



UNIVERSITAT POLITÈCNICA DE CATALUNYA  
BARCELONATECH

Escola Tècnica Superior d'Enginyeries  
Industrial i Aeronàutica de Terrassa

Titulació:

**Grau en Enginyeria de Tecnologies Aeroespacials**

Alumne:

**Alexandre Cortiella Segarra**

Títol TFG:

**Study of numerical techniques for structural optimization in aeronautics**

Director TFG:

**Juan Carlos Cante Teran**

Codirector TFG:

**Álex Ferrer Ferré**

**Joaquín Hernández Ortega**

Tutor TFG:

**Daniel Garcia Almiñana**

Convocatòria de lliurament del TFG:

**juny 2014**

Contingut d'aquest volum: **MEMÒRIA**





UNIVERSITAT POLITÈCNICA DE CATALUNYA  
BARCELONATECH

Escola Tècnica Superior d'Enginyeries  
Industrial i Aeronàutica de Terrassa

---

## **Study of numerical techniques for structural optimization in aeronautics**

---

Escola Tècnica Superior d'Enginyeria Industrial i Aeronàutica de Terrassa  
(ETSEIAT)

**Grau en Enginyeria de Tecnologies Aeroespacials**

Director TFG:

**Juan Carlos Cante Teran**

Codirector TFG:

**Álex Ferrer Ferré**

**Joaquín Hernández Ortega**

Tutor TFG:

**Daniel Garcia Almiñana**

**June 2014**



[This page intentionally left blank]

## Abstract

The present project is concerned with the study of the most recent numerical simulations for structural optimization problems in aeronautics. Different techniques will be studied in order to solve non-linear constrained optimization problems with non-convex domains. In addition, the Finite Element Method will also be applied to handle the elastostatic behavior of structures.

The computational implementation of numerical methods will be carried out by the well-known programming environment MATLAB. Examples of benchmarks, validation and aeronautical applications will be considered.

The last part of this project will focus on the optimization of an aircraft structure using structural optimization techniques.

**Keywords:** structural optimization, topology optimization, SIMP, minimum compliance, Finite Element Method, elastostatics, Method of Moving Asymptotes, Optimality Criteria, aeronautical structures, numerical methods, MATLAB.

## Summary

The present project is divided in five parts: introduction, theoretical formulation, structural topology optimization code, aeronautical application and project management.

The introduction discusses the main objectives and scope of the project, proposed requirements, the reason why this project has been developed and the state of the art of structural optimization methods.

In the second part, the theoretical contents of the project are treated. This section presents the formulation of constrained optimization methods, the basic concepts of the finite element method applied to linear elastostatics and the foundations of structural topology optimization.

The third part talks about the structure of the topology optimization code developed in MATLAB and some common benchmarks to validate a proper functioning of the program. It is also discussed the pre and post processing stages, a general block diagram of the code is presented and typical benchmarks are analysed.

The forth part, aeronautical application, presents the selected aeronautical structural component to optimize. First, some examples of aeronautical components obtained by topology optimization are shown, then the initial design model of the target structure is developed and finally the final results are discussed.

The last part summarizes the project management issues. The final budget for the project as well as the project planning are provided. Additionally, the environmental aspects are also taken into account.

## Acknowledgements

I personally would like to thank the director Juan Carlos Cante and co-directors Àlex Ferrer and Joaquín Hernández for their collaboration during the development of the project. Their recommendations, support and motivation have made possible to achieve the proposed goals.

I would also like to show my gratitude to the tutor Daniel García-Almiñana and Víctor López Grimau for their follow-up and care of formal aspects of the project.

## Contents

<b>List of Figures</b>	viii
<b>List of Tables</b>	xiii
<b>Nomenclature</b>	xiv
<b>1 Introduction</b>	1
1.1 Aim . . . . .	1
1.2 Scope . . . . .	1
1.3 Requirements . . . . .	2
1.4 Justification . . . . .	2
1.5 State of the Art . . . . .	3
<b>I Theoretical formulation</b>	7
<b>2 Optimization Methods</b>	8
2.1 Introduction to constrained optimization . . . . .	8
2.1.1 Convex and non-convex optimization . . . . .	9
2.1.2 Necessary conditions for optimality . . . . .	10
2.1.3 Sufficient conditions for optimality . . . . .	11
2.2 Primal Methods . . . . .	12
2.3 Dual Methods . . . . .	18
2.4 Penalty and Barrier Methods . . . . .	22
<b>3 Finite Element Method in linear elastostatics</b>	26
3.1 Continuum formulation . . . . .	26
3.1.1 General formulation . . . . .	26
3.1.2 Principles of minimum potential energy and virtual displacements . . . . .	28

3.1.3	Matrix notation . . . . .	29
3.1.4	Equilibrium principles . . . . .	30
3.2	Finite Element discretization . . . . .	32
3.2.1	Direct Stiffness Method . . . . .	33
3.2.2	Element Stiffness Method . . . . .	36
3.2.2.1	Linear triangular element T3 . . . . .	37
3.2.2.2	Bilinear quadrilateral element Q4 . . . . .	38
<b>4</b>	<b>Structural Optimization</b>	<b>41</b>
4.1	Introduction . . . . .	41
4.2	Types of structural optimization problems . . . . .	42
4.3	Topology Optimization . . . . .	43
4.3.1	Minimum Compliance Design . . . . .	44
4.3.2	SIMP Method . . . . .	45
4.3.3	Solution methods to structural topology optimization problems . . . . .	47
4.3.3.1	Optimality Criteria Method (OC) . . . . .	47
4.3.3.2	The Method of Moving Asymptotes (MMA) . . . . .	49
4.3.3.3	Sensitivity analysis . . . . .	52
4.3.4	Numerical instabilities . . . . .	53
4.3.4.1	Checkerboard pattern . . . . .	53
4.3.4.2	Mesh-dependency . . . . .	55
4.3.4.3	Local minima . . . . .	56
4.3.5	General algorithm . . . . .	57
<b>II</b>	<b>Structural Topology Optimization Code</b>	<b>59</b>
<b>5</b>	<b>Pre and post processing</b>	<b>60</b>
5.1	Pre-processing . . . . .	60
5.2	Post-processing . . . . .	61
<b>6</b>	<b>Code Design</b>	<b>62</b>
6.1	Input parameters . . . . .	62
6.2	Block diagram . . . . .	63
<b>7</b>	<b>Benchmarks: Test application and validation</b>	<b>66</b>
7.1	Selected benchmarks . . . . .	66
7.1.1	Messerschmitt–Bölkow–Blohm (MBB) beam . . . . .	67
7.1.2	Long cantilever beam . . . . .	72
7.2	Convergence analysis . . . . .	74
7.3	Conclusions . . . . .	78

<b>III Aeronautical application</b>	<b>80</b>
<b>8 Introduction</b>	<b>81</b>
8.1 Structural optimization in the aerospace industry . . . . .	81
8.2 Proposed application . . . . .	83
<b>9 Initial Design</b>	<b>85</b>
9.1 Wing rib geometry . . . . .	85
9.2 Wing rib model . . . . .	86
9.2.1 CAD Model . . . . .	86
9.2.2 Mesh . . . . .	86
9.3 Loading conditions . . . . .	87
<b>10 Optimized Design</b>	<b>89</b>
10.1 Wing Box . . . . .	89
10.2 Leading Edge . . . . .	91
10.3 Trailing Edge . . . . .	93
10.4 Final design . . . . .	95
<b>IV Project Management</b>	<b>97</b>
<b>11 Budget</b>	<b>98</b>
<b>12 Environmental implications</b>	<b>99</b>
<b>13 Planning</b>	<b>101</b>
<b>14 Conclusions and future development</b>	<b>106</b>
<b>Bibliography</b>	<b>111</b>

## List of Figures

1.1 Optimized truss structures proposed by A.G.M. Michell (Bendsoe & Sigmund, 2003) . . . . .	3
1.2 Representation of a topology optimization problem based on material distribution. . . . .	4
1.3 Homogenization Design Method approach (Z.D. Ma et al, 2006) . . . . .	5
1.4 Conceptual illustration of the Level Set Method (D. Herrero, 2012) . . . . .	6
2.1 Bidimensional optimization problem subject to one equality constraint.	9
2.2 Example of convex and non-convex functions. . . . .	10
2.3 Example of convex and non-convex sets. . . . .	10
2.4 Procedure of the feasible direction method. . . . .	13
2.5 Diagram of the projection of $\nabla f(\mathbf{x}^k)$ onto $W^k$ in case of linear constraints. . . . .	15
2.6 Geometric interpretation of duality in $\mathbf{z}$ - $\mathbf{y}$ space. . . . .	21
2.7 Geometric interpretation of the dual gap between primal and dual problems. . . . .	22
2.8 Representation of penalty and barrier methods in one-dimensional case. . . . .	25
3.1 General 2D/3D domain with body forces and boundary conditions. .	27
3.2 Discretized domain with body and surface forces and boundary conditions. . . . .	33

3.3	Shape function $N_i$ in a bidimensional domain. . . . .	34
3.4	Shape function corresponding to local node 3 ( $N_3$ ) of a linear triangular element. . . . .	37
3.5	Table of connectivities of a general element. nel: number of elements, nodel: number of nodes per element. . . . .	38
3.6	Bilinear isoparametric mapping of a Q4 element. . . . .	39
3.7	Shape function corresponding to local node 3 ( $N_3$ ) of a bilinear quadrilateral element. . . . .	39
4.1	Example of sizing optimization problem. The design variables are the cross-section areas of each bar (Christensen & Klarbring, 2009). . . . .	42
4.2	Example of shape optimization. The design variables are the contours of the domain. Ref. <i>Optishape-TS, Quint Co.</i> . . . . .	43
4.3	Topology optimization problem. Given an initial ground structure, find the best material distribution to sustain given loads (Christensen & Klarbring, 2009). . . . .	43
4.4	Example of a minimum compliance design of a half MBB beam. Volume restriction: 50 %. . . . .	45
4.5	SIMP penalty function based on a power law. . . . .	46
4.6	Optimization of an MBB beam using SIMP method. On the left no penalization ( $p=1$ ) has been used; on the right, the same problem is solved applying a power of $p=3$ . . . . .	47
4.7	Approximation used in MMA (red) of a one dimensional case. In this case the lower asymptote $L^k$ is active due to the sign of the gradient of the original function (blue). . . . .	51
4.8	Example of the checkerboard pattern using SIMP method. . . . .	54
4.9	Representation of the enclosed elements by the sensitivity filter. . . . .	54
4.10	Example of mesh-dependency instability. a. Ground structure, b. solution using a 600-element mesh, c. solution using a 5400-element mesh (Sigmund & Peterson, 1998). . . . .	55
4.11	Continuation method applied to SIMP penalty factor $p$ . From a given iteration $k^*$ , the penalty factor will increase $\Delta p$ at each $k$ until a limit of $p$ . . . . .	56
4.12	General algorithm to solve a topology optimization problem. . . . .	58

5.1	Diagram of mesh generation using an external pre and post-processing software (GiD). . . . .	60
5.2	Diagram of post-processing using GiD. . . . .	61
6.1	Required input parameters. . . . .	63
6.2	Diagram of the global structural topology optimization code. Blue color: parameters only used in O.C./ Green color: parameters only used in MMA / Black color: parameters used in both. . . . .	65
7.1	Problem definition with reference domain (left) and analytical optimal solution (right) of an MBB beam (Rozvany, 2009). . . . .	67
7.2	MBB benchmark using a continuation method without sensitivity filter. . . . .	67
7.3	MBB benchmark using a continuation method and sensitivity filter $R_{min} = 2$ . . . . .	68
7.4	MBB benchmark using a fixed value of the penalty factor $p=3$ and sensitivity filter $R_{min} = 2$ . . . . .	68
7.5	Final design of the MBB Beam(150x50), $R_{min} = 2$ and using a continuation method. Compliance = 210.86460. . . . .	70
7.6	MBB benchmark using MMA and a continuation method without sensitivity filter. . . . .	70
7.7	MBB benchmark using MMA and a continuation method and sensitivity filter $R_{min} = 2$ . . . . .	71
7.8	$\nabla f_o \cdot (\rho_{k+1} - \rho_k)$ and compliance vs iteration number plot of MBB Beam using MMA. . . . .	71
7.9	Problem definition with reference domain (left) and analytical optimal solution (right) of a long cantilever beam (Rozvany, 1998). . . . .	72
7.10	Long cantilever beam problem (75x25) using OC method, a continuation approach and a filter radius $R_{min} = 1.5$ . . . . .	73
7.11	Long cantilever beam problem (75x25) using MMA method, a continuation approach and a filter radius $R_{min} = 1.5$ . . . . .	73
7.12	Convergence (tolerance) of MBB (30x10), $\rho_{0,e} = 0.5$ , filter radius $R_{min} = 2$ and $p = 3$ . . . . .	75
7.13	Convergence (compliance) of MBB (30x10), $\rho_{0,e} = 0.5$ , filter radius $R_{min} = 2$ and $p = 3$ . . . . .	75

7.14 Convergence (tolerance) of MBB (30x10), $\rho_{0,e} = 0.5$ , no filter radius and $p = 3$ . . . . .	76
7.15 Convergence (compliance) of MBB (30x10), $\rho_{0,e} = 0.5$ , no filter radius and $p = 3$ . . . . .	77
8.1 Optimization process of an A380 leading edge droop nose rib (Krog et al, 2002). . . . .	82
8.2 Topology optimization of an aircraft landing gear by Altair OptiStruct. . . . .	82
8.3 Topology optimization of the rear part of an A350 fuselage (G. Schuhmacher, 2008). . . . .	82
8.4 Structural parts of an aircraft wing (from <a href="http://www.zamandayolculuk.com">http://www.zamandayolculuk.com</a> ). . . . .	83
8.5 Common wing rib structural models of small/medium propelled aircraft. (a)Truss type wing rib structure. (b)Truss type wing rib structure with a continuous gusset. (c) Wing rib structure with circular holes and gusset (Ref. [12]). . . . .	84
9.1 Representation of the NACA 2412 airfoil. . . . .	85
9.2 Model of the NACA 2412 airfoil to be optimized. (1) Leading edge rib, (2) Wing box rib, (3) Trailing edge rib. . . . .	86
9.3 Pressure coefficient distribution along NACA 2412 chord at an AoA=5°. . . . .	87
9.4 Pressure coefficient distribution along NACA 2412 chord at an AoA=18°. . . . .	88
10.1 Optimized wing box design at an AoA = 5°. . . . .	90
10.2 Optimized wing box design at an AoA = 18°. . . . .	90
10.3 Final wing box design at an AoA = 18°. . . . .	91
10.4 Optimized leading edge design at an AoA = 18° and Rmin = 0.008. . . . .	92
10.5 Final leading edge design at an AoA = 18° and Rmin = 0.008. . . . .	92
10.6 Optimal trailing edge design at an AoA = 18° with Rmin = 0.005. . . . .	93
10.7 Optimal trailing edge design at an AoA = 18° with Rmin = 0.005 and clamped ends. . . . .	94
10.8 Final trailing edge design at an AoA = 18° and Rmin = 0.005. . . . .	94
10.9 Final wing rib design. . . . .	96
13.1 Main parts of the project. . . . .	101

13.2 Software development tree. . . . .	101
13.3 Aeronautical application tree. . . . .	102
13.4 Project management tree. . . . .	102
13.5 Theoretical formulation tree. . . . .	103
13.6 Gantt Chart of the project. . . . .	105
14.1 General design process with structural optimization. . . . .	107

**List of Tables**

7.1	Benchmark global input parameters. . . . .	66
7.2	Minimum compliance obtained by continuation and a fixed penalty factor p=3. . . . .	69
7.3	Convergence of MBB Beam (30x10) using OC method. . . . .	74
7.4	Convergence of MBB Beam (30x10) using MMA. . . . .	76
9.1	Geometrical parameters of NACA 2412 airfoil. . . . .	85
9.2	Mesh Data of the NACA 2412 model. . . . .	86
10.1	Wing rib global input parameters. . . . .	89
11.1	Total cost. . . . .	98
13.1	Task Breakdown of the project . . . . .	104

## Nomenclature

$\mathbf{A}_{e=1}^{nel}$	Assembly operator
$\Gamma_D$	Dirichlet boundary conditions set
$\Gamma_N$	Neumann boundary conditions set
$\lambda, \mu$	Lagrange multipliers
$\phi(\lambda, \mu)$	Dual function
$\rho$	Fictitious density distribution vector
$\ell(\cdot)$	Linear operator
$\epsilon_{ij}$	Strain tensor
$B(\mathbf{x})$	Barrier function (Barrier methods)
$c$	Penalty parameter (Penalty methods)
$C_{ijkl}$	Elasticity tensor
$E$	Subset of equality constraints
$I$	Subset of inequality constraints
$P(\mathbf{x})$	Penalty function (Penalty methods)
$P[\cdot]$	Projection matrix operator
$U$	Strain energy
$V$	Potential energy of external loads

$\kappa$	Barrier parameter (Barrier methods)
$\mathbb{D}$	Dual problem
$\mathbb{P}$	Primal problem
$\mathcal{L}$	Lagrangian function
$\nabla$	Gradient operator
$\sigma_{ij}$	Stress tensor
$\mathbf{b}$	Body force vector
$\mathbf{d}^k$	Feasible direction vector
$\mathbf{g}$	Constrained displacement vector
$\mathbf{h}$	Surface force or traction vector
$f_0(\mathbf{x})$	Objective or cost function
$g(\mathbf{x})$	Inequality constraint function
$h(\mathbf{x})$	Equality constraint function
$V_f$	Volume fraction
$W^k$	Working set at iteration k
$a(\cdot, \cdot)$	Bilinear operator
AoA	Angle of attack
FEM	Finite Element Method
ISA	International Standard Atmosphere
MMA	Method of Moving Asymptotes
OC	Optimality Criteria Method
$p$	SIMP penalty factor
$R_{\min}$	Sensitivity filter radius
SCP	Sequentially Convex Programming
SIMP	Solid Isotropic Material with Penalization
SO	Structural Optimization
STO	Structural Topology Optimization

# 1

## Introduction

### 1.1 Aim

The aim of the project is to study different numerical techniques and develop a numerical tool for structural optimization. The resultant tool will be used to optimize an aeronautical structure.

### 1.2 Scope

This section states the contents and boundaries of the project, areas to be discussed and depth of study.

- This project will focus on designing a software tool for structural optimization.
- Only solid and linear materials will be taken into account.
- Only structural topology optimization will be discussed.
- The numerical techniques will be applied to aeronautical field.
- No experimental tests will be performed.
- The numerical techniques will be applied to solve non-linear and non-convex problems.
- There will be a part of theoretical development, implementation and applications.
- The code will be written in MATLAB programming language.
- An external pre and post processor will be used in order to mesh and show results.

## 1.3 Requirements

The following requirements of the project are to be fulfilled:

- Development of a software tool capable of solving structural optimization problems.
- The tool must be accurate and fast enough to solve different reference test examples successfully.
- The code is required to be robust such that all allowed input parameters are handled by the code and result in a valid state.
- The code must be implemented in a programming environment such that modifications and extensions can be done by other developers.
- The tool must be developed modularly.
- The project is required to be delivered by June of 2014.

## 1.4 Justification

Structural optimization (S.O.) has become a popular tool in engineering companies due to the fact that speeds up the decision making process and leads to an overall cost reduction of structural engineering projects.

Optimization means making one thing the best of a bunch of possibilities. This fact is crucial in an environment in which companies seek for convincing customers that their products are the best.

In recent years, structural optimization tools have been increasingly accepted in automotive, aerospace or other related industries for structural design processes. One of the main advantages that make S.O. an essential tool nowadays is the fact that it has redefined the concept design phase of structural components.

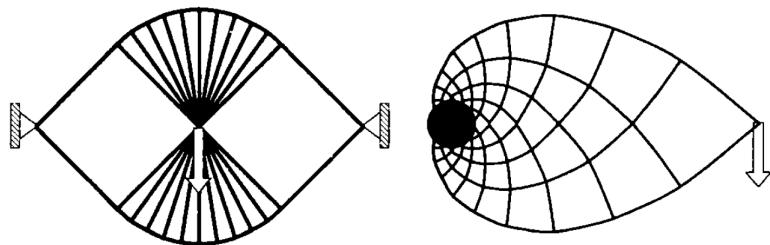
Ref. [1] points out this fact: "in the concept phase of a design process, the freedom of the designer is limited only by the specifications of the design. Today, the decision on how a new design should look is based largely upon a benchmark design or on previous designs. The decision making is based on the experience of those involved in the design process. Structural optimization can be introduced to enhance the process. The concept can be based on results of a computational optimization rather than on estimations".

The above statement confirms that S.O. tools are efficient in the way that without them experience of engineers and results based on previous projects are needed. In early design phases S.O. tools would cause huge savings in costs and time.

In aerospace industry it is important to optimize structures due to the fact that weight is one of the main factors that influences entire projects. A weight reduction implies savings in material costs, less fuel consumption and less environmental damage. These are the reasons so that S.O. tools are becoming popular and increasingly supported by aerospace industry.

## 1.5 State of the Art

Structural optimization may have origin back to 1904 by an Australian inventor and mechanical engineer named Anthony G.M. Michell. It is widely considered that he was the first person to publish a paper on structural optimization. He derived the optimality criteria for the least weight layout problem and some optimized structures that he proposed are known as *Michell structures*.



**Figure 1.1:** Optimized truss structures proposed by A.G.M. Michell (Bendsoe & Sigmund, 2003)

Some 60 years later, in 1960, L.A. Schmit introduced the idea of coupling both finite element methods and mathematical programming algorithms in order to obtain optimum designs for structural systems. He is considered as the precursor of the modern formulation of structural optimization. Although Schmit's contributions were revolutionary, the scientific community did not embrace them due to limitations in computational performance, only a few design variables could be handled. This method was restricted mostly to sizing problems of frame structures.

Around 1970s, a new optimization approach was presented by Prager et al (analytically) and Venkayya et al (numerically) usually known as *optimality criteria method*. This approach divided the professional community in supporters of this method and supporters of mathematical programming methods.

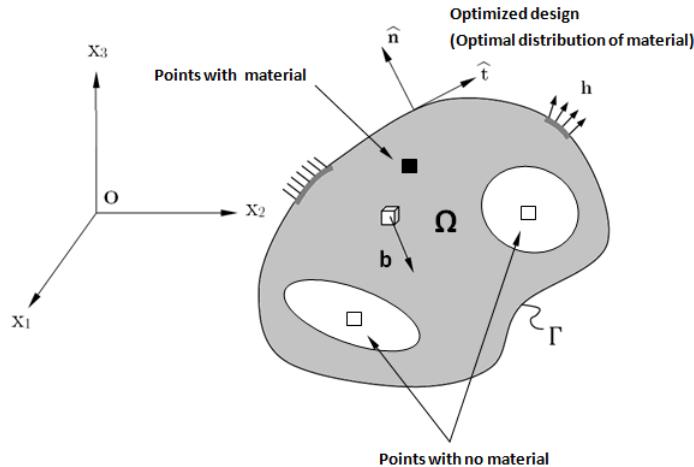
Zienkiewicz and Campbell solved in 1973 a type of structural optimization problems based on optimizing the shape of the structure. The main complication of shape problems arises from the fact that the geometry of a structure is the design variable.

In 1976, Rozvany and Prager presented the first general theory of topology optimization termed as *optimal layout theory*. This model had important contributions to numerical methods and continuum-type structures.

It was not until 1990s, since the publication of Bendsøe and Kikuchi (1998) that numerical structural optimization methods have been extensively investigated. This publication introduced a new structural optimization approach, called homogenization method, based on the microstructure of the material. This method was stated in terms of minimum compliance (maximum stiffness) approach.

Nowadays there exist four main groups related to structural topology optimization: SIMP method, Homogenization Design Method, ESO approach and the Level Set Method.

The first two methods are based on continuum material distribution approach, they find the best material layout of a domain subject to loads and boundary conditions by means of alternating solid and void regions of material (Fig. 1.2). The original problem is integer (material or no material) and these methods formulate a continuum approach in order to find the optimal solution.



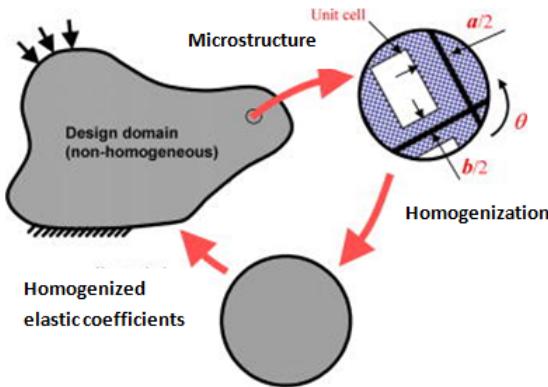
**Figure 1.2:** Representation of a topology optimization problem based on material distribution.

SIMP (**S**olid **I**sotropic **Material with **P**enalization) method was proposed by Bendsøe in 1990s in order to solve the minimum compliance design problem.**

It is based on finding the best material distribution (density) of a reference domain in order to obtain the stiffest structure with a prescribed volume. This model considers a continuum distribution of material and then a penalization is added in order to get 0-1 designs (material or no material). There exist another approach that minimizes the mass of the structure subject to a stress limitation.

The minimum compliance problem using the SIMP model can be solved by any mathematical programming algorithm but it is commonly solved by the *Optimality Criteria method* (Bendsøe & Sigmund, 2003) or by the *Method of Moving Asymptotes* (Svanberg, 1987).

While SIMP considers each element as a particle of homogeneous and isotropic material, the Homogenization Design Method (HDM) takes into account the microstructure and each particle is divided in cells that have microstructure information. Then, the material is homogenized with the low scale information. This fact makes the HDM suitable for optimizing composite materials.



**Figure 1.3:** Homogenization Design Method approach (Z.D. Ma et al, 2006)

Therefore, the SIMP method is a macroscale method while HDM takes care of the microscale.

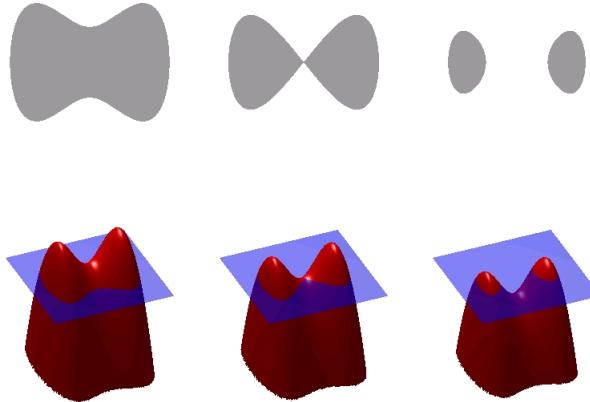
The third method for structural topology optimization, ESO (**E**volutionary **S**tructural **O**ptimization) was originally proposed in the early 1990s by professors M.Xie and G.Steven. It is based on gradually removing material that is considered less important to the structure with respect some parameter value or "criterion function". In this manner, a criterion function is calculated for each element, and at each iteration some elements with the lowest criterion function value are removed (Rozvany, 2008).

There exist a popular variation of the ESO approach called BESO (Bi-directional ESO, Yang et al. 1998) which follows the same principle of ESO but new material can be reintroduced to the structure.

The last main group is the Level Set Method (LSM), introduced in the late 1980's by S. Osher and J.A. Sethian. This method was originally used in computational fluid dynamics and image processing fields, but in the last years has become popular in topology optimization of structures.

LSM is a numerical technique for tracking shapes or boundaries. This method applied to structural topology optimization has some similarities with the formulation of the SIMP method. Both methods are applied to a FE-discretization and try to find which elements are solid or void in the optimal structure.

The advantage of the Level Set Method is that one can apply numerical methods involving curves and surfaces on a fixed cartesian grid without parametrization. LSM is usually combined with a mathematical tool called topological derivative.



**Figure 1.4:** Conceptual illustration of the Level Set Method (D. Herrero, 2012)

Topology optimization has become very popular in the recent years and its applications are gradually increasing. Due to its success, commercial FEA software tools have incorporated structural optimization algorithms in their codes. Some examples are: MSC/Nastran, OptiStruct by Altair, Ansys with the ATOM extension, TOSCA software commonly implemented in Abaqus, GENESIS, COMSOL...

In the last fifteen years, structural topology optimization has evolved from a method in development to a fundamental tool in the design process of a wide range of industrial projects.

## **Part I**

# **Theoretical formulation**

# 2

## Optimization Methods

### 2.1 Introduction to constrained optimization

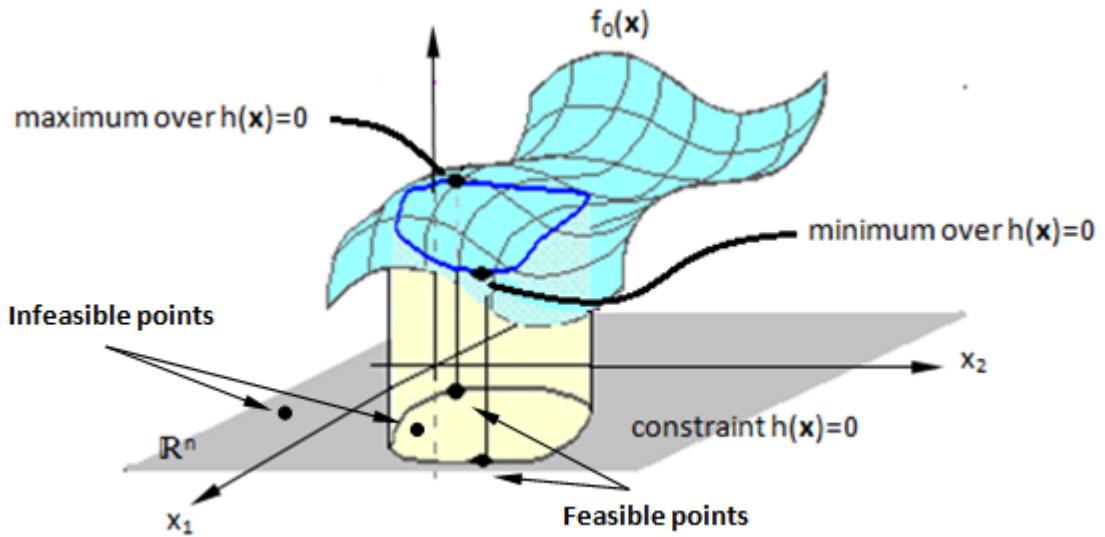
In mathematical programming, constrained optimization is the methodology to extremize (maximize or minimize) a target or objective function, with respect to some variables which are related by means of the so called constraint functions. These constraint functions are usually separated into two groups: equality constraints and inequality constraints.

Equality constraints are those which satisfy  $h(\mathbf{x}) = 0$ ,  $\mathbf{x} \in \mathbb{R}^n$  and inequality constraints satisfy  $g(\mathbf{x}) \leq 0$ ,  $\mathbf{x} \in \mathbb{R}^n$ . Sometimes inequality constraints functions that are constant and define a limit on the variables are taken apart, called box constraints.

