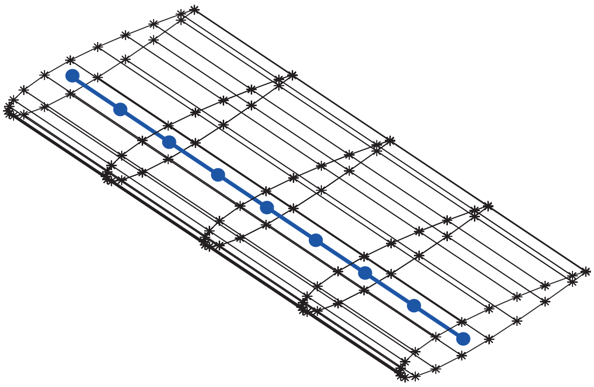


Efficient Structural Optimization of Aircraft Wings

Tiago Alexandre Reis Ramalho Moutinho Freire

Thesis to obtain the Master of Science Degree in

Aerospace Engineering

Supervisor: Prof. André Calado Marta

Examination Committee

Chairperson: Prof. Filipe Szolnoky Ramos Pinto Cunha
Supervisor: Prof. André Calado Marta
Member of the Committee: Prof. Aurélio Lima Araújo

February 2017

Acknowledgments

I would like to thank my supervisor, Professor André Calado Marta, for all his support and guidance throughout the development of this thesis. His vast knowledge in engineering and wise advice on how to tackle problems were essential to produce this work.

I wish to express my gratitude to my friends João Almeida, for taking the time to explain the structure of the original framework he developed and for always being there to clarify my doubts, and to Duarte Rondão for proofreading this dissertation and giving insightful advice.

I would also like to thank the amazing people I met during my studies in Lisbon, Delft and Victoria, and to express my gratitude for having them around in good and bad moments.

Last but not least, I would like to thank my family, specially my mother, for always believing in my abilities and encouraging me to follow my dreams.

Resumo

Actualmente, os métodos de gradiente constituem uma das ferramentas mais utilizadas na optimização multidisciplinar de aeronaves. No entanto, estes métodos necessitam de calcular a sensibilidade das funções de interesse em relação às variáveis de projecto, constituindo uma das etapas de maior exigência computacional no processo de optimização, uma vez que estas são frequentemente obtidas por métodos de aproximação altamente dependentes da dimensão do problema. Assim, este trabalho tem como objectivo desenvolver uma ferramenta de optimização eficiente, para uma fase preliminar do projecto de uma asa, com a utilização da informação exacta do gradiente. Primeiramente, é realizado um levantamento dos vários métodos de análise de sensibilidade existentes com exemplos de aplicação ao projecto de vigas modeladas com elementos finitos. Posteriormente, uma ferramenta para a representação estrutural de uma asa tridimensional é adaptada em três blocos com o desenvolvimento dos respectivos módulos para o cálculo das sensibilidades, utilizando os métodos de diferenciação automática, derivação simbólica e adjunto. Um estudo paramétrico é apresentado com base numa asa de referência e as sensibilidades totais são calculadas com a ferramenta desenvolvida e verificadas com o método das diferenças finitas. Por fim, são apresentados exemplos de optimização estrutural partindo do caso de referência. O objectivo de minimização da massa total da asa é alcançado, sendo notável o aumento da eficiência no processo de optimização com a utilização da ferramenta desenvolvida, traduzido pela redução dos tempos de computação para, aproximadamente, metade e um terço, quando comparados com os métodos das diferenças finitas progressivas e centradas, respectivamente.

Palavras-chave: optimização estrutural, métodos de gradiente, análise de sensibilidade, método adjunto, diferenciação automática, método dos elementos finitos.

Abstract

Nowadays, gradient-based methods are one of the most widely used tools in aircraft Multidisciplinary Design Optimization. However, these methods require the computation of the sensitivities of the interest functions with respect to the design variables, representing one of the most computationally expensive steps in the optimization process, since these are frequently obtained by approximation methods that are highly dependent on the number of design variables. Therefore, the main objective of this work is to develop an efficient optimization tool for wing preliminary design, using exact gradient information. Firstly, a survey on the existent sensitivity analysis methods is conducted, with the application to a beam design problem modeled with finite elements, providing valuable insight in the implementation process and advantages of each method. Subsequently, a tool to represent a 3D wing structure is adapted into three blocks and the correspondent modules for sensitivity computation are developed, with the application of the automatic differentiation, the symbolic derivative and the adjoint methods. A parametric study is presented for a reference wing case and the total sensitivities are computed with the developed framework and verified with the finite difference method. Lastly, structural optimization tests, using the reference case as the initial design point, are performed. The objective of minimizing the wing mass is achieved with a remarkable increase in computational efficiency in the optimization process, translated in a reduction of the computational time to, roughly, half and one third when compared to the forward difference and central difference methods, respectively.

Keywords: structural optimization, gradient-based methods, sensitivity analysis, adjoint method, automatic differentiation, finite element method.

Contents

Acknowledgments . . . iii

Resumo . . . v

Abstract . . . vii

List of Tables . . . xiii

List of Figures . . . xv

Nomenclature . . . xvii

Glossary . . . xxi

1 Introduction 1

1.1 Motivation . . . 1

1.2 Aeroelastic Tool Background . . . 3

1.3 Objectives . . . 5

1.4 Thesis Outline . . . 6

2 Optimization Methods 7

2.1 Optimization Background . . . 7

2.2 Gradient Based Methods . . . 8

2.2.1 Unconstrained Optimization . . . 8

2.2.2 Constrained Optimization . . . 10

2.3 Gradient-Free Methods . . . 11

2.4 Choice of the Method . . . 12

3 Sensitivity Analysis 13

3.1 Symbolic Differentiation . . . 13

3.2 Finite Difference Method . . . 14

3.3 Complex Step Method . . . 14

3.3.1 Finite Difference and Complex Methods Comparison . . . 15

3.4 Alternative Analytic Methods . . . 17

3.4.1 Direct Sensitivity Method . . . 17

3.4.2 Adjoint Sensitivity Method . . . 18

3.5 Automatic Differentiation . . . 18

3.6 Benchmark . . . 22

3.6.1 Model Definition . . . 23

3.6.2 Sensitivity Analysis . . . 27

4 Structural Framework 33
4.1 Equivalent Cross-sectional Properties . . . 34
4.1.1 Thin-Wall Assumption . . . 34
4.1.2 Implementation Procedure . . . 36
4.1.3 Application Example . . . 40
4.2 Equivalent Beam Element Properties . . . 41
4.3 Finite Element Model . . . 42
4.3.1 Constitutive Equations . . . 42
4.3.2 3D Finite Element Model Definition . . . 43
4.3.3 Strong and Weak formulations . . . 44
4.3.4 Domain Discretization . . . 44
4.3.5 Shape Functions . . . 45
4.3.6 Element Stiffness Matrix Formulation . . . 46
4.3.7 Force Vector Determination . . . 48
4.3.8 Bondary Conditions Application and Global System of Equations Solution . . . 50

5 Parametric Study 51
5.1 Wing Baseline Configuration . . . 51
5.2 Functions of Interest . . . 52
5.3 Internal Parameters Influence . . . 53
5.3.1 Spars Location . . . 53
5.3.2 Spar and Skin Thickness . . . 56
5.4 Material Parameters Influence . . . 58
5.5 Summary of Results . . . 59

6 Sensitivity Analysis Framework 61
6.1 Automatic Differentiation . . . 62
6.2 Symbolic Differentiation . . . 66
6.3 Adjoint Sensitivity Module . . . 67
6.3.1 Vertical Tip Displacement . . . 68
6.3.2 Tip Rotation . . . 70
6.4 Total Sensitivity . . . 72

7 Wing Structural Optimization 75
7.1 Optimizer . . . 75
7.2 Bound Constrained Optimization . . . 76
7.3 Fully Constrained Optimization . . . 78

8 Conclusions 83
8.1 Achievements . . . 83
8.2 Future Work . . . 84

Bibliography 85

A Automatic Sensitivity Analysis Benchmark with Finite Difference Methods 89

B Explicit Derivatives of the Element Stiffness Matrices 91

List of Tables

3.1 Elementary functions listing and chain rule sequential forward propagation 20

3.2 Beam parameters definition 27

3.3 Sensitivity analysis results with the adjoint method 29

3.4 Optimum steps and relative errors when evaluating the sensitivities of f_1 using forward and central difference methods 30

3.5 Comparison of the relative errors for different sensitivity analysis methods 32

4.1 Equivalent cross-sectional properties module inputs and outputs 34

4.2 Application example of the function `fequiprops.m` with respective input and output variables 40

4.3 Equivalent cross-sectional properties module inputs and outputs 41

4.4 Strong and weak formulations for the axial, bending and torsion problems 44

4.5 Shape functions 46

5.1 Reference wing baseline parameters 51

5.2 Effect of the parameters studied on the output functions 60

6.1 Sensitivities of the cross-sectional properties with respect to the internal parametric variables and material properties 63

6.2 Absolute value of the sensitivities errors using the forward finite difference method with a relative step size of 10^{-4} 64

6.3 Sensitivity of the shear center location with respect to the parameters x and the relative error using the forward finite difference method with a relative step size of 10^{-4} 65

6.4 CPU time (Δt) for calculating the first module's sensitivities 65

6.5 CPU time (Δt) for calculating all the third module's sensitivities 71

6.6 Sensitivity of the interest functions with respect to the geometrical wing parameters 73

6.7 Relative sensitivity errors of the interest functions with respect to the geometrical wing parameters, when using the forward finite difference method with a relative stepsize of 10^{-4} 73

7.1 Inital values and bounds of the design variables for the bounded optimization problem . . 77

7.2 Initial and final optimized design variable values and output functions for the bounded constrained problem 77

7.3 Benchmark of excecution time, iteration numbers and function evaluations for three different sensitivity calculation methods for the bounded optimization problem 78

7.4 Initial and final optimized design variable values and output functions for the fully constrained problem 79

7.5 Benchmark of excecution time, iteration numbers and function evaluations for three different sensitivity calculation methods, for the fully constrained optimization problem 80

A.1 Absolute values of the error of the sensitivities calculated computed for a step size of 10^{-6} 89

A.2 Absolute value of the sensitivities errors using the forward finite difference method with a relative step size of 10^{-8}. 89

List of Figures

1.1 Airbus global market forecast data 1

1.2 Aircraft concepts according to different disciplines 2

1.3 Flowchart illustrating the program‘s calculation process during aeroelastic analysis 3

1.4 Evolution of the internal wing structure during the optimization process 4

1.5 Evolution of the external wing structure during the optimization process 4

1.6 Flowchart illustrating the implementation structure of the aeroelastic framework 5

3.1 Error of sensitivity estimates given by the complex step, forward finite differences and central differences 16

3.2 Representation of the governing equation, interest function, design and state variables for an arbitrary system 17

3.3 Schematics of the automatic differentiation process 19

3.4 Decomposition of f_1 and f_2 in elementary functions 20

3.5 *ADiMat* application example. 22

3.6 Cantilevered beam representation 23

3.7 Finite element representation 24

3.8 Logarithmic plot of $\varepsilon(\frac{\mathrm{d}f_1}{\mathrm{d}b})$ as a function of the relative step size 31

4.1 Structural framework 33

4.2 Thin-wall wingbox approximation of a NACA0015 airfoil, represented in the airfoil’s reference frame 34

4.3 Thin-wall beam representation 35

4.4 Thin wall section represented in its centroidal reference frame 36

4.5 Wingbox open section representation, in the centroidal frame 37

4.6 Decomposition of a closed wingbox with shear flow distribution q_s into an open wingbox with a shear flow distribution q_{sb} and torque T_0, introducing a constant shear flow 39

4.7 Cross-sectional configuration and shear center location in the airfoil reference frame . . . 41

4.8 3D-beam element representation 43

4.9 Wing discretization example 45

4.10 3D-beam finite element in a global reference frame 45

4.11 Sequential rotations representation 48

4.12 Example of an aircraft wing and wake discretization and the respective collocation points 49

4.13 Fluid forces applied at each collocation point 50

5.1 Reference wing configuration 52

5.2 Wingbox configurations by changing the front and rear spars positions from the reference case 54

5.3 Influence of the spars locations in the wing mass and the root torsional moment 54
5.4 Influence of the spars location in the tip displacement and rotation 55
5.5 Influence of the spars location in the wing root maximum stress 55
5.6 Representation of the thicknesses of the wingbox segments 56
5.7 Influence of the wingbox segments' thickness in the wing mass and applied torsional moment 57
5.8 Influence of the wingbox segments' thickness in the tip vertical displacement and rotation 57
5.9 Influence of the wingbox segments' thickness in the wing root maximum stress 58
5.10 Influence of the material density in the wing mass 58
5.11 Influence of the material properties in the vertical displacement and rotation 59

6.1 Structural sensitivity analysis framework 61
6.2 Sensitivities and relative errors of the sensitivities of the vertical tip displacement with respect to the element bending stiffness EI_{yy} 69
6.3 Sensitivities and relative errors of the sensitivities of the tip rotation with respect to the torsional stiffness GJ 71
6.4 Sensitivities and relative errors of the sensitivities of the tip rotation with respect to the shear center location SC_x 71

7.1 Structural optimization framework 75
7.2 Convergence of the objective function 77
7.3 Change in the internal structural configuration resultant from optimization 78
7.4 Internal structural configurations before and after the fully constrained optimization problem 80
7.5 Convergence history of the objective function and constraints 81

Nomenclature

Greek symbols

α	Angle of attack.
$\boldsymbol{\epsilon}$	Strain vector.
$\boldsymbol{\sigma}$	Stress vector.
δ	Kronecker delta.
δ_{tip}	Vertical tip displacement.
Γ	Dihedral angle.
Λ	Sweep angle.
λ	Lagrangian multiplier for equality constraints. Taper ratio.
μ	Lagrangian multiplier for inequality constraints.
ϕ	Shape function.
ψ	Adjoint vector.
ρ	Density.
ρ_l	Linear density.
σ	Normal stress.
τ	Shear stress.
θ	Twist angle.
θ_{tip}	Tip rotation.
ε	Normalized error.

Roman symbols

$\mathbf{c}$	Cross-section information vector.
$\mathbf{d}$	Internal structure design vector.
$\mathbf{f}$	Force vector.
$\mathbf{v}$	Final design vector.
$\mathbf{x}$	Design vector. Average finite element properties vector.

A	Cross-sectional area.
a	Wingbox section's length.
B	Strain-displacement matrix.
b	Cross-section's width.
C	Stiffness tensor.
c	Airfoil's chord.
D	Elastic stiffness matrix.
E	Young's modulus.
EA	Axial stiffness.
EI	Bending stiffness.
f	Interest function.
f_{obj}	Objective function.
G	Shear modulus.
g	Inequality constraint.
GJ	Torsional stiffness.
h	Equality constraint. Cross-section's height.
J	Jacobian matrix.
K	Stiffness matrix.
L	Beam's length.
l	Number of equality constraints.
M	Bending moment.
m	Number of equality constraints.
N	Shape function matrix.
n	Number of design variables.
p	Pressure.
q	Vertical distributed load. Shear flow.
Q_i	Generalized applied forces.
R	Residual function.

SC	Shear center location.
T	Transformation matrix.
t	Thickness.
u	Displacement.
V	Potential energy.
v	Weight function.
w	Vertical displacement.
x	Design variable.
x_1/c	Front spar's relative location.
x_2/c	Rear spar's relative location.
y	State variable.

Subscripts

∞	Free-stream condition.
A	Refers to the front spar.
av	Average value.
B	Refers to the upper skin.
C	Refers to the rear spar.
c	Cross-sectional value.
D	Refers to the lower skin.
max	Maximum value.
n	Normal component.
s	Tangential component.
x, y, z	Cartesian components.

Superscripts

*	Optimal design point.
B	Lower bound.
T	Transpose.
U	Upper bound.

Glossary

ADiMat Automatic Differentiation for Matlab is a software for the computation of precise and efficient derivatives of Matlab programs.

AD Automatic Differentiation is a method to numerically evaluate the derivative of a function specified by a computer program by applying the chain rule repeatedly.

AM Adjoint Method is a numerical method for efficiently compute the gradient of a function in a numerical optimization problem.

CFD Computational Fluid Dynamics is a branch of fluid mechanics that uses numerical methods and algorithms to solve problems that involve fluid flows.

CSD Computational Structural Mechanics is a branch of structure mechanics that uses numerical methods and algorithms to perform the analysis of structures and its components.

FAR Federal Aviation Regulations are rules prescribed by the Federal Aviation Administration governing all aviation activities in the United States.

FD Finite Differences are approximation methods to estimate the derivatives of a function.

FEM Finite Element Method is a numerical approach for finding approximate solutions to boundary value problems for partial differential equations.

KKT Karush–Kuhn–Tucker conditions are first order necessary conditions for a solution to be optimal in nonlinear programming.

MDO Multi-Disciplinary Optimization is an engineering technique that uses optimization methods to solve design problems incorporating two or more disciplines.

NACA National Advisory Committee for Aeronautics, under which series of airfoil shapes are developed.

NLP Nonlinear Programming is the process of obtaining the solution of an optimization problem defined by a system of equalities and inequalities constraints, over a set of real variables, along with an objective function to be minimized, where some of the constraints or the objective function are nonlinear.

RANS Reynolds-Averaged Navier-Stokes are time-averaged equations of motion for fluid flow.

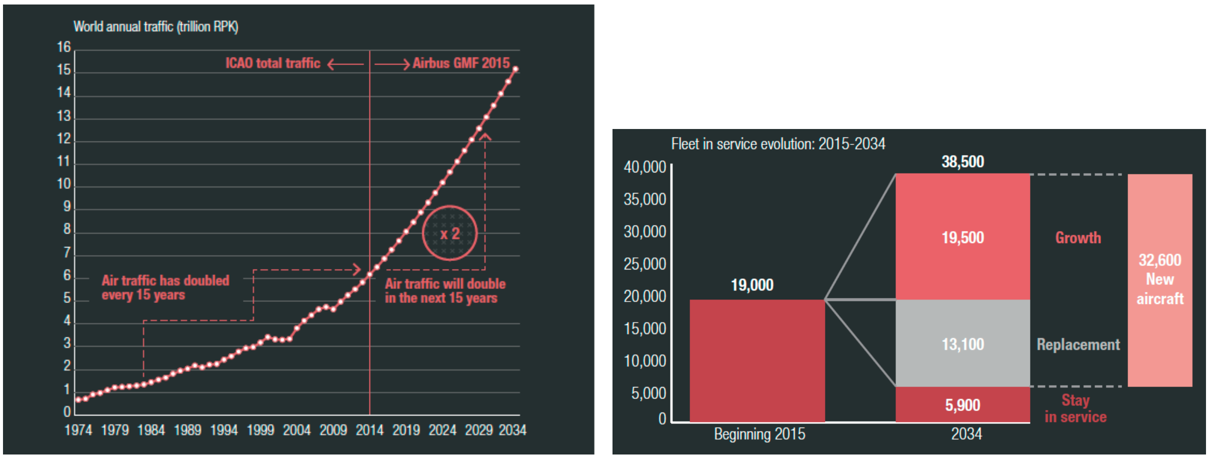
RPK Revenue Passenger Kilometers measures the traffic for an airline flight and is calculated by multiplying the number of revenue-paying passengers aboard the vehicle by the distance traveled.

Chapter 1

Introduction

1.1 Motivation

Slightly more than one hundred years ago, the Wright brothers were able to successfully build and fly the first sustained, controlled, powered and heavier-than-air manned aircraft. Over the past century, a remarkable progress in the history of aviation was observed and many different configurations of aircraft were developed. According to Airbus Global Market Forecast [1], the world's annual air traffic in 2014 was approximately 6.1 trillion RPK (revenue passenger kilometers). There are currently 19000 transport aircraft in service and, in the next 15 years, the air traffic is estimated to double (Figure 1.1a), demanding 32600 new passenger and freighter aircraft (Figure 1.1b).



(a) World annual traffic (in trillion RPK)

(b) Fleet in service evolution from 2015 to 2034

Figure 1.1: Airbus global market forecast data [1].

With a growing trend of the aviation market, new aircraft need to be introduced, keeping in mind the concept of sustainable development and an awareness on limited resources. Therefore, more efficient technology is highly desirable. In order to improve global aircraft design, several disciplines (such as structures, aerodynamics, avionics, propulsion, materials, among others) must be simultaneously taken into account as a coupled Multidisciplinary Design Optimization (MDO) problem. If each individual discipline was responsible for designing an aircraft, incompatible concepts would be presented, as illustrated in Figure 1.2.

The physical problems of each discipline should be translated into mathematical models, depending on a set of parameters that characterize a certain design. By varying these parameters, different concepts are generated and the main goal is to establish a compromise among the disciplines to generate

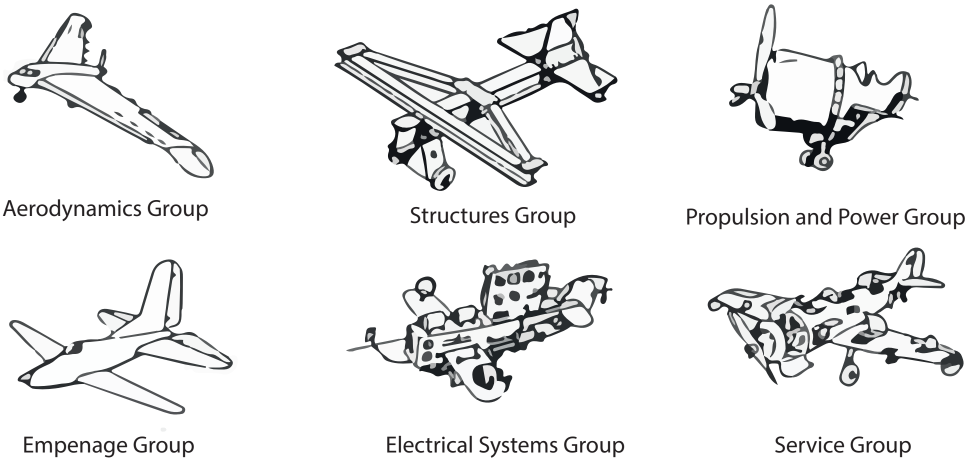


Figure 1.2: Aircraft concepts according to different disciplines (adapted from [2]).

better concepts. This is achieved by introducing optimization tools in multidisciplinary design frameworks. According to Sobieszczanski-Sobieski and Haftka [3], there has been a rising interest in MDO applications in aerospace systems, but many challenges still exist concerning computational expense and organizational complexity. Thus, more efficient optimization tools are still necessary to be developed.

One of the possible solutions to accelerate optimization processes is to use gradient-based algorithms with highly efficient sensitivity computation methods [4], in order to lead the optimizer to an optimum design point, fast and accurately. The adjoint method is a good example of an efficient approach for gradient calculation, being the computational time practically independent on the number of design variables [5]. This method can be employed in many different disciplines as testified in the following examples:

- Akgün et al. [6] applied the adjoint method to different aerospace structures subjected to multiple load cases, using Von Mises stress, local buckling and displacement values as constraints, exploring a constraint lumping strategy, named the Kreisselmeier–Steinhauser functional, to enhance the adjoint method efficiency;

- Jensen et al. [7] studied the application of the adjoint method to linear elastodynamic structural problems, introducing some alterations to the traditional approach, in order to correct the inconsistent sensitivities in transient responses, leading to more reliable solutions;

- Kennedy and Hansen [8] presented an application of an adaptation of the adjoint method in a composite manufacturing process optimization framework. The main objective was to improve the failure load of thick, laminated, thermosetting composite members, by imposing constraints in the autoclave temperature in order to ensure that the composite was uniformly cured;

- Luo et al. [9] demonstrated the advantages, in terms of efficiency, of applying the adjoint method in several aerodynamic problems including the drag reduction of airfoils, wings and wing-body configurations, as well as in the improvement of the aerodynamic performance in turbine and compressor blade rows;

- Martins et al. [10] applied the adjoint method to calculate the sensitivities in a coupled high-fidelity aero-structural problem. The aero-structural adjoint formulation was proved to be efficient and accurate, specially in cases where the number of design variables is much larger than the number of interest functions. This was illustrated with two drag minimization problems in a conventional swept-wing with 190 shape design variables each.

1.2 Aeroelastic Tool Background

Almeida [11] developed an aeroelastic tool to study the behavior of aircraft wings. The structure of the wing is modeled with 3D equivalent beam finite elements along its correspondent elastic axis and utilizes a panel method code to solve the fluid dynamics equations. In order to couple both fluid and structural problems, staggered (or loosely-coupled) algorithms were implemented, including both volume-continuous and volume-discontinuous methods.

The aeroelastic calculation procedure is represented in Figure 1.3. Firstly, the Computational Fluid Dynamics (CFD) and Computational Structural Dynamics (CSD) meshes are generated based on the input information, followed by a steady-state CFD computation to calculate the initial aerodynamic loads, which will lead to an initial structural response. Afterwards, a time step is applied, the aerodynamic forces are recalculated for the deformed configuration and are transfered to the structural mesh. The new deformed configuration is then calculated and the cycle is repeated until convergence is attained.

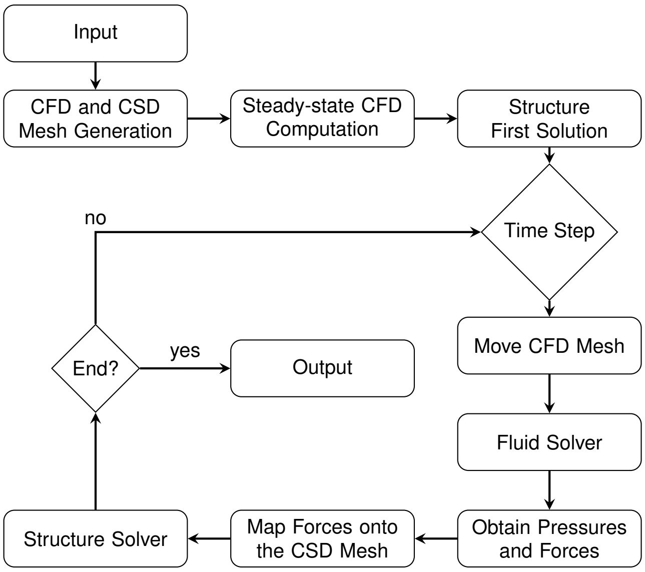
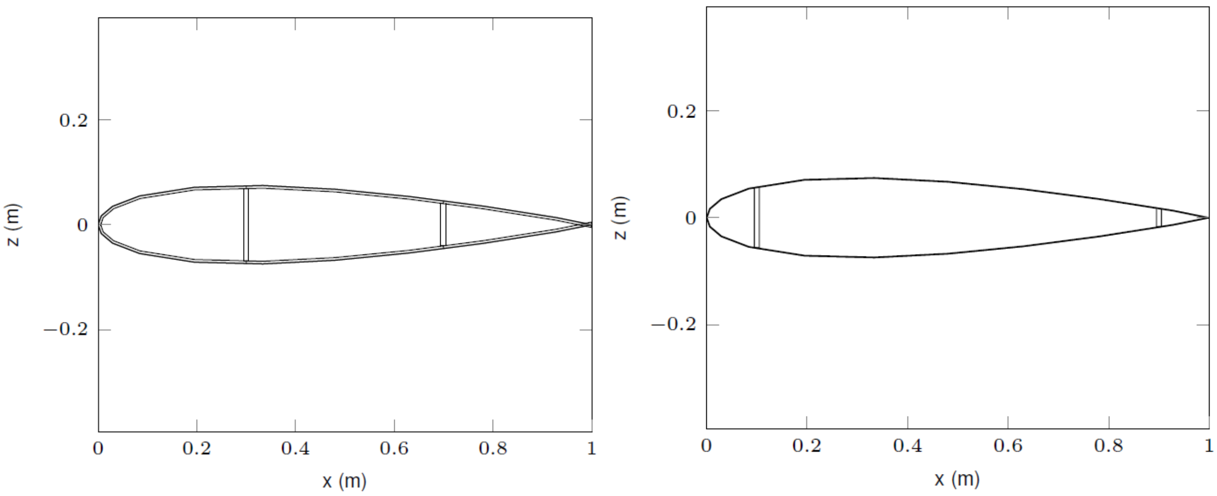

Input
CFD and CSD Mesh Generation
Steady-state CFD Computation
Structure First Solution
no
Time Step
yes
End?
Output
Move CFD Mesh
Fluid Solver
Structure Solver
Map Forces onto the CSD Mesh
Obtain Pressures and Forces

Figure 1.3: Flowchart illustrating the program‘s calculation process during aeroelastic analysis [11].

