters. To achieve clustering, we are given a number of clusters K say, with $K < N$ and a measure of similarity between observations $s(x_i, x_j)$. We therefore want to find the optimal many-to-one mapping $C : \{1, \ldots, N\} \to \{1, \ldots, K\}$, that assigns to observations one and only one cluster, e.g $C(i) = k$ assigns the i^{th} observation to cluster k. This mapping is a many-to-one mapping, since in principle, we want to explain the best we can the data hence, the number of clusters should be far less than the number of observations. Formally, a clustering algorithm seeks a mapping $C^\star$ minimizes the 'within-cluster' similarity $WC(C)$:

$$WC(C) = \frac{1}{2} \sum_{k=1}^{K} \sum_{C(i)=k} \sum_{C(j)=k} s(x_i, x_j) \tag{4.1}$$

where the $\frac{1}{2}$ factor accounts for the fact that each pair of observations lying in the same cluster k is counted twice. We can also equivalently maximizes the 'between cluster'

dissimilarity $BC(C)$:

$$BC(C) = \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\sum_{C(j)\neq k} s(x_i, x_j) \tag{4.2}$$

The equivalence between minimizing (4.1) and maximizing (4.2) follows from the fact that the sum $WC(C) + BC(C)$ is constant for all $C \in A(N, K)$, where $A(N, K)$ is the set of all different assignments of K objects within a set of N objects, since for any C :

$$\begin{aligned} WC(C) + BC(C) &= \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\Big(\sum_{C(j)=k} s(x_i, x_j) + \sum_{C(j)\neq k} s(x_i, x_j)\Big) \\ &= \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} s(x_i, x_j) \end{aligned}$$

which does not depend on C. Clustering is, at least formally, a simple combinatorial problem: once we are given a similarity measure s, clustering simply aims at minimizing or maximizing the quantities below over the finite set $A(N, K)$, set of $k-$permutations (sequences without repetitions) of n elements. Unfortunately, the size of $A(N, K)$ increases very fast making practical clustering based on these optimizations unfeasible whenever N is big ($N > 10$), recall that $A_n^k = \frac{n!}{(n-k)!}$. It should be noted however that this approach does not suffer the curse of dimensionality usually encountered in supervised learning and also this formal clustering problem is non-parametric in the sense it simply assigns one cluster to each observation and it uses the raw data with no specific parametric model. We end this section with the non-parametric description of the most popular algorithm for clustering : K-means, we will see later that K-means has a rather natural parametric description from the Gaussian mixture point of view.

4.2.2 K-means algorithm

In practice, standard clustering algorithms seek ways to simplify the problem through the following iterative process :

1. Start with an initial partition of the data : C^0
2. At each iterative step, through a given prescription, assign clusters to observations that improve the performance function, e.g minimize (4.1) or maximize (4.2).

3. When there is no better assignment for the given prescription, stop the clustering

Benefit of that simple procedure is that it gives a rather straightforward way to cluster data, once a similarity measure and a prescription have been chosen; the drawback is it can converge to sub-optimal clustering. For K-means algorithm, the similarity measure is the classical squared l_2-norm

$$s(x_i, x_j) = ||x_i - x_j||^2 \tag{4.3}$$

Hence, the within-cluster similarity is

$$WC(C) = \sum_{k=1}^{K} N_k \sum_{C(i)=k} ||x_i - \hat{x}_k||^2 \tag{4.4}$$

where N_k is the number of observation lying in cluster k and $\hat{x}_k$ is the mean of all observations lying in cluster k. This leads to the following algorithm, known as K-means algorithm (described for instance in (Friedman *et al.* 2001))

1. Start from an initial clustering C^1 (at random for instance).
2. From clustering at step n, C^n, solve for any $k = 1 \dots K$, $\quad \text{argmin}_{m_1,\dots,m_p \in \mathbb{R}^p} \sum_{c(i)=k}^{N} ||x_i - m||^2$, which leads to $m_k^{n+1} = \hat{x}_k$.
3. From the set of means $m_1^{n+1}, \dots, m_K^{n+1}$, assign a new clustering C^{n+1} with $C^{n+1}(i) = \text{argmin}_{k=1\dots K} ||x_i - m_k^{n+1}||^2$
4. Repeat, step 2 and 3 until convergence

K-means algorithm is an approximate procedure to solve

$$\min_{C, \{m_1,\dots,m_K\}} \sum_{k=1}^{K} N_k \sum_{C(i)=k} ||x_i - m_k||^2 \tag{4.5}$$

In K-means, we use the classical Euclidean norm. Owing to its simplicity, this algorithm is one of the most popular clustering algorithms. Nonetheless, in some situations, it might not capture all the complexity of the data. To explain this, we will have a more probabilistic point of view of K-means that will show how to improve K-means algorithm.

4.2.3 Mixture laws

Mixture laws offer a way to represent the underlying structure of complex data. In the sequel, we will only focus to the finite case of mixture laws. We say that a real random vector X comes from a mixture model of K laws with density probability functions $(g_k)_{l=1...K}$ if its probability density function is

$$x \to g(x) = \sum_{k=1}^{K} \alpha_k g_k(x) \tag{4.6}$$

where the α_k's weight each law into the mixture with $\forall k, \alpha_k \geq 0$ and $\sum_{k=1}^{K} \alpha_k = 1$. In the latter, we consider parameterized laws, i.e each law g_k is determined through a record θ_k of parameters. So we can note it

$$g(x) = \sum_{k=1}^{K} \alpha_k G(x, \theta_k) \tag{4.7}$$

For univariate normal distribution, $\theta_k = \{\mu_k, \sigma_k^2\}$ where μ_k is the expectation and σ_k^2 the variance. For multivariate normal distribution in $\mathbb{R}^p$, $\theta_k = \{\mu_k, \Gamma_k\}$ where $\mu_k \in \mathbb{R}^p$ and Γ_k is the variance-covariance matrix, symmetric positive definite matrix of size p. We already see that the number of parameters to be set can grow quickly with the number of individual components K and even more quickly with the dimension p. For instance, for fully free multivariate Gaussian mixture in dimension p where fully free means that all the terms of variance-covariance matrices are to be estimated, the number of free parameters is

$$K + K(p + \frac{p(p+1)}{2})$$

summing up for instance with $K = 10$ and $p = 5$ to 210 terms. Mixture density estimation consists in finding the parameters that best describe the sample $(x_i)_{i=1...n}$. This is to be achieved through maximum likelihood estimation. Namely, we will try to find the mixture parameters that are the most likely to give the sample. As we will see in the next section, maximum likelihood estimation consists in a nonlinear optimization problem. The increasing number of parameters, as in the example mentioned, show that classical nonlinear programming techniques (SQP...) will rapidly become unfeasible. EM algorithm is a statistical algorithm that aims at solving this type of problem.

4.2.4 Estimation of mixture parameters and mixture components parameters through maximum likelihood : EM algorithm

Our objective is to fit a mixture model to data. An important and classical measure of the goodness of a probabilistic model, i.e a probability density function $x \to g_\theta(x)$ with parameters θ is the **likelihood**. By 'measure of goodness' we mean a measure w.r.t parameters θ and independent real data sample $\mathcal{X} = (x_1, \ldots, x_N)$, the likelihood $\mathcal{L}(\theta; \mathcal{X})$ is then

$$\mathcal{L}(\theta; \mathcal{X}) = \prod_{i=1}^{N} g_\theta(x_i) \tag{4.8}$$

Likelihood function is to be thought as a function of θ with fixed $\mathcal{X}$. This measure is to be maximized over the space of parameters. Indeed, $\mathcal{L}(\theta; X)$ is the density of the observed data under the model g_θ. We often consider the logarithm of the likelihood, called log-likelihood

$$l(\theta; \mathcal{X}) = \sum_{i=1}^{N} \log g_\theta(x_i) \tag{4.9}$$

and instead of maximizing the log-likelihood, we often minimize the negative of the log-likelihood. The maximum likelihood estimation seeks $\hat{\theta}$ that minimize $-l(\theta; \mathcal{X})$. In case of simple probability density function, e.g one simple univariate normal distribution where the mean μ and the standard deviation σ^2, the log-likelihood happens to be minimum at the classical estimate $\hat{\mu} = \frac{1}{N}\sum_{i=1}^{N} x_i$ and $\hat{\sigma}^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \hat{\mu})$ and exhibits simple variations (convex or so). But with more sophisticated density function such as mixture models, this function may exhibit numerous local minima. This behavior and the high number of optimization variables prevent from using classical deterministic optimization algorithms.

The EM algorithm aims at solving maximum likelihood parametric estimation problems for mixture densities as $\sum_k \alpha_k G(\theta_k, x)$. The EM algorithm was first described in (Dempster *et al.* 1977).

The idea is to introduce a "latent" discrete random variable κ. This random variable is called latent since it is not observed. κ ranges from 1 to K and indicates which component of the mixture is to be used. We therefore now consider the mixture law of X in the context of the joint distribution of (X, κ). Notice that this law is the marginal

law of X. Indeed that the random couple (κ, X) is governed by

$$\forall k, \Omega,\ \mathbb{P}(\kappa = k, X \in \Omega) = \alpha_k \int_\Omega G(\theta_k, x)dx$$

Now suppose that the latent variable κ is observed, let us note $(\mathcal{K}, \mathcal{X}) = \{(\kappa_i, X_i)_i\}$ this virtual sample and let $\mathcal{X}_k$ be the sub-sample of $\mathcal{X}$ of the data x_i for which the associate latent variable k_i is equal to k, let N_k be the size of $\mathcal{X}_k$. Then it is easy to get the maximum likelihood estimation for θ_k and α_k, namely

$$\hat{\theta}_k = \arg\max_\theta l(\theta; \mathcal{X}_k), \qquad \hat{\alpha}_k = \frac{N_k}{N}$$

Actually, we do not know the k_i's whenever they have a physical existence. If the parameter set $\theta = \{\theta_k, \alpha_k\}$ was known, k_i could be recovered through a Bayes posterior estimation

$$\gamma_{i,k}(\theta) = \mathbb{P}(K_i = k \mid \theta, X = X_i) \tag{4.10}$$

which is called the *responsibility* of model k for observation i. Indeed, an estimation of the latent variable may be estimated through the MAP (maximum a posteriori) estimation:

$$\hat{k}_i = \arg\max_k \gamma_{i,k}(\theta)$$

A similar method is used in the proposed technique for "hard clustering". We can now derive the EM algorithm as an iterative relaxation of these two steps. Let us describe the $(n+1)$th iteration

1. Take the estimates at the previous step $\{(\hat{\theta}_k^{(n)}, \hat{\alpha}_k^{(n)})_k\}$
2. Expectation Step. Compute associate responsibilities $\hat{\gamma}_{i,k}^{(n)}$ for $i = 1 \dots N,\ k = 1 \dots K$:
$$\hat{\gamma}_{i,k}^{(n)} = \frac{\hat{\alpha}_k^{(n)} g_{\hat{\theta}_k^{(n)}}(x_i)}{\sum_{j=1}^K \hat{\alpha}_j^n g_{\hat{\theta}_j^n}(x_i)} \tag{4.11}$$
3. Maximization Step. Compute the weighted maximum likelihood estimators for each component of the mixture:
$$\hat{\theta}_k^{(n+1)} = \arg\max_\theta \left[\sum_{i=1}^N \hat{\gamma}_{i,k}^{(n)} \log G(\theta, x_i) \right] \tag{4.12}$$
$$\hat{\alpha}_k^{(n+1)} = \frac{\sum_{i=1}^N \hat{\gamma}_{i,k}^{(n)}}{\sum_{j=1}^K \sum_{i=1}^N \hat{\gamma}_{i,j}^{(n)}} \tag{4.13}$$

The algorithm is decreasing in a monotonous way but convergence towards local minima is possible in the general case (see (Wu 1983)).

4.2.5 Towards supervised clustering: regression

So far, we described mixture laws and clustering from an unsupervised point of view. In this section, we outline the path from unsupervised learning towards supervised learning and more precisely the methods that we will describe in details in the next section. In fact, this technique is part of semi-supervised technique since it does use unsupervised learning tools to achieve supervised learning.

We now assume that we have $(x_i, y_i)_{i=1\dots N}$ a set of N observed data points from a random variable $Z = (X, Y)$, where X is vector-valued random variable lying in $\mathbb{R}$ and Y a random variable lying in $\mathbb{R}$. Instead of solely modeling the marginal law of X, we can consider Z as a vector-valued random variable lying in $\mathbb{R}^{p+1}$ and model its density. Once this density has been estimated we can easily derive the marginal law of X. To illustrate our point, we will take the example of Gaussian mixtures of K components and get the marginal law. Suppose we have set the number of clusters to K. We estimate through EM algorithm the parameters of the K multivariate Gaussians in $\mathbb{R}^{p+1}$ such that

$$Z \sim \sum_{k=1}^{K} \alpha_k \mathcal{N}(\mu_k, \Gamma_k), \tag{4.14}$$

where the α_k's are the mixture parameters, i.e for all $k \in i \dots K$, $\alpha_k \in [0, 1]$ and $\sum_{k=1}^{K} \alpha_k = 1$ and $\mu_k \in \mathbb{R}^{p+1}$ is the mean of the Gaussian distribution k and denote

$$\mu_k = \begin{pmatrix} \mu_k^X \\ \\ \mu_k^Y \end{pmatrix}, \tag{4.15}$$

where $\mu_k^X \in \mathbb{R}^p$ is the $X-$coordinates of the mean μ_k and $\mu_k^Y \in \mathbb{R}$ is the $Y-$coordinate of the mean μ_k. Γ_k is the variance-covariance matrix of size $p+1$ and denote

$$\Gamma_k = \begin{pmatrix} \Gamma_k^X & \nu_k \\ \\ \nu_k^T & \xi_k \end{pmatrix}, \tag{4.16}$$

where Γ_k^X is the covariance-variance matrix of X for Gaussian k, $\nu_k \in \mathbb{R}^p$ is $\mathrm{Cov}(X, Y)$ for Gaussian k and $\xi_k = \mathrm{Var}(Y) \in \mathbb{R}$.

From this, we can derive the marginal law of X. Recall that κ is the associated latent variable added to Z to mean which component it belongs to. More precisely, $\kappa_i = k$ means that the i^{th} data belongs to component k. From the conjoint law of (X, Y), we can derive the law of $X|\kappa = k$

$$X|\kappa = k \sim \mathcal{N}(\mu_k^X, \Gamma_k^X) \tag{4.17}$$

and since

$$< X = x_i >= \cup_{k=1}^{K} < X = x_i \cap \kappa = k > \tag{4.18}$$

and finally, we have

$$X \sim \sum_{i=1}^{K} \alpha_k \mathcal{N}(\mu_k^X, \Gamma_k^X) \tag{4.19}$$

4.3 Posterior probabilities, hard and soft clustering and Bayes optimal boundaries

Once a mixture law has been fitted to the data,e.g all parameters have been estimated through EM algorithm, we first need to cluster the learning data and provide from these parameters a classifier that will be used for new data where outcome is unknown. This is done through maximum a posteriori estimation, we therefore need to estimate the probability for a new entry point to lie on each cluster without knowing its response.

4.3.1 Bayesian estimation of maximum a posteriori: probabilistic and geometric clusterings

We can now compute the posterior probabilities, that is to say, the probability for a given $(x, y) \in \mathbb{R}^{p+1}$ to lie within cluster k_i. It is given by Bayes' formula where κ denotes the discrete random variable associated with the clusters

$$\mathbb{P}(\kappa = k_i|(X, Y) = (x, y)) = \frac{\mathbb{P}(\kappa = k_i)\mathbb{P}((X, Y) = (x, y)|\kappa = k_i)}{\sum_{k=1}^{K} \mathbb{P}(\kappa = k)\mathbb{P}((X, Y) = (x, y) \mid \kappa = k)}.$$

For the particular case where (X, Y) is assumed to be a Gaussian mixture model : $(X, Y) \sim \sum_{k=1}^{K} \alpha_k \mathcal{N}(\mu_k, \Gamma_k)$ we have for all $k \in \{1, \ldots, K\}$

$$\begin{aligned} \mathbb{P}(\kappa = k) &= \alpha_k, \\ (X, Y) \mid (\kappa = k) &\sim \mathcal{N}(\mu_k, \Gamma_k), \end{aligned}$$

which leads with $z = (x, y)$ to

$$\mathbb{P}(\kappa = k_i | (X, Y) = (x, y)) = \frac{\det(\Gamma_{k_i})^{-\frac{1}{2}} \alpha_{k_i} e^{-\frac{1}{2}(z-\mu_{k_i})^T \Gamma_{k_i}^{-1}(z-\mu_{k_i})}}{\sum_{k=1}^{K} \det(\Gamma_k)^{-\frac{1}{2}} \alpha_k e^{-\frac{1}{2}(z-\mu_k)^T \Gamma_k^{-1}(z-\mu_k)}}. \tag{4.20}$$

The a posteriori probability in Equation (4.20) represents a fuzzy probabilistic clustering. Indeed, being a probability, this quantity need not be 0 or 1, it represents the probability for a given observation (x, y) to be explained by law k_i. It therefore offers a simple soft clustering of the observed data. Indeed, consider

$$(\mathbb{P}(\kappa = k_i | (X, Y) = (x, y)))_{k_i \in [1, \ldots, K]} \tag{4.21}$$

which is a vector in $\mathbb{R}^K$ with components lying in $(0, 1)$ and we say that (x, y) is **soft-clustered** with respect to the mixture law, the point is given a probability to lie in each cluster defined by the mixture law. But we could have said that one point should lie in one cluster and only one, this will lead us to the hard clustering version. We first describe the hard clustering version from the probabilistic point of view and then describe it from a geometric point of view by means of Mahalanobis distance. These two approaches do not coincide. Actually, the probabilistic one is the more general since not all probability distributions induce distances (e.g stable law). Nonetheless, in the case of K-means clustering they do coincide. We first begin with the probabilistic approach instead of giving each point a set of K probabilities to lie in all clusters, we may want to assign each point to one and only one cluster. Being probabilities, we say that the cluster i an observation (x, y) lies in is simply the law i that gives the highest posterior probability

$$i := \text{argmax}_{k_i \in [1, \ldots, K]} \mathbb{P}(\kappa = k_i | (X, Y) = (x, y) \tag{4.22}$$

At this point, we define so-called geometric clusters with respect to the mixture law, as said earlier this geometric point of view can not be applied to all mixture laws, since

we need to define a distance from the laws parameters, if we think of a non-parametric law, such as stable distributions which are not Levy, normal or Cauchy . Assume again that our mixture law is a Gaussian mixture. We can define the **Mahalanobis distance** associated to component law i of parameters (μ_i, Γ_i) which is

$$D_M(\omega_1, \omega_2) = ||\omega_1 - \omega_2||_{\Gamma^{-1}} = \sqrt{(\omega_1 - \omega_2)^T \Gamma_i^{-1} (\omega_1 - \omega_2)} \tag{4.23}$$

where ω_1 and ω_2 are in $\mathbb{R}^p$. It does define a distance in $\mathbb{R}^p$ since the inverse of the variance-covariance matrix Γ_i is also a positive definite symmetric matrix. We then define the geometric cluster i in the set of observations as all the observations that are the closest to the center i (i.e component mean μ_i) with respect to the Mahalanobis distance inherited by the variance-covariance matrix Γ_i.

We can now go back to K-means algorithm and interpret from these points of view and make the probabilistic approach resemble the geometric one. Indeed, in section 1, we presented K-means as a geometric algorithm, but if we assume that the data come from the following mixture

$$X \sim \sum_{i=1}^{K} \alpha_i \mathcal{N}(\mu_i, \sigma^2 Id) \tag{4.24}$$

where the weights are the same $\alpha_i = \frac{1}{K}$ and $\sigma \in \mathbb{R}$. We see that the probabilistic hard clustering for an observation X assigns it to cluster i for which the Mahalanobis distance (which shrinks down to to the Euclidean distance) is the minimum. This is precisely the way, an observation is given a cluster in standard K-means. To follow this geometric interpretation, we can say that K-means finds centers of balls that best capture the structure of the data, while **EM clustering generalizes this by allowing the balls to be mapped into ellipsoids** (where the half-axes directions are the eigenvector of the variance-covariance matrix and their length are governed by the inverse of the corresponding values) and also to be weighted amongst them. The other way around is to explicit K-means as a probabilistic algorithm, this allows us to derive a soft version of K-means where each point is given a set of K probabilities. This gives insight to K-means algorithm and also explain some of the drawbacks of this algorithm, since the we assumed that the mixture law is $\sum_{i=1}^{K} \alpha_i \mathcal{N}(\mu_i, \sigma Id)$, that is all laws share the same variance-covariance matrix $\sigma^2 Id$ which is diagonal. This strong hypothesis is known in

statistics as **homoscedascity**, this is quite a strong hypothesis since the underlying assumption for the data is that dimensions are not correlated.

4.3.2 Practical issues : choice of the number of clusters

The choice of the number of components is extremely important. We present here three strategies to set this number. The first two ones are classical tools from unsupervised learning where the number of clusters is roughly speaking chosen through maximum likelihood estimation penalized by the number of parameters to estimate (AIC, BIC). Cross-validation and bootstrap may be also used to estimate the best number. A more recent suggestion from Tibshirani et al. in (Tibshirani *et al.* 2001) known as the **gap curve** is described. We will go back to this issue by presenting an original way which finds the best number of clusters to the end of regression by means of simple and cheap predictors (linear and experts).

4.3.2.1 Classical tools : AIC, BIC

We briefly introduce classical metrics to assess the quality of the fit for a given number of clusters and then design procedures to derive the best number (if any) of clusters. First, recall the basis metrics to estimate the goodness of a fit is the log-likelihood since it is directly the probability to observe the data under the parameters. This explains why these metrics will be used in most procedures to find the best number of clusters. First, let us define the **Akaike Information Criterion** and the **Bayesian Information Criterion** as a log-likelihood weighted by the number of free parameters to set, in (Burnham & Anderson 2004), these two metrics are thoroughly described. The underlying assumption of that the model that best explains the data is the simplest with respect to the number of free parameters. This relies on a more general assumption usually known as the Occam's razor that roughly speaking says that it is always better to reduce the number of entities when the additional cost is moderate. This simple assumption turns out to be in the statistical learning the same idea as limiting the number of parameters to prevent the model building from overfitting. Historically, the first criterion was the Akaike Information Criterion (AIC) defined as

$$AIC = 2k - 2l(\theta; X) \tag{4.25}$$

where k denotes the number of free parameters. Bayesian Information Criterion penalizes more the number of free parameters and also makes use of the number of observations in the learning sample, this criterion (BIC) is defined as

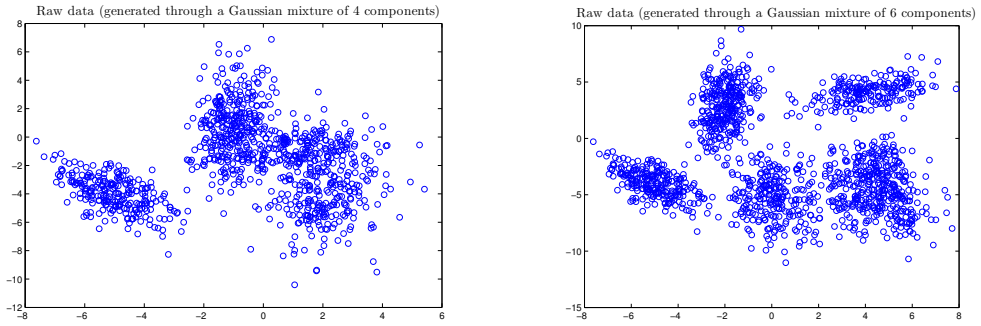
$$BIC = k\log(n) - 2l(\theta; X) \tag{4.26}$$

where n is the number of observations. Again, when comparing two models, as in our case the mixture density estimation for different number of components, with these criteria we favor the ones with lower AIC or BIC. It is worth noting that, most of modern implementations of EM algorithm or EM clustering (e.g MIXMOD, EMMIX,...) offer AIC or BIC to choose the best number of clusters.

4.3.2.2 Gap statistics

Another approach suggested in (Tibshirani *et al.* 2001) is to compare the different metrics for the observed sample with the same metrics fir data lying in the same design space but uniformly generated. For instance, the idea is to compare the two log-likelihoods for a varying number of clusters and more specifically the difference between them. Indeed, when computing the log-likelihood of the uniform data which has not specific structure and so which should not be well explained by finite mixtures, we are likely to detect a gap (or in statistics jargon an 'elbow') in the differences between the two curves. This gap is expected to be the highest for the best number of clusters.

To illustrate these different criteria, we used the following examples. We generated samples from a Gaussian mixture of 4 components and 6 components to see whether or not the correct number of components can be detected through the use of log-likelihood, AIC, BIC and their corresponding gap curve. Raw data are represented for both mixtures in Fig. 4.1. These are mixtures of different numbers of components in $\mathbb{R}^2$, the associated samples occupies a domain $\Omega = [x_{min}x_{max}] \times [y_{min}y_{max}]$. We generated in each case the same number of samples uniformly distributed. These samples are referred to as artificial data. For $i = 2 \dots K_{max}$ clusters, the EM algorithm is applied both on the real data and on the artificial data and the associated criteria are computed. Results of these computations are depicted Fig. 4.2. We first observe that the gap curves associated to both AIC and BIC are obviously the same for they are defined up to an additive constant. In both cases, we observe that the maximum of the gap



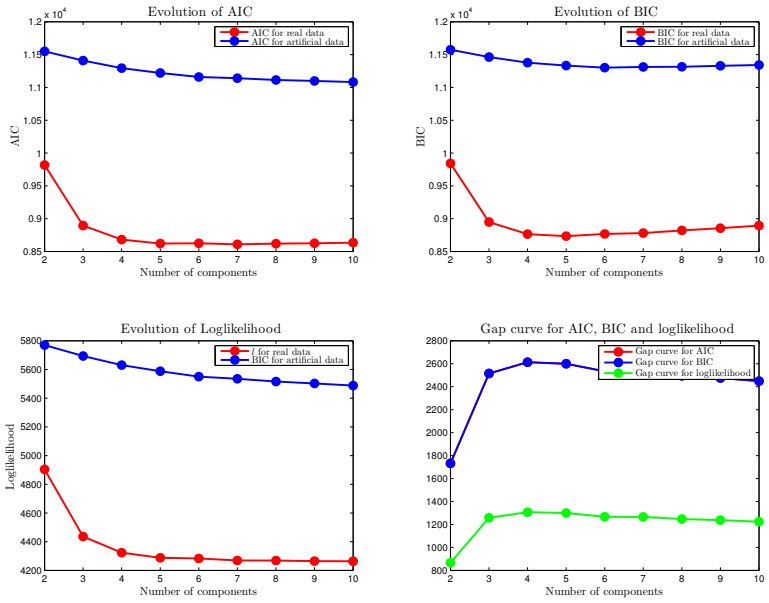
(a) Gaussian mixture of 4 components (b) Gaussian mixture of 6 components

Figure 4.1: Two different mixtures of Gaussians in $\mathbb{R}^2$

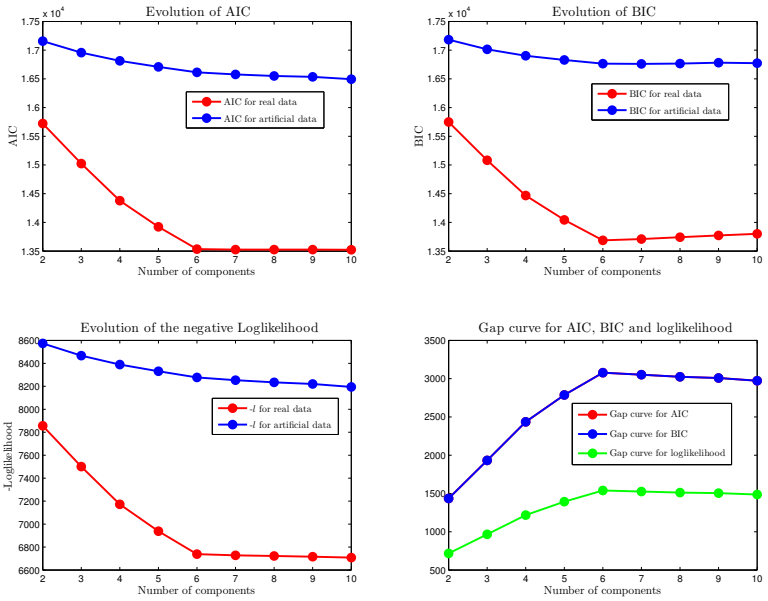
curves are given for the right number of clusters, even though it does not make a narrow peak around the correct number of clusters. This even more flat for log-likelihood.

4.4 Building the overall expert for the regression case

In this section, we most specifically turn on the case of regression and show how the mixture density estimation (e.g for Gaussian mixture for instance) previously described from an unsupervised point of view can be used to derive an algorithm for supervised learning. We will focus to regression but the latter can be also applied to classification problems. We will keep the same notations as in the previous sections and assume we have $(x_i, y_i)_{i=1...N}$ a set of N observations, we will also consider that at this point mixture density estimation has been done through EM algorithm and that we then have an estimation of the mixture parameters. We do not need to restrict ourselves to the case of Gaussian mixtures though we use them as an example since we can derive easy analytical formulas from Gaussian mixture and also a geometric interpretation from the Mahalanobis distance. Nevertheless, we can derive the same procedure for arbitrary distributions with analytical probability density functions, since at the very end, we need first to maximize log-likelihood and hence need analytical formulas for this density, and second to be able to write the marginal distribution for X (without knowing the outcome Y). For instance, the same procedure can be easily derived for Students or χ^2 distributions. We should however note that the computation of the marginal law might not be as straightforward as in the Gaussian case, but this can be



(a) AIC, BIC, log-likelihood and associated gap statistics for the case of a mixture of 4 Gaussians



(b) AIC, BIC, log-likelihood and associated gap statistics for the case of a mixture of 6 Gaussians

Figure 4.2: Usual criteria for assessing the goodness of the fit: AIC, BIC, log-likelihood and associated gap curves. These gap curve are obtained by taking the difference between each criterion for the raw data and the corresponding criterion for artificial uniform data

still approximated numerically.
Note that at the very end, we want to be able to estimate

$$\mathbb{E}(Y|X=x) \tag{4.27}$$

which is the expectation of Y knowing that the entry is x. Suppose now that we hard-clustered our learning data (joint data) into clusters $(C_i)_{i=1...K}$, and that we build a surrogate model $f_i(x) = \mathbb{E}(Y|X=x, \kappa=i)$. We can now write

$$\mathbb{E}(Y|X=x) = \sum_{i=1}^{K} \mathbb{E}(Y|X=x, \kappa=i)\mathbb{P}(\kappa=i|X=x) \tag{4.28}$$

where

$$\mathbb{P}(\kappa=i|X=x) = \frac{\mathbb{P}(\kappa=i)\mathbb{P}(X=x|\kappa=i)}{\sum_{j=1}^{K}\mathbb{P}(X=x|\kappa=j)} \tag{4.29}$$

The final approximation model is simply described by Eq. (4.28). In the rest of this section, we describe the different steps to achieve to get this final model and a formal algorithm that can be used in any context (classification or regression). We though restrict test cases and examples to regression.

4.4.1 Building local expert

The learning basis is split into K learning sub-bases and an expert f_k is trained over each sub-basis k. Any surrogate model can be used, we give results obtained using the following different local experts :

- linear regression: the most simple expert (apart from the constant expert which leads to radial basis functions regression). It can be computed easily and the multivariate linear regression parameters are directly given by the Gaussian component parameters

$$\begin{aligned} f_k(x) &= \frac{\mathrm{Cov}^k(X,Y)}{\mathrm{Var}^k(X)}(x - \mathbb{E}^k(X)) + \mathbb{E}^k(Y) \\ &= (\Gamma_k^X)^{-1}\nu_k(x - \mu_k^X) + \mu_k^Y. \end{aligned} \tag{4.30}$$

  In that case, once the EM algorithm is done, all the parameters of the Mixture of Experts (MoE) are computed. In this particular case, clustering and learning are not separated as in (Jordan & Jacobs 1994). This MoE is therefore very cheap

to compute. Note that a numerical instability (apart from EM algorithm that might converge very slowly) can arise from the inversion of the variance-covariance matrices Γ_k (for clustering) and Γ_k^X (to build local experts and combine them). This should be done carefully using for instance QR factorization[1].

- quadratic regression: the original response surfaces, which are quadratic polynomials over $\mathbb{R}^d$ as extensively described in (Myers *et al.* 2009). They are also relatively inexpensive to build however there is no easy formula that can be derived from the Gaussian components parameters. In our case, we computed it in a simple way taking care of the inversion of the system.

- artificial neural networks: we use here the classical Multi Layer Perceptron (MLP) as a local expert; MLP models are thoroughly described in (Haykin 2008). We use one-layer networks and the number of hidden neurons is classically determined through a cross-validation procedure and the network is trained using Levenberg-Marquardt algorithm.

- moving least squares (MLS): the MLS expert is the most complicated expert to compute for it is an implicit model that needs to be recomputed at each new evaluation point. We implemented a moving least squares method based on the Backus-Gilbert approach that can be found in (Fasshauer 2005). We also implemented a golden ratio search to optimize the hyper-parameter σ (width of the Gaussian kernel, see appendix B). A landmark paper on MLS is (Levin 1998) and a brief introduction can be found in (Nealen & Darmstadt 2004).

- radial basis function (RBF): radial basis functions models are linear combination of radial functions (e.g Gaussian, thin plate...), these radial functions are usually centered around learning points and the coefficients of the linear combination are found through least-squares minimization. In case there are as many radial functions as learning points (each radial being centered at each point), the linear system to be minimized through least-squares happens to be square and the RBF models interpolates the learning points. An important aspect of these two models

[1] All these matrices are symmetric positive definite but they can become nearly-singular especially in case of redundant data (linearity), QR factorization performs better than Gaussian reduction and even Choleski factorization. A good introduction to these methods can be found in (Saad 1992)

(MLS and RBF) is that they are the basis of meshless and pseudo-spectral methods in the related field of approximation theory of pde's. Another very important aspect of these approximation models is that they feature, provided the function f to approximate is smooth (e.g infinitely continuously differentiable), **spectral convergence in $||.||_\infty$ norm**, which is convergence faster than any polynomial convergence (e.g typical convergence rate is $O(N^{-m})$) for any m where N is the number of learning points) and in case the function f to approximate is analytical (e.g can be developed as a power series everywhere on its definition domain), the convergence is even faster and is $O(c^N)$, where $c \in (0, 1)$. A good overview is these specific methods can be found in (Fasshauer 2007) and (Wendland 2005). However, it should be noted that the functions to approximate are usually not regular (e.g buckling) and this spectral convergence results are usually derived in 'ideal' conditions such as simple domain or input space (rectangle or hypercubes) and then do not fit our requisites.

Nonetheless, any surrogate model can be used (kriging, support vector regression, multivariate adapted regressive splines) as a local expert and can be perfectly improved using an ensemble of surrogate models, or boosting (Meir & Ratsch 2003). Moreover, a local expert can be a black-box or itself an MoE and so on.

4.4.2 Combining local experts

Once we build our local experts f_k, we want to predict the response y for a new entry $x \in \mathbb{R}^d$. This is done by combining them. We form a linear combination of the local experts f_k. A natural idea is that this linear combination should not be constant over the whole input space and should give more weight to expert f_k when x gets closer to the center of cluster k with respect to the natural Mahalanobis distance inherited by the variance-covariance matrix of Gaussian k. Namely the global model $\hat{f}$ is

$$\hat{f}(x) = \sum_{i=1}^{K} \beta_i(x) f_i(x), \tag{4.31}$$

where $\beta(.) = (\beta_1(.), \ldots, \beta_K(.))$ is an expert that gives the local weights (local in the sense that it does depend on x). A natural gating network would be

$$\beta(x) = (\mathbb{P}(\kappa = 1|X = x), \dots, \mathbb{P}(\kappa = K|X = x)), \tag{4.32}$$

such that the global model would be

$$\hat{f}(x) = \sum_{i=1}^{K} \mathbb{P}(\kappa = i|X = x) f_i(x). \tag{4.33}$$

Equation (4.33) is the classical probability expression of mixture of experts (as it can be found in (Jordan & Jacobs 1994)). Note that this expression may represent a lot of different situations and therefore a lot of different MoE's. For instance, the weighted least squares can be interpreted as an Moe using that equation . To use Equation (4.33) we need to know what is the law of κ knowing that $X = x$ and without knowing that $Y = y$, this can be easily obtained with the Gaussian parameters found by EM algorithm. Indeed, from the conjoint law $(X, Y) \sim \sum_{k=1}^{K} \alpha_k \mathcal{N}(\mu_k, \Gamma_k)$, we can derive the law of $X|\kappa = k$ without knowing Y

$$X|\kappa = k \sim \mathcal{N}_d(\mu_k^X, \Gamma_k^X), \tag{4.34}$$

such that the global Gaussian Mixture Model law of X is

$$X \sim \sum_{k=1}^{K} \alpha_k \mathcal{N}(\mu_k^X, \Gamma_k^X). \tag{4.35}$$

Note that this Gaussian Mixture Model is different from the one we would have obtained by applying EM only on the inputs X's for it is the projection on the input space of the conjoint law. Therefore we can derive equivalently the posterior probability from Bayes' formula

$$\mathbb{P}(\kappa = k_i|X = x) = \frac{\det(\Gamma_{k_i}^X)^{-\frac{1}{2}} \alpha_{k_i} e^{-\frac{1}{2}(x-\mu_{k_i}^X)^T \Gamma_{k_i}^{X\,-1}(x-\mu_{k_i}^X)}}{\sum_{k=1}^{K} \det(\Gamma_k^X)^{-\frac{1}{2}} \alpha_k e^{-\frac{1}{2}(x-\mu_k^X)^T \Gamma_k^{X\,-1}(x-\mu_k^X)}}. \tag{4.36}$$

Note that the global model defined with Equation (4.33) and Equation (4.36) is completely smooth. In the sequel, this mixture of experts will be referred to as **smooth mixture**. At this point, we can think of another way of combining the local surrogate models that takes more advantage from the clustering made by the EM algorithm. Indeed, based on the Gaussian parameters estimated on the clustering step, we can

predict which cluster a new given entry x lies in and simply assign to this entry x the corresponding local surrogate model. This means that we can, at least formally, define a partitioning of the whole input space: $\mathcal{X} = \cup_{k=1}^{K} \mathcal{X}_k$ and simply define the law of k knowing that $X = x$ as a uniform discrete law such that the global model would be

$$\hat{f}(x) = \sum_{k=1}^{K} \chi_{\mathrm{Cl}_k^X}(x) f_k(x) \tag{4.37}$$

where

$$\chi_{\mathrm{Cl}_k^X}(x) = \begin{cases} 1 & \text{if} \quad k = \mathrm{argmax}_{j=1,\dots,K} \mathbb{P}(\kappa = j | X = x) \\ 0 & \text{if} \quad k \neq \mathrm{argmax}_{j=1,\dots,K} \mathbb{P}(\kappa = j | X = x) \end{cases} \tag{4.38}$$

and Equation (4.42) defines the most simple mixture of experts where a new entry x is given a cluster k and the predicted value is simply $f_k(x)$. An important feature of this mixture of experts is that it is not continuous, indeed at the boundary between two adjacent clusters[1] the two adjacent local surrogate models need not to match resulting in a **globally discontinuous model**. This mixture of experts will be referred to as **hard mixture of experts**. In the case when the functions to approximate is discontinuous or derivative-discontinuous the hard mixture version is likely to be more accurate than the smooth version. Besides being discontinuous, the hard mixture version may create artificial discontinuities where the original function does not have ones (see Fig. 2. d)). In this article, we are mainly concerned with approximating functions that are objective or constraints functions of an optimization problem that is to be solved on the basis of a gradient-based method, this is why we will focus on the smooth mixture of experts. Indeed, in such applications, we are not only concerned with the accuracy of the approximation but also with the regularization of the original approximation. In terms of accuracy the hard version is likely to perform better (see for instance Fig. 2. d)) but the optimization algorithm may fail to converge due to the non-differentiability of the global approximation model.