Mathematically, a general constrained optimization problem can be defined as follows,

$$\begin{aligned}
 & \underset{x}{\text{minimize}} && f_0(\mathbf{x}) \\
 & \text{subject to} && h_i(\mathbf{x}) = 0 \quad i = 1, \dots, m \\
 & && g_j(\mathbf{x}) \leq 0, \quad j = 1, \dots, l \\
 & && \mathbf{x}_{min} \leq \mathbf{x} \leq \mathbf{x}_{max}
 \end{aligned} \tag{2.1}$$

where  $m$  and  $l$  are the number of equality and inequality constraints respectively.



**Figure 2.1:** Bidimensional optimization problem subject to one equality constraint.

A point  $\mathbf{x}^* \in \mathbb{R}^n$  is said to be feasible if it satisfies all constraints and infeasible otherwise (Fig. 2.1). The set of all feasible points is called feasible region.

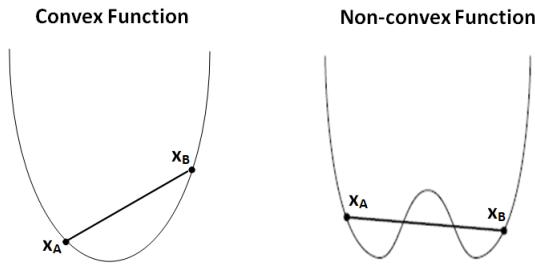
### 2.1.1 Convex and non-convex optimization

Convex behaviour in optimization problems is a very important property because the local minimum of the problem is also the global minimum. Thus, once a stationary point is reached, then the global minimum is obtained.

A constrained optimization problem (Eq. 2.1) is said to be convex if:

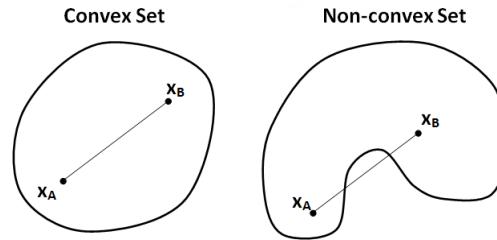
- The objective function  $f_0$  is convex
- The set of inequality constraints  $g_j(\mathbf{x}) \leq 0, j = 1, \dots, l$  is convex
- The set of equality constraints  $h_i(\mathbf{x}) = 0 i = 1, \dots, m$  is affine

A function is convex if the line segment between any two points on the graph of the function lies above the graph (Fig. 2.2).



**Figure 2.2:** Example of convex and non-convex functions.

A set is convex if for every pair of points within the set, every point on the straight line segment that joins the pair of points is also within the set (Fig. 2.3).



**Figure 2.3:** Example of convex and non-convex sets.

While in convex problems the local minimum is also the global one, the global minimum is not guaranteed if reached a stationary point in non-convex problems.

### 2.1.2 Necessary conditions for optimality

In order to obtain an optimal solution for a general nonlinear constrained problem (Eq. 2.1) some conditions must be satisfied. Assuming that the problem satisfies some regularly conditions, the first order necessary conditions for optimality are called Karush-Kuhn-Tucker conditions (KKT). It is important to remark that KKT conditions are a generalization of the Lagrange multipliers method, where inequality constraints are also taken into account.

#### Theorem 2.1.1 : Karush-Kuhn-Tucker Theorem

Let  $\mathbf{x}^*$  be a optimal solution of the problem (2.1) and assume that  $f_0(\mathbf{x})$ ,  $h_i(\mathbf{x})$  ( $i = 1, \dots, m$ ) and  $g_j(\mathbf{x})$  ( $j = 1, \dots, l$ ) are continuously differentiable functions and  $\mathbf{x}^*$  is a regular point. Then there exist unique Lagrange multipliers  $\lambda$  and  $\mu$  that satisfy the following conditions:

- **Stationarity**

Defining the Lagrangian as  $\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f_o(\mathbf{x}) + \sum_{i=1}^m \lambda_i \cdot h_i(\mathbf{x}) + \sum_{j=1}^l \mu_j \cdot g_j(\mathbf{x})$ , the stationarity condition states,

$$\nabla_x \mathcal{L}(\mathbf{x}^*, \boldsymbol{\lambda}, \boldsymbol{\mu}) = 0 \quad (2.2)$$

- **Primal feasibility**

$$h_i(\mathbf{x}^*) = 0 \quad i = 1, \dots, m \quad (2.3)$$

$$g_j(\mathbf{x}^*) \leq 0 \quad j = 1, \dots, l$$

- **Dual feasibility**

$$\mu_j \geq 0 \quad j = 1, \dots, l \quad (2.4)$$

- **Complementary slackness**

$$\mu_j \cdot g_j(\mathbf{x}^*) = 0 \quad j = 1, \dots, l \quad (2.5)$$

There are also second order necessary conditions for optimality that involve the second partial derivatives of the Lagrangian with respect to  $\mathbf{x}$ . For further information, see Ref. [20].

### 2.1.3 Sufficient conditions for optimality

The necessary conditions must be satisfied for optimality but they don't assure that any point which satisfies them is optimal. Except for the convex problems, KKT are not sufficient conditions.

Sufficient conditions are difficult to verify and are stated by a set of conditions called constraint qualifications. These conditions impose certain restrictions on the constraint functions in order to guarantee optimality.

Reference [14] describes in detail the necessary and sufficient conditions for non-linear constrained optimization problems.

Typically, numerical algorithms try to find points that satisfy KKT conditions and the global optimum is found within these KKT points. However an important fact is that for convex problems, there is only one optimal solution. Thus, a KKT point in a convex problem is a global optimum.

The following sections deal with some common methods for finding optimal solutions of the problem described in (Eq. 2.1). There are many methodologies to solve it, ones are better in some cases but not in others and some are extensions or generalizations of unconstrained optimization methods.

The constrained optimization methods that will be presented are divided into primal methods, dual methods and penalty and barrier methods.

There exist a forth group called primal-dual methods which combines methodologies of the previous groups. This method is not explicitly described in this section but more information can be found on Ref. [20].

All methods discussed are gradient based, they take information from the gradient of the objective and/or constraints functions. Methods such as Genetic Algorithms are not treated in this project.

## 2.2 Primal Methods

Primal methods work on the original problem directly by searching through the feasible region for the optimal solution. Each point at each iteration is feasible and the objective function evaluated at these points decreases constantly.

Primal methods have three significant advantages (Luenberger & Ye, 2008): (1) if the algorithm finishes before confirming optimality, then the last current point is feasible, (2) if the algorithm generates a convergent sequence, then the terminating point is a local constrained optimal solution and (3) algorithms based on primal methods are general, do not depend on a particular problem structure.

### 2.2.1 Feasible Direction Methods

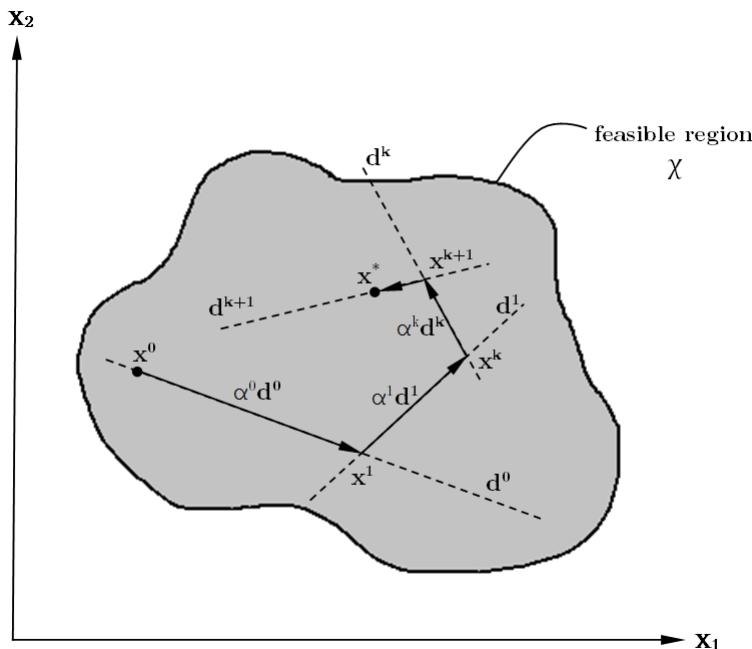
The basic idea of this method is to start from a feasible point  $\mathbf{x}^0$ , find a direction which is feasible  $\mathbf{d}^k$  and points towards the decrease of the objective function. Then, set the step length  $\alpha^k$  of the feasible direction, which determines the next iterative point  $\mathbf{x}^{k+1}$  (Eq. 2.6). This method is iterative and will continue until a stopping criterion is reached. Figure (2.4) shows the procedure of this method.

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha^k \mathbf{d}^k \quad (2.6)$$

The condition for which the feasible direction must reduce the objective function is the following,

$$\nabla f(\mathbf{x}^k)^T \mathbf{d}^k < 0 \quad (2.7)$$

which means that the feasible direction and the gradient of the objective function form an angle between  $90^\circ$  and  $270^\circ$ . Since the gradient points towards the direction of maximum rate of change, the feasible direction points towards some point such that the objective function decreases.



**Figure 2.4:** Procedure of the feasible direction method.

The general algorithm for feasible direction method is the following:

---

#### Algorithm 1 : Feasible Direction Method

---

- **Step 0:** Find an initial point  $\mathbf{x}^0$  such that  $\mathbf{x}^0 \in \chi$  ( $k=0$ )
  - **Step 1:** Find a search direction  $\mathbf{d}^k$  such that  $\mathbf{d}^k$  is feasible and decreases the objective function.
  - **Step 2:** Determine a step length  $\alpha^k$  such that  $f(\mathbf{x}^k + \alpha^k \mathbf{d}^k) < f(\mathbf{x}^k)$ ,  $\alpha^k > 0$  and  $\mathbf{x}^k + \alpha^k \mathbf{d}^k \in \chi$
  - **Step 3:** Find next feasible point  $\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha^k \mathbf{d}^k$  such that  $\mathbf{x}^{k+1} \in \chi$
  - **Step 4:** If the stopping criterion is fulfilled, stop. Otherwise  $k=k+1$  and go to Step 1.
-

Algorithms such as *Zoutendijk* or *Frank-Wolfe* solve constrained optimization problems using feasible direction methods.

### 2.2.2 Active Set Methods

This method consists of partitioning inequality constraints into two groups: those which are active and those which are inactive (they are ignored). If we consider the original problem

$$\begin{aligned} & \underset{x}{\text{minimize}} && f_0(\mathbf{x}) \\ & \text{subject to} && h_i(\mathbf{x}) = 0, i \in E \\ & && g_j(\mathbf{x}) \leq 0, j \in I \end{aligned} \tag{2.8}$$

the working set  $W^k$  is defined as the subset of constraints that are active at the current point  $\mathbf{x}^k$ .

$$W^k = E \cup \{j \in I : g_j(\mathbf{x}^k) = 0\} \tag{2.9}$$

Hence, at each iteration the algorithm moves through the surface of the current working set  $W^k$  to an improved point. The working set may change from iteration to iteration. At each point  $\mathbf{x}^k$  the original problem (2.8) with equality and inequality constraints is reduced to an equality constrained problem.

$$\begin{aligned} & \underset{x}{\text{minimize}} && f_0(\mathbf{x}^k) \\ & \text{subject to} && h_i(\mathbf{x}^k) = 0 \cup g_j(\mathbf{x}^k) \leq 0, i, j \in W^k \end{aligned} \tag{2.10}$$

For a given working set  $W^k$  suppose that the point  $\mathbf{x}^k$  solves problem (2.10), therefore it must satisfy

$$\nabla f(\mathbf{x}^k) + \sum_{j \in W^k} \lambda_j \nabla g_j(\mathbf{x}^k) = 0 \tag{2.11}$$

Solving the necessary condition one gets the Lagrange multiplier. If  $\lambda_j \geq 0, \forall j \in W^k$ , then  $\mathbf{x}^k$  is a local solution to problem (2.8); if  $\lambda_j < 0$ , then the constraint associated with the multiplier is dropped from  $W^k$ . Through iterations, if the point  $\mathbf{x}^k$  encounters a new equality constraint  $g_j(\mathbf{x}^k) \leq 0, j \notin W^k$ , then the constraint is added to the updated working set  $W^{k+1}$ .

The active set theorem assures the convergence of the equality problem (2.10) to the solution of the original problem (2.8) if (2.10) is well-defined with a unique non-degenerate solution.

For nonlinear problems the objective function and equality and inequality constraints may be approximated by a quadratic approximation and affine constraints respectively.

The basic algorithm of active set methods is the following,

---

**Algorithm 2 : Active Set Method**


---

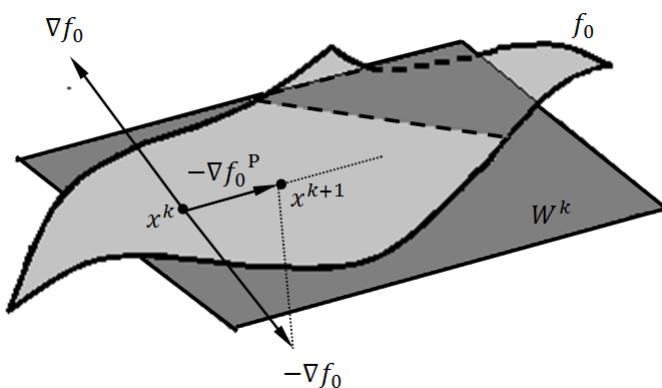
- **Step 0:** Choose a working set  $W^k$ .
  - **Step 1:** Find  $\mathbf{x}^{*k}$  and  $\lambda_j^* \forall j \in W^k$  by solving (2.10) and (2.11).
  - **Step 2:**
- ```

if  $\mathbf{x}^{*k}$  is feasible and  $\lambda_j^* \geq 0, \forall j \in W^k$  then
    local optimal solution is obtained
else Change the set  $W^k$  by adding or dropping constraints and go to Step 1.
end if

```
- 

### 2.2.3 Gradient Projection Methods

The gradient projection method is characterized by projecting the negative gradient of the objective function onto the surface generated by the active constraints (i.e. working surface) in order to define a direction of movement and find the next point (Fig. 2.5). Although this method works for nonlinear constraints, it is more effective when handling linear constraints.



**Figure 2.5:** Diagram of the projection of  $\nabla f(\mathbf{x}^k)$  onto  $W^k$  in case of linear constraints.

This method focuses on finding a search direction  $\mathbf{d}^k$  tangent to the working surface and select an appropriate step length that leads to a point  $\mathbf{x}^{k+1}$  that decreases the objective function and does not violate any constraint.

Given a  $\mathbf{x}^k$  the algorithm searches along the direction of movement  $\mathbf{d}^k$  such that

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha^k [-\nabla f(\mathbf{x}^k)]^P \quad (2.12)$$

where the superindex P denotes projection onto the working surface. So the direction of movement is  $\mathbf{d}^k = P[-\nabla f(\mathbf{x}^k)]$ , where P is the projection matrix.

The general algorithm is noted below:

---

### Algorithm 3 : Gradient Projection Method

---

- **Step 0:** Find the subspace of active constraints  $W^k$ .
  - **Step 1:** Calculate the projection matrix P and obtain  $\mathbf{d}^k$ .
  - **Step 2:** If  $\mathbf{d}^k \neq 0$ , find  $\alpha^k$  such that  $\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha^k \mathbf{d}^k$  is feasible and decreases the objective function.  
Set  $\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha^k \mathbf{d}^k$  and return to Step 0.
  - **Step 3:** If  $\mathbf{d}^{*k} = 0$ , find the lagrange multipliers  $\lambda_j^k$  associated with the active constraints.
    1. If  $\lambda_j^k \geq 0 \ \forall j \in W^k$ , stop. A local optimal solution is obtained.
    2. If  $\lambda_j^k < 0$ , drop the constraint associated to that multiplier and return to Step 1.
- 

For nonlinear constraints, the working surface may be curved and the projection of  $-\nabla f(\mathbf{x}^k)$  onto the subspace is tangent to the working surface. In order to overcome this difficulty, one may move along the projected  $-\nabla f(\mathbf{x}^k)$  to a point  $\mathbf{y}$ . Then a move perpendicular to the tangent subspace will intersect the feasible region to the next point  $\mathbf{x}^{k+1}$ . This procedure will continue until the local optimum value is reached. For a more detailed description see Refs. [20, 28].

#### 2.2.4 Reduced Gradient Methods

The reduced gradient method was formulated to solve nonlinear problems with linear constraints. However, it was extended to solve nonlinear constraints in the so-called generalized reduced gradient method. This method is characterized because the design variables are partitioned into basic and non-basic groups.

Consider a general nonlinear problem with upper and lower bounds on design variables (Eq. 2.13),

$$\begin{aligned} & \underset{x}{\text{minimize}} \quad f_0(\mathbf{x}) \\ & \text{subject to} \quad h_i(\mathbf{x}) = 0, \quad i = 1, 2, \dots, m \\ & \quad g_j(\mathbf{x}) \leq 0, \quad j = 1, 2, \dots, l \\ & \quad \alpha_q^l \leq x_q \leq \alpha_q^u, \quad q = 1, 2, \dots, n \end{aligned} \tag{2.13}$$

It is possible to transform this problem with equality and inequality constraints into a problem with only equality constraints by introducing a slack variable. The inequality constraints are written in the form  $g_j(\mathbf{x}) + x_{n+j} = 0$  such that  $x_{n+j} \geq 0$ . Therefore, the actual problem becomes,

$$\begin{aligned} & \underset{x}{\text{minimize}} \quad f_0(\mathbf{x}) \\ & \text{subject to} \quad h_j(\mathbf{x}) = 0, \quad j = 1, 2, \dots, l+m \\ & \quad \alpha_q^l \leq x_q \leq \alpha_q^u, \quad q = 1, 2, \dots, n+m \end{aligned} \tag{2.14}$$

where the lower and upper bounds are taken as 0 and a large number respectively.

The idea of this method is to split the design variables into two groups:  $\mathbf{y}$  as independent and  $\mathbf{z}$  as dependent on  $\mathbf{y}$ . Thus,  $\mathbf{x} = \begin{bmatrix} \mathbf{y} \\ \mathbf{z} \end{bmatrix}$  where  $\mathbf{y} = [y_1 \dots y_{n-l}]$  and  $\mathbf{z} = [z_1 \dots z_{m+l}]$ . Differentiating both the objective function  $f(\mathbf{y}, \mathbf{z})$  and the constraints  $h_j(\mathbf{y}, \mathbf{z})$  one gets,

$$df = \nabla_{\mathbf{y}}^T f d\mathbf{y} + \nabla_{\mathbf{z}}^T f d\mathbf{z} \tag{2.15}$$

$$dh_j = \nabla_{\mathbf{y}}^T h_j d\mathbf{y} + \nabla_{\mathbf{z}}^T h_j d\mathbf{z} \tag{2.16}$$

alternatively,

$$d\mathbf{h} = [A]d\mathbf{y} + [B]d\mathbf{z} \tag{2.17}$$

Assuming nondegeneracy and feasibility of  $\mathbf{x}$ , if  $\mathbf{x}$  is varied a small amount  $d\mathbf{x}$  then  $\mathbf{x} + d\mathbf{x}$  must be feasible. Therefore,  $d\mathbf{h}(\mathbf{x}) = 0$ . Taking  $d\mathbf{z}$  as a function of  $[A], [B]$  and  $d\mathbf{y}$  and introducing it into (Eq. 2.15) the variation of the objective function with respect to  $\mathbf{y}$  is,

$$\frac{df}{d\mathbf{y}} = \nabla_{\mathbf{y}}^T f + \nabla_{\mathbf{z}}^T f [B]^{-1} [A] = \mathbf{G}_r \tag{2.18}$$

which is called the reduced gradient.

The algorithm of generalized reduced gradient methods is the following,

---

**Algorithm 4 : Generalized Reduced Gradient Method**


---

- **Step 0:** Select the design and state variables. Start with an initial trial vector  $\mathbf{X}$  and identify  $\mathbf{Y}$  and  $\mathbf{Z}$ .
  - **Step 1:** Compute the generalized reduced gradient  $\mathbf{G}_r$ .
  - **Step 2:** Check convergence. If  $\mathbf{G}_r < \delta$ , then stop. If it is not satisfied go to next step.
  - **Step 3:** Determine the search direction  $\mathbf{d}$  and find the minimum along this direction.
  - **Step 4:** Update the vector  $\mathbf{X}$  and go to step 1 to repeat the process.
- 

## 2.3 Dual Methods

### 2.3.1 Duality theory

The term duality refers to the fact that optimization problems may be expressed in two ways: primal ( $\mathbb{P}$ ) and dual ( $\mathbb{D}$ ) viewpoints. It may be tedious to solve nonlinear problems with nonlinear constraints using primal methods directly. Dual methods can solve nonlinear problems more easily than primal methods, nevertheless some requirements are needed. Both primal and dual problems are stated as shown below,

| Primal $\mathbb{P}$                                                                                 | Dual $\mathbb{D}$                                                                                   |
|-----------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|
| $\min_{\mathbf{x} \in \mathcal{X}} [\max_{\mathbf{y} \in \mathcal{Y}} \Pi(\mathbf{x}, \mathbf{y})]$ | $\max_{\mathbf{y} \in \mathcal{Y}} [\min_{\mathbf{x} \in \mathcal{X}} \Pi(\mathbf{x}, \mathbf{y})]$ |

where  $\Pi$  is any function with two vector variables.

Consider a nonlinear problem

$$\begin{aligned}
 & \underset{x}{\text{minimize}} \quad f_0(\mathbf{x}) \\
 & \text{subject to} \quad h_i(\mathbf{x}) = 0, \quad i = 1, 2, \dots, m \\
 & \qquad \qquad \qquad g_j(\mathbf{x}) \leq 0, \quad j = 1, 2, \dots, l \\
 & \qquad \qquad \qquad \mathbf{x} \in \chi
 \end{aligned} \tag{2.19}$$

and define  $\Pi(\mathbf{x}, \mathbf{y})$  as the Lagrangian  $\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f_o(\mathbf{x}) + \sum_{i=1}^m \lambda_i h_i(\mathbf{x}) + \sum_{j=1}^l \mu_j g_j(\mathbf{x})$  where  $\mu_j \geq 0 \forall j$ .

Variables  $\mathbf{x}$  and  $(\boldsymbol{\lambda}, \boldsymbol{\mu})$  are called primal and dual variables respectively and  $\mathcal{Y} = \{\boldsymbol{\lambda}, \boldsymbol{\mu} : \mu_j \geq 0, j = 1, \dots, l\}$ .

Duality defines a minmax problem equivalent to  $\mathbb{P}$  and a maxmin equivalent to  $\mathbb{D}$ .

Thus, the primal problem becomes

$$\mathbb{P} \left\{ \begin{array}{l} p^* = \min_{\mathbf{x} \in \chi} \max_{\boldsymbol{\mu} \geq 0, \boldsymbol{\lambda}} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \\ \text{subject to} \\ h_i(\mathbf{x}) = 0, i = 1, 2, \dots, m \\ g_j(\mathbf{x}) \leq 0, j = 1, 2, \dots, l \end{array} \right. \quad (2.20)$$

where

$$\max_{\boldsymbol{\mu} \geq 0, \boldsymbol{\lambda}} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \begin{cases} f_o(\mathbf{x}) & \text{if } \mu_j \geq 0 \ \forall j \\ +\infty & \text{otherwise} \end{cases} \quad (2.21)$$

which is equivalent to (Eq. 2.19).

In other hand, the dual problem is defined as

$$\mathbb{D} \left\{ d^* = \max_{\boldsymbol{\mu} \geq 0, \boldsymbol{\lambda}} \phi(\boldsymbol{\lambda}, \boldsymbol{\mu}) \right. \quad (2.22)$$

where  $\phi(\boldsymbol{\lambda}, \boldsymbol{\mu})$  is called dual function

$$\phi(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \min_{\mathbf{x} \in \chi} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \quad (2.23)$$

Two important theorems state the relationship between the primal  $\mathbb{P}$  and the dual  $\mathbb{D}$  problems: the weak and strong duality theorems.

### Theorem 2.3.1 : Weak Duality theorem

For both  $\mathbb{P}$  and  $\mathbb{D}$  with feasible solutions, weak duality holds:  $d^* \leq p^*$

### Theorem 2.3.2 : Strong Duality theorem

Suppose in (2.19) that  $\mathbf{h}$  is regular with respect to  $\chi$  and there exist a point  $\mathbf{x} \in \chi$  which is feasible. Then, the strong duality holds:  $d^* = p^*$

These theorems show that the dual problem gives a lower bound to the primal problem and if strong duality is satisfied, then both problems are equivalent. Strong duality is not satisfied in general, but it usually holds for convex problems.

The condition that guarantee strong duality for convex problems is called a Slater's constraint qualification condition and states that convex problems  $\mathbb{P}$  and  $\mathbb{D}$  are equivalent if the problem is strictly feasible.

The difference between the optimal solutions of  $\mathbb{P}$  and  $\mathbb{D}$  when strong duality is not satisfied is called duality gap. If strong duality exists, then the duality gap is 0 and  $d^* = p^*$ .

From a practical point of view, duality results more effective when it is applied locally. Thus, global duality theory can be replaced by a restrictive but more useful local duality. This restrictions are related to convexity assumptions.

### **Theorem 2.3.3 : Local Duality theorem**

*Suppose that the problem*

$$\begin{aligned} p^* &= \min_{\mathbf{x} \in \chi} \max_{\mu \geq 0, \lambda} \mathcal{L}(\mathbf{x}, \lambda, \mu) \\ &\text{subject to} \\ h_i(\mathbf{x}) &= 0, \quad i = 1, 2, \dots, m \\ g_j(\mathbf{x}) &\leq 0, \quad j = 1, 2, \dots, l \end{aligned} \tag{2.24}$$

*has a local solution at  $\mathbf{x}^*$ , which is regular, with corresponding value  $p^*$  and a Lagrange multipliers  $\lambda^*$  and  $\mu^*$  such that  $\mu_j \geq 0, \forall j$ . Suppose also that the Hessian of the Lagrangian  $\mathcal{L}(\mathbf{x}, \lambda, \mu)$  at  $\mathbf{x}^*$  is positive definite. Then the dual problem has a local solution at  $(\lambda^*, \mu^*)$  with corresponding value  $p^*$  and  $\mathbf{x}^*$  as the corresponding point to  $(\lambda^*, \mu^*)$  in the definition of  $\phi(\lambda, \mu)$ .*

### **2.3.2 Geometric interpretation**

Lagrange multipliers have an important geometric interpretation in dual problems. In order to simplify the procedure, only inequality constraint will be considered. Consider a set  $\mathcal{G}$  defined as

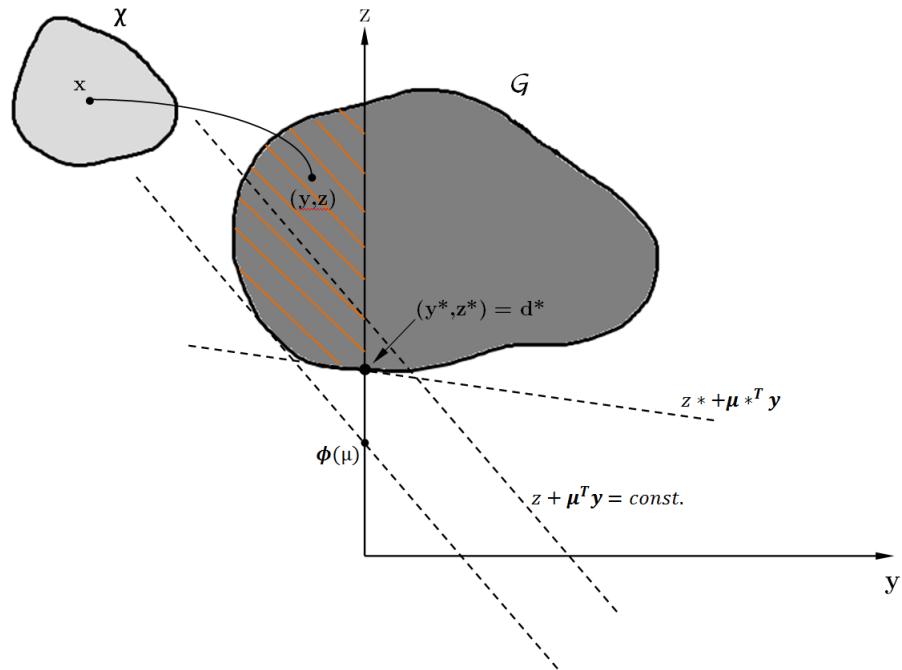
$$\mathcal{G} = \{(\mathbf{y}, z) : z = f_o(\mathbf{x}), \mathbf{y} = \mathbf{g}(\mathbf{x}) \quad \forall \mathbf{x} \in \chi\} \tag{2.25}$$

which is the mapping of  $\chi$  onto the  $z - \mathbf{y}$  space defined by the objective function and the inequality constraints. For  $\mu_j \geq 0$ , the dual function takes the form

$$\phi = z + \boldsymbol{\mu}^T \mathbf{y} \quad (2.26)$$

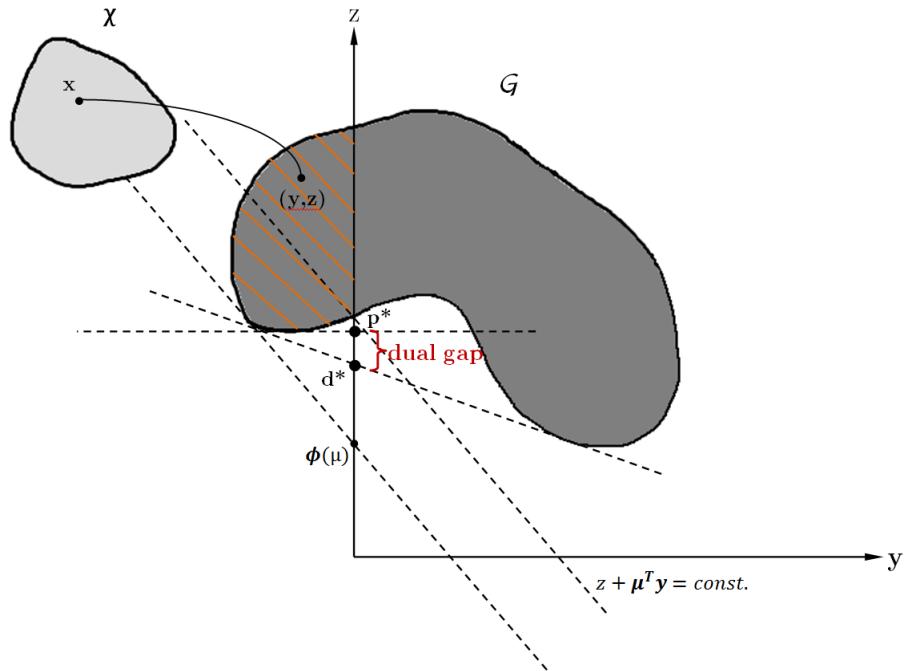
which is a straight line in  $z - \mathbf{y}$  space.