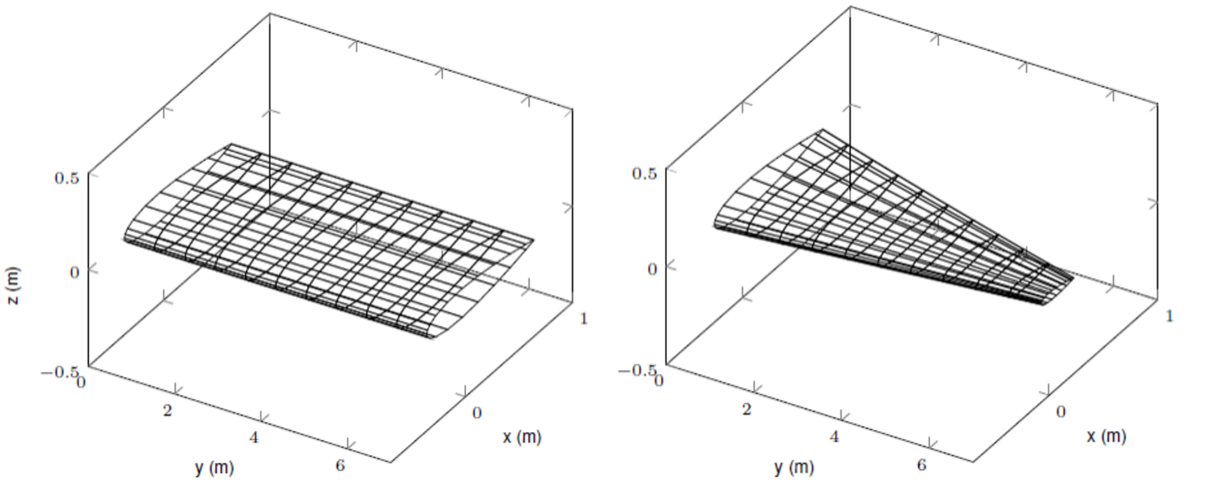
Besides the aeroelastic capabilities of the developed framework, static aero-structural analysis may also be performed. Almeida [11] used this capability to perform a gradient-based optimization to minimize the wing's total mass while fixing the lift coefficient, with maximum stress and tip deflection constraints. Taper ratio, sweep angle, dihedral angle, twist angle at the wing tip, angle of attack, relative spars location, and spar and skin thicknesses were used as design variables. The initial and final internal and external configurations are represented in Figures 1.4 and 1.5, respectively. Even though an optimal solution was obtained, the optimization was proven to be quite inefficient. Since the forward finite difference method was used to estimate the gradient information, the entire aero-structural code had to be evaluated ten times per iteration in the optimization process: one for the considered design point plus nine times for the design perturbations (one per design variable). Therefore, strategies to make this process more efficient are required to be investigated.



(a) Before optimization.

(b) After optimization.

Figure 1.4: Evolution of the internal wing structure during the optimization process [11].



(a) Before optimization.

(b) After optimization.

Figure 1.5: Evolution of the external wing structure during the optimization process [11].

1.3 Objectives

The main objective of this thesis is to develop an efficient optimization tool for the structural problem in the aero-structural framework presented in Section 1.2. The optimization algorithm shall rely on gradient-based information to find the best solution.

A wing baseline configuration shall be defined and a parametric analysis must be performed in order to check how the input variables impact the output functions of interest. Subsequently, a subset of parameters shall be selected to be used in the optimization problem.

An adjoint structural solver must be implemented and validated for the present case and other alternative sensitivity analysis methods must be studied and compared.

After developing the sensitivity analysis module to compute the total gradient information, some optimization cases must be tested to prove the efficiency of the developed structural optimization framework.

In Figure 1.6 it is represented, in solid boxes, the implementation steps of the aeroelastic framework developed by Almeida [11]. The dashed boxes correspond to the proposed additional modules to be introduced with this work. The adjoint structural solver shall compute an adjoint vector to be used in the determination of the interest functions' sensitivities with respect to the structural properties. Subsequently, the former sensitivities must be multiplied by the gradient of the structural properties with respect to the structural shape parameters, in order to obtain the total sensitivity to be used by the optimizer.

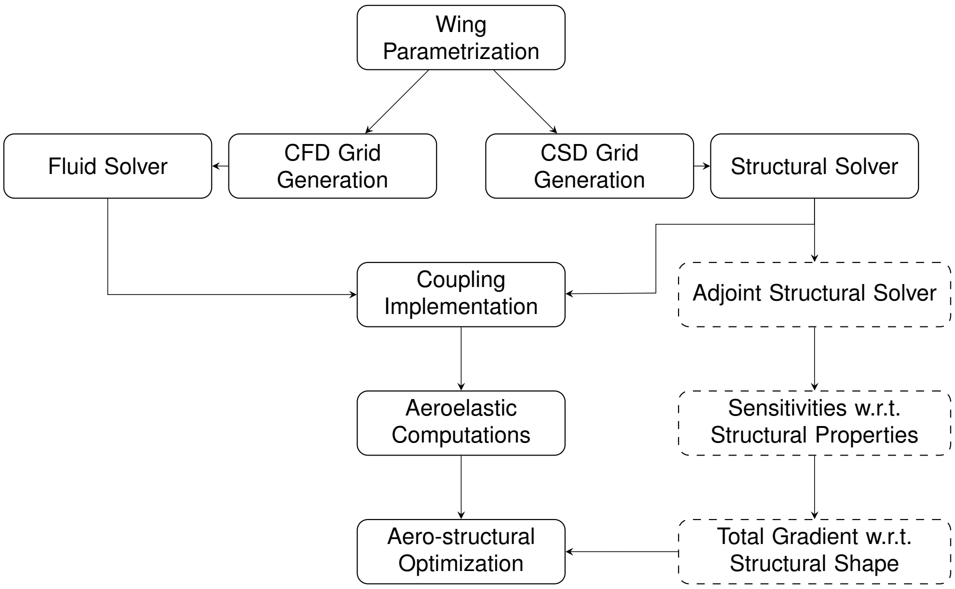


Figure 1.6: Flowchart illustrating the implementation structure of an aeroelastic framework. Solid boxes correspond to work previously done. Dashed boxes correspond to the new implemented features.

1.4 Thesis Outline

The structure of the thesis is organized as follows:

Chapter 2 provides an overview of the main concepts in optimization problems, with a formal mathematical definition. A general classification into gradient-based and gradient-free optimization problems is presented and the respective advantages and disadvantages are addressed.

Chapter 3 presents the definition of sensitivity analysis and explores the different methods to obtain gradient information, with practical examples in structural problems.

Chapter 4 describes the framework modules to assess the wing structural behavior. Adaptations in the original framework are performed to make it suitable for an exact gradient estimation in a later phase.

Chapter 5 covers a parametric study of the internal structural parameters and material properties, by varying some baseline parameters and evaluating the influence in the output structural responses.

Chapter 6 characterizes the sensitivity framework that computes the exact first-order derivative information to be used in the gradient-based optimization problems.

Chapter 7 consolidates the previously developed modules and links the framework to the optimizer. Two optimization cases are presented as examples of application.

Chapter 8 summarizes the main achievements with this thesis and provides recommendations for future work.

Chapter 2

Optimization Methods

This chapter provides an introduction to optimization problems. In Section 2.1, the main concepts in optimization will be addressed, including the classification of several types of optimization problems. Definitions of objective function, design variables and constraints will be provided. In Sections 2.2 and 2.3, an overview of gradient-based and gradient-free methods is presented, respectively, highlighting the main advantages and disadvantages of each method, leading to the selection of the best approach in Section 2.4.

2.1 Optimization Background

According to Rao [12], optimization can be defined as the act of obtaining the best result under given circumstances, and is a field that has been growing for the past few decades. This area has a direct application in engineering problems. An engineering system may be characterized by a group of quantities which may be regarded as variables. These variables, in the design process, are called design or decision variables and a set of design variables corresponds to a design vector. Normally, there are restrictions that must be satisfied to produce an acceptable design, which are called design constraints. An objective or cost function corresponds to a criterion selected for evaluating different designs expressed as a function of the design variables.

A proper classification of an optimization problem is essential to choose the best algorithm to solve the problem. According to Hicken et al. [13] , an optimization problem can be characterized based on the following different aspects:

Linearity: a problem may be considered linear programming if the objective function and constraints are linear, quadratic programming if the objective function is quadratic and constraints are linear, or nonlinear programming if the objective function or constraints are nonlinear;

Convexity: a problem is said to be convex if convex functions are minimized over convex sets. Otherwise it is called non-convex;

Design Variables: a problem may consider a single variable or multivariables, and these variables may be discrete or continuous;

Constraints: if there are conditions to be satisfied when optimization is performed, a problem is said to be constrained. If no limitations are imposed, it is said to be unconstrained;

Data: an optimization problem may be characterized based on whether it treats deterministic data or stochastic data;

Time: depends on whether the problem is static or dynamic.

Despite the broad classification of optimization problems previously presented, all the optimization methods can be divided in either gradient-based or gradient-free methods, depending on, whether or not, information on the derivatives of the objective and constraint functions is required, as it will be presented in the following subsections.

2.2 Gradient Based Methods

Gradient-based methods are applied in design problems which have smooth objective functions and constraints, using gradient (or sensitivity) information to find an optimal solution. According to Marta [14], most of the gradient-based algorithms comprise two subproblems:

1. Establishing the search direction.
2. Minimizing along that direction (or one-dimensional line search).

Zingg et al. [15] denote that the main advantages of gradient-based methods are the rapid convergence to the optimum solution by exploiting gradient information and the fact that gradient-based methods provide a clear convergence criterion, assuring that the step size is under a certain order of magnitude and allowing local optimum identification.

One of the disadvantages when using gradient based methods is that they tend to fail when considering noisy or discontinuous objective functions, or categorical design variables. Another limitation is that these methods find a local optimum instead of guaranteeing a global optimum. Furthermore, the optimization solution may be influenced by the initial design point choice, since different starting points may lead to different search directions, due to the gradient information at the considered different points.

Different optimality conditions related to the function's derivatives can be found, depending on whether the problem is unconstrained or constrained and on the order of the derivative. Therefore, a presentation of both unconstrained and constrained methods is addressed in the following subsections.

2.2.1 Unconstrained Optimization

An unconstrained optimization problem is formally defined as

$$\begin{array}{ll} \text{minimize} & f_{obj}(\mathbf{x}) \\ \text{w.r.t} & \mathbf{x} \in \mathbb{R}^n \end{array} , \tag{2.1}$$

where $f_{obj} : \mathbb{R}^n \to \mathbb{R}$ is an objective function and $\mathbf{x}$ is the design vector.

Considering that f_{obj} is twice continuously differentiable in $\mathbb{R}^n$, $\nabla f_{obj}(\mathbf{x})$ is the gradient of f_{obj}, and $\nabla^2 f_{obj}(\mathbf{x})$ is the Hessian of f_{obj}, it is possible to define first and second order optimality conditions. The first order optimality condition relies only on the gradient information and it is given by

$$||\nabla f_{obj}(\mathbf{x}^*)|| = 0 \; . \tag{2.2}$$

By satisfying Equation (2.2), it is guaranteed that the design point $\mathbf{x}^*$ is a stationary point and, consequently, it can be a maximum, a minimum or a saddle point.

The second-order optimality conditions rely not only on the gradient, but also on the Hessian matrix information of f_{obj}. The necessary second order conditions for optimality are given by

$$||\nabla f_{obj}(\mathbf{x}^*)|| = 0 \qquad \text{and} \qquad \nabla^2 f_{obj}(\mathbf{x}^*) \text{ is positive semi-definite.} \tag{2.3}$$

However, to guarantee that x^* is a minimum, the sufficient conditions for optimality must be verified:

$$||\nabla f_{obj}(\mathbf{x}^*)|| = 0 \qquad \text{and} \qquad \nabla^2 f_{obj}(\mathbf{x}^*) \text{ is positive definite.} \tag{2.4}$$

Several unconstrained gradient-based minimization methods can be found in literature, some based on the first order optimality conditions and others using second order optimality conditions. One of the first methods was developed by Cauchy and is called the Steepest Descent Method [16]. This method has a linear convergence rate and uses the gradient information of the objective function ∇f_{obj} to evaluate the search direction at each iteration and to establish one of the convergence criteria, as described in Algorithm 1.

Algorithm 1: Steepest Descent Method Algorithm

```
Input  : function ($f_{obj}$), initial point ($\mathbf{x}_0$), gradient convergence criteria ($\varepsilon_g$), absolute convergence
         criteria ($\varepsilon_a$) and relative convergence criteria ($\varepsilon_r$)
Output: Local minimum of $f_{obj}$ ($\mathbf{x}^*$)
begin
    repeat
        Calculate $\nabla f_{obj}$
        if $||\nabla f_{obj}|| \leq \varepsilon_g$ then
            converged
        else
            compute normalized search direction $p_k = -\nabla f_{obj}/||\nabla f_{obj}||$
        end
        Line search to find step length $\alpha_k$ in $p_k$ direction
        Update current point $x_{k+1} = x_k + \alpha_k p_k$
        Evaluate $f_{obj}(x_{k+1})$
        if $|f_{obj}(x_{k+1}) - f_{obj}(x_k)| \leq \varepsilon_a + \varepsilon_r |f_{obj}(x_k)|$ then
            converged
        else
            set $k = k + 1$, $x_k = x_{k+1}$
        end
    until converged;
    stop
end
```

Other unconstrained optimization algorithms can be found such as: the Conjugate Gradient Method [17], which is similar to the steepest descent method, but takes into account the gradient history in the definition of the search direction; Newton's Method [18], which is a quadratic convergence rate method, that uses first and second order derivative information to find the optimum; Modified Newton's Methods [19], which introduce an alteration to Newton Method by ensuring positive definiteness of the Hessian matrix and defining a step size parameter to improve solution in highly nonlinear functions; Quasi-Newton Methods [20], which only use gradient information and build second order information based on an approximate Hessian matrix derived from the function and gradient history, by forcing it to be symmetric and positive definite and Trust Region Methods [21] which result from minimizing a function within certain regions where the quadratic model represents a good approximation of the real function.

2.2.2 Constrained Optimization

According to Belegundu and Chandrupatla [22], most engineering optimization problems can be mathematically expressed as nonlinear programming (NLP) problems,

$$\begin{array}{lll} \text{minimize} & f_{obj}(\mathbf{x}) & \\ \text{w.r.t} & \mathbf{x} \in \mathbb{R}^n & \\ \text{subject to} & g_j(\mathbf{x}) \leq 0\ , & j = 1, ..., \ell \\ \text{and} & h_k(\mathbf{x}) = 0\ , & k = 1, ..., m \\ \text{and} & \mathbf{x}^L \leq \mathbf{x} \leq \mathbf{x}^U, & \end{array} \tag{2.5}$$

where $f_{obj}(\mathbf{x})$ is the objective function, with $\mathbf{x} = (x_1, x_2, ..., x_n)^T$ being the design vector of n design variables, which is limited by lower and upper bound vectors $\mathbf{x}^L$ and $\mathbf{x}^U$, respectively. The optimization problem is subjected to ℓ inequality constraints, $g_j(\mathbf{x})$, and to m equality constraints, $h_k(\mathbf{x})$. The term feasible design is used to describe a set of design variables that satisfies the constraints.

Optimality conditions can also be derived for the constrained optimization problem. However, these conditions are not as straightforward as in Equation (2.2), for the unconstrained case. It is first necessary to define the Lagrangian function ($\mathcal{L}$) as

$$\mathcal{L}(\mathbf{x}, \lambda) = f(\mathbf{x}) + \sum_{j=1}^{l} \mu_j g_j(\mathbf{x}) + \sum_{k=1}^{m} \lambda_k h_k(\mathbf{x})\ , \tag{2.6}$$

where λ_k and $\mu_j \geq 0$ are Lagrange multipliers defined for each constraint $h_k = 0$ and $g_j \leq 0$, respectively. It is assumed that f_{obj}, h_k and g_j are continuously differentiable over $\mathbb{R}^n$.

If $\mathbf{x}^*$ is an optimal solution of the optimization problem, then there exist Lagrange multipliers μ^* and λ^* that satisfy the conditions:

$$\nabla_x \mathcal{L} = 0 \Rightarrow \frac{\partial \mathcal{L}}{\partial \mathbf{x}} = \frac{\partial f}{\partial \mathbf{x}} + \sum_{j=1}^{l} \mu_j \frac{\partial g_j}{\partial \mathbf{x}} + \sum_{k=1}^{m} \lambda_k \frac{\partial h_k}{\partial \mathbf{x}} = 0 \tag{2.7a}$$

$$g_j \leq 0, \quad j = 1, ..., \ell \tag{2.7b}$$

$$h_k = 0, \quad k = 1, ..., m \tag{2.7c}$$

$$\mu_j \geq 0, \quad j = 1, ..., \ell \tag{2.7d}$$

$$\mu_j g_j = 0, \quad j = 1, ..., \ell\ . \tag{2.7e}$$

The Equations (2.7) are referred as the Karush-Kuhn-Tucker (KKT) optimality conditions, corresponding to the necessary conditions for nonlinearly constrained problems.

If a sufficient condition is requested for the constrained optimization case, then the following additional condition must be added to Equations (2.7a) to (2.7e):

$$\nabla^2\mathcal{L}(\mathbf{x}^*) = \nabla^2 f_{obj}(\mathbf{x}^*) + \sum_{j=1}^{l} \mu_j \nabla^2 g_j(\mathbf{x}^*) + \sum_{k=1}^{m} \lambda_k \nabla^2 h_k(\mathbf{x}^*) \text{ is positive definite.} \tag{2.8}$$

The positive definiteness $\nabla^2\mathcal{L}(\mathbf{x}^*)$ must be verified only in the feasible subspace (M), which is defined by the tangent plane to all active constraints,

$$M = \{\mathbf{y} : \nabla h_k^T(\mathbf{x}^*)\mathbf{y} = 0, j = 1, ..., \ell \text{ and} \tag{2.9}$$

$$\nabla g_j^T(\mathbf{x}^*)\mathbf{y} = 0 \text{ for all } j \text{ for which } g_j(\mathbf{x}^*) = 0 \text{ with } \mu_j > 0\}. \tag{2.10}$$

where $\mathbf{y}$ corresponds to any vector that satisfies the previous conditions.

Some constrained gradient-based methods can be found in literature, such as: the Gradient Projection Method [22], which finds the search direction by projecting the steepest descent direction onto the subspace tangent to the active constraints; the Feasible Directions Method [23] which determines a direction that is both feasible and descent, instead of following constraint boundaries (as in the projection method). Then, a line search along that direction will yield an improved design by simultaneously trying to keep away from the constraint boundaries and reducing the objective function as much as possible; The Sequential Quadratic Programming (SQP) [24], corresponding in a way to the application of Newton's method to the KKT conditions, which finds an approximate solution of a succession of quadratic programming subproblems in which a quadratic model of the objective function subject to linearized constraints is minimized.

2.3 Gradient-Free Methods

Gradient-free methods, or zeroth order methods, are optimization methods that do not require derivative information to find a solution and are based solely on the value of the objective function.

A great advantage in gradient-free methods is that they do not require a strong set of assumptions or global properties on the optimization problem. The function's evaluation is usually treated as a "black box" and there is no need to calculate sensitivity information. Another advantage, when compared to gradient-based methods, is that several algorithms are able to solve discrete optimization problems and these methods usually tolerate noise in the objective function. Furthermore, gradient-free methods are able to find a global optimum, instead of a local optimum.

The main disadvantage in gradient-based methods is that they are quite sensitive to the dimension of the problems, increasing the computational cost due to the need of evaluating the interest functions multiple times. Besides that, it may also be difficult to establish an ending criterion.

An example of gradient-free algorithms are the so-called "population based" algorithms, which engage a set of solutions (instead of a single solution) that are updated at every iteration, simulating the evolution or behavior of a population. Some examples of population based methods are: Genetic Algorithms, Particle Swarm Optimization and Ant Colony Methods.

Genetic Algorithms [25] are heuristic methods inspired in the natural evolution of species. In these methods, a gene corresponds to a design variable, an individual represents a design point, and a population corresponds to a group of individuals. Firstly, a population is initialized, covering the design space. Afterwards, a selection of the best individuals is performed (by evaluating the objective function) and these are subjected to crossover, which correspond to a combination of genes in order to originate the next generation. Some randomness is included through mutation, in order to maintain diversity in the population. Finally, the new generation is introduced in the population and this process is carried on until a predefined stopping criterion is met, which can be, for instance, a certain number of generations.

Particle Swarm Optimization [26] corresponds to another example of a population based method. In this method, a population of particles is randomly distributed in a search space with random individual initial positions and velocities. The objective function is evaluated for each particle's position and then, the movement of the particles is updated by both their self experience (cognitivism) and social interaction with other particles (sociocognition). It is expected that the particles will move towards the best solutions. Different stopping criteria, presented by Zielinski et al. [27], can be applied to this method such as the "improvement-based criteria", which means that the optimization run should stop if only a certain level of improvements are accomplished over time, or "distribution-based criteria", which are related to the closeness among individuals, since, usually, a group of individuals tend to converge to an optimum solution.

Ant Colony Algorithms developed by Dorigo and Stützle [28], are inspired in the ant colony behavior of looking for the highest amount of food (correspondent to the objective function) within a certain area (associated with a discrete set). At first, ants wander randomly looking for food, and once they find it, they leave a trail of pheromones on the way back to the nest, with an intensity depending on the quantity and quality of food. Thus, the more favorable paths are strongly marked with pheromones, leading the other ants towards their objective, and optimizing the solution. In the meantime, some ants are still randomly search for closer food sources. Once the food source is insufficient, the route is no longer populated with pheromones and slowly decays. In optimization processes, the pheromone trails are modeled with adaptive memory.

2.4 Choice of the Method

Based on the overview of gradient-based and gradient-free methods presented in the previous sections, it is necessary to justify the selected approach to be applied in this work. Since a highly efficient optimization is required, a fast convergence method with clear convergence criteria appears to be the best option. Furthermore, the structural model to be used in the optimization is well established and the parameters to be treated as design variables are considered to be continuous. Methods that are dependent on the dimension of the problem shall be avoided since, if a high number of design variables is chosen, the optimization process will require a high number of function evaluations to determine an optimum design, being computationally expensive. For all the reasons stated above, gradient-based methods will be used in this thesis. Note that in theory, these methods do not depend directly on the dimension of the problem. However, the method used to compute the gradient information may or not depend on the number of design variables, being one of the topics of discussion in the next chapter.

Chapter 3

Sensitivity Analysis

According to Choi and Kim [29], design sensitivity analysis consists in the computation of the dependence of an interest function (f) with respect to the design variables (x). Interest (or performance) functions can either be objective functions or constraints, and the dependence on the design variables can either be explicit or implicit. The results of a sensitivity analysis provide the necessary quantitative information to guide on which direction to go in order to improve a certain design. This information can be visually displayed so that engineers can carry out ”what-if” studies to make trade-off determinations. In the case of automated processes, the sensitivity information is provided to optimizer with the purpose of leading to a better design configuration. The computation of sensitivities is considered one of the most computational expensive steps in the gradient-based optimization process.

Sensitivity analysis methods can be divided in approximation and analytic methods. The former include finite difference and complex step methods which yield approximate result of the interest function's sensitivity. The latter exploits differential calculus concepts, such as symbolic differentiation and the chain rule, to obtain the true sensitivity values, being only affected by computational errors, when implemented.

In the following subsections, these different sensitivity analysis methods will be presented and a discussion on accuracy, computational efficiency and implementation complexity will be addressed.

3.1 Symbolic Differentiation

The first analytical method to evaluate the sensitivity of a function is the symbolic differentiation. A function f is said to be differentiable with respect to x, if the following limit exists:

$$\frac{\mathrm{d}f}{\mathrm{d}x} \equiv \lim_{\Delta x \to 0} \frac{f(x+\Delta x) - f(x)}{\Delta x} \quad . \tag{3.1}$$

The limit represented in Equation (3.1) is known for several functions, allowing an explicit derivation of the function's sensitivity. Nevertheless, it is restricted to explicit functions and may become computationally expensive or even impracticable to calculate in high dimensionality problems.

Both free and commercial computational tools that can handle symbolic mathematics exist. For instance, *SymPy* is a free library written in Python™ language for symbolic mathematics. An example of a commercial tool is the *Symbolic Math Toolbox* from MathWorks®, that includes the MuPAD™ language with a Notebook interface to interactively display the symbolic differentiation results.

3.2 Finite Difference Method

The finite difference method allows sensitivity estimation by approximating the derivative of a function with a quotient of differences. Considering an interest function f, a design variable x, and a small perturbation in the design Δx, the forward and backward finite difference expressions are, respectively, given by:

$$\frac{\mathrm{d}f}{\mathrm{d}x} = \frac{f(x+\Delta x) - f(x)}{\Delta x} + \mathcal{O}(\Delta x) \tag{3.2a}$$

and

$$\frac{\mathrm{d}f}{\mathrm{d}x} = \frac{f(x) - f(x-\Delta x)}{\Delta x} + \mathcal{O}(\Delta x)\ . \tag{3.2b}$$

These two expressions are first-order estimations, since the order of the truncation error is $\mathcal{O}(\Delta x)$. If a second-order error estimation is desired, one can use the central-difference formula:

$$\frac{\mathrm{d}f}{\mathrm{d}x} = \frac{f(x+\Delta x) - f(x-\Delta x)}{2\,\Delta x} + \mathcal{O}[(\Delta x)^2] \tag{3.3}$$

Equations (3.2a), (3.2b) and (3.3) can be obtained from a Taylor series expansion and its derivation can be found in the work of Chaudhry [30].

The main advantages in finite difference methods are the simplicity in the implementation process and the fact that the program that computes f can be treated as a ”black box”, since only the input and output values are required to be known.

Considering the multivariable case, where x becomes a n-dimension vector, the cost of calculating the sensitivity of f is proportional to the number of design variables n: in the forward and backward difference methods, the function f needs to be evaluated $n+1$ times, and in the central difference method it requires $2n$ evaluations. This is the main disadvantage since it becomes computationally too expensive (and sometimes even unfeasible) in large-scale problems involving many design variables. Furthermore, there is the need to select a small step-size to minimize truncation error, but at the same time guaranteeing that subtractive cancellation errors do not become dominant, making the task of choosing a design perturbation step-size a challenge.

3.3 Complex Step Method

The complex step method uses complex variables to calculate the sensitivity of a function and arises as a solution to overcome the problem of subtractive cancellation in the finite difference approximation methods, leading to accurate and robust results. This method is based on Lyness's [31] work on using complex variables to produce differentiation formulas. Squire and Trapp [32] presented a simplified version for the first derivative approximation and illustrated this method with numerical examples, comparing it with the central difference method.

Being f an analytic function, one can consider the following Taylor series expansion with a pure imaginary step perturbation $i\Delta x$:

$$f(x+i\Delta x) = f(x) + i\Delta x\, f'(x) - (\Delta x)^2 \frac{f''(x)}{2!} - i(\Delta x)^3 \frac{f'''(x)}{3!} + \ldots\,, \tag{3.4}$$

where f' represents the first order derivative of f, f'' the second order derivative of f, and so on. Evaluating the imaginary part on both sides of the Equation (3.4) and reorganizing it, yields

$$f'(x) = \frac{\mathrm{Im}[f(x+i\Delta x)]}{\Delta x} + (\Delta x)^2 \frac{f'''(x)}{3!} + \ldots\,. \tag{3.5}$$

Therefore, the sensitivity equation with a second-order error estimation is given by

$$\frac{\mathrm{d}f(x)}{\mathrm{d}x} = \frac{\mathrm{Im}[f(x+i\Delta x)]}{\Delta x} + \mathcal{O}[(\Delta x)^2]\,. \tag{3.6}$$

Note that, with this approach, subtractive cancellation errors do not occur for the first order derivative estimation due to the non existence of difference operations in Equation (3.6).