Our algorithm, called **IMAGE (Improved Metamodeling Approximation through Gaussian mixture of Experts)**, is presented here and results of this method are given in the following sections with the help first of simple test cases and buckling computations approximations. This method is also illustrated step-by-step on a one-dimensional test case in Fig. 4.3.

[1] This boundary is often known in Probability as the **optimal Bayes classifier**

1. Assemble Z the conjoint learning basis where $z_i \in \mathbb{R}^{d+1}$ contains inputs $x_i \in \mathbb{R}^d$ and output $y_i \in \mathbb{R}$, see Fig. 4.12. a)

$$Z = \begin{pmatrix} x_1^{(1)} & \dots & x_n^{(1)} \\ \vdots & & \vdots \\ x_1^{(d)} & \dots & x_n^{(d)} \\ y_1 & \dots & y_n \end{pmatrix}. \tag{4.39}$$

2. Set the number of clusters K. In Fig. 4.3 the number of clusters was set to 3.

3. Apply EM algorithm to Z with K to get $\hat{\alpha}_k$, $\hat{\mu}_k$ and $\hat{\Gamma}_k$, estimates of the real Gaussian parameters.

4. Hard clustering of the data, see Fig. 4.3 b). $z_i = (x_i, y_i)$ belongs to cluster k_i where

$$k_i = \text{argmax}_{j=1,\dots,K} \mathbb{P}(j|(X,Y) = (x_i, y_i)). \tag{4.40}$$

   where $\mathbb{P}(\kappa = j|(X,Y) = (x_i, y_i))$ is computed using Equation (4.20)

5. Split the conjoint learning basis into K clusters $Z = \cup_{i=1}^{K} Z^{(i)}$.

6. For $i = 1 \dots K$

   a) Remove outliers using for instance Mahalanobis distance.

   b) Split randomly $Z^{(i)} = Z^{(i)}_{learn} \cup Z^{(i)}_{test}$ into learning basis and test basis.

   c) Train local expert f_i on $Z^{(i)}_{learn}$, choose the best expert f_i with $Z^{(i)}_{test}$, see Fig. 4.3 c).

7. Combine all the experts for the smooth version with

$$\hat{f}(x) = \sum_{i=1}^{K} \mathbb{P}(\kappa = i|X = x) f_i(x). \tag{4.41}$$

   where $\mathbb{P}(k = i|X = x)$ is computed using Eq. (4.36), and the hard mixture with

$$\hat{f}(x) = \sum_{k=1}^{K} \chi_{\text{Cl}_k^X}(x) f_k(x) \tag{4.42}$$

   see Fig. 4.3 d) where we plotted both smooth and hard versions of the mixture of experts.

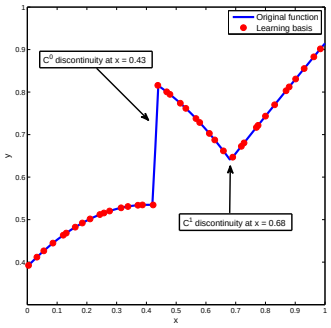
8. Estimate the generalization errors with the help of another data base B_V (validation basis) if available. If not, cross-validation or bootstrap techniques can be applied.

4.4.3 Simple analytical examples

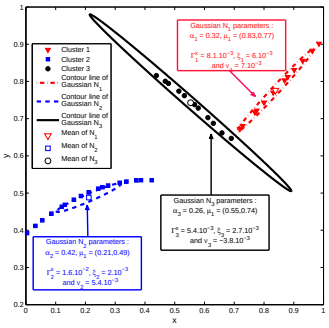
In this section, the performance of this algorithm is evaluated on very sample examples. Our main objective is to be able to detect and grasp region of continuity or smoothness of the function to approximate. We therefore applied our algorithm on two simple test cases, the first being the absolute value function over $[-1, 1]$, the second being a piecewise continuous function defined with three linear components. For the absolute value function, a Gaussian noise of equal variance all over the domain is added. In the second case, a Gaussian noise is added with different variances for each linear component. To assess the goodness of the fit, we compared with overall approximation models: quadratic regression and artificial neural network (built all over the domain). Results are depicted Fig. . In each case, the number of clusters was determined by finding the first 'elbow' of the gap curves. We observe that the clustering detects the region where the function is linear in each case. The local experts in these simple examples are set to be linear. However in each case, the hard recombination **creates artificial discontinuities** near kinks of the original function.

4.5 Application of this original strategy to buckling computations approximation

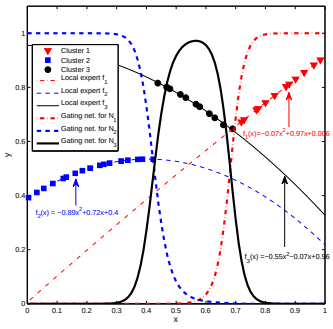
We end this section with a practical application test case of the algorithm previously described. This test case illustrates the approximation power of our EM clustering method. While initially derived to handle the approximation of functions that feature discontinuities and derivatives discontinuities, our method can be also applied, without any restriction, to perfectly smooth functions approximation to improve the accuracy of the approximation. Indeed, we build experts on small regions of the design space and then recombine them to get a global surrogate model in a smooth way, this method can achieve higher accuracy than a sole surrogate models due the localization of the experts. Indeed, an expert is, roughly speaking, located around the mean of the law



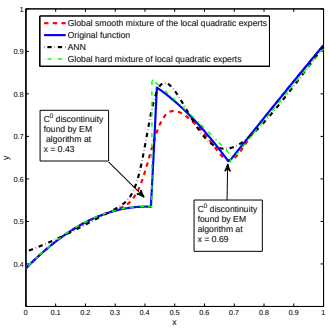
(a) `samgen_1D_1` test case and learning basis.



(b) Results of the EM clustering and associated Gaussian laws

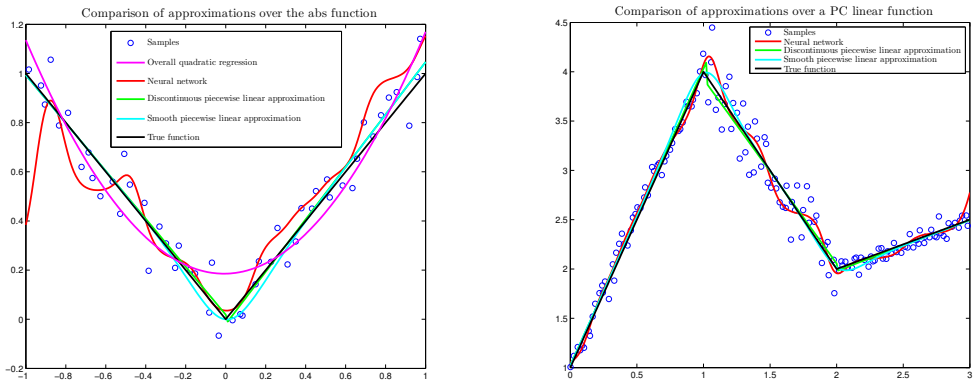


(c) Local quadratic experts and gating networks



(d) Global predictor

Figure 4.3: Sketch of the proposed algorithm for a 1D test case `samgen_1D_1` a) Original function and learning basis. Note that `samgen_1D_1` features two $\mathcal{C}^0$ and $\mathcal{C}^1$ discontinuities. EM clustering is expected to separate them well enough to locally build an accurate surrogate model.
b) EM clustering on the learning basis for $K = 3$. We depict the contour lines of the quadratic function associated to the variance-covariance matrix Γ_i at 3×10^{-6} for $i = 1 \dots 3$. Note that these lines are simply the balls of radius 3×10^{-6} centered at μ_i for the 3 Mahalanobis distances associated with the variance-covariance matrices.
c) Local quadratic experts and gating networks. See that the gating network associated to cluster 2 is quite steep. This is due to the relative small size of cluster 2.
d) Global surrogate models obtained through the mixture of local quadratic experts. We plotted the soft and the hard mixture versions. We also depicted a reasonably good artificial neural network to compare. We observe that the smooth predictor is very accurate on clusters 1 and 3 is a little bit less accurate on cluster 2. While the artificial neural network does not generalize very well at the boundaries of the domain, the global smooth predictor performs better at these boundaries due to the local behavior of the surrogate models. The hard mixture predictor is much more accurate since it does not regularize the discontinuities. It creates though an artificial $\mathcal{C}^0$ discontinuity at $x = 0.69$.

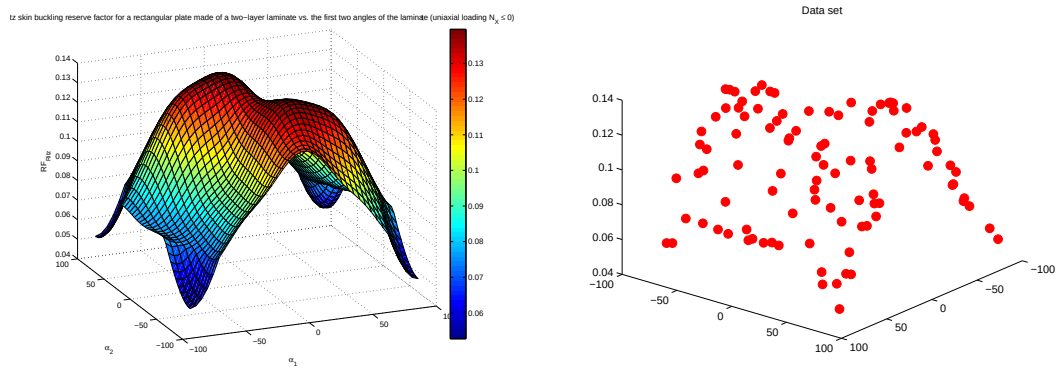


(a) Absolute value function with Gaussian noise (b) Piecewise linear function with Gaussian noise

Figure 4.4: Comparison of the hard and smooth piecewise linear approximation with overall approximation models over simple derivative-discontinuous functions

component it is associated with and the variance-covariance matrix **supports directions** (principal components of the data lying in the associated cluster) where this expert is to be **mainly applied**. Therefore, we can expect, several experts 'localized' and trained with fewer examples to perform better than a sole surrogate model trained with all the data, even in the case of smooth functions, where there is apparently no obvious need for clustering the learning basis.

This following test case is in the spirit of the buckling description of the former chapter, we want to approximate the critical buckling load factor λ_{cr} except that in this case the design variables are no more lamination parameters but directly ply angles of two-layer laminate. The aim of that test case is not only to illustrate the technique and show some results but also to illustrate that ply angles are not appropriate design variables for laminated composite design and optimization since we will see that buckling is not convex or concave w.r.t these variables. In this case recall that the partial differential equation that models linear buckling for a thin rectangular plate of width a and length b made of a two-layer laminate $[\alpha_1\backslash\alpha_2]$ simply-supported at the edges



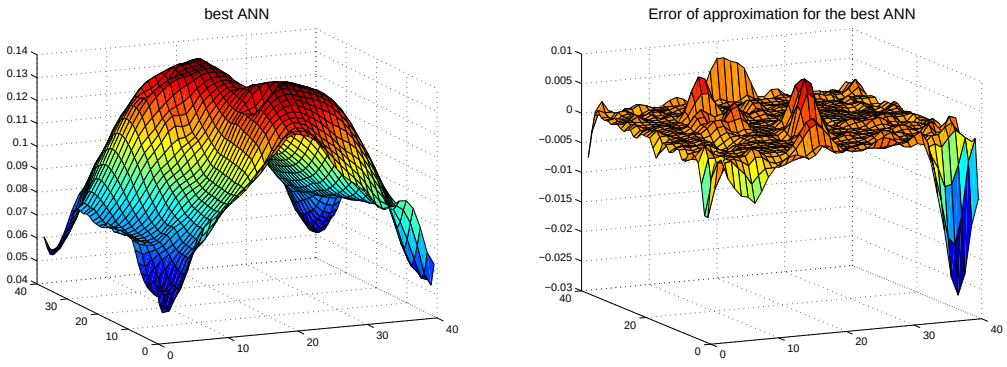
(a) Original function (b) Data base

Figure 4.5: Original function to approximate and data base

$$D_{11}(\alpha_1,\alpha_2)\frac{\partial^4 w}{\partial x^4} + 2D_{12}(\alpha_1,\alpha_2)\frac{\partial^4 w}{\partial x^2 y^2} + 2(D_{16}+D_{26})(\alpha_1,\alpha_2)(\frac{\partial^4 w}{\partial x^3 y} + \frac{\partial^4 w}{\partial y^3 x}) \quad (4.43)$$

$$+4D_{66}(\alpha_1,\alpha_2)\frac{\partial^4 w}{\partial x^2 y^2} + D_{22}(\alpha_1,\alpha_2)\frac{\partial^4 w}{\partial y^4} = N_x\frac{\partial^2 w}{\partial x^2} \quad (4.44)$$

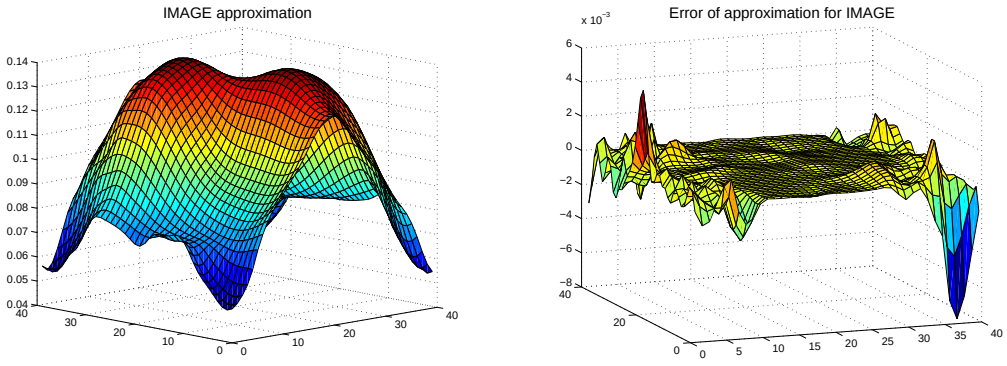
where w is the unknown transverse displacement, N_x the load case and D are stiffness terms that depend on a nonlinear manner on α_1, α_2. The buckling reserve factor (RF) is defined as being the first eigenvalue of 4.43. In some cases, when D_{16} and D_{26} are zero, analytical solutions can be derived, but in the general case, an approximation method is to be used. For that type of problem and geometry, Rayleigh-Ritz methods are often used since they provide an approximation of the buckling reserve factor faster than Finite Elements Methods. Note that Rayleigh-Ritz methods are often less precise than FEM but they are much easier to implement. In this regression case, the function to approximate was the reserve factor as a function of α_1, α_2. In Fig. B.3(b), we depicted the original function computed on a fine mesh, the learning samples we used to build surrogate models. We tried several neural networks for a different number of neurons and depicted Fig. 4.5 the best neural net estimated through cross-validation together with the generalization error over the fine mesh. Finally, we depicted Fig. 4.7 the surrogate model obtained with the original procedure we describe in this chapter. We clearly see that the error has the same distribution for both surrogate models but is lower in the case of our procedure.



(a) ANN approximation

(b) Error for the ANN

Figure 4.6: Best ANN approximation over the whole domain and error



(a) IMAGE approximation

(b) Error for IMAGE approximation

Figure 4.7: Approximation and error for the original method presented

4.6 Application to the approximation of buckling critical factor over lamination parameters and forces

In this section, we evaluate our original algorithm over the functions described and studied in Chapter 3. We want to approximate the buckling critical load factor λ_{cr} over lamination parameters and forces (loads per unit length) in case first of biaxial compression and also in case of combined loadings including shear loading. The resulting approximations models are then inserted into an optimization loop where we want to maximize λ_{cr} over lamination parameters. The results are also compared with the maximization of λ_{cr} over feasible stacking sequences and a strategy to find the optimal stacking from the continuous optimum is presented and applied.

4.6.1 Piecewise polynomial approximation in lamination space

The behavior of $\lambda_{cr}(\xi_D^1, \xi_D^2)$ for orthotropic laminates for combined loading (with low shear) seems to be piecewise linear (or close to be linear). We then want to approximate λ_{cr} based on a mixture of linear experts (referred to as MOL). The number of clusters is chosen through a range of potential clusters by minimizing the generalization error for the mixture of linear experts. We depicted Fig. 4.9 the result of the EM clustering for 4 clusters on the orthotropic data for combined loading. We see that the hyperplanes are correctly detected through the clustering.

For pure shear and non-orthotropic laminates, the phase portraits showed that the linear dependence does not hold, however λ_{cr} does not seem to behave in a very complicated way over each cluster. We then build an approximation of λ_{cr} on the basis of a Mixture of Quadratic models (MOQ).

4.6.2 Piecewise polynomial approximation in forces space

When the loading conditions vary, the dependence of $1/\lambda_{cr}$ seems to get less simple as linear as the plate endures less compression and more complex loading with tension for instance. However for pure compression loading cases, $1/\lambda_{cr}$ is close to be piecewise linear, we then build an approximation model as a mixture of quadratic models since it is expected to be able to handle different situations. It should be noted that if ordinary least squares are used, a linear model can not totally be recovered from a polynomial

regression due to the $\mathcal{L}_2$ minimization that regularizes polynomial coefficients. So more advanced techniques should be used to build purely linear experts and quadratic experts in different regions (best-subset selection, lasso, least angles regression, compressed sensing see the corresponding chapter in (Friedman *et al.* 2001)).

4.6.3 Piecewise rational approximation in the general case

Based on the behavior of λ_{cr} with respect to lamination parameters and of $1/\lambda_{cr}$ with respect to loads per unit length N, we want to build an approximation model as a mixture of (quadratic) rational experts

$$f_i(x) = \frac{a_0 + a_1\xi_D^1 + ... + a_{n-1}(\xi_D^2)^2 + a_n(\xi_D^3)^2}{b_0 + b_1 N_x + ... + b_{n-1}N_y^2 + b_n N_{xy}^2} \tag{4.45}$$

such models will be referred to as Mixture of Rationals (MOR).

4.6.4 Using directly Rayleigh-Ritz active buckling modes

It is worth noting when using Rayleigh-Ritz method, we usually get an extra information by looking at the eigenvector w_{cr} associated to λ_{cr}. Indeed, it can be written

$$w_{cr} = \sum_{i=1}^{N} \alpha_i u_i \tag{4.46}$$

where the u_i's are the basis functions. These basis functions can be associated to 'natural' modes (that is eigenfunctions of the unperturbed operator or over a more simple geometry). Suppose we take functions of the type

$$\sin(i\pi x/a)\sin(j\pi y/b) \tag{4.47}$$

We can associate each eigenvector w_{cr} to the 'closest natural mode' simply by finding the largest (in magnitude) component w_{cr}^{ij} of w_{cr}. However, this process can lead to wrong associated modes, in the case where we have two or more large components in w_{cr}, the numerical approximation is likely to invert the corresponding natural modes ('mode crossing'). Next there exists another difficulty in detecting and approximating mode per mode that is the veering phenomenon. Veering consists in two eigenvalues that get close and suddenly and abruptly diverge from each other (or veer). This sudden change makes the function more difficult to follow and the associated modes

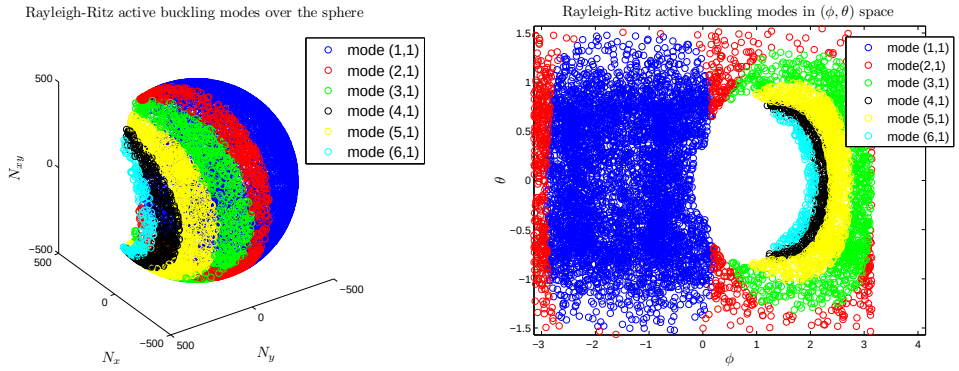
are no more orthogonal and starts to interact with each other, making impossible to follow one mode within the veering zone. Veering is a quite classical phenomenon in structural dynamics, but to the author's knowledge no veering example has been reported for buckling studies, however veering could be quite hard to detect and may be mixed up with crossing. Veering is reported to happen in any parameter dependent generalized eigenvalue problems. Next, finite element methods buckling do not provide such information, since the critical buckled shape (w_{cr}) is defined over local, mesh-dependent, non natural basis functions ($\mathcal{P}_1$ basis functions...). Post-treatment of the transverse displacement is needed when identifying the corresponding natural mode. However, in case this information is provided, this offers a natural way to approximate the critical buckling load by approximating mode per mode and building classifiers that gives the corresponding natural mode to a new input and hence apply the corresponding approximation model. This can be done since the regions that separates each natural buckling mode are usually quite simple in the input space, we depicted Fig. 4.8, the different natural modes when the flux vary both over the flux sphere and over spherical coordinates angles space (θ, ϕ), since the active mode does not depend on the norm of the flux. We clearly see Fig. 4.8 a) that the associated regions are quite simple over the sphere but for the (θ, ϕ) space Fig. 4.8 b) the regions are not easily separated, which prevents from building accurate classifiers. Such an approximation strategy could lead to interesting results but it is only possible when Rayleigh-Ritz are used and when we have access to the eigenvectors. This is not as general as the method we want to apply.

4.7 Numerical results

We describe the different results obtained with the suggested strategies to approximate λ_{cr} in the different situations

- For fixed loading: we approximate λ_{cr} first for orthotropic laminates (over ξ_D^1 and ξ_D^2) and then for [0/45/90] laminates (over ξ_D^1, ξ_D^2 and ξ_D^3) for both combined loading and pure shear loading.

- For a fixed non-orthotropic stacking sequence: we approximate λ_{cr} first for biaxial loading (over N_x and N_y) and then for combined loading (over N_x, N_y and N_{xy}).



(a) Associated natural buckling modes in the flux spaces

(b) Associated natural buckling modes in (θ, ϕ) space

Figure 4.8: Active buckling modes found by Rayleigh-Ritz method for different representations a) directly in the (N_x, N_y, N_{xy}) spaces for normalized flux b) in the spherical coordinates (θ, φ)

- In the general case when stacking and loading vary: we approximate λ_{cr} over ξ_D^1, ξ_D^2, ξ_D^3, N_x, N_y and N_{xy}.

In all situations, the learning stage of the approximation models is based on three different bases: learning basis B_A, test basis B_T and validation basis B_V. The learning in itself is directly based on B_A, to compare several approximation model built over B_A, we use B_T to rank them and chose the best one and finally at the end of this selection process, the generalization error and the other accuracy metrics are computed over the validation basis B_V. Following for instance (Friedman *et al.* 2001), we have a classical

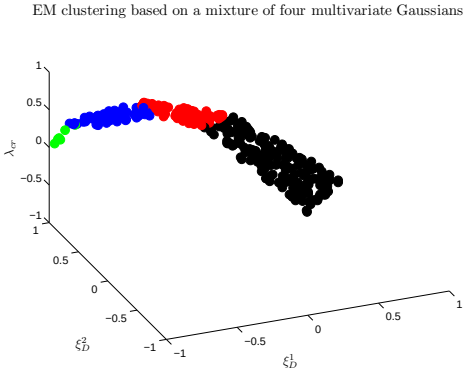
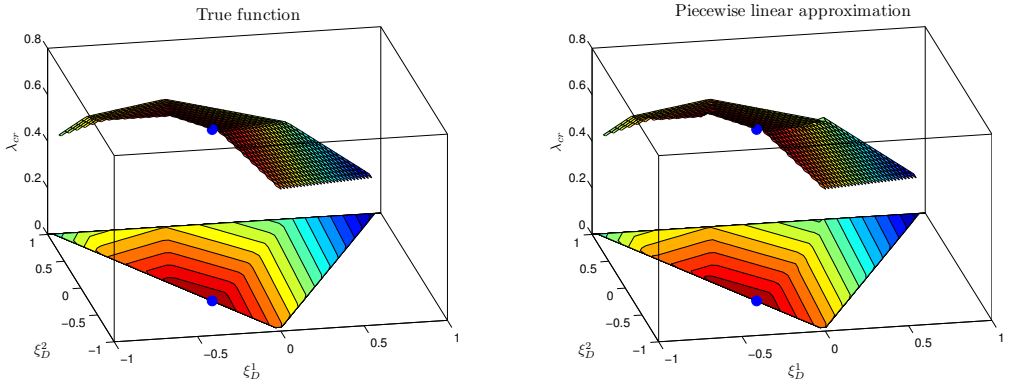


Figure 4.9: Automatic clustering (normalized output)



(a) Original function computed over a fine grid

(b) Piecewise linear approximation

Figure 4.10: Original function and piecewise linear approximation for buckling of orthotropic laminates

partition of 60 % for B_A, 20 % for B_T and 20 % for B_V. To assess the goodness of the fit, we use several classical metrics, amongst them, we use the *Mean Squared Error* (MSE) and *Lack Of Fit* (LOF) defined as

$$\text{MSE} = \frac{1}{N_V} \sum_{i=1}^{N_V} |\hat{f}(x_i) - y_i|^2 \tag{4.48}$$

and

$$\text{LOF} = \frac{\text{MSE}}{\text{Var}(\{y_i\}_{i=1\ldots N_V})} \tag{4.49}$$

where N_V is the number of points in the validation basis, y is the real output function and $\hat{f}$ is the approximation model. The LOF simply divides the MSE by the variance of the output to avoid scaling effect. However the knowledge of only the MSE might not be enough for accuracy evaluation. Indeed, first it does not provide any clue regarding the relative error and more important it is an average over a whole basis and does not give any insight on the distribution of the error. Suppose we have two different approximation models, one giving a very good approximation for a majority of points and being very bad for a small amount of points, it will be equivalent in terms of MSE to an approximation model that has a medium error for all points, while the first approximation model is likely to be preferable when used within a population-based algorithm (such as genetic algorithms). This is why, we add several metrics

- $\mathcal{L}_1$ and $\mathcal{L}_\infty$ errors:

- Mean relative error:

$$\hat{E}_{rel} = \frac{1}{N_V} \sum_{i=1}^{N_V} \frac{|\hat{f}(x_i) - y_i|}{|y_i|}$$

- Mean absolute error:

$$\hat{E}_{abs} = \frac{1}{N_V} \sum_{i=1}^{N_V} |\hat{f}(x_i) - y_i|$$

- Max relative error:

$$E_{rel}^{max} = \max_{i=1...N_V} \frac{|\hat{f}(x_i) - y_i|}{|y_i|}$$

- Max absolute error:

$$E_{abs}^{max} = \max_{i=1...N_V} |\hat{f}(x_i) - y_i|$$

- Cumulative error distribution.
  - $\alpha-$ quantiles:

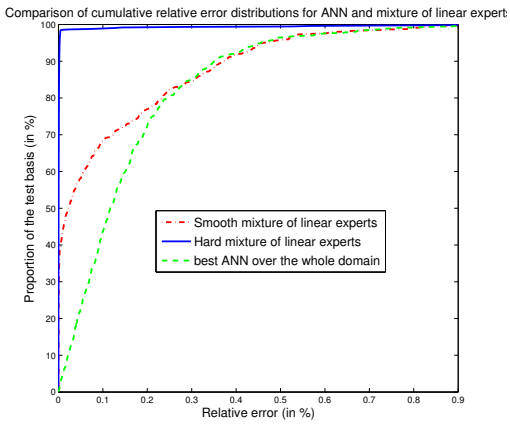
$$Q_\alpha = \frac{\text{Card}\{x_i \in B_V \quad \text{s.t} \quad \frac{|\hat{f}(x_i) - y_i|}{\hat{f}(x_i)} \leqslant \alpha\%\}}{N_V}$$

    the proportion of the validation points that are below a relative error of $\alpha\%$.
  - Quantile curve: curve obtained by making α varying from 0% to $E_{rel}^{max}\%$

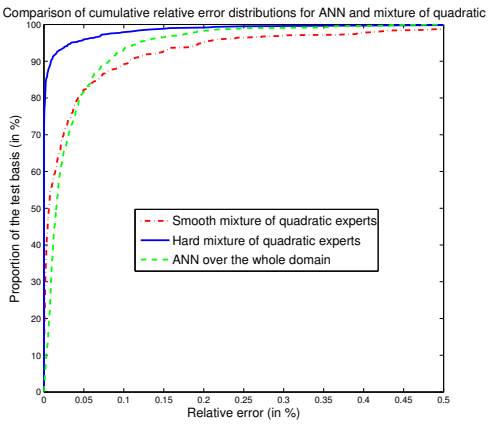
Note also that the number of points used in each case corresponds to the total points used for all bases. To the end of comparison, in each case we compared with a reasonably good artificial neural network built over the whole domain with classical techniques such as cross-validation for the choice of the number of hidden units.

4.7.1 Approximation models for varying stacking

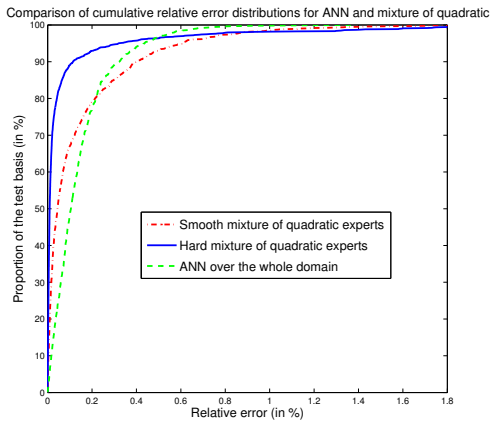
For the approximation of λ_{cr} for orthotropic laminates, we used $N = 420$ points for the first loading case (combined, $N_x = -50N.mm^{-1}$, $N_y = 20N.mm^{-1}$ and $N_{xy} = 20N.mm^{-1}$) and $N = 430$ points for the second loading case (pure shear). For the combined loading case, a mixture of linear experts was built and for the pure shear case, a mixture of quadratic models. We see Tab. 4.1 that the mixture models performs equally as the artificial neural network (ANN) for the pure shear and performs



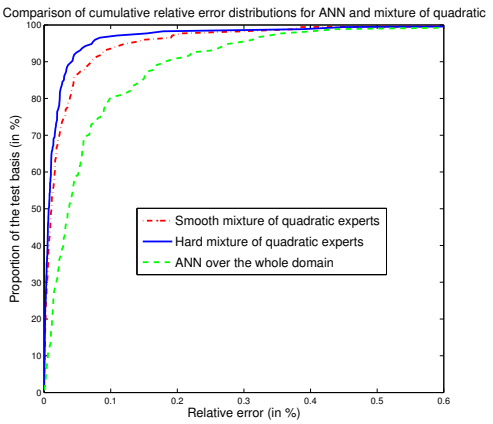
(a) Comparison of approximations of λ_{cr} over ξ_D^1 and ξ_D^2 on Miki's triangle for combined loading $N_x = -50N.mm^{-1}$, $N_y = 20N.mm^{-1}$ and $N_{xy} = 20N.mm^{-1}$



(b) Comparison of approximations of λ_{cr} over ξ_D^1 and ξ_D^2 on Miki's triangle for pure shear loading $N_x = 0N.mm^{-1}$, $N_y = 0N.mm^{-1}$ and $N_{xy} = 20N.mm^{-1}$



(c) Comparison of approximations of λ_{cr} over ξ_D^1, ξ_D^1, ξ_D^2 and ξ_D^3 for combined loading $N_x = -50N.mm^{-1}$, $N_y = 20N.mm^{-1}$ and $N_{xy} = 20N.mm^{-1}$



(d) Comparison of approximations of λ_{cr} over ξ_D^2 and ξ_D^3 for pure shear loading $N_x = N_y = 0N.mm^{-1}$ and $N_{xy} = 20N.mm^{-1}$

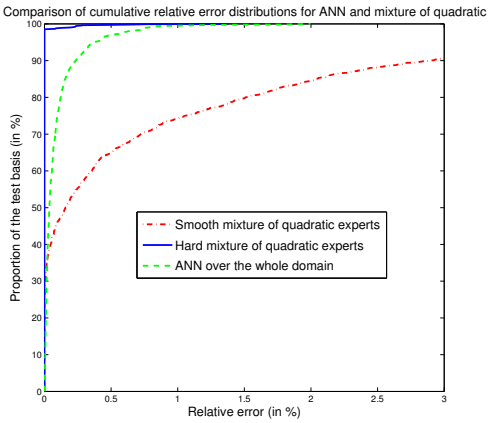
Figure 4.11: Quantile curves for orthotropic and non-orthotropic laminates under combined loading and pure shear loading

better for the combined loading. The quantiles curves shown in Fig. 4.12 and the quantiles $Q_{1\%}$ and $Q_{0.1\%}$ show that the hard recombination of mixture models even when they show the same accuracy in terms of MSE are much more accurate for a large amount of validation points. For combined loading for instance, the quantile curve is very close the y-axis, which means the approximation model is close to be exact but for a few points for which the mixture model gives bad accuracy. This might be due to misclassification, the wrong local expert is called for these few points, while the ANN regularizes the error over the whole domain. The ANN is of a reasonably good accuracy over all validation points but can not achieve very high accuracy even for a small amount of validation points. Regarding the relative error measures, the mixture models always perform better than ANN.

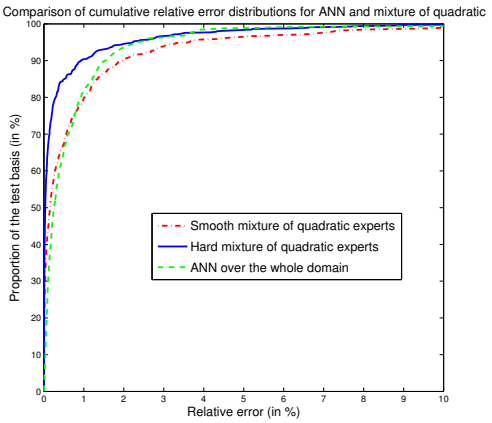
In the case of non-orthotropic laminates, $N = 853$ points were used for combined loading and $N = 768$ points in the case of pure shear loading. Results are in Tab. 4.2, again we observe the same behavior, the mixture models are by far better in terms of relative error while the MSE show the same tendency. The quantile curves clearly shows that the hard recombination of mixture models are better for a majority of points. Regarding the smooth recombination, no clear conclusion can be drawn since it performs better than ANN for pure shear loading and it is less accurate for combined loading. To illustrate the algorithm, we depicted Fig. 4.7 the results of the clustering for $K = 4$ clusters (determined by the strategy already presented) and we also compare Fig. 4.10 the real function with the approximation model.

4.7.2 Approximation models for varying loading conditions

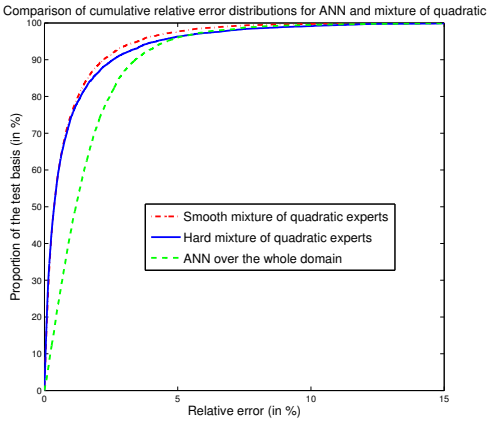
Based on the observations made, λ_{cr} is approximated through its reciprocal, the result of the approximation is then inverted and the accuracy is measured on λ_{cr}. For the first case (no shear), $N = 468$ points were used with $N_x \in [-100, 0]$ and $N_y = [-50, 50]$ and for the second case 940 points were used with $N_x \in [-100, 0]$, $N_y = [-50, 50]$ and $N_{xy} = [0, 50]$. In both cases, the stacking sequence that was fixed was a non-orthotropic one $[-45/-45/0/45/90/-45/0]_S$. Results are in Tab. 4.3. In that case, mixture of quadratic models performs better for all accuracy metrics. Quantiles measures and quantiles curves show the same tendency as in the former test cases. Mixture models are much more accurate for a large number of points of the validation basis.



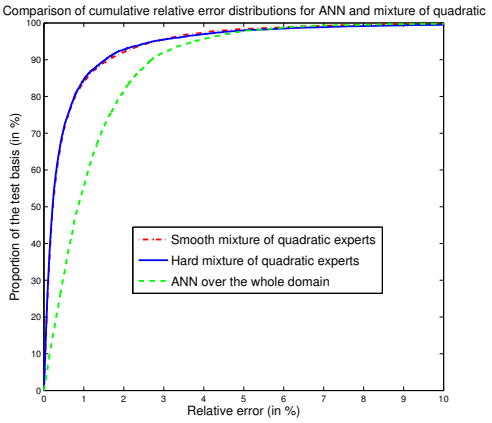
(a) Comparison of approximations of λ_{cr} over N_x and N_y for a non-orthotropic laminate



(b) Comparison of approximations of λ_{cr} over N_x, N_y and N_{xy} for an non-orthotropic laminate



(c) Comparison of approximations of λ_{cr} over ξ_D^1, ξ_D^2, ξ_D^3, N_x and N_y



(d) Comparison of approximations of λ_{cr} over ξ_D^1, ξ_D^2, ξ_D^3, N_x, N_y and N_{xy}

Figure 4.12: Quantile curves for the approximation of λ_{cr} a) over N_x and N_y b) over N_x, N_y and N_{xy} c) over ξ_D^1, ξ_D^2, ξ_D^3, N_x1 and N_y d) ξ_D^1, ξ_D^2, ξ_D^3, N_x1, N_y and N_{xy}

Loading	Combined loading		Pure shear loading	
Type	MOL	ANN	MOQ	ANN
MSE	$8.5.10^{-8}$	$7.8.10^{-7}$	$2.1.10^{-6}$	$1.7.10^{-6}$
LOF(%)	$9.1.10^{-4}$	$8.5.10^{-3}$	$8.5.10^{-4}$	$6.7.10^{-4}$
$\hat{E}_{rel}(\%)$	7.10^{-3}	$1.6.10^{-1}$	$8.8.10^{-3}$	$3.4.10^{-3}$
$\hat{E}_{abs}$	$3.1.10^{-5}$	$6.3.10^{-4}$	$2.4.10^{-4}$	$8.1.10^{-4}$
$E_{rel}^{max}(\%)$	1.16	1.6	0.9	0.5
E_{abs}^{max}	$4.6.10^{-3}$	$5.4.10^{-3}$	$2.5.10^{-2}$	$7.5.10^{-3}$
$Q_{1\%}(\%)$	99.7	99.7	100	99.7
$Q_{0.1\%}(\%)$	98.8	43.6	97.9	92.9

Table 4.1: Results for the approximation of λ_{cr} for orthotropic laminates for combined loading and pure shear loading

4.7.3 Approximation models in the general case

In that case, we make both stacking sequence and loading vary. We distinguished two cases, one with non-orthotropic laminates and no shear loading (biaxial loading including tension/compression). Dimension of regression in that case is 5: ξ_D^1, ξ_D^2, ξ_D^3, N_x and N_y. We used $5,400$ points ranging in Miki's tetrahedron for lamination parameters and $N_x \in [-100, 0]$ and $N_y \in [-50, 50]$. In the second, we added shear loading, dimension of regression was then 6 and we used $8,400$ points within the same domain with $N_{xy} \in [0, 50]$. Results are in Tab. 4.4. For both regressions, results show that ANN's give lower MSE, but for all the other accuracy metrics, the mixture of rational experts give a better accuracy. Regarding quantile errors, we have the same tendency as in the other cases, that is mixture models gives a better accuracy for most points than ANN's.