Taking into account the dual problem  $\mathbb{D}$ ,  $\phi$  minimizes  $z + \boldsymbol{\mu}^T \mathbf{y}$  over the set  $\mathcal{G}$  in  $z - \mathbf{y}$  space with respect to  $\mathbf{x}$  (i.e. moves along the ordinate axis  $z$ ). Once  $z + \boldsymbol{\mu}^T \mathbf{y}$  is minimized over  $\mathcal{G}$ , then  $\phi$  is maximized with respect to  $\boldsymbol{\mu}$  (i.e. slope of  $\phi$ ) (Fig. 2.6).



**Figure 2.6:** Geometric interpretation of duality in  $z - \mathbf{y}$  space.

Nevertheless, if strong duality theorem is not fulfilled, then the duality gap appears (Fig. 2.7).



**Figure 2.7:** Geometric interpretation of the dual gap between primal and dual problems.

## 2.4 Penalty and Barrier Methods

Penalty and Barrier methods are procedures that transform a constrained problem into an unconstrained problem.

In case of penalty methods, the transformation is done by penalizing the objective function with some parameter  $c$ , which determines the magnitude of the penalization due to the violation of some constraints. In other hand, barrier methods favours the points inside the feasible region over those near the boundary with a certain parameter  $\kappa$ .

### 2.4.1 Penalty Methods

Consider the problem

$$\begin{aligned} & \underset{x}{\text{minimize}} && f_0(\mathbf{x}) \\ & \text{subject to} && \mathbf{x} \in \mathcal{C} \end{aligned} \tag{2.27}$$

where  $\mathcal{C} = \{\mathbf{x} \in \mathbb{R}^n : g_j(\mathbf{x}) \leq 0, j = 1, \dots, l \text{ } \& \text{ } h_i(\mathbf{x}) = 0, i = 1, \dots, m\}$  and  $f_0$  is continuous in  $\mathbb{R}^n$ .

The penalty method replaces problem (2.27) to an unconstrained problem of the form,

$$\underset{x}{\text{minimize}} \quad f_0(\mathbf{x}) + cP(\mathbf{x}) \quad (2.28)$$

where  $P(\mathbf{x})$  is a penalty function and  $c$  is a positive constant parameter that holds the severity of the penalization due to the violation of constraints of the original problem.

The penalty function  $P(\mathbf{x})$  must satisfy the following conditions:

- (i)  $P(\mathbf{x})$  is continuous
- (ii)  $P(\mathbf{x}) \geq 0, \forall \mathbf{x} \in \mathbb{R}^n$
- (iii)  $P(\mathbf{x}) = 0$  if and only if  $\mathbf{x} \in \mathcal{C}$

and takes the following form,

$$P(\mathbf{x}) = \sum_{i=1}^m \Theta(h_i(\mathbf{x})) + \sum_{j=1}^l \Psi(g_j(\mathbf{x})) \quad (2.29)$$

where  $\Theta$  and  $\Psi$  are penalty functions for equality and inequality constraints respectively.

Usually  $\Theta(h_i(\mathbf{x})) = h_i(\mathbf{x})^\delta$  and  $\Psi(g_j(\mathbf{x})) = \max\{0, g_j(\mathbf{x})^\epsilon\}$ ,  $\delta = 2$  and  $\epsilon = 2$ .

The basic algorithm of the penalty method is the following,

---

#### Algorithm 5 : Penalty Method

---

- **Step 0:** Select a sequence  $\{c^k\}_{k=0}^\infty$  such that  $0 \leq c^k < c^{k+1}$  and a stopping criterion  $\epsilon$ .
  - **Step 1:** Choose an initial point  $\mathbf{x}^k$  and set  $k$  to 0.
  - **Step 2:** Defining  $q(c^k, \mathbf{x}^k) = f_0(\mathbf{x}^k) + c^k P(\mathbf{x}^k)$ .
   
  **while**  $|q(c^k, \mathbf{x}^k) - f_0(\mathbf{x}^k)| > \epsilon$  **do**
    - $\mathbf{x}^{k+1} = \text{argmin} [q(c^k, \mathbf{x})]$
    - $k = k+1$
    - end while**
  - **Step 3:** Local optimal solution  $\mathbf{x}^* = \mathbf{x}^k$
-

An important theorem that guarantees global convergence of this method states,

### **Theorem 2.4.1 : Convergence of the penalty method**

*Any limit point of the sequence  $\{\mathbf{x}\}$  generated by the penalty method is a solution to the associated original primal problem (2.27).*

### **2.4.2 Barrier Methods**

Barrier methods are also called interior point methods because they generate a barrier on the boundary of the feasible region and they avoid search algorithms from leaving the region. A disadvantage of this method is that it doesn't hold equality constraints, only inequalities.

Similarly to the penalty function methods, barrier methods solve problems of the form

$$\begin{aligned} & \underset{x}{\text{minimize}} && f_0(\mathbf{x}) \\ & \text{subject to} && \mathbf{x} \in \mathcal{B} \end{aligned} \tag{2.30}$$

where  $\mathcal{B} = \{\mathbf{x} \in \mathbb{R}^n : g_j(\mathbf{x}) \leq 0, j = 1, \dots, l\} \neq \emptyset$ .

The approximate unconstrained problem is

$$\begin{aligned} & \underset{x}{\text{minimize}} && f_0(\mathbf{x}) + \kappa B(\mathbf{x}) \\ & \text{subject to} && \mathbf{x} \in \text{interior } \mathcal{B} \end{aligned} \tag{2.31}$$

In this case,  $\kappa \geq 0$  is the barrier parameter and  $B(\mathbf{x})$  the barrier function.  $B$  is defined in the interior of the feasible region such that

- (i)  $B(\mathbf{x})$  is continuous
- (ii)  $B(\mathbf{x}) \geq 0$
- (iii)  $B(\mathbf{x}) \rightarrow \infty$  as  $\mathbf{x}$  approaches the boundary of the feasible region

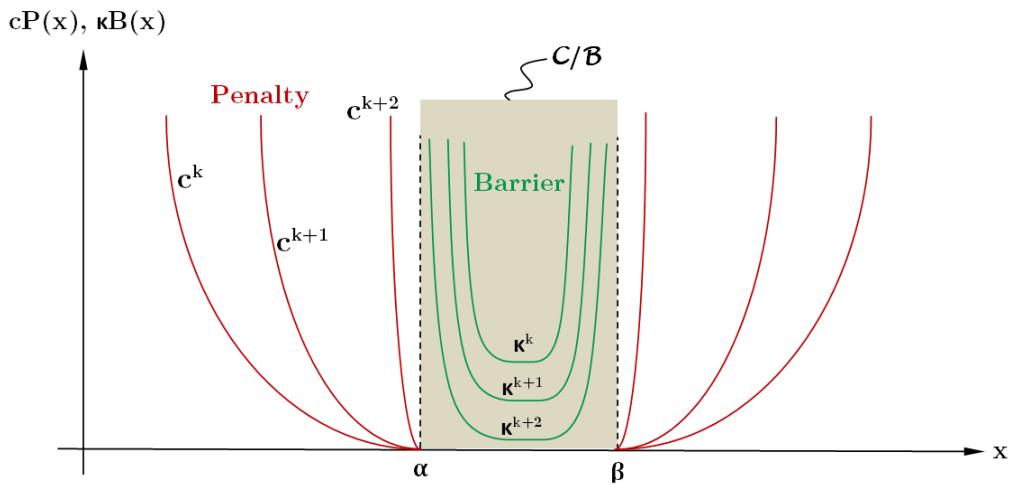
The following barrier functions are the most common,

$$B(\mathbf{x}) = - \sum_{j=1}^l \frac{1}{g_j(\mathbf{x})} \tag{2.32}$$

$$B(\mathbf{x}) = - \sum_{j=1}^l \log[-g_j(\mathbf{x})] \tag{2.33}$$

The algorithm for this method is analogous to the penalty method (algorithm (5)) but the parameter  $\kappa$  decreases ( $\kappa^k > \kappa^{k+1} \geq 0$ ). The same theorem (2.4.1) can be applied for convergence of the barrier method.

Figure (2.8) illustrates the comparison between the two methods in one-dimensional case. In case of the penalty method, as  $c \rightarrow \infty$  the solution of the unconstrained problem tends to the original constrained problem. Similarly in the barrier method, as  $\kappa \rightarrow 0$  the boundary approaches the original feasible region and the original problem and its approximation by the barrier method tend to be equivalent.



**Figure 2.8:** Representation of penalty and barrier methods in one-dimensional case.

# 3

## Finite Element Method in linear elastostatics

The Finite Element Method (FEM) is a numerical method applied to boundary value problems in order to find an approximate solution for the governing partial differential equations by discretizing a domain with boundary conditions.

The discretization of the domain is generated by a mesh formed by multiple nodes connected among them which form finite elements. Once the mesh is implemented and boundary conditions defined, FEM solves the discretized problem and gives the solution at every node by using interpolating functions.

This method is widely used in mechanical engineering problems, but also is suitable to solve thermal, fluid and electromagnetic problems.

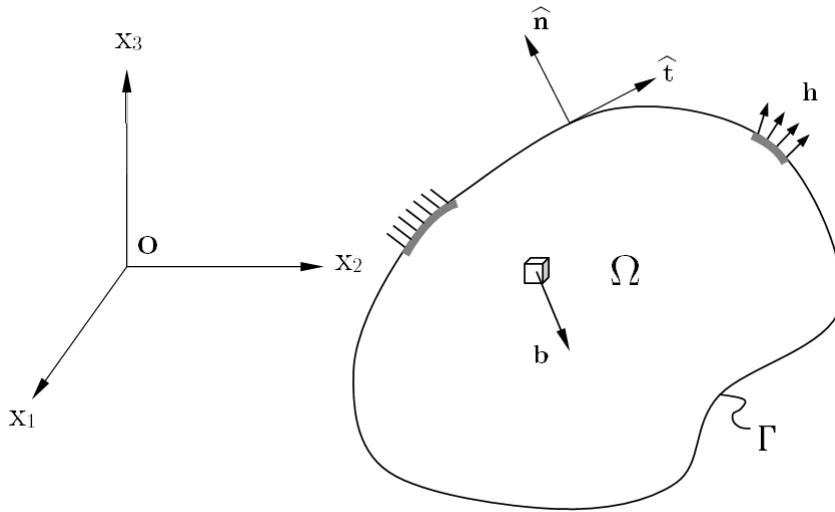
This section focuses on the basic formulation of the Finite Element Method applied to linear elasticity. First, a continuum formulation of the problem will be discussed and then the continuum domain will be discretized by using finite element techniques.

The general development of FEM can be found on Refs. [3, 15, 35].

### 3.1 Continuum formulation

#### 3.1.1 General formulation

Consider a domain with body and surface forces as well as boundary conditions (Fig. 3.1),



**Figure 3.1:** General 2D/3D domain with body forces and boundary conditions.

This domain represents an elastic body subject to certain boundary conditions and body  $\mathbf{b} = \{b_i\}_{i=1}^{nd}$  and surface  $\mathbf{h} = \{h_i\}_{i=1}^{nd}$ <sup>1</sup> forces. The boundary is divided into two subdomains  $\Gamma = \{\Gamma_D, \Gamma_N\}$ : Dirichlet  $\Gamma_D$  and Neumann  $\Gamma_N$  boundary conditions. Dirichlet conditions correspond to prescribed displacements and Neumann conditions to prescribed tractions, both on  $\Gamma$ .

Let  $\mathbf{u}(\mathbf{x}) = \{u_i\}_{i=1}^{nd}$  be the displacement vector field and assume small deformations (Eq. 3.1) and the constitutive equation for a linear elastic material (Eq. 3.2).

$$\epsilon_{ij} = \frac{1}{2} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad (3.1)$$

$$\sigma_{ij} = C_{ijkl} \cdot \epsilon_{kl} \quad (3.2)$$

Given  $b_i, g_i : \Gamma_{Di} \rightarrow \mathbb{R}$  and  $h_i : \Gamma_{Ni} \rightarrow \mathbb{R}$ , the problem is to find  $u_i : \Omega \rightarrow \mathbb{R}$  such that,

$$\begin{aligned} \sigma_{ij,j} + b_i &= 0 \text{ in } \Omega \\ u_i &= g_i \text{ on } \Gamma_{Di} \\ \sigma_{ij} \cdot n_j &= h_i \text{ on } \Gamma_{Ni} \end{aligned} \quad (3.3)$$

which are the elastostatic equilibrium equations.

<sup>1</sup>The index nd (number of dimensions) is nd=2 in 2D and nd=3 in 3D.

This expression is also called strong form.

### 3.1.2 Principles of minimum potential energy and virtual displacements

The principle of minimum total potential energy can be stated as: "*Of all kinematic admissible displacements satisfying compatibility and boundary conditions, those that satisfy the equilibrium equations extremize the total potential energy. If the extremum corresponds to a minimum, then the equilibrium state is stable*".

Let  $\Pi$  be the total potential energy of a solid body defined as the sum of the strain energy  $U$  and the potential of external loads  $V$ ,

$$\Pi = U + V \quad (3.4)$$

For linear elastic materials, the strain energy per unit volume is given by,

$$U = \frac{1}{2} \int_{\Omega} \sigma_{ij} \epsilon_{kl} d\Omega \quad (3.5)$$

and the potential of external loads is,

$$V = - \int_{\Omega} b_i u_i d\Omega - \int_{\Gamma} h_i u_i d\Gamma \quad (3.6)$$

In order to find the extremum of the total potential energy  $\Pi$ , variational calculus principles are applied,

$$\delta\Pi = \delta(U + V) = 0 \quad (3.7)$$

The above statement leads to (Eq. 3.8) after some calculations and applying (Eq. 3.2),

$$\delta\Pi = \int_{\Omega} \frac{1}{2} C_{ijkl} (\delta\epsilon_{kl} \epsilon_{ij} + \epsilon_{kl} \delta\epsilon_{ij}) d\Omega - \int_{\Omega} b_i \delta u_i d\Omega - \int_{\Gamma} h_i \delta u_i d\Gamma = 0 \quad (3.8)$$

Considering elastic behaviour, the tensor  $C_{ijkl}$  is symmetric ( $C_{ijkl} = C_{klij}$ ) and the following expression is obtained,

$$\int_{\Omega} C_{ijkl} \epsilon_{kl} \delta\epsilon_{ij} d\Omega = \int_{\Omega} b_i \delta u_i d\Omega + \int_{\Gamma} h_i \delta u_i d\Gamma \quad (3.9)$$

This expression (Eq. 3.9) is known as *Principle of Virtual Displacements*. This principle is the basis of the Finite Element Method, also known as weak form of (Eq. 3.3).

### 3.1.3 Matrix notation

In order to simplify the computer implementation, a matrix notation will be used instead of an index notation from this section onwards. Furthermore, it will be assumed linear elastostatic, homogeneous and isotropic behaviour.

For a three-dimensional problem, the displacement field is given by,

$$\mathbf{u}(\mathbf{x}) = \begin{pmatrix} u_1(x_1, x_2, x_3) \\ u_2(x_1, x_2, x_3) \\ u_3(x_1, x_2, x_3) \end{pmatrix} \quad (3.10)$$

The strain tensor is adapted to vector form due to symmetry,

$$\boldsymbol{\epsilon}(\mathbf{u}) = (\epsilon_{11} \ \epsilon_{22} \ \epsilon_{33} \ \gamma_{23} \ \gamma_{13} \ \gamma_{12})^T \quad (3.11)$$

where  $\epsilon_{ij} = \frac{\partial u_i}{\partial x_j}$ ,  $\gamma_{ij} = \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} = 2\epsilon_{ij}$

The constitutive equation for linear elastic isotropic materials takes the form,

$$\boldsymbol{\sigma}(\mathbf{u}) = [\mathbf{D}] \cdot \boldsymbol{\epsilon}(\mathbf{u}) \quad (3.12)$$

where

$$\boldsymbol{\sigma}(\mathbf{u}) = (\sigma_{11} \ \sigma_{22} \ \sigma_{33} \ \tau_{23} \ \tau_{13} \ \tau_{12})^T \quad (3.13)$$

and

$$[\mathbf{D}] = \begin{pmatrix} 2\mu + \lambda & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & 2\mu + \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & 2\mu + \lambda & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{pmatrix} \quad (3.14)$$

Parameters  $\mu$  and  $\lambda$  are the Lammé coefficients and are related to Young's modulus  $E$  and Poisson's coefficient  $\nu$  in the following way,

$$\mu = \frac{E}{2(1 + \nu)} \quad (3.15)$$

$$\lambda = \frac{\nu E}{(1 + \nu)(1 - 2\nu)} \quad (3.16)$$

Strains are the derivatives of displacements with respect to the position, so the strain vector is written in matrix form as,

$$\boldsymbol{\epsilon} = [\nabla]\mathbf{u} \quad (3.17)$$

where

$$[\nabla]^T = \begin{pmatrix} \frac{\partial}{\partial x} & 0 & 0 & 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \\ 0 & \frac{\partial}{\partial y} & 0 & \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \\ 0 & 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \end{pmatrix} \quad (3.18)$$

### 3.1.4 Equilibrium principles

Any elastostatic problem can be formulated in three different ways, known as equilibrium principles which are essentially equivalent. These formulations are:

- (i) Principle of potential energy minimization (PEM)
- (ii) Principle of virtual work or Weak form (PVW)
- (iii) Governing partial differential equation or Strong form (PDE)

It can be proven that each of these formulations are equivalent between them

$$\text{PEM} \Leftrightarrow \text{PVW} \Leftrightarrow \text{PDE}$$

Defining the symmetric and bilinear operator as,

$$a(\mathbf{u}, \mathbf{v}) = \int_{\Omega} \boldsymbol{\epsilon}(\mathbf{v})[\mathbf{D}]\boldsymbol{\epsilon}(\mathbf{u}) d\Omega \quad (3.19)$$

linear operators,

$$(\mathbf{v}, \mathbf{b}) = \int_{\Omega} \mathbf{v}^T \mathbf{b} \, d\Omega \quad (3.20)$$

$$(\mathbf{v}, \mathbf{h}) = \int_{\Gamma_h} \mathbf{v}^T \mathbf{h} \, d\Gamma \quad (3.21)$$

and sets,

$$\mathcal{U} = \{\mathbf{u} : \mathbf{u} \in \mathbf{H}_1(\Omega) \mid \mathbf{u} = \mathbf{g} \text{ on } \Gamma_D\}$$

$$\mathcal{V} = \{\mathbf{v} : \mathbf{v} \in \mathbf{H}_1(\Omega) \mid \mathbf{v} = \mathbf{0} \text{ on } \Gamma_D\}$$

the principle of minimum total potential energy written in matrix notation states,

---

### Principle of energy minimization (PEM)

---

Find  $\mathbf{u} \in \mathcal{U}$  such that

$$\Pi(\mathbf{u}) \leq \Pi(\mathbf{v}) \quad \forall \mathbf{v} \in \mathcal{V} \quad (3.22)$$

where

$$\Pi(\mathbf{v}) = \frac{1}{2}a(\mathbf{v}, \mathbf{v}) - (\mathbf{v}, \mathbf{b}) - (\mathbf{v}, \mathbf{h})$$


---

As stated in section 3.1.2, applying the stationarity condition to  $\Pi$ , the principle of virtual displacements (also called principle of virtual work or strong form) is obtained,

---

### Principle of virtual work (PVW)/ Weak form

---

Given  $\mathbf{b} : \Omega \rightarrow \mathbb{R}^3$ ,  $\mathbf{g} : \Gamma_D \rightarrow \mathbb{R}^3$  and  $\mathbf{h} : \Gamma_N \rightarrow \mathbb{R}^3$ , find  $\mathbf{u} \in \mathcal{U}$  such that

$$\frac{1}{2}a(\mathbf{v}, \mathbf{u}) = (\mathbf{v}, \mathbf{b}) + (\mathbf{v}, \mathbf{h}) \quad \forall \mathbf{v} \in \mathcal{V} \quad (3.23)$$


---

where  $\mathbf{u}$  and  $\mathbf{v}$  are the admissible and virtual displacement vectors respectively.

Finally, the equilibrium equation is also called strong form (partial differential equation formulation) and states (as mentioned in section 3.1.1) the following,

---

**Partial Differential Equation (PDE)/ Strong form**


---

Given  $\mathbf{b} : \Omega \rightarrow \mathbb{R}^3$ ,  $\mathbf{g} : \Gamma_D \rightarrow \mathbb{R}^3$  and  $\mathbf{h} : \Gamma_N \rightarrow \mathbb{R}^3$ , find  $\mathbf{u} : \Omega \rightarrow \mathbb{R}^3$  such that

$$\begin{aligned} [\nabla]^T \boldsymbol{\sigma} + \mathbf{b} &= \mathbf{0} \text{ in } \Omega \\ \mathbf{u} &= \mathbf{g} \text{ on } \Gamma_D \\ [\mathbf{N}] \boldsymbol{\sigma} &= \mathbf{h} \text{ on } \Gamma_N \end{aligned} \tag{3.24}$$


---

where

$$[\mathbf{N}] = \begin{pmatrix} n_1 & 0 & 0 & 0 & n_3 & n_2 \\ 0 & n_2 & 0 & n_3 & 0 & n_1 \\ 0 & 0 & n_3 & n_2 & n_1 & 0 \end{pmatrix}$$

is the normal vector written in matrix notation.

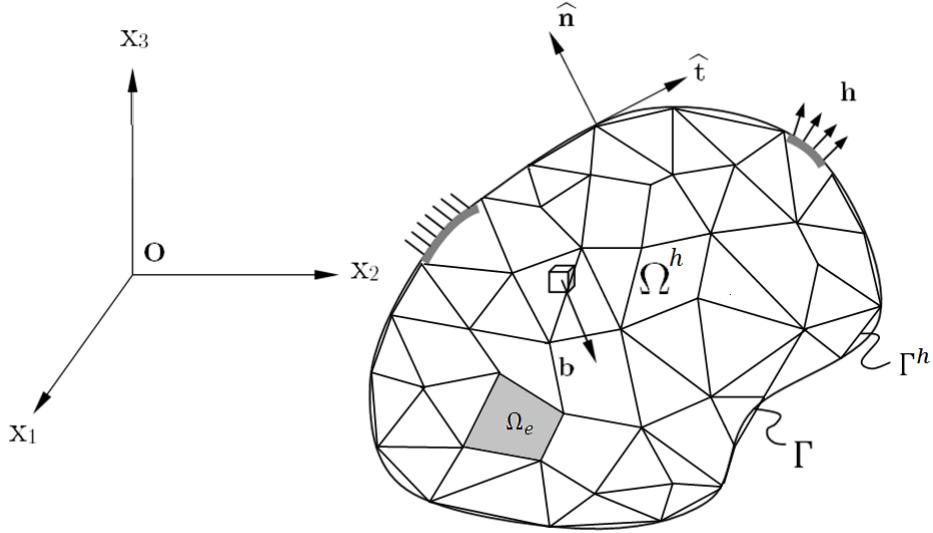
## 3.2 Finite Element discretization

As discussed before, FEM focuses on solving governing partial differential equations in the so-called weak formulation. The procedure to solve the problem is the following:

1. Approximate spaces  $\mathcal{U}$  and  $\mathcal{V}$  in discretized spaces  $\mathcal{U}^h$  and  $\mathcal{V}^h$ .
2. Reformulate the weak form problem in  $\mathcal{U}^h$  and  $\mathcal{V}^h$  discretized spaces.
3. Solve the weak form discretized problem.

In order to generate the discretized spaces, a partition of the domain  $\Omega$  into elemental domain  $\Omega_e$  is performed (Fig. 3.2).

Let  $E^h = \{\Omega_e\}_{e=1}^{nel}$  be the set formed by all elements.



**Figure 3.2:** Discretized domain with body and surface forces and boundary conditions.

The domain  $\Omega^h = \bigcup_{e=1}^{n_{el}} \Omega_e$  and discretized spaces of  $\mathcal{U}$  and  $\mathcal{V}$  are defined as

$$\mathcal{U}^h = \{\mathbf{u}^h : \mathbf{u}^h \in \mathbf{C}^0(E^h); \mathbf{u}^h \in P_n(\Omega^h); \mathbf{u}^h = \mathbf{g} \text{ on } \Gamma_D\}, \quad \mathcal{U}^h \subset \mathcal{U}$$

$$\mathcal{V}^h = \{\mathbf{v}^h : \mathbf{v}^h \in \mathbf{C}^0(E^h); \mathbf{v}^h \in P_n(\Omega^h); \mathbf{v}^h = \mathbf{0} \text{ on } \Gamma_D\}, \quad \mathcal{V}^h \subset \mathcal{V}$$

where  $P_n(\Omega^h)$  is the space of polynomials less or equal to degree  $n$  on  $\Omega^h$ .

Therefore, the discretized version of (Eq. 3.23) is,

---

### Discretization of the weak form

---

Given  $\mathbf{b} : \Omega^h \rightarrow \mathbb{R}^3$ ,  $\mathbf{g} : \Gamma_D \rightarrow \mathbb{R}^3$  and  $\mathbf{h} : \Gamma_N \rightarrow \mathbb{R}^3$ , find  $\mathbf{u}^h \in \mathcal{U}^h$  such that

$$\int_{\Omega^h} [\nabla] \mathbf{v}^{hT} [\mathbf{D}] [\nabla] \mathbf{u}^h d\Omega = \int_{\Omega^h} \mathbf{v}^{hT} \mathbf{b} d\Omega + \int_{\Gamma_h} \mathbf{v}^{hT} \mathbf{h} d\Gamma; \quad \forall \mathbf{v}^h \in \mathcal{V}^h \quad (3.25)$$


---

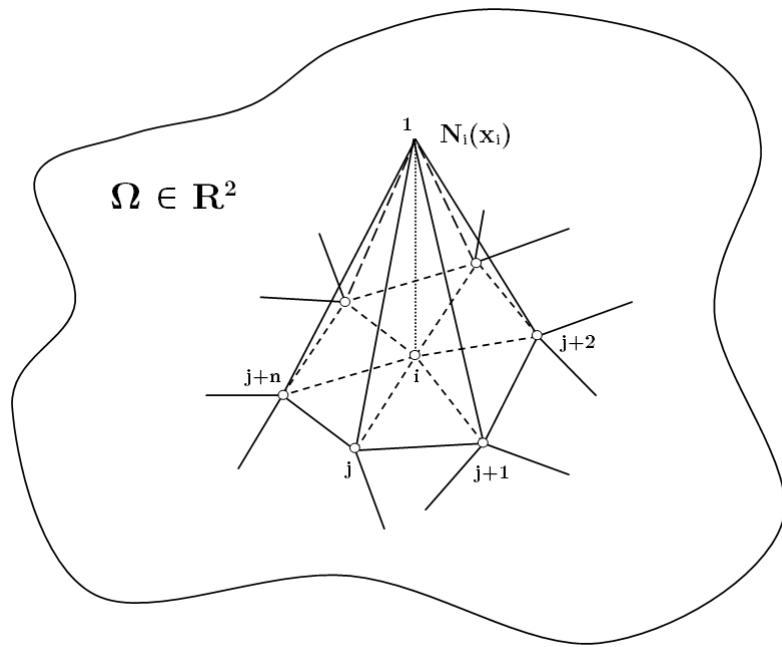
### 3.2.1 Direct Stiffness Method

The basis of FEM discretization is an approximation of the displacement field (Eq. 3.26) by using interpolating functions, also called base or shape functions, at each node of the discretized domain (Fig. 3.3).

$$u_i^h(\mathbf{x}) = \sum_{A=1}^n N_A(\mathbf{x}) d_{iA}, \quad i = 1, \dots, nd \quad (3.26)$$

where  $\{N_A\} \in \mathcal{V}^h$  are the system of piecewise global shape functions,  $d_{Ai}$  the values of nodal displacements in dimension  $i$ ,  $A$  refers to the nodal number and  $n$  refers to the total number of nodes. The main property of shape functions is,

$$N_i(\mathbf{x}_j) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \quad (3.27)$$



**Figure 3.3:** Shape function  $N_i$  in a bidimensional domain.

If prescribed displacements are taken into account,  $\mathbf{u}^h(\mathbf{x})$  may be written as the sum of free  $\bar{\mathbf{u}}^h(\mathbf{x})$  (unknowns) and restricted  $\mathbf{g}^h(\mathbf{x})$  displacements,

$$\mathbf{u}^h(\mathbf{x}) = \bar{\mathbf{u}}^h(\mathbf{x}) + \mathbf{g}^h(\mathbf{x}) \quad (3.28)$$

and using the Euclidean basis vectors  $\{\mathbf{e}_i\}_{i=1}^3$ ,

$$\bar{\mathbf{u}}^h(\mathbf{x}) = \bar{u}_i^h(\mathbf{x}) \mathbf{e}_i \quad (3.29)$$

where  $\bar{u}_i^h(\mathbf{x}) = \sum_{A \in \mathcal{F}} N_A(\mathbf{x}) d_{iA}$

$\mathcal{F}$  and  $\mathcal{C}$  denote the sets of free (not prescribed) and constrained (prescribed) nodes. Defining  $\mathcal{M}$  as the set of total nodes, it is clear that  $\mathcal{M} = \mathcal{F} \cup \mathcal{C}$ . The same is done with  $\mathbf{g}^h$  on  $\mathcal{C}$  and  $\mathbf{v}^h$  on  $\mathcal{F}$  using  $g_{iA}$  and  $c_{iA}$  as their nodal displacements respectively.