As in the finite difference case, the cost of estimating a sensitivity is directly proportional to the number of design variables n, becoming a disadvantage in using this method for problems with a great number of design variables. Note that complex arithmetic also increases the computational cost, but may pay-off when compared to finding an acceptable step-size on a finite difference method. Nevertheless, it is a trustworthy method to benchmark results. However, if a similar implementation is performed to obtain second order derivatives (that may be useful to estimate an Hessian matrix, for instance), roundoff errors occur. Thus, a small step-size cannot be arbitrarily chosen, as in for the first derivative approximation case in the finite difference methods, and a different approach should be employed. This goes beyond the scope of this thesis and the work of Lai and Crassidis [33] should be consulted for further details.

In order to implement the complex step method, it may be necessary to redefine some arithmetic functions and logical operators that are not defined for complex arguments in some programming languages, so that the evaluation of f with complex arguments is possible. For instance, in MATLAB® the absolute (`abs`) function needs to be redefined since its complex definition is non-analytic. `Max` and `min` functions should also be adapted because they use relation logic operators, which are normally not defined for complex arguments. A solution to deal with logic relation operations is to redefine the operators to compare only real parts of the complex number. The function `admDiffComplex` from *ADiMat*, a tool to differentiate MATLAB® code that will be presented in more detail in subsequent sections, allows the use of the complex step method.

3.3.1 Finite Difference and Complex Methods Comparison

Martins [34] performed a comparison among complex-step derivative, forward finite difference and central finite difference methods. The considered function was

$$f(x) = \frac{\exp(x)}{\sqrt{\sin^3(x) + \cos^3(x)}}\,. \tag{3.7}$$

The purpose of this study was to compare the two previous methods when calculating the sensitivity of f with respect to x, as far as the rate of convergence, subtractive cancellation and accuracy are concerned. The normalized error of the sensitivity $\varepsilon(f')$ is defined as

$$\varepsilon(f') = \frac{|f' - f'_{ref}|}{|f'_{ref}|} \, , \tag{3.8}$$

where f' is the sensitivity evaluated with the different approximation methods, and f'_{ref} is the exact derivative value at a certain point. The normalized error (ε) is represented in Figure 3.1, in a logarithmic plot, as a function of the step-size (h) for $x = 1.5$. Forward (Equation (3.2a)) and central (Equation (3.3)) differences schemes were used for the finite differences methods.

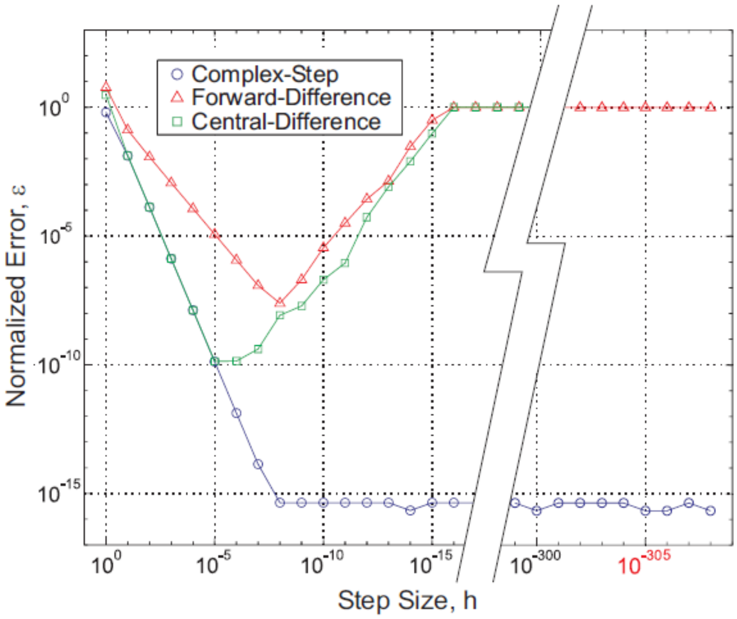


Figure 3.1: Error of sensitivity estimates given by complex step (blue), forward finite difference (red) and central difference (green) with the analytic method result as the reference value [34].

The first aspect to point out, from the initial slope observation in Figure 3.1, is that the rate of convergence for the forward-difference method is linear (associated with the $\mathcal{O}(\Delta x)$ approximation), as for the central-difference and complex step methods is quadratic (related to the $\mathcal{O}[(\Delta x)^2]$ approximation), as expected.

Secondly, it is possible to observe the effect of subtractive cancellation in the forward-difference method for step sizes below 10^{-8}, since the relative error (ε) increases with a decrease of the step size. For the central-difference method this occurs for step sizes below 10^{-5}. If the step size h is smaller than 10^{-16}, no changes will be observed in the output of the function f, leading to a zero sensitivity ($\varepsilon = 1$). In the complex-step method, no subtractive cancellation is observed, also as expected.

Evaluating accuracy in the approximation for each optimum step size, it is possible to state that the complex-step method is the most accurate, followed by the central-difference method and, lastly, the forward-difference method. Also note that, for the complex-step method, for step sizes below 10^{-8}, the normalized error converges to, approximately, 10^{-15}. When using this method, the lowest step size allowable corresponds to 10^{-308} since, for smaller steps, underflow will occur and the sensitivity will be zero.

3.4 Alternative Analytic Methods

According to Martins [35], alternative analytic approaches such as the adjoint and direct methods are the most efficient and accurate in sensitivity analysis. However, these methods require previous knowledge on the governing equations, which is a drawback when compared to other sensitivity analysis methods. For a general physical system, the governing equations can be represented in residual form,

$$\mathcal{R}(x_n, y_i(x_n)) = 0 \ , \tag{3.9}$$

where x_n is the design vector and y_i is the state variable vector, represented in the index notation. Also, consider a function of interest $f = f(x_n, y_i)$ that depends on the previous two vectors. A schematic representation of the considered case is shown in Figure 3.2.

In order to evaluate the total sensitivity of f, it is necessary to use the chain rule to capture not only the explicit influence of the design variables, but also their implicit influence associated to the state variables as

$$\frac{\mathrm{d}f}{\mathrm{d}x_n} = \frac{\partial f}{\partial x_n} + \frac{\partial f}{\partial y_i}\frac{\mathrm{d}y_i}{\mathrm{d}x_n} \ . \tag{3.10}$$

The partial derivative $\frac{\partial f}{\partial x_n}$, in Equation (3.10), is given by differentiating the explicit terms in the interest function's expression. In order to solve the second term on the right-hand side, two approaches are found: the direct sensitivity method and the adjoint sensitivity method, which are addressed in the following subsections.

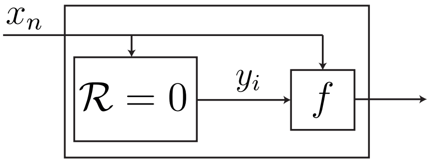


Figure 3.2: Representation of the governing equation ($\mathcal{R}$), interest function (f), design variables (x), state variables (y_i) for an arbitrary system.

3.4.1 Direct Sensitivity Method

The first approach consists in using direct equations to evaluate the function of interest's sensitivity. Since the governing Equation (3.9) must always be satisfied, it is a necessary condition that the total derivative of the residuals with respect to any design variable is always equal to zero,

$$\frac{\mathrm{d}\mathcal{R}}{\mathrm{d}x_n} = \frac{\partial \mathcal{R}}{\partial x_n} + \frac{\partial \mathcal{R}}{\partial y_i}\frac{\mathrm{d}y_i}{\mathrm{d}x_n} = 0 \ , \tag{3.11}$$

which is equivalent to

$$\frac{\partial \mathcal{R}}{\partial y_i}\frac{\mathrm{d}y_i}{\mathrm{d}x_n} = -\frac{\partial \mathcal{R}}{\partial x_n} \ . \tag{3.12}$$

Since the partial derivative of a performance function with respect to the state variables ($\frac{\partial f}{\partial y_i}$) can usually be explicitly obtained and the total derivative of the state variables with respect to design variables ($\frac{\mathrm{d}\, y_i}{\mathrm{d} x_n}$) is obtained by solving Equation (3.12), all the necessary information to solve Equation (3.10) is supplied.

Note that, by using the direct sensitivity method, Equation (3.12) has to be solved n times, due to the term on the right hand side of this equation. Thus, the computational cost of using the direct method will depend considerably on the number of design variables.

3.4.2 Adjoint Sensitivity Method

The second approach to evaluate a function of interest's sensitivity uses the adjoint formulation. Combining Equations (3.10) and (3.12), results

$$\frac{\mathrm{d} f}{\mathrm{d}\, x_n} = \frac{\partial\, f}{\partial\, x_n} - \underbrace{\frac{\partial\, f}{\partial\, y_i} \left[\frac{\partial \mathcal{R}}{\partial y_i}\right]^{-1}}_{\psi^T} \frac{\partial\, \mathcal{R}}{\partial\, x_n} \,. \tag{3.13}$$

An auxiliary adjoint vector ψ is defined and can be determined by solving the adjoint equation given by:

$$\left[\frac{\partial \mathcal{R}}{\partial y_i}\right]^T \psi = \left[\frac{\partial\, f}{\partial\, y_i}\right]^T \,. \tag{3.14}$$

Note that, with this method, the matrix Equation (3.14) only has to be evaluated once, considering that there is only one interest function. If k interest functions are regarded, the previous equation has to be evaluated k times. Therefore, the calculation of the adjoint vector is independent on the number of design variables, and the total sensitivity with respect to the design can be evaluated with a small additional computational cost.

To summarize: even though the partial derivative terms necessary in both methods are the same, the order of operations is different, affecting the computational cost in sensitivity calculation. If the number of interest functions is greater than the number of design variables, the direct method is preferable. Conversely, if the number of design variables exceeds the number of interest functions, the adjoint method shall be chosen.

3.5 Automatic Differentiation

Automatic differentiation, also known as algorithmic or computational differentiation, is another method used in numerical simulation programs to obtain the required derivatives efficiently and accurately. According to Bischof et al. [36], the sensitivity is obtained by applying the differentiation chain rule of Calculus systematically throughout the computer code, at elementary operator and intrinsic functions levels. This approach avoids approximation errors that exist in the finite difference method, and can be generated automatically, without the need of hand-coding, representing an advantage with respect to the alternative analytic methods. Another advantage of this method when compared to the alternative analytic methods is that, if major modifications need to be performed in the main program, a new updated version of the derivative computation is generated quickly, as for the latter case major adaptations may need to be implemented.

Consider a certain function $y = f(x)$. The goal is to computationally evaluate the function $f(x)$ and its derivatives $f'(x)$. Regard `f(x){...}` and `df(x){...}` as the computational versions of $f(x)$ and $f'(x)$, respectively. Also consider the schematic shown in Figure 3.3. A possible approach to compute both the function and its derivative values would be to differentiate the function $f(x)$ by hand or with the aid of symbolic differentiation (from 1. to 3.) and then to code both the function (from 1. to 2.) and its derivatives (3. to 4.). Alternatively, with the automatic differentiation, only function $f(x)$ needs to be coded (from 1. to 2.), avoiding the explicit calculation of $f'(x)$. The automatic differentiator will automatically calculate the associated derivatives (from 2. to 4.), whether it is through generated explicit code or through operator overloading, as it will be discussed later on in this chapter.

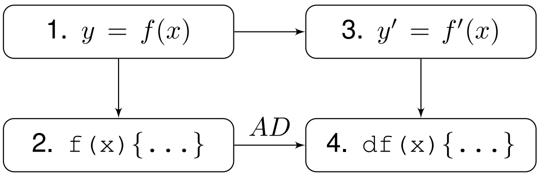


Figure 3.3: Schematics of the automatic differentiation (AD) alternative process (adapted from Berland [37]).

According to Berland [37], the procedure to implement an automatic differentiation algorithm comprises three steps:

1. Identify the intrinsic functions, based on the original source code;

2. Evaluate the derivatives of intrinsic functions;

3. Use the chain rule to obtain the desired derivative.

Martins [34] states that any program may be decomposed into m elementary functions T_i, with $i = 1, ..., m$, and considers that $t_i = T_i(t_1, ..., t_{i-1})$ correspond to the elementary variables. The chain rule of differentiation for a generic $\frac{\partial t_i}{\partial t_j}$, can be written as

$$\frac{\partial t_i}{\partial t_j} = \delta_{ij} + \sum_{k=j}^{i-1} \frac{\partial T_i}{\partial t_k}\frac{\partial t_k}{\partial t_j} \ , \tag{3.15}$$

where δ_{ij} is the Kronecker delta.

Two modes may be used to propagate the derivatives throughout the chain rule: the forward mode or the reverse mode. In the former, the bottom index j is chosen and kept fixed, varying the index i from j until the desired derivative (ultimately to m). In the latter approach, the i index is fixed, varying the index j, in a descendant order, from i until the desired quantity (ultimately down to 1).

The automatic differentiation method can be illustrated with a simple example. Consider a problem where two scalar functions (f_1 and f_2) depend on two variables (x_1 and x_2),

$$f_1(x_1, x_2) = \cos x_1 + x_1 \exp(x_2) \ , \tag{3.16}$$

$$f_2(x_1, x_2) = \cos^2 x_1 + x_2 \ . \tag{3.17}$$

In Figure 3.4, a schematic decomposition of f_1 and f_2 in elementary functions and operations (T_i), with the respective elementary variables (t_i), is presented.

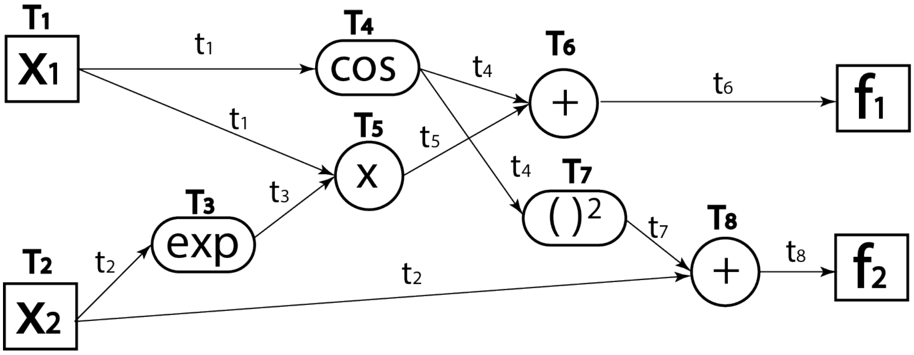


Figure 3.4: Decomposition of f_1 and f_2 in elementary functions.

Elementary functions	**Forward chain rule propagation**
$t_1 = T_1(x_1) = x_1$	$\dfrac{\partial t_1}{\partial x_1} \equiv \dfrac{\partial t_1}{\partial t_1} = 1$
$t_2 = T_2(x_2) = x_2$	$\dfrac{\partial t_2}{\partial x_1} \equiv \dfrac{\partial t_2}{\partial t_1} = 0$
$t_3 = T_3(t_2) = \exp(t_2)$	$\dfrac{\partial t_3}{\partial t_1} = \dfrac{\partial T_3}{\partial t_2}\dfrac{\partial t_2}{\partial t_1} = 0$
$t_4 = T_4(t_1) = \cos(t_1)$	$\dfrac{\partial t_4}{\partial t_1} = \dfrac{\partial T_4}{\partial t_1}\dfrac{\partial t_1}{\partial t_1} = -\sin t_1$
$t_5 = T_5(t_1, t_3) = t_1 \times t_3$	$\dfrac{\partial t_5}{\partial t_1} = \dfrac{\partial T_5}{\partial t_1}\dfrac{\partial t_1}{\partial t_1} + \dfrac{\partial T_5}{\partial t_3}\dfrac{\partial t_3}{\partial t_1} = t_3$
$t_6 = T_6(t_4, t_5) = t_4 + t_5$	$\dfrac{\partial t_6}{\partial t_1} = \dfrac{\partial T_6}{\partial t_4}\dfrac{\partial t_4}{\partial t_1} + \dfrac{\partial T_6}{\partial t_5}\dfrac{\partial t_5}{\partial t_1} = -\sin t_1 + t_3$
$t_7 = T_7(t_4) = t_4^2$	$\dfrac{\partial t_7}{\partial t_1} = \dfrac{\partial T_7}{\partial t_4}\dfrac{\partial t_4}{\partial t_1} = 2\, t_4$
$t_8 = T_8(t_2, t_7) = t_2 + t_7$	$\dfrac{\partial t_8}{\partial t_1} = \dfrac{\partial T_8}{\partial t_7}\dfrac{\partial t_7}{\partial t_1} + \dfrac{\partial T_8}{\partial t_2}\dfrac{\partial t_2}{\partial t_1} = 2\, \cos t_1$

Table 3.1: Elementary functions listing and chain rule sequential forward propagation.

In Table 3.1, in the left column, the elementary function decomposition is listed and, in the right column, the chain rule forward propagation is represented. Note that the propagation is performed regarding Equation (3.15) by keeping the the index $j = 1$ (held constant) and varying i from 1 to 8. The terms $\frac{\partial T_i}{\partial t_k}$ are obtained by explicitly differentiating the expressions on the left side column, and the $\frac{\partial t_k}{\partial t_j}$ terms are obtained from previous calculation steps in the chain. Note that the terms $\frac{\partial T_i}{\partial t_k} = 0$ are omitted. With the presented chain propagation sweep on forward mode, the entries of the first column vector of the Jacobian matrix, are calculated as

$$\frac{\partial f_1}{\partial x_1} = \frac{\partial t_6}{\partial t_1} = -\sin(x_1) + \exp(x_2)\ , \tag{3.18}$$

$$\frac{\partial f_2}{\partial x_1} = \frac{\partial t_8}{\partial t_1} = 2\cos(x_1)\ . \tag{3.19}$$

For the reverse chain rule propagation case, the terms $\frac{\partial t_k}{\partial t_j}$ are calculated from the end to the beginning of the chain. For instance, to obtain the derivatives $\frac{\partial f_1}{\partial x_1} = \frac{\partial t_6}{\partial t_1}$ and $\frac{\partial f_1}{\partial x_2} = \frac{\partial t_6}{\partial t_2}$, it is necessary to fix the index $i = 6$ and evaluate the cycle for $j = 5, ..., 1$.

Thus, it is possible to conclude that one sweep of forward mode determines one column vector of the Jacobian matrix (represented in red in Equation (3.20)), and one sweep on the reverse mode calculates one row vector of the Jacobian (represented in blue in Equation (3.20)). Therefore, the reverse mode is desirable if there are more variables than functions, and the forward mode is best suited if the number of functions is greater than the number of variables.

$$J = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{pmatrix} \tag{3.20}$$

However, in the reverse mode it is necessary to run the code once forward, storing all the intermediate variables, and once backward to apply the backwards chain rule. The total number of operations is independent of n, but the necessary memory is not. Therefore, the cost in terms of memory when using this method may be a drawback, becoming even prohibitive in certain cases. In the forward mode, this does not occur since the computation of gradient of each variable proceeds with the computation of each variable. Furthermore, the forward approach is easier to implement. In this work, only the forward mode will be used.

The implementation of the forward automatic differentiation can be performed either by source code transformation (1.) or by operator overloading (2.):

1. According to Hogan [38], the source code transformation utilizes a ”source-to-source compiler” that analyses the original source code, parses the information and adds the extra code associated with the derivative calculation. This new file is then compiled as the original. The main advantages with this approach is that it is possible to be implemented in all computer languages and tends to generate faster code. However, some disadvantages are identified, such as the fact that enlarged generated code is practically unreadable, making it more difficult to detect possible bugs. Furthermore, if changes are made in the source code, one must rerun the parser before compiling the new derivative code version.

2. The second approach utilizes operator overloading, a feature that is available in modern computer languages, where different operators have different implemented functionalities depending on their arguments. This feature allows the definition of ”active variables”, which can be seen as dual numbers, where the first part of the number is associated to the function’s value and the dual part of the number contains its derivative information. The operators (or mathematical functions) are redefined to perform not only the standard correspondent operation, but also to deal with the dual component by using the definition of derivate for that operator. The main advantages inherent to this approach are the fact that it is easier to code the automatic differentiation tool

and there is no need to generate a new file from source code manipulation. However, it is only possible to implement it in languages that support operator overloading. Another disadvantage in this approach is that compilers lag behind with operation overloading, and the whole code runs slower.

ADiMat Review

Automatic Differentiation for MATLAB® (*ADiMat*) is a tool developed in MATLAB® by Bischof et al. [39] that uses the source code transformation technique to perform automatic differentiation. This process is divided in two steps:

1. Source code transformation: the master file, corresponding to a `function.m` file, contains the top level function and is used as input. A differentiation code is obtained based on this master file through code parsing and manipulation. If the forward mode option is chosen, a `g_function.m` file is generated, that is able to calculate the gradient information.

2. Transformed code evaluation: the transformed code `g_function.m` is used to calculate the desired derivative information, with the aid of a freeware runtime environment.

In order to use *ADiMat* it is necessary to execute the `ADiMat_startup` command within MATLAB®, in the *ADiMat* installed directory. This will add the necessary directories of *ADiMat* to the current path and set the environment variables in order to be possible to use the *ADiMat* libraries. In Figure 3.5, it is presented the code for obtaining the Jacobian matrix (J) of the previously presented example (recall Figure 3.4):

```
admTransform(@F); % produces the differentiated function g_F.m
adimat_derivclass('opt_derivclass'); % select runtime environment
S = eye(2); % create seed matrix
[g_x1 g_x2] = createSeededGradientsFor(S, x1, x2); % create derivative inputs
[g_f1 f1 g_f2 f2] = g_F(g_x1, x1, g_x2, x2); % run the differentiated function
J = admJacFor(g_f1, g_f2); % extract derivative values
```

Figure 3.5: *ADiMat* application example.

Due to a great number of syntactic constructs in MATLAB® , not all features are completely implemented in *ADiMat*. The syntactic constructs are parsed, but some of their meaning is only partially or not at all known to *ADiMat*. If *ADiMat* encounters an unknown construct, a warning message is printed, but the differentiation process continues. Therefore, attention must be paid if warning messages are output and the differentiated code has to be checked.

3.6 Benchmark

As a first approach to the implementation of sensitivity analysis methods to structural problems, a cantilevered beam with a point load (P) applied at the tip will be considered. This beam has a rectangular cross-section shape with a constant width b along its span, and heights h_1 and h_2 on the first and second halves of its length L, respectively, as represented in Figure 3.6. In order to study this problem, an Euler-Bernoulli formulation will be adopted, meaning that the plane cross sections perpendicular to the beam's axis remain plane and perpendicular to the axis after deformation [40].

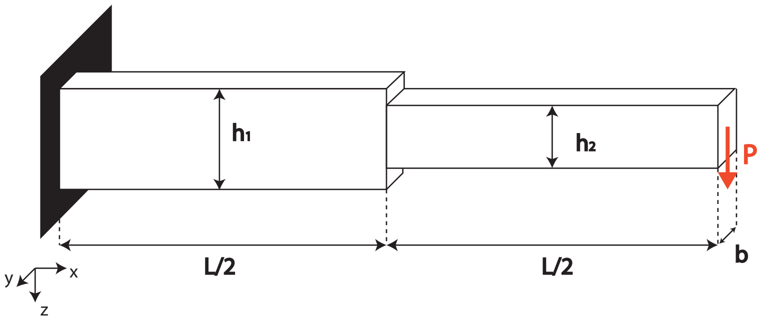


Figure 3.6: Cantilevered beam representation.

In the following subsections, a definition of the structural model using the Finite Element Method will be presented and a sensitivity analysis will be performed.

3.6.1 Model Definition

According to Reddy [41], the Finite Element Method (FEM) is a numerical method that is used to solve real world problems with complex physics, geometry and boundary conditions. This is performed by dividing the overall problem's domain in many subdomains, and approximating the solution of the governing equation within each subdomain with simplified functions, such as polynomials. The overall solution will be a collection of the subdomain solutions, making sure that each segment of the subdomain solution (and possibly derivatives up to a chosen order) are continuous at the connecting points. The implementation of the Finite Element Method may be divided in the following steps:

1. Differential governing equation definition (strong formulation);
2. Domain discretization;
3. Weak form formulation;
4. Shape and weight functions definition;
5. Element matrix formulation;
6. Global matrix and force vector assembly;
7. Nodal boundary conditions application;
8. Global system of equations solution;
9. Postprocessing of the solution.

The Euler-Bernoulli finite element formulation will be used to illustrate the steps in the application of the finite element method, based on the work of Reddy [41].

1. Differential governing equation

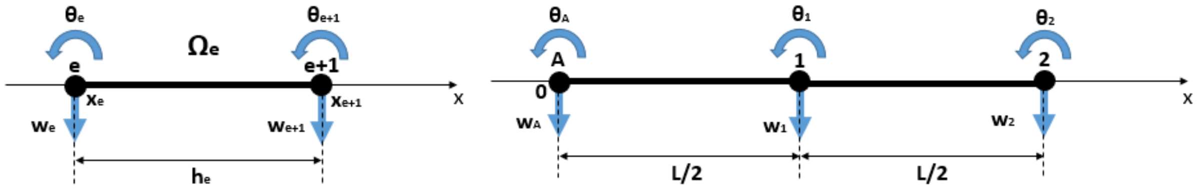
The first step in the FEM method, is to identify the differential governing equation that describes the problem. In this case, the bending deformation is characterized by the following one-dimensional fourth-order governing differential equation,

$$\frac{d^2}{dx^2}\left(EI_{yy}\frac{d^2w}{dx^2}\right) - q(x) = 0\;, \tag{3.21}$$

where E is the Young's modulus of the material, I_{yy} is the second moment of area about the y axis, w is the vertical displacement and $q(x)$ is a vertical distributed load along the beam's length (which is zero in the present case).

2. Domain discretization

The next step is to divide the domain Ω into a n_{fe} number of finite elements Ω_e. The generic finite element used as reference is represented in Figure 3.7a. This finite element of length h_e has 2 nodes, with two degrees of freedom each: a vertical displacement w and a rotation θ. The rotations correspond to the derivative of the vertical displacement with respect to the beam spanwise coordinate ($\frac{\mathrm{d}w}{\mathrm{d}x}$). As an example, the cantilevered beam is discretized into two finite beam elements of length $L/2$, as shown in Figure 3.7b.



(a) Generic finite element $\Omega_e = (x_e, x_{e+1})$.

(b) Beam domain discretization in 2 finite elements.

Figure 3.7: Finite element representation.

3. Weak form formulation

The weak form of a differential equation corresponds to an integral version of the residual of the differential equation multiplied by a weight function $v(x)$, in which the differentiation is transferred from the dependent variable (w, in this case) to the weight function v, so that all boundary conditions of the problem may also be included in the integral statement.

Considering Equation (3.21), multiplying it by an arbitrary weight function $v(x)$, that is twice differentiable with respect to x, and integrating in the element domain, yields

$$\int_{x_e}^{x_{e+1}} v\left[\frac{\mathrm{d}^2}{\mathrm{d}x^2}\left(EI\frac{\mathrm{d}^2w}{\mathrm{d}x^2}\right) - q(x)\right]\mathrm{d}x = 0\;. \tag{3.22}$$

Integrating by parts twice, results

$$\int_{x_e}^{x_{e+1}}\left[-\frac{\mathrm{d}v}{\mathrm{d}x}\frac{\mathrm{d}}{\mathrm{d}x}\left(EI\frac{\mathrm{d}^2w}{\mathrm{d}x^2}\right) - v\,q(x)\right]\mathrm{d}x + \left[v\frac{\mathrm{d}}{\mathrm{d}x}\left(EI\frac{\mathrm{d}^2w}{\mathrm{d}x^2}\right)\right]_{x_e}^{x_{e+1}} = 0\;, \tag{3.23a}$$

and

$$\int_{x_e}^{x_{e+1}} \left(EI \frac{\mathrm{d}^2 v}{\mathrm{d}x^2} \frac{\mathrm{d}^2 w}{\mathrm{d}x^2} - v q(x) \right) \mathrm{d}x + \left[v \frac{\mathrm{d}}{\mathrm{d}x} \left(EI \frac{\mathrm{d}^2 w}{\mathrm{d}x^2} \right) - \frac{\mathrm{d}v}{\mathrm{d}x} EI \frac{\mathrm{d}^2 w}{\mathrm{d}x^2} \right]_{x_e}^{x_{e+1}} = 0 \, . \tag{3.23b}$$

Note that, in the integral term of Equation (3.23b), the differentiation terms are equally distributed between the independent variable (w) and the weight function (v), and two boundary expressions appear evaluated at each boundary node. The terms $EI\frac{\mathrm{d}^2 w}{\mathrm{d}x^2}$ and $-\frac{d}{dx}\left(EI\frac{\mathrm{d}^2 w}{\mathrm{d}x^2}\right)$, are related to the external applied moments (M) and forces (V), respectively.