4.7.4 Conclusion on the accuracy of the approximation

We see that in many cases, mixture models offer quite a **good performance in terms of the accuracy of the approximation**. For most cases, the **hard recombination was found to be better than the smooth recombination and a reasonably good artificial neural network**. In case one type of variables, amongst stacking or loading, is fixed, the accuracy is very high allowing to use the approximation models within an optimization process. When material and loading vary the quality of the

Loading	Combined loading		Pure shear loading	
Type	MOQ	ANN	MOQ	ANN
MSE	$6.9.10^{-4}$	$2.7.10^{-4}$	$1.7.10^{-4}$	$9.3.10^{-4}$
LOF(%)	3.10^{-2}	$1.2.10^{-2}$	$3.4.10^{-4}$	$1.9.10^{-3}$
$\hat{E}_{rel}(\%)$	$7.7.10^{-2}$	$1.4.10^{-1}$	3.10^{-2}	$7.6.10^{-2}$
$\hat{E}_{abs}$	$6.4.10^{-3}$	$1.2.10^{-2}$	7.10^{-3}	$1.8.10^{-2}$
$E_{rel}^{max}(\%)$	5.1	1.3	0.47	0.9
E_{abs}^{max}	$3.8.10^{-1}$	$8.8.10^{-2}$	$8.9.10^{-2}$	0.15
$Q_{1\%}(\%)$	98.2	99.8	100	100
$Q_{0.1\%}(\%)$	89	48.1	93.1	79.9

Table 4.2: Results for the approximation of λ_{cr} for non-orthotropic laminates for combined loading and pure shear loading

approximation degrades, this is of course due to the dimension but also of the growing complexity of λ_{cr} much modes are indeed covered when making material and loading vary. What these results suggest is that approximation models should take advantage of this mode-based structure to improve the accuracy of the approximation. However, doing so requires a fairly good number of learning points within each mode, which is not an easy task to obtain without prior knowledge.

4.8 Applications to optimization: maximization of the buckling critical load factor

4.8.1 Optimization problem

In order to illustrate the interest of the approximation of λ_{cr}, we consider a stacking sequence optimization to maximize buckling critical factor of a composite plate under combined loading. The optimization problem can be stated as follows

$$\begin{cases} \max \lambda_{cr}(\xi_1^D, \xi_2^D, \xi_3^D) \\ \text{s.t.} \quad 2|\xi_D^1| - \xi_D^2 - 1 \leqslant 0 \\ \qquad 2|\xi_D^3| + \xi_D^2 - 1 \leqslant 0 \end{cases}$$

where some parameters are fixed: the material, the dimensions of the plate a and b, the total number of plies n, the loading with N_x, N_y and N_{xy}.

Loading	Biaxial loading		Combined loading	
Type	MOQ	ANN	MOQ	ANN
MSE	$1.93.10^{-6}$	$4.8.10^{-5}$	$3.6.10^{-3}$	$6.6.10^{-3}$
LOF(%)	$2.1.10^{-4}$	5.10^{-3}	$2.6.10^{-1}$	$4.7.10^{-1}$
$\hat{E}_{rel}(\%)$	$5.1.10^{-3}$	$9.3.10^{-2}$	4.10^{-1}	$7.1.10^{-1}$
$\hat{E}_{abs}$	1.10^{-4}	2.10^{-3}	$1.1.10^{-2}$	$1.7.10^{-2}$
$E_{rel}^{max}(\%)$	1.4	2	11.1	29.4
E_{abs}^{max}	$3.9.10^{-2}$	$9.9.10^{-2}$	1.1	1.1
$Q_{1\%}(\%)$	99.9	99.4	90.4	81.8
$Q_{0.1\%}(\%)$	98.8	75.6	65.9	28.9

Table 4.3: Results for the approximation of λ_{cr} for varying loadings (biaxial loading and combined loading)

4.8.2 Two approaches

Two approaches will be compared to solve this problem:

- an explicit approach: it consists in listing all the feasible stacking sequences for a fixed number of plies n and computing the buckling load with a Rayleigh-Ritz method. The optimization problem is written as follows

$$\max_{\text{feasible SS}} \lambda_{cr}(\text{SS})$$

  where the acceptable stacking sequence space is generated by listing the whole of the possible combinations of angles $[0°, \pm 45°, 90°]$ which comply with the manufacturing rules: balanced ($\pm 45°$), symmetric, contiguity (maximum of 4 equal successive layer orientations), disorientation (two consecutive ply orientation should not differ from more than 45°), and finally a proportion rule with a minimum of 8% for each orientation. The buckling critical factor λ_{cr} is computed with a Rayleigh-Ritz (RR) method on the subset of feasible stacking sequences. The optimal stacking sequence $\text{SS}^\star$ is referred as the reference solution of the optimization problem.

- a two-step approach composed of a continuous optimization in the out-of-plane lamination parameters space followed by a post-identification to find the associated optimal stacking sequence.

Loading	Biaxial loading		Combined loading	
Type	MOR	ANN	MOR	ANN
MSE	$7.6.10^{-3}$	$3.4.10^{-3}$	$1.9.10^{-2}$	7.10^{-3}
LOF(%)	$3.8.10^{-1}$	$1.7.10^{-1}$	$3.8.10^{-1}$	$1.3.10^{-1}$
$\hat{E}_{rel}(\%)$	0.9	1.6	0.67	1.25
$\hat{E}_{abs}$	$3.2.10^{-2}$	$3.6.10^{-2}$	$3.3.10^{-2}$	$3.7.10^{-2}$
$E_{rel}^{max}(\%)$	20.9	40.5	31	22.4
E_{abs}^{max}	1.23	0.7	3.64	1.56
$Q_{1\%}(\%)$	73.7	43.6	84.6	55.6
$Q_{0.1\%}(\%)$	19.5	4.1	28.6	6.6

Table 4.4: Results for the approximation of λ_{cr} for varying non-orthotropic laminates and for varying flux

1. solve the continuous optimization problem

$$\begin{cases} \max \lambda_{cr}(\xi_1^D, \xi_2^D, \xi_3^D) \\ \text{s.t.} \quad 2|\xi_D^1| - \xi_D^2 - 1 \leqslant 0 \\ \qquad 2|\xi_D^3| + \xi_D^2 - 1 \leqslant 0 \end{cases}$$

   and determine the optimal out-of-plane lamination parameters $(\xi^D)^\star = (\xi_1^D, \xi_2^D, \xi_3^D)^\star$. The concavity of λ_{cr} ensures that any local maximum is a global maximum and that a gradient-based methods (such as SQP,...) will converge to a global maximum.

2. from the optimal vector $(\xi^D)^\star$ find the associated $SS^\star$ by solving

$$\begin{cases} \max \lambda_{cr}(SS) \\ \text{s.t.} \quad \xi^D(SS_{\text{feasible}}) \in \mathcal{N}((\xi^D)^\star) \end{cases}$$

   where $\mathcal{N}((\xi^D)^\star)$ denotes the neighborhood of the optimal vector $(\xi^D)^\star$. To define this neighborhood, we consider the ball $\mathcal{B}((\xi^D)^\star, r)$ of center $(\xi^D)^\star$ and of radius r chosen such that the cardinal of points inside $\mathcal{B}$ is equal to 20 (fixed number). The reason for choosing the best stacking within a neighborhood is again related on the concavity of λ_{cr}. Indeed, this ensures that the discrete optimum can not be too far away from the continuous optimum. There is no reason however that the discrete optimum is the closest discrete stacking sequence. This local research could definitely be improved by constructing a new distance that will weights the influence of each ξ_D^i (based on the Hessian of λ_{cr} at the optimum for instance).

	explicit	2-step with RR	2-step with MOQ
optimal SS	$[-45/-45/-45/0/45/45/0/0/45/90]_s$		
Error at optimum (%)	exact	exact	0.03%
Number of λ_{cr} evaluations	7032	$269+20$	$39+20$

Table 4.5: Optimization result with the combined loading $N_x = -50N.mm^{-1}$, $N_y = 20N.mm^{-1}$ and $N_{xy} = 20N.mm^{-1}$.

4.8.3 Comparison of the optimization results

We consider the Graphite/Epoxy material given in Tab. 3.2.4, a plate of 650×250 mm with 10×2 plies according to the symmetry. A piecewise polynomial regression of degree 2 (MOQ) is build to approximate the buckling critical factor λ_{cr} and is used in the optimization process to reduce computational time (compared with the RR approach).

4.8.3.1 A combined loading example

First the combined loading $N_x = -50N.mm^{-1}$, $N_y = 20N.mm^{-1}$ and $N_{xy} = 20N.mm^{-1}$ is considered. We report in table 4.5 the results obtained with the two approaches: in all cases we obtained the same optimal stacking sequence, the only difference results in the number of objective function evaluations: 7032 by listing all the feasible subset of stacking sequences with 10×2 plies, 269 function evaluations for the continuous optimization by using the 2-step approach with the RR method to compute λ_{cr} instead of 39 evaluations with the MOQ approximation. For the SS-identification process, 20 more evaluations of λ_{cr} are required to build the neighborhood $\mathcal{B}((\xi^D)^\star, r)$ of the optimal vector and then choose the optimal stacking sequence.

The convergence of the optimization process is compared on Fig. 4.13. Due to the smoothness of the quadratic approximate model (MOQ), the number of iterations is drastically reduced with 9 iterations instead of 21 with the RR method. Fig 4.14 shows the feasible out-of-plane lamination parameters in Miki'tetrahedron. The considered neighborhood of the continuous solution is also represented and contains the feasible optimal solution. As expected, we can see that these optimal parameters are on the boundary of the domain. The use of a piecewise polynomial approximation of the λ_{cr}

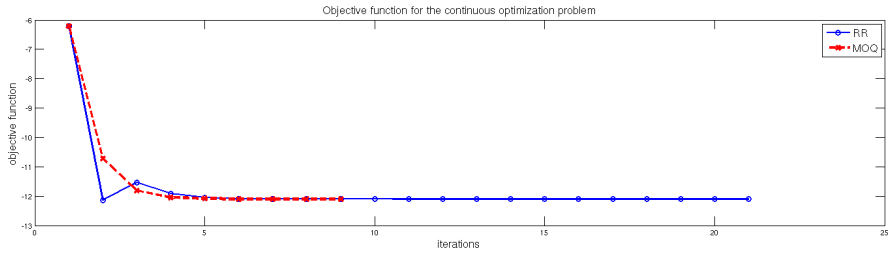


Figure 4.13: Objective function during the optimization process with the combined load

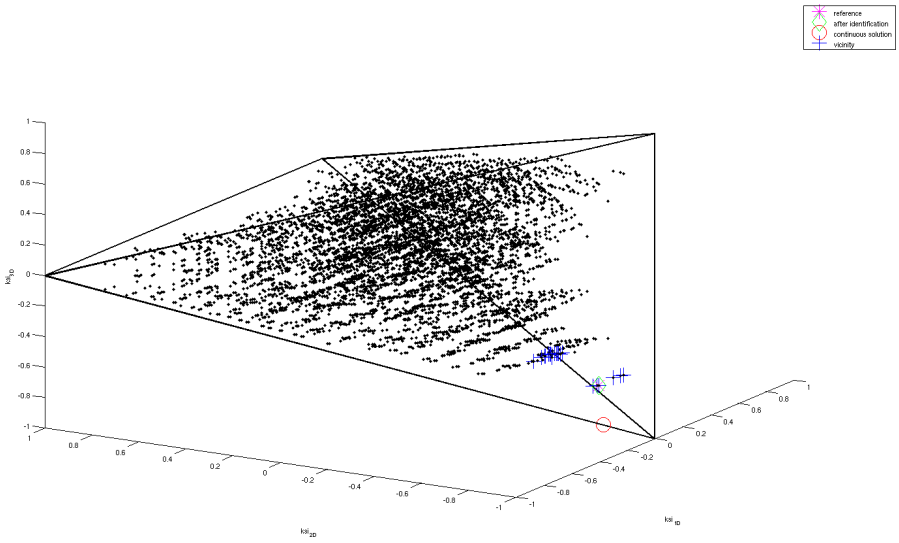


Figure 4.14: Optimal solution in the lamination parameters space (the 7032 feasible ξ^D are illustrated by a black dot market) under the combined loading

	explicit	2-step with RR	2-step with MOQ
optimal SS	$[-45/-45/-45/-45/90/45/45/45/45/0]_s$		
Error at optimum (%)	exact	exact	0.1%
Number of λ_{cr} evaluations	7032	$38+20$	$32+20$

Table 4.6: Optimization result with the shear loading $N_x = N_y = 0N.mm^{-1}$ and $N_{xy} = 20N.mm^{-1}$.

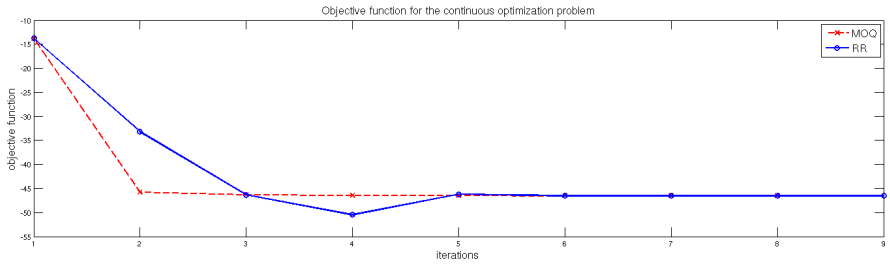


Figure 4.15: Objective function during the optimization process with the shear loading

offers many advantages in term of convergence, computing time and makes it possible in this case to find the reference solution.

4.8.3.2 A pure shear loading example

To confirm these first very promising results, a second loading case is considered with $N_x = Ny = 0\text{N.mm}^{-1}$ and $N_{xy} = 20\text{N.mm}^{-1}$. The results are presented in Table 4.6. As previously, the reference stacking sequence is found by the 2-step approach in both cases: with the RR method or the MOQ method to compute λ_{cr}. The number of iterations is very close and Fig. 4.15 confirms that the objective function converges in a similar way (8 or 9 iterations). One more time, the optimal solution belongs to the boundary of the Miki' tetrahedron as shown in Fig. 4.16.

4.8.4 Conclusion of the optimization applications

The use of our approximate model of λ_{cr} is particularly useful for the optimization strategy developed here. The approach in 2 stages with MOQ makes it possible to find the reference solution with few evaluations of the objective function, and in a negligible computational time. If the number of plies n would be higher, acceptable space would be larger and thus one could clarify all the combinations with difficulty. One will prefer

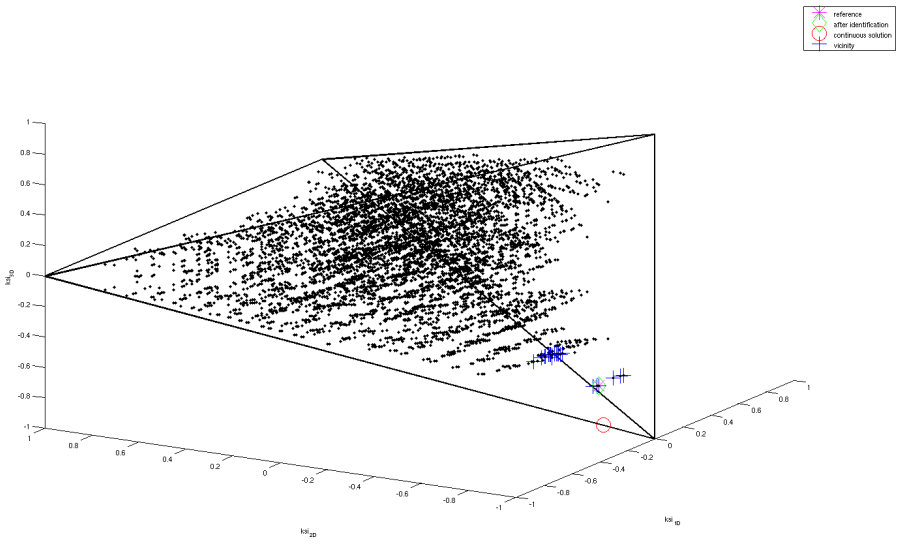


Figure 4.16: Optimal solution in the lamination parameters space (the 7032 feasible ξ^D are illustrated by a black dot market) under the shear loading

in this case to use an evolutionary algorithm (genetic algorithm, ...) coupled with the MOQ approximation for the objective function to perform the stacking sequence identification.

4.9 EM estimation of hyperparameters for memory-based methods

In this section, we show that the hyperparameters of well-known approximation methods in the case of regression : rbf, moving least squares can be thought as mixture density parameters and therefore can be set through maximum likelihood estimation for mixture densities. In particular, the support points for regressive RBF's and kernel centers and kernel smoothing methods are interpreted as means of the same number of individual probability laws. We also show the the well-known regression technique of Weighted Least Squares can be seen as a soft homoscedastic Gaussians mixture of polynomial experts. Finally, we carry out a locally varying weight for Moving Least Squares based on fully-free Gaussian mixture where the weight function is interpreted as the expectation of the variance-covariance matrices.

These methods arise from different areas of supervised learning : for instance radial basis functions are usually derived from classical neural networks theory, more precisely, they are a particular case of multilayer perceptron where the activation functions of the neurons are univariate Gaussian probability density functions. Kernel smoothing methods like local polynomial regression (or moving least squares in the engineering context) arise from statistics theory where they were historically derived from the apparently unsupervised methods for density estimation (like Parzen windowing or nearest neighbors methods). Although they are quite different, they do share a common feature, when used in supervised learning (e.g regression) the prediction for a new entry needs to be computed at each time. Unlike neural networks that provide a quite simple closed-form expression of the model, these methods do not provide easy to handle analytical expression. If we think of interpolating radial basis functions all the data base is needed to predict the outcome of a new entry, indeed the analytical expression they produce uses all the inputs values. Local polynomial regression are even more complicated since they do not provide any closed-form expression, and the model needs to be computed for any new prediction (i.e an optimization problem of small size must be solved for each new point prediction). In all cases, all the data or a subsequent part of the data is used for a new prediction, making this kind of methods useless from the compression point of view.

4.9.1 Radial basis network

The major idea of radial basis functions network is to explain the data by a model of the form

$$f(x) = \sum_{i=1}^{M} \alpha_i \phi_i(||x||) \tag{4.50}$$

where the one-dimensional functions ϕ_i's are called radial functions since they only depend on the norm of the entry. There many choices for the type of radial functions, but we will focus on Gaussian density function since there are by far the most popular and also because they provide a connection with mixture density estimation. Note that each radial function is expected to locally capture the structure of the data and they are defined together with so-called support points $(x_i)_{1...M}$ where the radial functions

are centered at. In our case of Gaussian density functions, these ϕ_i turns out to be

$$\phi_x(||x||) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2\sigma^2}(x-x_i)^T(x-x_i)} \tag{4.51}$$

Note suppose we have set the x_i and σ to some values, then building the radial basis functions network (i.e computing the α_i's) is straightforward through classical least squares estimation

$$min_{\alpha\in\mathbb{R}^M} \sum_{i=1}^{N} (y_i - \sum_{j=1}^{M} \alpha_j e^{-\frac{1}{2\sigma^2}(x-x_i)^T(x-x_i)})^2 \tag{4.52}$$

whose solution can be easily derived analytically.

Note that, when building interpolating radial basis functions network, we take as many basis functions as observations in the sample, the support points x_i are set to be observed inputs. In that case, 4.52 has as many equations as variables and there exists one and only one solution $\alpha^\star$ that makes the residual sum of squares vanish. This procedure however cannot be applied to very large data sets in high dimension and still needs an estimation of σ the standard deviation (or in the kernel smoothing methods jargon: the *width* of the Gaussian kernel). We need therefore good procedures to set σ and the x_i's. We could extend the optimization problem 4.52 with adding the x_i's and the σ as optimization variables but the resulting optimization problem would no longer be convex and the algorithms used to solve this kind of problem are exactly the same as in neural network training (BFGS, Levenberg-Marquardt). Another approach is to estimate the hyperparameters x_i and σ separately from the α_i's. It is where mixture density estimation come to help. Indeed, these hyperparameters can be thought as means and variance of Gaussian mixture of M components. Once a number of M radial basis functions has been chosen, we apply EM algorithm on the joint database for an homoscedastic mixture ($\Gamma_i = \sigma^2 Id$) and get maximum likelihood estimators of the support points x_i's and σ by computing the marginal laws of X exactly as we did in the former section. We do need to restrict to homoscedascic mixture and we could run the EM algorithm for fully free Gaussian mixtures. This would result in different variance-covariance matrices for supports points and would allow the model f to capture more sophisticated behavior of the underlying function. Finally a common

practice with radial basis functions is to renormalized basis functions $\hat{\phi}_i$ of the form

$$\hat{\phi}_i(x) = \frac{\phi_i(x)}{\sum_{j=1}^{M} \phi_j(x)} \tag{4.53}$$

to avoid regions of the input space where all ϕ_i's are low-valued, which creates holes. Note that in the context of mixture of experts, rbf networks with renormalized basis functions are exactly mixture of constant experts for a soft clustering. Indeed, the form of f is nothing else than a mixture of expert where the local expert is a sole scalar value ($f_k(x) = \alpha_k$ with the same notations as in the former section). In this context we see that α_k is somehow related to the expectation of the fuzzy cluster C_k (weighted expectation of Y).

4.9.2 Weighted Least Squares

We give here an interpretation of the weighted least squares regression. Say we have $\mathcal{X} = (x_1, \ldots, x_n) \in \mathbb{R}^d$ together with their output values $\mathcal{Y} = (y_1, \ldots, y_n)$. A local Weighted Least Squares regression at point $\hat{x} \in \mathbb{R}^d$ consists in finding the best multivariate polynomial approximation $f_{\hat{x}}$:

$$f_{\hat{x}} = \text{argmin}_{p \in \prod_s^d} \sum_{i=1}^{n} \theta(||x_i - \hat{x}||)||p(x_i) - y_i||^2 \tag{4.54}$$

where Π_s^d is the set of multivariate polynomials of $d-$variables and degree s and θ is a symmetric positive and decreasing function : maximum value at 0 and 0 when $|x| \to +\infty$: which can be strictly positive or compactly-supported. θ is called a weight function. A very popular choice for θ is

$$\theta(r) = \frac{1}{\sqrt{(2\pi)^d \sigma^{2d}}} e^{-\frac{1}{2}\frac{r^2}{\sigma^2}} \tag{4.55}$$

This regression $f_{\hat{x}}$ is a local approximation which is valid only in the neighborhood of $\hat{x}$. Note that the moving least squares regression consists in a global weighted least squares by making the $\hat{x}$ varying (or moving) over the whole domain $\mathbb{R}^d$, this global model happens to be continuous and even smooth whenever the weight function is smooth (Levin 1998). We depicted several WLS regression on different points together with the MLS regression Fig. 4.17. Another way to get a global model is by using a

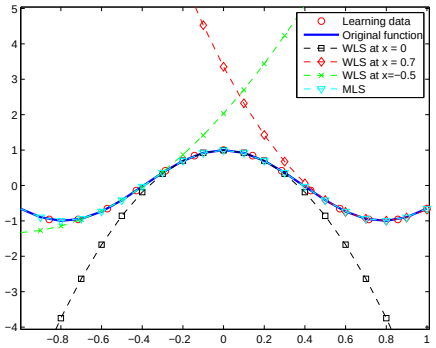


Figure 4.17: Weighted least squares regressions (polynomials of degree 2 and $\sigma = 0.1$) of the function $x \to \cos(4x)$ at different points and moving least squares regression

partition of unity. Namely, we have K so called support points $\hat{x}_1, \ldots, \hat{x}_K$, a local WLS regression $\hat{f}_{\hat{x}_j} = \hat{f}_j$ is built at each support point $\hat{x}_j$ and the global model is

$$F_{WLS}(x) = \sum_{i=1}^{K} \beta_i(x) \hat{f}_i(x) \tag{4.56}$$

where

$$\beta_j(x) = \frac{e^{-\frac{1}{2}\frac{||x-x_j||^2}{\sigma^2}}}{\sum_{i=1}^{K} e^{-\frac{1}{2}\frac{||x-x_i|}{\sigma^2}}} \tag{4.57}$$

This popular model has an obvious interpretation in terms of MoE and soft clustering. Assume that the law[1] of $X \sim \sum_{j=1}^{K} \frac{1}{K} \mathcal{N}_d(\hat{x}_j, \sigma^2 I_d)$. In that case, we have, keeping the same notations as in section 2

$$\mathbb{P}(X = x|j) = \frac{1}{\sqrt{(2\pi)^d \sigma^{2d}}} e^{-\frac{1}{2}\frac{||x-\hat{x}_j||^2}{\sigma^2}} \tag{4.58}$$

and

$$\mathbb{P}(j|X = x) = \frac{e^{-\frac{1}{2}\frac{||x-\hat{x}_j||^2}{\sigma^2}}}{\sum_{i=1}^{K} e^{-\frac{1}{2}\frac{||x-\hat{x}_i||}{\sigma^2}}} \tag{4.59}$$

[1]In the former examples we focused on Gaussian mixture models that were fully free, i.e. all the parameters of the Gaussian mixture models are not constrained and EM algorithm estimates all the parameters. There are more simple Gaussian mixture models that assume that all the means are the same or all of the variance-covariance matrices are of the form $\sigma^2 I_n$ (this hypothesis is known in statistics as homoscedascity)

and we see that

$$F_{WLS}(x) = \sum_{i=1}^{K} \mathbb{P}(\kappa = i | X = x) \hat{f}_i(x) \tag{4.60}$$

and the WLS model is an MoE for $X \sim \sum_{j=1}^{K} \frac{1}{K} \mathcal{N}_d(\hat{x}_j, \sigma^2 I_d)$ using a soft partitioning and where local experts are multivariate polynomial regression experts weighted by the probabilities of the soft partitioning.

4.10 Towards the numerical implementation of decomposition/bilevel schemes

We presented an original strategy to approximate typical buckling constraints that appear in our large-scale optimization problem. We observed that this strategy achieves a reasonable accuracy for idealized buckling computations. In the next chapter, we get to the practical implementation of our bilevel scheme by first presenting a reasonable test case and finding out the real optimum first by monolevel methods. This innovative strategy will be used in the construction of our real aircraft structure test case in Chapter 7.

4. ORIGINAL STRATEGY TO APPROXIMATE DISCONTINUOUS AND DERIVATIVE DISCONTINUOUS FUNCTIONS

5 Monolevel optimization schemes

In this chapter, we describe the different mono-level schemes that can be used to solve Problem (1.8) on the basis of the equivalence of optimization problems described in Chapter 2. Indeed, we want to compare the results of multilevel schemes on the basis of mono-level schemes results. Moreover, the different mono-level schemes that we describe in this section will help us to favor one decomposition of the optimization variables amongst several. Indeed, the classical AIO (All-In-One) uses all local design variables without using directly stiffness terms y, while BAO formulations does use explicitly Y terms. These different mono-level schemes are the basis of the multilevel schemes that we will compare and apply in the next chapter. In particular, we will see that different mono-level schemes imply different multi-level schemes (AIO scheme implies non-hierarchical methods such as the one presented in the next chapter) while BAO-like formulations imply hierarchical methods as Target Rigidity, this was already outlined in Chapter 2 but in this chapter, we make this appear for the real structural optimization problem we want to solve.Before describing the mono-level schemes, we start first with a description of the test case we will use throughout this chapter and in the next chapter and second with a brief description of sensitivity analysis.

5.1 Formal description of the test case

To compare the different decomposition methods we investigated in Chapter 2, we used the same test case as in (Merval 2008): **the 10-bar truss**. This is a classical example in structural optimization. It is a truss made of 10 bars (tension/compression only). Each of these 10 bars is I-profiled. This truss and the dimensions of the I-profile are depicted Fig. . A load case is set on nodes 4 and 5 and we optimize the dimensions of the truss for this load case under the constraints of local buckling of the web of the bar, Euler buckling, tension and compression/tension yield stress limits. We did not impose maximal displacements nor continuity design constraints.

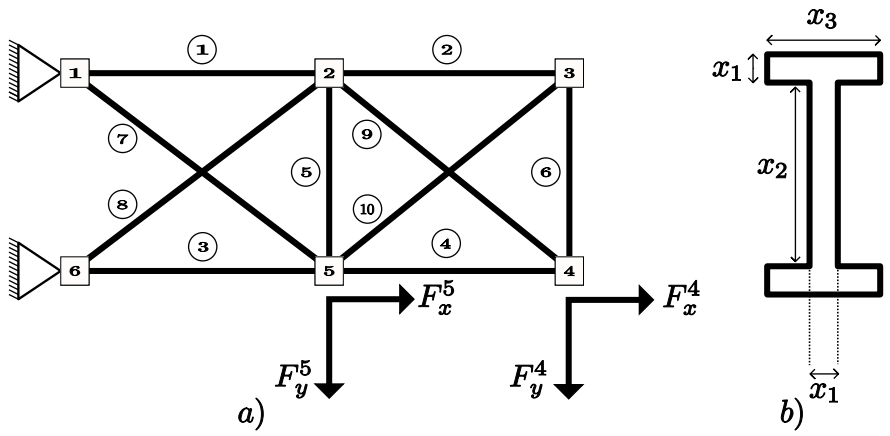


Figure 5.1: a) 10 bar truss b) I-profile

We note that this test case gathers all the difficulties of the problem (**??**) for any modification of the detailed dimensions changes the stress distribution and we have global-local constraints that link the global stress redistribution with local design.

5.1.1 Analytical expression of the objective function

We keep the convention, defined in the first chapter, that the dimensions are sorted in the bars order.

$$X = (x_1^{(1)}, x_2^{(1)}, x_3^{(1)}, \ldots, x_1^{(10)}, x_2^{(10)}, x_3^{(10)}) \tag{5.1}$$

The weight of the straight bars is easily obtained, with L the length of the bar and ρ_1 the volumic density

$$m^{(i)}(x_1^{(i)}, x_2^{(i)}, x_3^{(i)}) = (x_1^{(i)} x_2^{(i)} + 2x_1^{(i)} x_3^{(i)}) L \rho_1 \tag{5.2}$$

For $i = 1 \ldots 6$, the weight of the diagonal bar is :

$$m^{(i)}(x_1^{(i)}, x_2^{(i)}, x_3^{(i)}) = (x_1^{(i)} x_2^{(i)} + 2x_1^{(i)} x_3^{(i)}) \sqrt{2} L \rho_2 \tag{5.3}$$

For bars $i = 7 \ldots 10$. we consider that the diagonal bars do not have the same volumic density to allow us to amplify the hyper-static configuration of the truss.
Global weight is therefore

$$M(X) = \sum_{i=1}^{N} m^{(i)}(x_1^{(i)}, x_2^{(i)}, x_3^{(i)}) \tag{5.4}$$

or

$$M(X) =< X, AX > \tag{5.5}$$

with

$$A = \begin{pmatrix} A_1 & & & & \\ & \ddots & & & \\ & & A_i & & \\ & & & \ddots & \\ & & & & A_N \end{pmatrix} \tag{5.6}$$

where

$$A_i = \begin{cases} \rho_1 L \begin{pmatrix} 0 & 1 & 2 \\ 1 & 0 & 0 \\ 2 & 0 & 0 \end{pmatrix} & \text{for} \quad i = 1 \dots 6 \\ \rho_2 L\sqrt{2} \begin{pmatrix} 0 & 1 & 2 \\ 1 & 0 & 0 \\ 2 & 0 & 0 \end{pmatrix} & \text{for} \quad i = 7 \dots 10 \end{cases} \tag{5.7}$$

Weight is therefore a quadratic function where we sort the nodes such that the matrix M would be of minimal band. This quadratic function is not definite ($\det(A) = \prod_{i=1}^{N} \det(A_i) = 0$), nor positive ($\{-2.13, ..., 2.13...\} \in \mathrm{Sp}(M)$) and therefore the function M is not convex. Nonetheless, the function $\hat{M}$ is convex for variables Y. Practically, it means that most of the usual optimization gradient-based methods are not guaranteed to converge to the global minimum when applied over x variables.

5.1.2 Stress computations and constraints

5.1.2.1 Stress computation

We briefly review the computation of stress inside the bars. Recall that we are in the linear elastic domain (small perturbations) and we look for the approximate solution of the linear elastic equations with the hypothesis that bars are only subject to compression and tension. The computation is then

1. initial detailed configuration of the truss : $X \in \mathbb{R}^{30}$ and load case
2. computation of stiffness terms : $(A, I) \in \mathbb{R}^{10} \times \mathbb{R}^{10}$,with A the cross-section areas of bars and I the quadratic momentum.

$$\begin{aligned} A_i = &\quad x_1^{(i)} x_2^{(i)} + 2 x_1^{(i)} x_3^{(i)} \\ I_i = &\quad x_1^{(i)} \Big(\frac{\left(x_2^{(i)}\right)^3}{12} + \frac{\left(x_2^{(i)}\right)^2 x_3^{(i)}}{2} \Big) \end{aligned} \tag{5.8}$$

3. we loop on the elements (bars) and we assemble the stiffness matrix with the stiffness terms $K_A \in \mathcal{M}_{10}(\mathbb{R})$. We depicted Fig. 5.2, the pattern of this matrix. We know that the nonzero terms of this matrix are of the form $\frac{EA_i}{L_i}$ (with E Young modulus of the material and L_i length of the bar i) such that the partial derivative matrices $\frac{\partial K_A}{\partial A_i}$ used to compute analytical sensitivity are constant and easy to compute.

4. we assemble the external strength vector f associated with the load case specified

5. we solve

$$K_A d = f \tag{5.9}$$

to compute the displacements d.

6. we get from it the strain disitrbution ε_i and the stress distribution σ_i inside each bar i from Hooke's law ($\sigma_i = E\varepsilon_i$).

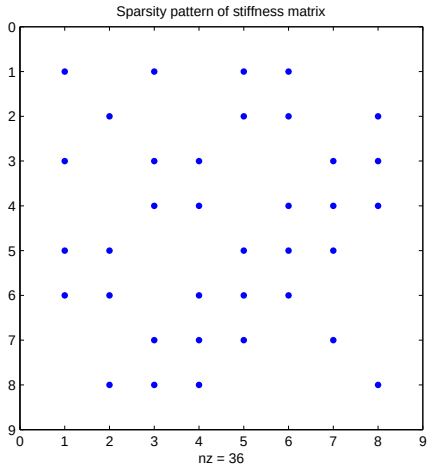


Figure 5.2: Pattern of the stiffness matrix K_A

5.1.2.2 Optimization constraints

The computations made before will allow us to express the optimization constraints. As said before, we do not consider for the moment design continuity constraints and

maximal displacement constraints. Namely, we do not have c^K_{glob} and c_{glob} in our optimization problem.

On the one hand, we have the local design constraints :

$$\begin{array}{l} 1 \leq x_1^{(i)} \leq 5 \\ 10 \leq x_2^{(i)} \leq 100 \\ 0.1 \leq x_3^{(i)} \leq 80 \end{array} \tag{5.10}$$

for $i = 1 \dots 10$.

On the other hand we have in the case of tension, the tensile yield stress, and in the case of compression, we have the local buckling constraint, the Euler buckling constraint and the yield stress

- tensile yield stress , for $\sigma_i \geq 0$

$$1 - \frac{\sigma_{tens}}{\sigma_i} \leq 0 \tag{5.11}$$

  where σ_{tens} material admissible

- compressive yield stress, for $\sigma_i \leq 0$

$$1 - \frac{\sigma_{comp}}{\sigma_i} \leq 0 \tag{5.12}$$

  where σ_{comp} compression allowable

- local buckling of the web :

$$1 - \frac{\sigma_{loc,i}}{\sigma_i} \leq 0 \tag{5.13}$$

  where $\sigma_{loc,i} = \frac{4\pi^2 E}{12(1-\nu^2)} \left(\frac{x_1^{(i)}}{x_2^{(i)}} \right)^2$

- Euler buckling :

$$1 - \frac{\sigma_{Eul,i}}{\sigma_i} \leq 0 \tag{5.14}$$

  where $\sigma_{Eul,i} = \frac{\pi^2 E I_i}{L_i^2}$

As we can see, any of these 4 constraints depend on σ_i and then the stress distribution and then depend on all the dimensions. To make the notation a little bit clearer, we did not make the dependence on X_i appears in the constraints but in reality we must write

$\sigma_i(x_1^{(1)}, x_2^{(1)}, x_3^{(1)}, \ldots, x_1^{(10)}, x_2^{(10)}, x_3^{(10)})$, $\sigma_{Eul,i}(x_1^{(i)}, x_2^{(i)}, x_3^{(i)})$ and $\sigma_{loc,i}(x_1^{(i)}, x_2^{(i)}, x_3^{(i)})$. Eventually, we have 10 constraints for each bar and then 100 constraints for the problem which is written formally

$$\begin{array}{ll} \min_{X\in\mathbb{R}^{30}} & M(X) =< X, AX >= \sum_{i=1}^{10} < X_i, A_i X_i > \\ \text{s. t.} & \left\{ \begin{array}{l} 1 \leq x_1^{(i)} \leq 5 \quad \text{for} \quad i = 1 \ldots 10 \\ 10 \leq x_2^{(i)} \leq 100 \quad \text{for} \quad i = 1 \ldots 10 \\ 0.1 \leq x_3^{(i)} \leq 80 \quad \text{for} \quad i = 1 \ldots 10 \\ 1 - \frac{\sigma_{tens}}{\sigma_i(x_1^{(1)},x_2^{(1)},x_3^{(1)},\ldots,x_1^{(10)},x_2^{(10)},x_3^{(10)})} \leq 0 \quad \text{for} \quad i = 1 \ldots 10 \\ 1 - \frac{\sigma_{comp}}{\sigma_i(x_1^{(1)},x_2^{(1)},x_3^{(1)},\ldots,x_1^{(10)},x_2^{(10)},x_3^{(10)})} \leq 0 \quad \text{for} \quad i = 1 \ldots 10 \\ 1 - \frac{\sigma_{loc,i}(x_1^{(i)},x_2^{(i)},x_3^{(i)})}{\sigma_i(x_1^{(1)},x_2^{(1)},x_3^{(1)},\ldots,x_1^{(10)},x_2^{(10)},x_3^{(10)})} \leq 0 \quad \text{for} \quad i = 1 \ldots 10 \\ 1 - \frac{\sigma_{Eul,i}(x_1^{(i)},x_2^{(i)},x_3^{(i)})}{\sigma_i(x_1^{(1)},x_2^{(1)},x_3^{(1)},\ldots,x_1^{(10)},x_2^{(10)},x_3^{(10)})} \leq 0 \quad \text{for} \quad i = 1 \ldots 10 \end{array} \right. \end{array} \tag{5.15}$$

Problem (5.15) is therefore a non-convex quadratic under nonlinear constraints. These constraints turn out not to be convex, as we will see in the following section.

5.1.3 Existence of a minimum and feasible set

We investigate a little deeper the optimization constraints together with the existence of a minimum of this problem.

5.1.3.1 Existence of a minimum

We denote $U^{ad} \in \mathbb{R}^{30}$ the set of all feasible configurations for (5.15), we denote $c^K : \mathbb{R}^{30} \longrightarrow \mathbb{R}^{40}$ optimization constraints that depend on σ_i. By continuity of σ_i, $\sigma_{loc,i}$ and $\sigma_{Eul,i}$ (note that $\sigma_{loc,i}$ is continuous over the feasible domain since $x_2^{(i)} \geq 10$), c^K is continuous and even smooth. The set of feasible configurations with respect to the *complicated* constraints (the constraints that depend on σ_i) is then the reciprocal of $(]-\infty, 0])^{40}$, c^K being continuous this set is then closed. On the other hand U^{ad} is bounded with respect to the first three *box* constraints. Eventually, U^{ad} is a compact set being a closed bounded subset of $\mathbb{R}^{30}$. M as a bilinear function, is continuous and the existence of at least one minimum of the global weight is therefore ensured. Note that M being non-definite and non-positive does not ensure the unicity of such a minimum. This is why, we do not denote any minimum X^* for it may not be unique (and numerical experiments tend to prove that there are several minima) but it makes sense to denote M^* the minimal value of the objective function at this or these minima.