Substituting these approximations of displacement fields for  $\mathbf{u}^h(\mathbf{x})$ ,  $\mathbf{g}^h$  and  $\mathbf{v}^h$  into the discretized weak formulation, one gets

$$\sum_{A \in \mathcal{F}} \sum_{B \in \mathcal{F}} c_{iA} \ a(N_A \mathbf{e}_i, N_B \mathbf{e}_j) \ d_{jB} = \sum_{A \in \mathcal{F}} c_{iA} \ (N_A \mathbf{e}_i, \mathbf{b}) + \sum_{A \in \mathcal{F}} c_{iA} \ (N_A \mathbf{e}_i, \mathbf{h})_\Gamma - \quad (3.30)$$

$$- \sum_{A \in \mathcal{F}} \sum_{B \in \mathcal{C}} c_{iA} \ a(N_A \mathbf{e}_i, N_B \mathbf{e}_j) \ g_{jB}$$

Introducing the ID array which is an array that returns global degrees of freedom given the node number A and local degree of freedom per node i,

$$ID(A, i) = \begin{cases} P = nd(A - 1) + i & \text{if } A \in \mathcal{F} \\ 0 & \text{if } A \in \mathcal{C} \end{cases} \quad (3.31)$$

to (Eq. 3.30) for  $P = ID(A, i)$  and  $Q = ID(B, j)$ ,

$$[\mathbf{K}] \cdot \mathbf{d} = \mathbf{F} \quad (3.32)$$

$$[\mathbf{K}] = \{K_{PQ}\}, \quad K_{PQ} = \mathbf{e}_i \int_{\Omega} [\mathbf{B}_A]^T [\mathbf{D}] [\mathbf{B}_B] d\Omega \mathbf{e}_j \quad (3.33)$$

$$\mathbf{d} = \{d_Q\} \quad (3.34)$$

$$\mathbf{F} = \{F_P\} \quad (3.35)$$

$$F_P = \sum_{A \in \mathcal{F}} c_{iA} \ (N_A \mathbf{e}_i, \mathbf{b}) + \sum_{A \in \mathcal{F}} c_{iA} \ (N_A \mathbf{e}_i, \mathbf{h})_\Gamma - \sum_{A \in \mathcal{F}} \sum_{B \in \mathcal{C}} c_{iA} \ a(N_A \mathbf{e}_i, N_B \mathbf{e}_j) \ g_{jB} \quad (3.36)$$

where  $[\mathbf{B}]$ , called deformation matrix, comes from  $\epsilon(N_A \mathbf{e}_i) = [\mathbf{B}] \cdot \mathbf{e}_i$ ,

$$[\mathbf{B}] = \begin{pmatrix} \frac{\partial N_A}{\partial x_1} & 0 & 0 \\ 0 & \frac{\partial N_A}{\partial x_2} & 0 \\ 0 & 0 & \frac{\partial N_A}{\partial x_3} \\ 0 & \frac{\partial N_A}{\partial x_3} & \frac{\partial N_A}{\partial x_2} \\ \frac{\partial N_A}{\partial x_3} & 0 & \frac{\partial N_A}{\partial x_1} \\ \frac{\partial N_A}{\partial x_2} & \frac{\partial N_A}{\partial x_1} & 0 \end{pmatrix}$$

$[\mathbf{K}]$  is called the stiffness matrix,  $\mathbf{F}$  load vector and  $\mathbf{d}$  displacement vector. So, the problem is reduced to calculate the displacement vector  $\mathbf{d}$  from (Eq. 3.32). This method is called Direct Stiffness Method.

### 3.2.2 Element Stiffness Method

Instead of integrating all over the domain  $\Omega$  and boundary  $\Gamma$  to obtain  $[\mathbf{K}]$  and  $\mathbf{F}$ , it is more efficient from a computational point of view to obtain elemental stiffness matrices  $[\mathbf{K}^e]$  and load vectors  $\mathbf{F}^e$  and then assemble them into a global ones. This elemental point of view is called Element Stiffness Method.

$$[\mathbf{K}] = \bigwedge_{e=1}^{nel} [\mathbf{K}^e] \quad (3.37)$$

$$\mathbf{F} = \bigwedge_{e=1}^{nel} \mathbf{F}^e \quad (3.38)$$

where  $\bigwedge_{e=1}^{nel}$  is the assembly operator which transforms local or elemental degrees of freedom into global degrees of freedom. The elemental stiffness matrix takes the following form,

$$[\mathbf{K}^e] = \int_{\Omega_e} [\mathbf{B}^e]^T [\mathbf{D}] [\mathbf{B}^e] d\Omega \quad (3.39)$$

In this case the shape functions depend on which type of element is used: n-node triangle element, m-node quadrilateral element, tetrahedral or hexahedral elements...

The following sections deal with two basic element types: triangular elements and quadrilaterals in a bidimensional case. However, there exist more types of elements such as high-order elements, 3D elements, structural elements... all depending on the kind of application in which they are used, precision and computational efficiency.

### 3.2.2.1 Linear triangular element T3

T3 element type is a 3-node triangular geometry which uses linear shape functions. In this case, the displacements are interpolated by shape functions in the following way,

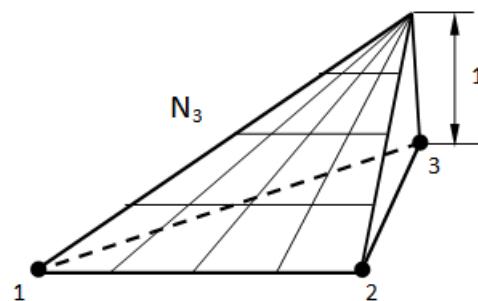
$$u_i^h(\mathbf{x}) = \sum_{a=1}^3 N_a(\mathbf{x}) d_{ia}, \quad i = 1, 2. \quad (3.40)$$

where shape functions take the form,

$$N_a(x, y) = \frac{a_a + b_a \cdot x + c_a \cdot y}{2 \cdot \Delta} \quad a = 1, 2, 3. \quad (3.41)$$

$\Delta$  = Area of the triangle 123

and a stands for local/elemental node numbers.



**Figure 3.4:** Shape function corresponding to local node 3 ( $N_3$ ) of a linear triangular element.

Each shape function is obtained by imposing that its value is 1 when  $x, y$  correspond to the local node A and 0 otherwise and then obtaining the coefficients  $a_a, b_a, c_a$ .

In this case the elemental  $[\mathbf{B}^e]$  matrix is the following,

$$[\mathbf{B}^e] = \begin{pmatrix} \frac{\partial N_1}{\partial x} & 0 & \frac{\partial N_2}{\partial x} & 0 & \frac{\partial N_3}{\partial x} & 0 \\ 0 & \frac{\partial N_1}{\partial y} & 0 & \frac{\partial N_2}{\partial y} & 0 & \frac{\partial N_3}{\partial y} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial y} & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial y} & \frac{\partial N_3}{\partial x} \end{pmatrix}$$

Since shape functions are linear and  $[\mathbf{B}^e]$  constant,  $[\mathbf{K}^e]$  can be obtained as,

$$[\mathbf{K}^e] = [\mathbf{B}^e]^T [\mathbf{D}] [\mathbf{B}^e] \cdot \Delta \quad (3.42)$$

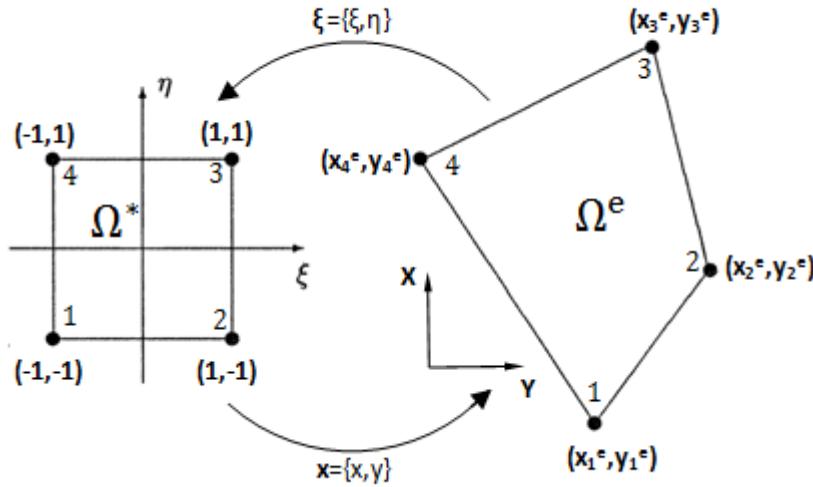
The global stiffness matrix and the global load vector are obtained through the assembly operator  $\underset{e=1}{\overset{nel}{\text{A}}}$ , that links local degrees of freedom with global ones. This relation is achieved by means of the table of connectivities (Fig. 3.5) and the ID array.

**Figure 3.5:** Table of connectivities of a general element. nel: number of elements, nodel: number of nodes per element.

### 3.2.2.2 Bilinear quadrilateral element Q4

The formulation of a linear triangle element can be extended to quadrilateral and high-order triangular elements. However, some complications arise related to the construction of shape functions. The consistency requirements for high order elements with curved boundaries becomes complicated (Felippa, 2004).

This problem can be solved using a nodal coordinate transformation in order to parametrize and thus generalize the geometry of any element. This is what the isoparametric formulation does. In case of a bilinear quadrilateral, the isoparametric representation can generalize square, rectangular or trapezoidal shapes into one parametric formulation in a space of natural coordinates  $\xi=(\xi, \eta)$ .



**Figure 3.6:** Bilinear isoparametric mapping of a Q4 element.

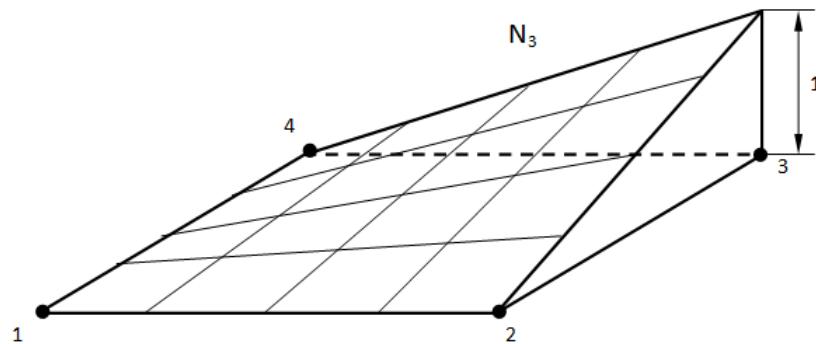
The mapping functions have the following appearance,

$$\mathbf{x}(\xi) = \sum_{a=1}^4 N_a(\xi) \cdot \mathbf{x}_a^e \quad a = 1, 2, 3, 4. \quad (3.43)$$

and assuming that  $\mathbf{x}$  is bilinear with respect to  $(\xi, \eta)$ , the shape functions become

$$N_a(\xi, \eta) = \frac{1}{4} \cdot (1 + \xi_a \cdot \xi) \cdot (1 + \eta_a \cdot \eta) \quad a = 1, 2, 3, 4. \quad (3.44)$$

where  $(\xi_a, \eta_a)$  are the natural coordinates of the local node  $a$ .



**Figure 3.7:** Shape function corresponding to local node 3 ( $N_3$ ) of a bilinear quadrilateral element.

Because of the presence of a forth node, the deformation matrix  $[\mathbf{B}^e]$  yields,

$$[\mathbf{B}^e] = \begin{pmatrix} \frac{\partial N_1}{\partial x} & 0 & \frac{\partial N_2}{\partial x} & 0 & \frac{\partial N_3}{\partial x} & 0 & \frac{\partial N_4}{\partial x} & 0 \\ 0 & \frac{\partial N_1}{\partial y} & 0 & \frac{\partial N_2}{\partial y} & 0 & \frac{\partial N_3}{\partial y} & 0 & \frac{\partial N_4}{\partial y} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial y} & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial y} & \frac{\partial N_3}{\partial x} & \frac{\partial N_4}{\partial y} & \frac{\partial N_4}{\partial x} \end{pmatrix}$$

and the elemental stiffness matrix is given by (Eq. 3.39).

In this case, since the deformation matrix is not constant, integration all over the elements is needed to obtain elemental stiffness matrices. In order to integrate numerically (Eq. 3.39), the method of Gaussian Quadrature is used.

The method of Gaussian Quadrature substitutes the integral form by a discrete form with weighted factors, a deformation term and the integrands evaluated at a particular points, called Gauss points.

In case of a Q4 element, the integral needed to obtain the stiffness matrix is transformed into,

$$[\mathbf{K}^e] = \int_{\Omega^e} [\mathbf{B}^e]^T [\mathbf{D}] [\mathbf{B}^e] d\Omega \approx \sum_{ig=1}^2 \sum_{jg=1}^2 w_{ig} w_{jg} \cdot \| \mathbf{J}(ig,jg) \| \cdot [\mathbf{B}^e]^T(ig,jg) \cdot [\mathbf{D}] \cdot [\mathbf{B}^e](ig,jg) \quad (3.45)$$

where  $\| \mathbf{J} \|$  is the determinant of the jacobian matrix, which indicates the amount of deformation of the quadrilateral element in the xy-space with respect to the square base in the  $\xi\eta$ -space.

$$[\mathbf{J}] = \begin{pmatrix} \frac{\partial x(\xi,\eta)}{\partial \xi} & \frac{\partial x(\xi,\eta)}{\partial \eta} \\ \frac{\partial y(\xi,\eta)}{\partial \xi} & \frac{\partial y(\xi,\eta)}{\partial \eta} \end{pmatrix} \quad (3.46)$$

Stresses and strains are computed element by element,

$$\boldsymbol{\epsilon}^e = [\mathbf{B}^e] \cdot \mathbf{u}^e \quad (3.47)$$

$$\boldsymbol{\sigma}^e = [\mathbf{D}] \cdot \boldsymbol{\epsilon}^e \quad (3.48)$$

were  $\mathbf{u}^e$  are the elemental displacements.

# 4

## Structural Optimization

### 4.1 Introduction

Structural Optimization is the process of making an assemblage of materials in the optimal way in order to sustain certain loads. Thus, the objective of this subject is to find the best structure that performs this task.

The term "best" is related to the maximization or minimization of the objective function which could be the weight of the structure, stiffness, energy...

The general mathematical formulation of a structural optimization problem may be stated as,

$$\begin{aligned}
 & \underset{x,y}{\text{minimize}} && f_0(\mathbf{x}, \mathbf{y}) \\
 & \text{subject to} && \text{Behavioral constraints } \mathbf{y} \\
 & && \text{Design constraints } \mathbf{x} \\
 & && \text{Equilibrium constraints}
 \end{aligned} \tag{4.1}$$

where

- **Objective function ( $f_0$ )**: function of some property of the design that is to be extremized.
- **Design variable ( $\mathbf{x}$ )**: variable that describes the design to be optimized. It may be a geometric or a material parameter.
- **State variable ( $\mathbf{y}$ )** : variable that represents the response of the structure (displacements, strains, stresses...).

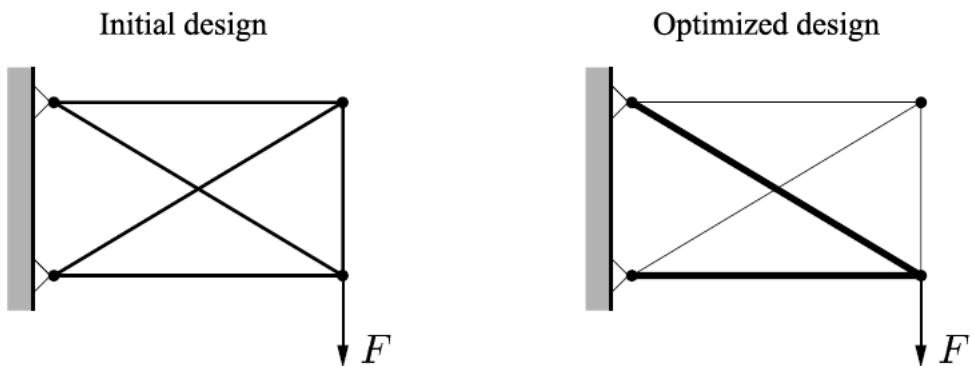
- **Equilibrium constraint:** equation that states an equilibrium condition of the physical problem. In elastostatics the equilibrium equation looks like  $[K] \cdot u = F$ , where  $[K]$  is the stiffness matrix,  $u$  the displacement vector and  $F$  the load vector.

## 4.2 Types of structural optimization problems

There exist three types of structural optimization problems which are classified in the following way:

### Sizing Optimization

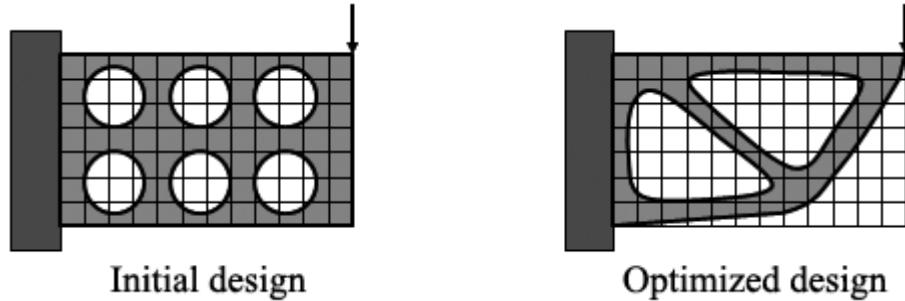
This type of problem is based on the optimization of structures when the design variables represent parameters of the material, cross-section areas of trusses or thickness of sheets. Figure (4.1) illustrates this kind of structural optimization.



**Figure 4.1:** Example of sizing optimization problem. The design variables are the cross-section areas of each bar (Christensen & Klarbring, 2009).

### Shape Optimization

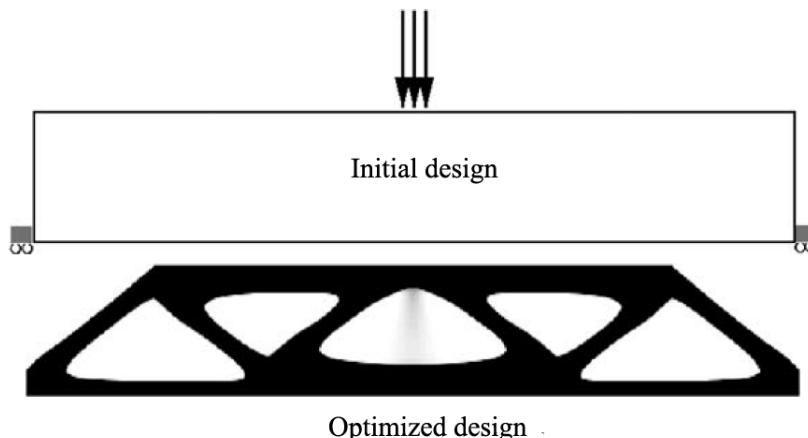
In shape optimization the design variables are parameters related to the contour or boundaries of the domain (Fig. 4.2). The goal is to find the optimum shape of the design domain subject to some constraints. This kind of problem doesn't create or remove new boundaries, only change their shape.



**Figure 4.2:** Example of shape optimization. The design variables are the contours of the domain. Ref. *Optishape-TS, Quint Co.*

### Topology Optimization

This type of structural optimization involves the determination of the optimal material distribution within a given design space under a certain loads subject to some constraints (Fig. 4.3). This is the most general form of structural optimization techniques.



**Figure 4.3:** Topology optimization problem. Given an initial ground structure, find the best material distribution to sustain given loads (Christensen & Klarbring, 2009).

## 4.3 Topology Optimization

Topology optimization is a type of structural optimization technique based on finding the optimal material layout given a set of loads and boundary conditions subject to constraints on the design.

The present project focuses on structural topology optimization technique applied to solid isotropic materials.

### 4.3.1 Minimum Compliance Design

The minimum compliance design problem is based on minimizing the external work (i.e. maximizing the global stiffness) of a ground structure (reference domain) subject to an equilibrium constraint, a limitation on its global volume  $V$  and bounds on design variables.

Although there are other formulations on the objective function and constraints (eg. minimum weight s.t. a stress constraint), this is the standard problem in structural topology optimization.

In this problem the design variables represent the local density of the domain, which must be positive.

Mathematically, the minimum compliance design can be stated as follows,

$$\begin{aligned} & \underset{u, \rho}{\text{minimize}} \quad \ell(\mathbf{u}) \\ & \text{subject to} \quad a(\rho, \mathbf{u}, \mathbf{v}) = \ell(\mathbf{v}) \\ & \qquad \int_{\Omega} \rho \, d\Omega \leq V \\ & \qquad \rho(\mathbf{x}) \geq 0, \quad \mathbf{x} \in \Omega \end{aligned} \tag{4.2}$$

where the same notation as section 2.1.4 has been used and  $\ell(\mathbf{u}) = (\mathbf{u}, \mathbf{b}) + (\mathbf{u}, \mathbf{h})$ .

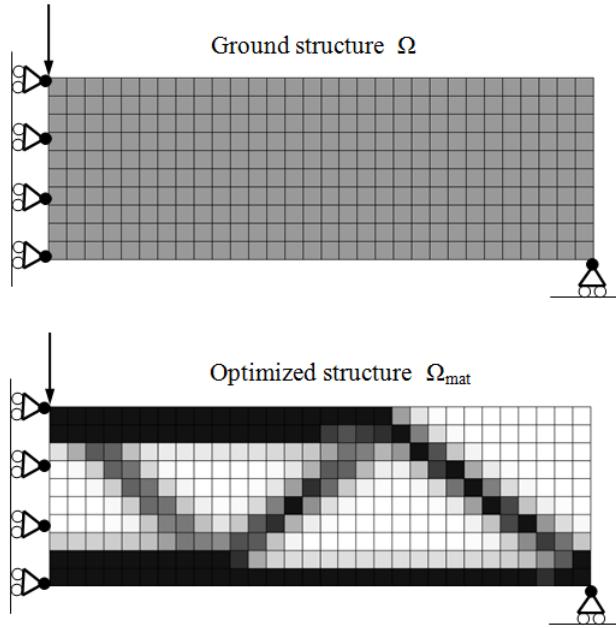
The same problem after FE-discretization becomes,

$$\begin{aligned} & \underset{u, \rho}{\text{minimize}} \quad \mathbf{F}^T(\rho) \cdot \mathbf{u}(\rho) \\ & \text{subject to} \quad \mathbf{v}^T \cdot \rho \leq V \\ & \qquad \rho_{min} \leq \rho \leq \rho_{max} \end{aligned} \tag{4.3}$$

where  $\mathbf{v}$  is the element volume vector,  $\rho$  is the fictitious element density vector and the dependence of nodal displacements  $\mathbf{u}$  on the element densities is achieved by the equilibrium equation  $\mathbf{u}(\rho) = [\mathbf{K}]^{-1}(\rho) \cdot \mathbf{F}(\rho)$ .

The fictitious element density is bounded between  $\rho_{min} = 0$  when the element has no material (void element) and  $\rho_{max} = 1$ , when the element is a material point.

Thus, in the topology design of a structure one is interested in finding the optimal distribution  $\Omega_{mat}$  of 0 or 1 densities (void or material region) which maximizes the global stiffness (minimizes the external work) on a given reference domain  $\Omega$  with a given amount of material  $V$ .



**Figure 4.4:** Example of a minimum compliance design of a half MBB beam. Volume restriction: 50 %.

### 4.3.2 SIMP Method

The ideal result after optimization would be a discretized 0-1 material distribution structure. However, although it is possible to formulate the compliance design problem in terms of integer (0-1) design variables, this leads to a different formulation and integer programming algorithms should be used.

For this reason, the most commonly used approach is to consider continuous design variables  $\rho$  ranging from 0 to 1 and introduce some kind of penalization that forces the solution to an almost 0-1 values of density.

This is what SIMP (Solid Isotropic Material with Penalization) method does. This approach adds the design variables into the minimum compliance problem through the stiffness tensor.

$$\mathbf{E}_{ijkl} = \phi(\rho(x)) \cdot \mathbf{E}_{ijkl}^0 \quad (4.4)$$

$$0 \leq \rho(x) \leq 1, \quad x \in \Omega$$

where  $\phi$  is a penalty function of  $\rho$  such that  $\phi(\rho = 0) = 0$  and  $\phi(\rho = 1) = 1$  and  $E_{ijkl}^0$  is the original (non-penalized) stiffness tensor. Hence, the stiffness tensor becomes 0 when  $\rho=0$  (no material) and 1 when  $\rho=1$  (material). After a FE-discretization and considering linear and isotropic materials, the penalty function can be applied to the Young's modulus of each element.

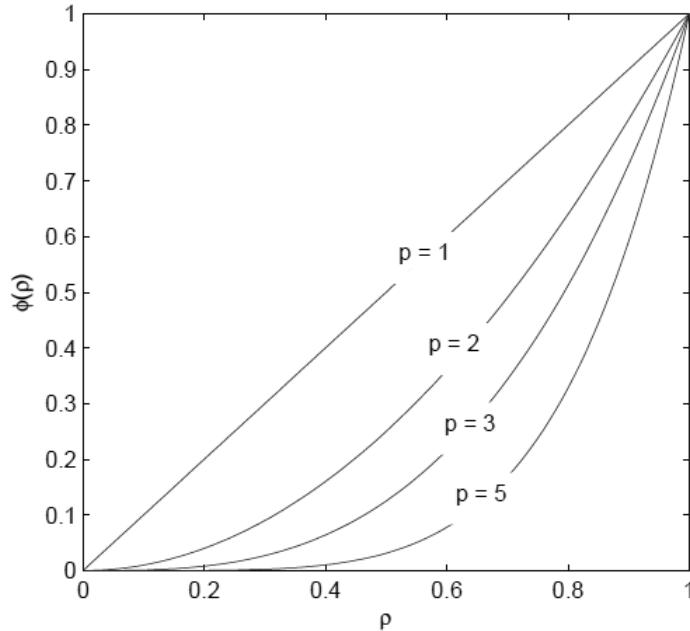
$$\mathbf{E} = \phi(\rho) \cdot \mathbf{E}^0 \quad (4.5)$$

SIMP approach may also be formulated by means of the elemental stiffness matrix,

$$[\mathbf{K}_e] = \phi(\rho) \cdot [\mathbf{K}_e^0] \quad (4.6)$$

One of the most commonly used SIMP penalty function is a power law,

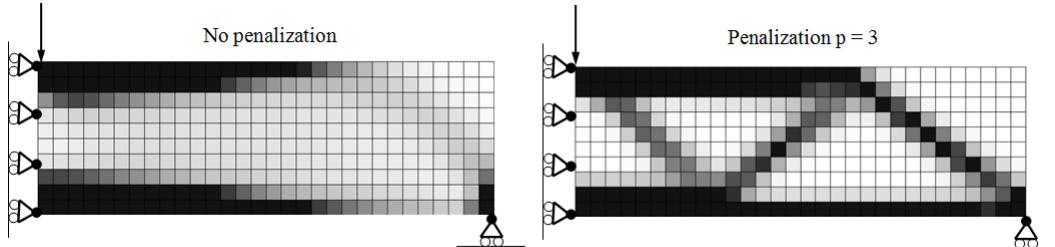
$$\phi(\rho) = \rho^p, \quad p > 1 \quad (4.7)$$



**Figure 4.5:** SIMP penalty function based on a power law.

Figure (4.5) shows the power-law function used in SIMP approach. Intermediate "grey" densities, which have no physical interpretation, are penalized more severely the higher the parameter  $p$  is. A typical value of  $p \geq 3$  is usually required to obtain almost 0-1 solutions.

Figure 4.6 compares the same problem without penalizing intermediate densities and with a penalty parameter  $p=3$ . Although the penalized problem is not a fully 0-1 design, it has a much better interpretation than the non-penalized case.



**Figure 4.6:** Optimization of an MBB beam using SIMP method. On the left no penalization ( $p=1$ ) has been used; on the right, the same problem is solved applying a power of  $p=3$ .

There exists other penalty functions such as an hyperbolic sine function, weight penalization function, RAMP function, Hashin-Strikman...

### 4.3.3 Solution methods to structural topology optimization problems

The compliance design problem is a nonlinear optimization problem that has as many design variables as the number of elements of the mesh, one inequality constraint and one side constraint bounding the design variables between 0 and 1.

This problem may be solved using nonlinear constrained optimization methods discussed in chapter 2. However, two classical optimization methods will be discussed in this section: the Optimality Criteria method (OC) proposed by M.P. Bendsøe and the Method of Moving Asymptotes (MMA) proposed by K. Svanberg. These two methods have been widely used in structural topology optimization problems.

#### 4.3.3.1 Optimality Criteria Method (OC)

The optimality criteria methods are classical methods to solve structural optimization problems applied to the minimum compliance. O.C. method is based on solving the Lagrangian of the compliance problem satisfying the KKT conditions.

The necessary conditions (KKT conditions) for optimality with respect to the design variable  $\rho$  must satisfy the stationarity condition. The Lagrangian of the minimum compliance is,

$$\mathcal{L}(\rho, \lambda, \mu^{-/+}) = \mathbf{F}^T \cdot \mathbf{u} + \lambda(\mathbf{v}^T \cdot \rho - V) + \mu^-(\rho_{min} - \rho) + \mu^+(\rho - \rho_{max}) \quad (4.8)$$

The stationarity condition of the Lagrangian with respect the design variable gives,

$$\frac{\partial \mathcal{L}}{\partial \rho_e} = -\mathbf{u}_e^T \cdot \frac{\partial [\mathbf{K}_e]}{\partial \rho_e} \cdot \mathbf{u}_e + \lambda \cdot v_e + \mu^+ - \mu^- = 0 \quad (4.9)$$

after subtracting the equilibrium equation multiplied by the displacements  $\mathbf{u}^T ([K] \mathbf{u} - \mathbf{F}) = 0$  to (Eq. 4.8). The rest of KKT conditions are the switching conditions,

$$\lambda \geq 0, \quad \mu^- \geq 0, \quad \mu^+ \geq 0, \quad \mu^-(\rho_{e min} - \rho_e) = 0, \quad \mu^+(\rho_e - \rho_{e max}) = 0 \quad (4.10)$$

For intermediate values of  $\rho_e$ , this is  $\rho_{e min} \leq \rho_e \leq \rho_{e max}$ , the stationarity condition after applying the SIMP approach (Eq. 4.6) and (Eq. 4.5) yields,

$$\frac{p \cdot \rho_e^{p-1} \cdot \mathbf{u}_e^T \cdot [\mathbf{K}_e^0] \cdot \mathbf{u}_e}{v_e} = \lambda \quad (4.11)$$

An important fact about this last equation is that the strain energy density-like term is constant and equal to the Lagrange multiplier  $\lambda$ .

The proposed update scheme for the design variable at each iteration by Ref. [4] is the following,

$$\rho_{k+1} = \begin{cases} \max((1 - \zeta) \cdot \rho_k, \rho_{min}) & \text{if } \rho_k \cdot B_k \leq \max((1 - \zeta) \cdot \rho_k, \rho_{min}) \\ \min((1 + \zeta) \cdot \rho_k, \rho_{max}) & \text{if } \rho_k \cdot B_k \geq \min((1 + \zeta) \cdot \rho_k, \rho_{max}) \\ \rho_k \cdot B_k & \text{otherwise} \end{cases} \quad (4.12)$$

$$B_k = \frac{p \cdot \rho_e^{p-1} \cdot \mathbf{u}_{ek}^T \cdot [K_e^0] \cdot \mathbf{u}_{ek}}{v_e \cdot \lambda_k}$$

where the displacements are calculated from the equilibrium equation at each iteration. The parameter  $\zeta$  is a move limit and may be adjusted for efficiency issues of the algorithm.

This algorithm adds material to regions where the specific strain energy density-like exceeds  $\lambda$  ( $B_k \geq 1$ ) and removes it if the energy is below ( $B_k \leq 1$ ). This procedure is performed unless  $\rho$  is out of bounds.