4. Shape functions

In this step of the Finite Element Method, the shape functions are defined using Galerkin's method. According to this method, the weight and the approximation functions are assumed to be equal. It is necessary to guarantee the essential boundary conditions, meaning that not only the chosen functions have to be continuous (associated with the continuity of displacements), but also their first derivative continuity has to be verified (associated with the continuity of rotations). Therefore, Hermite cubic polynomials are chosen as approximation functions, satisfying the previous conditions,

$$\phi_1 = 1 - 3s^2 + 2s^3; \quad \phi_2 = -h_e\,(s - 2s^2 + 2s^3); \quad \phi_3 = 3s^2 - 2s^3; \quad \phi_4 = -h_e(s^3 - s^2); \tag{3.24}$$

where $s = \frac{\bar{x}}{h_e}$ corresponds to a local relative coordinate within each element, along the local $\bar{x}$ direction.

5. Element matrix formulation

Applying the Galerkin's concept of shape functions in Equation (3.23), it is possible to identify a bilinear form (K_{ij}^e), associated to the stiffness matrix,

$$K_{ij}^e = \int_{x_e}^{x_{e+1}} EI \frac{\mathrm{d}^2 \phi_i^e}{\mathrm{d}x^2} \frac{\mathrm{d}^2 \phi_j^e}{\mathrm{d}x^2} \mathrm{d}x \, . \tag{3.25}$$

The linear form (F_i^e) can also be identified and it corresponds to the force vector,

$$F_i^e = \int_{x_e}^{x_{e+1}} \phi_i^e q \mathrm{d}x + Q_i^e \, , \tag{3.26}$$

where Q_i^e was used to represent the boundary terms related to shear forces and bending moments general terms.

In the present case, $q(x) = 0$ and $I = \frac{bh^3}{12}$. Thus, the general element stiffness matrix, $[K^e]$, is given by

$$[K^e] = \frac{Ebh^3}{6h_e^3} \begin{bmatrix} 6 & -3h_e & -6 & -3h_e \\ -3h_e & 2h_e^2 & 3h_e & h_e^2 \\ -6 & 3h_e & 6 & 3h_e \\ -3h_e & h_e^2 & 3h_e & 2h_e^2 \end{bmatrix} . \tag{3.27}$$

The general element force vector, $\mathbf{f^e}$, may be written as

$$\mathbf{f^e} = [Q_1^e \; Q_2^e \; Q_3^e \; Q_4^e]^T \, . \tag{3.28}$$

6. Global matrix and force vector assembly

The next step is to assemble a global stiffness matrix and force vector by combining the contribution of each finite element. The global assembly is obtained based on the interelement continuity of the primary variables (deflection and slope or rotation) and the interelement equilibrium of the secondary variables (shear forces and bending moments) at the nodes of the elements. This can be done with the aid of a connectivity table, which contains the information of the global position of each element's degrees of freedom. The result will be a squared symmetric stiffness matrix, $[K_{glob}]$, and a column vector, $\mathbf{f_{glob}}$, with a dimension equal to the total number of degrees of freedom.

7. Nodal boundary conditions

After the global assembly, the resultant stiffness matrix is singular, and, therefore, no solution of the problem can be obtained. Thus, it is necessary to determine the reduced form. In this case, the beam is clamped at the node A, and therefore, two degrees of freedom (the vertical displacement and rotation at point A) are constrained, being equal to zero. These primary boundary conditions correspond to eliminating the first and second rows and columns of the global stiffness matrix, resulting in the reduced stiffness matrix

$$[K] = \frac{Eb}{6h_e^3} \begin{bmatrix} 6(h_1^3 + h_2^3) & 3h_e(h_1^3 - h_2^3) & -6h_2^3 & -3h_e h_2^3 \\ 3h_e(h_1^3 - h_2^3) & 2h_e^2(h_1^3 + h_2^3) & 3h_e h_2^2 & h_e^2 h_2^3 \\ -6h_2^3 & 3h_e h_2^3 & 6h_2^3 & 3h_e h_2^3 \\ -3h_e & h_e^2 h_2^3 & 3h_e h_2^3 & 2h_e^2 h_2^3 \end{bmatrix} . \tag{3.29}$$

The force vector contains the secondary boundary conditions informations, which correspond to the external nodal loads. The first two rows of the global vector are removed, due to the clamped constraint, originating the reduced force vector

$$\mathbf{f} = [0\ 0\ P\ 0]^T \ , \tag{3.30}$$

with P being the applied load at the tip of the beam.

8. Global system of equations solution

Once the reduced stiffness matrix $[K]$ and force vector $\mathbf{f}$ are determined, it is possible to obtain the solution of the structural problem, by solving the matrix equation

$$[K]\,\mathbf{y} = \mathbf{f} \ . \tag{3.31}$$

The state vector $\mathbf{y} = [w_1\ \theta_1\ w_2\ \theta_2]^T$ will contain the resultant displacements and rotations of nodes 1 and 2.

9. Post-processing of the solution

The post-processing stage of the FEM corresponds to using the solution obtained from Equation (3.31), to represent the obtained results or to compute other quantities. An example of a post-processed quantity is the bending moment (M^e) applied at any point within the element Ω^e, which is given by

$$M^e(\bar{x}) = -EI \frac{\mathrm{d}^2}{\mathrm{d}\bar{x}^2} [\phi_1\ \phi_2\ \phi_3\ \phi_4]\, \mathbf{y^e} \ , \tag{3.32}$$

where $\bar{x}$ is the local position coordinate within the finite element and $\mathbf{y}^\mathrm{e}$ is the solution at the nodes that delimit finite element e.

3.6.2 Sensitivity Analysis

In order to perform a design sensitivity analysis of the clamped beam represented in Figure 3.6, it is necessary to define a baseline configuration. The beam specifications and the magnitude of the load applied are summarized in Table 3.2.

Design variables	**Value**	**Constant parameters**	**Value**
Cross-sectional width (b)	25 mm	Applied force (P)	100 N
Cross-sectional height segment 1 (h_1)	35 mm	Young's modulus (E)	70E9 Pa
Cross-sectional height segment 2 (h_2)	30 mm	Beam's length (L)	2 m

Table 3.2: Beam parameters definition.

In the present case, the design vector contains 3 design variables: $\mathbf{x} = [b \, h_1 \, h_2]^T$. The reduced problem has 4 state variables $\mathbf{y} = [w_1 \, \theta_1 \, w_2 \, \theta_2]^T$, which correspond to vertical displacements (w) and rotations (θ) of nodes 1 and 2.

Two interest functions will be used in this problem. The first function (f_1) corresponds to the tip vertical displacement, which is simply given by

$$f_1 = w_2 \; . \tag{3.33}$$

The second function of interest (f_2) is the bending moment at the built-in end, which is directly given by

$$f_2 = M_A = \frac{-EI}{h_e^2} \left. \begin{bmatrix} -6+12s \\ -h_e(-4+6s) \\ 6-12s \\ -h_e(6s-2) \end{bmatrix}^T \right|_{s=0} \cdot \begin{bmatrix} w_A \\ \theta_A \\ w_1 \\ \theta_1 \end{bmatrix} = \frac{-Ebh_1^3}{6h_e^2}(3w_1 + h_e\theta_1) \; , \tag{3.34}$$

where $w_A = 0$ and $\theta_A = 0$, since the beam is fixed at the point A.

Note that it is known that the moment at the built-in end will not depend on the design variables, if body forces are neglected, and therefore, a zero sensitivity is expected. Hence, this function will only be used to verify the code implementation.

The previous information is enough to determine the design sensitivity through the finite difference methods, by using Equations (3.2a), (3.2b) and (3.3), and the complex method, by using Equation (3.6). Since there are three design variables, it will be required to solve the structural Equation (3.31) four times to obtain sensitivities through forward finite differences or complex method: one for the current design plus three for the perturbed designs. For the central difference case, six evaluations must be performed: one positive and one negative design perturbation for each of the three design variables.

In order to use alternative analytical methods, additional information must be provided. The first step is to write Equation (3.31) in the residual form, using index notation,

$$\mathcal{R} = K_{ki}\, y_i - F_k = 0\ . \tag{3.35}$$

The partial derivatives of the residual Equation (3.35) with respect to the design and state variables are given by

$$\frac{\partial \mathcal{R}}{\partial x_n} = \frac{\partial K_{ki}}{\partial x_n}\tilde{y_i} \tag{3.36a}$$

and

$$\frac{\partial \mathcal{R}}{\partial y_i} = K_{ki}\ , \tag{3.36b}$$

where $\tilde{y_i}$ corresponds to the solution from the residual equation held constant.

Note that the force vector does not depend on the design nor on the state variables, meaning that $\frac{\partial F_k}{\partial x_n} = 0$ and $\frac{\partial F_k}{\partial y_i} = 0$.

The next step is to define the partial derivatives of the interest functions f_1 and f_2 with respect to the design variables (x_n), given, respectively, by

$$\frac{\partial f_1}{\partial x_n} = \left[\frac{\partial f_1}{\partial b}\ \frac{\partial f_1}{\partial h_1}\ \frac{\partial f_1}{\partial h_2}\right] = [0\ 0\ 0] \tag{3.37a}$$

and

$$\frac{\partial f_2}{\partial x_n} = \left[\frac{\partial f_2}{\partial b}\ \frac{\partial f_2}{\partial h_1}\ \frac{\partial f_2}{\partial h_2}\right] = \left[\frac{-Eh_1^3}{6h_e^2}(3w_1 + h_e\theta_1)\ \frac{-Ebh_1^2}{2h_e^2}(3w_1 + h_e\theta_1)\ 0\right]\ . \tag{3.37b}$$

The interest function's derivatives with respect to the state variables (y) are given by

$$\frac{\partial f_1}{\partial y_i} = \left[\frac{\partial f_1}{\partial w_1}\ \frac{\partial f_1}{\partial \theta_1}\ \frac{\partial f_1}{\partial w_2}\ \frac{\partial f_1}{\partial \theta_2}\right] = [0\ 0\ 1\ 0] \tag{3.38a}$$

and

$$\frac{\partial f_2}{\partial y_i} = \left[\frac{\partial f_2}{\partial w_1}\ \frac{\partial f_2}{\partial \theta_1}\ \frac{\partial f_2}{\partial w_2}\ \frac{\partial f_2}{\partial \theta_2}\right] = \left[\frac{Ebh_1^3}{2h_e^2}\ \frac{Ebh_1^3}{6h_e^2}\ 0\ 0\right]\ . \tag{3.38b}$$

The terms $\frac{\partial K_{ki}}{\partial x_n}$, are obtained by differentiating the reduced matrix $[K]$, represented in Equation (3.29), with respect to each design variable. For instance, $\frac{\partial K_{ki}}{\partial x_2}$ may be represented as

$$\frac{\partial\,[K]}{\partial h_1} = \frac{Eb}{2h_e^3}\begin{bmatrix} 6h_1^2 & 3h_eh_1^2 & 0 & 0 \\ 3h_eh_1^2 & 2h_e^2h_1^2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}\ . \tag{3.39}$$

Similarly, it is possible to obtain $\frac{\partial[K]}{\partial b}$ and $\frac{\partial[K]}{\partial h_2}$.

With all the previous information, the requirements to apply either the direct or the adjoint method are met. However, the order of application of the determined quantities will depend on the method. In the direct method, the direct Equation (3.12), becomes

$$K_{ki}\,\frac{\mathrm{d}\,y_i}{\mathrm{d}\,x_n} = -\frac{\partial\,K_{ki}}{\partial\,x_n}\,\tilde{y}_i\;. \tag{3.40}$$

Equation (3.40) must be evaluated three times, one for each design variable, since the term $\frac{\partial\,K_{ki}}{\partial\,x_n}$ is different for each case, and each evaluation has a cost similar to solving the structural problem solution given by Equation (3.31). Note that the calculation of $\frac{\mathrm{d}\,y_i}{\mathrm{d}\,x_n}$ from Equation (3.40) is independent of the number of interest functions and the solution is directly applied in Equation (3.11), to determine the total sensitivity ($\frac{\mathrm{d}\,f}{\mathrm{d}\,x_n}$).

In the adjoint method, the adjoint Equation (3.14), becomes

$$K_{ik}\,\psi_k = \frac{\partial f}{\partial y_i}\;. \tag{3.41}$$

Note that the symmetry property of the stiffness matrix, $K_{ik} = K_{ki}$, was used. Equation (3.41) is independent of the number of design variables, only depending on the number of the interest functions f. Thus, this equation has to be evaluated twice, since two functions of interest are considered. After calculating the adjoint vector for both functions, these shall be applied in Equation (3.13), to find the total sensitivity ($\frac{\mathrm{d}\,f}{\mathrm{d}\,x_n}$).

The last method used in this analysis is the automatic differentiation, which is obtained by running *ADiMat*, as explained in Section 3.5.

Sensitivity Reference Results

The design sensitivities obtained with the adjoint method are represented in Table 3.3. These values were chosen to be used as reference, since the only error associated to this method is the computational numerical precision. Even though this method was chosen, any of the other exact methods could also be used as the reference case.

Sensitivity	$\frac{\mathrm{d}f_1}{\mathrm{d}b}$ [-]	$\frac{\mathrm{d}f_1}{\mathrm{d}h_1}$ [-]	$\frac{\mathrm{d}f_1}{\mathrm{d}h_2}$ [-]	$\frac{\mathrm{d}f_2}{\mathrm{d}b}$ [N]	$\frac{\mathrm{d}f_2}{\mathrm{d}h_1}$ [N]	$\frac{\mathrm{d}f_2}{\mathrm{d}h_2}$ [N]
Value	-1.83134E-05	-3.19867E-05	-8.46561E-06	-1.47793E-12	9.09495E-13	-2.01836E-13

Table 3.3: Sensitivity analysis results with the adjoint method.

Based on these results, it is possible to confirm that the increase of b, h_1 or h_2, leads to a decrease in the tip deflection (f_1), since the sensitivity values are negative for the three cases. One can also state that the best way to decrease the tip deflection, will be to increase the cross-section's height near the built-in end (h_1). As it was expected, the sensitivities concerning the function of interest f_2, are negligible and these values are not exactly zero, due to computational errors.

Optimal Steps for Finite Differences

Recall Equation (3.8) that will be used to represent the relative errors (ε) of a certain sensitivity f', where f'_{ref} corresponds, in this case, to the sensitivity calculated with the adjoint method.

In Table 3.4 it is represented the relative optimum steps that minimize the sensitivity error of the interest function f_1 for each of the three design variables. This is performed for the forward and central difference methods and the value of the relative error ε is also presented for each situation. In order to obtain these values, the developed program was run thousands of times, exponentially decreasing the relative step size and storing this value, and the respective sensitivity error. Then, a search was performed throughout the stored sensitivity error list, to look for the minimum value of ε. Once this value was found, both the relative error and the correspondent optimum step were output.

Sensitivity $\left(\frac{\mathrm{d}f_1}{\mathrm{d}x_i}\right)$	**Forward difference**		**Central difference**	
	Optimum step $\left\lvert\frac{\Delta x_i}{x_i}\right\rvert$	$\varepsilon\left(\frac{\mathrm{d}f_1}{\mathrm{d}x_i}\right)$	**Optimum step** $\left\lvert\frac{\Delta x_i}{x_i}\right\rvert$	$\varepsilon\left(\frac{\mathrm{d}f_1}{\mathrm{d}x_i}\right)$
$\frac{\mathrm{d}f_1}{\mathrm{d}b}$	2.512E-08	1.56793E-08	7.943E-06	3.24468E-11
$\frac{\mathrm{d}f_1}{\mathrm{d}h_1}$	1.995E-08	4.51732E-08	1.000E-06	4.61789E-11
$\frac{\mathrm{d}f_1}{\mathrm{d}h_2}$	1.585E-08	3.80878E-09	1.000E-05	2.06275E-10

Table 3.4: Optimum steps and relative errors when evaluating the sensitivities of f_1 using forward and central difference methods.

In the forward difference case, the order of magnitude for the relative step size is 10^{-8} for all the design variables. In the central difference method it is 10^{-5} or 10^{-6}, depending on the design variable. Furthermore, one is able to check that, for the optimum steps, central difference yields better sensitivity results than forward differences, since the relative errors are lower in the former method. Note that, in practice, in the majority of the engineering problems, the true sensitivity is unknown, making it difficult to guarantee that a certain step is the optimal.

Complex Step - Avoiding subtractive cancelation

In Figure 3.8, the relative error of the tip displacement sensitivity with respect to the beam cross-sectional width, is presented in a logarithmic scale. Linear convergence is observed for the forward difference, until a step of approximately 10^{-8}. For smaller steps than 10^{-8}, subtractive cancellation effects are noticeable and the relative error increases, reaching a value of 1 at a relative step of, approximately, 10^{-16} .

For the central difference method, it can be observed, in the logarithmic scaled plot, the quadratic convergence of ε until a relative step size of, approximately, 10^{-5}. For relative step sizes between $[10^{-16}, 10^{-5}]$, subtractive cancellation occurs, and the relative error ramps up. For steps lower than 10^{-16}, a $+\Delta b$ or $-\Delta b$ perturbation in the design yields the same output of f_1, meaning that the sensitivity $\frac{\mathrm{d}f_1}{\mathrm{d}b}$ will be zero ($\varepsilon = 1$).

For the complex step method a quadratic convergence is also obtained, and subtractive cancellation does not occur. The relative error tends to a value of, approximately, $\varepsilon = 10^{-15}$, for steps lower than 10^{-9}. The real step size Δb can be reduced down to 10^{-308} and, for lower step sizes, underflow occurs, the perturbation Δb will be considered as zero and, therefore, the outputs of f_1 will be equal, meaning that $\epsilon = 1$.

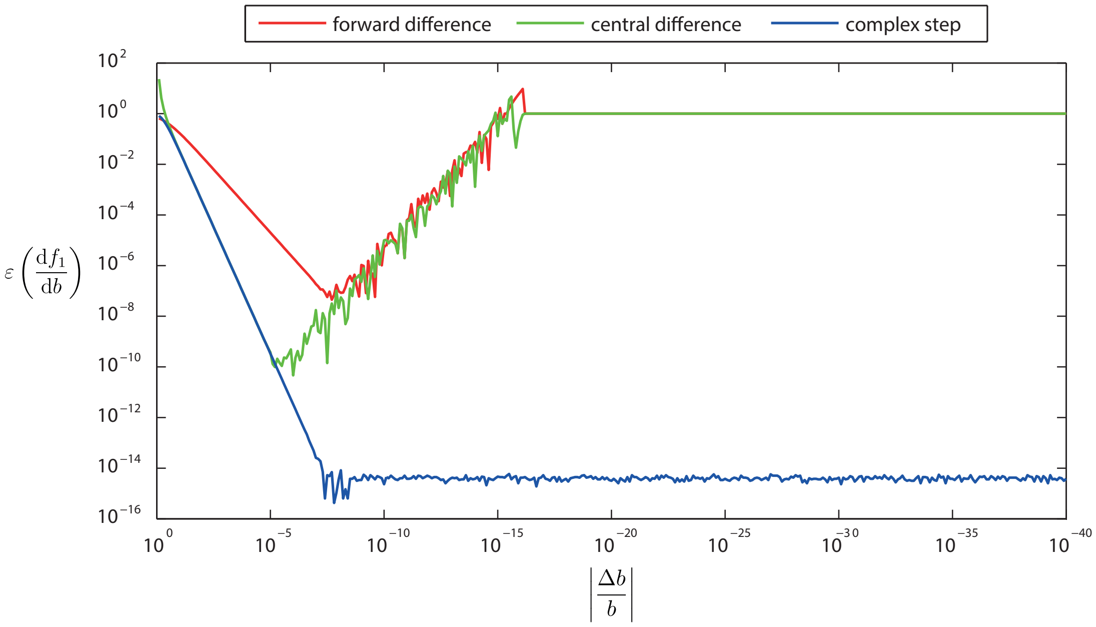


Figure 3.8: Logarithmic plot of $\varepsilon(\frac{\mathrm{d}f_1}{\mathrm{d}b})$ as a function of the relative step size.

Note about precision

MATLAB® uses the double-precision (or double) data type according to IEEE Standard 754, requiring 64 bits to store a value: 1 bit to represent the sign, 11 to represent the exponent and 52 to represent the fraction. With this standard, negative numbers are repesented in the range $[-1.79769 \times 10^{+308}; -2.22507\times10^{-308}]$ and positive numbers fall in the range of $\left[+2.22507 \times 10^{-308}; +1.79769 \times 10^{+308}\right]$. Numbers larger than $+1.79769 \times 10^{+308}$ or smaller than $-1.79769 \times 10^{+308}$ are assigned as positive and negative infinity, respectively, due to overflow. If a number is smaller than $\mid 2.22507\times10^{-308} \mid$, in absolute value, underflow occurs and a zero value is assigned.

Methods comparison

In Table 3.5, the relative errors of the sensitivity of the tip vertical displacement f_1 with respect to the beam width b, h_1 and h_2 are listed for the six types of sensitivity analysis cases, using the adjoint method as reference. The relative steps used in the finite difference methods (forward and central) were the optimum for each sensitivity. For the complex step method, a relative step size of 10^{-35} was chosen for the three design variables, since this step corresponds to a converged value of the relative error of the sensitivity. Note that the complex step, automatic differentiation, direct and adjoint methods result in similar sensitivities, being the differences observed in relative errors lower than 10^{-14}, which can be assumed negligible.

With this simple structural example it was possible to identify the advantages and challenges in each method. If the problem contains few design variables and an approximation of the sensitivity information is enough, any of the finite differences methods may be applied. However, the choice of the step size perturbation may not be straightforward. In this case, an optimal step size was determined but, in most cases, this is not possible and, even if it was, it would be computationally very inefficient.

To avoid dealing with the subtractive cancellation effects and step size selection in finite difference

Method	$\epsilon\left(\frac{\mathrm{d}f_1}{\mathrm{d}b}\right)$	$\epsilon\left(\frac{\mathrm{d}f_1}{\mathrm{d}h_1}\right)$	$\epsilon\left(\frac{\mathrm{d}f_1}{\mathrm{d}h_2}\right)$
Forward difference	1.5679E-08	4.5173E-08	3.8088E-09
Central difference	3.2447E-11	4.6179E-11	2.0628E-10
Complex step	9.4354E-15	4.0251E-15	5.0028E-15
Automatic differentiation	2.5901E-15	8.4739E-16	1.0806E-14
Direct	5.5503E-16	0.0000E+00	3.8021E-15
Adjoint	-	-	-

Table 3.5: Comparison of the relative errors for different sensitivity analysis methods.

methods, the complex method may be used as an alternative. This approximation method showed first order sensitivity results comparable with exact methods. However, this method is still highly dependent on the number of design variables and requires software that is capable of computing complex arithmetics.

If exact methods are preferred, the automatic differentiation may be utilized. This method required using an auxiliary tool, *ADiMat*, to generate the derivative code. Thus, if the source code is available and the used functions are able to be interpreted by the automatic differentiation program, this constitutes a possible approach to compute gradients. Nevertheless, the forward automatic differentiation is still dependent on the number of design variables.

Lastly, the alternative analytic methods were also proved to be an option to obtain exact derivative information, requiring the implementation of an auxiliary code exploiting the physics of the problem. The first approach was the direct method which was verified to be dependent on the number of design variables, being useful in cases where the number of functions of interest is greater than the number of design variables. The second approach was the adjoint method, which was proven to be practically independent on the number of design variables, being useful in cases where the number of design variables exceeds the number of functions of interest.

Thus, this study provided an overview of the available sensitivity analysis methods with some guidelines to select the appropriate approach to be applied to the structural framework later in Chapter 6.

Chapter 4

Structural Framework

In this chapter a description of the structural framework will be provided. A 3D-beam finite element model will be ultimately used to represent the structure of an aircraft wing. However, it is first necessary to obtain the equivalent wing properties to characterize each beam element. After defining the 3D-beam model it will be possible to apply certain load conditions, to generate the structural response of the wing.

The framework is divided into three main modules, as represented in shaded boxes in Figure 4.1. The first module determines the cross-sectional properties at a certain position along the wing span, with the use of a thin-wall wingbox approximation model. It will be used to evaluate the cross-sectional properties at the nodes locations. The second module performs a weighted average of the structural properties between two sections of the wing. It will be used to determine each beam element properties, based on the boundary (nodal) properties. The third module generates the finite element stiffness matrix based on the equivalent properties previously determined and the structural solution when a load $\mathbf{f}$ is applied.

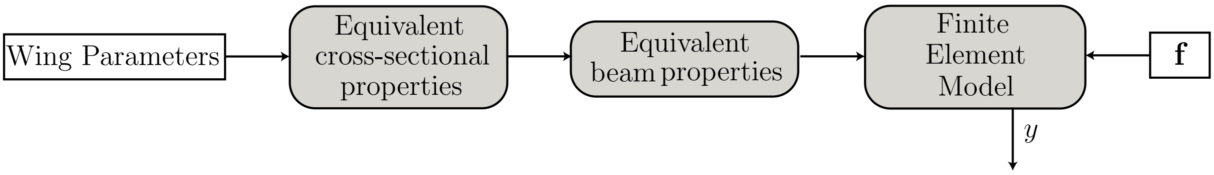

Wing Parameters
Equivalent cross-sectional properties
Equivalent beam properties
Finite Element Model
f
y

Figure 4.1: Structural framework.

Note that part of the structural tool from Almeida's framework [11] had to be redeveloped, since it was incompatible with algorithmic differentiation. The old tool to evaluate the structural properties was adapted from a third-party program with a complex structure, that could not be treated by the code parser, when generating the derivative code from the source file. Besides that, the original structural tool was built for composite blades that required running some modules to determine the orthotropic properties, unnecessary for structures made from isotropic homogeneous materials. Therefore, a new program had to be developed with a simplified structure, making sure every used function could be interpreted by *ADiMat*. Some changes were also introduced in the finite element model, more specifically in the definition of the rotation matrices, by using Euler angles, and in the polar moment of inertia (J) definition from the wingbox module, instead of assuming it equal to the sum of both moments of inertia.

In the following sections, each of the three steps of the framework will be explained in detail.

4.1 Equivalent Cross-sectional Properties

The first module in the structural framework calculates the equivalent cross-sectional properties based on the wing external and internal parameters, at a desired location along the wing span. A wingbox is firstly used to approximate the structural model, as represented in Figure 4.2.

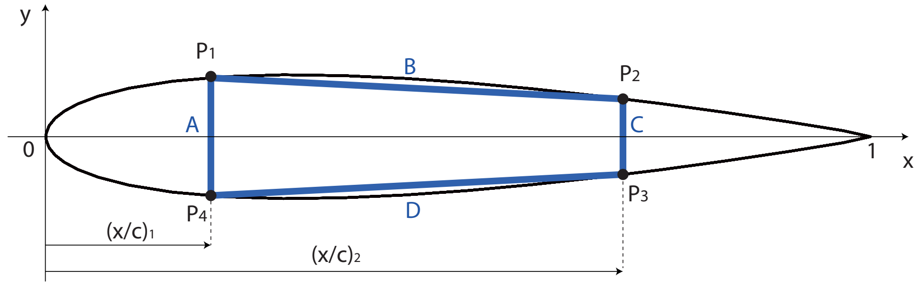


Figure 4.2: Thin-wall wingbox approximation of a NACA0015 airfoil, represented in the airfoil's reference frame.