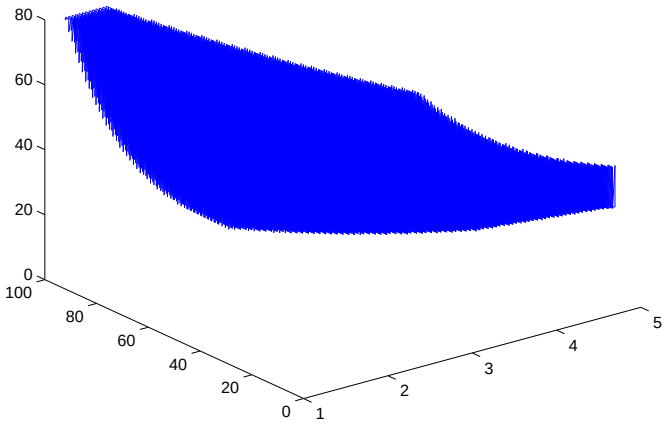


Figure 5.3: Section S_1 of the feasible set for bar 1

5.1.3.2 Feasible set

A question that naturally arises is the connexity of the feasible set since there is no reason for U^{ad} to be connected (the continuous reciprocal of a connected set need not be connected). In the case U^{ad} is connected, we can equally ask if it is also convex. For the moment, we can only look at numerical experiments. We draw a section of U^{ad} (we fix the dimension of bars 2 to 10) and explore the section of U^{ad} for bar 1. We depicted this section in Fig. 5.3 and Fig. 5.4.

The feasible set seems to be connected but not convex as we can see in Fig. 5.4.

We also investigated the pattern of the Jacobian matrix of constraints. Recall that ∇c^K that lies in $\mathcal{M}_{30,40}(\mathbb{R})$ contains the partial derivatives of each constraint with respect to all the design variables x_i. Note that this derivatives (also known as sensitivities) can be calculated explicitly (see next section). To show the structure of this Jacobian (that can be useful to implement new algorithms), we depicted the relevant terms of this matrix, namely the terms whose absolute value is more 10^{-3}, in Fig. 5.1.3.2. As we can see ∇c^K has an almost block diagonal pattern that can be explained by the fact the local dimensions of bar i have a bigger influence of the constraints that concerns bar i than the other, which is perfectly normal from a mechanical point of

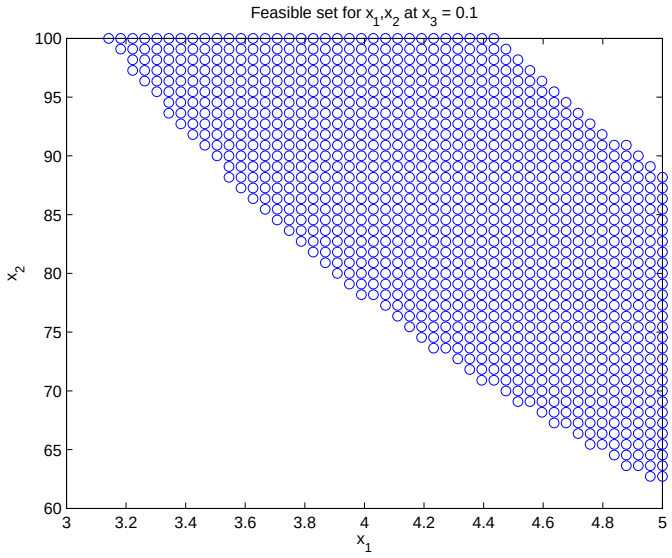


Figure 5.4: Section of S_1 for $x_3^{(1)} = 0.1$

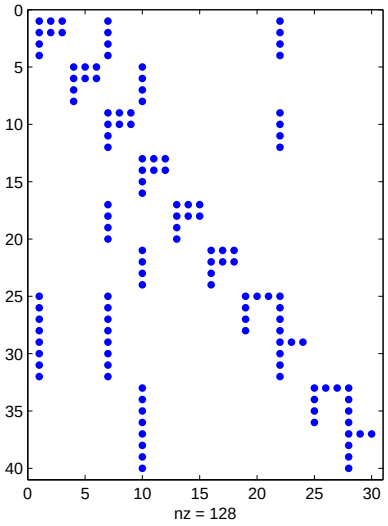


Figure 5.5: Pattern of the Jacobian matrix of the constraints. The x-axis represent the design variables sorted element by element and y-axis represent the constraints sorted element by element as well

view. But we see also that

- Dimensions $x_1^{(1)}$ and $x_1^{(3)}$ (namely the thickness of the flange of bar 1 and 3) have quite a big influence on the constraints of bar 7 and 8.
- Dimension $x_1^{(4)}$ has quite a big influence on the constraints of bars 9 and 10
- Dimension $x_1^{(7)}$ has quite a big influence on the constraints of bars 1 and 3.

5.2 Sensitivity analysis

In structural optimization problems, one often needs to compute derivatives of the constraints and the objective functions. Usually, the gradient of the objective function is quite easy since it is the weight of the structure or the area of a profile, namely it is a simple product between dimensions. The constraints are quite difficult to differentiate since it involves a finite-element analysis and therefore the resolution of a large system of equations. In the AIO case, it becomes quite cumbersome since the FEA is based on the stiffness variables while we want to differentiate the constraints with respect to the design variables.

5.2.1 AIO formulation

We saw that Problem (1.8) involves constraints ($c^{(i)}_{glob-loc}(Y^{(1)}, \dots, Y^{(N)}, X^{(i)})$) that are functions of both design variables X and the internal loads that depend on stiffness variables Y, which only depend on the design variables X. Eventually, such constraints boil down to only depend on design variables X. This leads to the so-called *nested formulation*) of the constraints, that is a formulation that only depends on the design variables and make not appear any stiffness variables, whilst they obviously appear in the computation of the constraints. We start by describing the simple case where we differentiate such a constraint $c(x, u(x))$ where x are the stiffness variables and u the displacements. The nested formulation is therefore $\hat{c}$

$$\hat{c}(x) = c(x, u(x)) \tag{5.16}$$

where

$$K(x)u(x) = f(x) \tag{5.17}$$

and we want the derivative of $\hat{c}$ with respect to x. This derivative is often called the *total derivative* of c. Applying the chain rule to (5.16) with respect to the $i-$th variable x_i we get

$$\frac{d\hat{c}}{dx_i}(x) = \frac{\partial c}{\partial x_i}(x, u(x)) + \frac{\partial c}{\partial u}(x, u(x))\frac{\partial u}{\partial x_i}(x) \tag{5.18}$$

where $\frac{\partial .}{\partial u}$ denotes the derivative with respect to the second variable. Note that usually the first two terms on the right side can be computed easily. We describe in the following section how to compute the last term $\frac{\partial u}{\partial x_i}(x)$.

5.2.2 Direct method and numerical aspects

To compute $\frac{\partial u}{\partial x_i}(x)$, we start with the differentiation of (5.17) with respect to x_i

$$\frac{\partial K}{\partial x_i}(x)u(x) + K(x)\frac{\partial u}{\partial x_i}(x) = \frac{\partial f}{\partial x_i}(x) \tag{5.19}$$

we assume that the load case does not depend on x and we obtain the following linear system to solve

$$K(x)\frac{\partial u}{\partial x_i} = -\frac{\partial K}{\partial x_i}(x)u(x) \tag{5.20}$$

to obtain $\frac{\partial u}{\partial x_i}$. This is called the *direct method* to compute sensitivities of the displacement u with respect to x_i. There exist other methods

- **Finite differences**: it consists in forming an approximation of the derivative, e.g forward or centered finite differences. Despite it is easy to implement, it might be inaccurate (especially when the stepsize is fixed or not controlled) and may require a huge number of finite element analyses, in case fast reanalysis techniques are to be used it can however lead to efficient performance.

- **Adjoint method**: it is also an analytical method as the direct method. It usually much cheaper than the direct method in case when there are few constraints. More precisely, when there are more constraints than design variables (which is our case and usual in structural optimization) the direct method is cheaper than the adjoint. this is the reason why we will not use it

- **Semi-analytical methods**: it consists in computing some intermediate derivatives by finite differences and the others by an analytical way. Typical intermediate derivative are the stiffness matrices derivatives. Suppose we have to differentiate a stiffness matrix K w.r.t A where the non zero terms in K are $\frac{EA}{L}$, finite-differences in that case are for instance exact. This is done for instance in `MSC.Nastran.SOL200`.

We end this section with a very important remark. When we want to compute the sensitivities of u for all design variables $(x_1, \ldots, x_N)$, we have to solve N linear systems of the form (5.20). This would lead to $N \times N^3 = N^4$ flops since solving a linear system is in general of order $\mathcal{O}(N^3)$. But we can make use of the fact it is the same matrix $K(x)$ for all systems and this matrix is symmetric positive definite. A good idea is therefore to first make the Choleski decomposition of $K(x)$, namely to compute L lower triangular matrix with positive diagonal terms such that $LL^T = K(x)$, this requires N^3 flops and solve system (5.20) by

1. Solve by backward substitution $LY = -\frac{\partial K}{\partial x_i}(x)u(x)$: N^2 operations
2. Solve by forward substitution $L^T X = Y$: another N^2 operations

This way, computation of all sensitivities boil down to a complexity of order $\mathcal{O}(N^3)$, this is precisely the same complexity as a FEA. Computations of all sensitivities boil down to only one finite element analysis.

5.2.3 Sensitivity analysis for the AIO formulation

We investigate the computation of analytical sensitivitie with respect to stiffness variables. In what follows, we extend this computation to the AIO case. More precisely, we want to differentiate the constraints with respect to x, while the FEA relies on stiffness variables Y.

Recall the following notations

- N the number of elements indexed by $i = 1 \ldots N$
- n_i is the number of design variables for element i
- $n = \sum_{i=1}^{N} n_i$ is the *total* dimension of the optimization problem

- $X = (x_1^{(1)}, \dots, x_{n_1}^{(2)}, \dots, x_1^{(N)}, \dots, x_{n_N}^{(N)})$ the design variables
- $Y = (Y_1, \dots, Y_N)$ the stiffness variables. Note that we assume there is only one stiffness variable per element, while in all generality there may be several.
- $\varphi : \mathbb{R}^n \mapsto \mathbb{R}^N$ the mapping from the design variables to the stiffness variables (area of profiles). Note that φ is pretty simple and the Jacobian of φ, $D\varphi(X) \in \mathcal{M}_{N,n}(\mathbb{R})$ is easy to obtain and is block diagonal

$$D\varphi(X) = \begin{pmatrix} \nabla\varphi_1(x_1^{(1)}, \dots, x_{n_1}^{(1)}) & & & & \\ & \ddots & & & \\ & & \nabla\varphi_i(x_1^{(i)}, \dots, x_{n_i}^{(i)}) & & \\ & & & \ddots & \\ & & & & \nabla\varphi_i(x_1^{(N)}, \dots, x_{n_N}^{(N)}) \end{pmatrix} \tag{5.21}$$

- $\Phi : \mathbb{R}^N \mapsto \mathbb{R}^N$ the mapping from the stiffness variables to the internal loads. Note that Φ is the FEA and what we saw in the preceding section allows us to from the Jacobian matrix of Φ at point Y $D\Phi(Y) \in \mathcal{M}_N(\mathbb{R})$

$$D\Phi(Y) = \begin{pmatrix} \vdots & & \vdots \\ \nabla\Phi_1(Y) & \dots & \nabla\Phi_N(Y) \\ \vdots & & \vdots \end{pmatrix} \tag{5.22}$$

Recall now that our global-local coupling constraints are written $c^K(X, \Phi(Y))$ and then can be written as well $c^K(X, (\Phi \circ \varphi)(X))$, this leads to the *nested multilevel formulation* of constraints

$$\hat{c}(X) = c^K_{glob-loc}(X, (\Phi \circ \varphi)(X)) \tag{5.23}$$

For the moment we consider a scalar constraint. Applying the chain rule twice we obtain the analytical expression of constraints derivative with respect to $x_j^{(i)}$

$$\frac{\partial \hat{c}}{\partial x_j^{(i)}}(X) = \frac{\partial c^K}{\partial x_j^{(i)}}(X, \Phi(\varphi(X))) + \frac{\partial c^K}{\partial y_i}(X, \Phi(\varphi(X))) \circ \left(\nabla\Phi_i(\varphi(X)) \circ \frac{\partial \varphi_i}{\partial x_j^{(i)}}(X) \right) \tag{5.24}$$

Where $\frac{\partial .}{\partial y_i}$ denotes the derivative with respect to the $i-$th second variable (namely strength F_i) and therefore is a scalar. Now consider vector constraints sorted in the order of the elements i

- $c : \mathbb{R}^n \times \mathbb{R}^N \mapsto \mathbb{R}^m$ with
- $m = \sum_{i=1}^{N} m_i$ where m_i is the number of coupling constraints of element i, e.g for the ten bar truss $m_i = 4$ and $m = 40$, $\hat{c}$ is therefore written as

$$c(X,Y) = \begin{pmatrix} c_1^{(1)}(x_1^{(1)}, \dots, x_{n_1}^{(1)}, Y) \\ \vdots \\ c_{m_1}^{(1)}(x_1^{(1)}, \dots, x_{n_1}^{(1)}, Y) \\ \vdots \\ c_1^{(N)}(x_1^{(N)}, \dots, x_{n_1}^{(N)}, Y) \\ \vdots \\ c_{m_N}^{(N)}(x_1^{(N)}, \dots, x_{n_1}^{(N)}, Y) \end{pmatrix} \tag{5.25}$$

- Denote by $D_X c(X,Y) \in \mathcal{M}_{m,n}(\mathbb{R})$ the Jacobian of the constraint c with respect to the the design variables X (in other words the partial Jacobian matrix of c with respect to the first *vector* variable X. $D_X c$ is obviously block diagonal since $c^{(i)}$ the vector constraint for element i does not depend on design variables of elements $j \neq i$

$$D_X c(X,Y) = \begin{pmatrix} Dc^{(1)}(X^{(1)},Y) & & & & \\ & \ddots & & & \\ & & Dc^{(i)}(X^{(i)},Y) & & \\ & & & \ddots & \\ & & & & Dc^{(N)}(X^{(N)},Y) \end{pmatrix} \tag{5.26}$$

  where $Dc^{(i)}(X^{(i)},Y) \in \mathcal{M}_{m_i,n_i}(\mathbb{R})$. Note that $D_X c$ can be computed by simple differentiation, it consists in differentiating the local constraints : local buckling,... For a large number of design variables this can become cumbersome but any symbolic derivator will do it.

- Denote also by $D_Y c(X,Y) \in \mathcal{M}_{m,N}(\mathbb{R})$ the Jacobian matrix of c with respect to the second *vector* variable, that is the vector of internal strength (or displacements, constraints...). In the usual case, only internal strength F_i in element i appears in the constraints $c^{(i)}$, (see for instance the constraints of the ten bar truss). This means that the derivatives of $c^{(i)}$ with respect to the internal forces F_j for $j \neq i$ is 0. This gives us a particularly sparse and easy to compute matrix

$D_Y c$

$$D_Y c(X,Y) = \begin{pmatrix} \nabla_{y_1} c^{(1)} & & & & \\ & \ddots & & & \\ & & \nabla_{y_i} c^{(i)} & & \\ & & & \ddots & \\ & & & & \nabla_{y_N} c^{(N)} \end{pmatrix} \tag{5.27}$$

where $\nabla_{y_N} \in \mathbb{R}^{m_i}$. For instance, for the ten bar truss, the nonzero elements of this matrix are only 1 and -1.

Finally, summing up ((5.24)) for $i = 1 \dots N$ and $j = 1 \dots n_i$ we end up with the following formula where $D\hat{c}(X)$ is the *total multilevel Jacobian matrix of the constraints* with respect to the design variables

$$D\hat{x}(X) = D_X c(X, \Phi(\varphi(X))) + D_Y c(X, \Phi(\varphi(X)))D\Phi(\Phi(\varphi(X)))D\varphi(X) \tag{5.28}$$

5.3 Mono-level methods

5.3.1 AIO Method

By AIO optimization (All-In-One), we mean the resolution of (6.12) at once. Namely we do not take into consideration the structure of the problem we do not look for a decomposition onto sub-problems. We solve (6.12) as a whole problem. This is of course possible for the ten bar truss problem given its relative small size (30 variables) but practically impossible for real large structures. The AIO formulation is exactly the one in (6.12).

5.3.2 Formal description of AIO

In AIO optimization, the optimization variables are only the local dimensions. Though the stiffness variables are implicitly considered in the computation of constraints. They are nonetheless really computed for the FEA relies on them. The formal AIO problem is therefore

$$\text{(AIO)} \qquad \begin{array}{ll} \min_{X \in \mathbb{R}^n} & M(X) = \sum_{i=1}^N m^{(i)}(X^{(i)}) \\ \text{sous} & \begin{cases} c^K_{glob}(\varphi(X)) \leq 0 \\ c^{(i)}_{glob-loc}(X^{(i)}, \Phi(\varphi(X))) \leq 0 \\ c_{glob}(X^{(1)}, \dots, X^{(N)}) \leq 0 \\ c^{(i)}_{loc}(X^{(i)}) \leq 0 \end{cases} \end{array} \tag{5.29}$$

Note that AIO is only the *nested* formulation of the original problem. We do not make the stiffness variables directly appear in the optimization problem.

5.3.3 Practical implementation

To solve by AIO, we used the `fmincon` function of Matlab. This is based on SQP algorithm. It forms a quadratic approximation of the objective function. Note that our objective function is already quadratic but not convex, while SQP algorithm uses a convex quadratic approximation based the Broyden-Fletcher-Goldsharb-Shanno (BFGS) formula that, roughly speaking, approximates the Hessian by a definite positive matrix. This approximation is usually not controlled by trust region methods (see (**?**) for further details). It also uses a linear approximation of constraints. Practically, we then computed analytically the gradient of the objective function and the Jacobian of the constraints for `fmincon` works better with analytical gradients instead of computing them by finite differences. Moreover, we scaled the detailed dimensions to make them lie in $[0,1]^{30}$ (this changes of course the analytical gradients). This is usual in optimization and improves accuracy and convergence speed.

5.4 SAND original formulation

The original SAND formulation (as described in [3] for instance) comes from MDO problems. The main idea is to convert a multi disciplinary problem into a mono-level problem, the analysis and the design are then performed at the same time (*Simultaneous ANalysis and Design*. In structural optimization, SAND formulation treat mechanical characteristics (displacement, strength...) as optimization variables *a priori* independent from the local variables and force them to be dependent through the equilibrium equation that appears as an optimization constraint that way the analysis is not made for the inequality constraints but only for N equality constraints (namely the equilibrium equation). That way, the Jacobian of the inequality constraints are much easier to compute since we differentiate with respect to X_i in one hand and on the other hand with respect to the strength F_i. More formally, the original SAND formulation for our problem is

$$\text{(SAND)} \qquad \begin{array}{ll} \min_{X,F\in\mathbb{R}^n\times\mathbb{R}^N} & M(X,F)=\sum_{i=1}^{N} m^{(i)}(X^{(i)}) \\ \text{sous} & \left\{\begin{array}{l} F=\Phi(\varphi(X)) \\ c^{K}_{glob}(F)\leq 0 \\ c^{(i)}_{glob-loc}(X^{(i)},F)\leq 0 \\ c_{glob}(X^{(1)},\ldots,X^{(N)})\leq 0 \\ c^{(i)}_{loc}(X^{(i)})\leq 0 \end{array}\right. \end{array} \tag{5.30}$$

On the one hand, we get more optimization variables ($n+N$ instead of n for AIO) and more constraints (m inequality constraints and N equality constraints while AIO only has m inequality constraints) but on the other hand, the constraints are much more simple in the sense we consider that the X_i's and the F_i are independent, the only coupling constraints are the N equality constraints (equilibrium equation). Unfortunately the first results for our test case show that convergence is much slower that in the AIO, though it does converge to the real optimum. The low rate of convergence can be explained that the N equality constraints are very hard to satisfy, this suggests that the analysis has to remain within the inequality constraints, nonetheless the idea of artificially separate variables that are strongly dependent among them is interesting, indeed it gives the problem a block angular form (see (Haftka & Gurdal 1992)). To explain this, we must investigate deeper the structure of the Jacobian of the constraints. Let us use the 10-bar truss. Suppose we sort the constraints in the order of the elements : the first four constraints are local constraints of bar 1, the next four constraints are the local constraints of bar 2 and so on. If we are to compute the Jacobian of the constraints (sorted in this order) we end up with the Jacobian matrix depicted Fig. where it has a dominating block diagonal form but it is though a full matrix (we made only appear elements greater than 10^{-3} but in reality all the elements are nonzero). One could ask if there would not be a way by reorganizing or changing the groups of variables to make this matrix have a nicer profile : block diagonal, block companion... This is achieved by SAND-like methods, which are the structural optimization versions of the BAO formulation of the second chapter. Indeed, suppose we organize our optimization variables this way : $(F_1,...,F_N,x_1^{(1)},...,x_{n_1}^{(1)},...,x_N^{(N)},...,x_{n_N}^{(N)}$ and keep the same order of the constraints. Now the local constraints of element i depend of course on all the F_i's (global sensitivities) and of the X_i's but no more on the X_i's for $i\neq j$. This is the key idea of SAND, that way the Jacobian of the constraints (and the problem) has a so called *block angular form* depicted Fig. 5.6.

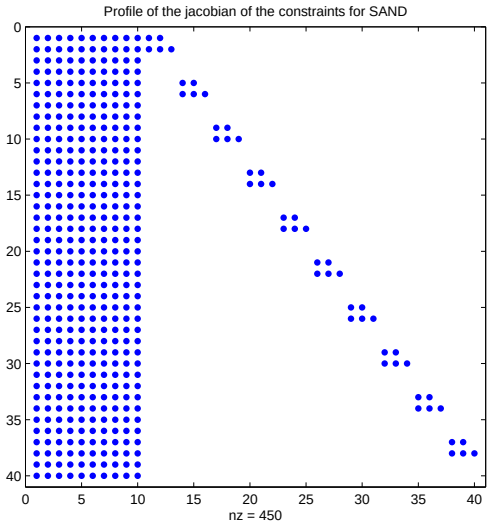


Figure 5.6: Profile of the Jacobian of the constraints for SAND and SAND2

Though the problem is bigger but has a nicer form. This is the reason why we derive the original following formulation, which exploits this artificial separation of variables and its block angular form but keep the FEA within the inequality constraints.

5.5 Block Angular form (SAND2)

Instead of separating forces and local dimensions, one could artificially separate stiffness variables and local dimensions and force them to dependent with equality constraints ($Y = \varphi(X)$), this way we have the following problem

$$\text{(SAND2)} \qquad \begin{array}{ll} \min_{X,Y \in \mathbb{R}^n \times \mathbb{R}^N} & M(X,Y) = \sum_{i=1}^{N} m^{(i)}(Y^{(i)}) \\ \text{s.t} & \begin{cases} Y = \varphi(X) \\ c^{K}_{glob}(Y) \leq 0 \\ c^{(i)}_{glob-loc}(X^{(i)}, \Phi(Y)) \leq 0 \\ c_{glob}(X^{(1)}, \ldots, X^{(N)}) \leq 0 \\ c^{(i)}_{loc}(X^{(i)}) \leq 0 \end{cases} \end{array} \tag{5.31}$$

which is precisely the BAO scheme described in the second chapter. However because of similarity with the SAND formulation, we keep the SAND-like names. Note that the objective function is now convex being linear.

5.6 Block Angular form new version (SAND3)

We can improve the former formulation (SAND2)

- with **skipping the equality constraints** $\varphi(X_i) = Y_i$ and
- with **eliminating one design variable**, e.g for ten bar truss $x_1^{(i)} = \frac{Y_i}{x_2^{(i)} + 2x_3^{(i)}}$
- Formally, we only consider a set of **reduced design variables** $\hat{X}^{(i)}$, the others can be deduced from them and the stiffness variables

It turns out that

- We **reduce the number of optimization variables**, it boils down to n
- We **skip the equality constraints** and therefore ensure compatibility between design and stiffness variables
- We just add one (linear and easy) constraint on the vanished design variables

This leads to the following formulation (SAND3):

$$\text{(SAND3)} \qquad \begin{array}{ll} \min_{\hat{X}, Y \in \mathbb{R}^{n-N} \times \mathbb{R}^N} & M(\hat{X}, Y) = \sum_{i=1}^{N} m^{(i)}(Y^{(i)}) \\ \text{s.t} & \begin{cases} c(\Phi(Y)) \leq 0 \\ c_{gl}^{(i)}(\hat{X}^{(i)}, \Phi(Y)) \leq 0 \\ c_{loc}^{(i)}(\hat{X}^{(i)}, Y_i) \leq 0 \end{cases} \end{array} \tag{5.32}$$

In definitive, 5.32 has the following characteristics

- **It has the same form and the same number of constraints and variables as AIO**
- **Sensitivities are still easy to compute and M is still convex**
- **However, in our example the relationship was pretty easy to derive since $Y = \phi(X)$ is a very simple formula. In case this equation is not as simple (just think of the quadratic momentum for I-shaped beam), an approximation method is to be used to inverse this formula (Newton or Quasi-Newton methods,...) making this transform not so easy and cheap**

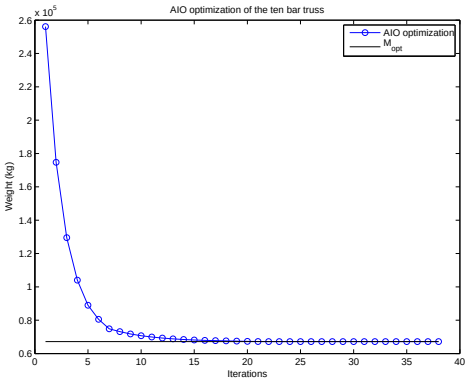


Figure 5.7: Usual behavior of AIO

5.7 Overall comparison and finding a good initial point

On the one hand, we tested the three methods (AIO, SAND2, SAND3) detailed in the preceding section on a bases of 80 load cases. More precisely, we make $F_4^X \times F_4^Y$ vary on $[-10000, -7000, -5000, 0, 5000] \times [] - 15000, -10000, -7000, -5000$, which gives 20 load cases and F_5^X, F_4^X in the same range, which gives 20 other cases. This is done for the case $\rho_1 = \rho_2$ and the case $\rho_2 = 100\rho_1$ since in this case the non-FSD optimality id reinforced and StiffOpt does converge to a very sub-optimal design. To test all the methods on the same conditions, we applied the same convergence criteria to all of them. convergence is obtained when difference between two iterations is below $TolM$ for the objective function and $TolX$, namely

- $||X^{k+1} - X^k|| < \text{TolX} = 10^{-3}$
- $|M(X^{k+1}) - M(X^k)| < \text{TolM} = 10^{-5}$

5.7.1 AIO

AIO convergence under these criteria is usually ensured after 30 iterations. The typical behavior of AIO is depicted Fig. 7

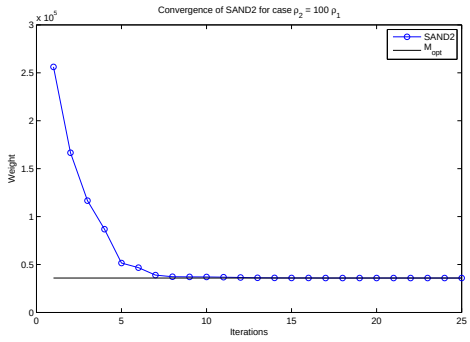


Figure 5.8: Average case for SAND2

5.7.2 SAND2

This new global method does much better than AIO both in terms of iterations : equivalent for case $\rho_1 = \rho_2$, but 25 % less in the case $\rho_2 = 100\rho_1$, but also in terms of computation time. The sensibilities are in this case much easier to compute analytically since the X_i's and the Y_i's are considered independent from each other there is no differentiation of Y_i's with respect to the X_i's (in terms of computation it saves a product of matrices per iteration). This is unexpected since SAND2 has more optimization variables (40 while AIO has only 30) and has $N = 10$ more constraints (equality $f(X_i) = Y_i$) but as said in [3] the original SAND formulation (where the strengths are added to the optimization variables while in our case it is the stiffness variables) prevents to continually reanalyze the structure and can be more efficient than AIO. The difference between SAND2 and AI0 is depicted Fig.

5.7.3 Finding a good initial design

We anticipate a bit with the next chapter where we detail the standard sub-optimal decomposition algorithm called StiffOpt. We will not describe it here but simply uses to speed up convergence of the mono-level schemes. Indeed it can be used to get a good initial point since it makes the weight rapidly decreases (after one or two iterations) the weight is quite close to convergence). We then re-investigated the mono-level methods to see if it could give good starting points. Indeed all of these formulations (AIO, SAND2, SAND3) seem very stable (i.e, not initial point too much depending and

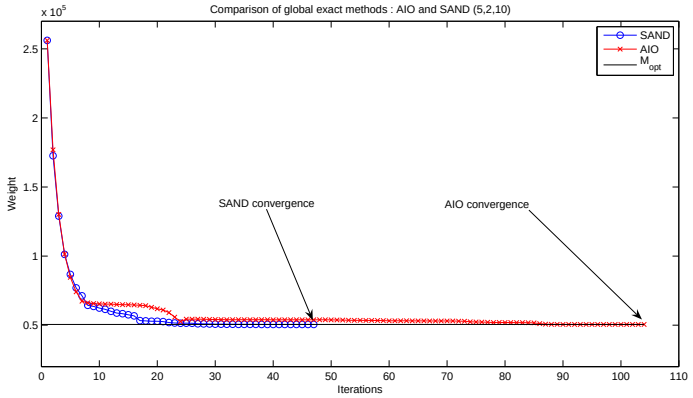


Figure 5.9: Comparison of SAND2 and AIO

converge for all load cases). Numerical experiments show that convergence is very much sped up when starting from a neighborhood of the optimum. One could naturally ask how to find a good initial point:

- As we will see in the next chapter, the algorithm called StiffOpt gives a **near optimal point very quickly** (since it converges in a few iterations and the computations can be distributed). The idea was then to **couple Stiffopt** with all of these formulations.
- We start from an initial (better if feasible) point and **run StiffOpt for one or two iterations** and use the configuration obtained as**an initial point for AIO, SAND2 and SAND3**

The algorithm is then

1. X_0 initial configuration
2. Run 1 or 2 iterations of StiffOpt : gives X_{stiff}
3. Start AIO, SAND2 or SAND3 (with the modifications needed) at X_{stiff}

Numerical results show that convergence is much faster with using this strategy. In average half of the number of iterations is required to converge (see detailed results). Results are even better with only one iteration of StiffOpt_R.

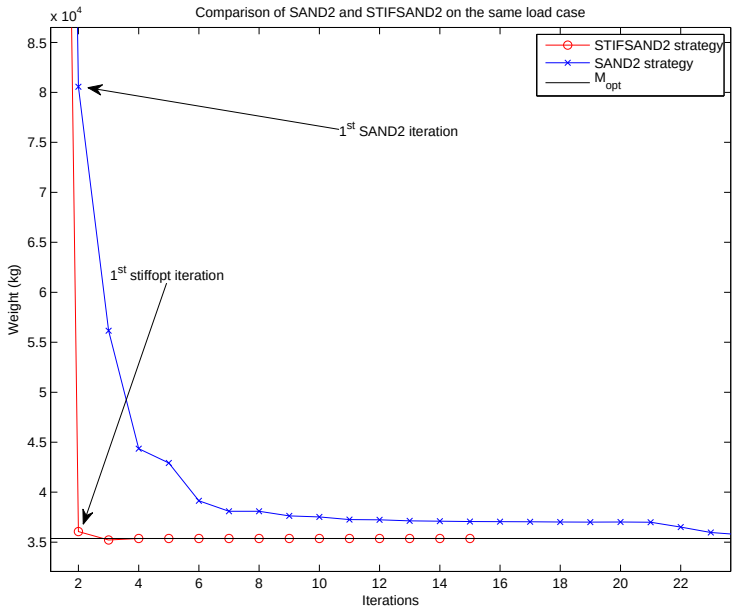


Figure 5.10: Comparison between STIFAND2 and SAND2

5.7.4 Overall comparison

We compared all the mono-level methods. Including the StiffOpt initialization to assess first the different optima and also see if one of the mono-level methods (that do not use the same set of design variables) seems to be better. In this case, this would give a natural decomposition onto variables that can be used for multilevel technique.

We see that the SAND2 and SAND3 methods perform slightly better than AIO. This favors decomposition/bilevel methods based on a decomposition with stiffness terms at upper level and a lower level made of local design variables in comparison with methods that would only decompose the problem over X.

5.8 Towards decomposition/bilevel schemes

We describe our test case and apply on it the different monolevel schemes we saw in the second chapter. We did not test and implement the IQP-BAO version, for it has mainly a theoretical interest. We see that getting the **analytic sensitivities is**

Formulation	N_iter	T (s)	Best case	Worst case
AIO	42	1.02	21	117
SAND2	39	0.83	20	101
SAND3	**36**	**0.63**	21	59
STIFAIO	**18.8**	1.01	4	40
STIFSAND2	22.4	1.4	4	43
STIFSAND3	19.5	**0.83**	4	47

Table 5.1: Synthesis of the results for mono-level methods

far from trivial and that the different **practical formulations we apply gives the same optimum**, which numerically proves that they are equivalent. The block-angular versions (SAND2, SAND3) seems also to **perform better in terms of number of iterations and execution time**. This results in more simple analytic sensitivies computations. Finally we also apply the StiffOpt algorithm presented in the next chapter to get a **good initial design that helps speed up convergence**. We can now turn to decomposition/bilevel schemes implementation and comparison in the next chapter.

6 Bilevel formulations from bi-stages techniques to real bilevel implementation

This chapter evaluates the different multilevel schemes in a realistic context. In particular, we emphasize the importance of the different implementations. Indeed, bilevel formulations as the ones described in Chapter 2, are only theoretical formulations, where some constraints of the upper level are defined through an implicit local optimization. This apparent simplicity of the formulation hides difficulties in terms of implementation, indeed, a bilevel problem may be solved approximately through the resolution of a sequence of optimizations, alternating between upper level optimization and lower level, ***'optimization then optimization'****. It can be solved as a well as it is formulated that is to say a real bilevel implementation,* ***'optimization in optimization'****. The different schemes are then applied and tested over the* 10 *bar truss test case. Alternating optimizations implementation is shown to be sub-optimal and we then turn to a real bilevel implementation of the Quasi Separable subsystem Decomposition that finds out the same optimum is the monolevel schemes.*

6.1 Differences between bi-stages and bilevel optimizations

When facing a bilevel optimization formulation, the question of the implementation is of huge importance. It is an optimization problem where some constraints are computed on the basis of some codes that include the resolution of an optimization. From a conceptual point of view, we could think of these constraints as implicit functions for which we have a code or software that produce the results. It is quite similar to mechanical constraints that involve the resolution of some non-linear problems (large deflections, non-Hookean materials,...). In such a case, the optimization should only see some executable that takes some inputs and gives the physical quantities that are to be constrained. This implies that finally the bilevel problem is in reality a monolevel

problem. It is often denoted this way in the mathematical literature on bilevel optimization. The same way, bilevel optimization can be understood in a different manner. Indeed, most of the structural optimization problems involve at some point equilibrium equations, or eigenproblems whose solutions also satisfy some minimization principles (the unknown displacement u minimizes the strain energy, the critical buckling eigenmode minimizes the Rayleigh quotient,...in terms of elliptic pde's this means that there are symmetric bilinear forms behind), which means even a mono-level optimization scheme such that AIO, solves at each iteration many different optimization problems over different variables and could be called multilevel.

We will not fuss over about the correct term and simply keep the former notations and denominations that is **bilevel optimization is an optimization that includes optimization within some of its constraints**. We should simply keep in mind that the bilevel formulation implies that optimization are performed at different levels. To get to a tractable implementation whenever gradient-based methods are concerned, one should definitely use post-optimal sensitivities to avoid finite-differences computations that would rerun the optimization many times. To help reduce complexity of the solution of the upper level problem, one common heuristic is to approximate the bilevel problems by a sequence of optimizations. As described in (Tosserams *et al.* 2009), the bilevel optimization can be solved

- either as **a real bilevel implementation**, fixing the Y's variables and solving at each global iteration all the optimization local sub-problems
- or **as sequential optimizations**, first solve (entirely) the optimization problem with the local variables fixed and then solve all the local optimization problems with the new Y's fixed and iterate back and forth.

We refer to the first implementation as bilevel implementation and to the second as alternating or bi-stages optimizations. We first describe the implementation and the results obtained for alternating optimizations and then turn to a real bilevel optimization for the QSD scheme.

6.2 Bi-stages schemes

6.2.1 Decomposition based on Fully Stressed Design criterion : StiffOpt

Early attempts to decompose large-scale structural optimization problems on the basis its natural decomposition into structural sub-elements gave rise to a mechanical optimality criteria known as Fully Stressed Desgin criterion. This means that for weight minimization at fixed internal loads, we can decompose the large problem made of N structural sub-elements into N local weight minimization at each element level. Coupled with the stress-ratioing technique, the FSD criterion leads to the algorithm called StiffOpt. This criterion was extensively covered in the eighties, it is a mechanical criterion that appears to be false for a lot of optimization problems. However, it is a quite simple algorithm that can be applied to any structure quite easily. When internal loads vary, this criteria in connection with the stress-ratio update technique usually offers fast convergence. Nonetheless, for hyper-static or redundant structures and also structure made of different materials[1] the FSD criterion is known to be sub-optimal. Classical test cases on truss showed even for simple stress constrained weight minimization the FSD criterion can lead to sub-optimal design for some loading cases. Even worst, buckling constraints make this criterion more false. However, since it can give good initial design points for other methods and that it is a first attempts to decompose, we briefly review the StiffOPt algorithm based on this criterion.

6.2.1.1 Brief description of StiffOpt

Roughly speaking, StiffOpt algorithm starts from an initial design (X local variables), computes the associated stiffness variables and the internal load redistribution. Under this fixed internal loads, every sub-element is optimized with respect to weight under global-local constraints (buckling). This results in new local design variables, and the stiffness terms are updated and so is the internal loads redistribution. Unlike other multilevel schemes presented in the sequel (Target Rigidity,...) StiffOpt does not include an upper level optimization but simply an update and a new FEA. It is worth noting that, if we take the original AIO problem and artificially modify its analytical sensitivities by cutting off the extra block diagonal entries (most of modern gradient-based optimizers

[1] And large-scale composite structures lie in this category whenever stacking sequences are optimization variables, since their stiffness moduli E_x, E_y and ν_{xy} vary with the orientation angles proportions.

(`fmincon` from MathWorks, `nlpql` from Saunders et al., `gcm` from Samtech) accept user-supplied analytical gradients of objective and constraints functions), we end up with same results as StiffOpt.

Indeed, StiffOpt is formally

1. For a given X^k, we compute Y^k and Φ^k
2. Solve

$$\text{(StiffOpt)} \qquad \begin{array}{ll} \min_{X\in\mathbb{R}^n} & M(X) = \sum_{i=1}^{N} m^{(i)}(X^{(i)}) \\ \text{s.t} & \begin{cases} \hat{c}^{(i)}(X^{(i)}, \Phi^k) \leq 0 \\ c_{loc}^{(i)}(X^{(i)}) \leq 0 \end{cases} \end{array} \tag{6.1}$$

3. Update X^{k+1} check convergence

6.2.1.2 Improvement of StiffOpt based on sensitivity

The former algorithm gives a natural way to improve StiffOpt. Indeed, we see that the internal loads are not updated whilst the local optimization process. This can be done by computing the sensitivities of stress distribution at step 1 and use them to update stress distribution using a Linear Approximation (or Reciprocal)

1. For a given X^k, we compute Y^k, Φ^k and $D^k\Phi = \left(\frac{\partial \Phi_i}{\partial Y_j}(Y_k)\right)_{1\leq i,j\leq N}$
2. Solve

$$\text{(StiffOpt\_L)} \qquad \begin{array}{ll} \min_{X\in\mathbb{R}^n} & M(X) = \sum_{i=1}^{N} m^{(i)}(X^{(i)}) \\ \text{s.t} & \begin{cases} \hat{c}^{(i)}(X^{(i)}, \Phi^k + (\varphi^i(X_i) - Y_k)\frac{\partial \Phi_i}{\partial Y_i}) \leq 0 \\ c_{loc}^{(i)}(X^{(i)}) \leq 0 \end{cases} \end{array} \tag{6.2}$$

3. Update X^{k+1} and check convergence

Note that we used the classical Linear Approximation, better results are obtained with the Reciprocal Approximation (more suitable for structural optimization problems)

$$\text{(StiffOpt\_R)} \qquad \begin{array}{ll} \min_{X\in\mathbb{R}^n} & M(X) = \sum_{i=1}^{N} m^{(i)}(X^{(i)}) \\ \text{s.t} & \begin{cases} \hat{c}^{(i)}(X^{(i)}, \Phi^k + (\varphi^i(X_i) - Y_k)\frac{Y^k}{\varphi^i(X_i)}\frac{\partial \Phi_i}{\partial Y_i}) \leq 0 \\ c_{loc}^{(i)}(X^{(i)}) \leq 0 \end{cases} \end{array} \tag{6.3}$$

Results are better than the original StiffOpt, as observed in Fig. 6.1, in terms of optimal weight and number of iterations.

