



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Escola Superior d'Enginyeries Industrial,
Aeroespacial i Audiovisual de Terrassa

Study of new design of lightweight structures for maximizing first eigenvalues

Document:

Bachelor final thesis report

Author:

Bernat Descalzi Marron

Director:

Àlex Ferrer Ferré

Degree:

Bachelor's degree in Aerospace Technology Engineering

Examination session:

Spring 2023

BACHELOR FINAL THESIS

Abstract

Structural optimization and design of lightweight structures is a crucial field of study in the aeronautical industry, especially in the transition towards greener ways of transport.

The main purpose of this thesis is to study the buckling optimizing problem of a clamped-clamped beam. The aim is to maximize the load at which the column begins to bend, which can be regarded as an eigenvalues problem, where the first eigenvalue is the critical buckling load and the eigenvectors are the buckling modes. This optimization problem is solved using an object-oriented code developed inside the *Swan* framework, using pre-implemented algorithms, namely the *fmincon* and *MMA* optimizers within *Swan*.

Based on previous works in the field, which were focused on the circular cross-section column, in this thesis new column cross-sections are proposed and studied, such as the rectangular section and the circular section with an inner hole. The obtained results are analysed using a sensitivity study to assess the influence of different initial parameters.

Resum

L'optimització estructural i el disseny d'estructures lleugeres és un àmbit d'estudi crucial en la indústria aeronàutica, especialment en la transició cap a mitjans de transport més ecològics.

El principal objectiu d'aquesta tesi és estudiar el problema d'optimització del vinclament d'una viga bi-encastada. L'objectiu és maximitzar la càrrega a la qual es produeix el vinclament, que es pot considerar un problema de valors propis, on el primer valor propi és la càrrega crítica de vinclament i els autovectors són els modes de vinclament. Aquest problema d'optimització es resol utilitzant un codi orientat a objectes desenvolupat dins del marc de treball *Swan*, utilitzant algorismes pre-implementats, concretament els optimitzadors *fmincon* i *MMA* dins de *Swan*.

Basant-se en els previs estudis en aquest àmbit, que majoritàriament es centren en la columna de secció circular, en aquesta tesi es proposen i s'estudien nous tipus de secció transversal, com ara la secció rectangular i la secció circular amb un forat concèntric interior. Els resultats obtinguts són analitzats utilitzant un estudi de sensibilitat per avaluar la influència de diferents paràmetres inicials.

Acknowledgements

I would like to express my most heartfelt gratitude to my director Dr. Àlex Ferrer Ferré. His support and encouragement were invaluable throughout the entire development of the thesis, and I could not have completed this project without his expertise, guidance and dedication.

I would also like to express my sincere gratitude to Aitziber Mañeru for her valuable assistance during the initial phases of the project. Her prompt responses to my inquiries, particularly on the topic of the introduction to *Swan* and her code, allowed me a fast adaptation to the project.

Finally, I would like to thank my family and friends for their support during this months and for always believing in me.

Contents

1 Introduction	8
1.1 Aim	8
1.2 Relevance	8
1.3 Scope	9
2 State of the art	11
2.1 Buckling problem	11
2.2 Buckling load optimization	12
3 Buckling eigenvalues problem	16
3.1 Buckling differential equation	16
3.2 Weak formulation	18
3.3 Finite Elements formulation	20
4 Optimization problem	22
4.1 Mathematical optimization	22
4.2 Buckling optimization	23
4.2.1 Shape functionals	24
4.3 Eigenvalues sensitivity	25
5 Code	29
5.1 The <i>Swan</i> framework	31
5.2 Object Oriented Programming	31

5.3 Code refactoring	32
6 Change of design variable	40
6.1 More than one design variables	43
6.2 Dimensional analysis	43
7 Sensitivity and results analysis	45
7.1 Circular section	45
7.2 Circular section with inner concentric hole	49
7.3 Rectangular section	54
8 Conclusions	60
A Cross-sections	63
A.1 Circular section	63
A.2 Square section	64
A.3 Rectangular section	65
A.4 Circumference with concentric hole section	66
A.5 Rectangular section with rectangular inner hole	67

List of Figures

2.1 Awkward suboptimal cross-sectional area distribution of clamped-clamped columns [6]	13
3.1 Deformed beam [10]	16
3.2 Forces acting on a beam's slice (dx) [10]	17
3.3 Hermite C^1 shape functions for a two-node beam [5]	21
5.1 Flowchart of the optimization algorithm [7]	30
5.2 A. Mañeru <i>IterativeProcessComputer</i> class UML diagram [7]	33
5.3 <i>SectionVariablesComputer</i> class structure	34
5.4 General UML graphic for the Euler Beam Optimizer code	35
5.5 Example of the 3D visualization of the column	36
5.6 Schematic representation of the matrices product	37
6.1 MMA, $\alpha = 0.5$, Random initial values, Cost = -51.07	41
6.2 MMA, $\alpha = 0.5$, Constant initial Value, Cost = -51	41
6.3 MMA, $\alpha = 0.25$, Random initial values, Cost = -52.24	41
6.4 fmincon, $\alpha = 0.5$, Random initial values, Cost = -51.07	42
6.5 Cantilever beam, MMA, $\alpha = 0.5$, Random initial values, Cost = -3.19	42
6.6 Hinged-hinged beam, MMA, $\alpha = 0.5$, Random initial values, Cost = -12.77	42
6.7 Schematic representation of the dimensional analysis	44
7.1 Circular section results	46
7.2 Circular with inner hole section results	50

7.3 N6 monitoring	52
7.4 Rectangular section results	55
7.4 Rectangular section results	56
A.1 Circular section	63
A.2 Square section	64
A.3 Rectangular section	65
A.4 Circular section with concentric inner hole	66
A.5 Hollow rectangle	67

List of Tables

2.1	Convergence of the eigenvalue simple solution with the mesh discretization in the clamped-clamped column with $\alpha = 0.5$ [7]	14
2.2	Numerical results of clamped-clamped column with fmincon and Augmented Lagrangian [7]	15
7.1	Reference cases initial configuration	47
7.2	Hindered cases initial configuration	47
7.3	Optimal case initial configuration	48
7.4	Other cases initial configuration	48
7.5	Reference cases initial configuration	51
7.6	Optimal cases initial configuration	51
7.7	Other cases initial configuration	52
7.8	Reference cases initial configuration	57
7.9	Hindered cases initial configuration	57
7.10	Optimal cases initial configuration	58
7.11	Other cases initial configuration	58

Chapter 1

Introduction

1.1 Aim

The aim of this thesis is to study the optimization problem of the buckling clamped-clamped column, which can be regarded as an eigenvalues problem, and propose new solutions using different types of cross-section geometries. To do so, a object-oriented code will be developed inside the framework *Swan* based on the previous work done by A. Mañeru [7]. The problem will be solved using different optimizers such as *fmincon* or *MMA*.

1.2 Relevance

Structural optimization is highly relevant in the field of engineering as it plays a crucial role in ensuring the safety, efficiency, and durability of structures. By using mathematical modeling and optimization techniques, structural engineers can design structures that are capable of withstanding external forces and loads while minimizing material usage and cost. This not only ensures the safety of the structure but also reduces the environmental impact by minimizing the use of materials. Structural optimization also allows engineers to explore new and innovative designs, which can lead to the creation of more efficient and sustainable structures.

In the aeronautical industry, structural optimization is particularly relevant in ecological terms as it can lead to significant reductions in fuel consumption and emissions. By using optimization techniques, engineers can design aircraft that are lighter and more aerodynamic, which in turn reduces the amount of fuel needed to operate the aircraft. This not only lowers operating costs but also reduces carbon emissions, making air travel more environmentally sustainable. Additionally, optimized structures can also improve the durability and lifespan

of aircraft, reducing the need for frequent replacements and further reducing environmental impact. Therefore, structural optimization is a key factor in achieving more sustainable and efficient air transportation, which is increasingly important as the demand for air travel continues to grow.

In particular, the appearance of new optimal designs is increasingly more relevant in the industry thanks to the growing capacity of additive manufacturing techniques, which allows to foresee a future where it will be possible to use new structures that were too complicated to fabricate at industrial level nowadays.

1.3 Scope

The development of this project will include:

- Preliminary study
 - Learning of Object Oriented Programming: Learning process of Object Oriented Programming in order to be able to work with the Swan Code
 - Theoretical review of the eigenvalue problem: Mathematical study of the eigenvalue problems (specifically, the buckling beam eigenvalue problem) and its formulation applied to the Finite Elements Method
 - Learning of Topology Optimization's main mathematical concepts and state of the art
 - Introduction to the Swan code: Main introduction to the Swan code, a topology optimization code developed in Matlab by Alex Ferrer, which will be the base code of this project
 - Detailed study of Aitziber Mañeru's Bachelor Final Thesis: *Study of numerical tools for the design of lightweight structures when considering the modal analysis*, since its developed code (Eigen Value branch of the Swan Code) and researches are going to be used as a starting point for this project. In fact, this project is based on one of the future lines of investigation proposed on Aitziber's work.
- Main study
 - Refactoring of the Eigen Values branch of the Swan Code: Refactoring of the Eigen Value branch in order to make it more readable, clear and clean.
 - Study of the interaction of the code with different optimizers: Performance analysis of different optimizers used for solving the problem. Development of code that adapts to working with different optimizers

- Improvement of the code in order to broaden its study cases: Development of new code to consider not only the case of solids of revolution but other kinds of cross sectional areas in the buckling beam problem.
- Analysis of the results obtained using different optimizers

Chapter 2

State of the art

In this chapter, a general overview of the previous works existing related to the optimization of the buckling column is presented.

Maximization of the buckling load is essential to enhance the overall structural stability by decreasing the possibility of reaching an unstable equilibrium position under any contemplated loading. Therefore, is essential to understand the buckling phenomena and how it has been studied along the years.

2.1 Buckling problem

The bucking phenomena arises as an elastic deformation before the stress level of a structure reaches the yield strength. It manifests itself as a sudden change in the shape of the structural components, with the appearance of important displacements transversal to the loading direction. The critical load at which buckling occurs is known as the buckling load or the Euler load. The buckling load depends on the geometry of the member, its material properties, and the boundary conditions of the system. Understanding the buckling behavior of structural members is crucial for the safe and efficient design of structures.

Buckling is included within the group of elastic instabilities. Those are a kind of instability occurring in elastic systems where the displacements are not linearly related to the forces applied to the system.

The buckling problem can be mathematically formulated as an eigenvalue problem, where the critical load is the eigenvalue and the buckling mode shape is the associated eigenvector. The solution of this problem involves finding the critical load and the corresponding mode shape, which can be obtained using analytical, numerical, or experimental methods.

The study of the buckling problem can be traced back to 1750 when Leonhard Euler and Daniel Bernoulli developed the bending theory of beams by analyzing their stresses and strains, which is a particular case of linear elasticity theories. [8] Their main postulation resulted in the equation that relates the bending moment (M) with the deflection of the beam, known as the Euler-Bernoulli equation.

$$M = -EI \frac{d^2 w}{dx^2} \quad (2.1)$$

where w is the transverse displacement of the beam, E the Young's modulus and I the second moment of area of the beam's cross section. Later on, Euler analysed the deformed beam from an energy point of view obtaining the following equation of the beam's total strain energy

$$E = \frac{1}{2} \int_0^L EI \left(\frac{d^2 w(x)}{dx^2} \right)^2 - \frac{1}{2} \lambda \int_0^L \left(\frac{dw(x)}{dx} \right)^2 \quad (2.2)$$

where λ is the applied axial load. The equilibrium position can be found at an extreme value of the functional

$$\frac{\partial E[w(x)]}{\partial w(x)} \quad (2.3)$$

which, for the case of a constant section and Young's modulus beam, leads to the Euler's critical buckling load:

$$P_{cr} = \frac{\pi^2 EI}{L^2} \quad (2.4)$$

Equation [2.2] can also be regarded as an eigenvalues problem where the critical axial load λ_c is the first eigenvalue.

2.2 Buckling load optimization

The problem of maximizing the critical buckling load of an elastic column was first introduced by Lagrange in 1773. The problem focused on a column with a variable cross-sectional area, and a given length and volume. The type of columns considered in this problem were solids of revolution, where $I = cA^2$. However, the solution initially obtained by Lagrange was incorrect, as he arrived to the cylinder shape as the optimal one.

Subsequently, the first optimal solution was found analytically by T.Clausen in 1851 for the case of a cantilever column. Contrary to Lagrange's approach, he minimized the volume

subject to a fixed buckling load, which is an equivalent problem. Later on, Tadjbakhsh and Keller (1962) derived analytical optimal solutions for clamped-clamped and clamped-hinged columns. Their findings led to a surprising conclusion for the clamped-clamped case: the area function $A(x)$ vanished at $1/4$ and $3/4$. In other words, zero cross sections appeared at some points along the beam, giving highly non-linear and non-feasible solutions.

All the aforementioned solutions are unimodal, meaning that they have a single buckling mode. However, Olhoff and Rasmussen (1977) discovered that the unimodal solution obtained by Tadjbakhsh and Keller (1962) for the clamped-clamped case was incorrect. Olhoff affirmed that Keller's analysis was mistaken when it comes to differentiability, since there's a shortage of continuity due to the appearance of multiple eigenvalues. They numerically determined the optimal solution to the problem and showed that the optimal solution is bimodal, meaning that it has two buckling modes at the same buckling load. Based on their work, finally, it was Steven J. Cox and Michael L. Overton in 1992 ([3]) who came to the right and definitive solution. [9]

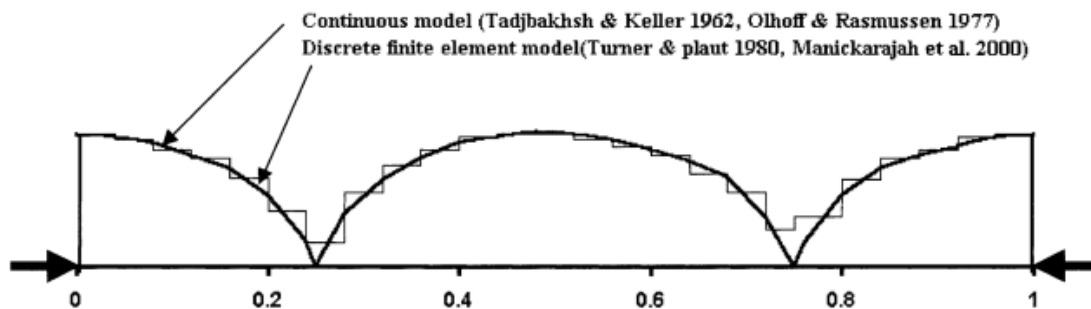


Figure 2.1: Awkward suboptimal cross-sectional area distribution of clamped-clamped columns [6]

As a point of start for this study, the work developed by Aitziber Mañeru in 2022 [7] is used as a reference. One part of her thesis was centered on implementing a code inside the *Swan* software that was able to solve via Finite Elements Method (FEM) the Cox and Overton's problem. As well as all the aforementioned works that have studied the optimization problem of a beam against buckling, Mañeru solved the problem defining the column as a solid of revolution. In fact, the design variable used was the area $A(x)$ while the circular shape was introduced through the second moment of area, which for the case of a circular area can be defined as $I = cA^2$. The optimization problem studied by Mañeru is formulated as follows:

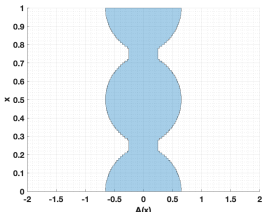
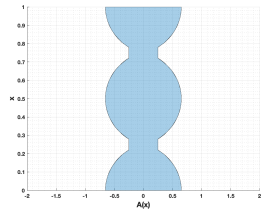
$$\min_A -\gamma(A) \quad s.t. \quad \begin{cases} \gamma - \lambda_{1,1}(A) \leq 0 \\ \gamma - \lambda_{1,2}(A) \leq 0 \\ V - V_{max} \leq 0 \\ \alpha \leq A \leq \beta \end{cases} \quad (2.5)$$

where α and β are the area lower and upper bounds, which as Cox and Everton analysed, are highly related to the appearance of single or multiple eigenvalues. $\lambda_{1,1}$ and $\lambda_{1,2}$ are the first and second eigenvalues respectively.