### 4.3.3.2 The Method of Moving Asymptotes (MMA)

The Method of Moving Asymptotes was proposed by K. Svanberg in 1987 (Ref. [29]) and has been commonly used in structural optimization problems. This method belongs to the group of sequentially convex programming (SCP) optimization algorithms to solve inequality constrained optimization problems. SCP is an iterative algorithm where the original nonlinear problem is approximated by a convex function, then this modified problem is solved and the solution is mapped into the original function. The algorithm is repeated until an optimal solution (i.e. stopping criterion) is reached.

Consider a general nonlinear optimization problem with inequality constraints,

$$P \begin{cases} \underset{x}{\text{minimize}} & f_0(\mathbf{x}) \\ \text{subject to} & g_i(\mathbf{x}) \leq 0 \quad i = 1, \dots, m \\ & \mathbf{x}_{\min} \leq \mathbf{x} \leq \mathbf{x}_{\max} \end{cases} \quad (4.13)$$

The general algorithm of MMA for solving a nonlinear optimization problem is to generate at each iteration a sequence of explicit subproblems according to the following steps:

---

#### Algorithm 6 : MMA Algorithm

---

- **Step 0:** Choose an initial starting point  $\mathbf{x}_0$  and set iteration index  $k = 0$ .
  - **Step 1:** Calculate both objective function and constraints  $f_i(\mathbf{x}_k)$  and their gradients  $\nabla f_i(\mathbf{x}_k)$  all evaluated at the current iteration point  $\mathbf{x}_k$ .
  - **Step 2:** Generate a subproblem  $P_k$  by replacing the original functions  $f_i$  (objective and constraints) of the original problem  $P$  by a convex approximation  $\hat{f}_i^k$ .
  - **Step 3:** Solve the subproblem  $P_k$  and let the solution  $\mathbf{x}_k$  obtained be the solution of the next iteration point  $\mathbf{x}_{k+1}$ . Set  $k = k+1$  and go to **Step 1** until a stopping criterion is reached.
- 

The above steps indicate the global algorithm to solve a general nonlinear problem sequentially. Therefore, in order to apply these steps it must be described: the approximated functions  $\hat{f}_i^k$  and how the subproblem  $P_k$  should be solved.

MMA defines the approximating functions as,

$$\hat{f}_i^k = r_i^k + \sum_{j=1}^n \left( \frac{p_{ij}^k}{U_j^k - x_j} + \frac{q_{ij}^k}{x_j - L_j^k} \right) \quad i = 0, 1, \dots, m \quad j = 0, 1, \dots, n \quad (4.14)$$

where

$$p_{ij}^k = (U_j^k - x_j^k)^2 \cdot \max \left( 0, \left. \frac{\partial f_i}{\partial x_j} \right|_{\mathbf{x}_k} \right) \quad (4.15)$$

$$q_{ij}^k = (x_j^k - L_j^k)^2 \cdot \max \left( 0, - \left. \frac{\partial f_i}{\partial x_j} \right|_{\mathbf{x}_k} \right) \quad (4.16)$$

$$r_i^k = f_i(\mathbf{x}_k) - \sum_{j=1}^n \left( \frac{p_{ij}^k}{U_j^k - x_j^k} + \frac{q_{ij}^k}{x_j^k - L_j^k} \right) \quad (4.17)$$

and parameters  $L_j^k$  and  $U_j^k$  are called moving asymptotes such that,

$$L_j^k < \alpha_j^k \leq x_j \leq \beta_j^k < U_j^k$$

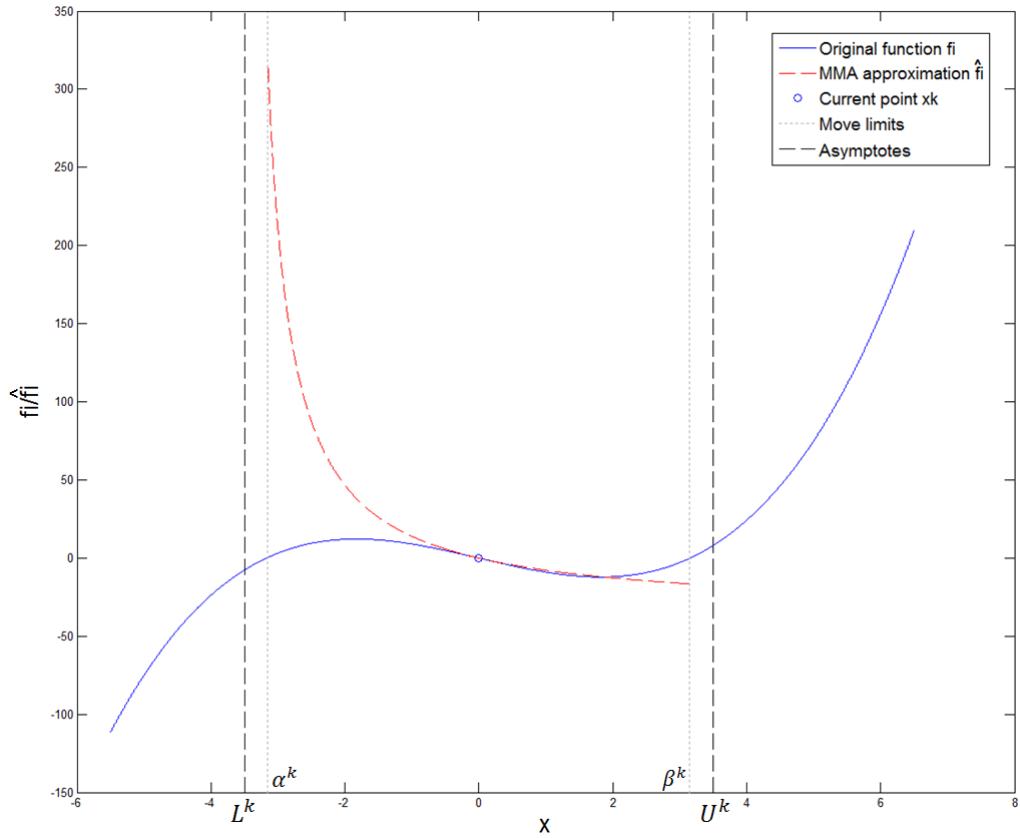
where  $\alpha_j^k$  and  $\beta_j^k$  are move limits at each iteration k.

It is important to note that when  $p_{ij}$  is non-zero,  $q_{ij}$  is zero and vice versa. Hence, depending on the sign of the gradient at each iteration point, only one of the asymptotes is used.

MMA approximation has some good properties to solve nonlinear optimization problems:

- Functions  $\hat{f}_i^k$  are a first order approximations such that  $\hat{f}_i^k = f_i$  and  $\nabla \hat{f}_i(\mathbf{x}_k) = \nabla f_i(\mathbf{x}_k)$ .
- $\hat{f}_i^k$  are explicit convex functions since the Hessian  $\nabla^2 \hat{f}_i(\mathbf{x}_k)$  is positive definite.
- $\hat{f}_i^k$  are chosen in such a way that each subproblem  $P_k$  is separable.
- Parameters  $\alpha_j^k, \beta_j^k, L_j^k$  and  $U_j^k$  can be tuned for convergence purposes.

Figure (4.7) shows the approximation of a nonlinear function used in MMA.



**Figure 4.7:** Approximation used in MMA (red) of a one dimensional case. In this case the lower asymptote  $L^k$  is active due to the sign of the gradient of the original function (blue).

The values of the asymptotes are changed at each iteration following the heuristic rule proposed by Ref. [31].

- For  $k = 1$  and  $k = 2$

$$L_j^k = x_j^k - 0.5 \cdot (x_{j \max} - x_{j \min}) \quad (4.18)$$

$$U_j^k = x_j^k + 0.5 \cdot (x_{j \max} - x_{j \min})$$

- For  $k \geq 3$

$$L_j^k = x_j^k - \gamma_j^k \cdot (x_j^{k-1} - L_j^{k-1}) \quad (4.19)$$

$$U_j^k = x_j^k + \gamma_j^k \cdot (U_j^{k-1} - x_j^{k-1})$$

where

$$\gamma_j^k = \begin{cases} 0.7 & \text{if } (x_j^k - x_j^{k-1}) \cdot (x_j^{k-1} - x_j^{k-2}) < 0 \\ 1.2 & \text{if } (x_j^k - x_j^{k-1}) \cdot (x_j^{k-1} - x_j^{k-2}) > 0 \\ 1.0 & \text{if } (x_j^k - x_j^{k-1}) \cdot (x_j^{k-1} - x_j^{k-2}) = 0 \end{cases} \quad (4.20)$$

The foundations of choosing these values is related to stabilization issues. If the process tends to oscillate, the asymptotes are taken closer to the current point in order to better capture the curvature of the original function. If, instead, the process tends to be slow and monotone, the asymptotes are taken away to speed it up.

In order to do so, design variables of the last two iterations and the asymptotes of the last iteration are needed.

The subproblem at each iteration takes the following form,

$$P_k \left\{ \begin{array}{ll} \text{minimize}_x & r_0^k + \sum_{j=1}^n \left( \frac{p_{0j}^k}{U_j^k - x_j} + \frac{q_{0j}^k}{x_j - L_j^k} \right) \\ \text{subject to} & r_c^k + \sum_{j=1}^n \left( \frac{p_{cj}^k}{U_j^k - x_j} + \frac{q_{cj}^k}{x_j - L_j^k} \right) \quad c = 1, 2, \dots, m \\ & \alpha_j^k \leq x_j \leq \beta_j^k \quad j = 1, 2, \dots, n \end{array} \right. \quad (4.21)$$

As mentioned before, subproblem  $P_k$  is a explicit, convex and separable approximation of the original one  $P$ . Therefore, it is suitable to solve by Lagrangian duality or interior point methods.

In this section it has been explained the original MMA. However, there exist updates of this method that are more efficient such as GCMMA, GBMMA, TGMMA... (Refs. [7, 19, 31])

### 4.3.3.3 Sensitivity analysis

Sensitivity analysis describes the methodology to calculate gradients of functions. This is, the way the function changes its value with respect to the design variable.

Both O.C. and MMA methods require the derivatives of the objective function and the constraint (only in MMA) to solve the compliance design problem. Therefore, a sensitivity analysis is needed.

The objective function can be expressed as a function of the densities and the displacements,

$$\frac{Df_o(\rho)}{D\rho_e} = \frac{\partial \hat{f}_o(\rho, \mathbf{u})}{\partial \rho_e} + \frac{\partial \hat{f}_o(\rho, \mathbf{u}(\rho))}{\partial \mathbf{u}} \cdot \frac{D\mathbf{u}(\rho)}{D\rho_e} \quad (4.22)$$

The derivative of the displacements is obtained by adding the derivative of the equilibrium equation  $[\mathbf{K}](\rho) \cdot \mathbf{u}(\rho) = \mathbf{F}(\rho)$  and using the adjoint method (Christensen & Klarbring, 2009) to calculate derivatives yields,

$$\frac{Df_o(\rho)}{D\rho_e} = 2 \cdot \mathbf{u}^T \cdot \frac{\partial \mathbf{F}(\rho)}{\partial \rho_e} - \mathbf{u}^T \cdot \frac{\partial [\mathbf{K}]}{\partial \rho_e} \cdot \mathbf{u} \quad (4.23)$$

In case of considering that the load vector does not depend on the mass distribution of the structure (i.e. design variables), such as weight or inertial forces, and adopting the SIMP method, (Eq. 4.23) becomes,

$$\frac{Df_o(\rho)}{D\rho_e} = -p \cdot \rho_e^{p-1} \cdot \mathbf{u}_e^T \cdot [\mathbf{K}_e^0] \cdot \mathbf{u}_e \quad (4.24)$$

On the other hand, the sensitivity of the volume constraint is just

$$\frac{Dg_c(\rho)}{D\rho_e} = \frac{\partial}{\partial \rho_e} \left( \sum_{e=1}^{nel} v_e \cdot \rho_e - Vf \cdot V_0 \right) = v_e \quad (4.25)$$

It is important to remark that for  $p = 1$  the problem (4.2) is convex because the Hessian of the Lagrangian is positive definite (Svanberg, 1994). However, a penalty factor  $p > 1$  makes the penalty function  $\phi(\rho)$  nonlinear and the minimum compliance problem becomes non-convex.

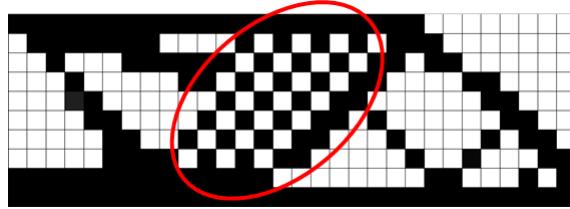
#### 4.3.4 Numerical instabilities

In displacement based topology optimization there exist solutions which are not desired by engineers. These complications are related to computational numerical instabilities that must be taken into account and soften their effect.

These numerical instabilities are classified into three categories: checkerboard pattern, mesh-dependency and local minima.

##### 4.3.4.1 Checkerboard pattern

Checkerboards (Fig. 4.8) are regions of alternating solid and void elements ordered in a checkerboard fashion. This instability has its origin related to features in the finite element discretization and it considers the checkerboard pattern as a good solution to stiffen the structure. Checkerboards usually appear due to the non-convergent behaviour of the FE-discretization.



**Figure 4.8:** Example of the checkerboard pattern using SIMP method.

There are two typical methods to avoid or soften checkerboards: discretizing the domain with high-order finite elements or using filtering techniques.

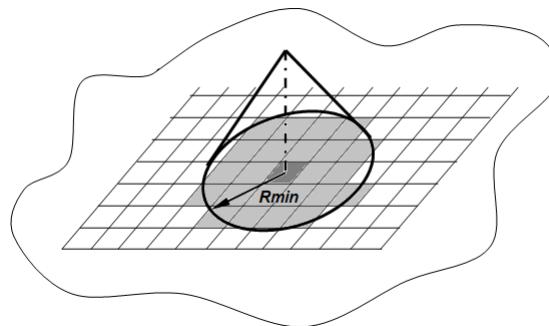
As discussed in Ref. [27], checkerboards in SIMP method are only prevented using 8 or 9-node elements if the penalty factor is small enough. However, a drawback using elements with a higher order leads to an increase of the computational time.

Filters are based on image processing techniques and usually give good results removing checkerboard patterns. They assign a value (density or sensitivity) on each element considering the information of neighbour elements.

The filtered value of the objective function sensitivity is a linear average of the sensitivity of the elements enclosed by the filter radius  $R_{min}$  (Eq. 4.26).

$$\frac{\partial \hat{f}_o}{\partial \rho_k} = \frac{1}{\rho_k \cdot \sum_{e=1}^{nel} \hat{H}_{ek}} \cdot \sum_{e=1}^{nel} \hat{H}_{ek} \cdot \rho_e \cdot \frac{\partial f_o}{\partial \rho_e} \quad (4.26)$$

where  $\hat{H}_{ek} = R_{min} - \text{distance}(e,k)$ .



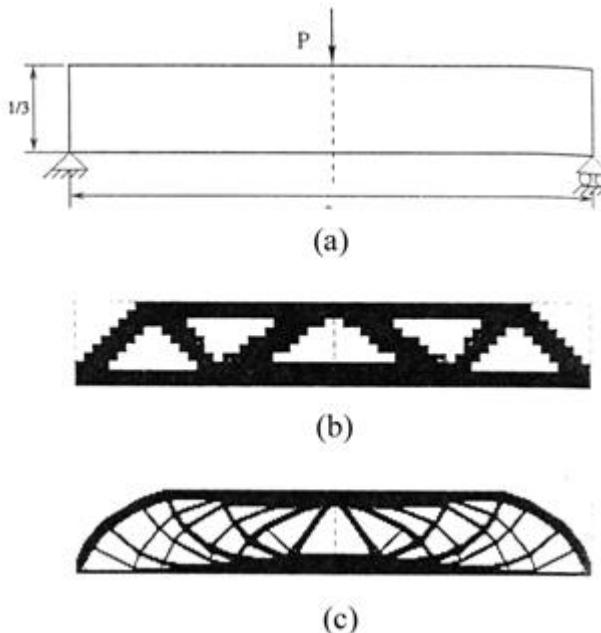
**Figure 4.9:** Representation of the enclosed elements by the sensitivity filter.

#### 4.3.4.2 Mesh-dependency

This instability refers to the problem of obtaining different solutions when changing the mesh discretization.

Ideally, a mesh refinement leads to a better description of the domain and boundaries. Thus, better results would be obtained and the optimal solution would be more detailed but not different from more coarse meshes.

Mesh-dependency instability occurs because of non-existence of solutions or non-unique solutions of the physical description of the problem (Sigmund & Peterson, 1998).



**Figure 4.10:** Example of mesh-dependency instability. a. Ground structure, b. solution using a 600-element mesh, c. solution using a 5400-element mesh (Sigmund & Peterson, 1998).

The most common methods to get rid of this instability is using filtering techniques such as perimeter control or mesh independent filter. The first one is based on limiting the size of void region perimeters and the second one (Bendsøe & Sigmund, 2003) modifies the design sensitivity of a specific element based on a weighted average of sensitivities of neighbour elements.

#### 4.3.4.3 Local minima

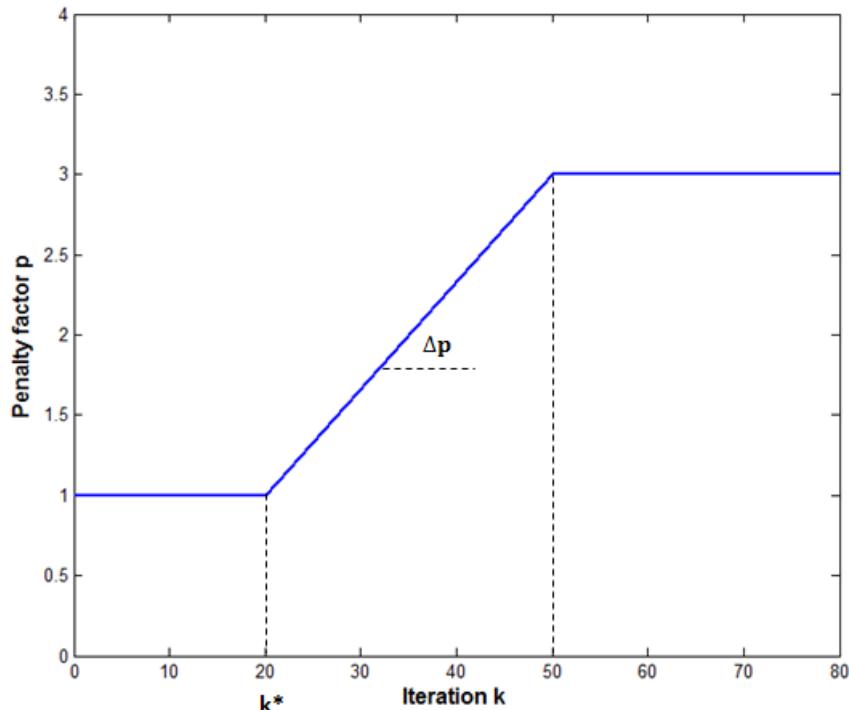
This instability refers to the problem of obtaining different solutions when modifying the algorithm parameters because they change the original problem (4.2).

The original minimum compliance problem is convex, however the penalty factor  $p > 1$  used in SIMP makes the objective function non-convex. Therefore, the minimum compliance problem becomes non-convex and a global minimum is not guaranteed if reached a stationarity point. Filter radius, move limits or initial density distribution also affect the convex behaviour.

This is one of the main disadvantages of the gradient based optimization methods. In order to solve this, heuristic methods such as continuation approaches are required to obtain global minima.

The idea of continuation methods is to start with algorithmic parameters that don't modify the original problem ( $p=1$ , no filter...) and gradually increase them along the iterations.

This approach is based on obtaining of a binary solution and the KKT conditions. Reference [22] examines the validity of the method.



**Figure 4.11:** Continuation method applied to SIMP penalty factor  $p$ . From a given iteration  $k^*$ , the penalty factor will increase  $\Delta p$  at each  $k$  until a limit of  $p$ .

### 4.3.5 General algorithm

The procedure for solving the minimum compliance design problem may be described in the following order:

---

#### **Algorithm 7 : STO General Algorithm**

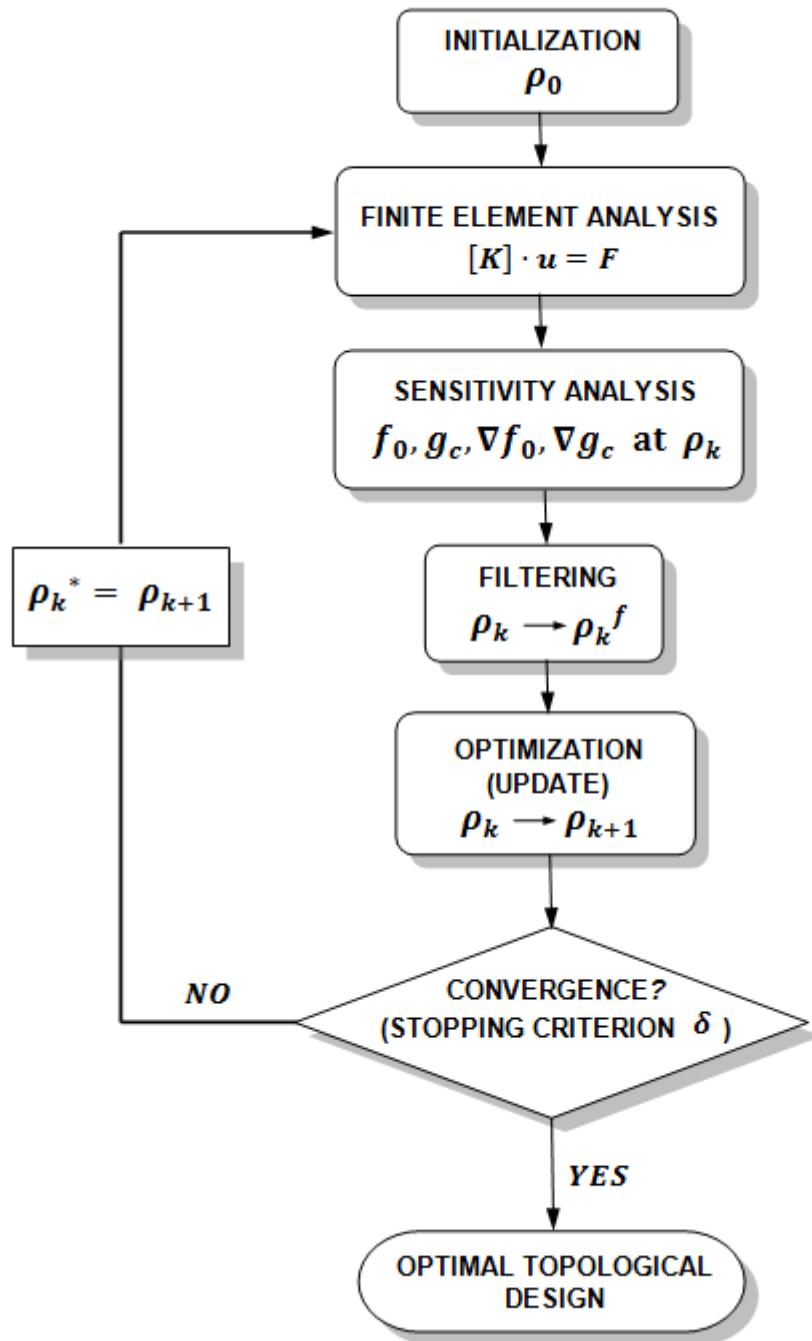
---

- I. Start with an initial design  $\rho_0$  and set the iteration counter k=0.
  - II. Calculate the nodal displacement vector for the current design variable  $\mathbf{u}(\rho_k)$  by performing a finite element analysis  $[\mathbf{K}](\rho_k) \cdot \mathbf{u}(\rho_k) = \mathbf{F}(\rho_k)$ .
  - III. Calculate the objective function  $f_0(\rho_k) = \mathbf{F}^T(\rho_k) \cdot \mathbf{u}(\rho_k)$  (compliance), the volume constraint  $g_c(\rho_k) = \mathbf{v}^T \cdot \rho_k - V$  and their gradients ( $\nabla f_0$  and  $\nabla g_c$ ) evaluated at the current iteration point k. (the calculation of constraint function and its gradient is only needed for MMA)
  - IV. Filtering process (if needed) to deal with numerical instabilities.
  - V. Solve the problem by a nonlinear optimization algorithm to obtain an updated value of the design variable  $\rho_{k+1}$ .
  - VI. Set k = k+1 and return to step II until a stopping criterion is reached.
- 

In step III the calculations of the gradients of the objective function and volume constraint depends on what kind of optimizer is used. O.C. method only needs the gradient of the objective function but MMA may need both the gradient of the objective function and the volume constraint.

Other optimizers use higher order gradients to perform the optimization process. For instance, SQP(Sequential Quadratic Programming) algorithm uses the second order gradient but the computational cost is increased.

Figure (4.12) shows a flowchart of a general iterative process to solve a structural topology optimization problem.



**Figure 4.12:** General algorithm to solve a topology optimization problem.

## **Part II**

# **Structural Topology Optimization Code**

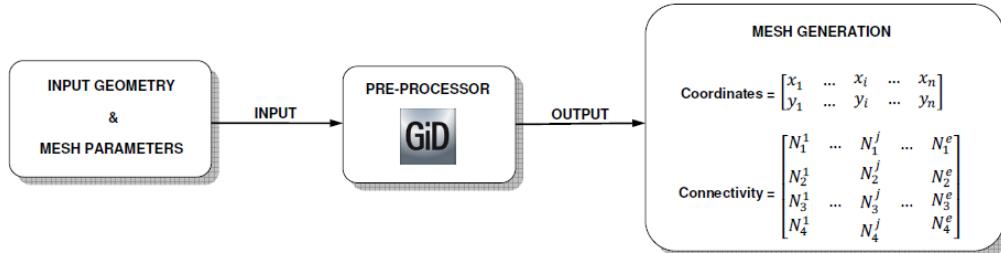
# 5

## Pre and post processing

### 5.1 Pre-processing

In order to generate the mesh of the reference domain, an external pre processing software will be used. This software tool allows you to choose the fineness of the mesh, element type (T3,Q4,T6,Q9...), unstructured or structured meshes...

Once input mesh parameters are selected and introduced, the preprocessor will give the coordinate and connectivity matrices as an output (Fig. 5.1). Then, these data will be sent to Matlab to start the optimization algorithm.



**Figure 5.1:** Diagram of mesh generation using an external pre and post-processing software (GiD).

The intermediate step between mesh files generated by GiD with a specific format and the implementation in MATLAB is performed by either Excel sheet or a txt. file.

## 5.2 Post-processing

As well as the pre-processing part, the post-process will also be carried out by the same external software tool.

In this case, the post-processor reads the files generated by the optimization algorithm. These files are the density vector  $\rho$ , the elemental strains  $\epsilon^e$  and the elemental stresses  $\sigma^e$  at each iteration step  $k$ .

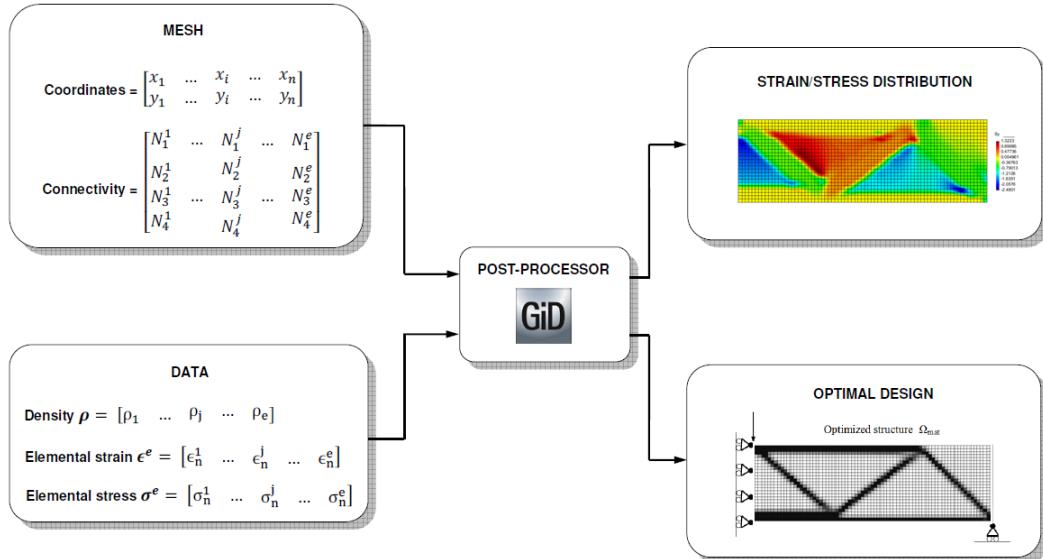


Figure 5.2: Diagram of post-processing using GiD.

Similarly to the pre-processing part, all data generated by MATLAB must be converted in a specific format readable by GiD. In this way, some subroutines are added to the code to adapt these data files into the required input format by GiD.

These subroutines are included in Appendix [C].

# 6

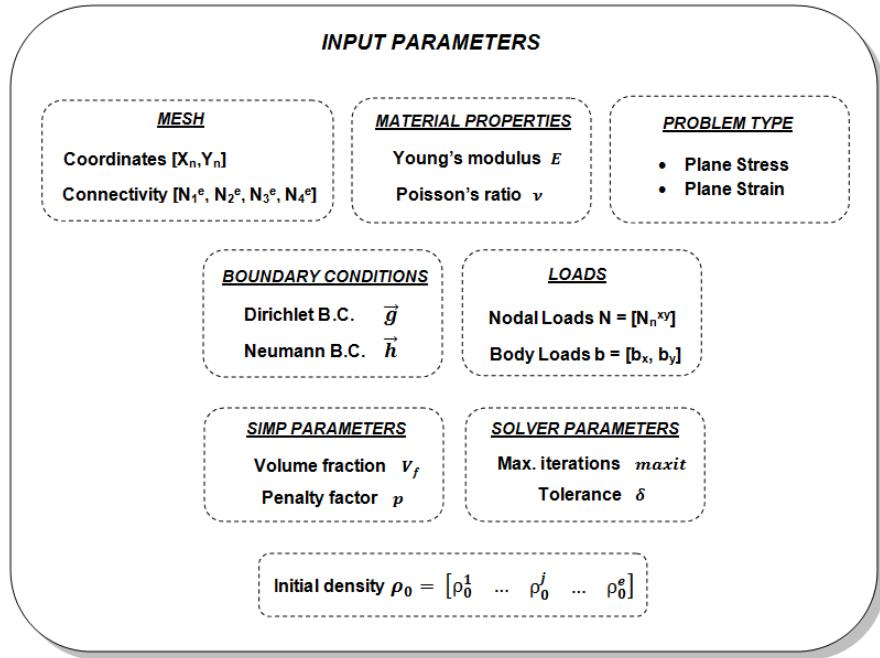
## Code Design

### 6.1 Input parameters

The global code, excluding the pre and post-processing parts, is organized modularly following the flowchart discussed in section 4.3.5.