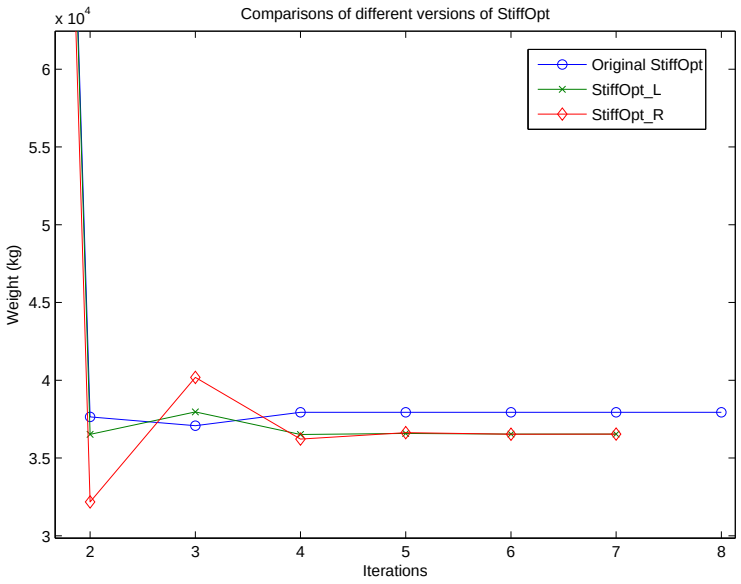


Figure 6.1: Comparison between different versions of StiffOpt

6.2.2 Bi-stages or alternating optimization schemes

This section is the most important since we apply here the most classical decomposition techniques in structural optimization, these were already outlined in Chapter 2, but we make them more precise here. However, it is worth noting that we did implement and test these schemes on the basis of bi-stages optimization (a.k.a alternating optimizations), which means that we first optimize the upper level and then, based on the results obtained at this level, we realize local optimizations where upper level parameters (mainly stiffness's and hence internal loads redistribution fixed). The local optimization results are sent to upper level to update it, this process goes back and forth and is repeated alternatively to reach convergence. This is a fixed point resolution, the underlying idea is that real bi-level programming is too computationally expensive to be interesting. We refer also refer to these implementations as 0-order schemes, since at this point, we do not consider post-optimal sensitivities of local optimizations. In the next section, we will describe a fully bilevel scheme : Quasi Separable Subsystem Decomposition that is simply the real bilevel implementation of the Maximum Margin scheme that we describe in this section. This real bilevel is made possible thanks to

the use of post-optimal sensitivities and surrogate models (in the next chapter).

6.2.2.1 Formal description

Recall that the typical bilevel decomposition has the following general form.

- Global-level (or system-level) :

$$\begin{array}{ll} \min_Y & M(Y) = \sum_{i=1}^{N} m_i(Y_i) \\ \text{s. t.} & \begin{cases} c^K_{glob}(Y_1, \ldots, Y_N) \leq 0 \\ \gamma_i(Y) \leq 0 \quad \text{for} \quad i = 1 \ldots N \end{cases} \end{array} \tag{6.4}$$

- where γ_i for $i = 1 \ldots N$ is the coupling function computed from the lower-level (or sub-element level) :

$$\begin{array}{ll} \min_{X_i} & h_i(X_i, Y) \\ \text{s. t.} & \begin{cases} c^{(i)}_{glob-loc}(Y, X_i) \leq 0 \\ c^{(i)}_{loc}(X_i) \leq 0 \end{cases} \end{array} \tag{6.5}$$

- that way the hard constraints (based on FEA) are only computed on the global-level and the N local optimizations can be performed concurrently

6.2.2.2 Target Rigidity

The main idea is to first realize optimization at upper level where internal loads vary, taking into account stress and strain constraints. This gives optimal stiffness terms (cross-sections, quadratic momentum whenever bending is involved...) and at local level we try to find the local design that best matches the stiffness terms (i.e the closest in terms of Euclidean distance over dimensions) and that satisfies local constraints (buckling). This way the internal loads are expected not to vary too much when updating the stiffness terms at upper level.

- **Target Rigidity** defines $\gamma_i(Y) = \frac{1}{2}|\varphi(X_i^*) - Y_i|^2$

- where X_i^* is

$$\begin{array}{ll} \text{argmin}_{X_i} & \frac{1}{2}||\varphi(X_i) - Y_i||^2 \\ \text{s. t.} & \begin{cases} c^{(i)}_{glob-loc}(Y, X_i) \leq 0 \\ c^{(i)}_{loc}(X_i) \leq 0 \end{cases} \end{array} \tag{6.6}$$

As already outlined, Target Rigidity is very close to Collaborative Optimization in the MDO framework. From constraint $\gamma_i(Y) \leq 0$ at upper level, we see that at convergence of the global-level optimization, we expect to have a local design that perfectly matches the stiffness variables and hence this local design is feasible since it is the results of local optimization that includes global-local constraints.

6.2.2.3 Maximum margin (a.k.a Constraint margin)

Recall that this scheme was carried out first by Haftka and Sobiesky in the 80's and then extended in a more general form by Haftka, Liu and Watson under the QSD scheme.

- **MaxMargin** defines $\gamma_i(Y) = \min_{X_i \in \varphi^{-1}(Y)} \{c^{(i)}_{glob-loc}(Y, X_i^*)\}$ and therefore $\gamma_i(Y) = -\mu_i^\star$
- where $\mu_i^\star$ is equal to

$$\begin{array}{ll} \text{argmax}_{\mu_i, X_i} & \mu_i \\ \text{s. t.} & \left\{ \begin{array}{l} c^{(i)}_{glob-loc}(Y, X_i) + \mu_i \leq 0 \\ c^{(i)}_{loc}(X_i) + \mu_i \leq 0 \\ Y_i = \varphi(X_i) \end{array} \right. \end{array} \tag{6.7}$$

Note that the very name **Maximum Margin** comes from the fact that the local optimization seeks the local design that least violates both the global-local constraints and the local constraints (e.g buckling constraints and local ratio constraints). In this case we maximize the margin (the feasibility) and constrain at upper level this margin to be positive. It would have been totally equivalent to minimize the negative margin and constrain at upper level this margin to be negative.

6.2.2.4 Local mass minimization MinMass

This scheme was carried out by Merval in his PhD thesis (Merval 2008).

- MinMass defines $\gamma_i(Y) = c^{(i)}_{glob-loc}(Y, X_i^*)$
- where X_i^* is

$$\begin{array}{ll} \text{argmin}_{X_i} & m_i(X_i) \\ \text{s. t.} & \left\{ \begin{array}{l} c^{(i)}_{glob-loc}(Y, X_i) \leq 0 \\ c^{(i)}_{loc}(X_i) \leq 0 \\ Y_i \leq \varphi(X_i) \end{array} \right. \end{array} \tag{6.8}$$

6.2.2.5 Mix scheme

- This scheme is **a combination of MinMass and Target Rigidity**
- $\gamma_i(Y) = \max\{c^{(i)}_{glob-loc}(Y, X_i^*), |\varphi(X_i^*) - Y_i|\}$
- where X_i^* is

$$\begin{array}{ll} \text{argmin}_{X_i} & \varepsilon\frac{1}{2}|\varphi(X_i) - Y_i|^2 + (1-\varepsilon)m_i(X_i) \\ \text{s. t.} & \begin{cases} c^{(i)}_{glob-loc}(Y, X_i) \leq 0 \\ c^{(i)}_{loc}(X_i) \leq 0 \\ Y_i \leq \varphi(X_i) \end{cases} \end{array} \tag{6.9}$$

6.2.2.6 Practical implementation as alternating optimizations

We see that a bilevel optimization involves solving local optimization problems to compute constraints of the global optimization problem. As outlined before, one way to reduce the complexity is to consider the stiffness terms and stress distribution are fixed during the local optimizations and we do not take into consideration the variation of the X_i's while optimizing the Y's at the global level. The algorithm is therefore:

1. Iteration k : given detailed configuration X^k
2. Resolution of $\{\min_Y M(Y) | c^K_{glob}(Y) \leq 0, c_{glob-loc}(Y, X^k) \leq 0\}$
3. We obtain $Y_1^k, \ldots, Y_N^k$
4. For $i = 1 \ldots N$, we solve $\{\min_{X_i} h_i(X_i, Y) | c_{glob-loc}(Y^k, X_i) \leq 0\}$
5. Update X^{k+1} and M^{k+1}
6. Check convergence for M and X^k and check feasibility of the structure, if convergence end if not go to 1.

6.3 Overall comparison for alternating optimizations implementation

6.3.1 StiffOpt

StiffOpt has a very good behavior in the first case ($\rho_1 = \rho_2$). Typical behavior in this case is depicted Fig. 6.2. We can see that we have a very fast decreasing weight and that StiffOpt always gives feasible configuration.

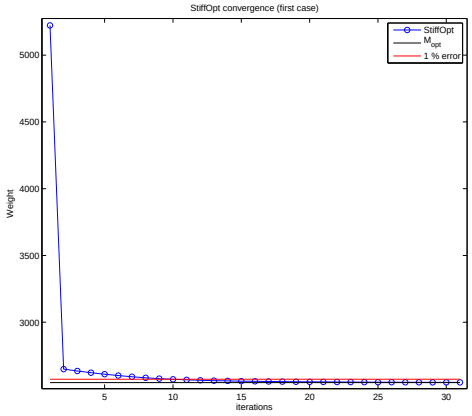
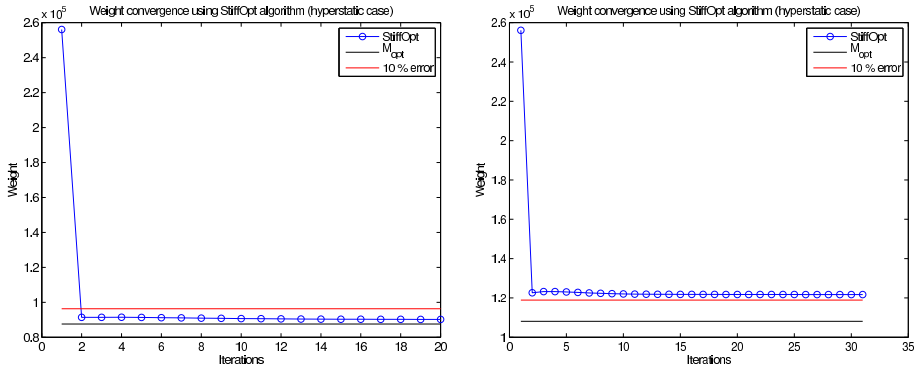


Figure 6.2: Usual behavior of StiffOpt in the first case

But in the second case ($\rho_2 = 100\rho_1$), StiffOpt does not behave that nice, we usually observe an error at convergence of $3 - 4\%$. Average behavior in that case is depicted Fig. **??** and for few cases the error goes up to 12%, this is depicted Fig. 6.3. We noted the following interesting fact that in the second case, the weight at optimum is nothing else than the weight of the optimum found in the first case with respect to the $\rho_2 = 100\rho_1$. In other words, if we take the optimal configuration found in the first case and compute its weight for $\rho_2 = 100\rho_1$ we find the same weight as StiffOpt. StiffOpt optimal configurations are actually very close in both cases.



(a) Average behavior of StiffOpt in the second case

(b) Worse behavior of StiffOpt

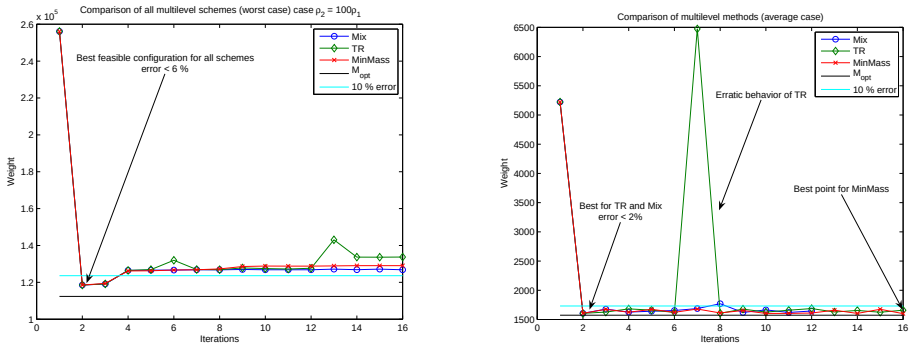
Figure 6.3: Average and worse cases for StiffOpt in the hyperstatic case

6.3.2 Overall comparison

Eventually, we compared the behavior of all methods for two load cases (in the second case $\rho_2 = 100\rho_1$). We obtained the following results over 40 different load cases for the ten bar truss

- **FSD criterion** : relative error 6% in 14 iterations
- **Target Rigidity** : relative error 3.2% in 5 iterations
- **MinMass** : relative error 3.1% in 4 iterations
- **Mix** : relative error 2.9% in 5 iterations
- **MaxMarge** : relative error 12% in 4 iterations

Finally, we depicted Fig. 6.4 and 6.5 the results we obtained for MinMass, Target Rigidity and the so-called Mix schemes, for the same starting points. Note in the appendix are shown the detailed results load by load case. Note also we do not converge for all load cases and the minimum weight reported is the best feasible configuration over the optimization history.



(a) Comparison of multilevel methods (worse case)

(b) Comparison of multilevel methods (average case)

Figure 6.4: Comparison of multilevel methods : worst and average cases in the second case ($\rho_2 = 100\rho_1$)

Finally, we depicted the erratic and ill-conditioned behavior of MaxMargin Fig. 6.6, where we studied the behavior for close starting points. The equality constraint seem

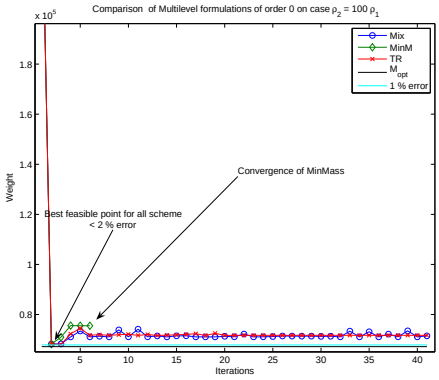


Figure 6.5: Comparison of multilevel methods : best in the second case ($\rho_2 = 100\rho_1$)

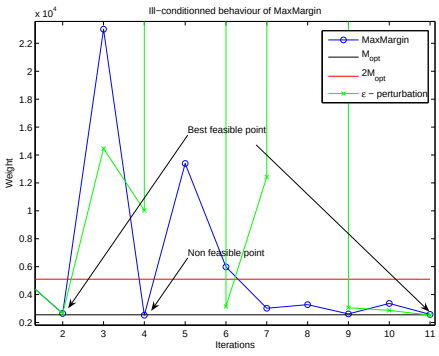


Figure 6.6: Erratic behavior of MaxMargin in the first case ($\rho_2 = \rho_1$)

As a conclusion, we observed that over the various bilevel/decomposition schemes, **Maximum Margin has a very erratic behavior** while **MinMass, Target Rigidity and the Mix schemes perform better**. However, all of them are **sub-optimal**.

6.4 Adaptation of QSD scheme to composite structural optimization problem

In this section, we first recall the QSD scheme, already presented in Chapter 2 and describe the adaptation we have made to solve our original problem first by relaxing the equality constraints as suggested in Chapter 2 and second with adaptation to specific composite considerations: lamination parameters as described in Chapter 3. In the application of this scheme to a realistic structure, we also applied our original strategy

to build approximation of buckling constraints computed through Airbus in-house tools. This is described in Chapter 7.

6.4.1 Adaptation to equality constraints

Haftka et al. carried out the initial scheme

- Global-level (or system-level) :

$$\begin{array}{ll} \min_Y & M(Y) = \sum_{i=1}^{N} m_i(Y_i) \\ \text{s. t.} & \left\{ \begin{array}{l} c_{glob}(\Phi(Y_1, \ldots, Y_N)) \leq 0 \\ \mu_i^\star(Y_i, \Phi_i) \leq 0 \quad \text{for} \quad i = 1 \ldots N \end{array} \right. \end{array} \tag{6.10}$$

- where $\mu_i^\star$ for $i = 1 \ldots N$ is the coupling function computed from the lower-level (or sub-element level) :

$$\begin{array}{ll} \min_{(\mu_i, X_i)} & \mu_i \\ \text{s. t.} & \left\{ \begin{array}{l} c_{glob-loc}^{(i)}(Y_i, \Phi_i, X_i) - \mu_i \leq 0 \\ c_{loc}^{(i)}(X_i) - \mu_i \leq 0 \\ Y_i = \phi(X_i) \end{array} \right. \end{array} \tag{6.11}$$

As observed in the former section, the **equality constraint makes the algorithm very unstable and ruins convergence**. **We adapted this formulation based on the inexact quadratic penalty block angular version presented in Chapter 2 (IQP-BAF)**. This allows us to add *budgets* b_i as suggested by Haftka, Liu and Watson.

- Global-level (or system-level) :

$$\begin{array}{ll} \min_{Y,b} & M(Y) = \sum_{i=1}^{N} m_i(Y_i) + \sum_{i=1}^{N} b_i \\ \text{s. t.} & \left\{ \begin{array}{l} c_{glob}(\Phi(Y_1, \ldots, Y_N)) \leq 0 \\ \mu_i^\star(Y_i, \Phi_i, b_i) \leq 0 \quad \text{for} \quad i = 1 \ldots N \end{array} \right. \end{array} \tag{6.12}$$

- where $\mu_i^\star$ for $i = 1 \ldots N$ is the coupling function computed from the lower-level (or sub-element level) :

$$\begin{array}{ll} \min_{(\mu_i, X_i)} & \mu_i \\ \text{s. t.} & \left\{ \begin{array}{l} c_{glob-loc}^{(i)}(Y_i, \Phi_i, X_i) - \mu_i \leq 0 \\ c_{loc}^{(i)}(X_i) - \mu_i \leq 0 \\ \frac{1}{2}||Y_i - \varphi(X_i)||^2 - b_i - \mu_i \leq 0 \end{array} \right. \end{array} \tag{6.13}$$

6.4.2 Adaptation to composite specific considerations

In this section, we explain how to adapt this QSD scheme to the specific composite considerations, in particular, we use here the different results obtained in the framework of this thesis. The main idea is that we want to decompose the main problem variables into two sets of distinct variables that will be of course connected to each other and keep this distinction between upper level variables, whose FEA depends on and lower level variables, whose buckling constraints depend on. For geometric dimensions, this connection was ensured through cross-section dimensions $Y = \varphi(X)$. Indeed, FEA depends only on Y and once the internal loads are computed on the basis of the Y's, the local design X's (geometric dimensions) are used to compute the stability constraints (buckling) with loads as parameters. For laminate, this is not that easy. Suppose for instance that we work with ply angles orientations as optimization variables, we can not separate them between stiffness-dependent variables and stability-dependent variables and the number of variables depends on the thickness of the laminate. However if we think of constitutive law's in terms of A, B and D tensors, we see that they are much better suited to decomposition. Indeed

- They can vary **continuously**
- There is always the **same number of variables** for they are 3 dimensional symmetric tensors
- In case of symmetric laminate ($B = 0$), the **in-plane (membrane) and out-of-plane (bending) behavior can be decoupled**

$$\begin{pmatrix} N \\ M \end{pmatrix} = \begin{pmatrix} A & 0 \\ 0 & D \end{pmatrix} \begin{pmatrix} \varepsilon \\ -\kappa \end{pmatrix} \tag{6.14}$$

  where ε is the strain tensor (plane strain $\varepsilon = (\varepsilon_x, \varepsilon_y, \varepsilon_{xy})$)), κ are the curvature components, N the forces and M moments. We see that the in-plane behavior does not depend on D.

We see that we have kind of a natural decomposition through this constitutive equation. Roughly speaking, everything that depends on internal loads redistribution will be linked to A and everything that depends on out-of-plane behavior (e.g buckling) will be linked to D. This quite general, for laminate, this means that

- membrane behavior (maximum deformation, Poisson ratio mismatch,...) will only depend on **proportion of orientations**, or equivalently in relative thickness. A depends on the proportion
- out-of-plane behavior will only **depend on the order of stacking**.

This advocates the use of A and D tensor variables as optimization variables. However, these variables are not very easy to handle. In particular it might not be very simple to describe the feasible space and the constraints connection between A and D. Furthermore, there are 12 variables. This is why we used another continuous representation, namely the lamination parameters ξ. We saw in Chapter 3 that they are continuous variables and they describe as well the in-plane behavior (ξ_A) as the out-of-plane (ξ_D). Moreover, some (but not all) manufacturability constraints can be imposed in the lamination space and much work has been done to write the connection constraints between ξ_A and ξ_D (compatibility equations). Next, out-of-plane lamination parameters are also quite a good choice for optimizations variables, since the skin buckling critical reserve factor is concave over ξ_D. However it is worth noting that the buckling constraints are still coupled with ξ_A and Y through internal loads redistribution. Indeed, recall that buckling critical reserve factor is defined as the smallest positive eigenvalue λ such that

$$D(\xi_D, X)\Delta^2 w = -\lambda N(\xi_A, Y)\Delta w \tag{6.15}$$

and Eq. (6.15) is a simplified form of the real buckling equation for composite plates, however we clearly see the dependence of the operators involved in buckling. In terms of the QSD scheme, this turns out to be the following representation

- $Y = (A_i, \xi_A^{(i)}, b_i)$ for $i = 1 \dots N$, where A_i is the cross section area for super-stringer and $\xi_A^{(i)}$ are the lamination parameters for skin and stringer and b_i the budget. Note that this is totally equivalent to use proportions of angles orientations since we have (for conventional $[0/90/45/-45]$ laminates) the following relationship

$$\xi_A^1 = p_0 - p_{90} \tag{6.16}$$

$$\xi_A^2 = p_0 + p_{90} - 2p_{45} \tag{6.17}$$

$$\xi_A^3 = 0 \tag{6.18}$$

whenever the laminate is balanced. We see that proportions can be easily derived from in-plane lp's and the other way around. In both cases, it boils down to only 2 variables per stacking. Note that all the maximum deformation, maximum displacement or maximum stress constraints can be computed on the basis of these variables.

- $X^{(i)} = (e_i, \xi_D^{(i)})$ where e_i are the local dimensions (e.g stringer profile dimensions,...) and $\xi_D^{(i)}$ are the out-of-plane lamination parameters of skin and stringer. For conventional $[0/90/45/-45]$ laminates $\xi_D^4 = 0$ but in general we have $\xi_D^3 \neq 0$ even for balanced laminates.

The optimization formulation is therefore

- $M(Y) = \sum_i^N A_i + \sum_i^N b_i$ the total weight penalized by the budgets.
- The local optimization is then

$$\begin{array}{ll} \min_{(\mu_i,(e_i,\xi_D^{(i)})} & \mu_i \\ \text{s. t.} & \begin{cases} RF^{(i)}_{glob-loc}(\Phi_i, e_i, \xi_D^{(i)}) - \mu_i \leq 0 \\ c^{(i)}_{comp}(\xi_A^{(i)}, \xi_D^{(i)}) \leq 0 \\ c^{(i)}_{loc}(, e_i) - \mu_i \leq 0 \\ \frac{1}{2}||A_i - \varphi(X_i)||^2 - b_i - \mu_i \leq 0 \end{cases} \end{array} \tag{6.19}$$

The reason why we do not use the margin in the compatibility constraints between ξ_A and ξ_D is that we numerically observed that the real feasible (manufacturable) stacking sequences often lie on the boundary of the domain defined by these equations.

6.5 Detailed results

In this section, we give results and details about the practical implementation for the academic test case, the 10 bar truss. This test case is complicated enough to see what are the key points of such an implementation. Roughly speaking, we obviously need an optimizer that accepts constraints computed through an optimization. This means that if the same algorithm is used at both levels it can be called inside the main call. The other main point is about sensitivity computations, we need to compute post-optimal sensitivity $\frac{\partial \mu_i^*}{\partial Y_i}$, if finite-differences are to be used this will become totally intractable, since another local optimizations are to be called for perturbed optimization parameters. Specific post-optimal sensitivity needs to be set.

6.5.1 Practical implementation

There is quite a big step between the formal description and the implementation of a bilevel optimization. In this section, we describe on a the (not so simple) example of the 10 bar truss how this implementation was made. For some points (especially chain ruling of sensitivities), we will see it was even more complicated than the MAAXIMUS panel implementation in the next chapter, so it is worthwhile to have quite a good picture of what is done for this test-case.

6.5.1.1 Post-optimal sensitivities and chain ruling

First, gradient based optimization methods are used, which means that we need to differentiate the optimal margin $\mu^\star$ with respect to Y terms and of course b budgets as well. This is a bit tricky since the parameters of the local optimization involved are not directly Y but quantities computed on the basis of these Y's namely forces (or loads). This particular derivative is of very huge importance since it couples all the sub-elements. Former 0-order multilevel schemes are in fact quite the same algorithm but this derivative set to 0. Roughly speaking, the upper level does not see the change in the local optimization induced by a perturbation of the parameters while they vary during the main loop optimization.

Regarding this post-optimal sensitivity, note that

- $\mu^\star(Y,b) = (\mu_1^\star(Y_1, \Phi_1(Y), b_1) \dots \mu_N^\star(Y_N, \Phi_N(Y), b_N))$ can be computed in a distributed way and analytical derivatives are available

$$\frac{d\mu_i^\star}{dY_i} = \frac{\partial \mu_i^\star}{\partial Y_i} + \frac{\partial \Phi_i}{\partial Y_i}\frac{\partial \mu_i^\star}{\partial \Phi_i} \tag{6.20}$$

- where $\frac{\partial \mu_i^\star}{\partial Y_i}$ and $\frac{\partial \mu_i^\star}{\partial \Phi_i}$ are post-optimum sensitivities computed with the Lagrange multipliers λ found at local optimization, based on post-optimal sensitivities results

$$\frac{\partial \mu_i^\star}{\partial Y_i} = \lambda^T \frac{\partial c^i}{\partial Y_i} \tag{6.21}$$

- where $\frac{\partial c^i}{\partial Y_i}$ is the derivative of local constraints w.r.t to Y_i and $\frac{\partial \Phi_i}{\partial Y_i}$ is the derivative of the FEA response found by direct or adjoint methods (e.g **SOL200** sensitivity analysis responses).

6.5.1.2 Optimizers

As outlined before, the optimizers should be called in the main loop whose constraints call the same optimizers. If we think pf μ_i^* as an implicit function that can be computed through any programming code, we therefore think of it as an external task. However, this can be done in Matlab with the `fmincon` function and also in Boss Quattro. In terms of optimization architecture, this means that we have a call to the optimizer in the function that computes the constraints. For instance, the main call will be of the kind

```
[x_opt,f_opt,exitflag,output,lambda] = fmincon(@(X)main_objective(X,strcase,truss),..
X_0,[],[],[],[],lbnd,ubnd,@(X)main_contr(X,lc,truss),options2);
```

where `main_contr` is of the following structure

```
function [con,con_eq,grad,grad_con_eq] = main_contr(X,lc,truss)

% Computation of yield stress constraints
...
% Sensitivity of yield stress constraints
...

% Local optimization based constraints computation and post-optimal sensitivity call
```

```
for i=1:10
    if nargout < 3
        [x_opt,f_opt,exitflag,output,lambda] = ...
        sub_opt([0 1 10 0.01],Y(i),phi(i),b(i),i,50,truss);
        ...
    elseif nargout > 2
        [x_opt,f_opt,exitflag,output,lambda,loc_rig,err_rig,...
        d_mu_d_y,d_mu_d_phi,d_mu_d_b] =...
        ...sub_opt([0 1 10 0.01],Y(i),phi(i),b(i),i,50,truss);
        ....
        % chain rule
        grad_con(20+i,1:10) = (d_mu_d_y.*(ind == i)'+...
        grad_phi(i,ind)'.*d_mu_d_phi);
        grad_con(20+i,11:20) = d_mu_d_b.*(ind == i);
...
end
```

where sub_opt is the local optimization calls

```
function [x_opt,f_opt,exitflag,output,lambda,loc_rig,...
  err_rig,d_mu_d_y,d_mu_d_phi,d_mu_d_b] =...
  ...sub_opt(X_0,Y_i,Phi_i,b_i,i,N_iter,truss)
...
[x_opt,f_opt,exitflag,output,lambda] = fmincon(@(X)sub_objective(X),...
X_0,[],[],[],[],lbnd,ubnd,...
@(X)sub_constr(X,Y_i,Phi_i,b_i,i,truss),options);
...
if nargout > 5
    ...
    if nargout > 7
        ...
        % post optimal sensitivities
        d_mu_d_y     = lambda.ineqnonlin'*[grad_y_c_1;grad_y_c_2;grad_y_c_3];
        ...
        % Post-optimal sensitivities comptutation
        d_mu_d_phi   = lambda.ineqnonlin'*[grad_phi_c_1;grad_phi_c_2;grad_phi_c_3];
        ...
```

```
        d_mu_d_b      = lambda.ineqnonlin'*[grad_b_c_1;grad_b_c_2;grad_b_c_3];
    end
end
```

6.5.2 Results for 10 bar truss

The main important result from this numerical experiments is that **QSD converges to the same optimum as AIO for all load cases**. We used in a first time all the same starting points as the ones previously used for alternating optimizations implementation. The typical behavior of QSD versus AIO is depicted Fig. 6.7 and 6.8.

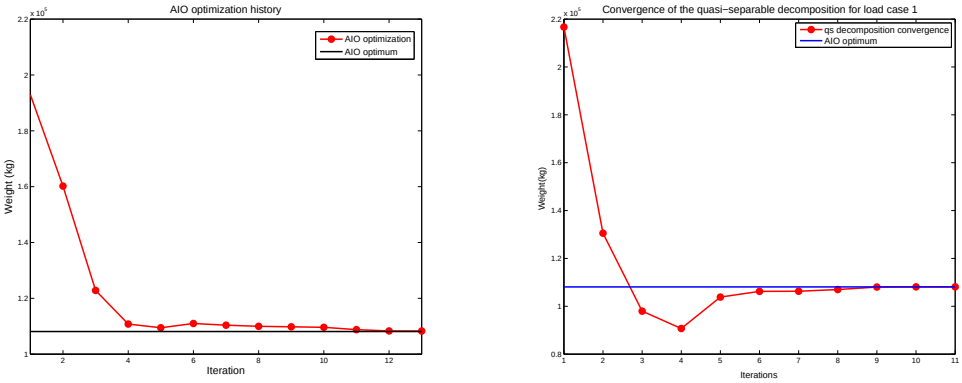


Figure 6.7: Comparison of AIO optimization and proposed decomposition on load case 1

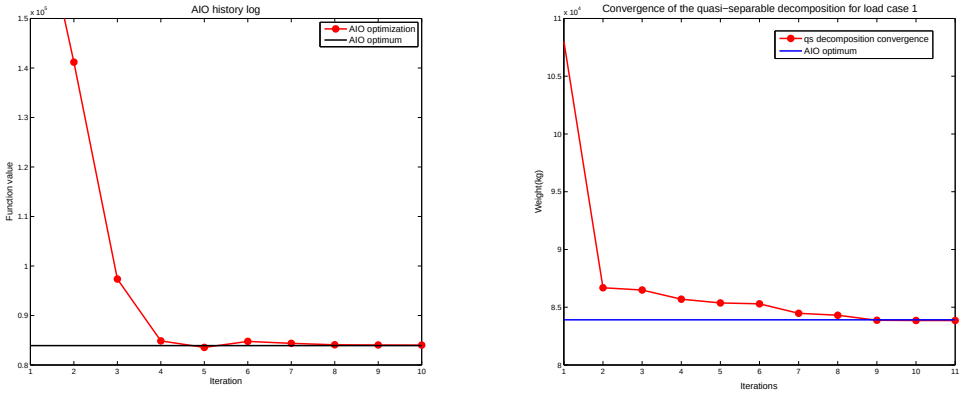


Figure 6.8: Comparison of AIO optimization and QSD on load case 2

To validate this behavior with many more optimizations, we extended our numerical experiments through a collaboration with NLR in the framework of the MAAXIMUS project (see the corresponding section **Collaborations & Publications**), we compared the AIO optimization and our adaptation of QSD optimization on the 10 bar truss

- For 21 different starting points randomly generated in $\mathbb{R}^{30}$
- For the 40 load cases
- Both configurations : isostatic ($\rho_1 = \rho_2$) and hyperstatic ($\rho_1 = 100\rho_2$)

Resulting in 1680 optimizations for which we compared the number of FEA's and overall optimizations iterations (global cycle optimizer) before convergence. **AIO converged for** 1634 **cases while QSD for** 1574 **cases**. For the optimizations that did converge, each time the optimum was the same as the AIO one, over the cases that both converged, we had

- **For AIO, on average :** 171 **FEA's (including sensitivities computations) and** 64 **global cycle iterations**
- **For QSD, on average :** 170 **FEA's (including sensitivities computations) and** 51 **global cycle iterations**

Fig. 6.9, 6.10, 6.11 shows the comparison between AIO and QSD for the first three load cases, for each load case, we depict the total number of finite elements call (stiffness assembly plus resolution of linear system), we distinguish the calls for simple constraints evaluation from the calls for sensitivities evaluations. The same way we compare the number of global iterations before convergence. Note that a feature of the SQP implementation is that it often calls several times the objective and constraint functions per optimization iteration. While others algorithms, like CONLIN for instance, call once the functions per iteration.

6.6 Towards a real aircraft structure

We saw that the alternating optimizations implementation **failed to attain the real optimum**, while the real **bilevel implementation of QSD allows to find the same**

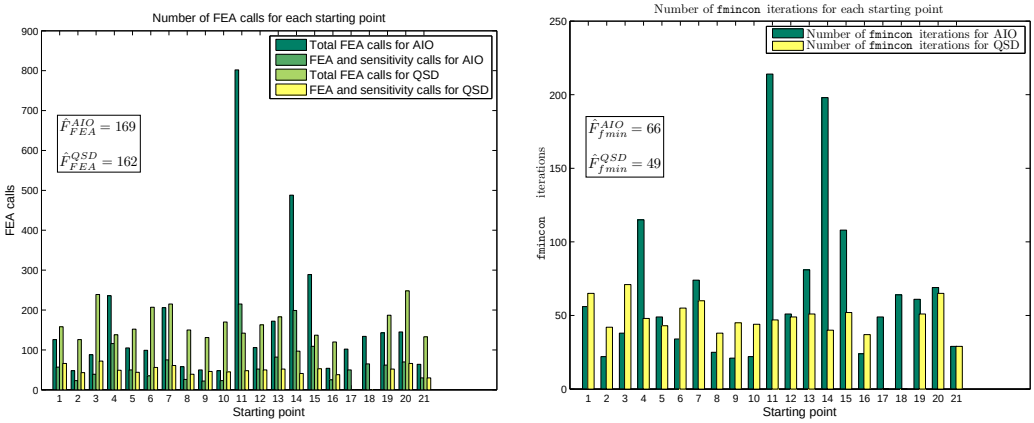


Figure 6.9: Comparison of AIO and QSD: load case 1

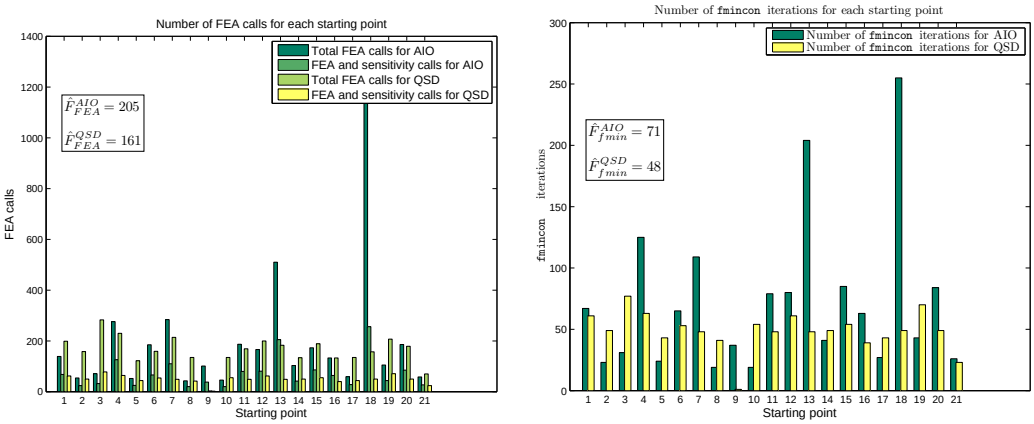


Figure 6.10: Comparison of AIO and QSD: load case 2

optimum as the monolevel methods. Based on this promising result we turn now to the application of QSD to real aircraft structure based on most the material we have presented so far.

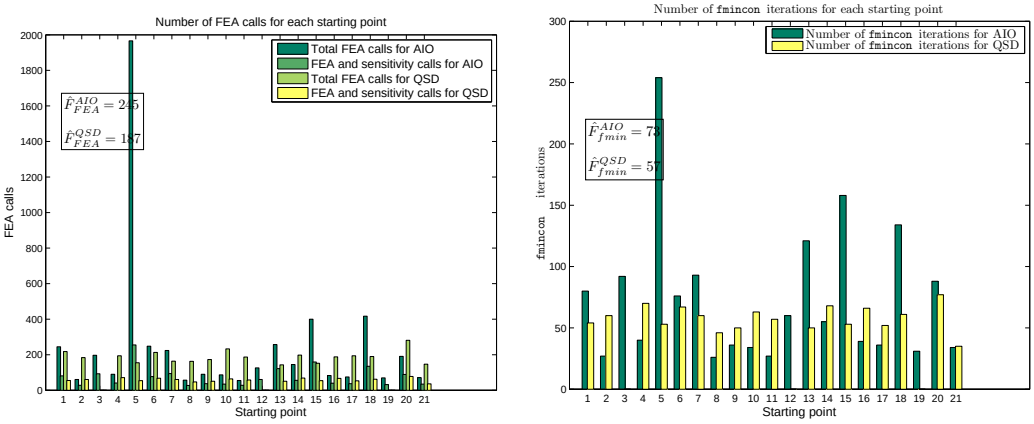


Figure 6.11: Comparison of AIO and QSD: load case 3

7 Application to MAAXIMUS panel

In this section, we describe the implementation of the QSD scheme that we previously described to a realistic aircraft structure, namely a panel of 6×8 *composite panels reinforced by* $\Omega-$*shaped stringers, this test case comes from the European research project MAAXIMUS (More Affordable Aircraft through eXtended, Integrated and Mature nUmerical Sizing) and then will be denoted the MAAXIMUS panel. We start with some specific applications to take into account the stacking sequences. In particular, we describe the construction of surrogate models of Airbus in-house buckling code for varying stacking sequences based on lamination parameters. The set of feasible stacking sequences for manufacturability constraints is briefly described in connection to the construction of approximation models of buckling constraints. We then turn on the implementation and gives some details about the different tools used (*`MSC.Nastran`*,* `Boss Quattro`*).*

7.1 Surrogate models of buckling constraints

We very briefly describe here adaptation to IMAGE that were done in order to take into account the mixed variables we naturally encounter when dealing with laminated composite design and optimization. With very few modification, EM clustering works exactly the same for mixed variables resulting in a globally continuous model over discrete value.