Although the optimization problem and her code are analysed in detail later on in this thesis (section 5), some of her most relevant results are summarized below.

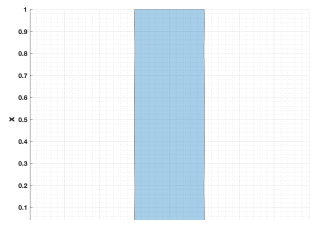
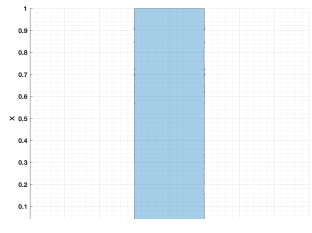
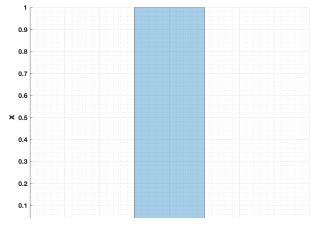
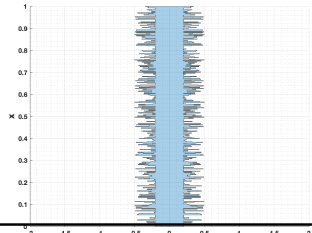
In concordance with the results obtained by Cox and Overton, in table 2.1 the main results obtained by A. Mañeru are shown.

Table 2.1: Convergence of the eigenvalue simple solution with the mesh discretization in the clamped-clamped column with $\alpha = 0.5$ 7

Inputs	Profile	Results
number of elements: 100 init. val. design variable: ones		iter cost: 51,048 vol: - 1,5e-5 constraint:
number of elements: 500 init. val. design variable: ones		cost: 51,0786 vol: - 10e-9

This results were obtained using the *MMA* optimizer, and the achieved cost was very similar to the one obtained by Cox and Everton (51.07), therefore verifying the code implemented. However, an issue aroused when trying to solve the problem using other optimizers such as *fmincon* or *Augmented Lagrangian*.

Table 2.2: Numerical results of clamped-clamped column with *fmincon* and Augmented Lagrangian [7](#)

Inputs	Profile	Results
Optimizer: <i>fmincon</i> number of elements: 400 init. val. design variable: ones		iter: 12 cost: 12 vol: - 1,5e-3
Optimizer: <i>fmincon</i> number of elements: 400 init. val. design variable: random		iter: 13 cost: 10 vol: - 1,6e-3
Optimizer: Augmented Lagrangian number of elements: 400 init. val. design variable: ones		iter: 10 cost: 10 vol: - 1,5e-3
Optimizer: Augmented Lagrangian number of elements: 400 init. val. design variable: random		iter: 20 cost: 10 vol: - 0,14

As can be observed in table [2.2](#), all the results obtained by *fmincon* and *Augmente Lagrangian* optimizers stayed at a cost value of 10, much below the one obtained using *MMA* and *Cox* and *Everton*, and the final geometry was simply the constant cross-section column.

Thus, in this project, the main goal is to obtain new solutions to the buckling optimization problem by using other types of cross sections such as the rectangular one or the circular with an inner concentric hole. The code developed by A. Mañeru is used as reference, which will be refactored and broadened to admit other design variables types than the area. Also, the code will be fixed to work with all the optimizers available in *Swan*.

Chapter 3

Buckling eigenvalues problem

In this section the differential equation that governs the buckling problem is obtained, then, its weak formulation is introduced in order to be able to formulate the problem via Finite Elements Method.

3.1 Buckling differential equation

The basic differential equation that governs the buckling problem can be derived applying the equilibrium equations to the beam's deformed geometry. In particular, a simply supported beam subjected to an axial compressive force (P) acting at both ends. Also, an auxiliary distributed lateral load (q) acting along the x -direction is introduced, that will later be set to 0. The following development is based on the Euler-Bernoulli's beams theory.

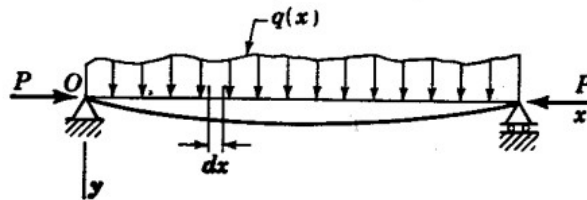


Figure 3.1: Deformed beam [10]

The equilibrium equations are applied to a differential beam element of length dx (Fig. 3.2). For simplicity reasons, the lateral load is considered constant along the beam's length.

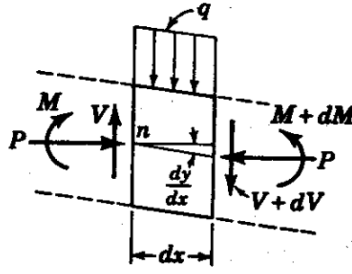


Figure 3.2: Forces acting on a beam's slice (dx) [10]

Applying equilibrium in the y direction gives:

$$-V + q dx + (V + dV) = 0 \longrightarrow q = -\frac{dV}{dx} \quad (3.1)$$

and, taking moments equilibrium at point n :

$$M + q dx \frac{dx}{2} + (V + dV)dx + P \frac{dy}{dx} dx - (M + dM) = 0 \quad (3.2)$$

neglecting second order derivatives gives:

$$V = \frac{dM}{dx} - P \frac{dy}{dx} \quad (3.3)$$

Following, the Euler-Bernoulli's equation that relates the bending moment (M) with the transverse displacement of the beam is introduced:

$$M(x) = EI \frac{d^2 y}{dx^2} \quad (3.4)$$

Combining equations [3.1], [3.3] and [3.4], the following expression can be obtained:

$$\frac{d^2}{dx^2} \left(EI \frac{d^2 y}{dx^2} \right) + P \frac{d^2 y}{dx^2} = q \quad (3.5)$$

Finally, the distributed lateral force q is set to 0 to obtain the differential equation that governs the buckling behaviour of an elastic beam caused by an axial compressive force P ([10]):

$$N \frac{d^2 w(x)}{dx^2} + \frac{d^2}{dx^2} \left(EI \frac{d^2 w(x)}{dx^2} \right) = 0 \quad (3.6)$$

Analytical results have been obtained for eq. 3.6 if the beam of study has constant section and Young's modulus along the beam axe. Which led to Euler's Critical Buckling load equation 2.4

However, if the objective is to obtain optimal geometries other than the constant section ones, the analytical solution is no longer useful. And another method has to be considered which can take into account partial differential equations (PDE) with variable sections.

3.2 Weak formulation

Once the governing equation of the Buckling problem is obtained (Eq. 3.6), it is then transformed to its weak form, in order to solve it using the Finite Element Method. Using the weak formulation allows to relax the derivative continuity requirements of the solution.

To begin with, eq. 3.6 is multiplied by a test function $v(x)$ and integrated over the domain Ω :

$$\int_{\Omega} v(x) \left[N \frac{d^2 w(x)}{dx^2} + \frac{d^2}{dx^2} \left(EI \frac{d^2 w(x)}{dx^2} \right) \right] dx = 0 \quad \forall v(x) \quad (3.7)$$

For practicality reasons, the equation is divided into two terms:

$$\int_{\Omega} v(x) N \frac{d^2 w(x)}{dx^2} dx + \int_{\Omega} v(x) \frac{d^2}{dx^2} \left(EI \frac{d^2 w(x)}{dx^2} \right) dx = 0 \quad (3.8)$$

The first term is integrated by parts once:

$$\int v N \frac{d^2 w}{dx^2} dx = - \int N \frac{dv}{dx} \frac{dw}{dx} dx \quad (3.9)$$

and the second term is integrated by parts twice:

$$\int v \frac{d^2}{dx^2} \left(EI \frac{d^2 w}{dx^2} \right) dx = \int \frac{d^2 v}{dx^2} EI \frac{d^2 w}{dx^2} dx \quad (3.10)$$

where the boundary terms have been eliminated due to the boundary conditions.

Finally, combining equation 3.9 and 3.10, the weak formulation of the Buckling problem is obtained (eq. 3.11). Notice that the fourth derivative of the transverse displacements ($w(x)$) no longer appears and only its second derivative is used. The solution functions obtained

will need a continuity requirement of type C^1 that ensures the integrability of the second derivative of the function. [5]

$$-N \int \frac{dv}{dx} \frac{dw}{dx} dx + \int \frac{d^2v}{dx^2} EI \frac{d^2w}{dx^2} dx = 0 \quad (3.11)$$

Following, the shape functions are introduced, which combined with the weak form will allow to obtain the discretized equations of the problem, in the form of finite elements equations. Considering that the exact solutions $w(x)$ and $v(x)$ belong to infinite dimensional spaces \mathcal{S} and \mathcal{V} , the approximation consists in changing w and v with functions belonging to finite dimensional spaces \mathcal{S}^h and \mathcal{V}^h , which are subsets of \mathcal{S} and \mathcal{V} . This new finite dimensional functions are called w^h and v^h . Leading us to the discrete form of the variational equation:

$$-N \int \frac{dv^h}{dx} \frac{dw^h}{dx} dx + \int \frac{d^2v^h}{dx^2} EI \frac{d^2w^h}{dx^2} dx = 0 \quad (3.12)$$

The approximated transverse displacement, and also the approximated test function, are described as the linear combination of shape functions which have to fulfill the smoothness required by the second derivative of the displacements filed. The shape functions used will be the same for both the trial and test functions, as in the Galerkin method, this is

$$w^h(x) = \sum_j^n w_j N_j(x) \quad (3.13)$$

$$v^h(x) = \sum_j^n v_j N_j(x) \quad (3.14)$$

Introducing eq. [3.13] and eq. [3.14] into eq. [3.12] the following equation is obtained:

$$\sum_j^n w_j [B - \lambda K] = 0 \quad (3.15)$$

where N has been renamed as λ , and B and K are respectively the Bending and Stiffness matrices [7]:

$$\begin{cases} K = \int \frac{dN_i}{dx} \frac{dN_j}{dx} dx & \text{Stiffness Matrix} \\ B = \int \frac{d^2N_i}{dx^2} EI \frac{d^2N_j}{dx^2} dx & \text{Bending Matrix} \end{cases} \quad (3.16)$$

3.3 Finite Elements formulation

Once the weak form of the governing equation of the Buckling problem is obtained, and has been discretized, the finite Elements Formulation is introduced from the element point of view.

The buckling problem that is being studied is a one dimensional problem, therefore, the beam must be discretized in elements along the x-direction.

As concluded previously, the approximate solutions, and the shape functions, for each beam element require C^1 continuity. The second derivatives of the transverse displacement appearing in the weak form of the problem require the existence of the rotation $\theta = \frac{dw}{dx}$ as a degree of freedom to maintain the continuity between elements. Therefore, each 2-nodes element will have four degrees of freedom: the transverse displacement at each node, and its rotations. 5 Thus, the nodal displacement vector can be described as

$$d^e = \begin{bmatrix} u_1^e \\ \theta_1^e \\ u_2^e \\ \theta_2^e \end{bmatrix} \quad (3.17)$$

where $u_1 = u(x_1)$, $\theta_1 = u'(x_1)$, $u_2 = u(x_2)$ and $\theta_2 = u'(x_2)$.

The interpolation functions selected for this case, which must be C^1 functions, are the Hermite polynomials. Below they are described for an element of length l^e using the parent domain ξ :

$$N_{u_1} = \frac{1}{4}(1 - \xi)^2(2 + \xi) \quad (3.18)$$

$$N_{\theta_1} = \frac{l^e}{8}(1 - \xi)^2(1 + \xi) \quad (3.19)$$

$$N_{u_2} = \frac{1}{4}(1 + \xi)^2(2 - \xi) \quad (3.20)$$

$$N_{\theta_2} = \frac{l^e}{8}(1 + \xi)^2(\xi - 1) \quad (3.21)$$

where

$$\xi = \frac{2x}{l^e} - 1 \quad \text{and} \quad -1 \leq \xi \leq 1$$

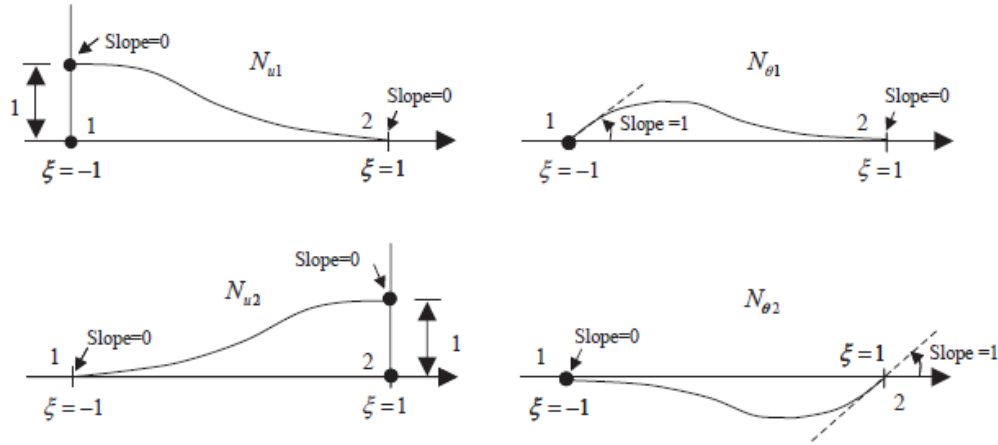


Figure 3.3: Hermite C^1 shape functions for a two-node beam [5]

As can be observed in figure 3.3, the Hermite polynomials have the following Kronecker delta properties:

$$N_{u_i}(x_j) = \delta_{ij} \quad \text{and} \quad \frac{dN_{\theta_i}}{dx}(x_j) = \delta_{ij} \quad (3.22)$$

concluding that they interpolate both the function and its derivatives at the nodes.