The shear center location, the axial, flexural and torsional rigidities, and the linear density are evaluated using a developed function called `fequiprops.m`, for the specified material properties, airfoil chord length, spars relative locations and wingbox segments' thickness. The flexural rigidities are evaluated relative to an auxiliary reference frame with the same orientation of the airfoil's reference frame, but passing through the shear center.

A detailed list of the input and output variables is presented in Table 4.1. This module is also able to output the centroid location, as well as the moments of inertia relative to the centroid, upon user request.

Input variables	Output variables
Chord length (c)	Shear center location (SC_x, SC_y)
Front and rear spars location (x_1/c, x_2/c)	Axial rigidity (EA)
Front and rear spars thickness (t_A, t_C)	Flexural rigidity about the x-axis (EI_{xx})
Upper and lower skin thickness (t_B, t_D)	Flexural rigidity about the y-axis (EI_{yy})
Young's modulus (E)	Torsional rigidity (GJ)
Shear modulus (G)	Linear density (ρ_l)
Material density (ρ)	

Table 4.1: Equivalent cross-sectional properties module inputs and outputs.

All of the previous output variables are determined using the thin-wall assumption, which will be described in the next subsection.

4.1.1 Thin-Wall Assumption

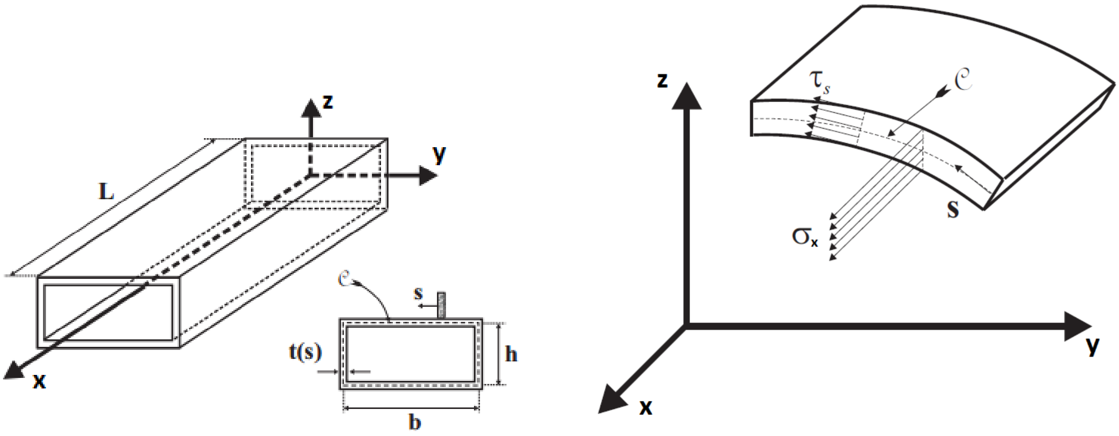
Larrosa [42] defines a structure as thin-walled if the wall thickness (t) is much smaller than a representative dimension of the cross-section (b):

$$\frac{t(s)}{b} \ll 1 \; . \tag{4.1}$$

In Figure 4.3a, an example of a rectangular cross-section thin-walled beam with width (b), height

(h) and length (L) is provided. The contour ($\mathcal{C}$) represents the mid-line of the thin-wall, along which a curvilinear variable (s) is defined with an arbitrary origin. In the thin-wall beam theory, it is assumed that the cross-section is infinitely rigid in its own plane, meaning that the normal stresses in the y- and z-direction (σ_y and σ_z) and the shear stress τ_{yz} are negligible. The relevant stress components are the axial stress σ_x, and the shear stresses τ_{xy} and τ_{yz}, which can be alternatively decomposed into τ_n and τ_s in an auxiliary reference frame, corresponding, respectively, to the normal and parallel directions to the contour $\mathcal{C}$. Since the wall is considered to be very thin, the stresses may be regarded as being constant across the thickness. Knowing that the shear stress τ_n must be zero at the outer surfaces of the wall, $\tau_n \approx 0$ within the structure, according to this theory (Megson [43]). A representation of the relevant stress components is found in Figure 4.3b. Instead of working with τ_s, it may be convenient to use the shear flow definition (q) as a shear force per unit length,

$$q = \tau\, t\,. \tag{4.2}$$



(a) Thin-walled box-beam example.

(b) Uniform axial and shear stresses distributions across the wall thickness.

Figure 4.3: Thin-wall beam representation. [42]

When using the thin-wall theory, the computation of sectional properties can also be simplified since the higher powers of t may be neglected. For a general thin section i, as represented in Figure 4.4, with length a_i, thickness t_i and an angle of inclination β_i relative to the centroidal $\bar{x}$-axis (defined positive counterclockwise), the moments and products of inertia relative to its centroid ($\bar{I}_{xx}^{(i)}$, $\bar{I}_{yy}^{(i)}$ and $\bar{I}_{xy}^{(i)}$) are calculated as

$$\bar{I}_{xx}^{(i)} = \frac{a_i^3\, t_i\, \sin^2(\beta_i)}{12}; \quad \bar{I}_{yy}^{(i)} = \frac{a_i^3\, t_i\, \cos^2(\beta_i)}{12}; \quad \bar{I}_{xy}^{(i)} = \frac{a_i^3\, t_i\, \sin(2\beta_i)}{24}\,. \tag{4.3}$$

Besides the geometric approximations, a linear elastic and homogeneous material model will be used. With the previous assumptions in mind, it is possible to implement the thin-walled beam box model approximation.

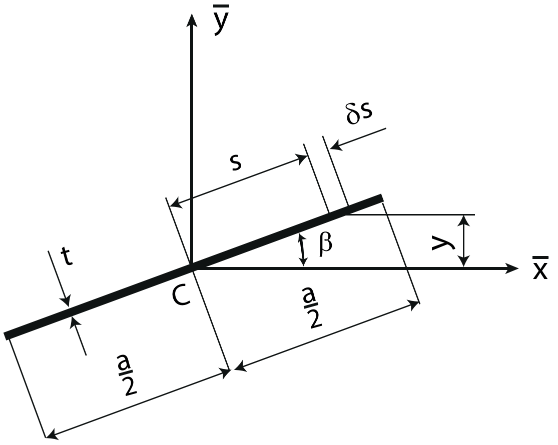


Figure 4.4: Thin wall section represented in its centroidal reference frame [43].

4.1.2 Implementation Procedure

The first step in the implementation is the definition of the internal structure's boundaries based on the airfoil characterization. A text file containing the normalized coordinate points defines the airfoil. Then, by inputting the spars' relative chordwise location $(x/c)_1$ and $(x/c)_2$, an upper and a lower limit points (P_i) will be calculated for each spar from spline interpolation among the airfoil coordinate points, as represented in Figure 4.2, defining the four wingbox vertices. The next step is to assign the thin sheet thickness to the front spar (t_A), upper skin (t_B), rear spar (t_C), and lower skin (t_D), as well as the material properties: density (ρ), Young's modulus (E) and shear modulus (G). All sheets are considered to be made from the same material and the upper and lower skins will be considered as one plane plate each (instead of curvilinear). The chord length (c) is then introduced to scale the cross-sectional model. The calculations presented in the following subsections is based on the work of Megson [43].

Shear Center Location

After defining the wingbox, it is necessary to determine the shear center location. The shear center is defined as the point where shear loads S_x and S_y are applied producing no beam twisting. To determine the shear center, the shear flow distribution has to be calculated using

$$q_s = q_b + q_{s0} \ , \tag{4.4}$$

where q_b corresponds to the basic shear flow, which is a shear flow distribution if the section was cut at a certain point (represented in Figure 4.5) and assumed to be loaded through its yet unknown shear center, and q_{s0} is the shear flow that needs to be added at the cut to produce the equivalent closed section shear flow. Note that the shear center location in Figure 4.5 is just merely representative. If the airfoil is symmetric and the upper and lower skin thicknesses are equal, it is known that the shear center will lie in the x-axis, due to symmetry along the horizontal axis.

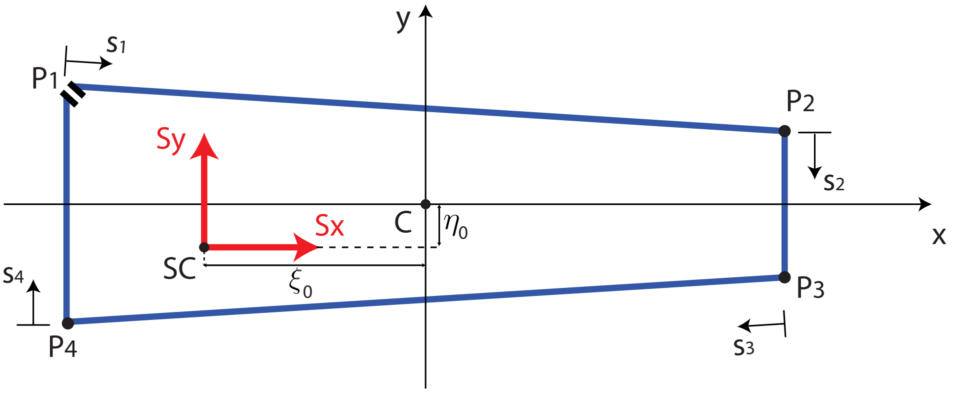


Figure 4.5: Wingbox open section representation, in the centroidal frame, with a cut at P_1. S_x and S_y correspond to the shear loads applied at the shear center.

Taking the origin of s at the cut of the section (where $q = 0$), the basic shear flow is given by

$$q_b = -\left(\frac{S_x I_{xx} - S_y I_{xy}}{I_{xx}\, I_{yy} - I_{xy}^2}\right) \int_0^{s^*} tx \, \mathrm{d}s - \left(\frac{S_y I_{yy} - S_x I_{xy}}{I_{xx}\, I_{yy} - I_{xy}^2}\right) \int_0^{s^*} ty \, \mathrm{d}s \;, \tag{4.5}$$

where t is the thickness distribution along the coordinate s, x and y are the coordinates relative to the centroid as a function of s, and I_{xx}, I_{yy} and I_{xy} correspond to the second moments of area and product of area relative to the centroidal frame, respectively. Therefore, it is first necessary to locate the centroid of the section and define the centroidal axis. The centroid of the entire section (x_C, y_C) is found from

$$(x_C,\, y_C) = \left(\frac{\sum_{i=1}^{4} x_{c_i}\, a_i t_i}{\sum_{i=1}^{4} a_i t_i}, \frac{\sum_{i=1}^{4} y_{c_i}\, a_i t_i}{\sum_{i=1}^{4} a_i t_i}\right) \;, \tag{4.6}$$

where a_i is the length, t_i is the thickness and (x_{c_i},y_{c_i}) is the centroid location of each thin-wall section in the airfoil reference frame.

The centroidal axis, represented in Figure 4.5, corresponds to the airfoil axis shifted to the centroid of the wingbox. The second moments of area and product of area relative to the centroidal frame (I_{xx}, I_{yy} and I_{xy}) are calculated by summing the individual contribution of each thin-wall section, given by

$$I_{xx}^{(i)} = \bar{I}_{xx}^{(i)} + a_i t_i (y_{c_i} - y_C)^2 \;, \tag{4.7a}$$

$$I_{yy}^{(i)} = \bar{I}_{yy}^{(i)} + a_i t_i (x_{c_i} - x_C)^2 \;, \tag{4.7b}$$

$$I_{xy}^{(i)} = \bar{I}_{xy}^{(i)} + a_i t_i (x_{c_i} - x_C)(y_{c_i} - y_C) \;. \tag{4.7c}$$

Equations (4.7) are obtained from applying the parallel axis theorem to Equations (4.3). With the moments and products of inertia calculated in the centroidal axis, it is possible to evaluate the basic shear flow distribution (q_{b_i}) at each thin-wall section i, as a function of the coordinate s (divided into four

s_i segments), by

$$q_{b_i}(s_i) = -\left(\frac{S_x I_{xx} - S_y I_{xy}}{I_{xx}\, I_{yy} - I_{xy}^2}\right) \int_0^{s_i^*} t_i\, x_i(s_i)\, \mathrm{d}s_i - \left(\frac{S_y I_{yy} - S_x I_{xy}}{I_{xx}\, I_{yy} - I_{xy}^2}\right) \int_0^{s_i^*} t_i\, y_i(s_i)\, \mathrm{d}s_i + q_{b_{i-1}}(a_{i-1}) \quad (4.8)$$

Note that t_i is the thickness of the segment i, which is constant within each segment, and $x_i(s_i)$ and $y_i(s_i)$ are the x and y coordinates in the centroidal reference frame, as a function of the variable s_i. The term $q_{b_{i-1}}(a_{i-1})$ corresponds to the basic shear flow evaluated at the end of the previous thin-wall segment, being defined as zero for $q_{b_0}(a_0)$.

The term q_{s0} in Equation (4.4), assuming that the loads are applied on the shear center, producing no twist on the structure, can be calculated from

$$q_{s,0} = -\frac{\oint \frac{q_b}{Gt}\,\mathrm{d}s}{\oint \frac{1}{Gt}\,\mathrm{d}s} \,. \quad (4.9)$$

Based on the Equations (4.8) for q_{b_i} and Equation (4.9) for $q_{s,0}$, and considering that the shear modulus (G) is constant, results

$$q_{s,0} = -\frac{\sum\limits_{i=1}^{4} \int_0^{a_i} \frac{q_{b_i}}{t_i}\mathrm{d}s_i}{\sum\limits_{i=1}^{4} \int_0^{a_i} \frac{1}{t_i}\mathrm{d}s_i} \,. \quad (4.10)$$

The moment (T) produced by the shear flow (q) about a certain point is given by

$$T = \oint pq\mathrm{d}s = \oint p\, q_b \mathrm{d}s + q_{s,0} \oint p\mathrm{d}s \,, \quad (4.11)$$

where p is the arm relative to the shear flow direction. In Equation (4.11), it was used the expression given by Equation (4.4) and the fact that $q_{s,0}$ is constant. Note that, taking the centroid as the reference point for calculating the moment T, results

$$\oint p\,\mathrm{d}s = 2A^* \,, \quad (4.12)$$

where A^* is the area enclosed by the wingbox.

From Figure 4.6, it is possible to write the expression from moment equilibrium,

$$S_y\, \xi_0 - S_x\, \eta_0 = \oint pq\mathrm{d}s = \oint pq_b\mathrm{d}s + \underbrace{2A^* q_{s,0}}_{T_0} \,. \quad (4.13)$$

Therefore, with the Equation (4.13), it is possible to locate the shear center. To determine the coordinate ξ_0, it is necessary to calculate the shear flow distribution caused by applying a shear load $S_y = 1$N and $S_y = 0$ N. To determine η_0, a shear load $S_x = -1$ N and $S_y = 0$ N, must be applied.

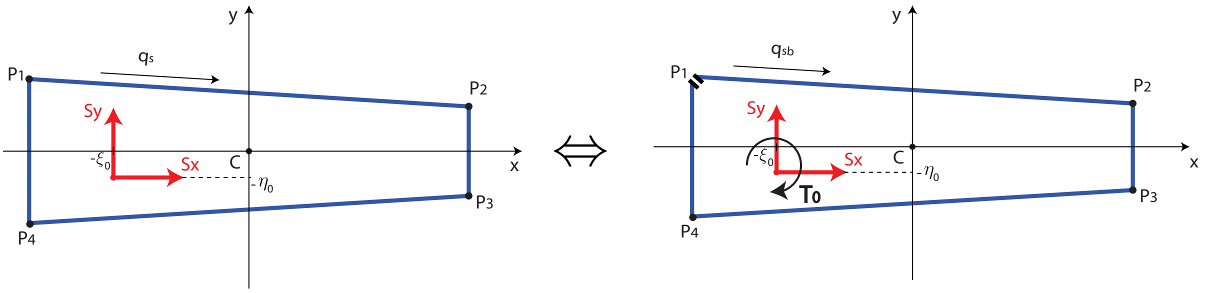


Figure 4.6: Decomposition of a closed wingbox with shear flow distribution q_s into an open wingbox with a shear flow distribution q_{sb} and torque T_0, introducing a constant shear flow.

Axial and Bending Stiffness

The total axial stiffness (EA) is given by the sum of the individual contribution of the homogeneous material thin-wall segments, calculated as

$$EA = E\sum_{i=1}^{n}(a_i t_i)\ . \tag{4.14}$$

Similarly, for the total bending stiffness EI_{xx} and EI_{yy} results

$$EI_{xx} = E\sum_{i=1}^{n}\left[\bar{I}_{xx}^{(i)} + a_i t_i (y_{c_i} - y_{SC})^2\right] \tag{4.15}$$

and

$$EI_{yy} = E\sum_{i=1}^{n}\left[\bar{I}_{yy}^{(i)} + a_i t_i (x_{c_i} - x_{SC})^2\right]\ , \tag{4.16}$$

where (x_{SC}, y_{SC}) are the coordinates of the shear center in the airfoil frame.

Torsional Stiffness

The rate of twist ($\frac{\mathrm{d}\theta}{\mathrm{d}z}$) of a closed section beam subjected to a torque (T) is given by

$$\frac{\mathrm{d}\theta}{\mathrm{d}z} = \frac{T}{4A^2}\oint\frac{ds}{Gt}\ . \tag{4.17}$$

Knowing that the rate of twist in a beam and the torque are related by the torsional stiffness (GJ),

$$T = GJ\frac{d\theta}{dx}\ . \tag{4.18}$$

Therefore, by comparing Equations (4.17) and (4.18), the torsional stiffness of a closed section is given by

$$GJ = \frac{4\,A^2}{\oint\frac{\mathrm{d}s}{Gt}}\ . \tag{4.19}$$

Linear Density

The linear density of the cross section is obtained using

$$\rho_l = \rho \sum_{i=1}^{n} (a_i t_i) \ , \tag{4.20}$$

where ρ is the material density.

4.1.3 Application Example

After implementing the module to calculate the cross-sectional properties, it is necessary to apply it to the desired spanwise sections. As it will be described in Section 4.3, the wing span will be divided in n_{fe} intervals, corresponding to the number of finite elements. The cross-sectional properties must be evaluated for the $n_{fe}+1$ nodes. If the wing structure's internal configuration is constant throughout the span, this evaluation only has to be performed once, since the properties will remain constant throughout the wing. However, if the wing's structural configuration is not constant, whether it is due to specification of different parameters at certain sections of the wing or by considering a tapered configuration, the module has to be run multiple times, in order to obtain the properties for each section. For the case of a tapered wing, the chord (c) at a position (ℓ_{wing}) along the direction of the mid span, will vary according to the following expression:

$$c(\ell_{wing}) = c_{root} \left(1 - \frac{1-\lambda}{b/2}\, \ell_{wing}\right) \tag{4.21}$$

where c_{root} is the root chord, λ is the taper ratio and b is the total wing span.

In Table 4.2, it is presented an application example of the developed function `fequiprops.m` with a NACA0015 airfoil. A reference case of a wingbox with the spars located at 25% and 75%, with a thickness of 20 mm each, is set as an input. The upper and lower segments of the wingbox are 5 mm thick and the reference material properties are chosen to represent an Aluminum structure.

Input variables		Output variables	
Variable	**Value**	**Variable**	**Value**
Chord (c)	1 m	Axial rigidity (EA)	7.15816E+08 N·m^2
Front spar location (x_1/c)	0.25	Flexural rigidity in yy (EI_{yy})	5.15071E+07 N·m^2
Rear spar location (x_2/c)	0.75	Flexural rigidity in xx (EI_{xx})	2.96814E+07 N·m^2
Front spar's thickness (t_A)	20 mm	Torsional rigidity (GJ)	1.81857E+06 N·m^2
Upper segment's thickness (t_B)	5 mm	Shear center's x coord. (SC_x)	0.45984 m
Rear spar's thickness (t_C)	20 mm	Shear center's y coord. (SC_y)	3.99174 E-09 m
Lower segment thickness (t_D)	5 mm	Linear density (ρ_l)	26.72381 kg·m^{-1}
Material's density (ρ)	2800 kg·m^{-3}		
Young's modulus (E)	75 GPa		
Shear modulus (G)	30 GPa		

Table 4.2: Application example of the function `fequiprops.m` with respective input and output variables.

The output solution for axial, flexural and torsional rigidities, as well as the shear center location and the linear density, are presented in the right column. The wingbox configuration with the position of the shear center is represented in Figure 4.7.

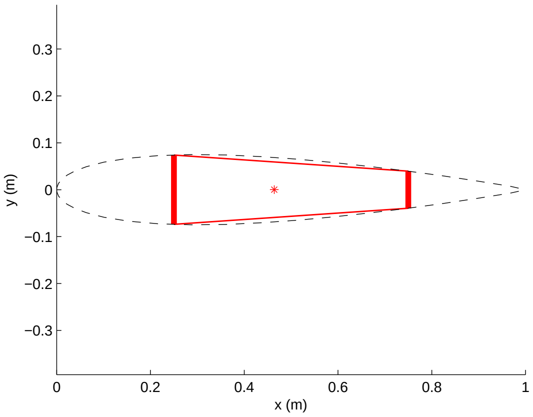


Figure 4.7: Cross-sectional configuration and shear center location in the airfoil reference frame.

4.2 Equivalent Beam Element Properties

The next step in the structural framework is to obtain the equivalent beam element properties based on the cross-sectional properties at the boundaries of each beam finite element. This module is developed to overcome the complexity of spanwise variable cross-section beam configurations, as it is verified, for instance, in tapered beam geometries. Two sets of structural properties are required as inputs and the average properties will be given as an output, as represented in Table 4.3

Input Variables (Boundary values)	**Output Variables (Element average value)**
Axial stiffnesses (EA_1, EA_2)	Average axial stiffness (EA_{av})
Bending stiffnesses in xx $(EI_{xx_1},\, EI_{xx_2})$	Bending stiffness in xx $(EI_{xx_{av}})$
Bending stiffnesses in yy $(EI_{yy1},\, EI_{yy_2})$	Bending stiffness in yy $(EI_{yy_{av}})$
Torsional stiffnesses $(GJ_1,\, GJ_2)$	Torsional stiffness (GJ_{av})
Boundary linear densities (ρ_{l_1}, ρ_{l_2})	Linear density $(\rho_{l_{av}})$

Table 4.3: Equivalent cross-sectional properties module inputs and outputs.

The utilized approach is applied in commercial finite element code and is suggested in ANSYS® [44]. The average area (A_{av}), average moments of inertia (I_{av}) and average polar moment of inertia (J_{av}) are calculated as

$$A_{\text{av}} = \frac{A_1 + \sqrt{A_1\, A_2} + A_2}{3} \ , \tag{4.22a}$$

$$I_{\text{av}} = \frac{I_1 + \sqrt[4]{I_1^3\, I_2} + \sqrt{I_1\, I_2} + \sqrt[4]{I_1\, I_2^3} + I_2}{5} \tag{4.22b}$$

and

$$J_{\text{av}} = \frac{J_1 + \sqrt[4]{J_1^3\, J_2} + \sqrt{J_1\, J_2} + \sqrt[4]{J_1\, J_2^3} + J_2}{5} \ . \tag{4.22c}$$

The indexes 1 and 2, correspond to the properties on the limits of the considered element (nodes I and

J in Figure 4.8).

The average linear density ($\rho_{l_{av}}$) of each finite element will be simply given by the arithmetic mean value at the boundaries.

4.3 Finite Element Model

In order to evaluate the structural response to an aerodynamic loading, a finite element model approach is chosen. As previously introduced in Chapter 3, this technique discretizes a domain into many subsets with a characteristic geometry and certain physical assumptions, defining the type of finite element. The finite elements are all connected through their nodes, and the assembly of all finite elements to portrait the global structural behavior is achieved by superimposing the element equations to obtain a system of equations of the type $[K]\,\mathbf{y} = \mathbf{f}$. The solution of this linear system of equations, presented in a matrix form, is obtained after applying the forces at the nodes and boundary conditions. The force vector $\mathbf{f}$ is generated by mapping the pressure forces calculated by a panel method code into the structural nodes. Note that the original global assembled stiffness matrix $[K]$ is singular, and, in order to obtain a solution, the columns and rows corresponding to the fixed degrees of freedom have to be eliminated. The stiffness matrix $[K]$ can be obtained, for instance, from energy considerations with the use of constitutive equations. In this module, the previously computed average properties of the finite elements and an external force vector $\mathbf{f}$ are used as inputs and an output vector $\mathbf{y}$ is generated as a solution, containing the information of the structural nodal deformation.

4.3.1 Constitutive Equations

According to Reddy [45], the constitutive equations correspond to relations that characterize an individual material and its reaction to applied loads. The Generalized Hook's law, corresponds to a linear constitutive model for infinitesimal deformation and relates stresses and strains. If stress components (σ_{ij}) are assumed to be linear functions of the strain components (ϵ_{kl}), then the constitutive equation for a three-dimensional body is given by

$$\sigma_{ij} = C_{ijkl}\,\epsilon_{kl} + \sigma^0_{ij}\ , \tag{4.23}$$

where C_{ijkl} is the fourth-order stiffness tensor and σ^0_{ij} is the stress components of a pre-stressed body. Since the principle of conservation of angular momentum requires the stress tensor to be symmetric ($\sigma_{ij} = \sigma_{ji}$) and being the strain tensor symmetric by definition ($\epsilon_{ij} = \epsilon_{ji}$), the number of independent material stiffness components is reduced to 6x6. Thus, the stiffness tensor can be represented in a matrix form $[D]$, and the constitutive tensor equation (4.23), can be rewritten in a matrix form as

$$\boldsymbol{\sigma} = [D]\ \boldsymbol{\epsilon}\ , \tag{4.24}$$

where $\boldsymbol{\sigma} = [\sigma_{xx}\ \sigma_{yy}\ \sigma_{zz}\ \tau_{yz}\ \tau_{xz}\ \tau_{xy}]^T$ and $\boldsymbol{\epsilon} = [\epsilon_{xx}\ \epsilon_{yy}\ \epsilon_{zz}\ 2\epsilon_{yz}\ 2\epsilon_{xz}\ 2\epsilon_{xy}]^T$.

For isotropic materials, whose properties are constant regardless the orientation of the system of axes to which the components of stress are referred [46], the matrix $[D]$ is given by

$$[D]^{-1} = \begin{bmatrix} \frac{1}{E} & -\frac{\nu}{E} & -\frac{\nu}{E} & 0 & 0 & 0 \\ -\frac{\nu}{E} & \frac{1}{E} & -\frac{\nu}{E} & 0 & 0 & 0 \\ -\frac{\nu}{E} & -\frac{\nu}{E} & \frac{1}{E} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G} \end{bmatrix} , \tag{4.25}$$

where E is the Young's modulus, ν is the Poisson's ratio and G is the shear modulus of the considered material. The shear modulus G is computed from the isotropic relation

$$G = \frac{E}{2(1+\nu)} . \tag{4.26}$$

4.3.2 3D Finite Element Model Definition

The first step is to decide which finite element type shall be used to represent the wing's structure. Since the main goal of the original tool developed by Almeida [11] was to rapidly generate concepts in preliminary design stages, assessing the most important structural responses, the low fidelity model was kept the same.

The so-called 3D beam finite element (Figure 4.8) corresponds to a unidimensional finite element model, delimited by two nodes with six degrees of freedom each: three displacement components (u_x, u_y, u_z) and three rotation components (θ_x, θ_y, θ_z). This general beam element results from the conjugation of a tension bar element, two bending beam elements and a torsion bar element. The formulation used in this framework is based on the work of Cook et al. [47] and Ottosen and Petersson [48], and following assumptions are made:

1. Plane sections perpendicular to the beam axis remain plane and normal to the beam axis after deformation (Euler-Bernoulli beam theory);
2. Axial deformation, bending (both in xz-plane and xy- plane) and twist are considered to be four decoupled problems.