7.1.1 Use of IMAGE for mixed variables for surrogate over lamination parameters

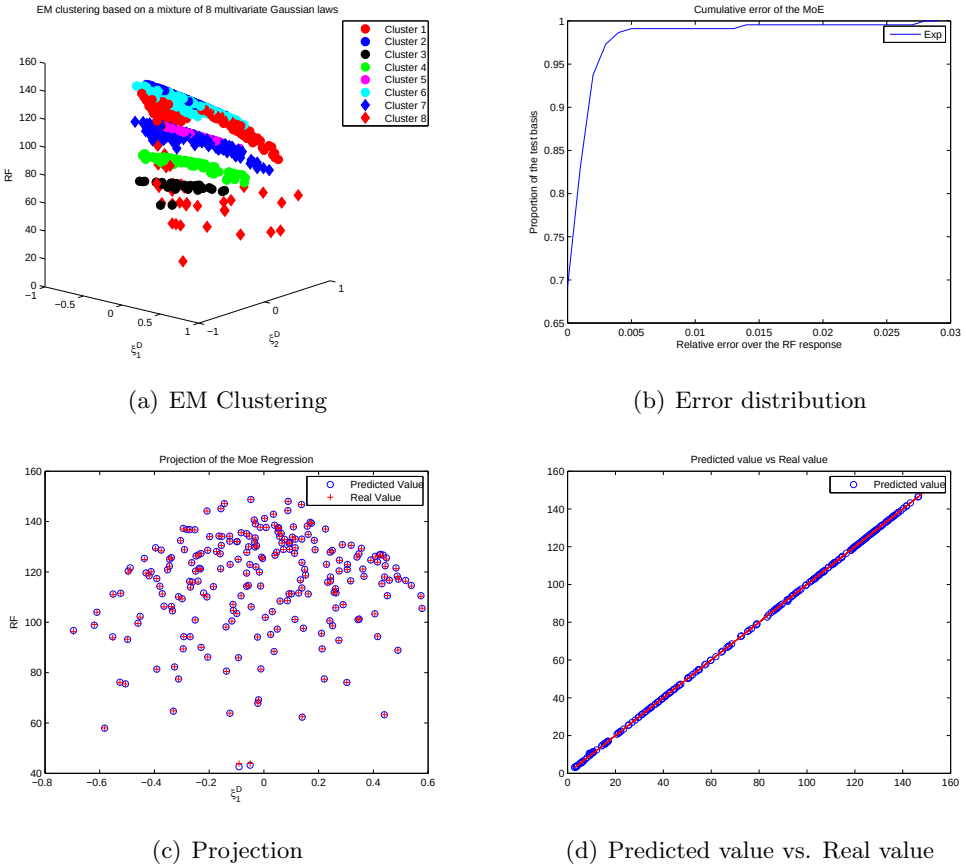
The idea of this extension is simply to add a small perturbation over the discrete values. Say we have $x = [x_1, \ldots, x_d]$ where there is a discrete dimension say x_1. If we run IMAGE, the EM algorithm will not manage to find clusters, essentially there

will be a dimension (discrete one) which does not vary. This leads to a very high ill-conditioning of the variance matrix of the associated cluster. In geometric terms, the ellipsoid become degenerate since one eigenvalue vanishes. We simply add a small perturbation ε over the diagonal of the variance-covariance matrix $\Gamma = \Gamma + \varepsilon I_d$. A rule of thumb that we carried out was to select ε such that $\varepsilon = \Delta x_i/100$ where Δx_i is the average step between successive discrete values of x_i. In the case when x_i is the laminate thickness for instance, $\Delta x_i = t_{ply}$ the elementary ply thickness.

7.1.2 Early results

We applied this modified IMAGE technique to build a surrogate model simply over lamination parameters coming from feasible stacking sequence and we also evaluate this surrogate over a test basis made of feasible stacking sequences.

In the first case, we used the Rayleigh-Ritz computations of the skin buckling of a laminate panel where only out-of-plane stiffness tensors appear. In that case, the dimension of regression was 4 : $(h, \xi_1^D, \xi_2^D, \xi_3^D)$ note that only h is discrete in that regression. The learning basis was made of $3,200$ points coming from feasible stacking sequences of number of plies ranging from 8 to 32.



(a) EM Clustering

(b) Error distribution

(c) Projection

(d) Predicted value vs. Real value

Figure 7.1: Results

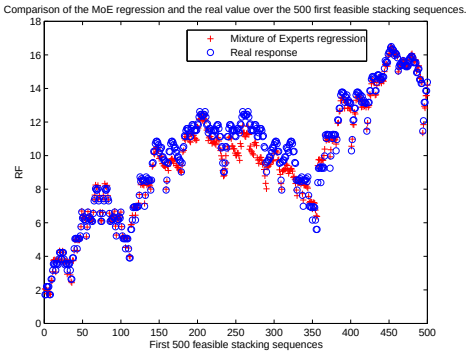
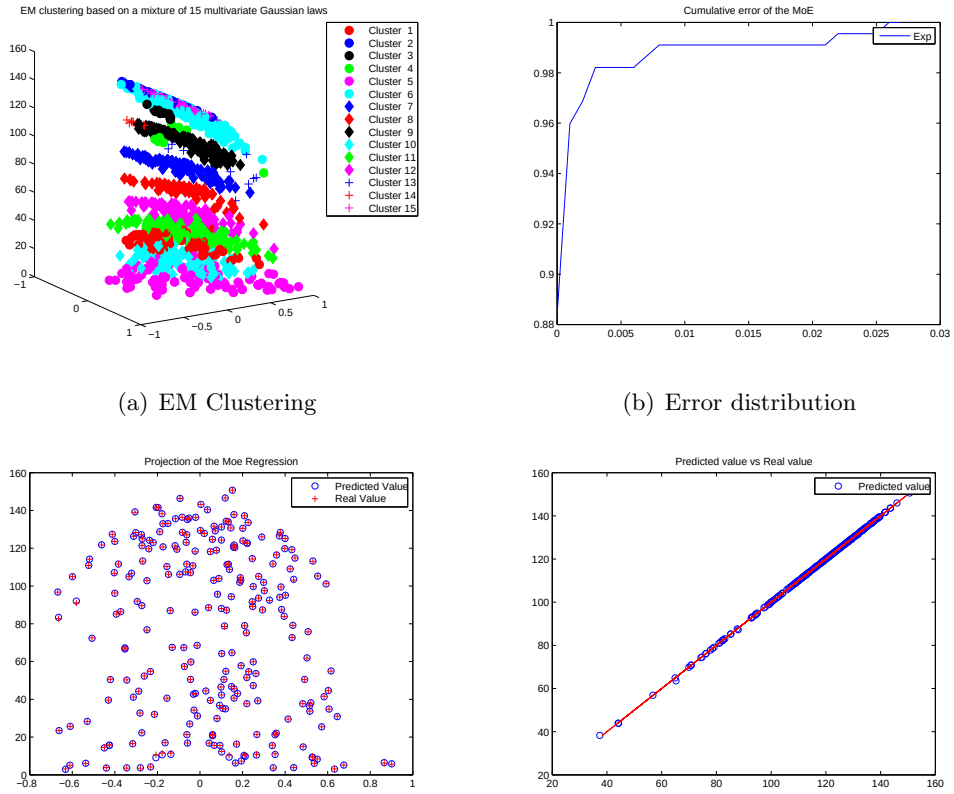


Figure 7.2: Comparison of the modified IMAGE Mixture of Experts predicted values and the real values for the first 500 feasible stacking sequences

In the second case, we used the same simple Rayleigh-Ritz computation of the skin buckling but for a stringer reinforced panel. Dimensions and stacking sequence of the stringer were fixed, and we considered internal load redistribution to make in-plane stiffness tensor appear in the expression. In that case, the dimension of regression was $6 : (h, \xi_1^A, \xi_2^A, \xi_1^D, \xi_2^D, \xi_3^D)$ note that in that case h, ξ_1^A, ξ_2^A are discrete in that regression. The learning basis was made of $5,400$ points coming from feasible stacking sequences of number of plies ranging from 8 to 32.



(a) EM Clustering

(b) Error distribution

(c) Projection

(d) Predicted value vs. Real value

Figure 7.3: Results

7.1.3 Airbus stability tool: COFUS

COFUS is a former stability in-house code at Airbus. It was aimed at computing stability critical reserve factor for curved stiffened panels. More precisely, it analyses

- Maximum deformation
- Buckling of skins, stringer, local web of stringer
- Global buckling of the whole cross-section
- Post-buckling
- Damage Tolerance

Different types of buckling are studied including skin buckling, stringer buckling (crippling), Euler buckling of the assembly (or global buckling), general buckling of the assembly of several super-stringers. However for MAAXIMUS test case we decided to froze stringer geometric dimensions and stringer web and cap stacking sequences (these two stacking can be different). This was aimed at reducing the complexity of the function. However, these fixed variables were based on previous optimizations were all variables were free with fixed internal loads. However it is worth noting that in our case even though the stringer variables were fixed, the local stringer varies still varies, since the thickness and stacking of skins and forces vary, changing the redistribution of forces at super-stringer (stiffening ratio hypothesis).

7.2 Practical implementation

A whole QSD implementation of MAAXIMUS panel implied

- An upper level optimizer that needs to be linked to a FEA (and sensitivity of FEA) tool. We took standard Boss Quattro optimization engine (`GCM` algorithm was chosen *Globally Convergent Method*)
- A FEA and sensitivity FEA tool. We chose `MSC.Nastran` and built a SOL200 (Optimization and sensitivity analysis solution of Nastran) session for the MAAXIMUS panel. We used classical Nastran cards `MAT2` to connect lamination parameters defined as `DESVAR`. Indeed, we created a link between G the normalized A tensor ($G = \frac{A}{h}$) defined over lp's (or equivalently proportions).
- A lower level optimization that could give the post-optimal sensitivities of the local optimizations. In terms of IT architecture, this means that the local optimization should be seen by the upper level as an external task; However at the

time of this implementation. Boss Quattro could not handle this since an optimization was different from an external task. Which means we had to implement the local optimization as a real external task that could not be done with Boss Quattro. For efficiency reasons, we chose `fmincon` optimizer of Matlab

The general framework is given in the following slides. However, it is worth noting that the trickiest part was not in the definition of tasks separately where a good knowledge of each tool was sufficient to handle each task (FEM, optimization,...) but **in the connection between different tasks**. Indeed, we had to tackle the following issues

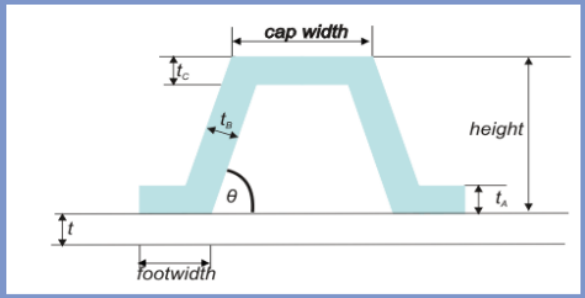
- SOL200 driver: to link the SOL200 FEA and sensitivity analysis, we had to write our own fit-to-purpose driver to Boss Quattro. One of the issue was about collecting the sensitivities terms and to write them in an appropriate format for Boss Quattro. This was done through a Shell script. However, it should be noted, there does exist a SOL200 driver in Boss Quattro but at the time when this implementation, the documentation was not clear about the different limitations (in terms of number of load cases for instance).

- Local optimizations external tasks: as explained before, we had to use the external task capability of Boss Quattro to link the different local optimizations. This was done `USERtask` files. We also had to write the sensitivities of the optimizations in the correct format for Boss Quattro.

- Chain ruling: despite the relative complexity of that test case, chain ruling sensitivity computation $\frac{d\mu_i^\star}{dY_j}$ were more simple to do that in the ten bar truss test-case. Indeed, we used the `Chain ruling` option from the optimization engine of Boss Quattro and this was automatically done.

WP 2.4 use case : Presentation

With local variables and constraints



- **Variables** :

$1.625\ mm \leq t_{skin} = 2\ mm \leq 10\ mm$

$1.625\ mm \leq t_a = 2\ mm \leq 4\ mm$

$0\ mm \leq \Delta t_c = 0\ mm \leq 4\ mm \qquad (t_c = t_a + \Delta t_c)$

Foot Width = 27mm

$15\ mm \leq$ Stiffener Height $= 25\ mm \leq 30\ mm$

Cap Width = 25 mm

$45° \leq$ Theta $\leq 63°$

$0.08 \leq$ Skin percentage $0° = 0.25 \leq 0.6$

$0.08 \leq$ Skin percentage $90° = 0.25 \leq 0.4$

$0.08 \leq$ Stringer web percentage $0° = 0.6 \leq 0.6$

Stringer web percentage $0° \leq$ Stringer cap percentage $0° \leq 0.6$

$0.08 \leq$ Stringer web/cap percentage $90° = 0.1 \leq 0.4$

- **Constraints :**

$RF_{Buckling} > 1$

$RF_{PostBuckling}$ (specific margin policy)

$RF_{StringerBuckling} > 1$

$0.4 < stiff\_ratio < 0.6$

$-3000\ \mu D < (\varepsilon_0, \varepsilon_{45}, \varepsilon_{135}, \varepsilon_{90})_{skin} < 5000\ \mu D$

$-2000\ \mu D < \varepsilon_{stringer} < 5000\ \mu D$

MAAXIMUS

7

Figure 7.4: Original use case description

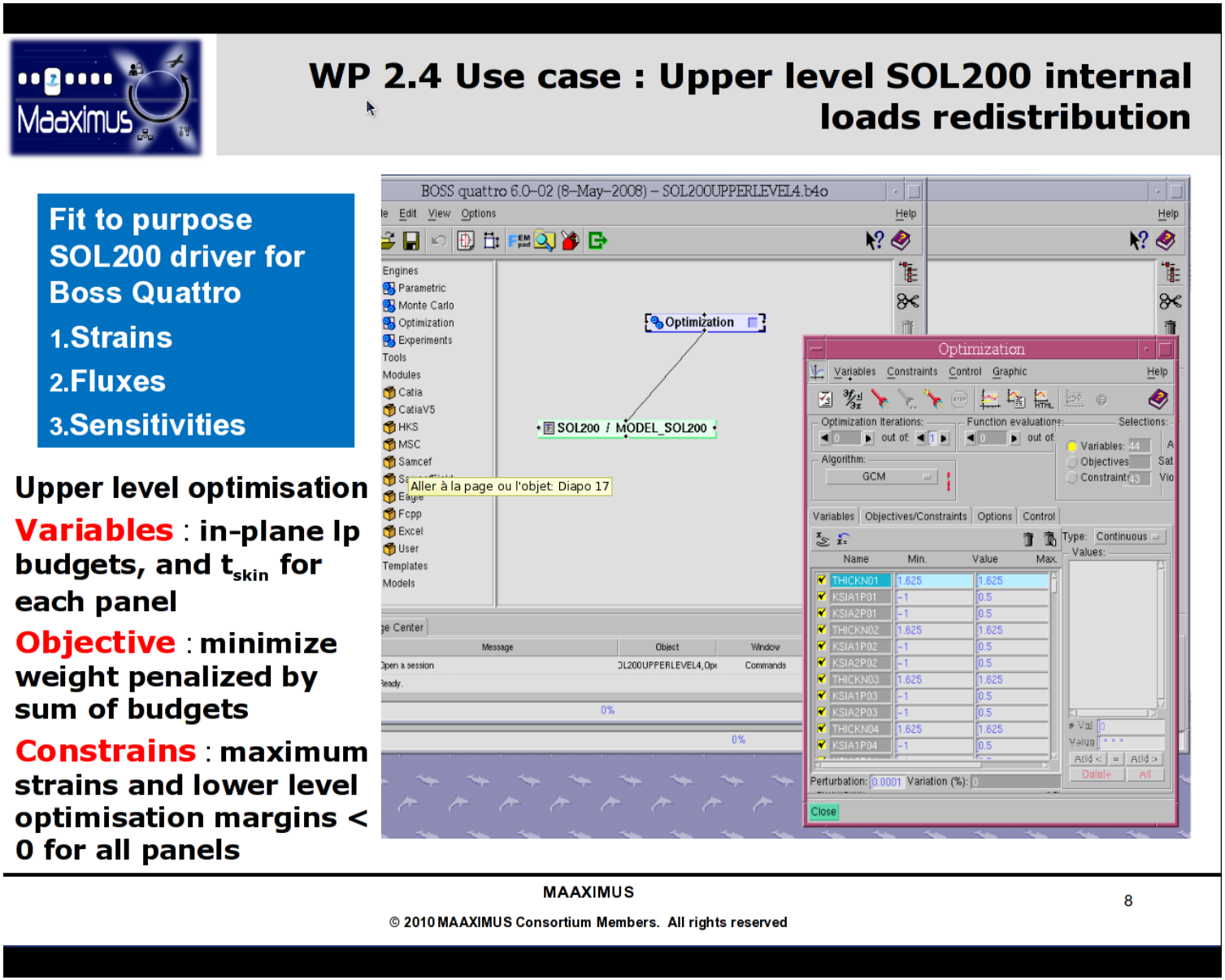


Figure 7.5: Upper level description

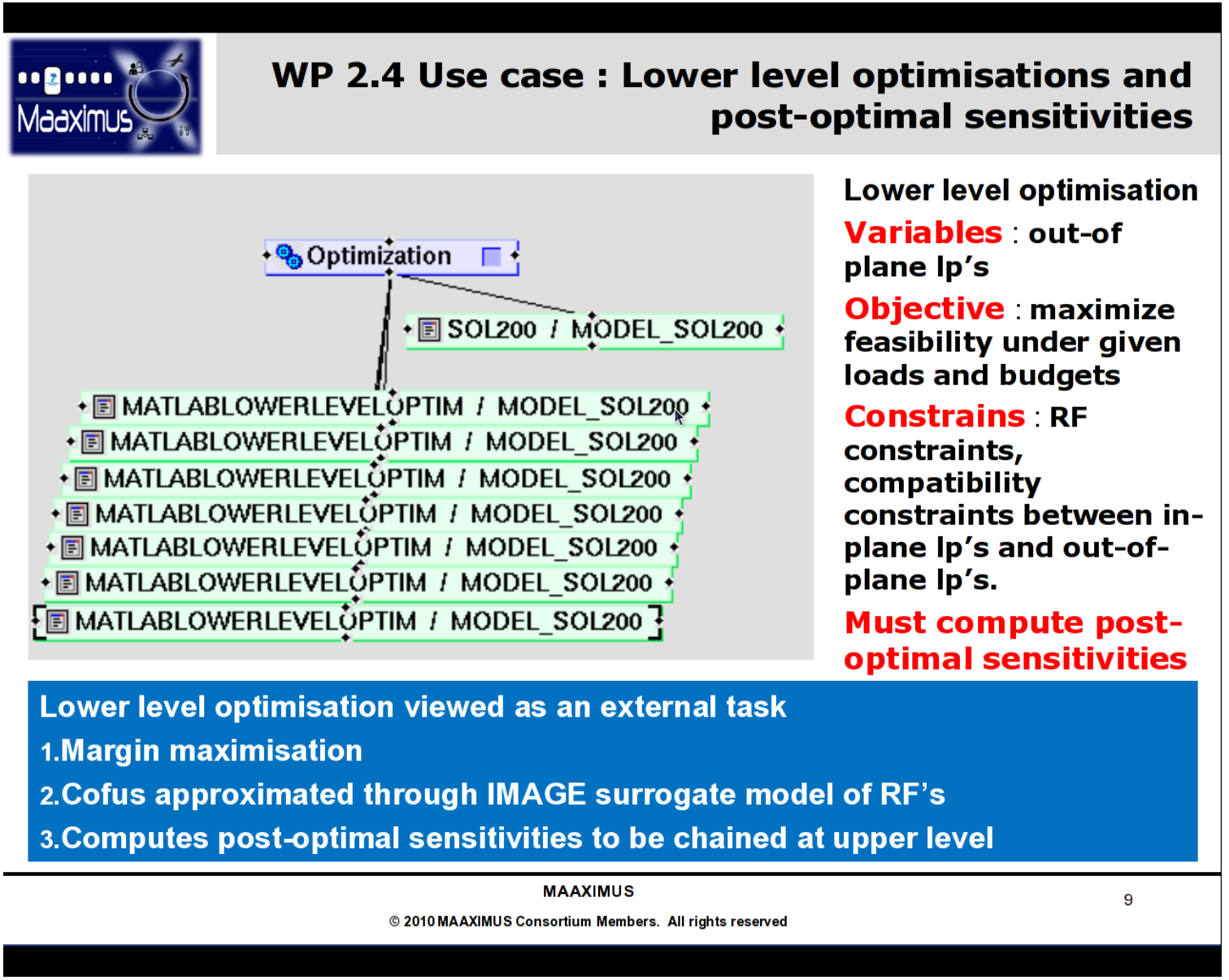


Figure 7.6: Lower level description

7.3 Post-identification

Once the upper level optimization is done, we have continuous variables for thicknesses and lamination parameters. We need to go back to discrete stacking sequence. To do we simply minimize the Euclidean distance between feasible stacking sequences and their continuous values

$$\min_{SS \in \text{set of feasible SS}} |h(SS) - h^\star| \sum_{i=1} |\xi_i(SS) - \xi_i\star| \tag{7.1}$$

However, to get consistent internal load redistribution, we did this post-identification in five steps

1. First **round off thicknesses and in-plane lamination parameters** (or equivalently proportions) to get all the discrete upper level Y variables. Based on these stiffness variables values, we reran static analysis to get the correct internal loads redistribution

2. Based, on these stiffness variables values and internal loads, we reran local optimizations to get the optimal out-of-plane lamination parameters (still continuous)

3. Identify amongst the stacking of given thickness and proportions the one(s) that give the closest out-of-plane lamination parameters.

4. Check whether the panel is still feasible with respect to buckling constraints computed with real COFUS code

the obvious drawbacks of that approach is that it might give a good set of stacking that minimizes the weight, however **there is no guarantee** that

- it gives the optimal one w.r.t to stability and strains constraints
- it gives blended stacking sequences

7.4 Detailed results on MAAXIMUS panel

Detailed results are given in the following pages. The main important result is that we found **the same optimum than AIO**. We found out an optimal weight of 26.9 kg with the QSD at the 4^{th} iteration and the same AIO session gave an optimal weight of

26.9 kg at the 7^{th} iteration. Since we used `GCM` algorithm, the number of finite elements calls is exactly the same as the number of iterations.

QSD Results on MAAXIMUS panel

This QSD implementation was ran on the Maaximus panel with initial loading

1,625	1,625	1,625	1,625	1,625	1,625	1,85005724	2,45467129
1,625	1,625	1,625	1,625	1,625	1,625	1,625	2,46125275
1,625	1,625	1,625	1,625	1,625	1,625	1,625	1,94380887
1,625	1,625	1,625	1,625	1,625	1,625	1,625	1,625
1,625	1,625	1,625	1,625	1,625	1,625	1,625	1,98846182
1,625	1,625	1,625	1,625	1,625	1,625	1,625	1,625

Weight at convergence was 26.9 kg (over continous variables). Thickness were rounded

13	13	13	13	13	13	15	19
13	13	13	13	13	13	13	19
13	13	13	13	13	13	13	16
13	13	13	13	13	13	13	13
13	13	13	13	13	13	13	16
13	13	13	13	13	13	13	13

Proportions were also rounded (in terms of in-plane lamination parameters) to get discrete proportion

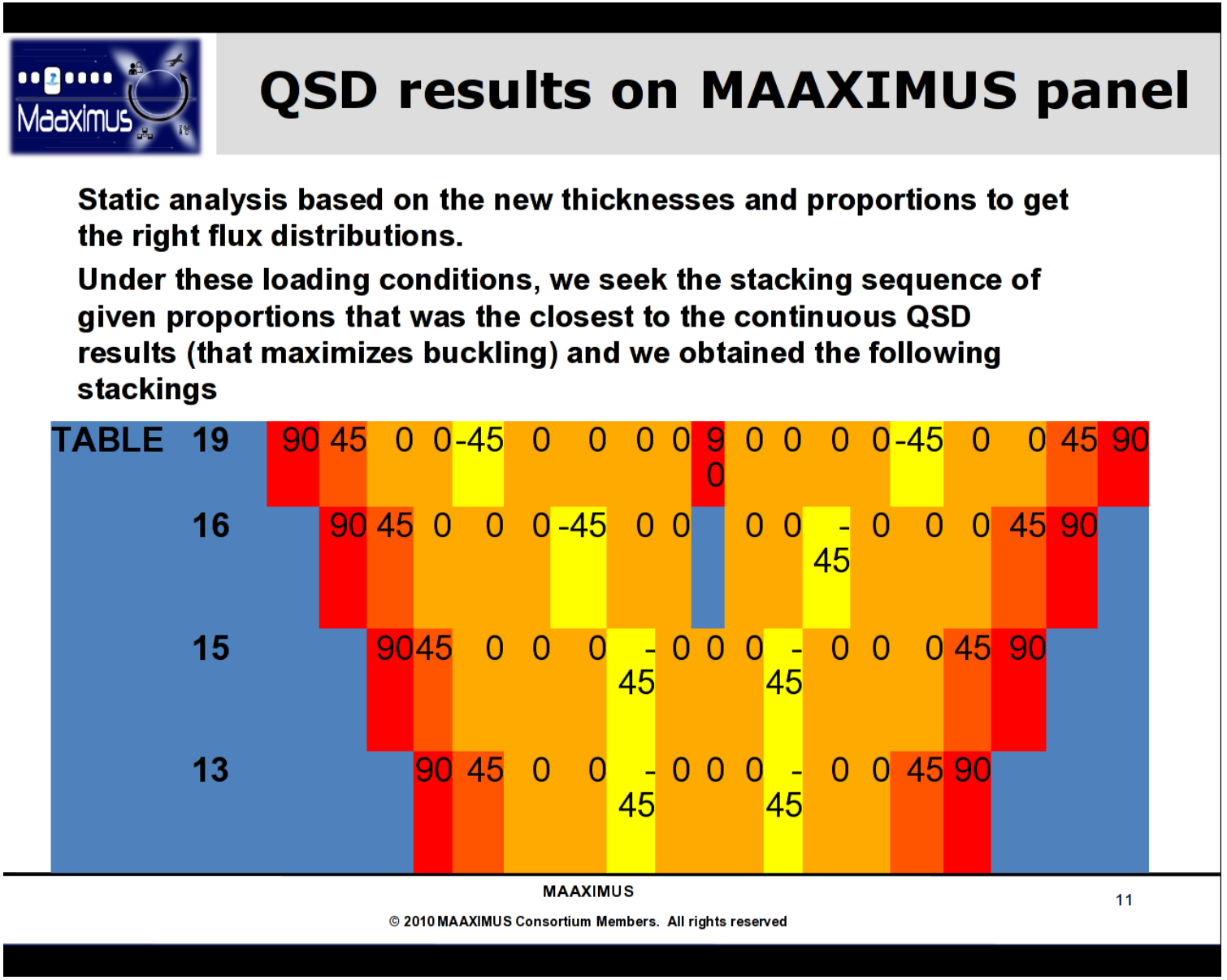


Figure 7.7: Lower level description

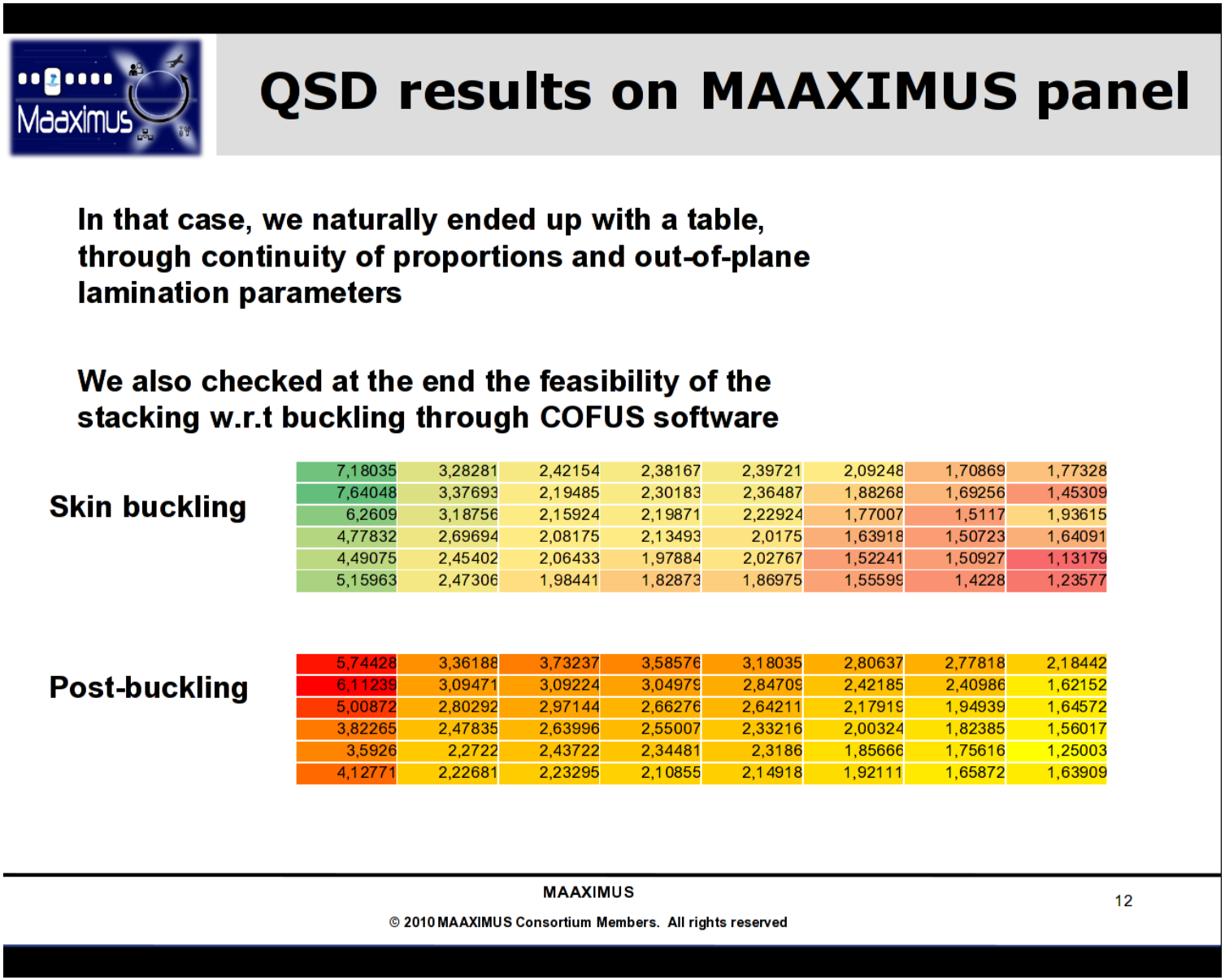


Figure 7.8: Lower level description

7.5 Conclusion and ways forward

We saw that we could derive a bilevel optimization scheme that gives the optimal weight for the simple test-case. As we saw it in the former chapter, as opposed to other decomposition/bilevel schemes that were implemented as alternating optimizations, this QSD scheme was found to be give the optimal weight for all loading cases, for most starting points with less FEA calls and less global cycles iterations than a monolevel scheme (AIO).

Based on the specific composite considerations, we were able to derive this scheme for more realistic structures including the MAAXIMUS panel test case. It has the following features

- **Allows to decompose the global problem into several local optimizations that can be solved in parallel.**
- **The composite representation adopted allows to integrate surrogate models of buckling constraints.**
- **It gives a lower bound of the weight, the optimal weight found by applying QSD is indeed the best conceptual design (continuous variables)**
- **However this weight will always be less than the one found by rounding off the thicknesses and proportions.**
- **There is no guarantee that the rounding-off process would give blended stacking sequences. However, we saw that in the MAAXIMUS test case, we saw that it naturally** gives a table.

Ways forward

- **Apply this decomposition scheme to more complex structures: barrel**
- **One improvement would be to allow the lower level to give directly stacking sequences instead of continuous lp's, through the use for instance of a continuous optimization that converge to discrete values (applying for instance ideas of topology optimization over angles densities**)

- **Find a way to enforce the continuity constraints or find a heuristic that allows to find an optimal set of blended stacking sequences based on the continuous variables values.**

Conclusion & Perspectives

We have presented a formal problem that naturally arises in the structural optimization for large-scale composites structures of aircraft. The analysis of the problem shown that decomposition and more specifically **bilevel optimization schemes could help treat this kind of problem**. The natural decomposition of such problems was mostly based on **the physics of the problems both from the natural decomposition of the global structure into sub-structural elements and also from the different levels of analysis of the mechanical behavior** used in the aerospace industry and more especially at Airbus. The **main innovations lied in first the type of decomposition and then bilevel scheme to use, then in the adaptation of the composite representation needed and also on the approximation of buckling constraints that features discontinuities**.

The analysis of the problem lied essentially in the three first chapters. We started with the formal description of the optimization problem and enforced the needs for a decomposition scheme and then a bilevel optimization formulation that matches the process of sizing used at Airbus. One important feature of the bilevel scheme we pointed out was that it should feature membrane mechanical behavior of the whole structure only at the upper level of optimization. This way the analysis of the whole structure, performed by the Finite Element Method is only performed at global level of optimization. The second level should feature only stability analyses. After presenting formally the problem, we then emphasized the need for fast stability analysis to perform local optimizations and then enforced the need for surrogate models of buckling constraints.

In the second chapter, basics of differentiable optimization are recalled to highlight two important facts. First we presented a quite rigorous framework to differentiate implicit functions defined through local optimizations. This was referred to as the

post-optimal sensitivity. After presenting these classical results, we gave an **original application of these results that allows us to differentiate eigenvalue problem** (and buckling is typically such as problem). If the formula for eigenvalue sensitivity were not new the way we obtain them was original. This chapter also illustrated the equivalence of the monolevel formulations, from the original problem, denoted as AIO, we introduce the classical block-angular formulation by using stiffness variables as optimization variables. The real coupling between sub-elements was treated as equality constraints. These equality constraints were then relaxed by inexact quadratic penalty, leading to three different instances of the original problems. The different multilevel formulations were then introduced in this framework as natural decomposition of each of this instance.

We turned to real composite representation in the third chapter. We introduced the practical composite representation that we adopted in this work, namely the **lamination parameters and investigate the dependence of the buckling with respect to these variables**. Besides from the buckling dependence over lamination parameters, this representation appeared to be **well adapted to our bilevel needs since it offers a natural decomposition between in-plane behavior (that feeds the FEA at upper level) and out-of-plane behavior (that feeds buckling analysis of lower level)**. Two important aspects are then shown, first buckling critical load factor (or Reserve Factor) is shown to be **concave over bending lamination parameters**. Then the very nature of the buckling constraints over lamination parameters and also over loadings was investigated. The buckling critical load was shown to **behave in a piecewise continuous manner**. An appropriate approximation strategy (or surrogate models construction) should take this into account.

On the basis of the observed behavior of buckling constraints, that is piecewise differentiable, an original and innovative strategy to take this feature into account was presented in Chapter 4. This strategy allows to **divide the input space into regions** where the functions are expected to behave more regularly. This strategy uses tools from **unsupervised statistical learning from clustering to probability density estimation**. Our strategy does rely on first density estimation performed through EM algorithm and then to clustering and local approximation models constructions. These

local approximation models are called experts and the whole strategy is an instance of mixture of experts. This strategy led to **an algorithm called IMAGE** that we tested and validated over the piecewise differentiable examples that are buckling constraints. This comparison shows a very good accuracy of the approximation when compared to a classical global surrogate model. This **strategy seems then appropriate to be used within an optimization process to approximate buckling constraints**.

The fifth chapter illustrated the monolevel instances of our optimization problem. We first presented a classical test case that gathers most of the difficulties of our initial problem: the 10 bar truss. However we can still handle this problem for it has a reasonable size. Analytical sensitivities computations were recalled since they are a valuable tool in structural optimization and they were also **chained with post-optimal sensitivities**. **The different monolevel schemes were presented in the realm of our test case**. Numerical experiments shown that the **block-angular transform algorithms, denoted in this chapter SAND2 and SAND3 because of their similarity to the SAND formulation, performed quite well** and were a numerical evidence for introducing stiffness variables as optimization variables.

The sixth chapter decomposed the monolevel formulation introduced in the former chapters. Different classical schemes are tested over the 10 bar truss test case. We gave insight intro differences of implementations, indeed bilevel formulation may be solved either as alternating optimizations (sequences of successive optimizations) or as a real **bilevel optimization scheme**. This resulted in a more sophisticated implementation which was presented in detail. On the basis of theoretical properties, we chose to implement the **Quasi separable subsystem Decomposition as a real bilevel optimization**. We adapted the QSD scheme to allow equality constraints through inexact quadratic penalty. In particular, post-optimal sensitivities were required to get to a tractable upper level optimization. Results on the 10 bar truss shown first that **the optimum was the same as the AIO and the number of global iterations and finite elements calls was slightly less than AIO**.

Finally in the seventh chapter most of the material presented in former chapters is used **to solve a real aircraft structure optimization problem**, namely a panel

made of 6×8 composite super-stiffeners. The **QSD scheme was adapted for lamination parameters at both level**, the local buckling constraints computed from an Airbus stability tool were **approximated through the original approximation method presented in the fourth chapter**. These approximations were used at the local optimization level and post-optimal sensitivities were retrieved from the local optimums and chained at upper level with internal loads sensitivities computed through a FEA solver. The implementation together with results were given, **the same optimum was found than the AIO method. Real stacking sequences were then derived from the continuous optimums by simple post-identification**.

In definitive, we answered the original problem and find ways to improve the process of sizing of large-scale composite structures in the aerospace industry. This work could be carried on through the investigation of both physics (buckling) and theoretical abstract properties arising from optimization and probability. This investigation was indeed necessary first to understand the quite complicated problematic and to propose new innovative strategies. This thesis has also been sprinkled with fruitful collaborations in many different areas such as comparisons of the IMAGE strategy with other innovative methods (EADS-IW, EADS-Russia, Onera), implementation of the IMAGE strategy in the Boss Quattro software (Samtech), composite laminate material (Airbus-UK), comparison of the bilevel implementations (NLR)...and many others things that are listed in the next part where we also mention publications.

Perspectives

We derived and adapted an existing bilevel optimization scheme to solve our initial problem. This scheme in conjunction with the new approximation method that we develop allowed us to get first **promising results for a real aircraft structure**. However this scheme together with the approximation method need to tested and validated on more complicated aircraft structures. This will be a first step to achieve, another one will be to compare such a bilevel optimization not only in terms of optimization iterations and finite elements calls but also on the basis of computer performance for a

real parallel session, which was not implemented in this work. Indeed, the main objective of decomposition/bilevel optimizations schemes is to make the resolution scalable and each global iteration can be largely reduced by parallel computations of both constraints and sensitivities. The optimal parallel architecture should have as many as processors as sub-elements to reach a good performance.

Regarding theoretical properties of the QSD scheme, we already mentioned some of them (as in (Haftka & Watson 2005)), one could ask whether or not we could derive local convergence properties as for instance in (DeMiguel & Murray 2006). This would be a very important feature, however such a convergence is not easy to reach since one might encounter theoretical pitfalls as the optimal value function not always differentiable. Another way of improvement would be to show that under reasonable assumption the upper level problem over lamination parameters is convex. This work is a first step towards this property since we investigated the local level optimization (i.e maximization of the buckling critical load). It is worth noting that even though we obtain some properties of convexity in simple cases, the real set of constraints for instance used in the panel test case (post-buckling) damage tolerance...) would also need investigation, in particular for their behavior over lamination parameters and forces.