Therefore, the approximation function of the transverse displacement is

$$u^e(x) = N_{u_1}u_1 + N_{\theta_1}\theta_1 + N_{u_2}u_2 + N_{\theta_2}\theta_2 \quad (3.23)$$

Written in a matricial way, we have:

$$u^e = [N_{u_1}^e(\xi) \quad N_{\theta_1}^e(\xi) \quad N_{u_2}^e(\xi) \quad N_{\theta_2}^e(\xi)] \begin{bmatrix} u_1^e \\ \theta_1^e \\ u_2^e \\ \theta_2^e \end{bmatrix} \quad (3.24)$$

In order to calculate the shape functions derivative with respect to x that appear in equation 3.16, the following expression is used

$$\frac{d}{dx} = \frac{l}{2} \frac{d}{d\xi} \quad \text{and} \quad \frac{d^2}{dx^2} = \frac{l^2}{4} \frac{d^2}{d\xi^2} \quad (3.25)$$

Finally, the bending and stiffness matrices can be rewritten as:

$$(B_e)_{ij} = \int_{-1}^1 l \frac{EI}{2} \frac{\partial^2 N_i}{\partial \xi^2} \frac{\partial^2 N_j}{\partial \xi^2} d\xi \quad (K_e)_{ij} = \int_{-1}^1 l \frac{1}{2} \frac{\partial N_i}{\partial \xi} \frac{\partial N_j}{\partial \xi} d\xi \quad (3.26)$$

Chapter 4

Optimization problem

In this section the formulation of the optimization problem is presented. First, there's a general overview of mathematical optimization. Then the buckling optimization problem is detailed using the general and the bound formulation.

4.1 Mathematical optimization

An optimization problem is a mathematical problem that involves finding the best solution among all possible solutions, typically subject to some constraints. The goal is to minimize or maximize a particular objective function, which measures the quality of the solution.

The classic constrained optimization problem can be described using the following nomenclature:

$$\begin{cases} \text{minimize } f_0(x) \\ \text{subject to } f_i(x) \leq b_i, \quad i = 1, \dots, m \end{cases} \quad (4.1)$$

Here, x represents the optimization variable or design variable, f_0 denotes the objective function, f_i refers to the equality and/or inequality constraints, and b_i represents the bounds on the constraints. A feasible value or vector is considered to be a solution of the optimization problem if it yields the lowest possible objective value while satisfying all of the constraints.

The form of the objective and constraint functions determine the class of the optimization problem. An optimization problem where the objective and constraint functions are linear is called a *linear program*, while if they are not linear is called a *nonlinear program*. [\[11\]](#)

In this work the algorithms behind the optimizers used are not going to be detailed, since it is not in the scope of the study. However, is necessary to acknowledge the requirements

needed by the optimizers to work. In the iterative process that an optimizer carries out, to calculate each next step, the algorithm requires both the value and the gradients of the cost and constraint functions. The two main optimizers used in this thesis are the *fmincon* and the *MMA*.

4.2 Buckling optimization

The optimal buckling design of a slender column may be defined as finding the maximum value of the critical buckling load for a given volume. A common alternative approach to the problem definition it may be to minimize the structural weight that satisfies a prescribed buckling load. However, this study is centered on the first definition.

General formulation

Following the mathematical nomenclature of an optimization problem, the buckling optimization problem can be defined as

$$\begin{aligned}
 & \underset{\rho}{max} \quad \lambda_1(\rho) \\
 & \text{s.t.} \quad \begin{cases} (B(\rho) - \lambda K)\phi = 0 \\ \alpha \leq \rho \leq \beta \\ \int A(\rho) dx \leq V_{max} \end{cases} \quad (4.2)
 \end{aligned}$$

which is read as: maximize the first eigenvalue, which is the objective function and depends on the design variable ρ , subject to the detailed constraints. The first constraint, which is an equality constraint, denotes the eigenvalues problem, i.e. that the eigenvalue λ_1 must be a solution of this equation. The second (inequality) constraint defines the lower (α) and upper (β) bounds of the design variable ρ , which have an important relevance in the appearance of multiple eigenvalues, as detailed J. Cox and L. Overton in [3]. The last (inequality) constraint defines the volume restriction, in a way that the volume can be less or equal than the maximum imposed volume, but never greater. This constraint could also be defined as an equality constraint, but for the case of study the first option is used give the optimizer more freedom to achieve different results.

For convenience with the optimizers used, the maximization problem (eq. 4.3) is transformed into an equivalent minimization problem by taking the negative value of the objective function [7]:

$$\begin{aligned}
 \min_{\rho} \quad & -\lambda_1(\rho) \\
 \text{s.t.} \quad & \begin{cases} (B(\rho) - \lambda K)\phi = 0 \\ \alpha \leq \rho \leq \beta \\ \int A(\rho) dx \leq V_{max} \end{cases}
 \end{aligned} \tag{4.3}$$

Bound formulation

In case that eigenvalues have a multiplicity of two, there will exist two associated eigenvectors ϕ_1 and ϕ_2 that correspond to the double eigenvalue λ_1 . Although the objective function is not differentiable, it is still possible to calculate directional derivatives in two directions. To achieve this, a method called the "Bound Formulation" must be utilized, which involves adding two new constraints to the problem. [4] The first step is to rename the cost function introducing a new scalar variable γ which will have to fulfill

$$\begin{aligned}
 \lambda_{1,1}(\rho) &\geq \gamma \\
 \lambda_{1,2}(\rho) &\geq \gamma
 \end{aligned} \tag{4.4}$$

where $\lambda_{1,1}$ and $\lambda_{1,2}$ are the double eigenvalues. Finally, the problem that is going to be solved in this work can be formulated using the bound formulation as

$$\begin{aligned}
 \min_{\rho} \quad & -\gamma(\rho) \\
 \text{s.t.} \quad & \begin{cases} \gamma - \lambda_{1,1}(\rho) \leq 0 \\ \gamma - \lambda_{1,2}(\rho) \leq 0 \\ \alpha \leq \rho \leq \beta \\ \int A(\rho) dx \leq V_{max} \end{cases}
 \end{aligned} \tag{4.5}$$

where

$$[B(\rho) - \lambda K] \phi = 0 \tag{4.6}$$

and

$$B(\rho) = I(\rho)B_0 \tag{4.7}$$

4.2.1 Shape functionals

Both, the objective function and the constraints can be considered as Shape Functionals. The values of this shape functionals, together with its gradients, are passed to the optimizer in order to find the new design variable distribution on each iteration.

Objective function

In this case the objective function, also named as Cost, takes the form of the scalar variable γ . Consequently, the shape function and its gradient are described as:

$$\begin{aligned} f_0 &= -\gamma \\ f'_0 &= 0 \end{aligned} \quad (4.8)$$

Constraints 1 & 2: Eigenvalues

Due to the bound formulation, the buckling problem is solved at the constraints introducing the two first eigenvalues in the following way:

$$\begin{aligned} f_1 &= \gamma - \lambda_1 \\ f_2 &= \gamma - \lambda_2 \end{aligned} \quad (4.9)$$

Owing to its complexity, the derivative of the eigenvalues is detailed next, in section [4.3](#)

4.3 Eigenvalues sensitivity

In order to calculate the gradient of the shape functionals that involve the eigenvalues solution of the buckling problem (axial loads), it is necessary to know its sensibility with respect to the design variable vector. Thereby, in this section a general formulation for the sensitivity analysis of finite element discretized structural eigenvalues problem is presented based on the work developed by A.P. Seyranian, E. Lund and N. Olhoff in [\[1\]](#). The analysis is divided in two parts considering the cases of single eigenvalues and multiple eigenvalues, which entail a more complex problem due to the non-differentiability.

Let's first consider the general eigenvalues problem definition

$$B\phi_j = \lambda_j K\phi_j \quad j = 1, \dots, n \quad (4.10)$$

where λ_j are the eigenvalues and ϕ_j the corresponding eigenvectors. The dimensions of the problem, n , indicate the number of possible eigenvalues, which will be sorted and numbered from the smallest to the highest in the following manner

$$0 < \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_j \leq \dots \leq \lambda_n \quad (4.11)$$

For further convenience, the eigenvectors are K-orthonormalized

$$\phi_j^T K \phi_k = \delta_{jk}, \quad j, k = 1, \dots, n \quad (4.12)$$

where δ_{jk} is the Kronecker's delta. The eigenvectors are also B-orthogonal, as is demonstrated premultiplying eq. [4.10](#) by ϕ_j^T

$$\phi_j^T B \phi_k = \lambda_j \delta_{jk}, \quad j, k = 1, \dots, n \quad (4.13)$$

Simple eigenvalues

For the case of simple eigenvalues the approach to obtain the sensibility is to differentiate eq. [4.10](#) by a design variable, x_i .

$$\frac{\partial B}{\partial x_i} \phi_j + (B - \lambda_j K) \frac{\partial \phi_j}{\partial x_i} = \frac{\partial \lambda_j}{\partial x_i} K \phi_j + \lambda_j \frac{\partial M}{\partial x_i} \quad i = 1, \dots, I \quad (4.14)$$

where the index I is the number of design variables.

Premultiplying by ϕ_j^T and applying the normalization in [4.12](#) the following expression is obtained for the eigenvalue sensitivity in the case of simple eigenvalues

$$\frac{\partial \lambda_j}{\partial x_i} = \phi_j^T \left(\frac{\partial B}{\partial x_i} - \lambda_j \frac{\partial K}{\partial x_i} \right) \phi_j \quad i = 1, \dots, I \quad (4.15)$$

Taking into account that the stiffness matrix (eq. [3.16](#)) is independent of the design variable, its term is eliminated.

$$\frac{\partial \lambda_j}{\partial x_i} = \phi_j^T \frac{\partial B}{\partial x_i} \phi_j \quad i = 1, \dots, I \quad (4.16)$$

Finally, must be noticed that the only term inside the bending matrix that is dependent on the design variable is the second moment of area (I). Thus, the following expression is obtained for the derivative of the bending matrix:

$$\frac{\partial B}{\partial x_i} = \frac{\partial I}{\partial x_i} B_e \quad e = 1, \dots, nElem \quad (4.17)$$

Multiple eigenvalues

The appearance of N multiple eigenvalues as a solution of the eigenvalue problem in Eq. [4.10](#) brings an added difficulty to the calculation of the eigenvalues sensitivity. This is due to the fact that each multiple eigenvalue λ_j will have a linearly independent eigenvector associated. Therefore, any linear combination of the eigenvectors will satisfy the eigenvalue problem.

$$\tilde{\lambda} = \lambda_j, \quad j = 1, \dots, n, \quad (4.18)$$

Assuming K-normalized eigenvectors

$$\phi_j^T K \phi_k = \delta_{jk}, \quad j, k = 1, \dots, n \quad (4.19)$$

a linear combination of eigenvectors is introduced in order to ensure its continuity

$$\tilde{\phi}_j = \sum_{k=1}^n \beta_{jk} \phi_k, \quad j = 1, \dots, n, \quad (4.20)$$

where β_{ij} are coefficients to be determined. The multiple eigenvalues sensitivities can be calculated introducing and solving a subeigenvalue problem. To do so, a small increment of an arbitrary design variable Δx_i is introduced, which will produce the following increment of the stiffness and bending matrices:

$$K + \frac{\partial K}{\partial x_i} \Delta x_i \quad \text{and} \quad B + \frac{\partial B}{\partial x_i} \Delta x_i, \quad i = 1, \dots, I \quad (4.21)$$

Thus, the eigenvalues and their respective eigenvectors for the perturbed design can be expressed as

$$\lambda_j(x_i + \varepsilon \Delta x_i) = \tilde{\lambda} + \varepsilon \mu_j(x_i, \Delta x_i) + o(\varepsilon), \quad j = 1, \dots, n \quad (4.22)$$

$$\phi_j(x_i + \varepsilon \Delta x_i) = \tilde{\phi}_j + \varepsilon \nu_j(x_i, \Delta x_i) + o(\varepsilon), \quad j = 1, \dots, n \quad (4.23)$$

where the coefficients μ_j and ν_j are the eigenvalue and eigenvector sensitivities respectively, still unknown. Then, substituting eq. [4.21](#), [4.22](#) and [4.23](#) into eq. [4.10](#) the following expression is obtained:

$$\left(\frac{\partial B}{\partial x_i} - \tilde{\lambda} \frac{\partial K}{\partial x_i} \right) \tilde{\phi}_j + (B - \tilde{\lambda} K) \nu_j = \mu_j K \tilde{\phi}_j \quad (4.24)$$

Next, premultiplying by ϕ_s^T gives

$$\phi_s^T = \left(\frac{\partial B}{\partial x_i} - \tilde{\lambda} \frac{\partial K}{\partial x_i} \right) \tilde{\phi}_j = \mu_j \phi_s^T M \tilde{\phi}_j, \quad s = 1, \dots, n \quad (4.25)$$

Taking into account that the terms $\tilde{\phi}_j$ were the linear combination of the eigenvectors ϕ_k (eq. [4.20](#)), and using the K-orthonormalization described in eq. [4.19](#), the following system of equations is obtained

$$\sum_{k=1}^N \beta_{jk} \left[\phi_s^T \left(\frac{\partial B}{\partial x_i} - \tilde{\lambda} \frac{\partial K}{\partial x_i} \right) \phi_k - \mu_j \delta_{sk} \right] \quad s = 1, \dots, n \quad (4.26)$$

Finally, and denoting that the stiffness matrix is independent of the design variable, the condition that imposes the existence of the nontrivial solution is

$$\det \left[\phi_s^T \frac{\partial B}{\partial x_i} \phi_k - \mu_j \delta_{sk} \right] = 0 \quad s, k = 1, \dots, n \quad i = 1, \dots, I \quad (4.27)$$

which is the equation that allows to obtain the sensitivities μ_j of multiple eigenvalues $\tilde{\lambda}$ for small changes of the design variable. Notice how eq. [4.27](#) acquires the form of a typical eigenvalue problem, where the sensitivities μ_j are the eigenvalues to be found.

Chapter 5

Code

To fulfill the goal of this study a software is required that is able to, on the one hand, solve through the Finite Element Method the differential equations that govern the Buckling problem, and on the other hand, optimize the beam, given an objective function and a set of constraints.

To do so, a code is developed encompassed inside the *Swan* framework, and based on the previous work and code developed by Aitziber Mañeru [7]. While Aitziber's code was able to optimize beams with circular cross-section, being the area the single design variable, now, the code is improved and made more versatile to allow it to handle a variety of different cross-section's geometries with more than one design variable, such as a rectangular or a hollow circular cross-sections.

In addition to functionality, the code adheres to a clean code philosophy, emphasizing readability and simplicity by avoiding redundancies and complexity. Moreover, the code follows an Object-Oriented Programming paradigm, providing a modular structure that allows reusability and flexibility.

The development of the work is divided in two main fields. The first is a refactoring phase that involves restructuring the code without changing its output and the second is a process of implementing new functionalities. The refactoring phase seeks to optimize and "clean" some functions and processes of the code to make it faster and cleaner. To ensure that the results obtained are not different from those obtained in the previous code, it is essential to have a system to verify them. For this reason, a test methodology is implemented that compares the old results with the new ones.

In figure 5.1 the flow chart of the optimization code main steps is shown. Two main parts can be identified. First, a initialization process takes part, where all the required data and classes are created and then passed to the next part, which is the optimization iterative process,

where, through an optimizer, the new proposed design variables distribution are calculated at each step until convergence is reached.