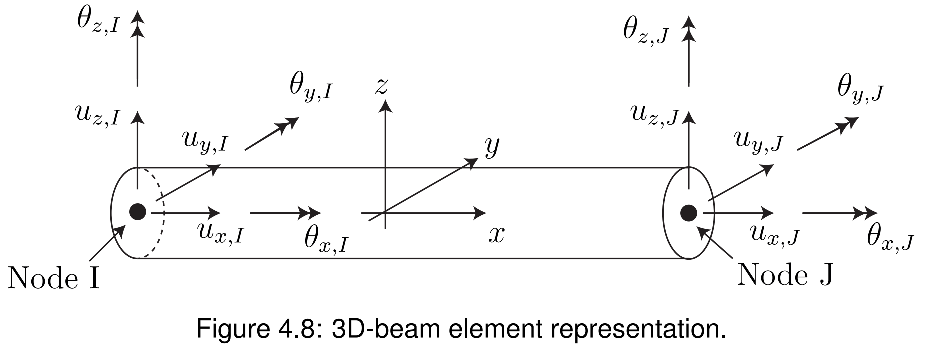


Figure 4.8: 3D-beam element representation.

4.3.3 Strong and Weak formulations

The strong equations that describe the physics of the axial, bending and torsion problems are summarized on the left hand side of Table 4.4. The variables EA, EI and GJ represent axial, bending and torsional rigidity, respectively, and the variables $b(x)$, $q(x)$ and $m(x)$ correspond to distributed axial, bending and twisting loads.

Problem	Strong formulation	Weak formulation
Axial	$\frac{d}{dx}\left(EA\frac{du_x}{dx}\right)+b(x)=0$	$[vN_1]_0^L-\int_0^L\left(\frac{dv}{dx}AE\frac{du_x}{dx}-vb\right)dx=0$
Bending (x-z)	$\frac{d^2}{dx^2}\left(EI_{yy}\frac{d^2u_z}{dx^2}\right)-q_1(x)=0$	$[vV_1]_0^L-\left[\frac{dv}{dx}M_1\right]_0^L-\int_0^L\left(\frac{d^2v}{dx^2}EI_{yy}\frac{d^2u_z}{dx^2}-vq_1\right)dx=0$
Bending (x-y)	$\frac{d^2}{dx^2}\left(EI_{zz}\frac{d^2u_y}{dx^2}\right)-q_2(x)=0$	$[vV_2]_0^L-\left[\frac{dv}{dx}M_2\right]_0^L-\int_0^L\left(\frac{d^2v}{dx^2}EI_{zz}\frac{d^2u_y}{dx^2}-vq_2\right)dx=0$
Torsion	$\frac{d}{dx}\left(GJ\frac{d\theta_x}{dx}\right)+m(x)=0$	$[vT_1]_0^L-\int_0^L\left(\frac{dv}{dx}GJ\frac{d\theta_x}{dx}-vm\right)dx=0$

Table 4.4: Strong and weak formulations for the axial, bending and torsion problems.

On the right hand side column of Table 4.4, it is presented weak formulation of the four problems. Each weak formulation problem contains an integral term, with the differentiation order evenly distributed between the shape function and the independent variable, and the boundary condition terms, applied at the nodal locations of each finite element (in this case at 0 and L). In the axial and torsion problems, two essential boundary conditions (one for each node) need to be applied, as for each bending problem, four essential boundary conditions (two for each node) are required.

In axial problems, the terms v are associated with the essential boundary conditions (specification of the axial displacements) and the terms N correspond to the natural boundary conditions (corresponding to the axial forces).

In bending problems, the terms v and $\frac{dv}{dx}$ are related to the essential boundary conditions (specification of displacements and rotations), and the terms V and M correspond to the natural boundary conditions (specification of shear loads and moments).

In torsional deformation problems, the terms v are associated with the essential boundary conditions (rotation angles) and the term T corresponds to the natural boundary conditions (torsional moments).

4.3.4 Domain Discretization

Once the finite element is defined, it is possible to proceed with the wing discretization. The number of finite elements (n_{fe}) to represent the wing's structure is chosen to be dependent on the number of spanwise aerodynamic panels (n_{ap}). For each spanwise set of aerodynamic panels, there will be 2 finite elements. The main reason to do so is to simplify mapping the aerodynamic forces into the structural nodes, since the collocation points will be in the same cross-sectional plane as some nodes. An example is presented in Figure 4.9.

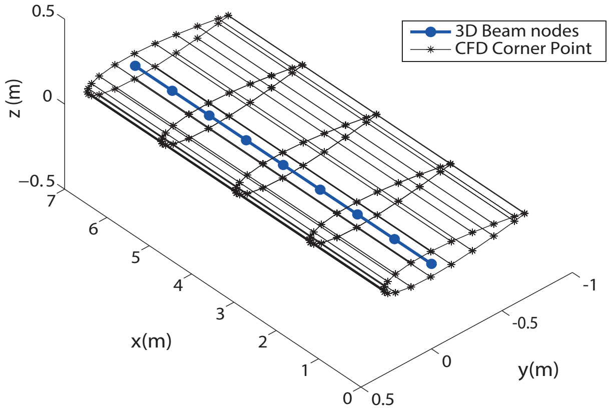


Figure 4.9: Wing discretization example with $n_{ap} = 4$ (corresponding to $n_{fe} = 8$) in the global reference frame.

4.3.5 Shape Functions

The displacement and rotation fields within each finite element can be modeled with the use of shape functions, based on interpolation of the correspondent nodal displacement values. These shape functions are essential to formulate the stiffness matrix, as presented in Chapter 3.

The local relative coordinate s is defined along the $\bar{x}$-direction within the finite element as

$$s = \frac{2\bar{x} - L}{L} , \tag{4.27}$$

where L is the length of the finite element. Note that $\bar{x}$ represents a local absolute coordinate along the finite element length, as represented in Figure 4.10.

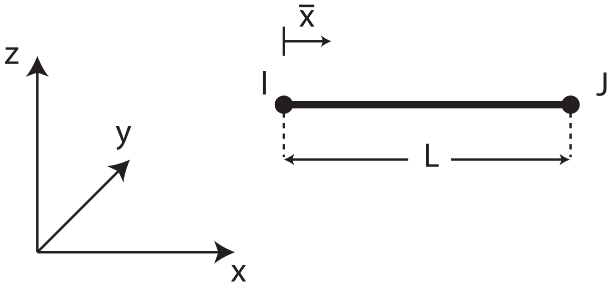


Figure 4.10: 3D-beam finite element in a global reference frame.

The shape functions used in the 3D beam finite element model [49] are represented in Table 4.5. The axial displacement (u_x) and torsion (θ_x) were modeled with linear shape functions (N_1 and N_2). For the transverse displacements' case (u_y and u_z), cubic shape functions along x were utilized (N_3, N_4, N_5 and N_6).

The full displacement field for a 3D beam element can be determined as

$$\mathbf{u} = [N]\,\mathbf{y} , \tag{4.28}$$

$N_1 = \frac{1}{2}(1-s)$	$N_3 = \frac{1}{2}\left[1 - \frac{s}{2}\left(3 - s^2\right)\right]$	$N_5 = \frac{1}{2}\left[1 + \frac{s}{2}\left(3 - s^2\right)\right]$
$N_2 = \frac{1}{2}(1+s)$	$N_4 = -\frac{L}{8}\left(1 - s^2\right)(1-s)$	$N_6 = \frac{L}{8}\left(1 - s^2\right)(1+s)$

Table 4.5: Shape functions.

where $\mathbf{u}$ is the vector of displacements, $\mathbf{y}$ is the vector of nodal displacements and $[N]$ is the matrix containing the shape functions. Equation (4.28) corresponds to the compact form of

$$\begin{Bmatrix} u_x(x) \\ u_y(x) \\ u_z(x) \\ \theta_x(x) \end{Bmatrix} = \begin{bmatrix} N_1 & 0 & 0 & 0 & 0 & 0 & N_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & N_3 & 0 & 0 & 0 & -N_4 & 0 & N_5 & 0 & 0 & 0 & -N_6 \\ 0 & 0 & N_3 & 0 & N_4 & 0 & 0 & 0 & N_5 & 0 & N_6 & 0 \\ 0 & 0 & 0 & N_1 & 0 & 0 & 0 & 0 & 0 & N_2 & 0 & 0 \end{bmatrix} \mathbf{y} \quad \text{(4.29a)}$$

$$\begin{Bmatrix} \theta_y \\ \theta_z \end{Bmatrix} = \begin{Bmatrix} \frac{\mathrm{d}u_z}{\mathrm{d}z} \\ \frac{\mathrm{d}u_y}{\mathrm{d}x} \end{Bmatrix} . \quad \text{(4.29b)}$$

where $\mathbf{y} = [u_{x,I}\ u_{y,I}\ u_{z,I}\ \theta_{x,I}\ \theta_{y,I}\ \theta_{z,I}\ u_{x,J}\ u_{y,J}\ u_{z,J}\ \theta_{x,J}\ \theta_{y,J}\ \theta_{z,J}\ ]^T$, with the indexes I and J representing two consecutive nodes delimiting a certain finite element, as illustrated in Figure 4.8.

4.3.6 Element Stiffness Matrix Formulation

The derivation of the stiffness matrix can be performed by using the concept of potential energy [11]. For a general finite element, the potential energy can be defined as

$$V = \frac{1}{2}\int_{\Omega} \boldsymbol{\sigma}^T \boldsymbol{\epsilon}\, \mathrm{d}\Omega - \int_{\Omega} \mathbf{f_v}^T \mathbf{u}\, \mathrm{d}\Omega - \int_{S} \mathbf{f_s}^T \mathbf{u}\, \mathrm{d}S\, , \quad \text{(4.30)}$$

where Ω corresponds to the element domain, S to the element's surface, $\mathbf{f_v}$ to the volume forces and $\mathbf{f_s}$ to the surface forces. Equation (4.30) can be rewritten using the constitutive, strain-displacement and shape function matrices as

$$V = \frac{1}{2}\mathbf{y}^T \int_{\Omega} [B]^T [D] [B]\, \mathrm{d}\Omega - \mathbf{y}^T \int_{\Omega} [N]^T \mathbf{f_v}^T \mathrm{d}\Omega - \mathbf{y}^T \int_{S} [N]^T \mathbf{f_s}^T \mathrm{d}S\, . \quad \text{(4.31)}$$

In this case, the stress vector, the elasticity matrix and the strain vector are slightly modified, to represent the axial force, bending moment in both planes and the torsional moment, as

$$\begin{Bmatrix} N_x \\ M_z \\ M_y \\ M_x \end{Bmatrix} = \begin{bmatrix} EA & 0 & 0 & 0 \\ 0 & EI_{zz} & 0 & 0 \\ 0 & 0 & EI_{yy} & 0 \\ 0 & 0 & 0 & GJ \end{bmatrix} \begin{Bmatrix} \epsilon_x(x) \\ \kappa_z(x) \\ \kappa_y(x) \\ \theta_(x) \end{Bmatrix} = [D]\, \boldsymbol{\epsilon}\, . \quad \text{(4.32)}$$

Note that the modified strain vector includes the axial strain, curvatures and twist angle. Thus, the modified strain-displacement relation is given by

$$\boldsymbol{\epsilon} = [B]\ \mathbf{y}\ , \tag{4.33}$$

where the matrix $[B]$ is calculated as

$$[B] = \begin{bmatrix} \dfrac{\partial}{\partial x} & 0 & 0 & 0 \\ 0 & \dfrac{\partial^2}{\partial x^2} & 0 & 0 \\ 0 & 0 & \dfrac{\partial^2}{\partial x^2} & 0 \\ 0 & 0 & 0 & \dfrac{\partial}{\partial x^2} \end{bmatrix} [N]\ . \tag{4.34}$$

The stiffness matrix of the element is identified from Equation (4.31) and is computed as

$$[K_\ell^e] = \int_\Omega [B]^T [D][B] d\Omega\ . \tag{4.35}$$

The resultant local symmetric stiffness element matrix $[K_\ell^e]$ is, therefore, given by

$$[K_\ell^e] = \begin{bmatrix} \frac{AE}{L} & 0 & 0 & 0 & 0 & 0 & -\frac{AE}{L} & 0 & 0 & 0 & 0 & 0 \\ & \frac{12EI_{zz}}{L^3} & 0 & 0 & 0 & \frac{6EI_{zz}}{L^2} & 0 & -\frac{12EI_{zz}}{L^3} & 0 & 0 & 0 & \frac{6EI_{zz}}{L^2} \\ & & \frac{12EI_{yy}}{L^3} & 0 & -\frac{6EI_{yy}}{L^2} & 0 & 0 & 0 & -\frac{12EI_{yy}}{L^3} & 0 & -\frac{6EI_{yy}}{L^2} & 0 \\ & & & \frac{GJ}{L} & 0 & 0 & 0 & 0 & 0 & -\frac{GJ}{L} & 0 & 0 \\ & & & & \frac{4EI_{yy}}{L} & 0 & 0 & 0 & \frac{6EI_{yy}}{L^2} & 0 & \frac{2EI_{yy}}{L} & 0 \\ & & & & & \frac{4EI_{zz}}{L} & 0 & -\frac{6EI_{zz}}{L^2} & 0 & 0 & 0 & \frac{2EI_{zz}}{L} \\ & & & & & & \frac{AE}{L} & 0 & 0 & 0 & 0 & 0 \\ & & & & & & & \frac{12EI_{zz}}{L^3} & 0 & 0 & 0 & -\frac{6EI_{zz}}{L^2} \\ & & & & & & & & \frac{12EI_{yy}}{L^3} & 0 & \frac{6EI_{yy}}{L^2} & 0 \\ & & & & & & & & & \frac{GJ}{L} & 0 & 0 \\ & & & & & & & & & & \frac{4EI_{yy}}{L} & 0 \\ & & & & & & & & & & & \frac{4EI_{zz}}{L} \end{bmatrix} . \tag{4.36}$$

In general, the local reference frame of the finite element is different from the global frame and it may be useful to represent the results in the global frame. The relation between the element coordinates and the global coordinates is given by

$$\mathbf{y}_\ell = [T^e]\, \mathbf{y}\ , \tag{4.37}$$

where $\mathbf{y}_\ell$ corresponds to the vector of displacements in the element Cartesian coordinates and $\mathbf{y}$ represents the vector of displacements in the global Cartesian coordinates. The elementary matrix rotation $[T^e]$ is defined as

$$[T^e] = \begin{bmatrix} [T] & 0 & 0 & 0 \\ 0 & [T] & 0 & 0 \\ 0 & 0 & [T] & 0 \\ 0 & 0 & 0 & [T] \end{bmatrix} , \tag{4.38}$$

where $[T]$ is the submatrix rotation defined as

$$[T] = \begin{bmatrix} \cos(\Lambda)\cos(\Gamma) & \sin(\Lambda)\cos(\Gamma) & \sin(\Gamma) \\ -\cos(\Lambda)\sin(\Gamma)\sin(\theta) - \sin(\Lambda)\cos(\theta) & -\sin(\Lambda)\sin(\Gamma)\sin(\theta) + \cos(\Lambda)\cos(\theta) & \sin(\theta)\cos(\Gamma) \\ -\cos(\Lambda)\sin(\Gamma)\cos(\theta) + \sin(\Lambda\sin(\theta) & -\sin(\Lambda)\sin(\Gamma)\cos(\theta) - \cos(\Lambda)\sin(\theta)) & \cos(\theta)\cos(\Gamma) \end{bmatrix} . \tag{4.39}$$

The angles Λ, Γ and θ correspond to the Euler angles used to describe the element rotation. According to the global reference frame (Figure 4.9), Λ corresponds to a ”sweep” rotation, Γ represents a subsequent ”dihedral” rotation and θ represents a ”twist” rotation, as illustrated in Figure 4.11.

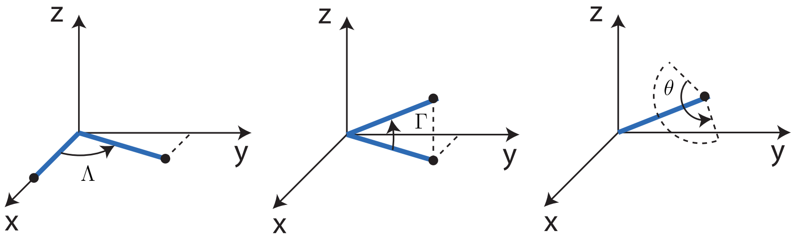


Figure 4.11: Sequential rotations representation.

Therefore, the element stiffness matrix in the global reference frame ($[K^e]$) may be calculated as follows:

$$[K^e] = [T_R]^T \, [K^e_\ell] \, [T_R] \ . \tag{4.40}$$

After evaluating the element stiffness matrix of each finite element, it is necessary to assemble them into a global stiffness matrix whose size corresponds to the total number of degrees of freedom in the system (6 times the number of nodes).

4.3.7 Force Vector Determination

After defining the stiffness matrix in the global reference frame, it is necessary to determine the global force vector containing the aerodynamic load information to be applied at the nodes of the beam finite elements. The method used by Almeida [11] was kept the same, but corrections in the external aerodynamic moments calculation were introduced. Furthermore, the aerodynamic code was detached from the framework source code and adapted into a function so that it could be evaluated separately without the need of running the entire framework.

Aerodynamic Model and Forces Computation

The aerodynamic analysis is performed by using a panel method code developed in MATLAB® by Cardeira [50]. This code uses an incompressible potential flow model, meaning that the flow is assumed to be incompressible, inviscid, irrotational and isentropic. Consequently, the model is not able to capture the physics of shock waves, shear stresses and heat transfer problems. Therefore, only analyses at subsonic regimes will be considered. Even though the panel method code has some disadvantages in terms of representing in detail the full flow physics, one of its main advantages is that the computational cost is considerably lower when compared to other high-fidelity methods, such as the Reynolds-averaged Navier–Stokes equations (RANS), being a useful tool for fast results in preliminary aero-structural analyses.

The steps in solving the numerical aerodynamic problem are the following:

1. Cardeira [50] has chosen to use first order source and doublet singularity elements to solve the Laplace equation and to apply Dirichlet boundary conditions, which were proven to lead to accurate results for aircraft wings without compromising computational time;

2. The wing geometry has to be discretized by choosing the number of airfoil and spanwise segment divisions, defining a grid of aerodynamic panels;

3. The collocation points are positioned in the center of each panel, as represented in Figure 4.12, and the influence coefficients are computed for each collocation point;

4. The singularity strengths and the velocity field can be obtained by solving a set of linear equations and the pressures may be post-processed, using the Bernoulli equation. Aerodynamic loads and coefficients are obtained by summing the contributions of each panel.

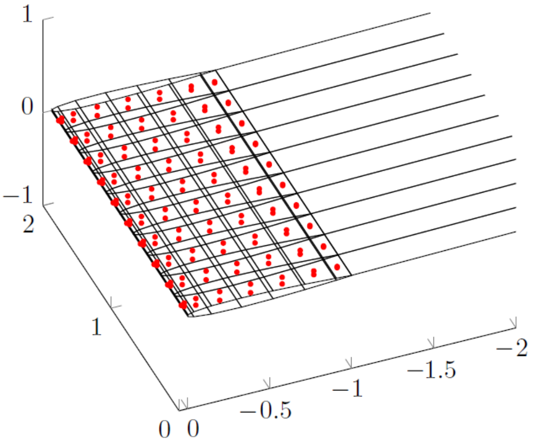


Figure 4.12: Example of an aircraft wing and wake discretization, with 16x10 panels and the respective collocation points represented in red. [11]

Cardeira's [50] aerodynamic code has a steady and unsteady capabilities. In the scope of this thesis, only steady computations are required to perform static aero-structural analyses.

Fluid-Structure Interface

Almeida [11] developed, in his thesis, a fluid-structure coupling interface between the aerodynamic panel method and the 3D beam structural model. In the aerodynamic model, the pressures are assumed to be constant within each panel, and the resultant forces applied at each panel's collocation point may be estimated by multiplying the area of each panel by the dynamic pressure $0.5\rho V_\infty^2$, where ρ represents the density of the fluid and V_∞ the free stream velocity. Figure 4.13 expresses a cross-sectional segment of the wing, where the blue circle represents a structural node (in blue) and the red dots correspond to the collocation points of the aerodynamic panels, which are all contained in the same cross-sectional plane. The force vectors are merely representative and do not portrait their real orientation nor magnitude. To transfer the forces in the panels to the structural nodes, each collocation point force ($\mathbf{f_i}$) is decomposed

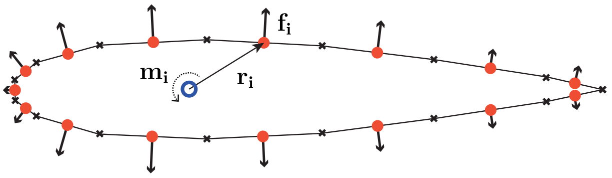


Figure 4.13: Fluid forces applied at each collocation point. [11]

into the respective global reference frame components, and the equivalent moments relative to the structural node are calculated using an auxiliary orientation vector $(\mathbf{r_i})$ defined from the structural point to the considered collocation point as

$$\mathbf{m_i} = \mathbf{r_i} \times \mathbf{f_i} \; . \tag{4.41}$$

Almeida [11] validated his code for a bending case. However, some errors were identified when using the resultant torsional moment information. This was not evident in the bending motion validation since both cases were considered to be decoupled. These errors were identified in the implementation of Equation 4.41. Since the cross product definition was not being correctly applied, some of the individual contribution of each collocation point to the total torsional moment presented incorrect values in sign, which were adjusted in this new version of the framework.

Aerodynamic function

The aerodynamic code was separated from the main aerostructural framework code, and adapted as a MATLAB® function called `Faero.m`. The required input parameters are a text file containing the coordinates of the airfoil, the spanwise length of the wing, the number of spanwise panels, the sweep and dihedral angles, the root chord, the taper ratio, the twist distribution, the freestream velocity, the angle of attack, the density of the fluid and the shear center node coordinates. The output of this function a load vector $\mathbf{f}$ with the components directly represented in the global framework.

4.3.8 Bondary Conditions Application and Global System of Equations Solution

In order to obtain the structural solution of the problem, the non-singular reduced stiffness matrix has to be determined. To perform so, it is necessary to apply the proper boundary conditions to the problem. In this case, the wing is modeled as a clamped beam, meaning that the first six degrees of freedom are fixed corresponding to the node of the first element, that is located at the wing root section. Thus, the reduced stiffness matrix is obtained by eliminating the first six rows and columns from the global stiffness matrix. The corresponding first six entries of the external force vector are removed, as well.

Therefore, all the conditions required to obtain the static structural solution of the problem $(\mathbf{y})$ are fulfilled, which results from the solution of the matrix Equation 3.31.

Chapter 5

Parametric Study

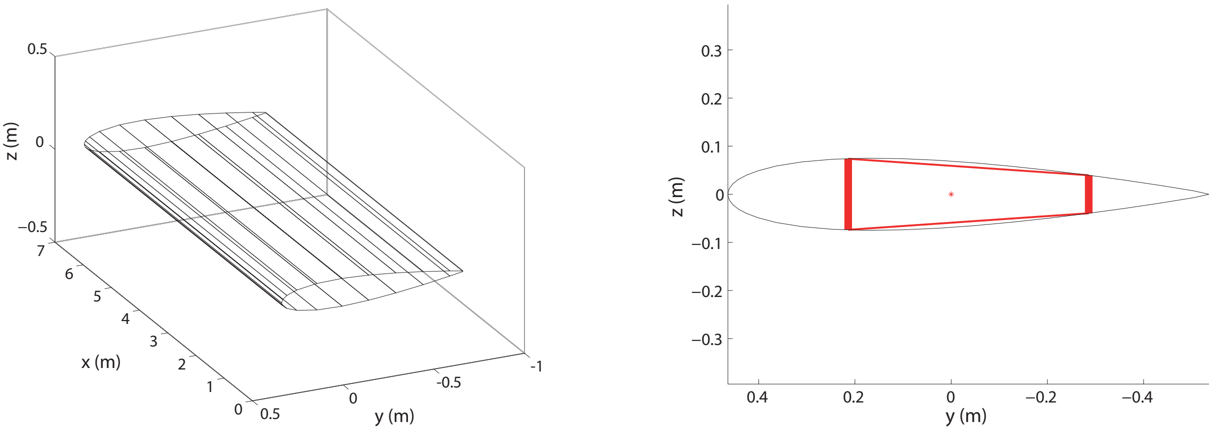
In this chapter, a baseline wing configuration is presented and a parametric study is performed. The main goal is to verify how output functions are influenced by changing some parameters from the baseline configuration. Only internal geometric parameters and material properties will be considered in these studies. The external geometric parameters will not be addressed in the scope of this thesis since these also influence the aerodynamic loading. Therefore, the external forces are calculated only once with the aerodynamic function `Faero.m`. The resultant forces will be constant throughout the analyses, however the torsional moment may vary, depending on the specific case, due to a possible change in the shear center location. This problem is simply overcome by recalculating the equivalent moment relative to the new shear center position. The other external moment components are approximately zero. At the end of the chapter, a table of input variables and output functions will be presented, showing the qualitative behavior of the former relative to the latter, in the surroundings of the baseline configuration.

5.1 Wing Baseline Configuration

The baseline wing is selected as a straight, untapered and unswept configuration. The wing has a zero dihedral angle, a NACA0015 symmetric airfoil and is mounted with a 2º angle of attack relative to the free air stream. For the aerodynamic forces calculation, the preset grid of 40 x 20 panels (chord and spanwise direction, respectively) were used. The external and internal wing geometries, the material properties and the flight condition are summarized in Table 5.1. The graphical representation of external and internal wing configurations are presented in Figures 5.1a and 5.1b, respectively.

External Geometric Parameters		**Internal Geometric Parameters**	
Half-wing spanwise length (L)	7 m	Front spar location (x_1/c)	0.25
Root chord (c_{root})	1 m	Rear spar location (x_2/c)	0.75
Angle of attack (α)	2º	Front spar thickness (t_A)	20 mm
Sweep (Λ)	0º	Upper skin thickness (t_B)	5 mm
Dihedral (Γ)	0º	Rear spar thickness (t_C)	20 mm
Taper ratio (λ)	1	Lower skin thickness (t_D)	5 mm
Twist at the tip (ϑ_{tip})	0º		
Material Properties		**Flight Condition**	
Density (ρ)	2800 kg/m^3	Altitude (h)	1370 m
Young's modulus (E)	75 GPa	Air density (ρ_{air})	1 kg/m^3
Shear modulus (G)	30 GPa	Freestream velocity (V_∞)	75 m/s

Table 5.1: Reference wing baseline parameters.



(a) External geometry.

(b) Internal geometry.

Figure 5.1: Reference wing configuration.

5.2 Functions of Interest

After the wing baseline definition, it is necessary to identify the functions of interest correspondent to the structural discipline. This is performed by running the structural framework and analyzing the change in output values. The identified interest or performance functions will be used as objective and constraints in the optimization problem. Three types of output variables will be covered:

1. Variables directly evaluated from the geometric configuration, such as the wing mass;
2. Variables obtained from the finite element solution, such as the vertical displacement and rotation of the wing tip;
3. Variables resultant from post-processing the finite element solution, such as forces, moments and stresses applied at the wing root.

In fact, forces, moments and stresses applied at the root can be calculated without solving the finite element problem. However, since the finite element solution is available, the previous quantities can be determined with a small additional implementation cost.

The first interest function in this parametric study is the total wing mass (m_{wing}). It is assumed to be equivalent to the wingbox mass, given by

$$m_{wing} = \sum_{i=1}^{n_{fe}} \rho_{l_i} l_i \ , \tag{5.1}$$

where ρ_i is the linear density and l_i is the length of the finite element i. The mass of the wing is directly proportional to the wing's weight, which is often desired to be minimized in aircraft design, to achieve better performance.

The second interest function corresponds to the wing tip vertical displacement (δ_{tip}) and it is directly obtained from the finite element solution vector $\mathbf{y}$ since

$$\delta_{tip} = u_{z,n_{fe}+1} \ , \tag{5.2}$$

where $u_{z,n_{fe}+1}$ corresponds to the u_z displacement component of the last node $(n_{fe}+1)$. The vertical tip displacement may be used as a constraint function in an optimization problem, for instance, to assure that the tip displacement does not surpass a predefined maximum allowable value.