The code is formed by three main modules: finite element module, sensitivity module and optimization module. Each one depends on the previous module following the order just described.

Once the problem and reference domain are selected and generated on the pre-processing step, then the input parameters of the problem must be defined. Figure 6.1 shows the input parameters required to run the program.



**Figure 6.1:** Required input parameters.

## 6.2 Block diagram

As discussed previously, the code has three main modules. First, the finite element module is responsible for calculating the displacements, strains and stresses with the SIMP modification on the elemental stiffness matrix and the given initial density vector.

Then, both objective function and volume constraint as well as their derivatives evaluated at the current density are calculated by the sensitivity module. Next, these values together with some input parameters are sent to the optimization module which takes care of updating the density vector. The updating process may be performed by either OC method or MMA, only a small change on the input variables is needed.

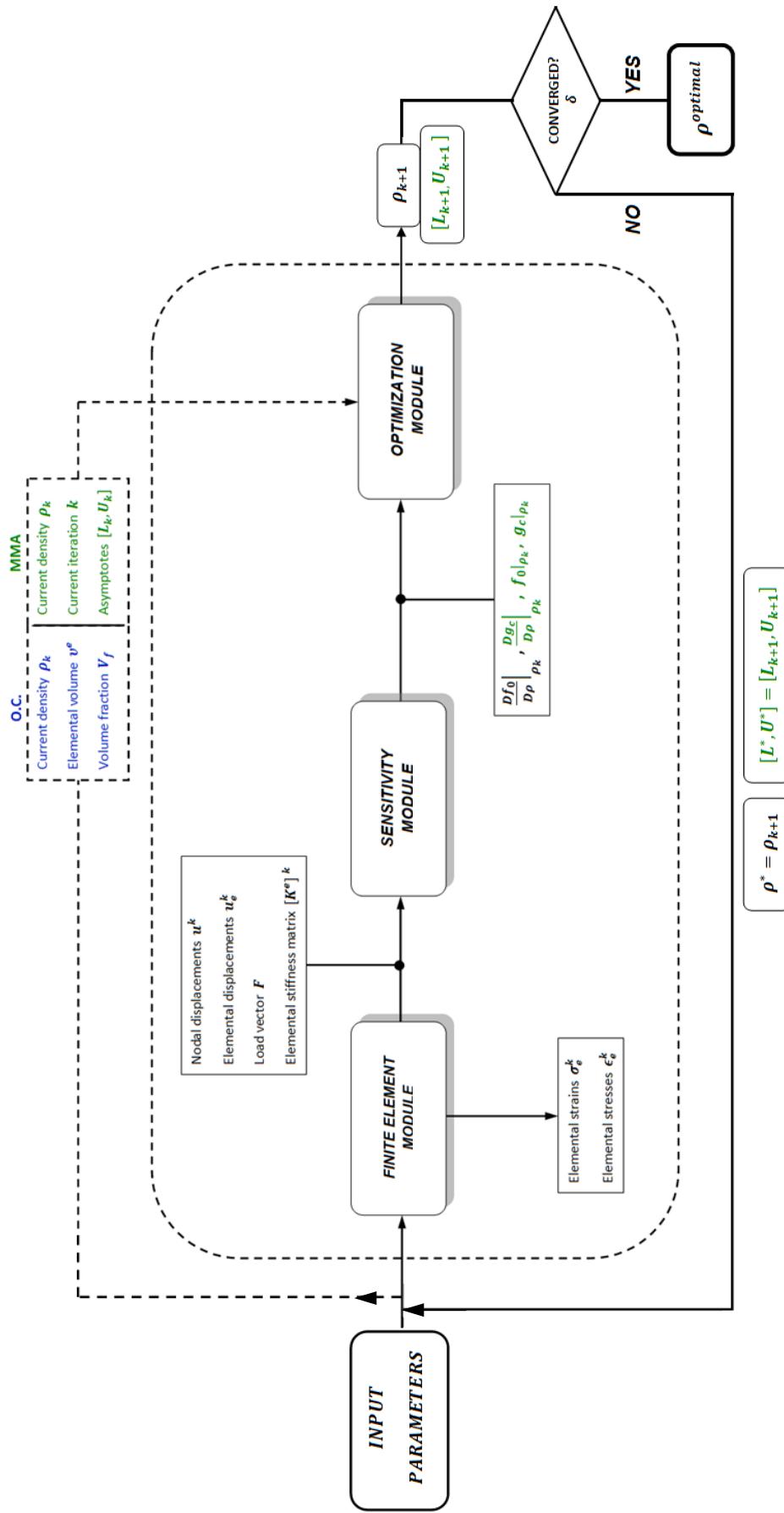
Finally, unless a stopping criterion is reached, the algorithm goes back to the first module with the updated parameters as the previous ones.

**Note:** The filter algorithm has not been included in the block diagram. For the sensitivity filter, only a small change on the sensitivity module would be required.

A block diagram of the general code is presented below (Fig. 6.2). The MATLAB code<sup>1</sup> can be found in Appendix [C].

---

<sup>1</sup>The final code may vary slightly from the general block diagram presented in this section.



**Figure 6.2:** Diagram of the global structural topology optimization code. Blue color: parameters only used in O.C./ Green color: parameters only used in MMA / Black color: parameters used in both.

# 7

## Benchmarks: Test application and validation

### 7.1 Selected benchmarks

In order to verify and validate the code, some tests have been performed. From topology optimization literature, there are two commonly used benchmarks: the MBB beam and the long cantilever beam.

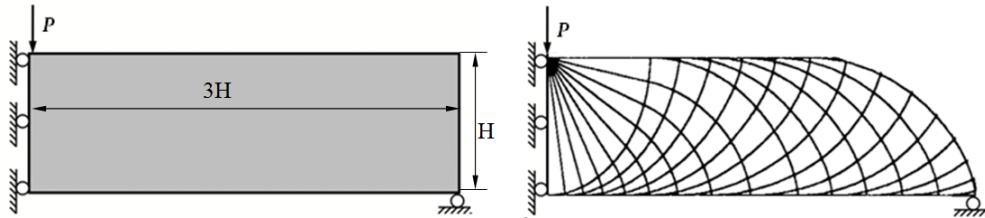
The problem definition and reference domains as well as the analytical optimized solution of each benchmark are presented in the following sections.

All results were obtained with the following parameters.

**Table 7.1:** Benchmark global input parameters.

| Parameter                   | Value  |
|-----------------------------|--------|
| Young's modulus $E$         | 1      |
| Poisson's coefficient $\nu$ | 0.3    |
| Initial density $\rho_0$    | Random |
| Load $P$                    | 1      |

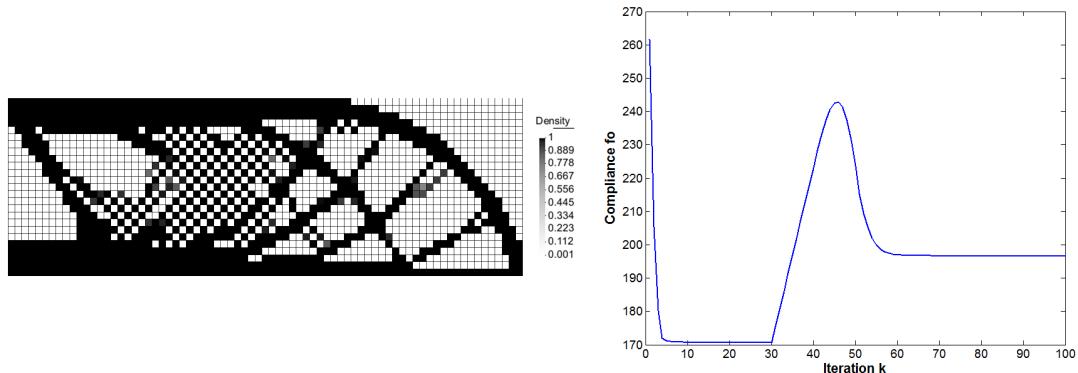
### 7.1.1 Messerschmitt–Bölkow–Blohm (MBB) beam



**Figure 7.1:** Problem definition with reference domain (left) and analytical optimal solution (right) of an MBB beam (Rozvany, 2009).

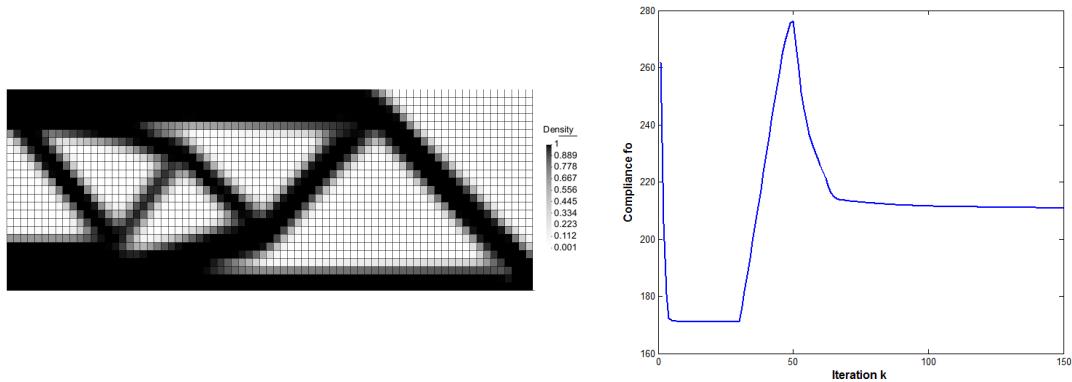
#### Optimality Criteria

First, the test was performed with no filter and using a continuation scheme ( $\Delta p = 0.1$ ,  $k^* = 30$ ). The following solution was obtained:



**Figure 7.2:** MBB benchmark using a continuation method without sensitivity filter.

The presence of checkerboard is noticeable in the central region. However, the solution tends to the analytical result. Because of the checkerboard instability, a method to avoid it is needed. Thus, a filter radius of  $R_{min} = 2$  is applied.

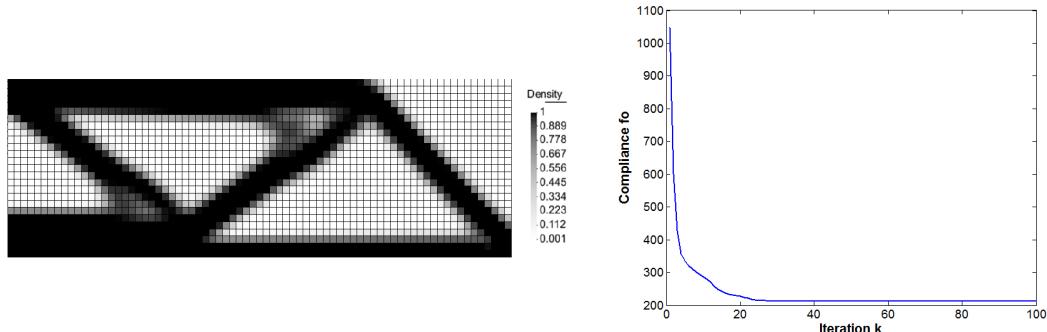


**Figure 7.3:** MBB benchmark using a continuation method and sensitivity filter  $R_{min} = 2$ .

The first impression is that the solution has changed dramatically, but if one looks closely the filter radius left the fundamental trusses and removed those ones which are not so important. Although the sensitivity filter has heuristic basis, the point is that it removed the checkerboard pattern. Moreover, it was checked that for various meshes the result is the same.

The effect of the continuation method is also considered. If one puts aside the continuation approach, then using a fixed parameter  $p$  may lead to a local solution instead of the global one.

The following result was obtained using a suggested fixed value  $p = 3$ .



**Figure 7.4:** MBB benchmark using a fixed value of the penalty factor  $p=3$  and sensitivity filter  $R_{min} = 2$ .

As it can be seen, the solution has changed. This is because the algorithm prematurely converged to a local solution.

It's important to note the difference between plots of figures (7.3) and (7.4). Both figures represent the convergence of the compliance through the iterations.

The first one (Fig. 7.3) has three regions: a first convergence, then a increase of the compliance and finally a second convergence. This is because the value of  $p$  is increasing  $\Delta p$  at each iteration from 1 to 3 starting at  $k^*=30$ . Therefore, the first convergence corresponds to a convex problem ( $p=1$ ) until  $k^*=30$ . The compliance is the lowest but the density distribution is not sufficiently penalized, so the  $p$  value is gradually increased to 3.

The continuation approach used in this project increases the value of  $p$  with an increment of  $\Delta p$  and then the next iteration takes the new value with  $p_{new} = p + \Delta p$ . Another approach would be to increase the value of  $p$  and let the algorithm converge with this  $p + \Delta p$  value, once it converged increase again  $p$  and so on.

A drawback of this approach is that a much higher iteration number is needed. For this reason the first approach is used and although the new point  $k+1$  is not the convergence value at  $p + \Delta p$ , it is a better point for the next iteration.

Finally, once  $p=3$  is reached, this value remains constant and this leads to a second convergence.

Otherwise, in figure (7.4), the value of  $p$  is fixed to 3, the problem is not changed at each iteration and the convergence gives a typical monotonous plot.

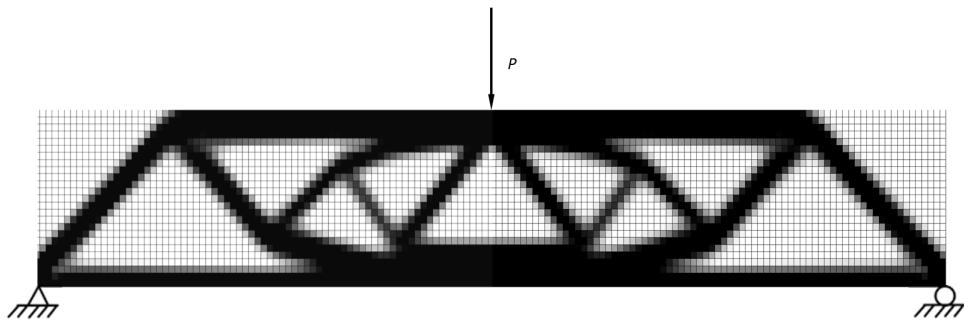
The table below shows the comparison between using the continuation method and using a fixed value of the penalty factor.

**Table 7.2:** Minimum compliance obtained by continuation and a fixed penalty factor  $p=3$ .

| Case                      | Compliance |
|---------------------------|------------|
| <b><math>p = 3</math></b> | 213.00162  |
| <b>Continuation</b>       | 210.86460  |

In order to mitigate the effect of the initial density vector, several attempts were performed. Fortunately, all results using sensitivity filter converged to the same solution.

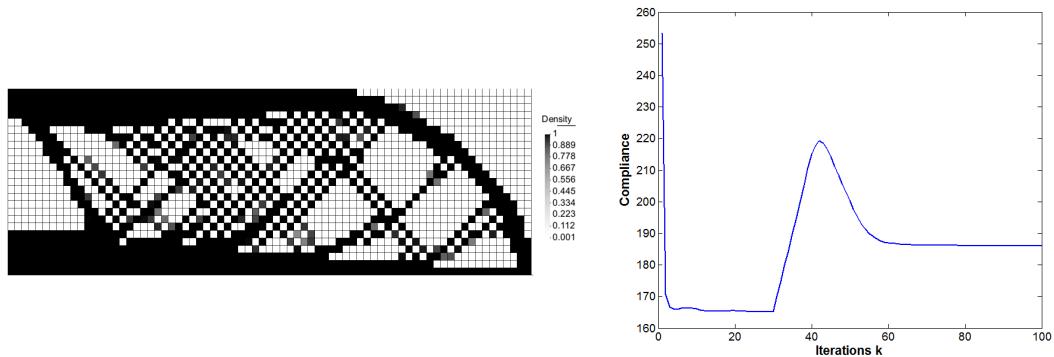
The final MBB optimal design is presented below (Fig. 7.5).



**Figure 7.5:** Final design of the MBB Beam( $150 \times 50$ ),  $R_{min} = 2$  and using a continuation method. Compliance = 210.86460.

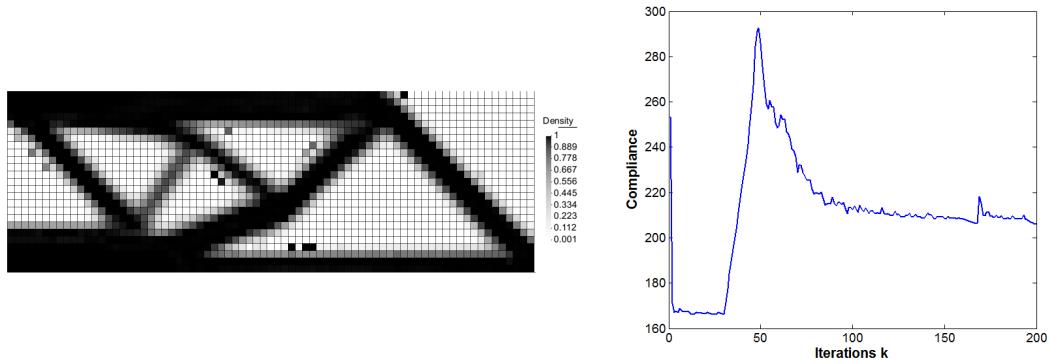
### Method of Moving Asymptotes

In case of MMA the following result was obtained with no filter and the same continuation scheme as the used in OC.



**Figure 7.6:** MBB benchmark using MMA and a continuation method without sensitivity filter.

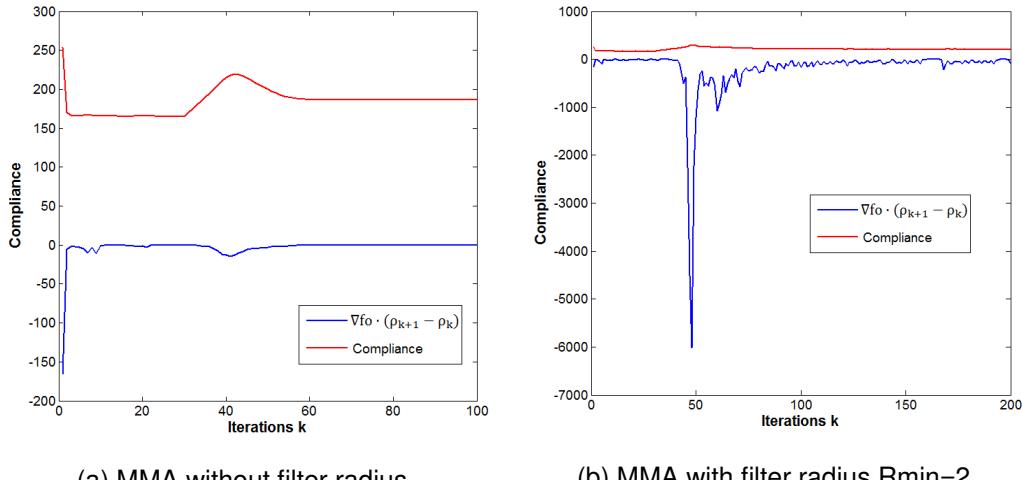
The result obtained is similar to the one obtained using OC method but the compliance is a little lower, so MMA converged to a better solution. Anyway the presence of checkerboard still appears and makes the geometry difficult to read. The following figure (Fig. 7.7) shows the solution after applying a filter radius  $R_{min} = 2$ .



**Figure 7.7:** MBB benchmark using MMA and a continuation method and sensitivity filter  $R_{min} = 2$ .

Although the result is essentially the same as the one obtained using OC, the first aspect to note is the presence of irregularities on the solution. There are elements which have a high density value when they should have almost 0 values. This might have happened due to the oscillating behaviour of the MMA when a sensitivity filter radius is applied. The compliance vs iteration number plot shows local oscillations. This makes the algorithm be non-monotonous, the iteration at  $k+1$  does not always reduce the compliance.

In order to verify results, it has been plotted the value of  $\nabla f_o \cdot (\rho_{k+1} - \rho_k)$  which always have to be negative to minimize the objective function.



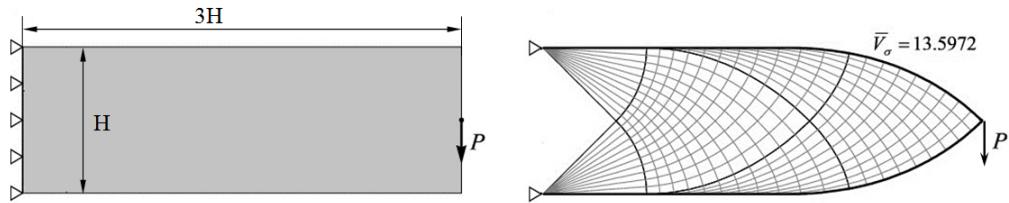
**Figure 7.8:**  $\nabla f_o \cdot (\rho_{k+1} - \rho_k)$  and compliance vs iteration number plot of MBB Beam using MMA.

An important think to note is that the value of  $\nabla f_o \cdot (\rho_{k+1} - \rho_k)$  never becomes positive in none of the cases (a) and (b) of (Fig. 7.8), so mathematically the objective function decreases. However, applying a sensitivity filter the objective function begins to oscillate and it takes values at iteration  $k+1$  which are higher than some values at iteration  $k$ . Thus, the algorithm doesn't take lower compliance values at all iterations.

This fact might be produced by the sensitivity filter radius because it has an heuristic origin. The filter radius does not have a mathematical background that satisfy the mathematical properties of optimization algorithms, it is just a tool used to obtain readable results. The sensitivity of each element is forced to change, so this may be the cause of oscillations.

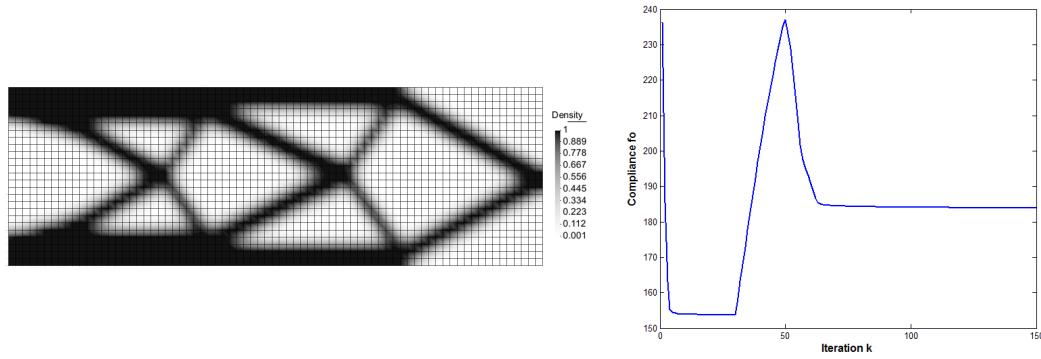
Nevertheless, the filter radius influenced the MMA but not the OC behaviour. This point should be studied carefully in further developments.

### 7.1.2 Long cantilever beam



**Figure 7.9:** Problem definition with reference domain (left) and analytical optimal solution (right) of a long cantilever beam (Rozvany, 1998).

A volume fraction  $V_f=0.5$  and a filter radius  $Rmin=1.5$  were considered in this benchmark. The figure presented below corresponds to the optimal solution using a mesh of 75x25 elements and a continuation scheme.

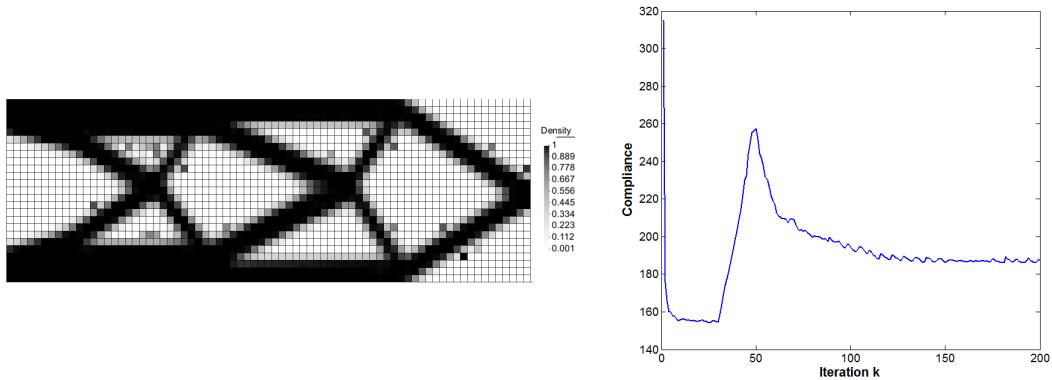


**Figure 7.10:** Long cantilever beam problem (75x25) using OC method, a continuation approach and a filter radius  $R_{min} = 1.5$ .

In this case (Fig. 7.10), the analytical and numerical solutions are very similar. The analytical solution has many secondary beams which together with the primary ones reinforce the whole structure. Otherwise, the numerical solution only takes into account the primary beams and gets rid of the secondary ones.

The convergence plot has the same pattern that the MBB one using a continuation scheme until  $p=3$  with  $\Delta p = 0.1$  and  $k^* = 30$ . The compliance obtained was 183.87377.

In case of MMA (Fig. 7.11), the optimal solution gives a similar result but with some irregularities because of the sensitivity filter. The same conclusions as the MBB beam can be deduced from this test using MMA.



**Figure 7.11:** Long cantilever beam problem (75x25) using MMA method, a continuation approach and a filter radius  $R_{min} = 1.5$ .

The averaged minimum compliance of the above figure is 187, which is close to that given by OC method. As discussed in previous MBB test, local oscillations are present again.

## 7.2 Convergence analysis

### Optimality Criteria

The convergence analysis of the OC method is performed using the MBB benchmark problem with a 30x10 element mesh. Filter radius and a fixed value of  $p$  were set to 2 and 3 respectively.

The stopping criterion used is similar to an L2 norm applied to two consecutive values of the density distribution vector (Eq. 7.1).

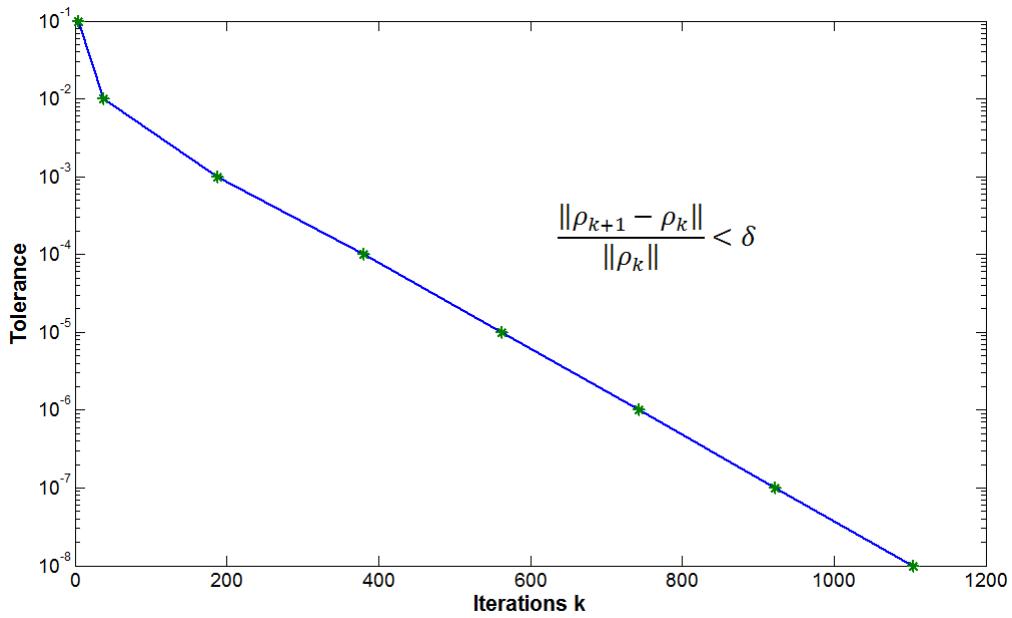
$$\frac{\|\rho_{k+1} - \rho_k\|}{\|\rho_k\|} \leq \delta \quad (7.1)$$

The selected range of tolerances goes from  $10^{-1}$  to  $10^{-8}$  by a factor of  $10^{-1}$ . The results of the number of iterations needed until convergence as well as the value of the compliance for each tolerance are listed in Table (7.3). The reference starting point was set to  $\rho_{0,e} = 0.5$  ( $e = 1, \dots, n.elem.$ ).

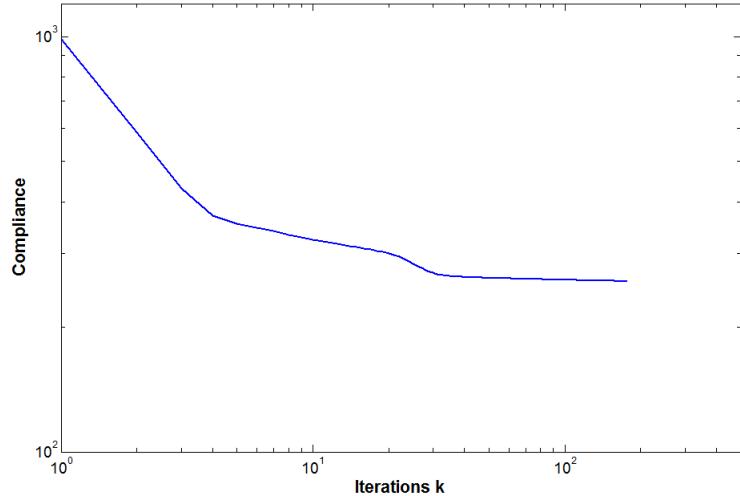
**Table 7.3:** Convergence of MBB Beam (30x10) using OC method.

| Tolerance $\delta$ | Iterations k | Compliance $f_0$ |
|--------------------|--------------|------------------|
| $10^{-1}$          | 4            | 404.05260        |
| $10^{-2}$          | 38           | 292.08218        |
| $10^{-3}$          | 188          | 280.19413        |
| $10^{-4}$          | 381          | 279.08632        |
| $10^{-5}$          | 563          | 278.99278        |
| $10^{-6}$          | 743          | 278.98336        |
| $10^{-7}$          | 923          | 278.98242        |
| $10^{-8}$          | 1104         | 278.98232        |

The logarithmic plots related to the values of the table above are the following,



**Figure 7.12:** Convergence (tolerance) of MBB (30x10),  $\rho_{0,e} = 0.5$ , filter radius  $R_{min} = 2$  and  $p = 3$ .



**Figure 7.13:** Convergence (compliance) of MBB (30x10),  $\rho_{0,e} = 0.5$ , filter radius  $R_{min} = 2$  and  $p = 3$ .