As far as global approximations of discontinuous or derivative-discontinuous functions are needed, the original strategy presented in this work could be applied to other fields of applications: basically any numerical computations that include a critical mode over many of them (vibrations, structural dynamics...) or any discontinuity in the response. In very different context, one possible application would be the approximation of CFD computations for different parameter (Mach number...) in such a case the response might be defined by regions. Regarding the very algorithms, many improvements and research could be achieved. First from a theoretical point of view, we have not found out what kind of discontinuities are very likely to be detected by the clustering part. The same way, suppose that the clustering does detect the real pieces of the functions, this is highly dependent on the sampling, would we observe convergence of the pieces to real pieces of the functions? In such a case, if the function to be approximated is smooth enough over each piece, one could use RBF or MLS approximations locally, would we get at the end a spectral convergence for piecewise analytic functions for

instance? No specific work was done in this thesis to enhance local experts, this would be a first step to achieve to improve the accuracy of the approximation.

The whole optimization problem and the bilevel scheme need to be modified to incorporate realistic continuity constraints, this is the difficult problem of blending, the plies should satisfy continuity all over the structure or at least large parts of the structure. This complicates much more the optimization problem for there is no easy way to insert inter-elements constraints, as already noted in (Liu *et al.* 2004) and (Liu 2001). In-plane lamination parameters would possibly be a coarse and rough way to ensure at least global continuity of the proportions at upper level optimization, such constraints would easily be integrated. However detailed blending stacking sequences is still a completely discrete and combinatorial problem, even though compatibility equations between out-of-plane and in-plane could help identify blended stacking sequences on the basis of their proportions.

Most of the innovation derived in this thesis can be applied to many different problems arising in engineering optimization. The frontier in between MDO and multilevel formulations is quite fuzzy and most of the material we presented and improve in this work can be applied to MDO problems.

Collaborations & publications

Collaborations

As already mentioned, most of this work was related to the European reserach project MAAXIMUS and more specifically the Work Package leaded by Airbus-France and Onera. This research was mainly conducted under both supervision of Airbus-France and Onera. Fruitful collaborations were initiated from first MAAXIMUS projects but also from EADS and Onera internal projects, we give here a list of the most significant collaborations of this thesis

- In the framework of MAAXIMUS Work Package 2.4 (led by Airbus-France), decision was taken **to compare the results of our bilevel approach (QSD) with the results from NLR (J. Vankan) who implemented a BLISS-like scheme**. The test case was the 10 bar truss that was formally specified in the frame of this thesis. Resulting in a quite intensive comparison whose results can be found in the NLR deliverable of WP 2.4. Many parameters where evaluated including the number of total FEA's, number of global iterations,... Results can be briefly summarized
  - In average, **QSD took slightly less FEA calls and global iterations to converge** than the BLISS like scheme of NLR.
  - In terms of local evaluation, **QSD called many more times the local analyses** (buckling) than the NLR's scheme since NLR has implemented an adaptive strategy for approximating local analyses. Our QSD scheme did not feature surrogate models, hence **comparison can hardly be done on the same basis**.
- In the framework of MAAXIMUS Work Package 2.3 (led by Onera), the IMAGE software was developped and exchanged through partners. In particular, decision

was taken **to integrate in the optimization software Boss Quattro (B. Colson) of Samtech the IMAGE algorithm**. This resulted in a collaboration with Samtech and an executable of the EM clustering was integrated on top of the surrogate models capability of Boss Quattro (we represented this integration Fig 7.9)

- In the EADS framework decision was taken to evaluate the capability of IMAGE as a prediction tool. Several comparisons over many test cases ranging from simple analytic test cases to the most recent Airbus stability tools were ran by EADS-IW. In particular, surrogate models of the current Airbus stability tool were constructed using IMAGE and MACROS solution for surrogate modeling and optimization, developed by DATADVANCE company, and accuracies of these surrogate models were compared. This was a challenging test case for the regression was in dimension 20, leading to very large databases (more than $200,000$ points). It turned out that MACROS results are more accurate (the error of MACROS is two times lower) but IMAGE results are also satisfactory."

- In the Onera framework, **the IMAGE algorithm was also tested over buckling critical factors approximation obtained through Detailed Finite Elements Methods (non linear FEM with Samcef code)**. These buckling computations came from the Department of Mechanics of Composite Structures (DMSC). In the framework of the Airbus/Onera project Artemis, **IMAGE was applied to aerodynamic numerical simulation results (*Applied Aerodynamics Department* DAAP)** because of the apparent piecewise behavior of the aerodynamic response to approximate (C_D drag coefficient) with respect to flight cases parameters.

- In the Airbus framework, **IMAGE was applied in an operational context to approximate some stability computations for the A-350(-1000)**. This was also the occasion to collaborate with Airbus-Spain.

- Finally, **IMAGE was applied to approximate modal displacements in structural dynamics with the Department of Structural Mechanics of ISAE-Sup'Aéro (J.Morlier)**. Recent collaboration also led to other approximation strategies including compressed sensing.

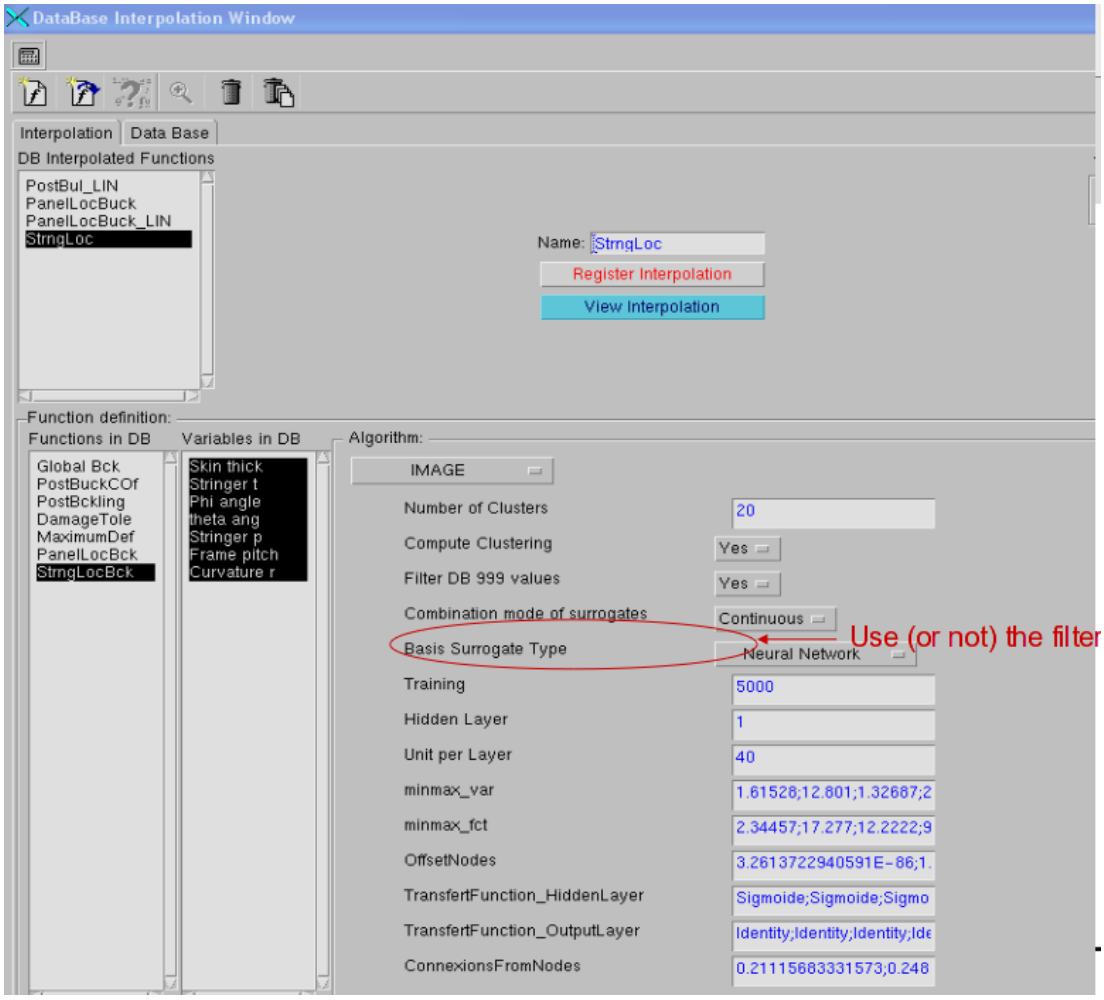


Figure 7.9: Integration of IMAGE into Boss Quattro

- This has also been the occasion for the author to **supervise four trainees at Airbus and ISAE-Sup'Aéro**.

Publications

Book chapter

- **On ensemble learning through mixtures probability densities estimation : from EM learning to memory based methods**, D. Bettebghor, M. Samuelides, to appear in *Recent advances in dynamics and control of neural net-*

works, Cambridge Scientific Publishers, UK.

Articles

- **Surrogate modelling approximation using a mixture of experts based on EM joint estimation**, D. Bettebghor, N. Bartoli, S. Grihon, J. Morlier, M. Samuelides, *Structural and Multidisciplinary Optimization*, 43, 2, 243–259, Springer, 2011.

- **Approximation of buckling critical factor for composite panels**, D. Bettebghor, N. Bartoli, *Structural and Multidisciplinary Optimization*, 46, 4, 561-584, Springer, 2011

- **Large-scale optimization of composites structures through Quasi Separable Decomposition bilevel scheme**, D. Bettebghor, S. Grihon, to be submitted

Research projects deliverables and technical reports

- **Utilisation de IMAGE et mise en place d'un processus itératif** D. Bettebghor, N. Bartoli, *Onera Technical Report*, 2012.

- **Embed surrogate models for multilevel optimization for composite fuselage barre**l, D. Bettebghor, N. Bartoli, M. Samuelides, *Onera MAAXIMUS Deliverable WP 2.4.9*, 2011.

- **Specification of IMAGE (Improving Metamodelling Approximation through Gaussian Experts) : application to the approximation of buckling analysis**, D. Bettebghor, **Airbus Technical Report**, 2010.

- **Review of metamodeling strategies for structural analysis and optimization : application to composite stiffened panels optimization**, D. Bettebghor, N. Bartoli, M. Samuelides, *Onera MAAXIMUS Deliverable WP 2.3.9*, 2010.

- **Review of global and multilevel strategies for large-scale structural optimization : towards composite fuselage optimization**, D. Bettebghor, *Airbus Technical Report*, 2009.

International peer-reviewed conferences

- **Comparison of methods for approximation of stability constraints for composite structures**, S. Alestra, D. Bettebghor, E. Burnaev, S. Grihon, P. Prikhodko, DYNACOMP 2012 *1st International Conference on Composite Structure Dynamics*, May 2012, Arcachon, France.

- **Compressed sensing applied to modal analysis and modeshapes reconstruction**, J. Morlier, D. Bettebghor, IMAC XXX *A Conference and Exposition on Structural Dynamics, Society for Experimental Mechanics*, January 2012, Jacksonville, U.S.A.

- **Surrogate-based bilevel optimization for laminated composite stiffened panels**, D. Bettebghor, S. Grihon, 2nd Int. Conf. Engineering Optimization, Lisbonne, 6-9 Sept. 2010.

- **Using Mixture of Experts Surrogate Models and EM learning for multilevel structural optimization**, D. Bettebghor, N. Bartoli, S. Grihon, A. Merval, J. Morlier, M. Samuelides, *ECCM 2010 IV European Conference on Computational Mechanics*, Paris, 16-21 May 2010.

- **Numerical optimisation of advanced composite structures**, D. Bettebghor, S. Grihon, *ECCM 2010 IV European Conference on Computational Mechanics*, Paris, 16-21 May 2010

French national peer-reviewed conferences

- **Enhancing approximation of critical buckling factor for laminated composites**, D. Bettebghor, M. Samuelides, N. Bartoli, *12th Onera-DLR Aerospace Symposium*, June 2012, Braunschweig, Germany

- **Optimisation biniveau de structures aéronautiques composites**, D. Bettebghor *Journée Scientifique de l'Onera - Optimisation multidisciplinaire*, Janvier 2012, Palaiseau

- **Applications des mélanges d'experts á l'approximation de calculs de stabilités pour les composites**, D. Bettebghor, N. Bartoli, *Journée Scientifique de l'Onera - Modèles de substitution*, Janvier 2012, Palaiseau

- **Approche en paramètres de stratification pour l'optimisation biniveau de structures composites**, D. Bettebghor, S. Grihon, N. Bartoli, J. Morlier, *10 ème colloque national en calcul de structures*, Mai 2011, Giens
- **Optimisation multiniveau de structures aéronautiques**, D. Bettebghor, M. Samuelides, S. Grihon, J. Morlier, A. Merval, *9 ème colloque national en calcul de structures*, Mai 2009, Giens
- **Conception optimale : outils d'aide au dimensionnement et au diagnostic de structures aéronautiques**, J. Morlier, D. Bettebghor, *Journee 3AF : Conception robuste des structures par la simulation*, Airbus France, Novembre 2009, Toulouse.

References

[Abu-Odeh & Jones 1998] A.Y. Abu-Odeh and H.L. Jones. *Optimum design of composite plates using response surface method.* Composite structures, vol. 43, no. 3, pages 233–242, 1998. 95

[Acar & Rais-Rohani 2009] E. Acar and M. Rais-Rohani. *Ensemble of metamodels with optimized weight factors.* Structural and Multidisciplinary Optimization, vol. 37, no. 3, pages 279–294, 2009. 131

[Akulenko & Nesterov 2005] L.D. Akulenko and S.V. Nesterov. High precision methods in eigenvalue problems and their applications. CRC Press, 2005. 113

[Alexandrov & Lewis 1999] N.M. Alexandrov and R.M. Lewis. *Comparative properties of collaborative optimization and other approaches to MDO.* In Proceedings of the first ASMO UK/ISSMO conference on engineering design optimization, pages 8–9. Citeseer, 1999. 89

[Alexandrov *et al.* 2000] N.M. Alexandrov, R.M. Lewis*et al. Analytical and computational properties of distributed approaches to MDO.* 2000. 89

[Allaire 2007] G. Allaire. Conception optimale de structures, volume 58. Springer Verlag, 2007. 64

[Antman & Renardy 2005] S.S. Antman and M. Renardy. Nonlinear problems of elasticity. Springer, 2005. 110

[Antman 2006] S.S. Antman. *Theodore von Kármán.* A Panorama of Hungarian Mathematics in the Twentieth Century I, pages 373–382, 2006. 109

[Armacost & Fiacco 1974] R.L. Armacost and A.V. Fiacco. *Computational experience in sensitivity analysis for nonlinear programming.* Mathematical Programming, vol. 6, no. 1, pages 301–326, 1974. 71

[Ashbaugh 1999] M.S. Ashbaugh. *Isoperimetric and universal inequalities for eigenvalues.* In Spectral theory and geometry: ICMS Instructional Conference, Edinburgh 1998, page 95. Cambridge Univ Pr, 1999. 112

[Ashton & Whitney 1970] JE Ashton and J.M. Whitney. Theory of laminated plates, volume 4. CRC Press, 1970. 100

[Autio 2001] M. Autio. *Optimization of coupled thermal-structural problems of laminated plates with lamination parameters.* Structural and Multidisciplinary Optimization, vol. 21, no. 1, pages 40–51, 2001. 96

[BARTHELEMY 1988] J.F. BARTHELEMY. *Improved multilevel optimization approach for the design of complex engineering systems.* AIAA journal, vol. 26, pages 353–360, 1988. 91

[Berge & Patterson 1963] C. Berge and E.M.W. Patterson. Topological spaces. Oliver & Boyd, 1963. 71

[Berkhin 2002] P. Berkhin. *Survey of clustering data mining techniques*, 2002. 133

[Bloom & Coffin 2000] F. Bloom and D. Coffin. Handbook of thin plate buckling and postbuckling. CRC Press, 2000. 110

[Bloomfield *et al.* 2009] MW Bloomfield, CG Diaconu and PM Weaver. *On feasible regions of lamination parameters for lay-up optimization of laminated composites.* Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science, vol. 465, no. 2104, page 1123, 2009. 95, 104, 105, 119

[Bradley *et al.* 1998] P.S. Bradley, U. Fayyad and C. Reina. *Scaling EM (expectation-maximization) clustering to large databases.* Microsoft Research Report, MSR-TR-98-35, 1998. 134

[Braun & Kroo 1997] RD Braun and IM Kroo. *Development and application of the collaborative optimization architecture in a multidisciplinary design environment*, 1997. 88

[Buhmann 2001] MD Buhmann. *Radial basis functions.* Acta numerica, vol. 9, pages 1–38, 2001. 130

[Burnham & Anderson 2004] K.P. Burnham and D.R. Anderson. *Multimodel inference: understanding AIC and BIC in model selection.* Sociological Methods & Research, vol. 33, no. 2, page 261, 2004. 133, 145

[Courant 1994] R. Courant. *Variational methods for the solution of problems of equilibrium and vibrations.* LECTURE NOTES IN PURE AND APPLIED MATHEMATICS, pages 1–1, 1994. 84, 113

[Daniel 1989] GAY Daniel. *Matériaux composites.* Hermès, Paris, 1989. 96, 100

[de Wit & van Keulen 2011] A. de Wit and F. van Keulen. *Framework for Multi-Level Optimization of Complex Systems.* Multiscale Methods in Computational Mechanics, pages 347–377, 2011. 61

[De Wit 2009] AJ De Wit. *A Unified Approach towards Decomposition and Coordination for Multi-level Optimization.* 2009. 61

[DeMiguel & Murray 2006] V. DeMiguel and W. Murray. *A local convergence analysis of bilevel decomposition algorithms.* Optimization and Engineering, vol. 7, no. 2, pages 99–133, 2006. 89, 249

[Dempe 2002] S. Dempe. Foundations of bilevel programming. Kluwer Academic Publishers, 2002. 76

[Dempster *et al.* 1977] AP Dempster, NM Laird and DB Rubin. *Maximum likelihood from incomplete data via the EM algorithm.* Journal of the Royal Statistical Society. Series B (Methodological), pages 1–38, 1977. 139

[Diaconu *et al.* 2002] C.G. Diaconu, M. Sato and H. Sekine. *Feasible region in general design space of lamination parameters for laminated composites.* AIAA journal, vol. 40, no. 3, pages 559–565, 2002. 95, 105

[Dreyfus 2005] G. Dreyfus. Neural networks: methodology and applications. Springer Verlag, 2005. 130

REFERENCES

[Fasshauer 2005] GE Fasshauer. *Dual bases and discrete reproducing kernels: a unified framework for RBF and MLS approximation.* Engineering Analysis with Boundary Elements, vol. 29, no. 4, pages 313–325, 2005. 150

[Fasshauer 2007] G.E. Fasshauer. Meshfree approximation methods with matlab, volume 6. World Scientific Pub Co Inc, 2007. 151

[Felippa 2000] CA Felippa. *ASEN 5367 course notes: advanced finite element methods.* Department of Aerospace Engineering, University of Colorado, Boulder, 2000. 39, 97

[Fiacco 1976] A.V. Fiacco. *Sensitivity analysis for nonlinear programming using penalty methods.* Mathematical programming, vol. 10, no. 1, pages 287–311, 1976. 71

[Fiacco 1983] A.V. Fiacco. Introduction to sensitivity and stability analysis in nonlinear programming, volume 165. Academic Pr, 1983. 70, 71, 76

[Fletcher 1981] R. Fletcher. *Practical Methods of Optimization: Vol. 2: Constrained Optimization.* JOHN WILEY & SONS, INC., ONE WILEY DR., SOMERSET, N. J. 08873, 1981, 224, 1981. 84, 85, 86

[Fleury 1989] C. Fleury. *CONLIN: an efficient dual optimizer based on convex approximation concepts.* Structural and Multidisciplinary Optimization, vol. 1, no. 2, pages 81–89, 1989. 40

[Forrester & Keane 2009] A.I.J. Forrester and A.J. Keane. *Recent advances in surrogate-based optimization.* Progress in Aerospace Sciences, vol. 45, no. 1-3, pages 50–79, 2009. 130

[Friedman *et al.* 2001] J. Friedman, T. Hastie and R. Tibshirani. *The elements of statistical learning*, 2001. 130, 132, 137, 161, 163

[Fukunaga *et al.* 1995] H. Fukunaga, H. Sekine, M. Sato and A. Iino. *Buckling design of symmetrically laminated plates using lamination parameters.* Computers & structures, vol. 57, no. 4, pages 643–649, 1995. 95

[Grenestedt 1989] J.L. Grenestedt. *A study on the effect of bending-twisting coupling on buckling strength.* Composite structures, vol. 12, no. 4, pages 271–290, 1989. 95

[Grenestedt 1990] JL Grenestedt. *Composite plate optimization only requires one parameter.* Structural and Multidisciplinary Optimization, vol. 2, no. 1, pages 29–37, 1990. 95

[Grenestedt 1991] JL Grenestedt. *Layup optimization against buckling of shear panels.* Structural and Multidisciplinary Optimization, vol. 3, no. 2, pages 115–120, 1991. 95, 116

[Gürdal *et al.* 1999] Z. Gürdal, R.T. Haftka and P. Hajela. Design and optimization of laminated composite materials. Wiley-Interscience, 1999. 41, 78, 93, 101

[Haftka & Gurdal 1992] R.T. Haftka and Z. Gurdal. Elements of structural optimization. Kluwer Academic Pub, 1992. 40, 54, 61, 74, 80, 198, 263

[Haftka & Watson 2005] R.T. Haftka and L.T. Watson. *Multidisciplinary design optimization with quasiseparable subsystems.* Optimization and Engineering, vol. 6, no. 1, pages 9–20, 2005. 91, 129, 249

[Haftka & Watson 2006] R.T. Haftka and L.T. Watson. *Decomposition theory for multidisciplinary design optimization problems with mixed integer quasiseparable subsystems.* Optimization and Engineering, vol. 7, no. 2, pages 135–149, 2006. 91

[Handbook 2007] A.I. Handbook. *FAA -H-8083-1A Aircraft weight and balance handbook.* Federal Aviation Administration, 2007. 33

[Harrison *et al.* 1995] P.N. Harrison, R. Le Riche and R.T. Haftka. *Design of stiffened composite panels by genetic algorithm and response surface approximations.* In AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, 36 th and AIAA/ASME Adaptive Structures Forum, New Orleans, LA, pages 58–68, 1995. 95

[Haykin 2008] S. Haykin. Neural networks: a comprehensive foundation. Prentice Hall, 2008. 150

[Henrot 2006] A. Henrot. Extremum problems for eigenvalues of elliptic operators: Antoine henrot. Birkhauser, 2006. 117

[Herencia *et al.* 2008a] J.E. Herencia, R.T. Haftka, P.M. Weaver and M.I. Friswell. *Lay-Up Optimization of Composite Stiffened Panels Using Linear Approximations in Lamination Space.* AIAA Journal, vol. 46, no. 9, pages 2387–2391, 2008. 119

[Herencia *et al.* 2008b] J.E. Herencia, P.M. Weaver and M.I. Friswell. *Optimization of anisotropic composite panels with T-shaped stiffeners including transverse shear effects and out-of-plane loading.* Structural and Multidisciplinary Optimization, vol. 37, no. 2, pages 165–184, 2008. 95, 104, 119

[Jones 1999] R.M. Jones. Mechanics of composite materials. Hemisphere Pub, 1999. 100

[Jordan & Jacobs 1994] M.I. Jordan and R.A. Jacobs. *Hierarchical mixtures of experts and the EM algorithm.* Neural computation, vol. 6, no. 2, pages 181–214, 1994. 132, 149, 152

[Kleijnen *et al.* 2005] J.P.C. Kleijnen, S.M. Sanchez, T.W. Lucas and T.M. Cioppa. *A user's guide to the brave new world of designing simulation experiments.* INFORMS Journal on Computing, vol. 17, no. 3, pages 263–289, 2005. 130

[Kohavi 1995] R. Kohavi. *A study of cross-validation and bootstrap for accuracy estimation and model selection.* In International joint Conference on artificial intelligence, volume 14, pages 1137–1145. Citeseer, 1995. 131

[Koiter 1962] WT Koiter. *Elastic stability and post-buckling behavior.* Nonlinear Problems, pages 257–275, 1962. 110

[Lascaux & Théodor 1987] P. Lascaux and R. Théodor. Analyse numérique matricielle appliquée à l'art de l'ingénieur, volume 2. Masson, 1987. 78

[Lebedev & Vorovich 2003] L.P. Lebedev and I.I. Vorovich. Functional analysis in mechanics. Springer Verlag, 2003. 110

[Lekhnitskii 1968] S.G. Lekhnitskii. *Anisotropic plates*. Technical report, DTIC Document, 1968. 100, 112

[Levin 1998] D. Levin. *The approximation power of moving least-squares*. Mathematics of Computation, vol. 67, no. 224, pages 1517–1532, 1998. 150, 179

[Liu *et al.* 2004] B. Liu, R.T Haftka and L.T Watson. *Global-local structural optimization using response surfaces of local optimization margins*. Structural and Multidisciplinary Optimization, vol. 27, no. 5, pages 352–359, 2004. 52, 91, 130, 250

[Liu 2001] B. Liu. *Two-level optimization of composite wing structures based on panel genetic optimization*. PhD thesis, Citeseer, 2001. 52, 250

[Löfgren 2011] K.G. Löfgren. *On Envelope Theorems in Economics: Inspired by a Revival of a Forgotten Lecture*. Umeå Economic Studies, 2011. 70

[Meir & Ratsch 2003] R. Meir and G. Ratsch. *An introduction to boosting and leveraging*. Lecture Notes in Computer Science, vol. 2600, pages 118–183, 2003. 151

[Merval *et al.*] A. Merval, M. Samuelides and S. Grihon. *Application of response surface methodology to stiffened panel optimization*. 47th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 1 - 4 May 2006, Newport, Rhode Island. 52, 130

[Merval 2008] A. Merval. *Application des modèles réduits à l'optimisation multi-niveau de structures aéronautiques*. Thèse SupAéro, 2008. 16, 24, 52, 56, 90, 130, 183, 213

[Miki & Sugiyama 1991] M. Miki and Y. Sugiyama. *Optimum design of laminated composite plates using lamination parameters*. In AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 32 nd, Baltimore, MD, Technical Papers. Pt. 1, volume 8, 1991. 94, 105

[Miki 1983] M. Miki. *A graphical method for designing fibrous laminated composites with required in-plane stiffness*. Japan Society for Composite Materials, Transactions, vol. 9, pages 51–55, 1983. 94, 105

[Myers *et al.* 2009] R.H. Myers, D.C. Montgomery and C.M. Anderson-Cook. Response surface methodology, process and product optimization using designed experiments. Wiley, 2009. 150

[Nealen & Darmstadt 2004] A. Nealen and TU Darmstadt. *An as-short-as-possible introduction to the least squares, weighted least squares and moving least squares methods for scattered data approximation and interpolation*. URL: http://www. nealen. com/projects, 2004. 130, 150

[Picard & Cook 1984] R.R. Picard and R.D. Cook. *Cross-validation of regression models*. Journal of the American Statistical Association, vol. 79, no. 387, pages 575–583, 1984. 131

[Ramanathan 1976] R.K. Ramanathan. *A multilevel approach in optimum design of structures including buckling constraints*. PhD thesis, University of California, 1976. 88

[Reddy 2004] J.N. Reddy. Mechanics of laminated composite plates and shells, volume 2. CRC press, 2004. 96, 121

[Rousselet & Chenais 1990] B. Rousselet and D. Chenais. *Continuité et différentiabilité d'éléments propres: application à l'optimisation de structures*. Applied Mathematics and Optimization, vol. 22, no. 1, pages 27–59, 1990. 95, 115, 122, 123

[Saad 1992] Y. Saad. Numerical methods for large eigenvalue problems. Manchester Univ Pr, 1992. 150

[Sanchez *et al.* 2008] E. Sanchez, S. Pintos and N.V. Queipo. *Toward an optimal ensemble of kernel-based approximations with engineering applications*. Structural and Multidisciplinary Optimization, vol. 36, no. 3, pages 247–261, 2008. 131

[Schutte & Haftka 2010] JF Schutte and RT Haftka. *Global structural optimization of a stepped cantilever beam using quasi-separable decomposition*. Engineering Optimization, vol. 42, no. 4, pages 347–367, 2010. 91

[Setoodeh *et al.* 2006] S. Setoodeh, M.M. Abdalla and Z. Gürdal. *Design of variable-stiffness laminates using lamination parameters*. Composites Part B: Engineering, vol. 37, no. 4-5, pages 301–309, 2006. 104

[Simon & Blume 1994] C.P. Simon and L. Blume. Mathematics for economists, volume 7. Norton, 1994. 262

[Simpson *et al.* 2008] T.W. Simpson, V. Toropov, V. Balabanov and F.A.C. Viana. *Design and analysis of computer experiments in multidisciplinary design optimization: a review of how far we have come or not*. In Proceedings of the 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 2008 MAO, Victoria, Canada, 2008. 130

[Smola & Schölkopf 2004] A.J. Smola and B. Schölkopf. *A tutorial on support vector regression*. Statistics and Computing, vol. 14, no. 3, pages 199–222, 2004. 130

[Sobieski] J. Sobieski. *A perspective on the state of multidisciplinary optimization (MDO)*. MDO workshop, Sept. 16 2010, Ft. Worth. 35

[Sobieszczanski-Sobieski & Haftka 1997] J. Sobieszczanski-Sobieski and R.T. Haftka. *Multidisciplinary aerospace design optimization: survey of recent developments*. Structural and Multidisciplinary Optimization, vol. 14, no. 1, pages 1–23, 1997. 53

[Sobieszczanski-Sobieski *et al.* 1981] J. Sobieszczanski-Sobieski, J.F. Barthelemy and KM RYAN. *Sensitivity of optimum solutions to problem parameters*. 1981. 71

[Sobieszczanski-Sobieski *et al.* 1985] J. Sobieszczanski-Sobieski, B.B. James and A.R. Dovi. *Structural optimization by multilevel decomposition*. AIAA journal, vol. 23, no. 11, pages 1775–1782, 1985. 91

[Stakgold 1971] I. Stakgold. *Branching of solutions of nonlinear equations*. SIAM review, vol. 13, no. 3, pages 289–332, 1971. 110

REFERENCES

[Svanberg 1987] K. Svanberg. *The method of moving asymptotes—a new method for structural optimization.* International journal for numerical methods in engineering, vol. 24, no. 2, pages 359–373, 1987. 40

[Terada *et al.* 2001] Y. Terada, A. Todoroki and Y. Shimamura. *Stacking sequence optimizations using fractal branch and bound method for laminated composites.* JSME International Journal Series A, vol. 44, no. 4, pages 490–498, 2001. 95

[Tibshirani *et al.* 2001] R. Tibshirani, G. Walther and T. Hastie. *Estimating the number of clusters in a data set via the gap statistic.* Journal of the Royal Statistical Society: Series B (Statistical Methodology), vol. 63, no. 2, pages 411–423, 2001. 145, 146

[Todoroki & Sasai 2002] A. Todoroki and M. Sasai. *Stacking sequence optimizations using GA with zoomed response surface on lamination parameters.* Advanced Composite Materials, vol. 11, no. 3, pages 299–318, 2002. 95

[Todoroki *et al.* 2003] A. Todoroki, K. Suenaga and Y. Shimamura. *Stacking sequence optimizations using modified global response surface in lamination parameters.* Advanced Composite Materials, vol. 12, no. 1, pages 35–55, 2003. 95

[Tosserams *et al.* 2009] S. Tosserams, L.F.P. Etman and JE Rooda. *A classification of methods for distributed system optimization based on formulation structure.* Structural and Multidisciplinary Optimization, vol. 39, no. 5, pages 503–517, 2009. 208

[Turner *et al.* 1956] MJ Turner, RW Clough, HC Martin and LJ Topp. *Stiffness and deflection analysis of complex structures.* J. Aero. Sci, vol. 23, no. 9, pages 805–823, 1956. 39

[Turvey 1995] G.J. Turvey. Buckling and postbuckling of composite plates. Springer, 1995. 110, 113

[van Beers & Kleijnen 2004] W.C.M. van Beers and J.P.C. Kleijnen. *Kriging interpolation in simulation: a survey.* In Proceedings of the 36th conference on Winter simulation, pages 113–121. Winter Simulation Conference, 2004. 130

[Ventsel & Krauthammer 2001] E. Ventsel and T. Krauthammer. Thin plates and shells. M. Dekker, 2001. 97

[Viana *et al.* 2009] F.A.C. Viana, R.T. Haftka and V. Steffen. *Multiple surrogates: how cross-validation errors can help us to obtain the best predictor.* Structural and Multidisciplinary Optimization, vol. 39, no. 4, pages 439–457, 2009. 131

[Walker *et al.* 1996] M. Walker, S. Adali and V. Verijenko. *Optimization of symmetric laminates for maximum buckling load including the effects of bending-twisting coupling.* Computers & structures, vol. 58, no. 2, pages 313–319, 1996. 95

[Wang & Shan 2007] G.G. Wang and S. Shan. *Review of metamodeling techniques in support of engineering design optimization.* Journal of Mechanical Design, vol. 129, page 370, 2007. 130

[Wendland 2005] H. Wendland. Scattered data approximation, volume 17. Cambridge Univ Pr, 2005. 151

[Whitney 1969] JM Whitney. *The effect of transverse shear deformation on the bending of laminated plates.* Journal of Composite Materials, vol. 3, no. 3, page 534, 1969. 121

[Wu 1983] C.F.J. Wu. *On the convergence properties of the EM algorithm.* The Annals of Statistics, vol. 11, no. 1, pages 95–103, 1983. 141

[Yang 2003] Y. Yang. *Regression with multiple candidate models: selecting or mixing?* Statistica Sinica, vol. 13, no. 3, pages 783–810, 2003. 131, 133

[Zerpa *et al.* 2005] L.E. Zerpa, N.V. Queipo, S. Pintos and J.L. Salager. *An optimization methodology of alkaline–surfactant–polymer flooding processes using field scale numerical simulation and multiple surrogates.* Journal of Petroleum Science and Engineering, vol. 47, no. 3-4, pages 197–208, 2005. 131

A More on optimization and numerical approximation

A.1 Quasiconcave functions and homogeneous functions

We recall here some basic definitions and properties of quasiconcave functions and homogeneous functions. In particular, we also give the Euler identity formula for homogeneous functions. Let $f : V \mapsto \mathbb{R}$ defined over a convex subset V of a vector space, f is *quasiconcave* if

$$\forall x, y \in V \quad \forall \theta \in (0,1) f(\theta x + (1-\theta)y) \geqslant \min(f(x), f(y)) \tag{A.1}$$

or equivalently, a quasiconcave function can be defined in terms of *upper level sets* $S_\alpha = \{x \in V | f(x) \geqslant \alpha\}$ that is, f is quasiconcave if for any $\alpha \in \mathbb{R}$, S_α is a convex set. This is the definition we used in this article. Quasiconcavity (and quasiconvexity) are a generalization of the well known concavity, a concave function is quasiconcave. Quasiconcave functions often appear in economics (e.g Cobb-Douglas function), game theory and optimization. These are assumptions for the minmax theorem.

Let $f : U \mapsto V$ a function defined a vector space U that maps to another vector space V. f is said to be *positive homogeneous* of degree α whenever

$$\forall x \in U \quad \forall \kappa \in \mathbb{R}^+ \quad f(\kappa x) = \kappa^\alpha f(x) \tag{A.2}$$

Linear functions are homogeneous functions of degree 1, the determinant function over $n \times n$ matrices is homogeneous of degree n. One important feature of homogeneous functions whenever they are continuously differentiable is that they admit the Euler identity characterization. $f : \mathbb{R}^n \mapsto \mathbb{R}$ is a positive homogeneous functions if and only if

$$\forall x \in \mathbb{R}^n \quad \sum_{i=1}^{n} x_i \frac{\partial f}{\partial x_i} = \alpha f(x) \tag{A.3}$$

To obtain (A.3), we simply differentiate (A.2) at κx with respect to κ to get

$$\sum_{i=1}^{n} x_i \frac{\partial f(\kappa x)}{\partial x_i} = \alpha \kappa^{\alpha-1} f(x). \tag{A.4}$$

It is worth noting that, if F is homogeneous of degree α and continuously differentiable, then its derivative is homogeneous of degree $\alpha - 1$. We end up with the following useful property, whose proof together with a lot of results and examples on quasiconcave and homogeneous functions can be found in (Simon & Blume 1994). Let f be a homogeneous function of degree 1 and quasiconcave then f is concave.

A.2 Rayleigh-Ritz approximation for buckling and sensitivity analysis

We briefly recall here the Rayleigh-Ritz method for buckling computations. Note that, Rayleigh-Ritz methods are also widely used in structural dynamics, vibrations, quantum chemistry,... The main idea is to write the solution of a partial differential equation system or eigenproblem as a linear combination of test functions. Unlike finite element methods, the test functions are defined over the whole domain and satisfy boundary conditions. These basis functions are usually based on a closed-form solutions of a close unperturbed problem. This means that Rayleigh-Ritz methods are used when solving a pde or eigenproblem close in some sense (simple geometry, simple differential operator) to a simple one and are not as general as finite element methods. They are also close to spectral methods in the sense the test functions are defined over the whole domain and their convergence properties are usually quite good whenever the problem remains close to an original simple problem. In addition to that, as in spectral methods, the discretization step lead to dense matrices and high accuracy can be achieved with much less degrees of freedom than in finite elements methods.

Consider the following Sturm-Liouville problem under its variational form: find $\lambda \in \mathbb{R}$ and $u \in \mathcal{V} - 0$ such that

$$a(u, u) - \lambda b(u, u) = 0 \tag{A.5}$$

with prescribed boundary conditions (e.g $u = 0$ on the boundary of the domain), with a and b being bilinear symmetric forms defined over $\mathcal{V}$, for sake of simplicity we

will assume that they are both coercive and restrict ourselves to eigenproblems. The minmax theorem (or variational characterization) for eigenvalues are local minima of the Rayleigh ratio

$$R(u) = \frac{a(u,u)}{b(u,u)} \tag{A.6}$$

To ensure unicity of eigenmodes it is convenient to add a normalization condition[1] over u (see (Haftka & Gurdal 1992), e.g $b(u,u) = 1$, $||b||_\infty = 1$. Let's take $b(u,u) = 1$. The process of computing eigenvalues and eigenmodes of (A.5) boils down to solving the following problem:

$$\begin{array}{ll} \min_{u \in \mathcal{V}} & R(u) \\ \text{s.t} & b(u,u) = 1 \end{array} \tag{A.7}$$

Instead of solving Problem (A.7), we solve it over a finite-dimensional space $\mathcal{V}_N$. Consider $(u_1, \ldots, u_N)$ the basis of $\mathcal{V}_N \in \mathcal{V}$ and let's denote for $v \in \mathcal{V}_h$

$$v(x,c) = \sum_{i=1}^{N} c_i u_i(x) \tag{A.8}$$

where $c \in \mathbb{R}^N$, and we can write finite-dimensional approximation of a and b

$$a(v,v) = a_N(c) = \sum_{i,j=1}^{N} a_{ij} c_i c_j \tag{A.9}$$

with

$$a_{ij} = a(u_i, u_j) \tag{A.10}$$

and

$$b(v,v) = b_N(c) = \sum_{i,j=1}^{N} b_{ij} c_i c_j \tag{A.11}$$

with

$$b_{ij} = b(u_i, u_j) \tag{A.12}$$

and denote by A and B the (symmetric positive definite) matrices of elements $(a_{ij})_{1 \leqslant i,j \leqslant N}$ and $(b_{ij})_{1 \leqslant i,j \leqslant N}$. We can write the KKT optimality conditions of Problem (A.7) over $\mathbb{R}^N$

$$\min_{c \in \mathbb{R}^N | b_N(c) = 1} \quad a_N(c) \tag{A.13}$$

[1] This is indeed necessary to ensure unicity of the eigenmodes, since they are defined up to a multiplicative constant. This is even more critical when we are to compute the derivative of the eigenmodes

and so $c \in \mathbb{R}^n$ a local minimum of Problem (A.13) necessary satisfies the following conditions (and any point satisfying this is also a local minim since a_N and b_N are convex)

$$\frac{1}{2}\nabla a_n(c) - \lambda \nabla b_N(c) = (A - \lambda B)c = 0 \tag{A.14}$$

with

$$b_N(c) = c^t B c = 1 \tag{A.15}$$

And the linear equation defined Eq. (A.14) has a nontrivial solution if and only if

$$det(A - \lambda B) = 0 \tag{A.16}$$

that the Lagrange multiplier λ is an eigenvalue of A relatively to B.