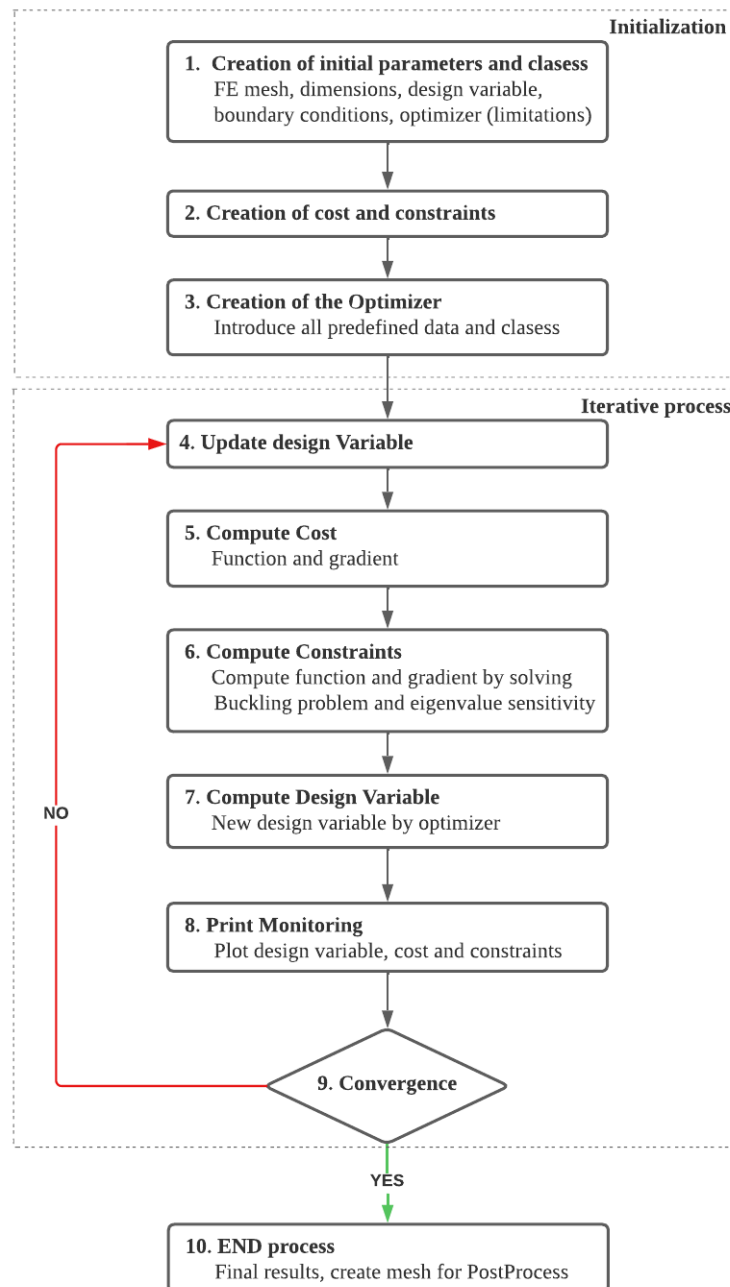


Figure 5.1: Flowchart of the optimization algorithm [7]

This section provides a brief introduction to the *Swan* framework and the main concepts of Object-Oriented Programming, followed by an analysis of the code's primary structure. Finally, the major issues addressed during the refactoring process are detailed, as well as the new functionalities that have been implemented.

5.1 The *Swan* framework

Swan is a versatile software designed for topology optimization. Initially created in 2017 as part of several bachelor theses, *Swan* has since expanded its capabilities to solve heat transfer problems, Navier-Stokes-based problems, and non-linear deformation elasticity problems. The software is written in MATLAB and uses an object-oriented approach for its main architecture. Over time, the architecture has been enhanced to address new problems and features. *Swan* follows a test-driven development approach by setting benchmark cases with analytical solutions to validate results. *Swan* is designed as an open source software, and therefore has been benefited by the work of several students in their bachelor and masters thesis. [2](#)

5.2 Object Oriented Programming

The code used to solve the buckling optimization problem, as well as the general *Swan*, is developed using object-oriented programming.

Object-oriented programming (OOP) is a programming paradigm, supported by main coding languages such as Python, C++ or MATLAB, that organises the codes around a modular structure. OOP enables to combine data (properties) and functions that interact with this data (methods) into modules called *objects*.

Objects are in fact instances of *classes*. Those are the templates that define the methods and properties included in a particular object. For instance, in the *Swan* code, an important class is the *Mesh*. This class contains all the methods related to the generation of different types of mesh, and within its properties there are the connectives and coordinates matrices. Once you want to create a particular mesh inside the code you create an object, which will be an instance of the class *Mesh*, and this object will have its particular values of connectivity and coordinates. Classes could be regarded as the passive actors in the code whereas objects are the active ones.

Object oriented softwares allows coping with high-complexity and can avoid code duplication by creating reusable objects with well-defined interfaces that hide the complexity of the underlying code. OOP it is well suited for programs that are large, complex, and actively

updated or maintained, as is the case of *Swan*, since it simplifies software development and maintenance.

One of the main strengths of Object Oriented Programming is the way objects are organised and the relationship that is established between them. These relations can be classified as *Composition* or *Inheritance* among others.

By Composition, an object contains other objects within itself. Whereas, in Inheritance, an object is a "child" of other object, in a way that it "inherits" (hence the name) all the properties and methods of the "parent" object, added to the ones that he has. The concept Composition over inheritance refers to the importance of prioritising the first over the second for versatility and flexibility. By using composition, it is possible to achieve code reuse by containing instances of other classes that implement the desired functionality rather than using the ones contained in a "parent" object.

Within large codes, object-oriented architectures can quickly become complex due to the many relationships between objects. Following clean code practices is thus important to keep the code manageable and easy to understand. By prioritizing readability and simplicity, developers can reduce the time and effort required to maintain the code, and minimize the risk of introducing bugs or errors. It is for this reason that object-oriented programming and clean code are concepts that must always come together. [\[2\]](#)

5.3 Code refactoring

General structure

When Aitziber's code was inherited, it was divided into two main different parts. The first one was the class *EulerBeamOptimizer*, where all the initialization parameters were defined, the mesh, the initial design variable vector and all the main classes were created and then passed to the class *IterativeProcessComputer*. However, the cost and constraints, as well as the *EigModes* class, were created inside the *IterativeProcessComputer*, as well as the initialization of the optimizer. This class constituted the second general part of the code, where the optimization iterative process took part.

One of the major drawbacks of this organisation was the nested structure that implied the transmission of a lot of information and objects from the "higher" class *EulerBeamOptimizer* to the "lower" class *IterativeProcessComputer*. Also, although the idea of the organisation was to have all the initialization part grouped in one class and the optimization part in another, in the end, what happened was that the objects were created in different classes without a strict criteria. And rigorously, the optimization process takes part inside the optimizer class, thus the *IterativeProcessComputer* class had no major functionalities implemented, becoming

redundant.

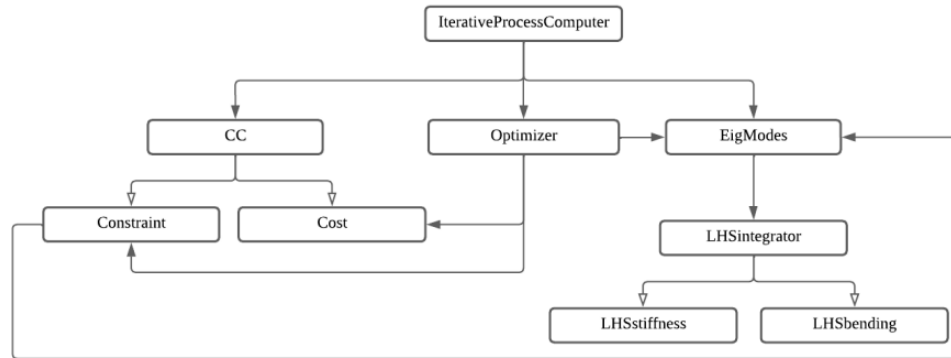


Figure 5.2: A. Mañeru *IterativeProcessComputer* class UML diagram 7

Having analyzed the main structure of the code, a new architecture is proposed that is more horizontal and readable. To begin with, the class *IterativeProcessComputer* is eliminated. Instead, all the classes that were created inside it are now created at the superior level *EulerBeamOptimizer*. The idea is to have the creation of all the relevant classes (i.e. the mesh, the design variable, the cost and constraints and the eigen modes) at the *EulerBeamOptimizer*, and then, passing the required ones to a new class called *OptimizerCreator* which deals with initializing and running the optimizer. All the not so relevant classes are now grouped inside one of the aforementioned classes. For instance, while before the creation of the bending and stiffness matrices took part at the same level as the creation of the *EigModes*, now, the *EigModes* includes them inside, being a more compact and consistent class that deals with everything related to solving the eigenvalues buckling problem.

Also, a new class is created, called *SectionVariablesComputer* that seeks to deal with all the data related with the cross section of the beam, and that is then required by the shape functionals and its gradients. The idea is that the class receives as an input data the design variable vector (x) and it returns as an output the area, its derivative, and the second moment of area with its respective derivative. This way, the *DesignVariable* class is exempt of performing all this calculation and just deals with the initialization of the design variables and storing its value over the iterations. On the other hand, the new class *SectionvariablesComputer* has all those calculations grouped, increasing this way the reusability between similar cross-sections geometries that can share some functions. Inside the class, another class is created according to the type of section that is being calculated. For the moment, this can only be a *QuadrilateralSection* or a *CircularSection*. However, this modular structure easily allows the implementation of other new section types.

Finally, another important refactoring implemented to the code has consisted of bringing all

the configuration parameters to the surface. This means, for example, to be able to change the maximum volume restriction or the type of optimizer used at the *EulerBeamOptimizer* level, and not inside a subclass. This feature adds the capability to test different simulations within a shorter time and increases the flexibility at the time of trying new configurations

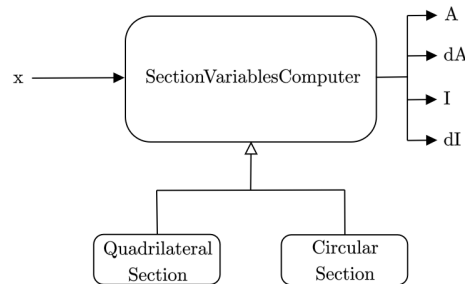


Figure 5.3: *SectionVariablesComputer* class structure

The problem found by A. Mañeru when using other optimizers than the MMA, that gave constant solutions, which has been detailed in section 2.2 has also been solved. The issue was related with the way the design variable restrictions were defined and passed to the optimizers, which was incompatible with the majority of optimizers. The code works now for all the optimizers implemented inside *Swan*.

In figure 5.4 the general architecture of the code is shown once the refactoring has been carried out. It is depicted using a UML (Unified Modelling Language) graphic, which is a standardized language used to visualize and document code. Here the black arrows denote *Composition*, in the sense that a class is using instances of other classes within itself, and the white arrows denote *Inheritance*, in the sense that the "child" classes inherit the methods and properties of the "parent" class.

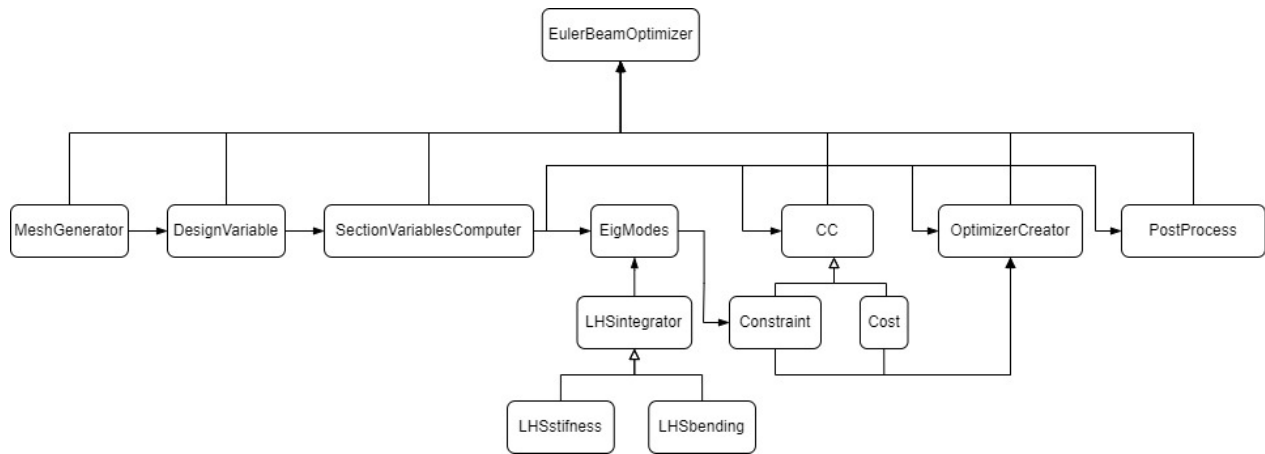


Figure 5.4: General UML graphic for the Euler Beam Optimizer code

Finally, the processes that are taking place at *EulerBeamOptimizer* class after the refactoring are listed below:

1. Initialize parameters
2. Mesh creation
3. Design variable vector creation
4. Section variables class creation
5. EigModes, Cost and Constraints creation
6. Optimizer creation
7. Solving the iterative problem through the optimizer
8. Post process

3D plots

A new functionality that has been implemented to the code is the capability of visualizing the resulting beams in 3-dimensional graphics. This feature allows a more realistic view of the obtained geometries, and therefore can lead to more interesting analysis of the results. Is of particular interest in the more than one design variable cross section designs, since the 2D view couldn't achieve to show changes occurring out of the plane.

This feature is implemented inside a class called *Plot3DBucklingColumn*. Basically, the class creates a group of level set functions that determine which areas of a background mesh contain material and which does not.

The level set curves of the function f are the set of points (curves) that given to the function, this returns a constant value c .

$$L_c(f) = \{(x_1, \dots, x_n) | f(x_1, \dots, x_n) = c\} \quad (5.1)$$

Is called a zero level-set when the constant value is zero.

Based on the zero-level set of a circle of radius r and center (x_0, y_0) , a function (eq. 5.2) is obtained that returns positives values if the point (x, y) is outside the circle, negative values if the point is inside the circle and zero if the point is on the boundary of the r -radius circle. Each type of cross section uses a different level set function.

$$L = (x - x_0)^2 + (y - y_0)^2 - r^2 \quad (5.2)$$

This is the basis of the system used to generate 3D representations of the obtained beams, as the one shown in figure 5.5.

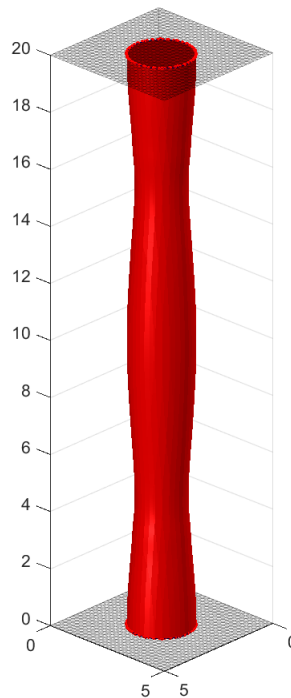


Figure 5.5: Example of the 3D visualization of the column

For loops elimination

Another important change to the code is the substitution of all the *For* loops that iterated through the total number of elements, usually with the following syntax:

Algorithm 1 For loop structure

```

for  $iElem = 1$  to  $nElem$  do
     $Syntax(iElem)$ 
end for
    
```

The idea behind this change is to optimize the code by using equally valid alternatives that aren't as computationally expensive as the loops through the elements. The loops that iterate through all the elements are a potential problem that needs to be solved, although for a small number of elements doesn't pose a significant problem, for high number of element's cases those loops would transform in a very time consumption computation.