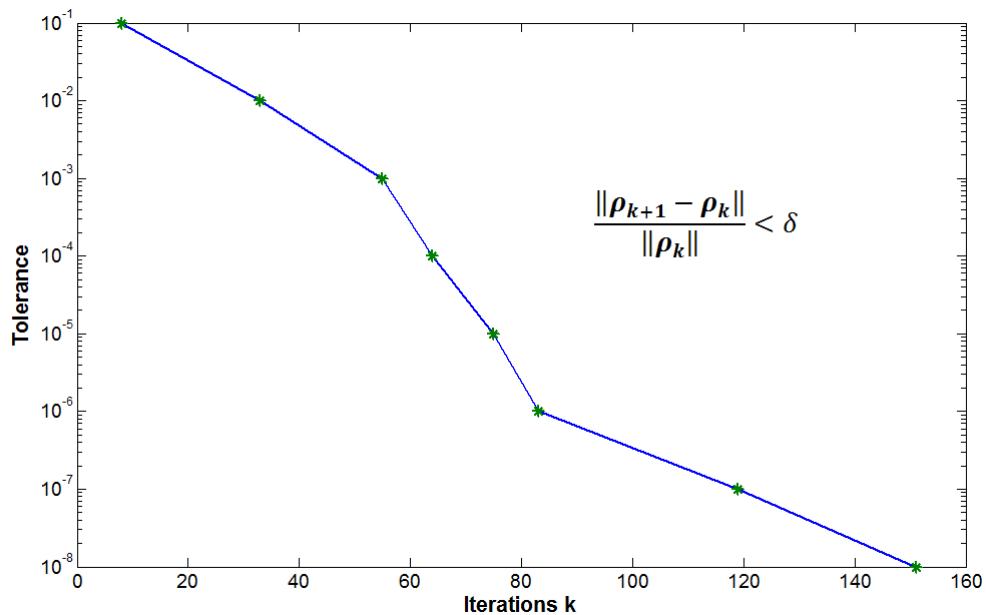
As it can be deduced from Figure (7.3) the rate of convergence in case of the Optimality Criteria method is linear.

## Method of Moving Asymptotes

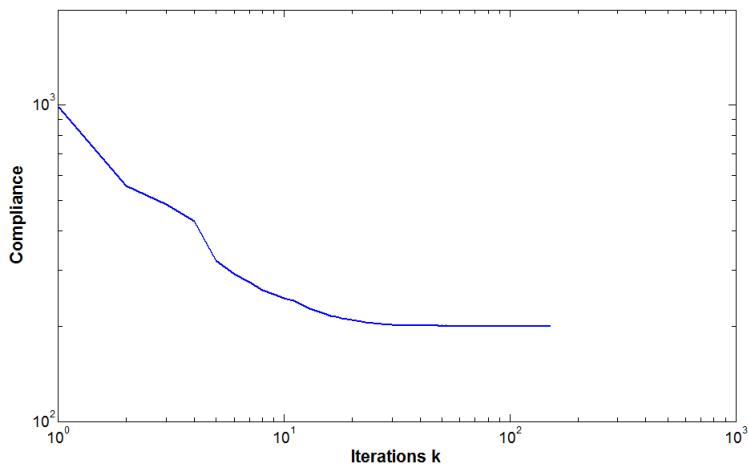
In case of MMA, the same stopping criterion and p value as OC were used. However, due to stability issues of the optimizer, the sensitivity filter was omitted. The initial starting point was also set to  $\rho_{0,e} = 0.5$  ( $e = 1, \dots, n.\text{elem}.$ ).

**Table 7.4:** Convergence of MBB Beam (30x10) using MMA.

| Tolerance $\delta$ | Iterations k | Compliance $f_0$ |
|--------------------|--------------|------------------|
| $10^{-1}$          | 8            | 259.25469        |
| $10^{-2}$          | 33           | 200.76155        |
| $10^{-3}$          | 55           | 200.22737        |
| $10^{-4}$          | 64           | 200.22971        |
| $10^{-5}$          | 75           | 200.22945        |
| $10^{-6}$          | 83           | 200.22904        |
| $10^{-7}$          | 119          | 200.22871        |
| $10^{-8}$          | 151          | 200.22867        |



**Figure 7.14:** Convergence (tolerance) of MBB (30x10),  $\rho_{0,e} = 0.5$ , no filter radius and  $p = 3$ .



**Figure 7.15:** Convergence (compliance) of MBB (30x10),  $\rho_{0,e} = 0.5$ , no filter radius and  $p = 3$ .

Figure (7.14) shows the convergence (tolerance vs iteration number) of the MMA. Although there exist is an unexpected change from tolerance  $10^{-3}$  to  $10^{-6}$ , the algorithm seems to adopt again the initial linear tendency after iteration 83. So, the MMA algorithm adopted in this case (with the dual subproblem solved by a Newton method) seems to have a linear rate. The unexpected change seems to be caused by small local oscillations.

In table (7.4) values of tolerances, iterations and compliance are shown. From the first starting point, the algorithm converges fast to a compliance of about 200.22.

With a tolerance value of  $10^{-3}$  a minimum value of the compliance is reached (200.22737), then the compliance increases (200.22971) and begins to decrease again. These small oscillations near the optimal point might be produced due to the heuristic rule used to update the asymptotes and the bound limits  $\alpha$  and  $\beta$ .

Although the MMA algorithm shows a faster convergence with less iterations than the OC method, the destabilization effect due to the sensitivity filter produces local oscillations and an increase on the global iteration number.

### 7.3 Conclusions

After analysing the benchmark problems, the following conclusions arise:

- The value of the parameters  $p$  and the filter radius  $R_{min}$  affect the optimal result. They may change the convex behaviour of the problem.
- A penalty parameter  $p > 1$  makes the minimum compliance design problem non-convex. Therefore, the optimal solution found may not be the global minimum because the optimization methods used (OC and MMA) are gradient-based.
- In order to mitigate the premature convergence to a local minimum instead of the global one, a continuation method for  $p$  should be used.
- The initial density distribution also affect the solution obtained. Thus, as the behaviour of the problem is not known, a random initial density distribution should be considered.
- Because of the heuristic choice of the filter radius value, some kind of continuation method may be used. However, after several attempts, a value of the radius which encloses the neighbour elements or slightly further seems to work well.
- In order to be more confident about the optimal solution reached, several attempts changing the values of the filter radius and the model of the continuation method might be a good option.
- The initial density vector  $\rho_0$  also affects the solution because the algorithm may converge to a non-optimal stationary point. A suggestion to this fact would be to run with different starting points and compare the compliance obtained at each attempt. The optimal solution would be that which has the lowest value of the compliance.
- The presence of oscillations in the MMA algorithm may be produced by the sensitivity filter radius which change the sensitivities breaking the mathematical background of optimization methods.
- Both OC and MMA methods have a linear convergence behaviour. MMA results faster than OC with a lower iteration number to converge.
- In order to obtain the closest solution to the global one, several tries tuning some parameters are required (continuation scheme, filter radius, starting point...).

- The optimal solution is not always the best one in terms of engineering. Some solutions obtained may have the lowest compliance but difficult to manufacture. In some cases it is better to consider a solution which has the value of the compliance slightly higher but easier to build.

## **Part III**

# **Aeronautical application**

# 8

## Introduction

### 8.1 Structural optimization in the aerospace industry

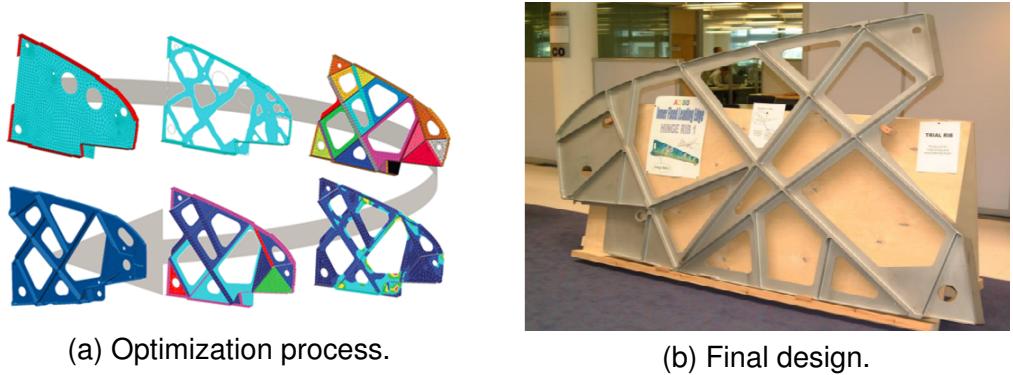
Over the last decades shape optimization brought important benefits and improvements in the design of aircraft structures. However, the design domain of the optimized structure was the same as the initial one with variations on the shape of the holes previously selected.

Topology optimization is able to make even better designs since the reference topologies and optimal domains are allowed to change.

Nowadays, topology optimization is a reality in aerospace industry. Companies like *Airbus*, *Boing* or *EADS* are using these numerical tools in order to obtain better designs and be more competitive.

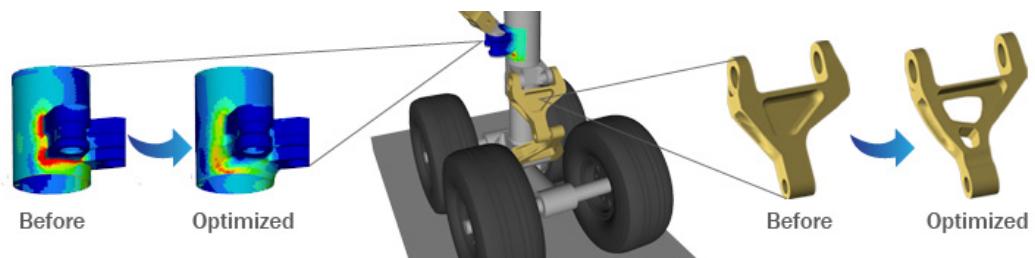
There are many examples of structural topology optimization applied in aerospace structures such as the design process of wings, landing gear systems, aerodynamic surfaces, fuselages...

The following figure (Fig. 8.1) shows the optimization process of a leading edge rib performed by Airbus. Topology optimization is performed in the second step.



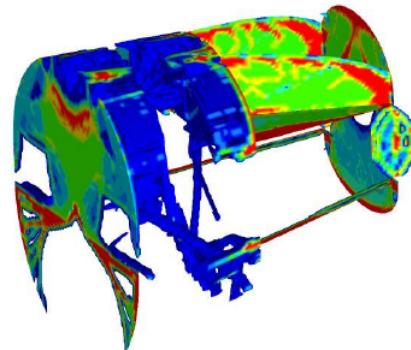
**Figure 8.1:** Optimization process of an A380 leading edge droop nose rib (Krog et al, 2002).

Other examples are the optimization of landing gear components,



**Figure 8.2:** Topology optimization of an aircraft landing gear by Altair OptiStruct.

and the rear part of an A350 fuselage,



**Figure 8.3:** Topology optimization of the rear part of an A350 fuselage (G. Schuhmacher, 2008).

Examples of structural topology optimization in space applications are small satellites such as *FASTSAT-HSV02*.

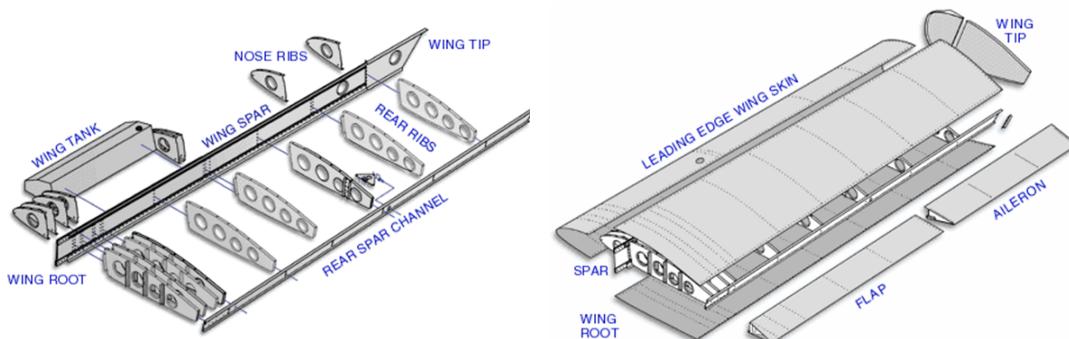
## 8.2 Proposed application

The proposed aeronautical application of this project is to optimize a wing rib structure subject to aerodynamic loads.

Wing ribs are structural pieces that combined with spars and stringers compose the structural part of an aircraft wing. Usually they extent from the leading edge to the trailing edge.

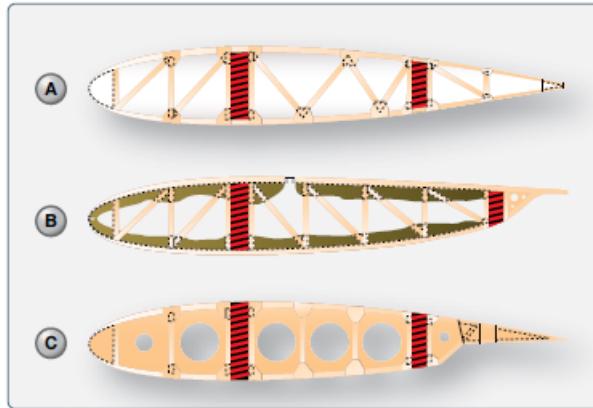
The main functions of wing ribs are listed below:

- Give a cambered shape to the airfoil for aerodynamic reasons
- Transmit aerodynamic loads from the skin and stringers to the spars
- Provide attachment points to other components
- Keep wing spars separated
- Provide booms to the fuel tanks



**Figure 8.4:** Structural parts of an aircraft wing (from <http://www.zamandayolculuk.com>).

Usually, the internal part of wing ribs are emptied with circular holes or made of truss structures so as to reduce weight. Figure (8.5) shows the most common wing rib layouts for a small/medium propelled aircraft.



**Figure 8.5:** Common wing rib structural models of small/medium propelled aircraft. (a)Truss type wing rib structure. (b)Truss type wing rib structure with a continuous gusset. (c) Wing rib structure with circular holes and gusset (Ref. [12]).

The first truss-like wing rib (Fig. 8.5A) offers a good stiffness thanks to the truss geometry with relatively light weight. In order to reinforce the truss, gussets are added.

The second type (Fig. 8.5B) is essentially the same as the previous one but with a continuous gusset. This provides the structure with a higher support and buckling resistance with a little weight increase.

The last one (Fig. 8.5C) is built with several circular holes spread over the wing rib. This structure is a little heavier than the truss-like structure but it is also easier to build.

Wing ribs are usually made of metal and/or wood for small aircraft and light metals, such as aluminium alloys, for larger ones.

The purpose of this section is to obtain an optimal wing rib subject to aerodynamic loads using structural topology optimization.

The selected wing rib model is based on the configuration of a Cessna 172 Skyhawk, a commonly used propelled airplane.

The model 172 is a four-seat, single-engine, high-wing fixed-wing aircraft. It is considered the world's most popular aircraft. As it is said in its webpage: "*With more than 43,000 aircraft with several model variants delivered, the Skyhawk is the best-selling, most-flown plane ever built.*" (Ref. [9])

## 9

## Initial Design

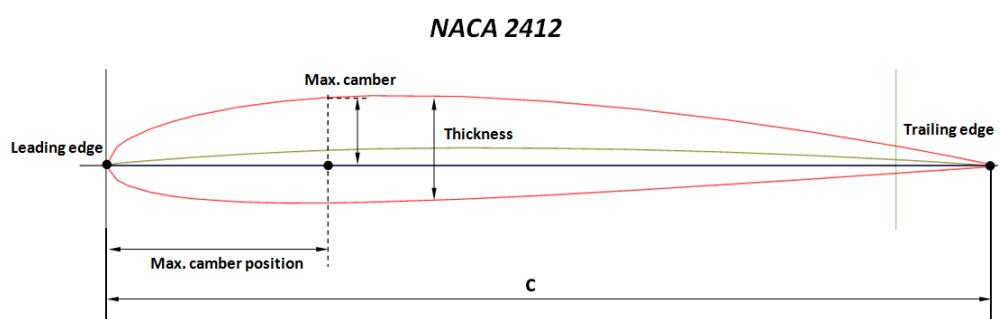
## 9.1 Wing rib geometry

The selected wing rib airfoil model, which takes part in the wing structure of the Cessna 172, is the NACA 2412. All geometrical data of this profile are listed below.

**Table 9.1:** Geometrical parameters of NACA 2412 airfoil.

| Parameters               | Value |
|--------------------------|-------|
| Max. camber <sup>1</sup> | 0.02c |
| Max. camber position     | 0.4c  |
| Max. thickness           | 0.12c |

\*c stands for chord, the line that goes from the leading edge to the trailing edge of the airfoil.

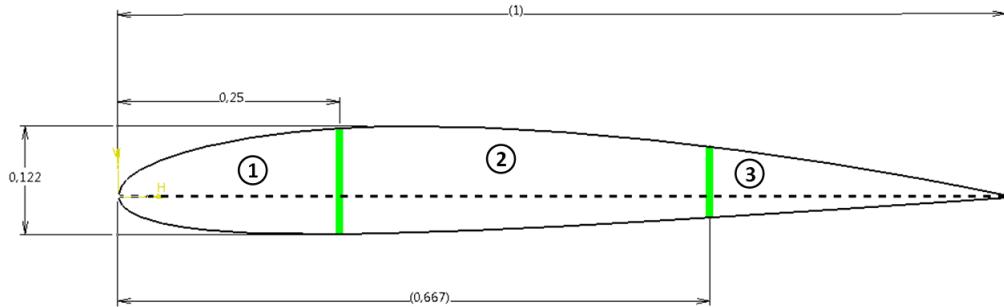


**Figure 9.1:** Representation of the NACA 2412 airfoil.

## 9.2 Wing rib model

### 9.2.1 CAD Model

The proposed wing rib model is divided in three sections which are clamped to a front and a rear spar (Fig. 9.2). Green lines represent front and rear spars and all measures are dimensionlessized with respect to the chord.



**Figure 9.2:** Model of the NACA 2412 airfoil to be optimized. (1) Leading edge rib, (2) Wing box rib, (3) Trailing edge rib.

### 9.2.2 Mesh

All meshes were generated by GiD preprocessor with a Q4 element type. Mesh data of the three sections are shown in the following table,

**Table 9.2:** Mesh Data of the NACA 2412 model.

| Section           | Elements | Nodes |
|-------------------|----------|-------|
| Leading Edge Rib  | 2244     | 2345  |
| Wing Box Rib      | 1992     | 2100  |
| Trailing Edge Rib | 1909     | 2016  |

The meshes of the three different parts can be found in Appendix [A].

### 9.3 Loading conditions

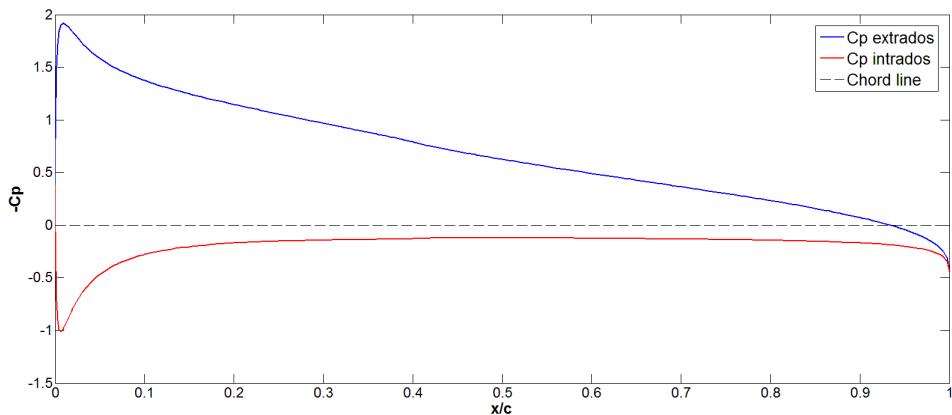
Before defining the wing rib loading conditions, some hypothesis must be considered. The assumptions in which the proposed model is based are listed below:

- Plane stress condition, no stresses act on the normal direction to the wing rib plane.
- Flight conditions according to Cessna 172 Skyhawk technical sheet (Appendix [F]).
- No inertial loading conditions<sup>2</sup>.
- No self-weight loading condition. It is assumed that aerodynamic loads are much higher than wing rib self-weight loads.
- XFLR5 inviscid fluid flow model.

Aerodynamic loads acting on the wing rib are computed using XFLR5 software. It has been considered a Reynolds number of  $5.2 \cdot 10^6$  and a Mach number of 0.2.

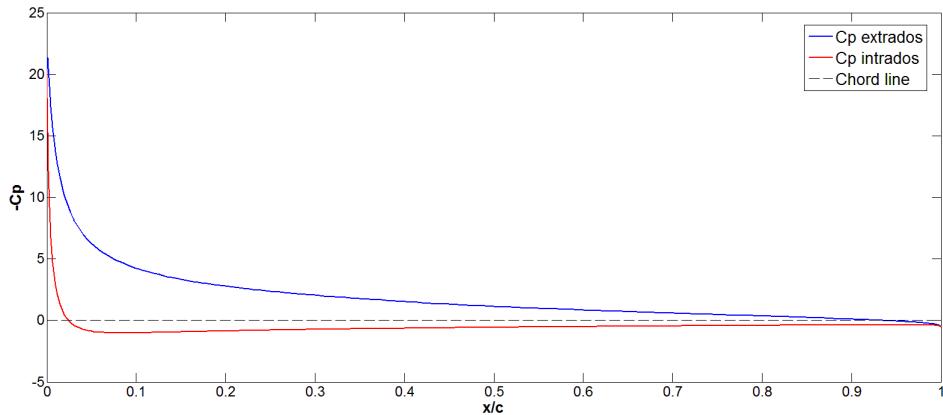
The aerodynamic loading conditions considered in this analysis are those related to a reference angle of attack (AoA)  $\alpha$  of  $5^\circ$  and an angle of attack of  $18^\circ$ , which gives the maximum lift force.

The following figures, (Fig. 9.3) and (Fig. 9.4), show the distribution of the pressure coefficient along the airfoil chord corresponding to an AoA of  $5^\circ$  and  $18^\circ$ .



**Figure 9.3:** Pressure coefficient distribution along NACA 2412 chord at an AoA= $5^\circ$ .

<sup>2</sup>Inertial loads are not taken into account due to the complexity associated to the optimization algorithm and non-convex behaviour of the objective function.



**Figure 9.4:** Pressure coefficient distribution along NACA 2412 chord at an AoA=18°.

In order to calculate the nodal loads acting on the finite element domain, the following process is performed:

- Obtain the pressure coefficient distribution Cp along the airfoil chord at extrados and intrados from XFLR5 software. Appendix [B] contains the Cp distribution values.
- Calculate the pressure distribution at intrados and extrados from Cp values. This step is performed using the following relation:

$$C_p = \frac{p - p_\infty}{\frac{1}{2} \cdot \rho_\infty \cdot V_\infty^2} \quad (9.1)$$

where  $p_\infty = 101325 \text{ Pa}$ ,  $\rho_\infty = 1.225 \text{ kg/m}^3$  and  $V_\infty = 64.83 \text{ m/s}$ . These values give the maximum pressure distribution according to Cessna 172 Skyhawk technical sheet and correspond to ISA atmosphere model.

- Interpolate the pressure nodal values from XFLR5 nodes with MATLAB spline function and evaluate the pressure at finite element nodal coordinates.
- Obtain the nodal aerodynamic force vector on the finite element boundary domain by linearly interpolating (using linear shape functions) the nodal pressure obtained in the previous step and then integrate all over the boundary.

Appendix [D] contains the code that calculates the nodal aerodynamic force vector given the Cp distribution from XFLR5 software.

# 10

## Optimized Design

As discussed in the previous chapter, the wing box geometry is divided into three parts. The three parts has been optimized separately due to the different boundary conditions applied to each one. In addition, the partition of the airfoil allows to use finer meshes than in the case of optimizing the whole airfoil.

The optimization method applied to each wing rib component is the OC method due to the stable behaviour against oscillations produced by the filtering technique.

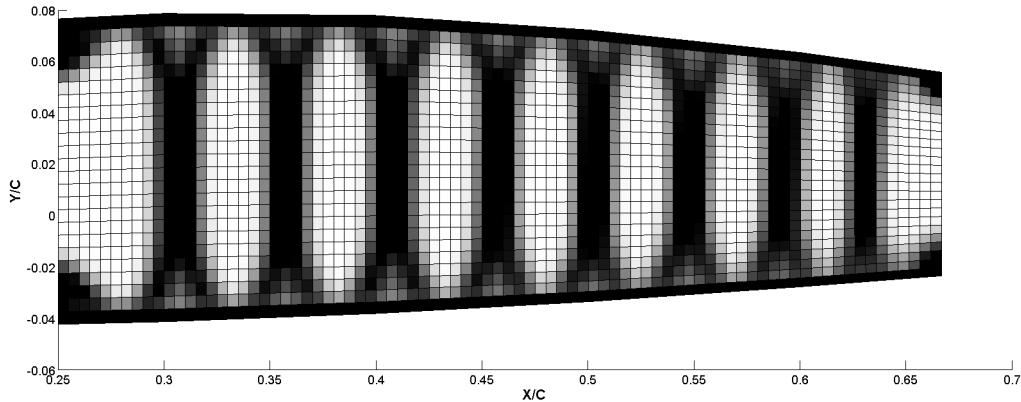
The material used for the whole wing rib structure is the aluminium alloy 7075-T6, a commonly used alloy in aeronautical structures. The parameters used are listed below:

**Table 10.1:** Wing rib global input parameters.

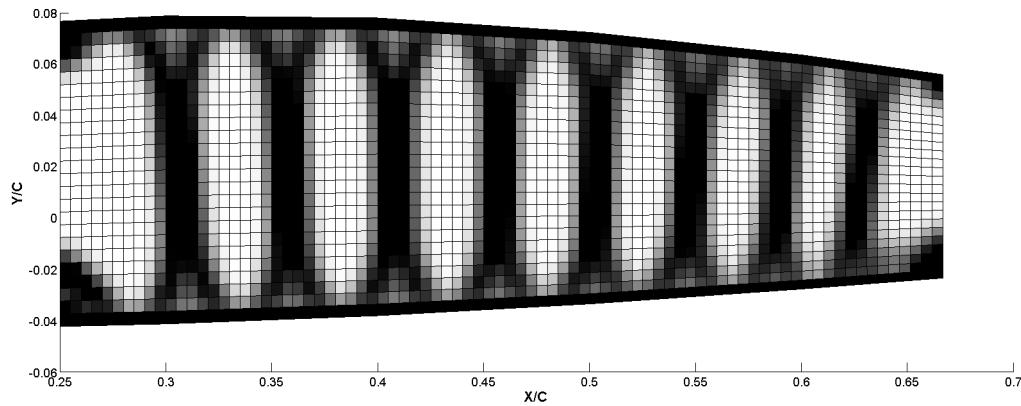
| Parameter                   | Value            |
|-----------------------------|------------------|
| Young's modulus $E$         | 100 GPa          |
| Poisson's coefficient $\nu$ | 0.33             |
| Initial density $\rho_0$    | Random           |
| Volume fraction $V_f$       | 0.5              |
| Load $P$                    | Pressure (XFLR5) |

### 10.1 Wing Box

The following results were obtained after optimizing the wing box section of the NACA 2412. Both the left side and the right side are clamped to the front and rear spars respectively. In addition, the density of upper and lower boundary elements has been forced to 1 to ensure solid material on that region.



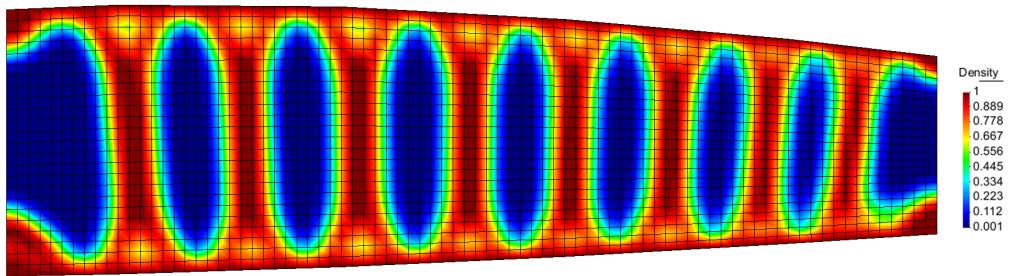
**Figure 10.1:** Optimized wing box design at an AoA = 5°.



**Figure 10.2:** Optimized wing box design at an AoA = 18°.

The optimal solution in both cases, AoA=18° and AoA=5°, look almost identical because the applied load is approximately linear (both cases) in this section. The result obtained is formed by vertical beams which absorb the pressure loads which compress the structure.

The following picture (Fig. 10.3) shows the final result with a smoothing filter and a colour distribution to make it more visual. This figure correspond to the critical case (AoA=18°).



**Figure 10.3:** Final wing box design at an AoA = 18°.

Stress distribution in X/Y direction and Von Mises can be found in Appendix [E]. The maximum Von Mises stress resulted 1.4205 MPa which is much lower than the yield strength of the aluminium alloy 7075-T6 ( $\sigma_Y = 500$  MPa).

However, using SIMP the stress distribution is distorted due to the presence of intermediate densities. For this reason, the correct way to perform a stress analysis would be to make a CAD model from the qualitative model obtained by topology optimization and then use a FEA software to analyse the stress distribution.

Buckling was not taken into account in the minimum compliance formulation. Therefore, it must be analysed if the optimal structure would collapse because of this instability. In any case, buckling isn't usually considered in this phase of design and it is carried out by sizing or shape optimization in future design stages.

## 10.2 Leading Edge

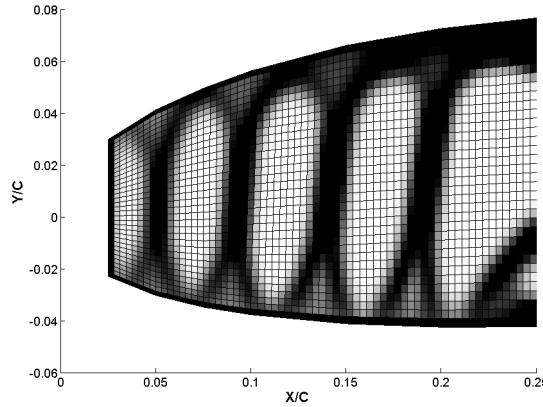
In this case, the optimization algorithm was applied to the AoA = 18° case because this section is loaded more severely at this angle of attack. Anyway, the results obtained in both cases doesn't differ much from each other. This section is clamped to the front spar and boundary elements have been forced to be solid.

It was also considered to remove the front side in order to prevent irregularities on the mesh geometry and let the preprocessor make a structured grid. In any case, the removed part is completely solid.

Unlike the wing box section case, the filter radius has a lot of incidence on the optimized designs. For this reason, many tries with different radii were performed until obtain a reasonable result.