To make this method more clear, let's take the example of composite plate buckling. As outlined in this paper, for simple geometry like plate, there exists closed-form solution whenever the laminate is orthotropic and the loading is biaxial. In that case, buckling equation boils down to

$$D_{11}\frac{\partial^4 w}{\partial x^4} + D_{22}\frac{\partial^4 w}{\partial y^4} + 2(D_{12} + 2D_{66})\frac{\partial^4 w}{\partial x^2 \partial y^2} = N_x \frac{\partial^2 w}{\partial x^2} + N_y \frac{\partial^2 w}{\partial y^2} \tag{A.17}$$

Closed form solution of Eq. (A.17) are obtained considering the following test functions

$$w_{ij}(x, y) = \sin(i\pi/x)\sin(j\pi/y) \tag{A.18}$$

and writing Eq. (A.17) we obtain the corresponding eigenvalues λ_{ij} as defined in Eq.(3.57). We therefore use $(w_{ij})_{i,j\in[1,N]}$ as test functions, leading to matrices A and B of size $N^2 \times N^2$ defined as

$$A_{(i-1)N+j,(k-i)N+l} = a_D(w_{ij}, w_{kl}) \tag{A.19}$$

and

$$B_{(i-1)N+j,(k-i)N+l} = b_N(w_{ij}, w_{kl}) \tag{A.20}$$

and λ_{cr} is found as the smallest positive eigenvalue of A relatively to B. It is worth noting here that when making the parameters of that buckling computation vary, there

is no need to reassemble each time the full matrices A and B. One easy way to save computation is to assemble elementary matrices once for all such that for instance

$$A_D = D_{11}A_{11} + \ldots + D_{16}A_{16} \tag{A.21}$$

where A_{ij} are matrices of size $N^2 \times N^2$ computed while performing the first analysis. This allows to perform new computations for new values of D faster, since the assembly becomes negligible and the main part of the computations lies in the eigensolver.

Suppose now we want to compute the sensitivity of λ_{cr} w.r.t to any design variable x.

$$\frac{\partial \lambda_{cr}}{\partial x} = w_{cr}^T(\frac{\partial A}{\partial x} + \lambda_{cr}\frac{\partial B}{\partial x})w_{cr} \tag{A.22}$$

with w_{cr} the associated eigenvector whenever λ_{cr} is simple.

A. MORE ON OPTIMIZATION AND NUMERICAL APPROXIMATION

B Detailed results for monolevel and bilevel schemes for the 10 bar truss

In this appendix, we give the detailed results for all methods and for all load cases over the ten bar truss test case.

We give the following results. For methods that gives the same optimum as AIO:

- $M^\star$ is the optimum of the objective at convergence
- N_iter is the number of iterations to converge
- T is the time (in seconds) of computation

B.1 AIO

(a) Node 4 case $\rho_2 = \rho_1$

Load	$M^\star$ (kg)	N_iter	T (s)
1	2.5483	17	0.3921
2	1.8925	17	0.3189
3	1.5132	21	0.3642
4	1.2636	32	0.4638
5	2.4083	18	0.3143
6	1.7723	26	0.4053
7	1.4115	25	0.4090
8	1.1642	19	0.3148
9	2.2800	23	0.3424
10	1.6706	29	0.4172
11	1.3083	20	0.3536
12	1.0654	11	0.2191
13	2.1796	23	0.3314
14	1.5729	22	0.3494
15	1.2246	23	0.3631
16	0.9290	31	0.4988
17	2.0959	33	0.4618
18	1.4887	30	0.4331
19	1.1614	15	0.2487
20	0.9999	16	0.2918
Mean		**22**	**0.36**

(b) Node 5 case $\rho_2 = \rho_1$

Load	$M^\star$ (kg)	N_iter	T (s)
1	1.2911	24	0.4758
2	1.0368	16	0.3285
3	0.8901	30	0.5205
4	0.7921	25	0.3661
5	1.2235	29	0.4293
6	0.9794	21	0.3789
7	0.8311	24	0.3260
8	0.7300	40	0.7651
9	1.1661	38	0.5726
10	0.9167	21	0.3744
11	0.7641	25	0.4373
12	0.6682	24	0.4294
13	1.1034	19	0.4146
14	0.8490	18	0.4001
15	0.6830	28	0.4636
16	0.5682	21	0.3689
17	1.0357	27	0.4831
18	0.7370	16	0.3013
19	0.6606	61	1.0421
20	0.6078	27	0.5306
Mean		**27**	**0.47**

(c) Node 4 case $\rho_2 = 100\rho_1$

Load	$M^\star$ (kg)	N_iter	T (s)
1	108.0930	29	0.6090
2	83.9084	26	0.5368
3	67.1776	34	1.0388
4	54.9260	33	0.7966
5	108.8153	25	0.6961
6	84.0800	34	0.8526
7	67.1866	56	1.3643
8	54.8726	30	0.6247
9	112.3718	31	0.8757
10	85.5374	36	0.8693
11	67.6381	28	0.6811
12	54.9854	31	0.6922
13	115.8133	32	0.7618
14	87.5727	48	1.4087
15	68.9753	43	1.0108
16	56.4905	78	2.1126
17	118.1611	72	1.7954
18	89.1981	34	1.1239
19	70.4585	36	1.0140
20	56.2708	28	0.7547
Mean		**38**	**0.98**

(d) Node 5 case $\rho_2 = 100\rho_1$

Load	$M^\star$ (kg)	N_iter	T (s)
1	61.5738	48	1.1506
2	49.7108	38	0.9090
3	41.4366	33	0.8756
4	35.3700	21	0.4934
5	61.9194	26	0.7293
6	49.7795	31	0.8593
7	41.4344	67	1.6093
8	35.3391	31	0.6973
9	64.9241	37	0.8450
10	50.5048	33	0.7184
11	41.6223	117	2.1125
12	35.3747	44	1.1601
13	69.1238	41	0.9630
14	53.4686	36	0.7939
15	42.8165	58	1.2260
16	35.9087	39	0.8238
17	72.1353	74	1.5962
18	54.2698	48	1.1931
19	43.3478	81	1.8566
20	36.2441	35	0.7802
Mean		**46**	**1.06**

Table B.1: AIO Load case applied at node 4 and 5 case $\rho_1 = 100\rho_2$

B.2 SAND2

(a) Node 4

Load	$M^\star$ (kg)	N_iter	T (s)
1	2.5483	24	0.3509
2	1.8925	40	0.5151
3	1.5132	29	0.7496
4	1.2636	40	0.8793
5	2.4083	22	0.3476
6	1.7723	32	0.5391
7	1.4115	28	0.5030
8	1.1642	38	1.0525
9	2.2800	26	0.3835
10	1.6706	50	1.0137
11	1.3083	27	0.6198
12	1.0654	36	0.8988
13	2.1796	34	0.6471
14	1.5729	37	0.8665
15	1.1967	29	0.5995
16	0.9290	31	0.6583
17	2.0959	34	0.5060
18	1.4887	53	0.9033
19	1.1614	24	0.5051
20	0.9999	23	0.4075
Mean		**32**	**0.64**

(b) Node 5 case $\rho_2 = \rho_1$

Load	$M^\star$ (kg)	N_iter	T (s)
1	1.2911	21	0.4467
2	1.0368	26	0.4451
3	0.8901	30	0.5175
4	0.7921	52	0.9753
5	1.2235	26	0.5101
6	0.9794	32	0.6206
7	0.8311	28	0.5616
8	0.7300	56	0.9329
9	1.1661	26	0.5709
10	0.9167	27	0.5884
11	0.7641	38	0.8274
12	0.6624	29	0.5780
13	1.1034	26	0.4236
14	0.8490	35	0.4815
15	0.6830	29	0.5411
16	0.5503	33	0.9868
17	1.0357	28	0.6075
18	0.7370	22	0.7213
19	0.6436	19	0.6195
20	0.6063	24	0.6390
Mean		**30**	**0.62**

(c) Node 4 case $\rho_2 = 100\rho_1$

Load	$M^\star$ (kg)	N_iter	T (s)
1	108.0930	30	0.5609
2	83.9084	34	0.5992
3	67.1776	30	0.5552
4	54.9260	28	0.6974
5	108.8153	27	0.5272
6	84.0800	24	0.3770
7	67.1866	22	0.4422
8	54.8726	21	0.4733
9	112.3718	34	0.5263
10	85.5374	37	0.5535
11	67.6381	30	0.5284
12	54.9854	27	0.4614
13	115.8133	36	0.5847
14	87.5727	36	0.6050
15	68.9753	37	0.5375
16	55.7383	21	0.4099
17	118.1611	40	0.5902
18	89.1981	40	0.6395
19	70.4395	37	0.4206
20	56.6674	27	0.4303
Mean		**30**	**0.52**

(d) Node 5 case $\rho_2 = 100\rho_1$

Load	$M^\star$ (kg)	N_iter	T (s)
1	61.5738	44	1.2314
2	49.7108	32	0.6404
3	41.4366	35	0.8103
4	35.3700	52	1.3719
5	61.9194	23	0.6647
6	49.7795	33	0.6804
7	41.4344	82	1.8426
8	35.3391	101	2.3263
9	64.9241	55	1.3356
10	50.5048	31	0.9265
11	41.6223	47	1.3052
12	35.3747	72	2.1568
13	69.1238	56	1.1381
14	53.4686	53	1.3098
15	42.8165	35	0.6882
16	35.9087	30	0.6396
17	72.1353	46	0.9084
18	54.2698	41	1.1469
19	43.3478	56	1.4749
20	36.2441	40	1.0091
Mean		**48**	**1.14**

Table B.2: SAND2 Load case at node 4 and 5

B.3 SAND3

(a) Node 4 case $\rho_2 = \rho_1$

Load	$M^\star$ (kg)	N_iter	T (s)
1	2.5483	45	0.8626
2	1.8925	25	0.4573
3	1.5132	26	0.5350
4	1.2636	40	0.7150
5	2.4083	24	0.4030
6	1.7723	33	0.5829
7	1.4115	29	0.5656
8	1.1642	23	0.4437
9	2.2800	36	0.6640
10	1.6706	35	0.6342
11	1.3083	24	0.5535
12	1.0654	22	0.4247
13	2.1796	29	0.5649
14	1.5729	30	0.6153
15	1.1967	29	0.5454
16	0.9290	28	0.4810
17	2.0959	29	0.5088
18	1.4887	42	0.7935
19	1.1614	20	0.4269
20	0.9999	27	0.4638
Mean		**30**	**0.56**

(b) Node 5 case $\rho_2 = \rho_1$

Load	$M^\star$ (kg)	N_iter	T (s)
1	1.2911	17	0.3972
2	1.0368	28	0.4593
3	0.8901	32	0.5349
4	0.7921	39	0.5597
5	1.2235	23	0.5295
6	0.9794	32	0.4035
7	0.8311	31	0.4852
8	0.7300	33	0.5157
9	1.1661	37	0.6764
10	0.9167	27	0.5634
11	0.7641	26	0.4817
12	0.6624	33	0.6346
13	1.1034	31	0.6560
14	0.8490	24	0.4343
15	0.6830	52	0.8862
16	0.5503	42	0.9944
17	1.0357	42	0.8428
18	0.7370	27	0.6906
19	0.6436	15	0.4246
20	0.6063	16	0.4754
Mean		**30**	**0.58**

(c) Node 4 case $\rho_2 = 100\rho_1$

Load	$M^\star$ (kg)	N_iter	T (s)
1	108.0930	31	0.5424
2	83.9084	28	0.4371
3	67.1776	25	0.4326
4	54.9260	25	0.5094
5	108.8153	32	0.5277
6	84.0800	23	0.3974
7	67.1866	27	0.3653
8	54.8726	26	0.4843
9	112.3718	39	0.7221
10	85.5374	44	0.5232
11	67.6381	28	0.3972
12	54.9854	21	0.4093
13	115.8133	35	0.3855
14	87.5727	40	0.6162
15	68.9753	41	0.6254
16	55.7383	27	0.3531
17	118.1611	54	1.0180
18	89.1981	43	0.6088
19	70.4395	43	0.5984
20	56.6674	51	0.6046
Mean		**34**	**0.53**

(d) Node 5 case $\rho_2 = 100\rho_1$

Load	$M^\star$ (kg)	N_iter	T (s)
1	61.5738	36	0.7296
2	49.7108	40	0.7949
3	41.4366	32	0.6307
4	35.3700	49	1.3436
5	61.9194	48	1.1839
6	49.7795	33	0.7171
7	41.4344	29	0.8028
8	35.3391	41	1.1493
9	64.9241	42	1.0099
10	50.5048	37	0.9126
11	41.6223	39	0.7543
12	35.3747	32	0.6977
13	69.1238	49	0.8926
14	53.4686	40	0.7526
15	42.8165	33	1.0303
16	35.9087	20	0.4311
17	72.1353	59	1.0485
18	54.2698	36	0.6377
19	43.3478	42	0.7307
20	36.2441	27	0.4833
Mean		**38**	**0.83**

Table B.3: SAND3 Load case at node 4 and 5

B.4 STIFAIO

(a) Node 4 case $\rho_2 = \rho_1$

Load	$M^\star$ (kg)	N_iter	T (s)
1	2.5483	12	0.6560
2	1.8925	10	0.8357
3	1.5132	15	0.7622
4	1.2636	14	0.7753
5	2.4083	9	0.8186
6	1.7723	12	0.8510
7	1.4115	14	0.8495
8	1.1642	10	0.7521
9	2.2800	14	0.8733
10	1.6706	10	0.8332
11	1.3083	14	0.8343
12	1.0654	10	0.7660
13	2.1796	14	0.9274
14	1.5729	13	0.8570
15	1.2246	14	0.7142
16	0.9290	15	0.7418
17	2.0959	21	0.9982
18	1.4887	16	0.7853
19	1.1614	10	0.7465
20	0.9999	6	0.6360
Mean		**12.65**	**0.8**

(b) Node 5 case $\rho_2 = \rho_1$

Load	$M^\star$ (kg)	N_iter	T (s)
1	1.2911	8	0.6673
2	1.0368	9	0.7178
3	0.8901	9	0.7203
4	0.7921	13	0.7972
5	1.2235	8	0.7837
6	0.9794	10	0.7559
7	0.8311	12	0.7326
8	0.7300	9	0.6885
9	1.1661	11	0.7778
10	0.9167	10	0.6844
11	0.7641	15	0.7890
12	0.6624	11	0.6798
13	1.1034	10	0.7714
14	0.8490	21	0.8028
15	0.6922	6	0.6326
16	0.5503	7	0.6056
17	1.0357	15	0.8970
18	0.7370	11	0.7442
19	0.6436	5	0.6601
20	0.6063	4	0.6189
Mean		**10.2**	**0.72**

(c) Node 4 case $\rho_2 = 100\rho_1$

Load	$M^\star$ (kg)	N_iter	T (s)
1	108.0930	22	1.0390
2	83.9084	15	0.9695
3	67.1776	15	0.8647
4	54.9260	12	0.8438
5	108.8153	20	1.1189
6	84.0800	15	1.0230
7	67.1866	14	0.9850
8	54.8726	10	0.8115
9	112.3718	33	1.4759
10	85.5374	23	1.1684
11	67.6381	16	1.0209
12	54.9854	9	0.9320
13	115.8133	40	1.8721
14	87.5727	30	1.4421
15	68.9753	24	1.1734
16	56.4905	27	0.8195
17	118.1611	28	1.4492
18	89.1981	28	1.3813
19	69.8998	16	0.9897
20	58.6315	16	0.8101
Mean		**20.65**	**1.10**

(d) Node 5 case $\rho_2 = 100\rho_1$

Load	$M^\star$ (kg)	N_iter	T (s)
1	61.5738	11	0.8040
2	49.7108	13	0.8393
3	41.4366	14	0.7244
4	35.3700	12	0.7478
5	61.9194	12	0.9475
6	49.7795	13	0.9270
7	41.4344	19	0.8484
8	35.3391	12	0.8600
9	64.9241	19	1.0344
10	50.5048	17	1.0022
11	41.6223	14	0.8526
12	35.3747	17	0.8371
13	69.1238	37	1.3270
14	53.4686	25	1.2873
15	42.8165	23	0.9318
16	37.2068	6	0.5979
17	72.1353	32	1.3062
18	54.2698	35	0.9709
19	43.9612	4	0.7238
20	36.5319	5	0.5942
Mean		**17**	**0.90**

Table B.4: STIFAIO Load case at node 4 and 5

B.5 STISAND

(a) Node 4 case $\rho_2 = \rho_1$

Load	$M^\star$ (kg)	N_iter	T (s)
1	2.5483	15	0.7206
2	1.8925	18	1.0373
3	1.5132	18	1.0794
4	1.2636	13	1.7451
5	2.4083	19	1.1175
6	1.7723	21	1.1797
7	1.4115	16	1.0212
8	1.1642	20	0.9436
9	2.2800	20	0.9485
10	1.6706	23	1.3884
11	1.3083	15	1.1015
12	1.0654	12	1.0273
13	2.1796	22	1.0366
14	1.5729	22	1.0687
15	1.2246	20	0.8310
16	0.9290	20	1.2015
17	2.0959	28	1.1132
18	1.4887	27	0.9026
19	1.1614	12	1.1353
20	0.9999	7	0.6492
Mean		**18.4**	**1.06**

(b) Node 5 case $\rho_2 = \rho_1$

Load	$M^\star$ (kg)	N_iter	T (s)
1	1.2911	29	1.1659
2	1.0368	16	0.7998
3	0.8901	11	1.1251
4	0.7921	12	0.9201
5	1.2235	18	2.0823
6	0.9794	15	1.0274
7	0.8311	18	1.1722
8	0.7300	9	1.5634
9	1.1661	15	1.0353
10	0.9167	12	2.2093
11	0.7641	18	1.1493
12	0.6624	18	1.8567
13	1.1034	18	1.0075
14	0.8490	29	0.9933
15	0.6922	9	0.7607
16	0.5503	4	0.6590
17	1.0357	28	1.7076
18	0.7372	27	2.4184
19	0.6436	6	1.2293
20	0.6063	4	0.6223
Mean		**15.8**	**1.27**

(c) Node 4 case $\rho_2 = 100\rho_1$

Load	$M^\star$ (kg)	N_iter	T (s)
1	108.0930	39	1.1832
2	83.9084	21	1.0172
3	67.1776	18	0.8739
4	54.9260	14	0.8615
5	108.8153	36	1.3174
6	84.0800	25	1.1641
7	67.1866	14	0.9377
8	54.8726	10	0.8690
9	112.3718	38	1.1702
10	85.5374	39	1.1682
11	67.6381	47	1.5054
12	54.9854	22	1.1332
13	115.8133	43	1.3155
14	87.5727	22	0.9919
15	68.9753	30	1.0501
16	56.4905	11	1.7475
17	118.1611	37	1.3546
18	89.6012	26	1.3289
19	69.8998	26	3.1187
20	58.6315	13	1.6101
Mean		**26.55**	**1.29**

(d) Node 5 case $\rho_2 = 100\rho_1$

Load	$M^\star$ (kg)	N_iter	T (s)
1	61.5738	18	1.0574
2	49.7108	25	3.1027
3	41.4366	16	1.1184
4	35.3700	9	1.0982
5	61.9194	22	1.2026
6	49.7795	30	2.1993
7	41.4344	14	1.0765
8	35.3391	10	0.9717
9	64.9241	26	1.2925
10	50.5048	20	1.8941
11	41.6223	42	3.3185
12	35.3747	12	1.0087
13	69.1238	21	2.0392
14	53.4701	10	1.7628
15	42.8165	34	2.4533
16	37.2068	4	1.3867
17	72.1367	10	0.8427
18	54.2698	35	2.3047
19	43.9612	4	1.4380
20	36.5319	4	1.7601
Mean		**18.3**	**1.66**

Table B.5: STIFSAND Load case at node 4 and 5

B.6 STIFSAND2

(a) Node 4 case $\rho_2 = \rho_1$

Load	$M^\star$ (kg)	N_iter	T (s)
1	2.5483	17	0.4176
2	1.8925	16	0.5011
3	1.5132	20	0.6021
4	1.2636	13	0.5942
5	2.4083	17	0.5012
6	1.7723	25	0.5405
7	1.4115	16	0.7156
8	1.1642	16	0.5621
9	2.2800	22	0.5346
10	1.6706	27	0.6932
11	1.3083	21	0.7560
12	1.0654	16	0.6962
13	2.1796	22	0.6684
14	1.5729	30	0.7202
15	1.2246	13	0.4375
16	0.9290	16	1.6134
17	2.0959	27	0.6903
18	1.4887	29	0.6032
19	1.1614	10	0.5887
20	0.9999	7	0.4607
Mean		**19**	**0.64**

(b) Node 5 case $\rho_2 = \rho_1$

Load	$M^\star$ (kg)	N_iter	T (s)
1	1.2911	13	0.5042
2	1.0368	17	0.7535
3	0.8901	11	0.3865
4	0.7921	15	0.4859
5	1.2235	22	0.5260
6	0.9794	20	0.4593
7	0.8311	17	0.4151
8	0.7300	10	0.4301
9	1.1661	17	0.4599
10	0.9167	12	0.4229
11	0.7641	24	0.5088
12	0.6624	16	0.4547
13	1.1034	17	1.2814
14	0.8490	21	0.8429
15	0.6922	9	0.4038
16	0.5503	4	0.3480
17	1.0357	31	0.7256
18	0.7370	20	0.4677
19	0.6436	6	0.4019
20	0.6063	4	0.3385
Mean		**15.3**	**0.53**

(c) Node 4 case $\rho_2 = 100\rho_1$

Load	$M^\star$ (kg)	N_iter	T (s)
1	108.0930	29	0.6681
2	83.9084	21	0.5862
3	67.1776	17	0.5550
4	54.9260	14	0.6887
5	108.8153	29	0.7600
6	84.0800	19	0.6226
7	67.1866	18	0.7807
8	54.8726	10	0.5767
9	112.3718	47	1.0063
10	85.5374	28	0.6764
11	67.6381	22	0.7799
12	54.9854	13	0.6238
13	115.8133	37	0.7248
14	87.5727	23	0.6389
15	68.9753	33	0.5913
16	56.4905	11	1.3584
17	118.1611	28	0.6987
18	89.1981	37	0.8382
19	69.8998	23	0.7025
20	58.6367	13	2.0168
Mean		**23.6**	**0.79**

(d) Node 5 case $\rho_2 = 100\rho_1$

Load	$M^\star$ (kg)	N_iter	T (s)
1	61.5738	19	0.5181
2	49.7108	20	0.4901
3	41.4366	15	0.4440
4	35.3700	13	0.6365
5	61.9194	18	0.8422
6	49.7795	14	0.6751
7	41.4344	13	0.5949
8	35.3391	15	0.7731
9	64.9241	17	0.7765
10	50.5051	19	1.3824
11	41.6223	15	0.5211
12	35.3747	12	0.6047
13	69.1238	23	0.9725
14	53.4701	13	1.3864
15	42.8165	34	1.4972
16	37.2069	4	1.1622
17	72.1367	9	0.5978
18	54.2698	30	0.8200
19	43.9612	4	0.3870
20	36.5319	4	2.4111
Mean		**15.5**	**0.87**

Table B.6: STIFSAND2 Load case at node 4 and 5

B.7 StiffOpt

(a) Node 4 case $\rho_2 = \rho_1$

	$M^\star$ (kg)	Err	Iter	T (s)
1	2.5498	0.06 %	30	3.09
2	1.8947	0.12 %	33	3.39
3	1.5162	0.20 %	11	1.49
4	1.2651	0.12 %	14	1.63
5	2.4098	0.06 %	29	3.25
6	1.7736	0.07 %	26	3.20
7	1.4129	0.09 %	17	2.18
8	1.1657	0.12 %	11	1.40
9	2.2816	0.07 %	31	3.72
10	1.6723	0.10 %	30	3.61
11	1.3098	0.11 %	12	1.73
12	1.0669	0.14 %	8	1.10
13	2.1811	0.07 %	26	3.20
14	1.5743	0.09 %	16	2.21
15	1.1977	0.09 %	16	1.93
16	0.9313	0.25 %	12	1.30
17	2.0975	0.07 %	26	3.41
18	1.4908	0.14 %	19	1.99
19	1.1632	0.16 %	9	1.01
20	1.0473	0.14 %	8	1.07
Mean.		**0.11 %**	**19.2**	**2.23**

(b) Node 5 case $\rho_2 = \rho_1$

	$M^\star$ (kg)	Err	Iter	T (s)
1	1.2940	0.23 %	27	2.78
2	1.0396	0.27 %	22	2.52
3	0.8926	0.28 %	19	2.07
4	0.7951	0.38 %	5	0.60
5	1.2262	0.22 %	23	2.70
6	0.9816	0.23 %	20	2.34
7	0.8334	0.28 %	18	1.98
8	0.7325	0.34 %	14	1.44
9	1.1683	0.19 %	21	2.53
10	0.9190	0.25 %	20	2.05
11	0.7666	0.32 %	20	2.17
12	0.6649	0.38 %	12	1.31
13	1.1057	0.21 %	20	2.19
14	0.8516	0.30 %	13	1.61
15	0.6863	0.48 %	8	0.95
16	0.5539	0.65 %	6	0.65
17	1.0382	0.24 %	14	1.90
18	0.7405	0.48 %	7	0.80
19	0.6480	0.68 %	8	0.83
20	0.6514	0.43 %	9	1.10
M		**0.34 %**	**15.3**	**1.73**

(c) Node 4 case $\rho_2 = 100\rho_1$

	$M^\star$ (kg)	Err	Iter	T (s)
1	121.71	12.59 %	30	3.24
2	90.19	7.50 %	33	3.56
3	71.13	5.90 %	16	1.93
4	57.94	5.50 %	14	1.64
5	121.56	11.72 %	29	3.42
6	90.13	7.20 %	26	3.40
7	71.09	5.81 %	17	2.22
8	57.77	5.28 %	11	1.39
9	121.36	8.0 %	31	3.84
10	90.06	5.29 %	30	3.57
11	70.86	4.78 %	12	1.72
12	57.48	4.54 %	8	1.06
13	121.54	4.94 %	26	3.37
14	90.21	3.02 %	19	2.48
15	70.52	2.25 %	16	1.96
16	56.61	0.21 %	12	1.25
17	121.82	3.10 %	26	3.48
18	90.77	1.77 %	19	2.11
19	71.14	0.98 %	9	1.04
20	57.38	4.11 %	8	1.09
M		**5.22 %**	**19.6**	**2.39**

(d) Node 5 case $\rho_2 = 100\rho_1$

	$M^\star$ (kg)	Err	Iter	T (s)
1	72.57	17.86 %	27	2.96
2	54.02	8.67 %	22	2.69
3	43.31	4.54 %	19	2.22
4	36.25	2.51 %	5	0.65
5	72.69	17.40 %	23	2.80
6	54.93	10.36 %	20	2.43
7	44.36	7.06 %	18	2.03
8	37.29	5.52 %	14	1.45
9	73.60	13.37 %	21	2.58
10	55.97	10.83 %	20	2.10
11	45.34	8.93 %	20	2.22
12	37.95	7.28 %	12	1.38
13	74.63	7.98 %	20	2.24
14	56.61	5.89 %	13	1.62
15	45.34	5.90 %	8	0.97
16	37.37	4.08 %	6	0.67
17	75.28	4.36 %	14	1.90
18	56.40	3.93 %	3	0.47
19	44.13	1.81 %	8	0.82
20	37.94	7.16 %	8	0.96
M		**7.77 %**	**15.05**	**1.76**

Table B.7: STIFFOPT Load case at node 4 and 5

B.8 Target Rigidity

(a) Node 4 case $\rho_2 = \rho_1$

Load	$M^\star$	Err	Iter	T (s)
1	2.6397	3.59 %	1	6.66
2	1.9119	1.03 %	9	20.44
3	1.5356	1.48 %	1	27.49
4	1.2854	1.73 %	1	19.89
5	2.5018	3.89 %	1	4.11
6	1.7982	1.46 %	1	21.73
7	1.4283	1.19 %	13	31.90
8	1.1909	2.29 %	1	19.88
9	2.3864	4.67 %	1	12.22
10	1.6727	0.13 %	10	25.08
11	1.3354	2.07 %	1	24.12
12	1.0920	2.50 %	1	15.31
13	2.2709	4.19 %	5	19.78
14	1.6048	2.02 %	1	24.92
15	1.2215	2.08 %	7	32.34
16	0.9602	3.36 %	1	16.50
17	2.1855	4.27 %	1	28.90
18	1.4665	1.49 %	4	24.28
19	1.1699	0.73 %	13	20.57
20	0.9855	5.77 %	10	13.78
M.		**2.50 %**	4.2	**20.5**

(b) Node 5 case $\rho_2 = \rho_1$

Load	$M^\star$	Err	Iter	T (s)
1	119.26	10.33 %	2	20.52
2	85.36	1.73 %	1	33.78
3	68.08	1.34 %	2	26.45
4	55.61	1.24 %	1	7
5	119	9.36 %	2	31.19
6	85.24	1.38 %	1	7
7	68.11	1.37 %	1	6.01
8	55.51	1.16 %	1	20.48
9	119.13	6.02 %	2	26.69
10	86.01	0.55 %	2	10.02
11	68.44	1.18 %	1	5.32
12	56.46	2.68 %	1	14.89
13	121.19	4.64 %	2	31.92
14	89.43	2.12 %	2	9.81
15	70.89	2.78 %	13	29.47
16	59.85	5.94 %	15	13.90
17	122.95	4.05 %	15	27.42
18	91.08	2.11 %	15	9.11
19	75.03	6.49 %	7	5.72
20	59.78	8.46 %	9	12.86
		3.75 %	**4.7**	**17.48**

Table B.8: Target Rigidity Load case at node 4

B.9 MinMass

(a) Node 4 case $\rho_2 = \rho_1$

Load	$M^\star$	Err	Iter	T (s)
1	2.6397	3.59 %	1	7.49
2	1.9026	0.53 %	3	13.16
3	1.5055	0.50 %	5	20.62
4	1.2854	1.73 %	1	23.18
5	2.5018	3.89 %	1	3.8
6	1.7679	0.25 %	13	20.07
7	1.4361	1.74 %	1	22.59
8	1.1909	2.29 %	1	22.88
9	2.3864	4.67 %	1	4.16
10	1.6749	0.25 %	5	20.03
11	1.3354	2.07 %	1	22.67
12	1.0920	2.50 %	1	18.27
13	2.2775	4.49 %	1	20.23
14	1.6000	1.72 %	15	20.76
15	1.2302	2.80 %	1	23.76
16	0.9601	3.35 %	1	19.52
17	2.1807	4.04 %	12	23.51
18	1.4678	1.41 %	13	16.43
19	1.1943	2.83 %	1	17.29
20	0.9907	5.27 %	15	16.48
		2.48 %	**4.65**	**17.84**

(b) Node 4 case $\rho_2 = 100\rho_1$

Load	$M^\star$	Err	Iter	T (s)
1	118.82	9.93 %	2	9.19
2	85.36	1.73 %	1	6.84
3	68.21	1.53 %	1	5.48
4	55.61	1.24 %	1	5.59
5	118.80	9.17 %	2	12.18
6	85.24	1.38 %	1	18.66
7	68.11	1.37 %	1	5.27
8	55.51	1.16 %	1	5.51
9	119.43	6.28 %	2	20.05
10	86.04	0.59 %	1	23.67
11	68.44	1.18 %	1	10.75
12	56.43	2.63 %	2	14.7
13	121.09	4.56 %	2	20.87
14	89.48	2.17 %	1	6.76
15	70.82	2.67 %	2	12.47
16	59.74	5.75 %	13	16.62
17	123.24	4.30 %	1	13
18	91.19	2.24 %	3	6.04
19	74.40	5.59 %	6	6.85
20	59.86	8.61 %	11	12.25
		3.7 %	**2.75**	**11.64**

Table B.9: MinMass Load case at node 4

B.10 Mix scheme

(a) Node 4 case $\rho_2 = \rho_1$

Load	$M^\star$	Err	Iter	T (s)
1	2.6397	3.59 %	1	3.42
2	1.9216	1.54 %	1	17.58
3	1.5356	1.48 %	1	20.46
4	1.2854	1.73 %	1	15.97
5	2.5018	3.89 %	1	4.74
6	1.7810	0.49 %	14	20.84
7	1.4326	1.49 %	15	26.02
8	1.1909	2.29 %	1	20
9	2.3864	4.67 %	1	4.23
10	1.6924	1.30 %	13	24.08
11	1.3354	2.07 %	1	20.5
12	1.0920	2.50 %	1	18.6
13	2.2595	3.67 %	9	23.02
14	1.6048	2.02 %	1	17.78
15	1.2302	2.80 %	1	25.47
16	0.9601	3.35 %	1	16.95
17	2.1278	1.52 %	13	26.77
18	1.4489	2.67 %	13	18.86
19	1.1943	2.83 %	1	21.82
20	0.9915	5.19 %	13	19.27
M.		**2.55 %**	**5.15**	**18.32**

(b) Node 4 case $\rho_2 = 100\rho_1$

Load	$M^\star$	Err	Iter	T (s)
1	121.36	12.27 %	2	25.57
2	85.36	1.73 %	1	25.02
3	68.21	1.53 %	1	29.16
4	55.61	1.24 %	1	25.00
5	120.31	10.56 %	2	26.41
6	84.20	0.14 %	5	19.83
7	68.11	1.37 %	1	6.44
8	55.51	1.16 %	1	19.93
9	119.10	5.99 %	2	24.9
0 10	85.99	0.53 %	2	19.98
11	67.42	0.32 %	3	29.67
12	56.18	2.17 %	3	19.09
13	120.31	3.88 %	7	17.78
14	88.64	1.22 %	3	16.25
15	70.84	2.70 %	2	15.22
16	59.82	5.89 %	15	16.71
17	122.48	3.65 %	15	17.15
18	91.03	2.05 %	12	17.96
19	74.53	5.78 %	15	22.61
20	59.83	8.56 %	8	13.43
M.		**3.64 %**	**5.05**	**20.4**

Table B.10: Mix Load case at node 4

B.11 MaxMargin

B. DETAILED RESULTS FOR MONOLEVEL AND BILEVEL SCHEMES FOR THE 10 BAR TRUSS

(a) Node 4 case $\rho_2 = \rho_1$

Load	$M^\star$	Err	Iter	T (s)
1	2.7840	9.25 %	7	19.02
2	1.9431	2.67 %	6	17.63
3	1.6871	11.49 %	1	18.4
0 4	1.6111	27.50 %	1	12.39
5	2.5358	5.30 %	1	22.1
0 6	1.9391	9.41 %	1	15.96
7	1.5471	9.60 %	1	21.13
8	1.4240	22.32 %	1	18.87
9	2.3958	5.08 %	1	20.65
10	1.7991	7.69 %	1	20.25
11	1.4208	8.60 %	1	24.27
12	1.3237	24.24 %	1	20.32
13	2.2558	3.50 %	3	11.21
14	1.5938	1.33 %	2	18.24
15	1.2924	8 %	1	19.94
16	0.9535	2.64 %	9	9.7
0 17	2.1941	4.68 %	1	21.34
18	1.6616	11.62 %	1	21.34
19	1.2337	6.23 %	2	14.34
20	1.0339	1.14 %	1	14.77
M.		**9.11 %**	**2.15**	**18.09**

(b) Node 4 case $\rho_2 = 100\rho_1$

Load	$M^\star$	Err	Iter	T (s)
1	165.41	53.03 %	9	20.78
2	98.46	17.34 %	4	23.45
3	90.72	35.04 %	4	25.86
4	58.42	6.36 %	1	24.79
5	114.32	5.06 %	10	24.02
6	107.44	27.78 %	10	24.04
7	75.43	12.27 %	3	13.15
8	58.27	6.19 %	1	22.12
9	116.12	3.34 %	6	20.94
10	108.25	26.55 %	4	28.46
11	68.43	1.17 %	9	19.57
12	59.12	7.52 %	1	12.03
13	119.10	2.84 %	3	23.70
14	112.85	28.86 %	6	21.24
15	71.32	3.40 %	1	19.32
16	64.06	13.40 %	2	15.62
17	134.31	13.67 %	3	23.32
18	99.18	11.19 %	3	11.72
19	73.17	3.85 %	1	14.52
20	65.20	18.30 %	4	15.91
M.		**14.86 %**	**4.25**	**20.22**

Table B.11: MaxMarge case at node 4 case $\rho_2 = 100\rho_1$

Optimisation biniveau de structures aéronautiques composites

Ce travail de thèse s'inscrit dans le domaine de l'optimisation de structures aéronautiques composites. On cherche à rendre possible le traitement de problèmes de dimensionnement de telles structures, telles que celles rencontrées dans l'industrie aéronautique. Ce type de problèmes présente deux aspects bloquants. En premier lieu, la taille des structures et le type de matériaux rendent le problème d'optimisation à la fois de très grande taille et de variables mixtes (continues, discrètes). D'autre part, le très grand nombre d'analyses de stabilité locale (flambage) nécessaires rend le problème d'optimisation très difficile à traiter en terme de coût de calculs. On cherche donc à résoudre le premier aspect au travers de schémas d'optimisation dits de *décomposition* qui permettent de décomposer le problème d'optimisation initial en une multitude de sous problèmes d'optimisations pouvant être résolus en parallèle et dont le couplage est résolu par un problème d'optimisation sur un ensemble de variables réduit. L'équivalence théorique entre les différents problèmes d'optimisation (en termes de minima locaux) est prouvée et on présente et développe un schéma adapté à la fois aux spécificités des composites et aux contraintes industrielles. Le second point est résolu de manière originale par le développement d'une stratégie d'approximation des contraintes de stabilité. Cette stratégie de *mélanges d'experts* se base sur des outils statistiques avancés et se révèle adaptée au comportement des composites. Les deux principales avancées de ce travail sont validées sur des cas test académiques et sur une structure aéronautique réaliste. Le fil directeur de ce travail est la mécanique des structures composites, néanmoins le caractère pluridisciplinaire du sujet nous a conduit à des incursions vers les domaines des statistiques (apprentissage), de l'analyse numérique (étude de l'équation aux dérivées partielles relative au flambage) et enfin de l'optimisation théorique.

Mots clés : Optimisation de structure, matériaux composites, optimisation biniveau, mélange d'experts, flambage, structures aéronautiques.

Bilevel optimization of large scale composite structures

This work lies in the field of aerospace composite structures optimization. We are interested in making possible the treatment of large scale optimization problems, as the ones encountered in aerospace design offices. Resolution of such problem needs two main obstacles to be removed. First one lies in the typical large size of problems and the mixed type of design variables: continuous (geometric dimensions, internal loads) and discrete (ply orientations for laminates). Second one lies in the tremendous amount of local stability analyses (buckling) to be performed in one standard optimization. First aspect is solved with the help of decomposition methods that allow breaking up the initial optimization problem in a multitude of optimization sub problems of reduced dimensions. These problems can be solved concurrently, however the internal load redistribution makes them coupled and an upper level optimization problem is needed to solve this coupling. Innovative solutions, both in terms of composite material mechanics representation and in terms of theoretical
optimization properties are presented. Second aspect is solved through the development of an innovative approximation scheme, tailored to buckling behaviour specificities, namely mixture of experts. In particular, the piecewise-like behaviour of such functions is considered. This innovative method relies on advanced statistical tools from unsupervised learning (clustering, law mixture). Finally the two main innovations are extensively discussed and tested over academic benchmark. They are eventually combined for a realistic structural optimization problem (fuselage panel) and allowed retrieving the same weight as traditional method with less iterations. Although the main theme is mechanics and structural optimization, the multidisciplinary aspect of the subject included some research questions and answers in statistics field (statistical learning), numerical analysis (buckling partial differential equation) and theoretical optimization.

Keywords: structural optimization, composite materials, bilevel optimization, mixture of experts, buckling, aerospace structures.