The most relevant case where this change has been implemented is when calculating the gradients of the eigenvalues, inside the *EigModes* class. Both in the case of simple or multiple eigenvalues, a loop through the elements was used to calculate eq. 4.16 and eq. 4.17. Which can be rewritten as

$$\frac{\partial \lambda_j}{\partial x_i} = \frac{\partial I}{\partial x_i} (\phi_j^T B_e \phi_j) \quad \text{for} \quad \begin{array}{l} i = 1, \dots, nDesignVar \\ e = 1, \dots, nElem \end{array} \quad (5.3)$$

The product of the eigenvectors by the bending matrix requires iterating through the elements to obtain the corresponding elemental bending matrix each time. To do so, a sub-vector of eigenvectors values corresponding to the global degrees of freedom of the respective element are used. And to obtain this sub-vectors a *index* operator is used. This process is described in algorithm 2

$$\begin{array}{c}
 \left[\begin{array}{c} \phi_{index}^T \end{array} \right] \cdot \left[\begin{array}{c} B_{iElem} \end{array} \right] \cdot \left[\begin{array}{c} \phi_{index} \end{array} \right] \\
 \begin{array}{ccc} 1 \times nDofElem & nDofElem \times nDofElem & nDofElem \times 1 \end{array}
 \end{array}$$

Figure 5.6: Schematic representation of the matrices product

Algorithm 2 Product by using a for loop through the elements

```

for  $iElem = 1$  to  $nElem$  do                                ▷ Loop through all the elements
   $index = ndofn \cdot (iElem - 1) + 1 : ndofn \cdot (iElem - 1) + ndofe$ 
   $dx = dI_{iElem}$ 
   $dfd_{x_{iElem}} = -dx \cdot (\phi_{index}^T \cdot B_{iElem} \cdot \phi_{index})$ 
end for
  
```

Alternatively, the following algorithm is proposed that avoids iterating through all the elements. Using the decomposition of the matrices product depicted in eq. [5.5](#), the product can be achieved iterating by rows and columns.

$$[\phi_1 \quad \phi_i \quad \dots \quad \phi_n] \cdot \begin{bmatrix} B_{11}^e & B_{1j}^e & \dots & B_{1n}^e \\ B_{i1}^e & B_{ij}^e & & B_{in}^e \\ \vdots & & \ddots & \vdots \\ B_{n1}^e & B_{nj}^e & \dots & B_{nn}^e \end{bmatrix} \cdot \begin{bmatrix} \phi_1 \\ \phi_j \\ \vdots \\ \phi_n \end{bmatrix} \quad i, j = 1, \dots, n \quad (5.4)$$

$$\sum_{j=1}^n \phi_j \left(\sum_{i=1}^n \phi_i B_{ij}^e \right) = \sum_{j=1}^n \sum_{i=1}^n \phi_i B_{ij}^e \phi_j \quad (5.5)$$

where n is the number of degrees of freedom by element, that in this case is $n = 4$.

However, here we still need to iterate the elements since this operation has to be repeated for each elemental bending matrix (B^e). The new idea introduced is to think in the bending matrix as a three-dimensional matrix, where each elemental matrix is stacked one behind the other so that the third dimension is the elements dimension. In this case, this bending matrix would become a $4 \times 4 \times nElem$ matrix. Then each item in the two-dimensional plane, i.e. the elemental degrees of freedom plane, can be transformed into a $nElem$ length vector. For instance, one vector would contain all the values in the position 11 of each elemental matrix. From here, the product of scalars in eq. [5.5](#) is transformed into a vectors product (element by element).

Algorithm 3 Product by avoiding iterating through the elements

```
Compute eigenvector by DOF
for  $iDof = 1$  to  $nDofE$  do
  for  $jDof = 1$  to  $nDofE$  do
     $B_{ij}(:, 1) = B_{elem}(iDof, jDof, :)$ 
     $w = \phi(:, iDof) \cdot B_{ij} \cdot \phi(:, jDof)$ 
     $W_{ab} = W_{ab} + w$ 
  end for
end for
```

Chapter 6

Change of design variable

In order to develop and broaden the cases of study, having in mind not only the solid of revolution case, but also other different types of cross sections, first, in this section a change of design variable is implemented. The problem is solved now with the column radius as the new design variable instead of the column area. While the problem remains the same: to maximize the first buckling eigenvalue of a clamped-clamped solid of revolution, now is reformulated using the cross section radius as design variable. The same results as in the area case should be obtained to validate the change. This change aims to establish the basis and procedures of future more complicated changes. The problem is now formulated as:

$$\min_r -\gamma(r) \quad s.t. \quad \begin{cases} \gamma - \lambda_{1,1}(r) \leq 0 \\ \gamma - \lambda_{1,2}(r) \leq 0 \\ V - V_{max} \leq 0 \\ \alpha \leq A(r) \leq \beta \end{cases} \quad (6.1)$$

Following, the results obtained by using as design variable the radius instead of the area are shown. Each figure displays the column profile, and the Cost and Volume Constraints evolution through the iterations. The cases are calculated with a 500 elements mesh for a unit length an volume column. *MMA* and *fmincon* have been used as optimizers. As can be observed, the costs obtained are equal to the ones obtained in A. Mañeru's work, which were the same obtained by Cox and Overton. Therefore, the change of design variable has been validated.

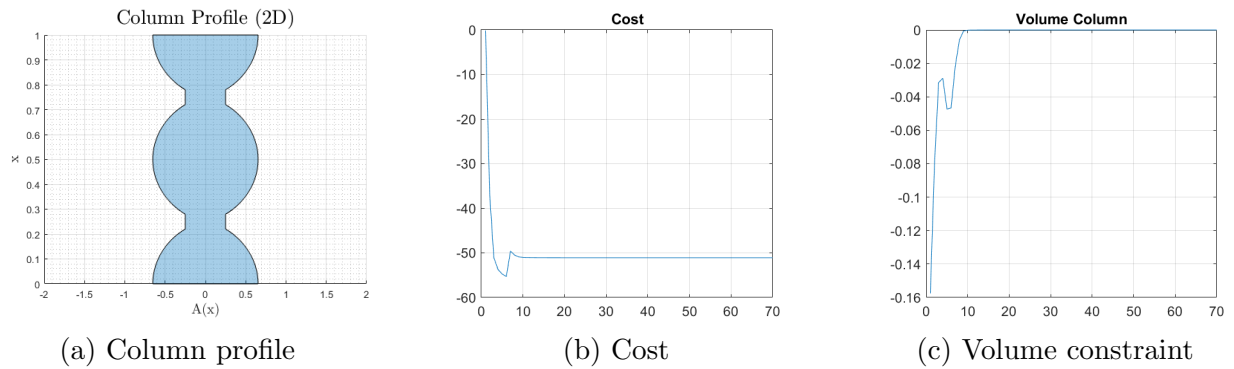


Figure 6.1: MMA, $\alpha = 0.5$, Random initial values, Cost = -51.07

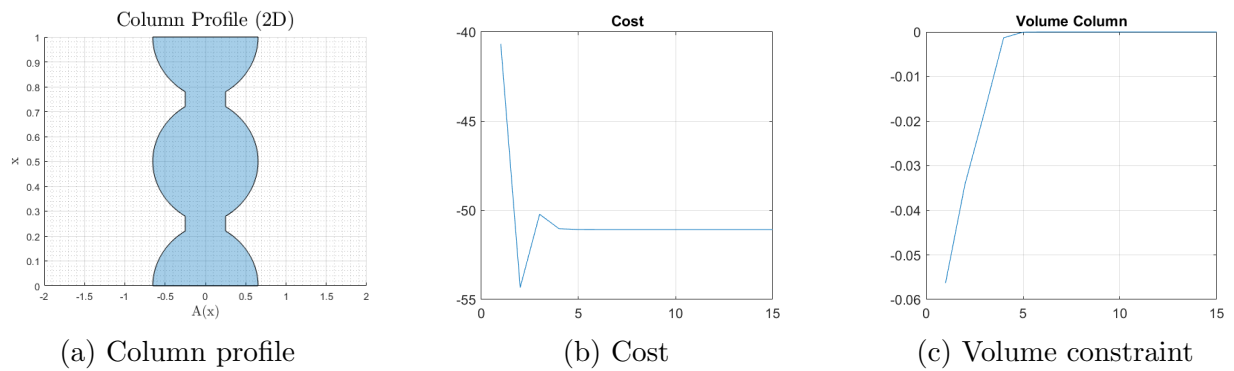


Figure 6.2: MMA, $\alpha = 0.5$, Constant initial Value, Cost = -51

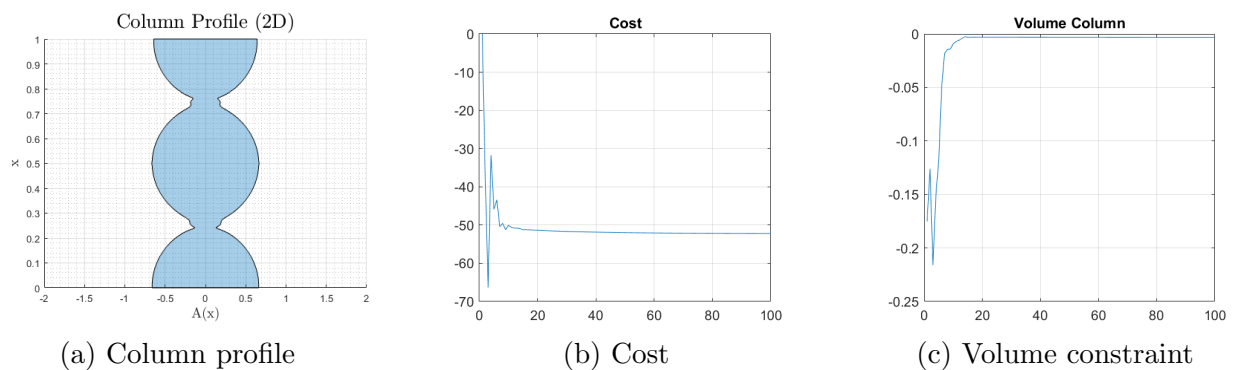


Figure 6.3: MMA, $\alpha = 0.25$, Random initial values, Cost = -52.24

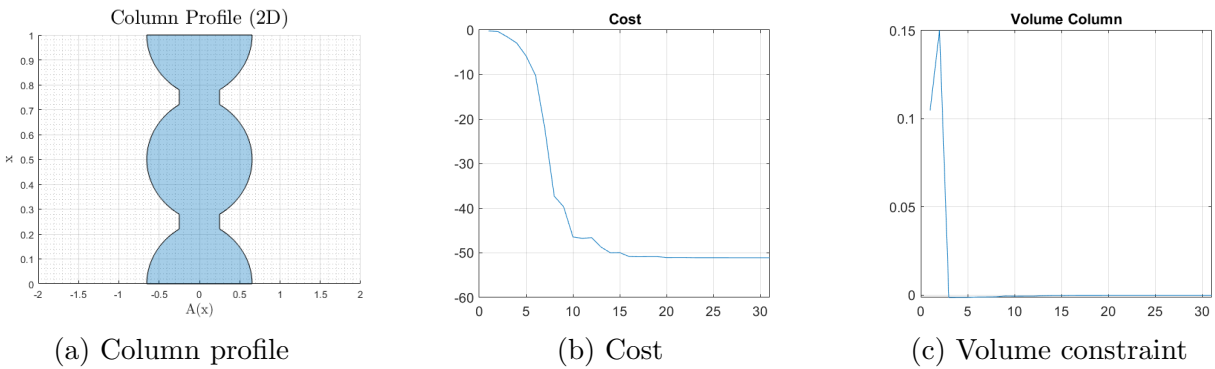


Figure 6.4: fmincon, $\alpha = 0.5$, Random initial values, Cost = -51.07

Also, the problem is simulated for the case of a cantilever beam and a hinged-hinged beam.

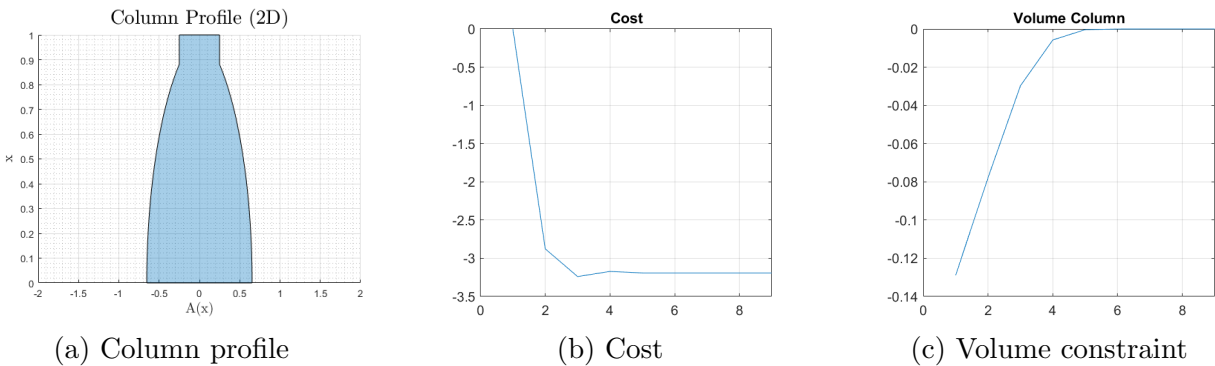


Figure 6.5: Cantilever beam, MMA, $\alpha = 0.5$, Random initial values, Cost = -3.19

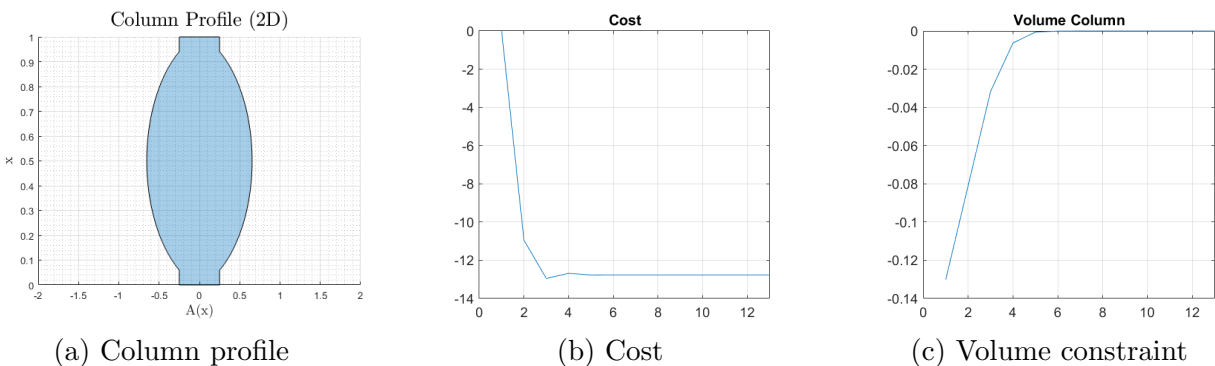


Figure 6.6: Hinged-hinged beam, MMA, $\alpha = 0.5$, Random initial values, Cost = -12.77

6.1 More than one design variables

In order to implement different more complex cross section geometries other than the circular one, the next step is to adapt the code to handle more than one design variables cases. First, it is necessary to adapt the design variables vector. This vector contains the values of the design variable at each element, therefore for more than one design variables, they are implemented stacking on variable after the other as is shown in equation [6.2](#)

$$x = [a_1 \dots a_n, b_1 \dots b_n] \quad n = 1, \dots, nElem \quad (6.2)$$

Inside the FEM equations developed in section [3.3](#) the cross-section type is defined using the second moment of area (I) inside the bending matrix. Thus, by changing the definition of I inside the code, the section is easily implemented with a code that receives as input the design variables vector (x), obtains the variables values at each element and calculates I, which is always a vector with the same length as the number of elements.