The tip rotation (θ_{tip}) is also obtained directly from the finite element solution vector $\mathbf{y}$ as

$$\theta_{tip} = \theta_{x,n_{fe}+1} \ , \tag{5.3}$$

where $\theta_{x,n_{fe}+1}$ corresponds to the θ_x rotation component of the last node $(n_{fe}+1)$. Tip rotation may also be used as a constraint function in optimization problems to guarantee that it does not exceed a certain value, for instance, an effective angle of attack that leads to tip stall.

The values of forces and moments applied at the wing root are calculated according to Equation (4.32) and N_x, M_x, M_y and M_z will be represented in the local reference frame of the first finite element, which coincide with the global reference frame for the baseline wing. According to the adopted 3D-Beam element from ANSYS [44], the maximum stress is computed as the direct axial stress plus the absolute values of both axial stress components originated from bending,

$$\sigma_x = \frac{N_x}{A} + \frac{M_z d_y}{I_{zz}} + \frac{M_y d_z}{I_{yy}} \ , \tag{5.4}$$

where d_y corresponds to the distance from the shear center to the considered spar, and d_z corresponds to half the length of that spar. Since there are two spars in this wingbox model, Equation (5.4) has to be evaluated twice for a corner point of each spar. Afterwards, it must be checked which of the two cases presents the highest stress value to be defined as the maximum axial stress (σ_{max}).

5.3 Internal Parameters Influence

In this section, a parametric analysis of the internal structure is performed. The influence of the spars' location along the airfoil and the thicknesses of the wingbox in the output functions are addressed. With reference to Figure 5.2, the wingbox segment A and C corresponds to the front and rear spars, respectively, and the segments B and D will be labeled as upper and lower skins, even though they do not truly represent the whole wing skin.

5.3.1 Spars Location

The chordwise position of the spars is the first identified parameter that influences the output functions. It is defined as the ratio of the spar's distance from the leading edge (x) of the airfoil to the airfoil's chord length (c). By varying the chordwise location a spar, not only the length of the considered spar changes, but also the length of the upper and lower wingbox sections (B and D) varies, according to the adopted wingbox model. Furthermore, a variation in the position of the spars causes a shift in the shear center location, as observed in Figure 5.2.

In Figure 5.3a, it is possible to observe that the total mass of the wing decreases when moving the front spar (A) in the rear direction. This occurs because an increase in x_1/c will lead to a reduction of the upper and lower wingbox segment's lengths. Thus, the linear density of the cross-section will be lower

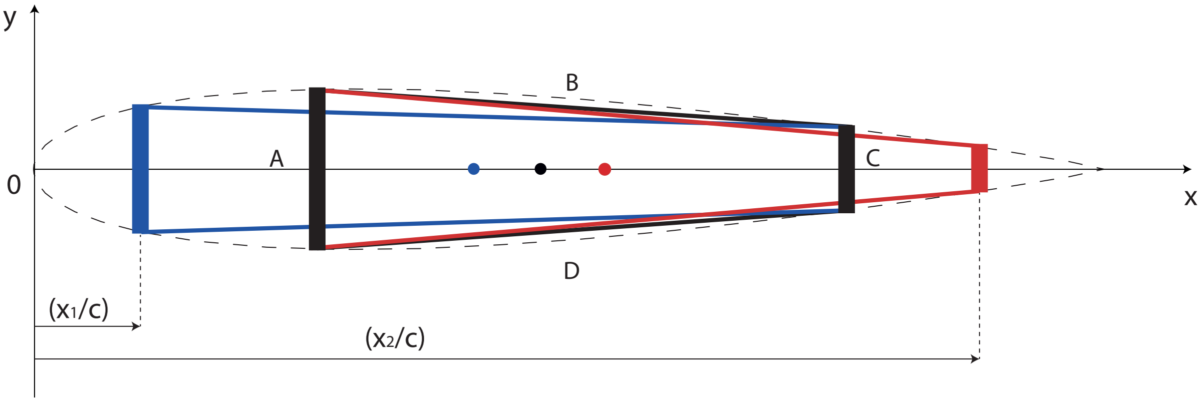
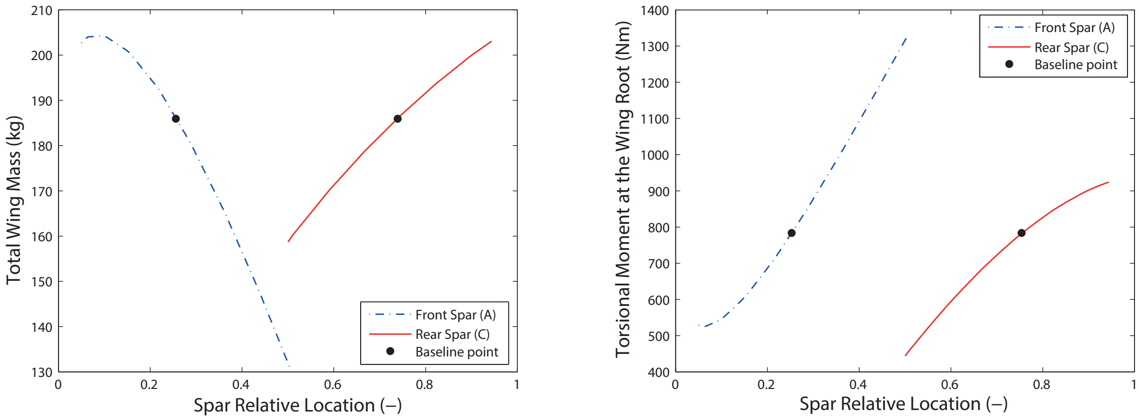


Figure 5.2: Wingbox configurations by changing the front spar A (blue) and the rear spar C (red) positions from the reference case (black). The dots represent the respective shear center locations, in the airfoil reference frame.

than in the reference case, since there is less material per cross-section segment, resulting in an overall weight reduction. The reverse happens when moving the rear spar (C) in the rearwards direction. Even though the rear spar's length is decreasing in this case, the upper and lower skins, will become longer, increasing the cross-sectional linear density, and, consequently, the total wing mass.



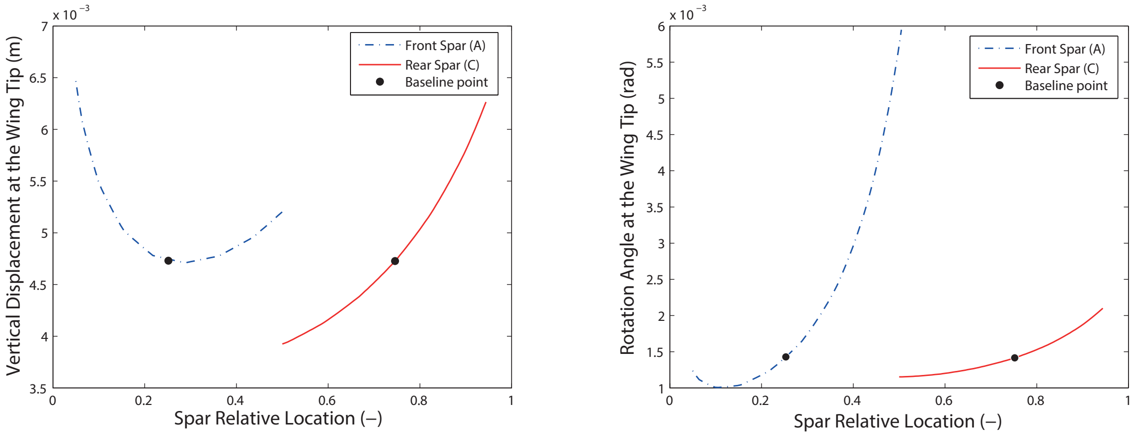
(a) Total wing mass as function of the relative chordwise location of the spars.

(b) Wing root torsional moment as function of the chordwise location of the spars.

Figure 5.3: Influence of the spars locations in the wing mass and the root torsional moment.

The total torsional moment M_x applied at the root of the wing also varies with the variation of the spars' location, as observed in Figure 5.3b. By moving either the front spar or the rear spar in the rearwards direction, the absolute value of the torsional moment increases, due to the rearwards shift in the shear center location.

In Figure 5.4a it is possible to observe the variation of the tip displacement as a function of the spars locations. By moving the front spar (A) rearwards up to 0.3, the vertical tip displacement decreases. For a front spar relative location of 0.3, the vertical tip displacement is minimum. This point corresponds to the maximum thickness of the airfoil, meaning that the front spar will reach the highest length possible leading to a higher value of the bending stiffness (EI_{yy} in the global reference frame). From that point on, the vertical tip displacement starts increasing, mainly due to the reduction of the front spar's length. The rear spar relative location also presents a significant influence in the tip vertical displacement. The closer the rear spar to the maximum airfoil thickness, the lower the vertical tip displacement.



(a) Tip vertical displacement as function of the relative location of the spars

(b) Tip rotation as function of the relative location of the spars.

Figure 5.4: Influence of the spars location in the tip displacement and rotation.

For the tip rotation case, the further rearwards the spars, the larger the tip rotation angle. The relative spars' locations influence both the magnitude of the applied torsional moment, due to the shift of the shear center, and the torsional rigidity (GJ). A combination of these two effects lead to the torsional response represented in Figure 5.4b.

In Figure 5.5, the maximum stress is represented as a function of the spars' locations. For the front spar location, it is possible to observe that the stress increases by moving the front spar rearwards, up to an x_1/c approximately equal to 0.36. From that point on the maximum stress decreases with the increase of x_1/c. By moving the rear spar further rearwards, the maximum stress increases. Note that since the axial load contribution (N_x) is zero and the bending moments M_y and M_z remain constant, the variation of the maximum stress results from a conjugation of the variation effects on d_y, d_z, I_{yy} and I_{zz}, according to Equation (5.4).

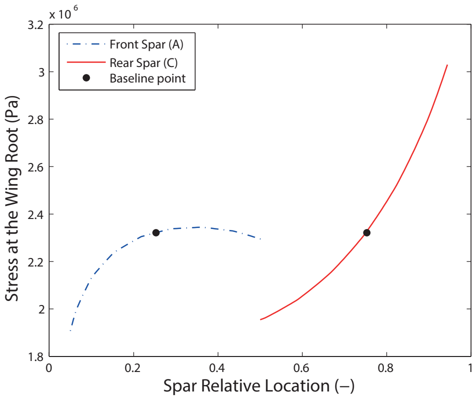


Figure 5.5: Influence of the spars location in the wing root maximum stress.

5.3.2 Spar and Skin Thickness

Besides the location of the spars relative to the chord length, other internal parameters, such as the thicknesses of the wingbox segments, influence the interest functions. In Figure 5.5 it is presented a partition of the wingbox in four segments for thickness specification. The front spar thickness (t_A) and the rear spars thickness (t_B) are independent from each other. The thicknesses of the segments B and D are considered to be equal and are changed simultaneously in the parametric studies, to guarantee symmetry along the airfoil x-axis. With this assumption, the product of inertia is zero relative to the represented frame is zero and both bending cases can be decoupled.

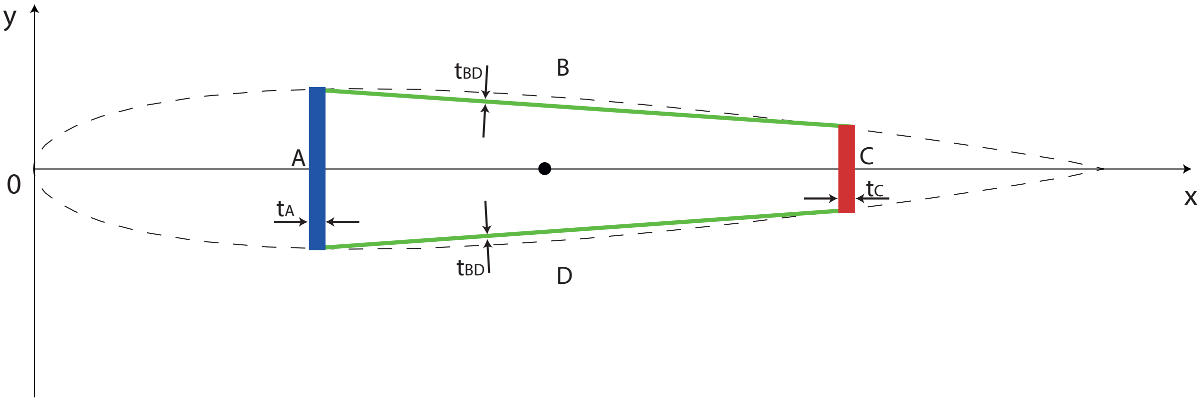
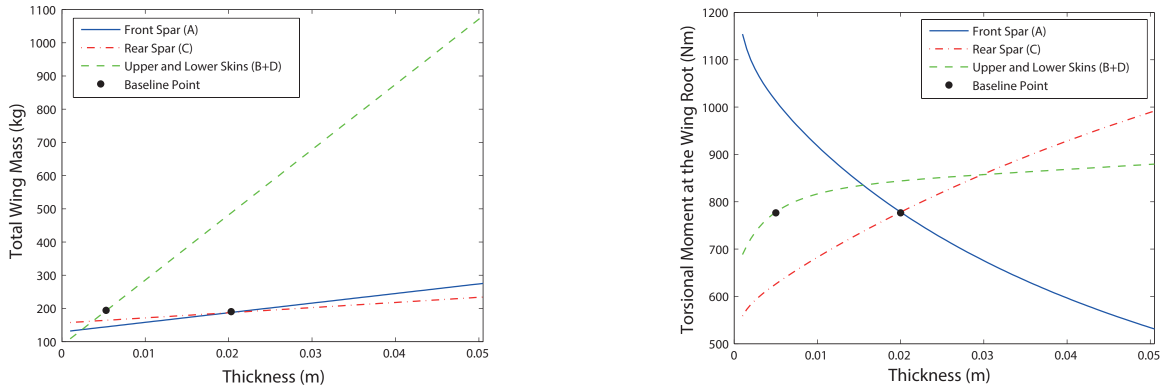


Figure 5.6: Representation of the thicknesses of the front spar (blue), the rear spar (red) and the upper and lower wingbox segments (green).

In Figure 5.7a, it can be verified that the wing mass increases linearly with the increase of thickness of the wingbox segments. Since the linear density is directly proportional to the cross-sectional area and the cross-sectional area is directly proportional to the thickness of the wingbox segments, it is confirmed that the linear density of the beam elements is directly proportional to the thickness of the wingbox. Note that the upper and lower wingbox segments (B and D) have a bigger influence in the wing mass than the spar's thickness segments (A and C). This happens because the sum of lengths B and D is greater than the lengths of spars A or C for the reference case.

In Figure 5.7b, it can be observed that an increase in the front spar thickness (t_A) leads to a lower applied torsional moment. This occurs due to the shift of the shear center in the leading edge direction, closer to the aerodynamic center of the airfoil located at the quarter chord. If the thicknesses of the rear spar (t_C) or of the upper and lower skins (t_{BD}) increase, the magnitude of the torsional moment will increase as well. This occurs due to the rearwards shear center shift with the increase of the thickness. Note that, according to Figure 5.7b, the thickness of front and rear spars seem to have a greater impact in the torsional moment applied at the wing root than the thickness of the skins. The bending moments M_y and M_z are not represented because they are constant and independent of the thicknesses. For the present case, the axial load (N_x) contribution is zero.

In Figure 5.8a, it is verified that an increase of the thickness of any section of the wingbox leads to a decrease in the vertical tip displacement, due to an increase in bending stiffness. Out of the three cases presented, the thickness of the front spar has the greatest impact in the vertical tip displacement.

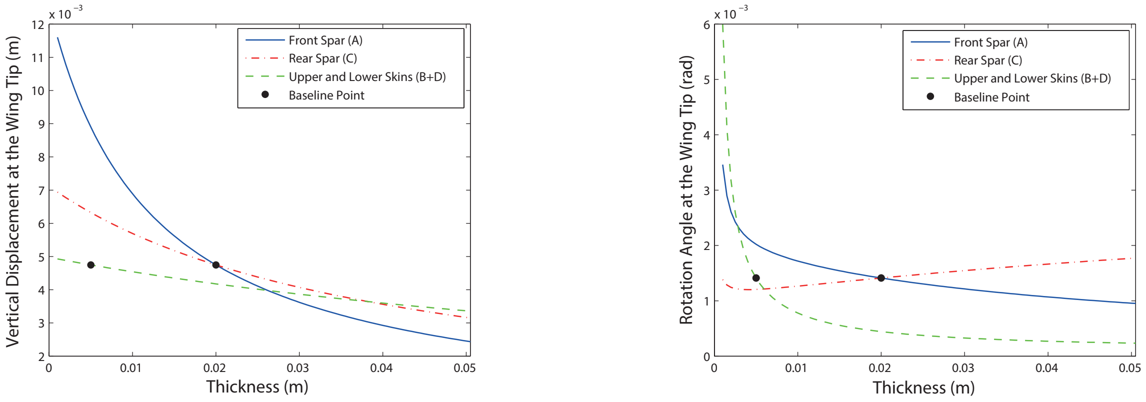


(a) Total wing mass as function of the thickness of the different wingbox segments.

(b) Torsional moment as function of the thickness of the different wingbox segments.

Figure 5.7: Influence of the wingbox segments' thickness in the wing mass and applied torsional moment.

As it was checked in the previous section, the tip rotation of the wing is influenced by two factors: the position of the shear center, that influences the magnitude of the torsional moment, and the torsional rigidity. Considering Figure 5.8b, it is observed that, for the front spar case, an increase in thickness leads to a reduction in the absolute value of the tip rotation. This occurs due to simultaneous increase in the torsional rigidity and decrease in the magnitude of the applied torsional moment. For the rear spar, an increase in thickness increases not only the magnitude of the torsional moment but also the torsional rigidity. However, the effect of the increase of the torsional moment prevails over the increase of torsional rigidity and an increase of the value of the tip rotation angle is observed. For the upper and lower skins thicknesses, an increase of thickness leads to an increase in the absolute value of the torsional moment and an increase in torsional rigidity. In this case, the increase of torsional rigidity prevails over the increase of torsional moment, resulting in a lower absolute value of the tip rotation angle with the increase of thickness.



(a) Vertical tip displacement as function of the thickness of the different wingbox segments.

(b) Tip rotation as function of the thickness of the different wingbox segments.

Figure 5.8: Influence of the wingbox segments' thickness in the tip vertical displacement and rotation.

Considering the maximum stress as a function of the wingbox thicknesses, it can be verified, from Figure 5.9, that an increase in t_A, t_{BD} or t_C, contributes to the decrease in the maximum stress at the wing root. For this reference case, the front thickness (t_A) has the greatest influence in the root stress.

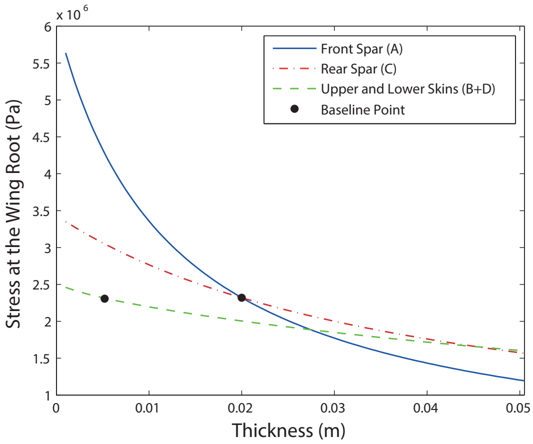


Figure 5.9: Influence of the wingbox segments' thickness in the wing root maximum stress.

5.4 Material Parameters Influence

In this section, the influence of the material properties on the interest functions will be addressed. Homogeneous isotropic materials will be considered. In reality, material properties are discrete variables in optimization problems, but for the sake of this parametric study, hypothetical continuum variations are assumed.

Density

The first property to be addressed is the density of a material. Out of the interest functions previously presented, this parameter will only influence the overall weight of the structure. In Figure 5.10, the trivial linear relation between the total mass of the wing and the material density is observed.

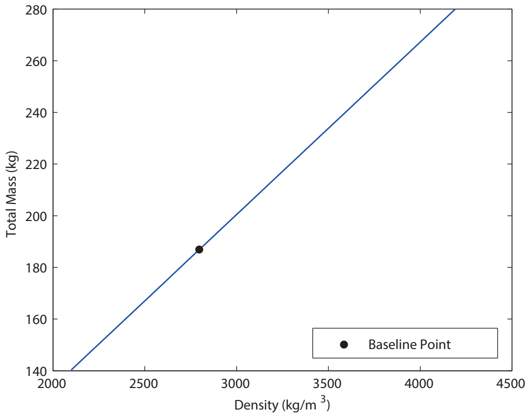
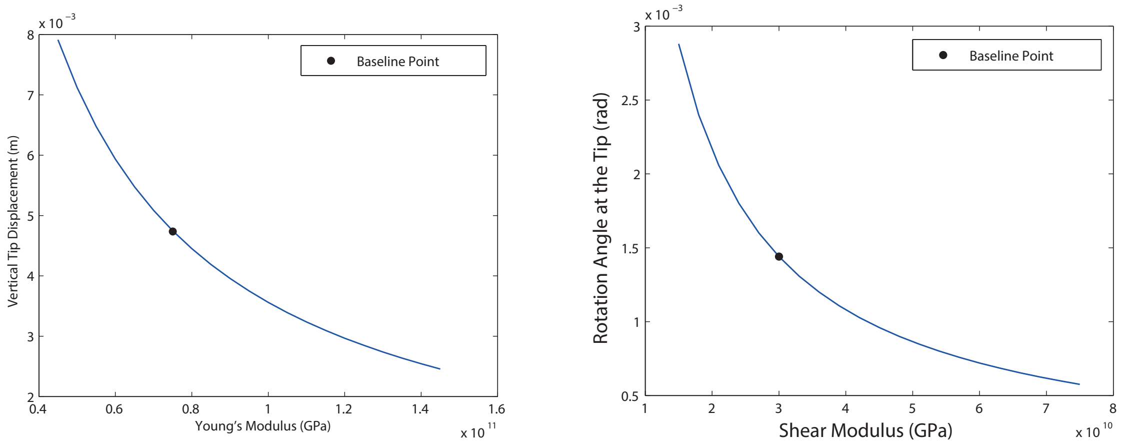


Figure 5.10: Influence of the material density in the wing mass.

Young's and Shear Modulus

Besides the material density, the Young's Modulus and the shear Modulus will also affect the output functions.



(a) Tip vertical displacement as function of the Young's Modulus.

(b) Tip rotation as function of the Shear Modulus Modulus.

Figure 5.11: Influence of the material properties in the vertical displacement and rotation.

In Figure 5.11a, it is observed that stiffer materials (higher Young's Modulus), lead to lower vertical tip's displacement. In Figure 5.11b it is possible to observe that, the larger the value of the Shear Modulus, the lower the tip rotation value.

In practice, these three variables cannot be manipulated individually, which means that a trade-off must be met when choosing the materials for a certain design. Besides these three properties, many other factors take place when choosing a material such as the resistance to corrosion, fatigue behavior, costs, machining and fabrication ability. According to Corke [51], aluminum remains the most broadly used metal for aircraft applications, due to low material cost, ease of machining, good corrosion resistance, and relatively high strength-to-weight ratio. Aluminum 2024-T3 will be used throughout this thesis.

5.5 Summary of Results

The obtained results are summarized in Table 5.2. This table presents the effects in the wing mass (m_{wing}), vertical tip displacement (δ_{tip}), tip rotation (θ_{tip}) and maximum root stress (σ_{max}) resultant from changing the internal structure configuration and material mechanical properties. Some of the observed results in Table 5.2 are intuitively verified. For instance, by increasing the thickness of any of the wingbox segments, the total mass increases. The stiffness of the cross-sections also increase with an increase in thickness, leading to a decrease in the deflection of the wing tip when subjected to external loads. As for the material properties, it is verified that an increase in the Young's modulus leads to a decrease in the bending deflection and the increase of the shear modulus reduces the tip torsional rotation.

Increase in	m_{wing}	δ_{tip}	θ_{tip}	σ_{max}
Front spar location (x_1/c)	decreases	decreases	increases	increases
Rear spar location (x_2/c)	increases	increases	increases	increases
Front spar thickness (t_A)	increases	decreases	decreases	decreases
Skin thickness (t_{BD})	increases	decreases	decreases	decreases
Rear spar thickness (t_C)	increases	decreases	increases	decreases
Density (ρ)	increases	does not change	does not change	does not change
Young's modulus (E)	does not change	decreases	does not change	does not change
Shear modulus (G)	does not change	does not change	decreases	does not change

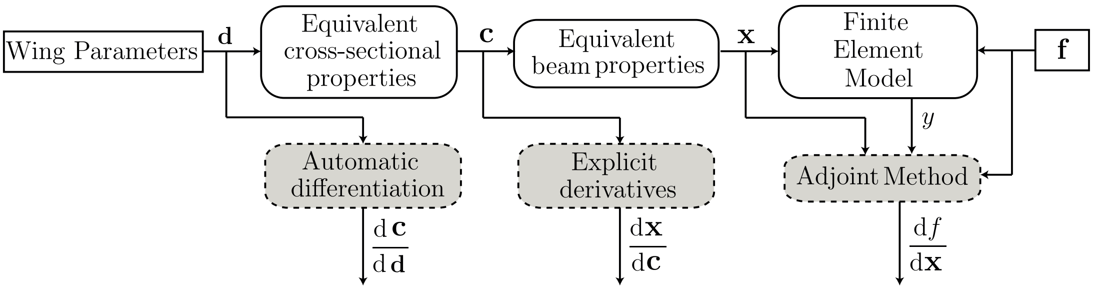
Table 5.2: Effect of the parameters studied on the output functions in the neighborhood of the baseline point.

Note that the presented values results are valid in the neighborhood of the baseline configuration point, meaning that this behavior is not globally guaranteed. However, the results related with the internal parameters will be useful in the next chapter to confirm if the sign in the total sensitivity analysis matches de behavior obtained with the parametric studies.

Chapter 6

Sensitivity Analysis Framework

In this chapter the calculation of sensitivities of the interest functions with respect to the selected design variables will be presented. Three auxiliary modules are introduced in the framework to compute the exact derivative information: the automatic differentiation, the explicit derivative and the adjoint method. These sensitivity blocks are represented in shaded dashed boxes in Figure 6.1.



Wing Parameters
$\mathbf{d}$
Equivalent cross-sectional properties
$\mathbf{c}$
Equivalent beam properties
$\mathbf{x}$
Finite Element Model
$\mathbf{f}$
y
Automatic differentiation
Explicit derivatives
Adjoint Method
$\frac{\mathrm{d}\mathbf{c}}{\mathrm{d}\mathbf{d}}$
$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\mathbf{c}}$
$\frac{\mathrm{d}f}{\mathrm{d}\mathbf{x}}$

Figure 6.1: Structural sensitivity analysis framework.

After selecting the design variables in the previous chapter, it is possible to define the design vector $\mathbf{d}$, which corresponds to a vector of dimension $5\,(n_{fe}+1)$, containing the parametrized internal structure configuration at each node location, being organized as

$$\mathbf{d} = [x_1/c \; x_2/c \; t_A \; t_{BD} \; t_C]_i^T \, , \qquad i = 1, ..., n_{fe}+1 \; . \tag{6.1}$$

The cross-section properties information vector $\mathbf{c}$ represents a vector of dimension $6(n_{fe}+1)$ containing the cross-section properties (EA_c, EI_{z_c}, EI_{y_c}, GJ_c and ρ_{l_c}) at the nodes locations, calculated in the shear center frame, as well as the shear center chordwise coordinate (SC_x), represented in the airfoil frame. This vector can be written as

$$\mathbf{c} = \mathbf{c_1} \cup \mathbf{c_2} \cup \mathbf{c_3} = \left[EA_c \, EI_{zz_c} \, EI_{yy_c} \, GJ_c\right]_i^T \cup [SC_{x_c}]_i \cup [\rho_{l_C}]_i \, , \qquad i = 1, ..., n_{fe}+1 \; , \tag{6.2}$$

where $\mathbf{c_1}$, $\mathbf{c_2}$ and $\mathbf{c_3}$ represent segments of the vector $\mathbf{c}$. The properties are organized in this way for a better understanding of the chain derivative calculation process, that will be presented in subsequent subsections.