Appendix [E] contains all results, in the Leading Edge section, at different filter radius values.

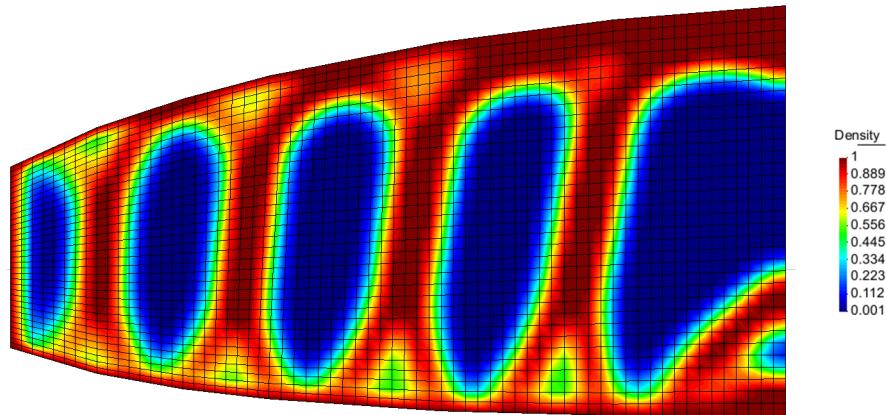
The final design follows the same pattern as the other sections and the algorithm tries to generate vertical beams in order to absorb the applied pressure loads.



**Figure 10.4:** Optimized leading edge design at an AoA = 18° and Rmin = 0.008.

The maximum Von Mises stress gave  $\sigma_{VM} = 3.1342$  MPa, which is also much lower than the yield limit of the material.

The final smoothed result of the leading edge section is presented below,



**Figure 10.5:** Final leading edge design at an AoA = 18° and Rmin = 0.008.

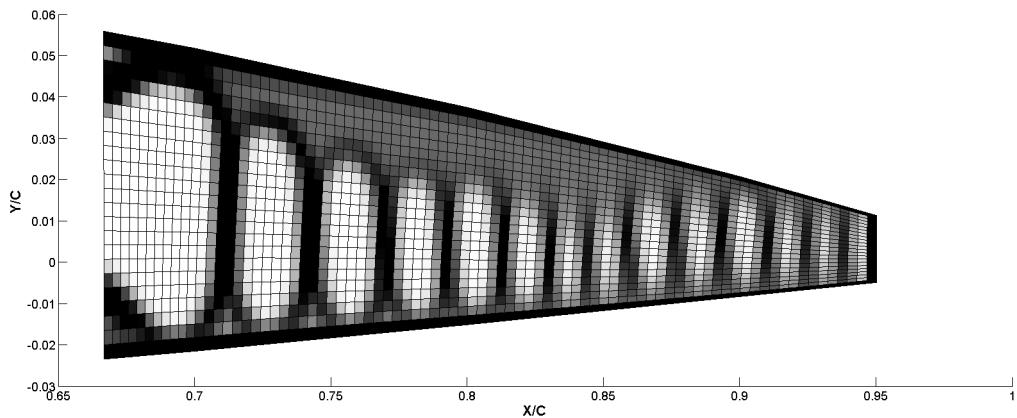
### 10.3 Trailing Edge

The trailing edge is clamped to the rear spar and free everywhere else. It was considered to remove the sharp end of this section in order to avoid mesh irregularities.

As well as the wing box section the boundary elements were forced to be solid. This case brought some problems to obtain fully interpretable optimal designs. Several filter radius were applied to find a reasonable solution. However, no one gave a clear result.

Appendix [E] collects all tests that were performed at angles of  $5^\circ$  and  $18^\circ$ . Both cases offer similar designs using the same values of the filter radius, but any case gives a clear 0-1 solution.

The best solution obtained was the one corresponding to a  $R_{min} = 0.005$ . As the two cases of angle of attack are similar, the chosen solution was the critical case (AoA  $18^\circ$ ). The following figure shows the result.



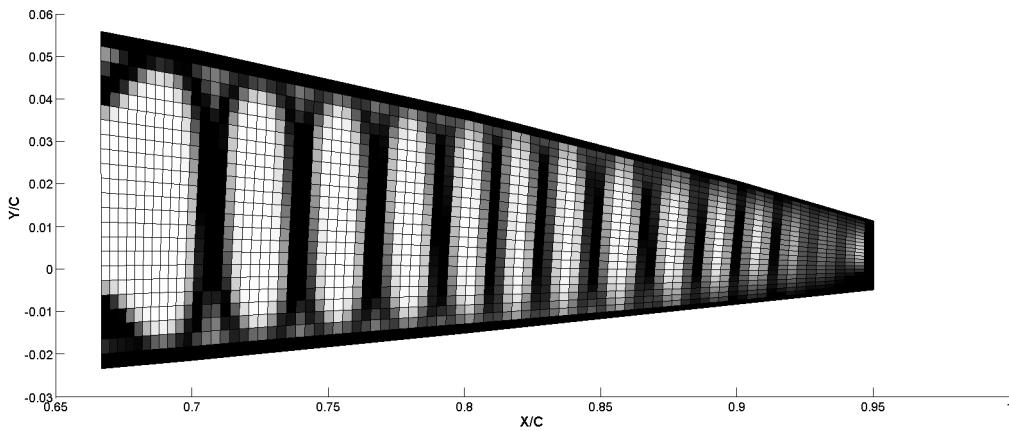
**Figure 10.6:** Optimal trailing edge design at an AoA =  $18^\circ$  with  $R_{min} = 0.005$ .

As one can see, the top of the leading edge has intermediate density values. Although it isn't clear if that region should be empty or solid, the same pattern of vertical beams arranged between elliptical holes as in the wing box case can be observed.

Some tests were performed to see the influence of the geometry or the load distribution shape on the optimal design. A simplified quadrilateral geometry of the original trailing edge and a uniform load distribution with the same overall force than the real case were tested. However, the same pattern of intermediate densities appeared.

Other possible causes of this problem could be the fact that, as the load is not dimensionless, it was too low to be taken into account by the optimizer. In any case, after testing the same design with the same load distribution augmented by 100, there wasn't any improvement.

The last approach to get rid of this problem was to clamp the free end. Although this approach differs a little from the real case, the optimal solution gave similar results to the wing box section. The result is shown in (Fig. 10.7).

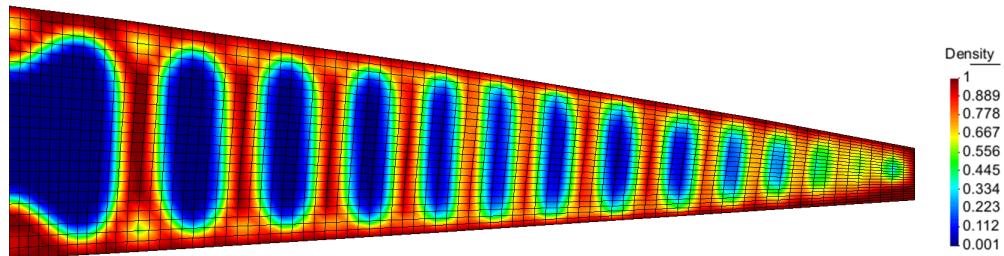


**Figure 10.7:** Optimal trailing edge design at an AoA = 18° with Rmin = 0.005 and clamped ends.

Both solutions in figures (Fig. 10.6) and (Fig. 10.7) are similar except for the upper side, where the solution with clamped ends has a clear interpretation.

According to results of the wing box and the leading edge sections, which have a similar pattern, the final design considered for the trailing edge is the case with clamped ends.

The trailing edge design, smoothed and coloured by a post-processing filter, is shown in the following figure.



**Figure 10.8:** Final trailing edge design at an AoA = 18° and Rmin = 0.005.

As in previous cases, the maximum Von Mises stress is much lower than the elastic limit.

## 10.4 Final design

The proposed final wing rib design, after applying structural topology optimization methods, is presented in (Fig. 10.9). This model was obtained from the optimization of each section of the wing rib.

The conversion from an almost a 0-1 design to a fully 0-1 design was performed by contouring and removing all regions with low density values and leaving those ones up to a certain value.

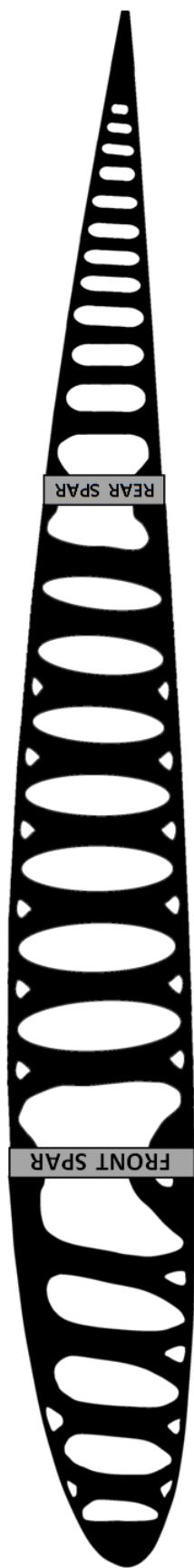


Figure 10.9: Final wing rib design.

## **Part IV**

# **Project Management**

# 11

## Budget

This section focuses on the economic issues of the project. All material and devices used to develop the project have been taken into account in order to present a final budget. It has also been pointed out needed material with no direct charge.

The final budget is divided in four subsections: working hours, software license, hardware expenses and technical literature. The full breakdown of the budget can be found on Budget volume.

### Total Budget

**Table 11.1:** Total cost.

| Concept              | Cost [€]     |
|----------------------|--------------|
| Working Hours        | 6832         |
| Software Licenses    | 2000         |
| Hardware expenses    | 1470         |
| Technical Literature | 56           |
| <b>TOTAL</b>         | <b>10358</b> |

# 12

## Environmental implications

- **About the project**

The present project has been entirely a theoretical study of numerical techniques. No environmental damage has been produced during the development of this work apart from the energy expenses when using computational tools.

The only concerns may be related to computational efficiency in the sense of reducing computing time when performing simulations. At any rate, only domestic computers with relatively low power requirements have been taken into account.

- **About the implications of structural optimization tools on the environment**

The application of structural topology optimization tools in the industry has a huge relevance in environmental issues.

In many cases, structural optimization implies lighter structures and a reduction on the material used. Therefore, this fact not only affects economical savings but also a more efficient way to use resources and reduce the residuals produced in the manufacturing process. An optimal structure usually needs less material and less energy to be built.

Other implications in the automotive and aerospace industry must also be taken into account.

On the one hand, a lighter structure in the automotive field means less friction and therefore less fuel consumption and environmental damage.

On the other hand, the same happens in the aerospace field, when engineers are always seeking for reducing weight. The use of structural optimization tools may help to achieve this purpose. A weight reduction on an aircraft means that less thrust is needed, less fuel will be consumed and less amount of pollution will be released.

# 13

## Planning

### 13.1 Work Breakdown Structure

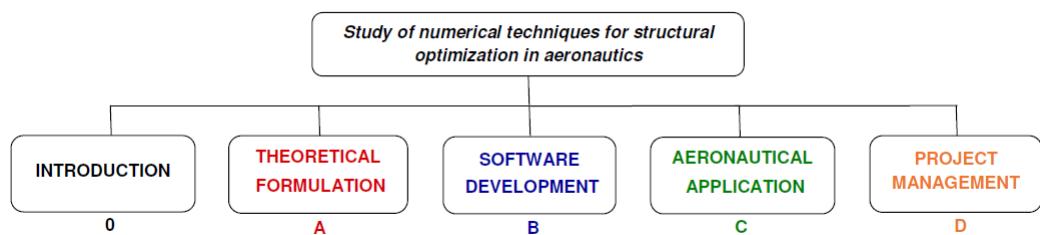


Figure 13.1: Main parts of the project.

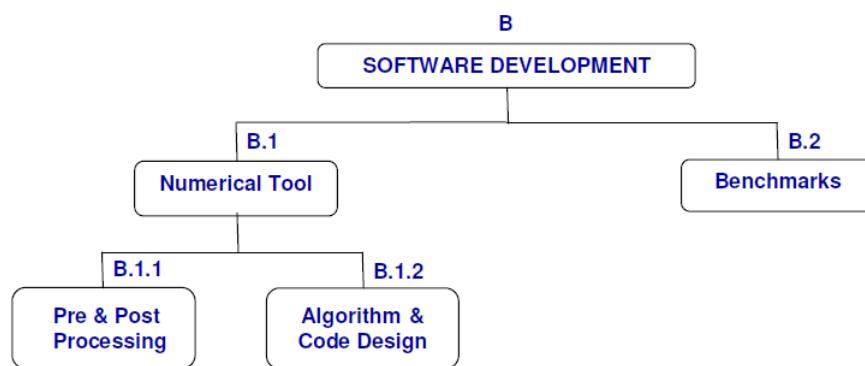
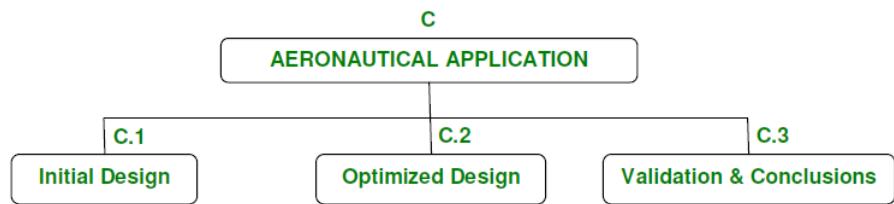
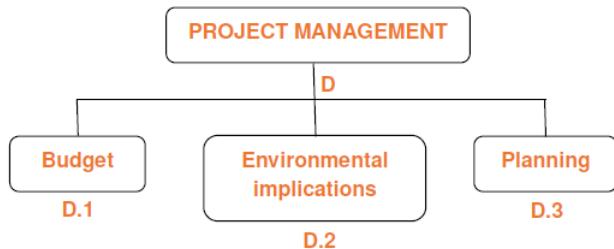


Figure 13.2: Software development tree.



**Figure 13.3:** Aeronautical application tree.



**Figure 13.4:** Project management tree.

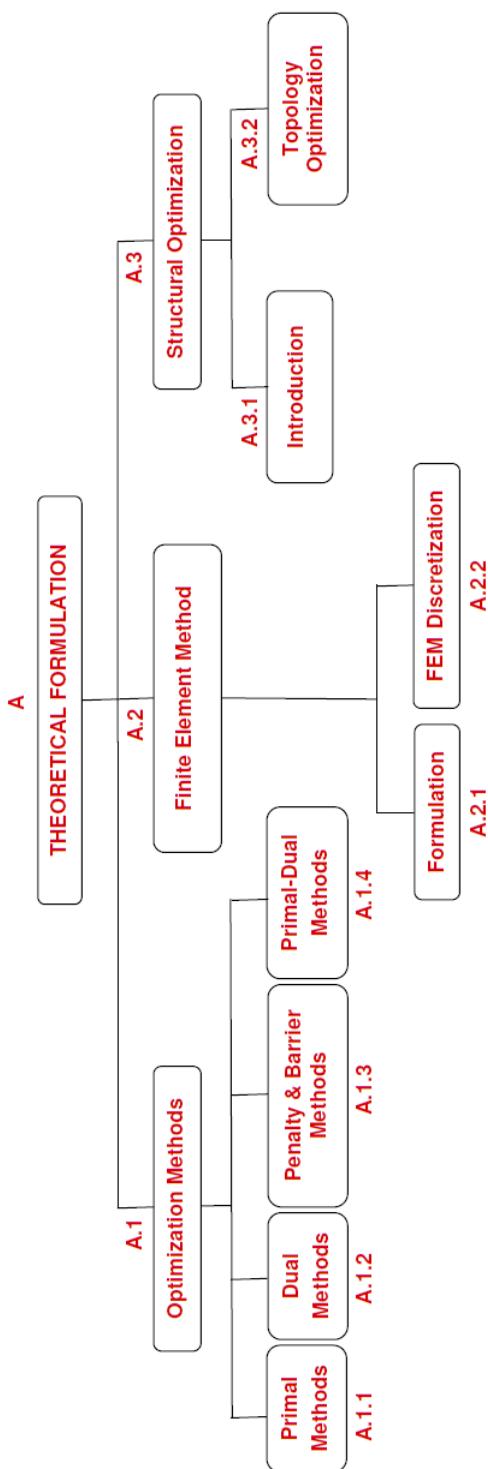


Figure 13.5: Theoretical formulation tree.

## 13.2 Gantt Chart

### 13.2.1 Tasks

**Table 13.1:** Task Breakdown of the project

| Name                            | Start Date | End Date |
|---------------------------------|------------|----------|
| Assignment of TFG               | 2/7/14     | 2/7/14   |
| Information research            | 2/7/14     | 3/2/14   |
| Start of TFG                    | 3/3/14     | 3/3/14   |
| Development of Project Charter  | 3/3/14     | 3/9/14   |
| Project Charter Delivery        | 3/10/14    | 3/10/14  |
| Theoretical Formulation         | 3/10/14    | 3/31/14  |
| Optimization Methods            | 3/10/14    | 3/24/14  |
| Linear Elasticity & FEA         | 3/17/14    | 3/31/14  |
| Topology Optimization           | 3/24/14    | 3/31/14  |
| Software Development            | 4/1/14     | 4/30/14  |
| Algorithm & Code Design         | 4/1/14     | 4/22/14  |
| Pre & Post Processing           | 4/8/14     | 4/22/14  |
| Test application & validation   | 4/23/14    | 4/30/14  |
| Aeronautical Application        | 5/2/14     | 5/23/14  |
| Initial & Optimized Design      | 5/2/14     | 5/16/14  |
| Validation & Conclusions        | 5/17/14    | 5/23/14  |
| Project Management              | 5/2/14     | 5/31/14  |
| TFG draft delivery              | 6/2/14     | 6/2/14   |
| TFG revision                    | 6/2/14     | 6/22/14  |
| Preparation of TFG presentation | 6/23/14    | 7/6/14   |
| TFG Delivery                    | 6/23/14    | 6/23/14  |
| End of TFG                      | 7/7/14     | 7/7/14   |

### 13.2.2 Diagram

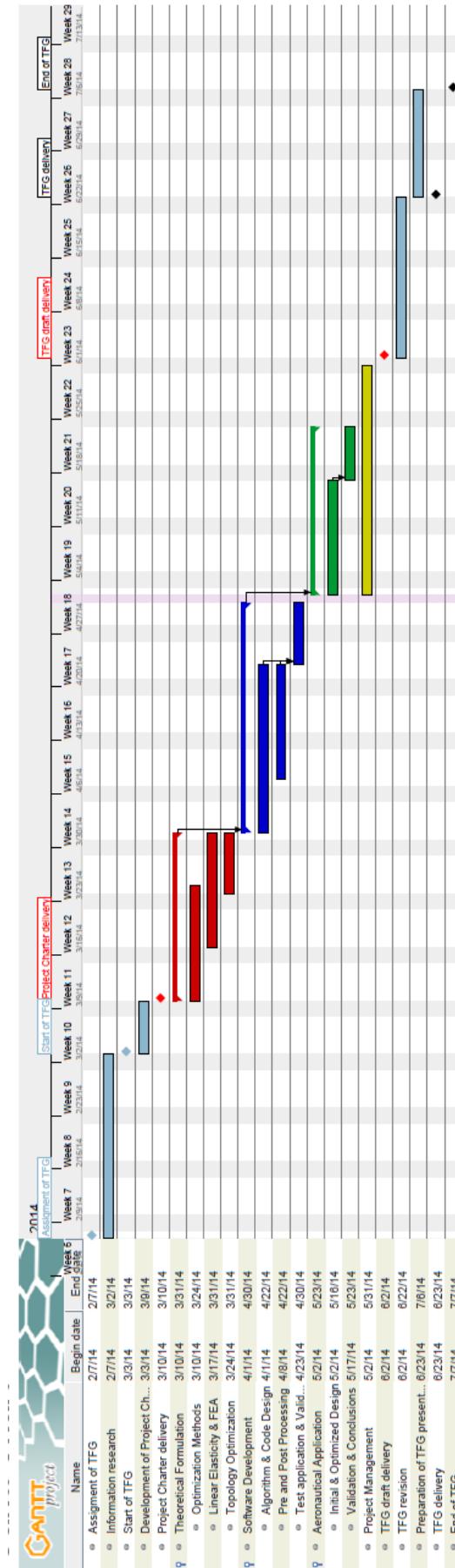


Figure 13.6: Gantt Chart of the project.

# 14

## Conclusions and future development

The present project was focused on the study of numerical techniques applied to structural optimization. Specifically, structural topology optimization was only taken into account because is the most general of the different structural optimization types.

The minimum compliance design formulation and the SIMP method were considered to optimize reference domains subject to loads and boundary conditions by means of material distribution. The SIMP approach is a good solution to avoid integer programming algorithms which can be difficult to implement. However, the drawback of this method is that there exist numerical instabilities which lead to inconsistent or non-interpretable topologies.

In order to overcome these drawbacks a filtering technique and a continuation approach were implemented. Although the sensitivity filter is effective in removing checkerboard patterns and mesh-dependency instabilities, it is heuristic without any mathematical optimization background. Thus, many tries with different values of the radius must be performed until obtain a reasonable solution. The effectiveness of the continuation method was successfully tested in the benchmarks section.

Two common optimizers, OC method and MMA, were used to minimize the compliance problem. These two methods have heuristic parameters and they need to be tuned.

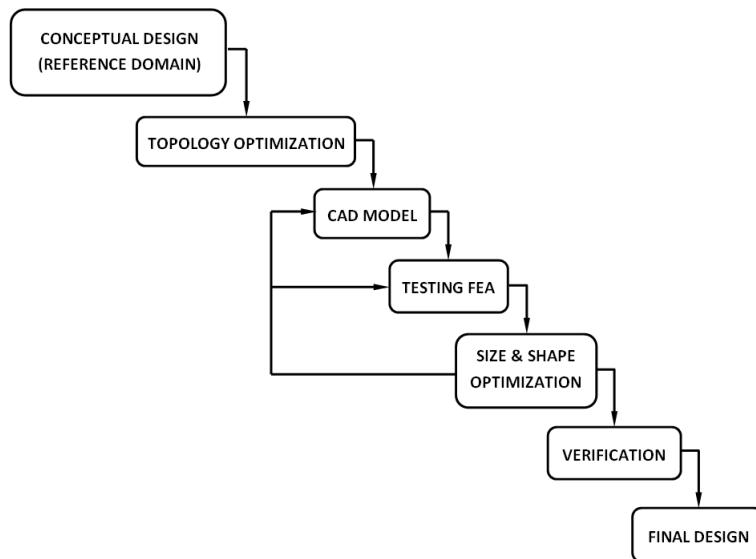
While the OC method is stable with a monotonous behaviour, the MMA is significantly faster but presents oscillations due to the heuristic sensitivity filter and leads to certain irregularities on the optimal material distribution.

The chosen aeronautical structure to be optimized was a wing rib, based on a Cessna 174 Skyhawk aircraft, subject to an aerodynamic load. The wing rib was divided into three sections which are clamped to a front and rear spars: leading edge, wing box and trailing edge.

Several attempts were performed with different filter radius until obtain interpretable and feasible solutions. While the wing box and leading edge structures gave reasonable optimal solutions, the trailing edge presented some irregularities. These complications could be produced by the type of load applied and/or its geometry. Nevertheless, some tests with a simpler geometry and an equivalent uniform load were performed with the same kind of irregularities.

For this reason, the final trailing edge design was obtained taking into account the results of the leading and wing box sections, ignoring the irregularities on the solution of the trailing edge and using the two clamped ends approach. In order to validate the proposed wing rib, a CAD model and a finite element analysis should be performed.

Topology optimization was applied to obtain a starting design point. The proposed optimal wing rib design is just an first estimation. The next steps and the general process, as suggested by Refs. [24, 34], to reach the final applicable design are shown in the following diagram.



**Figure 14.1:** General design process with structural optimization.

After the topology optimization process, the result must be understood and interpreted. Then a CAD model and a posterior validation by performing a finite element analysis is required. After that, the next step is the concept development considering additional engineering requirements such as buckling, fatigue or fracture constraints by using sizing and shape optimization techniques.

For example, sizing optimization can be formulated in terms of cross-sections of beams. Adding the buckling critical load as a constraint, the optimal cross-section would avoid this instability. In case of the final wing rib design, sizing optimization would find the optimal section of the vertical beams in order not to reach the buckling critical load.

Finally, the last step before getting the final design is to verify the model and consider if it is feasible to be build.

In some cases the final geometry is too complex and expensive to build. However, thanks to the advances on 3D-printing techniques with different materials, complex geometries are becoming viable and easier to handle.

Structural topology optimization method should be taken into account in any design process of structural components because it speeds up the design phase and this means savings in time and costs. In this manner, the concept design can be based on results of a computational optimization rather than on estimations, benchmarks or previous similar cases.

However, these methods are relatively recent and need to be improved with more emphasis on the mathematical properties in order to be more robust with less heuristic dependency.

## Bibliography

- [1] ALTAIR ENGINEERING, INC. *Hyperworks Release 11 – User’s manual*, release 11 ed. ALTAIR Engineering, Inc, Michigan, USA, 2009.
- [2] ANDREASSEN, E., CLAUSEN, A., SCHEVENELS, M., LAZAROV, B. S., AND SIGMUND, O. Efficient topology optimization in matlab using 88 lines of code. *Structural and Multidisciplinary Optimization* 43, 1 (2011), 1–16.
- [3] BATHE, K. *Finite Element Procedures*, 1st edition ed. Prentice Hall, USA, 1996.
- [4] BENDSÆ, M., AND SIGMUND, O. *Topology Optimization: Theory, Methods and Applications*, 2nd edition ed. Springer, Germany, 2003.
- [5] BOYD, S. P., AND VANDENBERGHE, L. *Convex optimization*. Cambridge university press, 2004.
- [6] BROWNE, P. *Topology Optimization of Linear Elastic Structures*. Ph.d. thesis, University of Bath, Bath, UK, 2013.
- [7] BRUYNEEL, M., DUYNSINX, P., AND FLEURY, C. A family of MMA approximations for structural optimization. *Structural and Multidisciplinary Optimization* 24, 4 (2002), 263–276.
- [8] CESSNA. *SkyHawk 172: Specification and Descripton*. Cessna, Textron, 2012.
- [9] CESSNA AIRCRAFT COMPANY. Skyhawk - the best-selling and most-flown aircraft ever built. (2013, January 1). Cessna. Retrieved May 15, 2014, from <http://www cessna com/single-engine/skyhawk?trk=NavSingle>, 2013.
- [10] CHRISTENSEN, P., AND KLARBRING, A. *An Introduction to Structural Optimization*, vol. 153. Springer, 2008.

- [11] DUYNSINX, P., BRUYNEEL, M., AND FLEURY, C. Solution of large scale optimization problems with sequential convex programming. *Danish Center For Applied Mathematics and Mechanics* (2009).
- [12] FEDERAL AVIATION ADMINISTRATION. *Aviation Maintenance Technician Airframe Handbook*, vol. 1. FAA, USA, 2012.
- [13] FELIPPA, C. A. Introduction to finite element methods. *University of Colorado, Boulder, http://www. colorado. edu/engineering/CAS/courses. d/IFEM. d* (2004).
- [14] HANSON, M., AND MOND, B. Necessary and sufficient conditions in constrained optimization. *Mathematical programming* 37, 1 (1987), 51–58.
- [15] HUGHES, T. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*, 1st edition ed. Dover, USA, 1987.
- [16] JAMES, K. *Aerostructural Shape and Topology Optimization of Aircraft Wings*. PhD thesis, University of Toronto, 2012.
- [17] KOKO, J. Vectorized matlab codes for linear two-dimensional elasticity. *Scientific Programming* 15, 3 (2007), 157–172.
- [18] KROG, L., TUCKER, A., AND ROLLEMA, G. Application of topology, sizing and shape optimization methods to optimal design of aircraft components. In *Proc. 3rd Altair UK HyperWorks Users Conference* (2002).
- [19] LI, L., AND KHANDELWAL, K. Two-point gradient-based MMA (TGMMA) algorithm for topology optimization. *Computers & Structures* 131 (2014), 34–45.
- [20] LUENBERGER, D., AND YE, Y. *Linear and Nonlinear Programming*, 3rd edition ed. Springer, USA, 2008.
- [21] MA, Z.-D., PIERRE, C., RAJU, B., AND KIKUCHI, N. Multidomain topology optimization for structural and material designs. *Journal of applied mechanics* 73, 4 (2006), 565–573.
- [22] RIETZ, A. Sufficiency of a finite exponent in SIMP (power law) methods. *Structural and Multidisciplinary Optimization* 21, 2 (2001), 159–163.
- [23] ROZVANY, G. I. A critical review of established methods of structural topology optimization. *Structural and Multidisciplinary Optimization* 37, 3 (2009), 217–237.

- [24] SCHUHMACHER, G. Numerical optimization methods in the aerospace design process. In *2nd European Hyperworks Technology Conference* (2008).
- [25] SHEVADE, S. Numerical optimization[Video]. <http://www.youtube.com/playlist?list=PL6EA0722B99332589>, 2012.
- [26] SIGMUND, O. A 99 line topology optimization code written in matlab. *Structural and Multidisciplinary Optimization* 21, 2 (2001), 120–127.
- [27] SIGMUND, O., AND PETERSON, J. Numerical instabilities in topology optimization: A survey on procedures dealing with checkerboards, mesh-dependencies and local minima. *Structural Optimization* 16 (1998), 68–75.
- [28] SINGIRESU, S. *Engineering Optimization: Theory and Practice*, 4th edition ed. John Wiley & Sons, USA, 2009.
- [29] SVANBERG, K. The method of moving asymptotes - a new method for structural optimization. *International Journal for Numerical Methods in Engineering* 24 (1987), 359–373.
- [30] SVANBERG, K. On the convexity and concavity of compliances. *Structural and Multidisciplinary Optimization* 7, 1 (1994), 42–46.
- [31] SVANBERG, K. MMA and GCMMA. Research report, Optimization and Systems Theory, Sweden, 2007.
- [32] VINSON, J. *Plate and Panel Structures of Isotropic, Composite and Piezoelectric Materials, Including Sandwich Construction*, 1st edition ed., vol. 120. Springer Netherlands, Netherlands, 2005, ch. Chapter 8: Theorem of Minimum Potential Energy, Hamilton's Principle and Their Applications, pp. 121–141.
- [33] WANG, S., AND TAI, K. Bar-system representation for topology optimization using genetic algorithms. *Engineering computations* 22, 2 (2005), 206–231.
- [34] WENJIONG, G. On challenges and solutions of topology optimization for aerospace structural design. *10th World Congress on Structural and Multidisciplinary Optimization* (2007).
- [35] ZHU, J., TAYLOR, R., AND ZIENKIEWICZ, O. *The finite element method: its basis and fundamentals*, 6th edition ed. Butterworth-Heinemann, UK, 2005.