For calculating the derivative of eigenvalues, equation [4.17](#) is used, which already considers multiple design variables x_i . Contrary to the I vector, the derivative of the eigenvalues vector will have the same length as the number of design variables times the number of elements, e.g. for a 2 design variables section with 500 elements the vector will have 1000 elements.

When implementing sections with more than one design variable it is important to choose the correct parameters that will characterize it, since a wise selection can avoid complexity to the code. For example, to define a rectangular section with a rectangular inner hole, four design variables are required. A possible selection of design variables could be the external and internal width and height. However, this choice would entail the addition of additional constraints in order to avoid non realistic solutions such as the case of the inner width being larger than the external width. Alternatively, another possible choice, which is the one used in this thesis, is the external rectangle width and height and each side thickness. This selection avoids the aforementioned problems without the need of adding more constraints which would add unnecessary complexity to the code.

6.2 Dimensional analysis

When analysing the results obtained by Cox and Overton and by A.Mañeru, it was noticed that the solutions at which they arrived hardly fulfilled the slenderness assumptions in the Euler-Bernoulli theory (i.e. that the beam must always have one dimension much bigger than the other two), used to develop the governing equations of the problem. Since sometimes the diameter of the section was longer than the unitary length of the column. This can't be considered a historically mistake in solving the buckling optimization problem. The idea

was to use a non-dimensional approach to the problem by using a column of unitary length, volume and even with a Young's modulus of value 1.

However, in this thesis it has been decided to use a more realistic approach in obtaining the results. Therefore a dimensional analysis is carried out for each type of cross section in order to fulfill an arbitrary slenderness ratio. This ratio is calculated by dividing the length of the column by a characteristic parameter of each type of section, for example for the case of a circular section the diameter is selected as a characteristic parameter. Once the correspondent parameter is calculated, then the cross sections reference area is computed. Finally, multiplying the area by the length the volume is obtained.

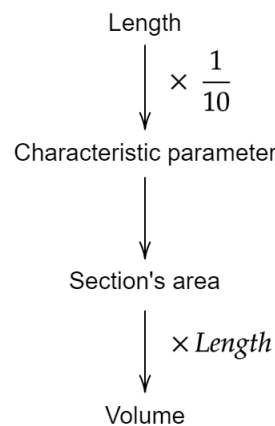


Figure 6.7: Schematic representation of the dimensional analysis

This analysis is used to establish the maximum volume constraint given a determined column length.

Chapter 7

Sensitivity and results analysis

In this section, a sensitivity analysis with the results of different cases is carried out. The aim is to study the influence of different simulation parameters in the results obtained. Several simulations are run changing different parameters such as the optimizer type, the initial values, the volume and design variable restrictions for each kind of design variable and cross section.

All the cases are based on a column of 20 units length and using a 500 elements mesh. Each case takes as a reference the volume and characteristic parameter values obtained from the pertinent dimensional analysis. However, those values are not considered immovable, since the objective of those analysis is to study its influence on the models, and they have to be varied to do so. Another point of view could be that of a reference uniform section column that takes as parameters the length, width and volume maximum values obtained from the dimensional analysis, which is then used to evaluate the slenderness of the solutions.

7.1 Circular section

The circular cross-section is a 1 design variable section. In this analysis it has been defined using the radius r .

The circular section column takes as a characteristic parameter the diameter of the section. Therefore, considering a 20 units length column, the dimensional analysis gives a radius of 1, and taking the column of constant circular section the maximum volume obtained is 20π .

Below, some of the most significant results obtained for the case of the circular cross section column of 20 units length are analysed. The design variables distribution of each result as well as its correspondent 3D column are shown in figure [7.1](#).

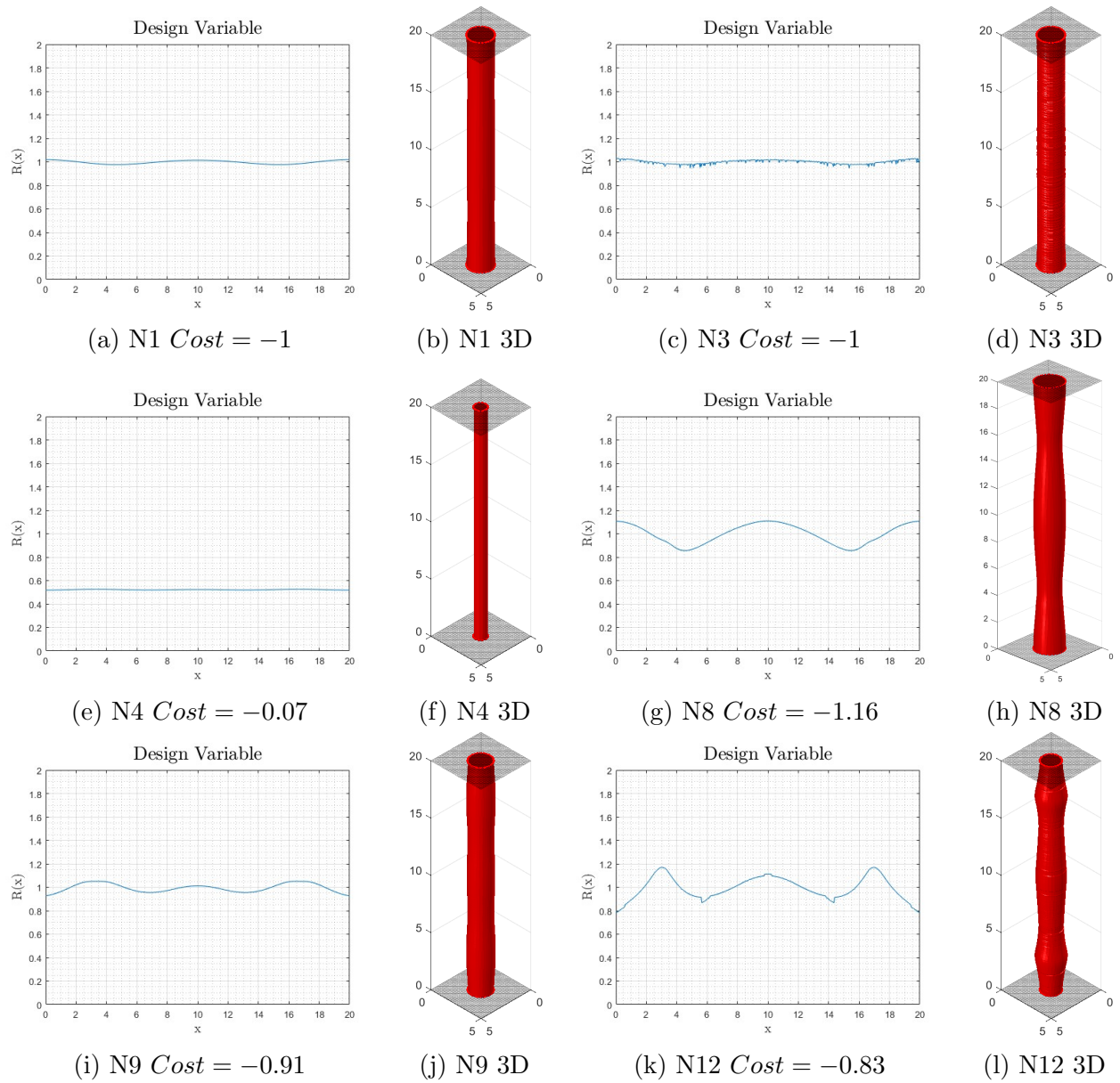


Figure 7.1: Circular section results

Reference case

Among all the results, a the reference case is first proposed. This case is characterized for having a constant radius of value 1 along the length. In other words, it is taking the geometry of the reference column obtained from the dimensional analysis. This basic solution obtains a cost of value -1 and all of them are obtained using *fmincon* optimizer. In table [7.1](#) the initial configurations that arrive to this solution are shown. The initial values can be constant or using a random distribution. The basic random distribution provides values from 0 to 1, however, this distribution can be escalated by multiplying it by a constant value.

Table 7.1: Reference cases initial configuration

N	Optimizer	Max. volume	Initial values		Des var restrictions		Cost
			r	Max	Min		
1	fmincon	20π	2	2	-	-1	
2	fmincon	20π	2*random	-	-	-1	
3	fmincon	20π	0.5*random	-	-	-1	

Adding the design variable constraints

When the maximum design variable is constrained to 1, the value of the radius in the reference column, the solutions obtained are achieving a maximum radius value of 0.5, as if the optimizer hindered the column to achieve higher values. Therefore, the solutions are never using the maximum volume available and the cost is lower in consequence.

Table 7.2: Hindered cases initial configuration

N	Optimizer	Max. volume	Initial values		Des var restrictions		Cost
			r	Max	Min		
4	fmincon	20π	2	1	-	-0.07	
5	fmincon	20π	0.5	1	-	-0.07	
6	fmincon	20π	2*random	1	-	-0.06	
7	fmincon	20π	0.5*random	1	-	-0.07	

Optimal case

An optimal solution is obtained using the *fmincon* optimizer, a constant initial value of $r = 2$ and no restrictions to the design variable maximum nor minimum value. This solution obtains a cost of $-1,16$, higher than in the basic case. The final geometry is wider at the ends and at the center of the column while thinner at $1/4$ and $3/4$ of the length. This fact resembles to the unitary column obtained by Cox and Everton, and seems to be a general tendency of the optimal columns obtained.

Table 7.3: Optimal case initial configuration

N	Optimizer	Max. volume	Initial values	Des var restrictions		Cost
			r	Max	Min	
8	fmincon	20π	2	-	-	-1.16

Other solutions

Other solutions, which are not optimal but that present another interesting geometry are shown below. This cases are obtained both with fmincon and MMA optimizer, however the design variable distribution is more accentuated using the MMA.

Table 7.4: Other cases initial configuration

N	Optimizer	Max. volume	Initial values	Des var restrictions		Cost
			r	Max	Min	
9	fmincon	20π	0.5	2	-	-0.91
10	fmincon	20π	2*random	2	-	-1.01
11	MMA	20π	0.5	2	0.3	-0.84
12	MMA	20π	2	2	0.3	-0.83

7.2 Circular section with inner concentric hole

The circular section with an inner concentric hole is a 2 design variables section. It is defined using the inner radius (r) and the thickness (e) as design variables. This means that this section is always symmetrical, despite the different configurations of the variables.

As a characteristic parameter to use for the dimensional analysis, for this section the external radius is selected. Therefore, the reference parameters used are the same as the circular section case.

As the other types of cross-sections presented, a different variety of results are obtained, having a strong influence the different initial configurations.

As initial values for this section, inner radius of 1 and 2, and thickness of $e = 0.3$ are considered, for both constant and random. Also, a sinusoidal distribution of the form

$$\begin{aligned} r &= 0.4 \cdot \sin(\theta) + 1 \\ e &= 0.2 \cdot \sin(\theta) + 0.3 \end{aligned} \tag{7.1}$$

is proposed, where θ goes from 0 to 5π radians.

As optimizers, *fmincon* and *MMA* are used. The maximum volume restriction is set to $20 \cdot \pi$, although some sensitivity analysis is also performed on the volume. And the maximum design variable restrictions are varied between non restriction at all, 2, 4 and 6. For the *MMA*, a minimum restriction is also required for the optimizer to work well, so minimum of 0,2 is set in those cases.

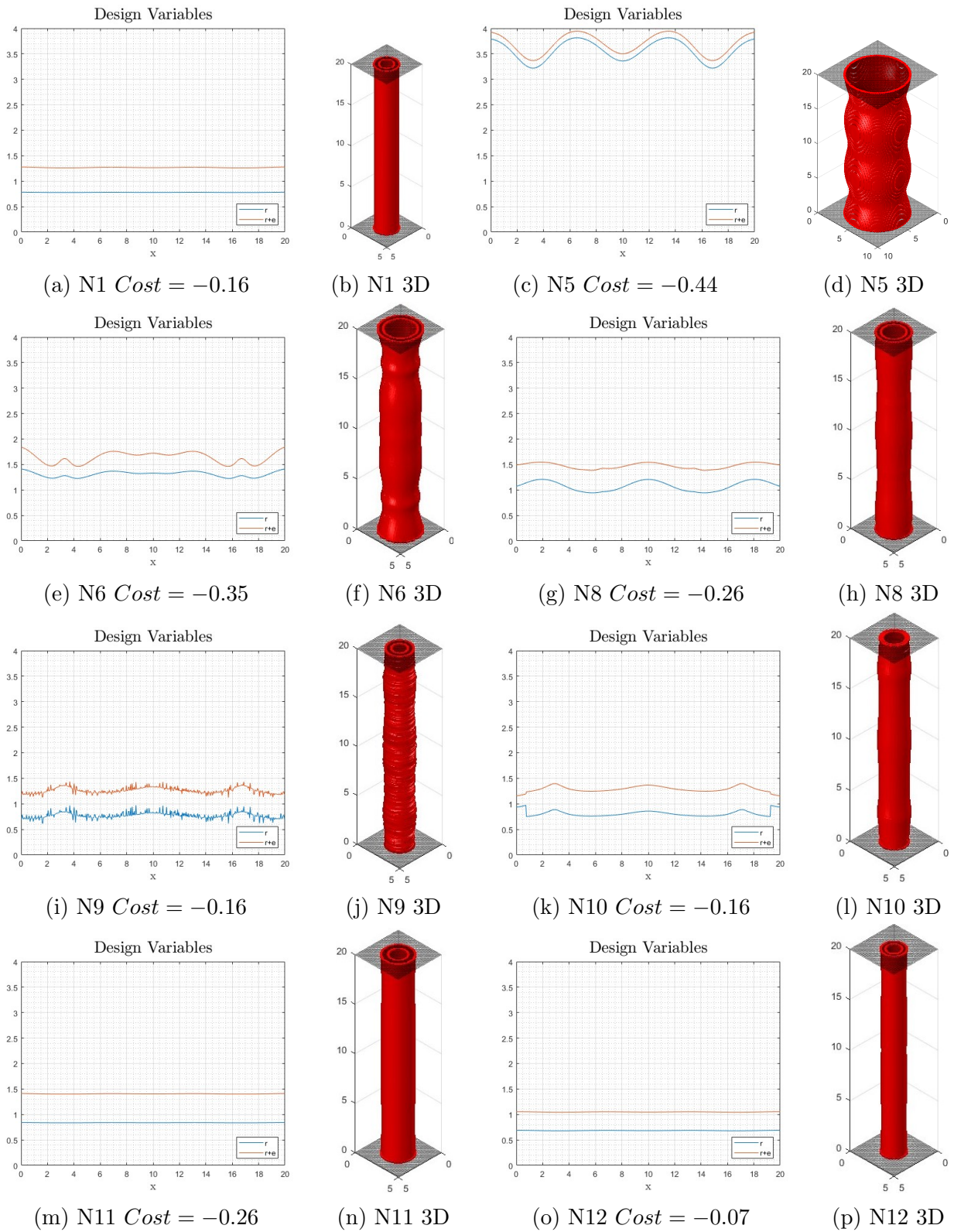


Figure 7.2: Circular with inner hole section results

Reference case

Among all the solutions obtained, the reference case can be differentiated, i.e. a column with a constant section along all the length. This base column takes an almost constant inner radius of $r = 0.8$ and a thickness of $e = 0.5$. Taking all the volume available, and therefore making the volume constraint 0. The cost reached for this solutions is -0.16 .

The different initial configurations that arrive to this solution are shown in table [7.8](#). Using MMA optimizer, a similar solution is achieved for the $r = 1$ and $e = 0.3$ initial values and $2/0.2$ as maximum and minimum design variables restriction, however, this solution is a little bit wider, obtaining a final constant radius $r = 1$ and thickness $e = 0.4$.

Table 7.5: Reference cases initial configuration

N	Optimizer	Max. volume	Initial values		Des var restrictions		Cost
			r	e	Max	Min	
1	fmincon	20π	1	0.3	2	-	-0.16
2	fmincon	20π	2	0.3	2	-	-0.16
3	fmincon	20π	random	$0.3*\text{random}$	2	-	-0.16
4	fmincon	20π	$2*\text{random}$	$0.3*\text{random}$	2	-	-0.16

Optimal cases

Another "family" of solutions obtained are the optimal ones. That is the solutions that are achieving a lower cost that -0.16 . Inside this group different solutions are found.

Table 7.6: Optimal cases initial configuration

N	Optimizer	Max. volume	Initial values		Des var restrictions		Cost
			r	e	Max	Min	
5	fmincon	20π	1	0.3	-	-	-0.44
6	fmincon	20π	1	0.3	4	-	-0.35
7	fmincon	20π	1	0.3	6	-	-0.33
8	fmincon	20π	sin	sin	4	-	-0.26

The best solution in terms of cost (-0.44) is the one depicted in figure [7.2c](#), case N5. However, due to not having any maximum design variable restriction, the optimizer here is looking for a solution as wider an thinner as possible. Arriving at values of $r = 3.8$ and $e = 0.15$, which are far from the reference constant section column of $r = 1$ obtained from the dimensional analysis.

Other interesting solutions, which have higher cost values than -0.44 , but that are more slender than the previous one are obtained giving as maximum design variable restriction values of 4 and 6, i.e. letting the optimizer consider wide solutions but with some limit imposed. (Case N6: Fig. 7.2e) For the first case, it is necessary to highlight how the optimizer first finds a constant section solution of $-0,25$ cost, however around the 25th iteration, a better solution is found which is the -0.35 cost. This behaviour can be observed in the cost monitoring graphic in figure 7.3.

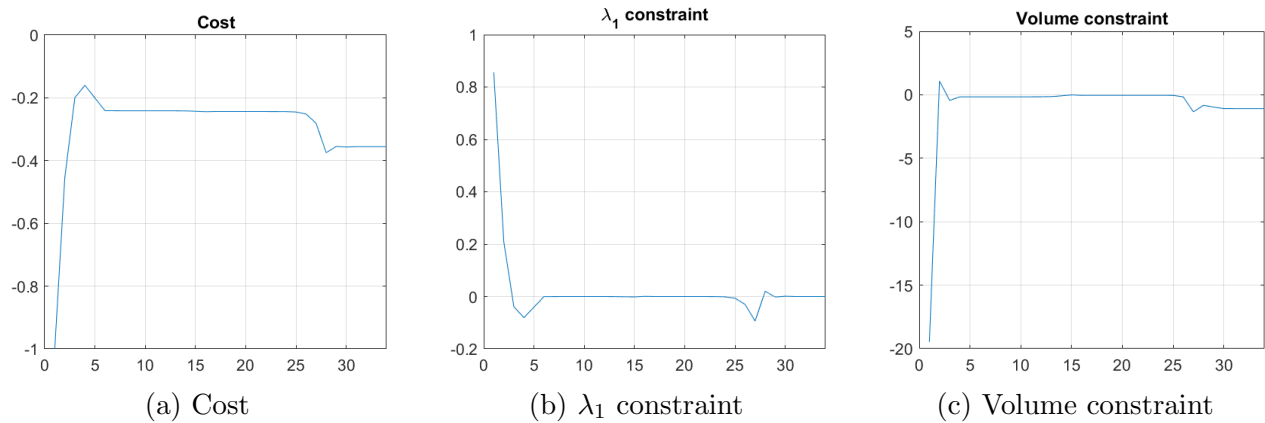


Figure 7.3: N6 monitoring

Finally, inside this group of better solutions than the basic case, a $-0,26$ cost solution is found using a sinusoidal initial distributions with the *fmincon* optimizer. This is achieved using a maximum design variable restriction of 4, since the same conditions with a restriction of 2 are giving the base case as a solution. (Case N8: Fig. 7.2g)

Other solutions

Apart from the cases already mentioned, other solutions are obtained that may not have lower cost values than the base case (-0.16), but that are arriving at solutions with different geometries than the constant section column.

Table 7.7: Other cases initial configuration

N	Optimizer	Max. volume	Initial values		Des var restrictions		Cost
			r	e	Max	Min	
9	MMA	20π	random	$0.3*\text{random}$	2	0.2	-0.16
10	MMA	20π	2	0.3	2	0.2	-0.16
11	fmincon	80	1	0.3	2	-	-0.26
12	fmincon	40	1	0.3	2	-	-0.07

Using the MMA optimizer two similar solutions are obtained which have a particular geometry consisting on two small peaks at the ends and a wider one in the middle. Both solutions achieve a cost of -0.16 and fulfill the slenderness requirements having inner radius of around $r = 0.9$. (Case N9: Fig. [7.2i](#) and Case N10: Fig. [7.2k](#))

Another set of cases analyse the sensibility of the volume constraint considering as maximum volume values of 80 and 40, to analyse the influence of higher and lower volumes keeping the maximum design variable value possible as 2. The solutions obtained in both cases can be considered as equivalents to the base case, with a constant cross section, but with greater and lower values respectively. (Case N11: Fig. [7.2m](#) and Case N12: Fig. [7.2o](#))

7.3 Rectangular section

The rectangular cross-section is a 2 design variables section. It is defined using both sides of the rectangle which are named as a and b .

The characteristic parameter used to analyse the slenderness of the beam is the length of the longest side, which for the case of a 20 units length column is set to 2. Meaning that for a constant section beam with both sides being 2, the maximum volume is 80.

Between the different initial configurations studied, the initial values of the sides used are: both sides equal at a value of 3, and one side longer than the other with the values 3 and 0.5. Also, initial values following a sinusoidal distribution of the form

$$\begin{aligned} a &= 0.4 \cdot \sin(\theta) + 2 \\ b &= 0.4 \cdot \sin(\theta) + 2 \end{aligned} \tag{7.2}$$

are tried. Where θ goes from 0 to 5π radians. The *MMA* and the *fmincon* optimizers are used in this case. The design variables maximum restrictions used are no restrictions, 2 and 4. With a minimum restriction for the MMA 0.3.

Following, some of the most relevant results are analysed.

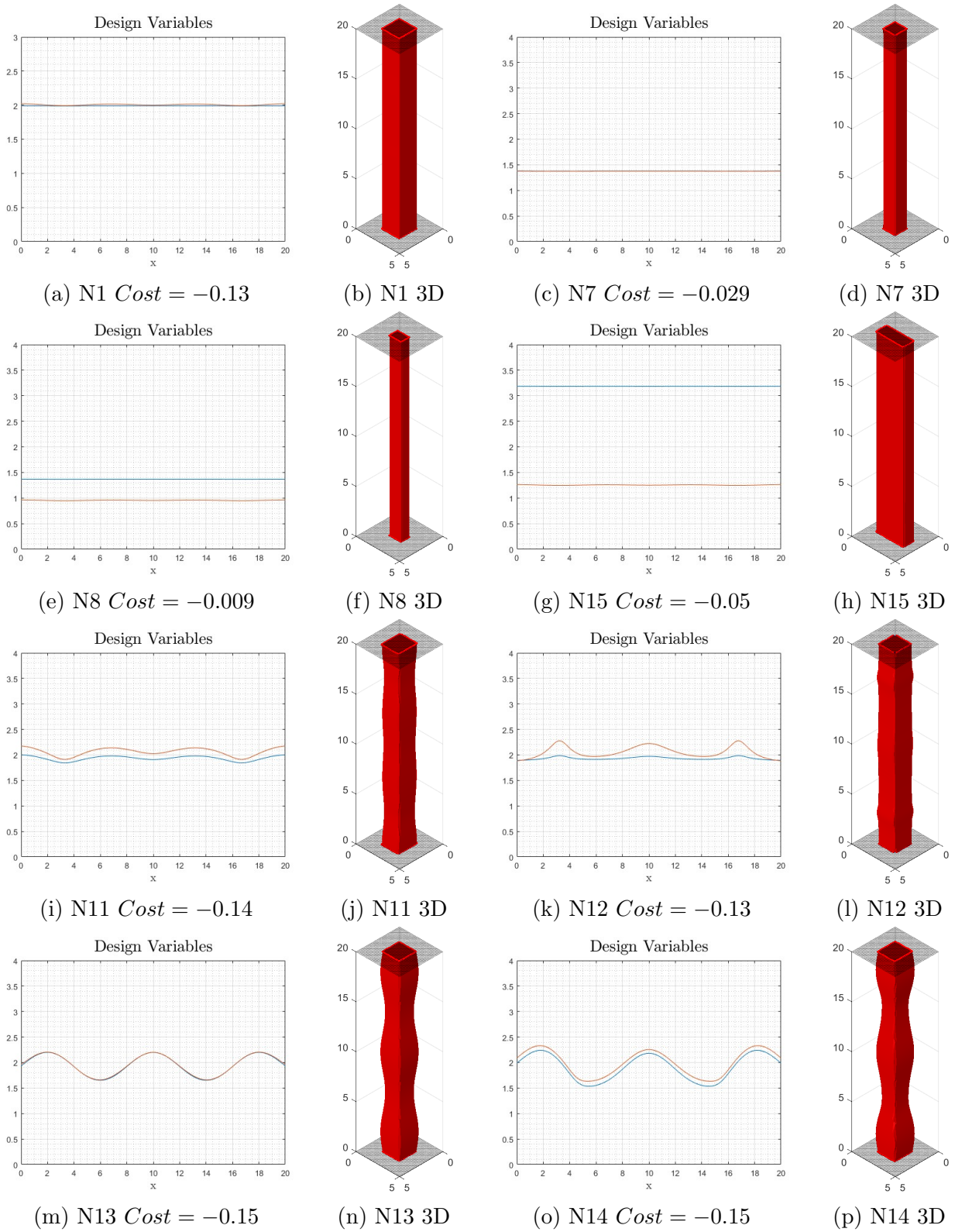


Figure 7.4: Rectangular section results

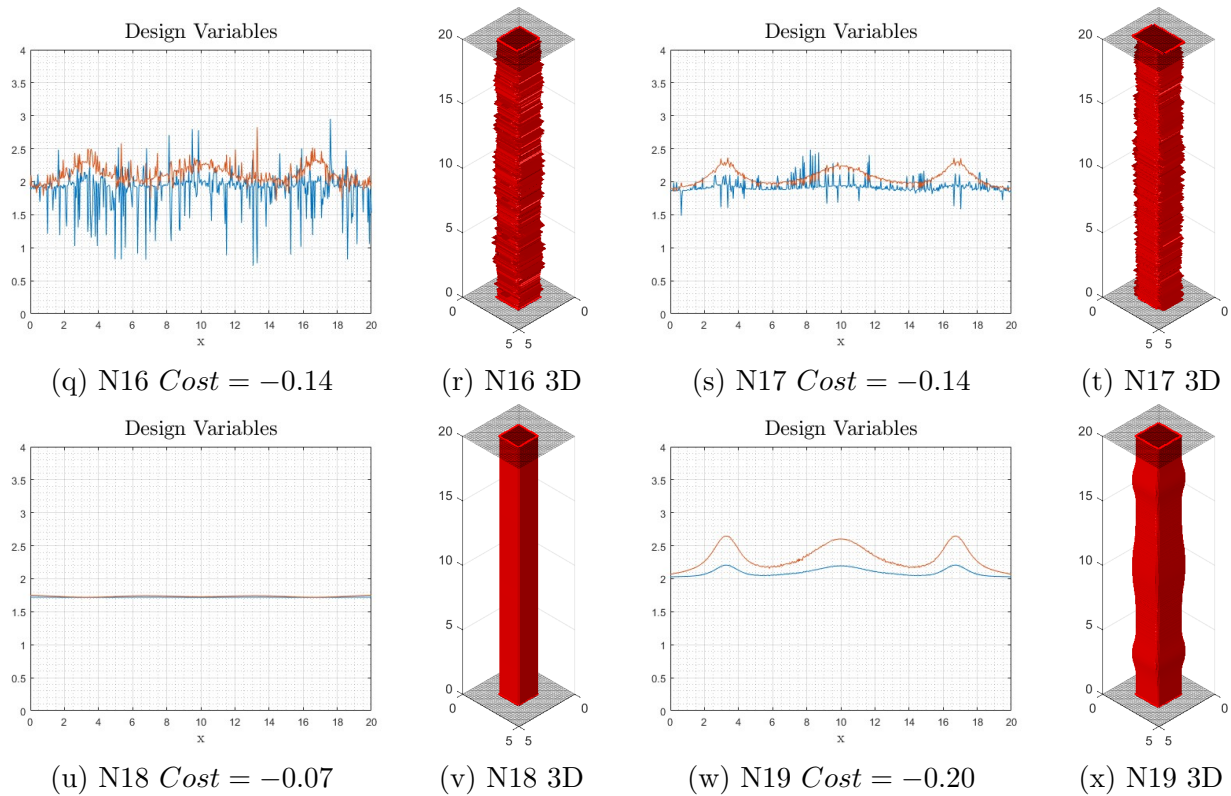


Figure 7.4: Rectangular section results

Reference case

As with the other cross-sections cases presented, the reference case is differentiated among all the solutions. This case is characterized by a constant cross-section along the length of the beam. This section takes the form of a square with both sides being almost equal, with a value of 2, i.e. taking the maximum volume available in the form of the reference slender column. This solution obtains a cost of -0.13 and is achieved only by *fmincon* optimizer when there is no maximum design variable restriction or when the restriction is 4.

Table 7.8: Reference cases initial configuration

N	Optimizer	Max. volume	Initial values		Des var restrictions		Cost
			a	b	Max	Min	
1	fmincon	80	3	3	-	-	-0.13
2	fmincon	80	3	3	4	-	-0.13
3	fmincon	80	3	0.5	4	-	-0.13
4	fmincon	80	sin	sin	-	-	-0.13
5	fmincon	80	3*rand	3*rand	4	-	-0.12
6	fmincon	80	3*rand	0.5*rand	4	-	-0.12

Adding design variable constraints

A curious phenomena is observed when the maximum design variable restriction is set to 2. It would be logical to think that the optimizers still can arrive to the base case of equal sided section of values $a, b = 2$. However, all the solutions achieved using this restriction are obtaining maximum side values of 1.4, and in no case a side value of 2 is obtained. Therefore, they are never using the maximum volume available and the volume constraint is never 0. Furthermore, all those solutions are of constant section, thinner square columns than the basic case. The costs of those solutions varies form -0.008 to -0.029 .

Table 7.9: Hindered cases initial configuration

N	Optimizer	Max. volume	Initial values		Des var restrictions		Cost
			a	b	Max	Min	
7	MMA	80	3	3	2	0.3	-0.029
8	MMA	80	3	0.5	2	0.3	-0.009
9	fmincon	80	3	3	2	-	-0.008
10	fmincon	80	3	0.5	2	-	-0.008

Optimal solutions

Although no much lower than -0.13 cost solutions are obtained, there is a group of solutions with different geometries from the constant section column which give values between -0.13 and -0.15 cost.

Table 7.10: Optimal cases initial configuration

N	Optimizer	Max. volume	Initial values		Des var restrictions		Cost
			a	b	Max	Min	
11	MMA	80	3	3	4	0.3	-0.14
12	MMA	80	3	0.5	4	0.3	-0.13
13	fmincon	80	sin	sin	4	-	-0.15
14	MMA	80	sin	sin	4	0.3	-0.15

Using MMA optimizer, depending on the initial values, two slightly different geometries are found. Starting with $a, b = 3$ and with a maximum restriction of 4, case 11, a cost of -0.14 is obtained. In figure [7.4i](#), the design variable values can be observed, being the column wider at the ends and the middle. On the other hand, beginning with $a = 3$ and $b = 0.5$, the cost is -0.13 and the geometry is different. One of the sides is of almost a constant value 2, while the other is showing three peaks at 3, 10 and 17 units length. Consequently augmenting the general area in those parts. (Fig. [7.4k](#))

If we use as initial conditions a sinusoidal distribution of the design variables, eq. [7.2](#), the final geometry is different. Both cases, N13 and N14, using fmincon and MMA respectively, obtain a sinusoidal solution with a cost of -0.15. Therefore, those cases are the optimal ones from all the configurations tested.

Other solutions

Other solutions different from the aforementioned, which are not optimal but that arrive to distinct geometries, are presented below.

Table 7.11: Other cases initial configuration

N	Optimizer	Max. volume	Initial values		Des var restrictions		Cost
			a	b	Max	Min	
15	fmincon	80	3	0.5	-	-	-0.05
16	MMA	80	3*rand	3*rand	4	0.3	-0.14
17	MMA	80	3*rand	0.5*rand	4	0.3	-0.14
18	fmincon	60	3	3	4	-	-0.07
19	fmincon	100	3	3	4	-	-0.20

Using the *fmincon* optimizer, starting with initial side values of $a = 3$ and $b = 0.5$ and without imposing any design variable restriction a constant rectangular section column is obtained. It

arrives to a cost of -0.05 and the sides take a constant value of $a = 3.18$, $b = 1.25$. (Case N15: Fig. [7.4g](#))

Using MMA optimizer, and starting with a random initial distribution of the design variables, solutions N16 and N17 are found. The resulting geometries are unrealistic and hardly feasible, however a tendency can be sensed towards a solution similar to the one obtained in case N12.

Finally, two more cases, N18 and N19, study the influence of the maximum volume constraint. In case 18 the volume constraint is set at 60, and the result is the basic case, i.e. constant square section, but with lower side values, that are 1.7 instead of 2. However, when the maximum volume is 100, (case 19) instead of arriving to the basic solution, a slightly different geometry is obtained resembling the one obtained in case N12.

Chapter 8

Conclusions

The main objective of this thesis has been to investigate new designs of lightweight structures to maximize the buckling critical load, which is the first eigenvalue of the problem. With the exception of the implementation of the rectangular section with an inner hole, that due to a lack of time hasn't been possible to make it work correctly, all other objectives have been successfully accomplished.

A change of design variable has been implemented, for the same circular cross-section shape that was studied by A. Mañeru. Her work, as well as the historical research done in this field, was developed using the circular section's area (A) as a design variable. However, in this thesis, the radius (r) has been implemented as design variable. The results obtained using the radius have been consistent with those obtained using the area. In particular, the -51 cost value achieved by Cox and Everton has been obtained for the case of a unitary volume and length column.

However, the primary focus in this thesis has been the research of new designs for maximizing buckling first eigenvalues. Two new cross-section geometries were studied, and their solutions were analyzed. Adapting the code to solve columns with more than one design variable was required for these new sections, such as the rectangular section or the circular section with a concentric inner hole.

After analyzing the solutions of the circular, circular with inner hole, and rectangular sections, some general conclusions have been drawn.

It has been observed that in all three sections a basic case is obtained which consists in a constant section column. This basic solution takes the form of the reference column obtained from the dimensional analysis, as is the case of the cylinder for the circular section case and the square prism for the rectangular case.

Another characteristic observed is related to the functioning of the design variables restriction.

When the maximum value of the design variable is set as the reference value obtained from the dimensional analysis, e.g. 2 for the rectangle case, the solutions obtained never reach that value, instead they stay always some margin below it. However, when a higher margin is introduced, the solutions do reach the reference value, e.g. when the maximum value is set to 4 for the rectangle case the obtained values are around 2. In other words, the reference values are never reached if the same reference value is used as a maximum restriction. This behaviour is related to the way the optimizers look for the optimal solution and is a feature that could be deepened in further studies.

Among all the solutions that have been studied there are two kind of distributions of the design variable that appear repeatedly in the optimal cases. One of those solutions is similar to the results obtained by Cox and Overton in the sense that distributes the material making the column wider at its ends and center (e.g. fig. [7.1g](#)). The other solution tends to expand the section at around $1/6$ and $5/6$ of the length and at its center (fig. [7.4k](#)).

Future lines of investigation

As further lines of investigation, the development of the buckling optimization problem can be broadened to other cross sections types, for example, regarding the typical sections used in industrial designs, as are the I and T-shaped sections. Parametrizing the sections with other design variables. For example, the circular hole case could not necessarily be concentric, adding more freedom to the generation of solutions.

Other studies could address the implementation of different materials in the optimization of the column. This change would be introduced by changing the Young's modulus (E), and the study could go from the influence of different materials in the final solutions obtained to the development of more than one material designs.

Finally fixing the implementation of the rectangular section with a rectangular hole, which was a initial feature in the scope of the thesis but that in the end has failed to be implemented, is the most direct way of continuing the work developed in this project.

Bibliography

- [1] E. Lund A.P. Seyranian and N. Olhoff. “Multiple eigenvalues in structural optimization problems”. In: *Structural and Multidisciplinary Optimization* 8 (1994), pp. 207–227.
- [2] Ton Creus Costa. “Study of numerical methods for the design of large lightweight structures”. Universitat Politècnica de Catalunya, 2022.
- [3] Steven J. Cox and Michael L. Overton. “On the optimal design of columns against buckling”. In: *SIAM Journal on Mathematical Analysis* 23.2 (1992), pp. 287–325.
- [4] Benoît Descamps and Rajan Filomeno Coelho. “A lower-bound formulation for the geometry and topology optimization of truss structures under multiple loading”. In: *Structural and Multidisciplinary Optimization* 48 (2013), pp. 49–58.
- [5] Jacob Fish and Ted Belytschko. *A first course in finite elements*. John Wiley & Sons Ltd, 2007. ISBN: 9780470035801.
- [6] Karam Y. Maalawi. “Buckling optimization of flexible columns”. In: *International Journal of Solids and Structures* 39 (2002), pp. 5865–5876.
- [7] Aitziber Mañeru. “Study of numerical tools for the design of lightweight structures when considering the modal analysis”. Universitat Politècnica de Catalunya, 2022.
- [8] Lucía Gómez Rodríguez. “Optimal design of columns against buckling”. Universidad de Castilla-La Mancha, 2014.
- [9] A.P. Seyranian and O.G. Privalova. “The Lagrange problem on an optimal column: old and new results”. In: *Structural and Multidisciplinary Optimization* 25 (2003), pp. 393–410.
- [10] Stephen P. Timoshenko and James M. Gere. *Theory of elastic stability*. 2nd. Mc Graw Hill, 1985. ISBN: 0-07-Y85821-7.
- [11] Stephen Boyd; Lieven Vandenbergh. *Convex Optimization*. 7th ed. New York: Cambridge University Press, 2009. ISBN: 9780521833783.

Appendix A

Cross-sections

Following, the main parameters of the different cross sections implemented in the code are detailed. The design variables that uses each kind of section as well as the area, inertia and its derivatives equations are presented.

A.1 Circular section

To define the circular section, the radius (r) is selected as design variable.

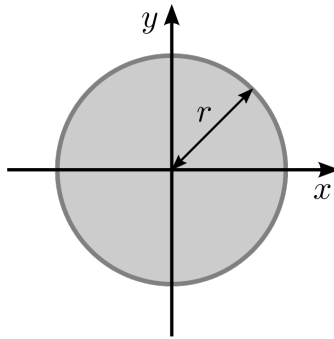


Figure A.1: Circular section

$$A = \pi r^2 \tag{A.1}$$

$$\frac{\partial A}{\partial r} = 2\pi r \tag{A.2}$$

$$I = \frac{\pi}{4} r^4 \quad (\text{A.3})$$

$$\frac{\partial I}{\partial r} = \pi r^3 \quad (\text{A.4})$$

A.2 Square section

The square section is a particular one design variable case of the rectangular section. To define it the length of one side is used as design variable (a).

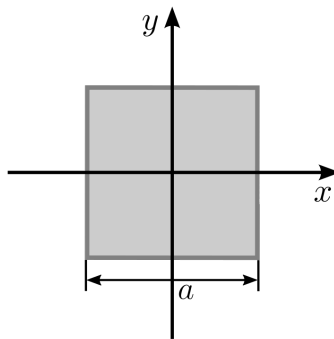


Figure A.2: Square section

$$A = a^2 \quad (\text{A.5})$$

$$\frac{\partial A}{\partial a} = 2a \quad (\text{A.6})$$

$$I = \frac{a^4}{12} \quad (\text{A.7})$$

$$\frac{\partial I}{\partial a} = \frac{a^3}{3} \quad (\text{A.8})$$

A.3 Rectangular section

The rectangular section is a two design variables section which is characterized by the length of each side a, b .

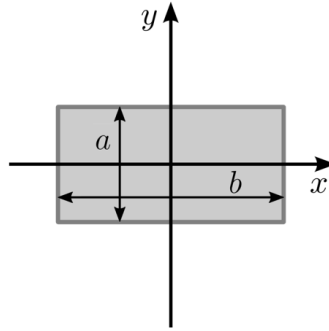


Figure A.3: Rectangular section

$$A = a b \quad (\text{A.9})$$

$$\frac{\partial A}{\partial a} = b \quad (\text{A.10})$$

$$\frac{\partial A}{\partial b} = a \quad (\text{A.11})$$

$$I = \frac{a b^3}{12} \quad (\text{A.12})$$

$$\frac{\partial I}{\partial a} = \frac{b^3}{12} \quad (\text{A.13})$$

$$\frac{\partial I}{\partial b} = \frac{a b^2}{4} \quad (\text{A.14})$$

A.4 Circumference with concentric hole section

The circular section with concentric inner hole is a two design variables section. To parametrize it, the inner radius and the thickness have been selected as design variables.

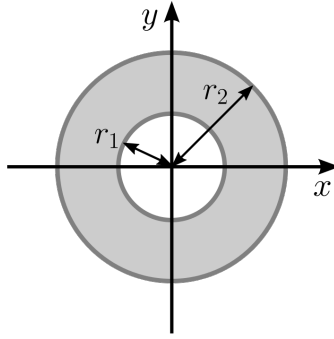


Figure A.4: Circular section with concentric inner hole

$$A = \pi(R_{ext}^2 - R_{int}^2) = \pi((r + e)^2 - r^2) \quad (\text{A.15})$$

$$\frac{\partial A}{\partial r} = 2\pi e \quad (\text{A.16})$$

$$\frac{\partial A}{\partial e} = 2\pi(r + e) \quad (\text{A.17})$$

$$I = \frac{\pi}{4}(R_{ext}^4 - R_{int}^4) = \frac{\pi}{4}((r + e)^4 - r^4) \quad (\text{A.18})$$

$$\frac{\partial I}{\partial r} = \pi((r + e)^3 - r^3) \quad (\text{A.19})$$

$$\frac{\partial I}{\partial e} = \pi(r + e)^3 \quad (\text{A.20})$$

where r is the inner radius and e the thickness so that the external radius is $r + e$.

A.5 Rectangular section with rectangular inner hole

The rectangular section with a rectangular inner hole is a four design variables section. To characterize it the length of the external sides and the width of each side are elected as design variables.

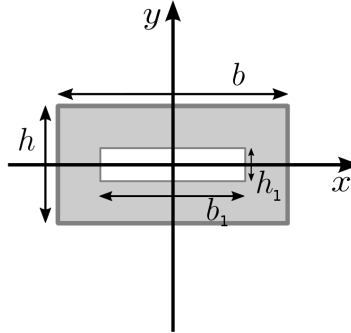


Figure A.5: Hollow rectangle

$$h, b, e_h, e_b$$

$$b_1 = b - 2 \cdot e_b \quad (\text{A.21})$$

$$h_1 = h - 2 \cdot e_h \quad (\text{A.22})$$

$$I_x = \frac{bh^3 - b_1h_1^3}{12} = \frac{bh^3 - (b - 2 \cdot e_b)(h - 2 \cdot e_h)^3}{12} \quad (\text{A.23})$$

$$\frac{\partial I_x}{\partial b} = \frac{h^3 - (h - 2 \cdot e_h)^3}{12} \quad (\text{A.24})$$

$$\frac{\partial I_x}{\partial h} = \frac{3bh^2 - 3(b - 2 \cdot e_b)(h - 2 \cdot e_h)^2}{12} \quad (\text{A.25})$$

$$\frac{\partial I}{\partial e_b} = \frac{(h - 2 \cdot e_h)^3}{6} \quad (\text{A.26})$$

$$\frac{\partial I}{\partial e_h} = \frac{(b - 2 \cdot e_b)(h - 2 \cdot e_h)^2}{2} \quad (\text{A.27})$$

$$A = h \cdot b - h_1 \cdot b_1 = h \cdot b - (h - 2 \cdot e_h) \cdot (b - 2 \cdot e_b) \quad (\text{A.28})$$

$$\frac{\partial A}{\partial b} = h - (h - 2 \cdot e_h) \quad (\text{A.29})$$

$$\frac{\partial A}{\partial h} = b - (b - 2 \cdot e_b) \quad (\text{A.30})$$

$$\frac{\partial A}{\partial e_b} = 2 \cdot (h - 2 \cdot e_h) \quad (\text{A.31})$$

$$\frac{\partial A}{\partial e_h} = 2 \cdot (b - 2 \cdot e_b) \quad (\text{A.32})$$