The vector $\mathbf{x}$, of dimension $6\,n_{fe}$, contains the average material properties of each finite element (EA, EI_{zz}, EI_{yy}, GJ, ρ_l), and the shear center location at the end node of the finite element (SC_x) as

$$\mathbf{x} = \mathbf{x_1} \cup \mathbf{x_2} \cup \mathbf{x_3} = [EA \, EI_{zz} \, EI_{yy} \, GJ]_j^T \cup [SC_{x_c}]_j \cup [\rho_l]_j \, , \qquad j = 1, ..., n_{fe} \; . \tag{6.3}$$

For a general interest function f, the calculation of its sensitivity with respect to the design variables can be written as

$$\frac{\mathrm{d}f}{\mathrm{d}\mathbf{d}} = \frac{\mathrm{d}f}{\mathrm{d}\mathbf{x}} \cdot \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\mathbf{c}} \cdot \frac{\mathrm{d}\mathbf{c}}{\mathrm{d}\mathbf{d}} \,, \tag{6.4}$$

where:

- $\frac{\mathrm{d}\mathbf{c}}{\mathrm{d}\mathbf{d}}$ is calculated through automatic differentiation (see Section 6.1);
- $\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\mathbf{c}}$ is obtained by using explicit differentiation (see Section 6.2);
- $\frac{\mathrm{d}f}{\mathrm{d}\mathbf{x}}$ is found by applying the adjoint method (see Section 6.3).

Depending on the interest function f, it may not be necessary to run all the modules in the optimization chain. For instance, in order to calculate the sensitivity of the wing mass, the adjoint module is not required, since the wing mass is independent of the structural response caused by the aerodynamic loading. However, if the tip displacement is chosen as a performance function, it will be required to go through the adjoint method module, since this is an output of the structural response.

In the following subsections, the three sensitivity calculation modules, that represent the terms of the right-hand side of Equation (6.4) are highlighted in Figure 6.1 , will be presented and, in the last subsection, it will be shown how to compute the global sensitivity of the interest functions.

6.1 Automatic Differentiation

The first stage in the gradient calculation process requires to compute the sensitivities of the cross-sectional properties $\mathbf{c}$ with respect to the parametrized geometric variables $\mathbf{d}$. An exact derivative method is required for this stage, excluding the previously presented approximation sensitivity methods. Since an explicit symbolic differentiation is difficult to be derived due to the complexity of the problem and the alternative analytic sensitivity methods are not suitable for this problem, since the finite element solution and state variables are not involved in this part of the program, automatic differentiation was chosen to be applied.

The function `fequiprops.m`, developed to calculate the equivalent cross-sectional properties at a certain cross-section location, presented in Section 4.1, is automatically differentiated with the aid of *ADiMat*. A function `g_fequiprops.m` is generated, containing the code to calculate the derivative information. Even though the number of input variables of `fequiprops.m` (10) is slightly greater than the number of output variables (7), a forward mode was chosen since it is had less usage limitations in *ADiMat*, when compared to the reverse mode. According to the developers [39], these limitations concern, for instance, some missing built-in functions or problems in dealing with variables defined within loops.

In order to test the function `g_fequiprops.m`, the baseline example presented in Table 4.2, was chosen. Table 6.1 summarizes the results obtained for the sensitivities of the cross-sectional properties (EA, EI_{zz}, EI_{yy}, GJ and ρ_l) with respect to a set of variables x listed in the first column. Note that these properties were already converted from the shear center reference frame to the global structural frame and some of the computed sensitivities will not be used in the following optimization case, namely, the sensitivities with respect to the material properties (ρ, E and G), since these will be assumed constant. This table is organized in an upper part, containing the variables related to the cross-section configuration, and a lower part comprising the variables related to the material properties.

Variable (x)	$\frac{\partial EA}{\partial x}$	$\frac{\partial EI_{zz}}{\partial x}$	$\frac{\partial EI_{yy}}{\partial x}$	$\frac{\partial GJ}{\partial x}$	$\frac{\partial \rho_l}{\partial x}$
Chord length (c)	7.1582E+08	9.2216E+07	3.2094E+07	5,4557E+06	2,6724E+01
Front spar loc. (x_1/c)	-6.5586E+08	-1.4094E+08	6.3684E+06	-2.9093E+06	-2.4485E+01
Rear spar loc. (x_2/c)	3.4756E+08	1.1117E+08	-3.3573E+07	-4.2930E+05	1.2975E+01
Front spar thick. (t_A)	1.1075E+10	5.0658E+08	9.2292E+08	3.1698E+06	4.1347E+02
Upper skin thick. (t_B)	3.7588E+10	3.1684E+09	1.3537E+08	1.7213E+08	1.4033E+03
Rear spar thick. (t_C)	5.9215E+09	4.8457E+08	4.9346E+08	1.6948E+06	2.2107E+02
Lower skin thick. (t_D)	3.7588E+10	3.1684E+09	1.3537E+08	1.7213E+08	1.4033E+03
Density (ρ)	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	9.5442E-03
Young's Modulus (E)	9.5442E-03	6.8676E-04	3.9575E-04	0.0000E+00	0.0000E+00
Shear Modulus (G)	0.0000E+00	-9.5173E-22	1.1121E-29	6.0619E-05	0.0000E+00

Table 6.1: Sensitivities of the cross-sectional properties with respect to the internal parametric variables and material properties.

Based on the analysis of the sensitivities for the axial rigidity ($\frac{\partial EA}{\partial x}$), it is possible to state that, due to the positive sign of the derivative, an increase of any design variable related to the wingbox configuration, except for the front spar relative location (x_1/c), leads to an increase of axial rigidity. The same is verified for the bending stiffness EI_{zz} case. For the bending stiffness EI_{yy}, the only variable that presents a negative sensitivity corresponds to the rear spar relative location (x_2/c), meaning that an increase in x_2/c leads to a decrease in the bending stiffness EI_{yy}. The torsional rigidity GJ increases with the increase in thickness of any wingbox segment and also with an increase in the chord length, but decreases when moving any of the spars rearwards. The linear density (ρ_l) increases with an increase in chord length, in thickness of any wingbox segment or with a rearwards shift of the rear spar. Note that by moving the front spar rearwards (x_1/c), a decrease in the cross-sectional linear density is observed, since the length of both upper and lower skins becomes shorter. The lower part of Table 6.1, presents the following trivial information: the density of the material only influences the cross-sectional linear density sensitivity, the shear modulus only affects the torsional rigidity sensitivity and the Young's modulus influences exclusively axial and both bending stiffnesses.

Even though the previous qualitative analyses, based on the sensitivity sign, seem coherent with the expected variations, it is necessary to validate the sensitivity values quantitatively. In order to benchmark these results, approximate values of the sensitivities were calculated with the forward finite difference approximation method. The approximated results were found to be similar to the solutions obtained with the automatic differentiation method and are represented in Table 6.2 in terms of relative errors, using the automatic differentiation solution as reference. A relative perturbation step of 10^{-4} was selected for the finite differences derivative estimations, to closely approximate the true sensitivity values but, at the same time, to avoid the effect of subtractive cancellation. Other relative steps, such as 10^{-6} and 10^{-8}, were also tested and included in Appendix A. However some of the estimated sensitivities exhibited smaller relative errors and others presented higher errors, not being verified an overall consistent trend in relative error variation. This occurs due to the fact that each sensitivity has an individual optimal step, and, therefore, subtractive cancellation effects occur, in each case, for different relative step sizes.

According to Table 6.2, the largest relative error obtained with the forward finite difference benchmark, with a value of approximately 0.118%, was found for $\frac{\partial GJ}{\partial x_2/c}$. All the other variables presented lower relative errors, validating the automatic differentiation tool. The entries N/A correspond to the cases where the relative error calculation is non-applicable due to division by zero.

Variable (x)	$\varepsilon\left(\dfrac{\partial EA}{\partial x}\right)$	$\varepsilon\left(\dfrac{\partial EI_{zz}}{\partial x}\right)$	$\varepsilon\left(\dfrac{\partial EI_{yy}}{\partial x}\right)$	$\varepsilon\left(\dfrac{\partial GJ}{\partial x}\right)$	$\varepsilon\left(\dfrac{\partial \rho_l}{\partial x}\right)$
Chord length (c)	1.3323E-15	6.6368E-05	1.1278E-05	1.0000E-04	6.4357E-13
Front spar location (x_1/c)	2.7817E-07	1.1547E-05	1.4343E-07	1.6712E-05	2.7817E-07
Rear spar location (x_2/c)	7.2508E-07	3.9424E-05	7.8455E-06	1.1852E-03	7.2508E-07
Front spar thickness (t_A)	2.2213E-12	3.0904E-05	8.7548E-13	9.6505E-05	2.6338E-12
Upper skin thickness (t_B)	2.5566E-12	3.6599E-07	1.0157E-05	5.2672E-05	4.5465E-12
Rear spar thickness (t_C)	3.3950E-13	1.6540E-05	3.3845E-13	9.8127E-05	4.5098E-12
Lower skin thickness (t_D)	2.6610E-13	3.6599E-07	1.0157E-05	5.2672E-05	3.2209E-12
Density (ρ)	N/A	N/A	N/A	N/A	6.4324E-13
Young's Modulus (E)	2.1811E-15	1.7761E-13	3.7902E-13	N/A	N/A
Shear Modulus (G)	N/A	N/A	N/A	4.6513E-13	N/A

Table 6.2: Absolute value of the sensitivities errors using the forward finite difference method with a relative step size of 10^{-4}.

The sensitivities of the shear center location with respect to a list of parameters x and the respective relative errors are presented in Table 6.3. Note that, for convenience, these variables are kept represented relative to the airfoil frame. The chordwise shear center coordinate (SC_x) corresponds to the distance of the shear center from the leading edge of the airfoil. Positive values of $\frac{\partial SC_x}{\partial x}$ correspond to a variation on the location of the shear center in the rearwards direction along the chord, for a positive variation of x. Thus, for this configuration, by increasing either the chord of the airfoil (c), the thickness of the upper and lower skins (t_B and t_D) or the thickness of the rear spar (t_C), the shear center moves in the trailing edge direction. The same occurs for a rearwards shift in the front and rear spars (x_1/c and x_2/c). However, the increase of the front spar thickness (t_A) leads to a shift of the shear center position towards the leading edge, according to the negative sensitivity value. These sensitivities will be important to compute the total sensitivity of the tip rotation with respect to the design variables, since they depend on the magnitude of the torsional moment, which is dependent on the shear center location.

For the shear center coordinate SC_y (in the airfoil frame), it is possible to state that the influence

Variable (x)	$\dfrac{\partial SC_x}{\partial x}$	$\dfrac{\partial SC_y}{\partial x}$	$\varepsilon\left(\dfrac{\partial SC_x}{\partial x}\right)$	$\varepsilon\left(\dfrac{\partial SC_y}{\partial x}\right)$
Chord length (c)	4.6432E-01	3.8353E-09	6.3369E-08	3.1887E-06
Front spar location (x_1/c)	5.0424E-01	1.2271E-07	3.6436E-06	6.2725E-06
Rear spar location (x_2/c)	3.1147E-01	1.5900E-09	2.9460E-05	2.0102E-04
Front spar thickness (t_A)	-3.3701E+00	4.6136E-08	3.2061E-05	2.1757E-04
Upper skin thickness (t_B)	1.3502E+00	6.5265E+00	2.9327E-05	5.9314E-05
Rear spar thickness (t_C)	2.3410E+00	-3.3303E-08	1.6668E-05	1.7466E-04
Lower skin thickness (t_D)	2.7663E+00	-6.5265E+00	1.8248E-05	5.9603E-05
Density (ρ)	0.0000E+00	0.0000E+00	N/A	N/A
Young's Modulus (E)	0.0000E+00	0.0000E+00	N/A	N/A
Shear Modulus (G)	7.5731E-29	-3.8638E-28	N/A	N/A

Table 6.3: Sensitivity of the shear center location (SC_x, SC_y) with respect to the parameters x and the relative error using the forward finite difference method with a relative step size of 10^{-4}.

of the listed variables in Table 6.3 is negligible, apart from the upper and lower skin thicknesses terms. Note that these two exception terms ($\frac{\partial SC_y}{\partial t_B}$ and $\frac{\partial SC_y}{\partial t_D}$) are symmetric, which means that if they are forced to be equal, in order to maintain symmetry along the chordwise axis (allowing the decoupling of both bending components), the overall contribution to the shear center vertical shift will be zero. Also note that, for a homogeneous wingbox, the material properties ρ, E and G do not influence the position of the shear center. The previous results were benchmarked using the forward finite difference method with a relative step size of 10^{-4}. The relative errors are also presented in Table 6.3 being the highest error in the order of magnitude of 10^{-5}, validating the results obtained for the sensitivities.

According to Table 6.4, the time required to compute all the previously presented sensitivities with the automatic differentiation method is 5.85 times higher than the finite differences approximation method. However, the results obtained with the former method are exact, which makes it preferable than the latter since the absolute differences in computational time between both methods are much smaller than a second. Also note that the represented time does not include the time that the parser takes to generate de differentiated code. These time intervals, obtained using the `tic` and `toc` functions from MATLAB®, are quite small and may not be very accurate.

Method	**Finite differences (FD)**	**Automatic differentiation (AD)**	**Ratio AD/FD**
Δt (s)	0.024582	0.143791	5.85

Table 6.4: CPU time (Δt) for calculating all the first module's sensitivities using an Intel(R) Core(TM) i5-4200U CPU @ 2.30Ghz.

Note that not all of variables x previously analyzed will be used in the sensitivity chain framework. Since the studied reference wing has a taper ratio of $\lambda = 1$, meaning that the chord length (c) is constant throughout the span, and maintaining this parameter as a constant, the sensitivity with respect to the variable c will not be required. Furthermore, the thicknesses t_B and t_D will be considered to be equal, so that the structure is symmetric along the y-axis and so that the product of inertia $I_{yz} = 0$, in the global reference frame.

For each run of `g_fequiprops.m`, the necessary derivative information may be organized into three different element matrices,

$$\left[\frac{\mathrm{d}\mathbf{c_1}}{\mathrm{d}\mathbf{d}}\right]_{elem} = \begin{bmatrix} \frac{\partial EA_c}{\partial x_1/c} & \frac{\partial EA_c}{\partial x_2/c} & \frac{\partial EA_c}{\partial t_A} & \frac{\partial EA_c}{\partial t_{BD}} & \frac{\partial EA_c}{\partial t_C} \\ \frac{\partial EI_{zz_c}}{\partial x_1/c} & \frac{\partial EI_{zz_c}}{\partial x_2/c} & \frac{\partial EI_{zz_c}}{\partial t_A} & \frac{\partial EI_{zz_c}}{\partial t_{BD}} & \frac{\partial EI_{zz_c}}{\partial t_C} \\ \frac{\partial EI_{yy_c}}{\partial x_1/c} & \frac{\partial EI_{yy_c}}{\partial x_2/c} & \frac{\partial EI_{yy_c}}{\partial t_A} & \frac{\partial EI_{yy_c}}{\partial t_{BD}} & \frac{\partial EI_{yy_c}}{\partial t_C} \\ \frac{\partial GJ_c}{\partial x_1/c} & \frac{\partial GJ_c}{\partial x_2/c} & \frac{\partial GJ_c}{\partial t_A} & \frac{\partial GJ_c}{\partial t_{BD}} & \frac{\partial GJ_c}{\partial t_C} \end{bmatrix} , \tag{6.5a}$$

$$\left[\frac{\mathrm{d}\mathbf{c_2}}{\mathrm{d}\mathbf{d}}\right]_{elem} = \begin{bmatrix} \frac{\partial \rho_{l_c}}{\partial x_1/c} & \frac{\partial \rho_{l_c}}{\partial x_2/c} & \frac{\partial \rho_{l_c}}{\partial t_A} & \frac{\partial \rho_{l_c}}{\partial t_{BD}} & \frac{\partial \rho_{l_c}}{\partial t_C} \end{bmatrix} , \tag{6.5b}$$

and

$$\left[\frac{\mathrm{d}\mathbf{c_3}}{\mathrm{d}\mathbf{d}}\right]_{elem} = \begin{bmatrix} \frac{\partial SC_x}{\partial x_1/c} & \frac{\partial SC_x}{\partial x_2/c} & \frac{\partial SC_x}{\partial t_A} & \frac{\partial SC_x}{\partial t_{BD}} & \frac{\partial SC_x}{\partial t_C} \end{bmatrix} . \tag{6.5c}$$

Note that the term element, in this case, is not associated with the finite elements, but with each nodal cross-section. If all the cross-sections are forced to be equal, these element matrices will also be equal for every cross-section, only needing to calculate the derivative code once. The respective global derivative matrices $\left[\frac{\mathrm{d}\mathbf{c_1}}{\mathrm{d}\mathbf{d}}\right]$,$\left[\frac{\mathrm{d}\mathbf{c_2}}{\mathrm{d}\mathbf{d}}\right]$ and $\left[\frac{\mathrm{d}\mathbf{c_3}}{\mathrm{d}\mathbf{d}}\right]$ are obtained by assembling the element matrices presented above.

6.2 Symbolic Differentiation

The second stage in the gradient calculation process evaluates the sensitivities of the finite element average structural properties with respect to the structural cross-sectional properties of the two nodes that delimit the finite element (marked with the subscripts 1 and 2).

Since the derivatives of Equations (4.22a), (4.22b) and (4.22c) are trivial, the analytical symbolic approach was chosen to be applied. Therefore, the partial derivatives expressions for constant material properties are given by

$$\frac{\partial EA}{\partial EA_{c_1}} = \frac{1}{3}\left(1 + \frac{1}{2}\sqrt{\frac{A_{c_2}}{A_{c_1}}}\right) , \tag{6.6a}$$

$$\frac{\partial EI}{\partial EI_{c_1}} = \frac{1}{5}\left(1 + \frac{3}{4}\sqrt[4]{\frac{I_{c_2}}{I_{c_1}}} + \frac{1}{2}\sqrt{\frac{I_{c_2}}{I_{c_1}}} + \frac{1}{4}\sqrt[4]{\left(\frac{I_{c_2}}{I_{c_1}}\right)^3}\right) \tag{6.6b}$$

and

$$\frac{\partial GJ}{\partial GJ_{c_1}} = \frac{1}{5}\left(1 + \frac{3}{4}\sqrt[4]{\frac{J_{c_2}}{J_{c_1}}} + \frac{1}{2}\sqrt{\frac{J_{c_2}}{J_{c_1}}} + \frac{1}{4}\sqrt[4]{\left(\frac{J_{c_2}}{J_{c_1}}\right)^3}\right) . \tag{6.6c}$$

The expressions for $\frac{\partial EA}{\partial EA_{c_2}}$, $\frac{\partial EI}{\partial EI_{c_2}}$ and $\frac{\partial GJ}{\partial GJ_{c_2}}$ are obtained from changing, in Equation (6.6), A_{c_1} for A_{c_2}, I_{c_1} for I_{c_2}, J_{c_1} for J_{c_2} and vice-versa.

The sensitivities $\frac{\partial \rho_l}{\partial \rho_{l_{c_1}}}$ and $\frac{\partial \rho_l}{\partial \rho_{l_{c_2}}}$ correspond to 0.5 each by definition of the mean value between two variables. Being this derivation straightforward, no numerical benchmark with other methods were performed.

These expressions were, however, checked with the MuPAD™ symbolic differentiation tool. The sensitivity information is organized in element matrices as

$$\left[\frac{\mathrm{d}\mathbf{x_1}}{\mathrm{d}\mathbf{c_1}}\right]_{elem} = \begin{bmatrix} \frac{\partial EA}{\partial EA_{c1}} & 0 & 0 & 0 & \frac{\partial EA}{\partial EA_{c2}} & 0 & 0 & 0 \\ 0 & \frac{\partial EI_{zz}}{\partial EI_{zz_{c1}}} & 0 & 0 & 0 & \frac{\partial EI_{zz}}{\partial EI_{zz_{c2}}} & 0 & 0 \\ 0 & 0 & \frac{\partial EI_{yy}}{\partial EI_{yy_{c1}}} & 0 & 0 & 0 & \frac{\partial EI_{yy}}{\partial EI_{yy_{c2}}} & 0 \\ 0 & 0 & 0 & \frac{\partial GJ}{\partial GJ_{c1}} & 0 & 0 & 0 & \frac{\partial GJ}{\partial GJ_{c2}} \end{bmatrix} \tag{6.7a}$$

and

$$\left[\frac{\mathrm{d}\mathbf{x_3}}{\mathrm{d}\mathbf{c_3}}\right]_{elem} = \left[\frac{\partial \rho_l}{\partial \rho_{l_{c1}}} \quad \frac{\partial \rho_l}{\partial \rho_{l_{c2}}}\right] . \tag{6.7b}$$

Similarly to the previous module, the element matrices are assembled to originate the global matrices $\left[\frac{\mathrm{d}\mathbf{x_1}}{\mathrm{d}\mathbf{c_1}}\right]$ and $\left[\frac{\mathrm{d}\mathbf{x_3}}{\mathrm{d}\mathbf{c_3}}\right]$.

6.3 Adjoint Sensitivity Module

For the third sensitivity analysis module, the alternative analytic method was chosen to be applied since the usage of the governing equations may be availed to achieve better efficiency. The adjoint method for sensitivity computation is chosen because the number of intermediate design variables x ($5\,n_{fe}$) is greater than the number of interest functions dependent of the structural solution (2), making it more computationally efficient than the direct method.

The adjoint module can be divided into two parts: the first part corresponds to the adjoint equation solution, where an equation similar to the structural governing equation is solved a number of times equal to the number of interest functions dependent on the structural solution; the second part corresponds to using the solution of the first part, the adjoint vector $\boldsymbol{\psi}$, in order to calculate the total sensitivity of the correspondent interest functions.

The adjoint solution is obtained by solving the adjoint equation:

$$[K]^T \boldsymbol{\psi} = \left[\frac{\partial f}{\partial \mathbf{y}}\right]^T . \tag{6.8}$$

This equation is equivalent to Equation (3.41) but represented in the matrix form. Note that since the stiffness matrix $[K]$ is symmetric, by definition, one gets $[K]^T = [K]$.

The implementation of the adjoint solver may be divided in three steps:

1. Obtain the assembled reduced stiffness matrix $[K]$, which was previously stored when solving the structural finite element problem;

2. Define the adjoint force vector ($\frac{\partial f}{\partial \mathbf{y}}$) for each of the considered interest functions;

3. Solve Equation (6.8) and obtain the correspondent adjoint vector $\boldsymbol{\psi}$.

The sensitivity equation of the function f with respect to a variable x is obtained by evaluating

$$\frac{\mathrm{d}f}{\mathrm{d}x} = \frac{\partial f}{\partial x} - \boldsymbol{\psi}^T \left(\frac{\partial [K]}{\partial x} \tilde{\mathbf{y}} + \frac{\partial \mathbf{f}}{\partial x} \right) \ , \tag{6.9}$$

which is equivalent to Equation (3.13), written in the matrix form for a given variable x. The process of computing the total sensitivity involves the steps:

1. Calculate the explicit derivative of the interest function with respect to dependent terms for each variable x $\left(\frac{\partial f}{\partial x}\right)$;
2. Compute the derivatives of the stiffness matrix relative to the considered variable x $\left(\frac{\partial [K]}{\partial x}\right)$;
3. Calculate the derivatives of the force vector with respect to the variable x $\left(\frac{\partial \mathbf{f}}{\partial x}\right)$;
4. Use the previously calculated structural solution vector $\tilde{\mathbf{y}}$ and the adjoint vector $\boldsymbol{\psi}$ and determine the total sensitivity $\left(\frac{\mathrm{d}f}{\mathrm{d}x_n}\right)$ by solving Equation (6.9).

For this stage, the intermediate vector $\mathbf{x}$ contains the material properties of the finite elements EA, EI_{zz}, EI_{yy}, GJ and also the shear center location SC_x at the end node of each finite element. Regarding the stiffness matrix $[K]$, no explicit dependence on the shear center location is verified, being only dependent explicitly on the material properties. The element derivatives of the stiffness matrix are presented in the Appendix B. For the force vector $\mathbf{f}$, no explicit dependence on the material properties is observed, only depending explicitly on the shear center location.

Also note that instead of using the derivatives of the entire stiffness matrix and force vector, as well as the entire global solution from the structural and adjoint equations, Equation (6.9) can be evaluated at the element level by only using the components correspondent to the considered finite element. This may be performed because each set of intermediate variables x is independent of the properties of the remaining finite elements. Thus, an element solution may be computed for each finite element and, at the end, the sensitivity vector may be assembled.

For each interest function, the element sensitivity analysis vector is given by

$$\left[\frac{\mathrm{d}f}{\mathrm{d}\mathbf{x_1}}\right]_{elem} = \left[\frac{\partial f}{\partial EA} \ \frac{\partial f}{\partial EI_z} \ \frac{\partial f}{\partial EI_y} \ \frac{\partial f}{\partial GJ}\right] \tag{6.10a}$$

and

$$\left[\frac{\mathrm{d}f}{\mathrm{d}\mathbf{c_2}}\right]_{elem} = \left[\frac{\partial f}{\partial SC_x}\right] \ . \tag{6.10b}$$

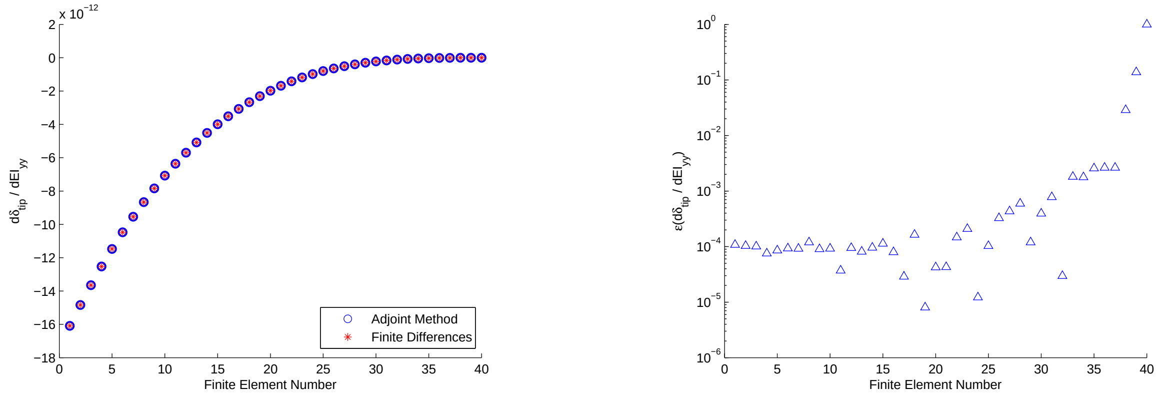
In the following subsections, Equation (6.9) will be applied to two different interest functions: the vertical tip displacement and the tip rotation.

6.3.1 Vertical Tip Displacement

The first interest function sensitivity calculation to be implemented is the vertical tip displacement. Since there are no explicit design variable terms in the tip displacement expression (Equation (5.2)), the term $\frac{\partial f}{\partial x}$ vanishes from Equation (6.9). Also, note that the external force vector $\mathbf{f}$ is explicitly independent on the finite element properties ($\mathbf{x_1}$). Furthermore, recalling that the vertical tip displacement is described by the decoupled bending motion in the $y0z$-plane, it means that only the components F_z and M_y of the external loads will impact the tip displacement output values. Since the external aerodynamic forces are maintained constant in the optimization case, F_z is constant, and the moments M_y are always zero for the considered problem. Thus, the variation of the external force vector with respect to the chordwise location of the shear center location ($\mathbf{x_2}$) is zero, meaning that Equation (6.9) becomes

$$\frac{\mathrm{d}\,\delta_{tip}}{\mathrm{d}\,x_1} = -\boldsymbol{\psi_1}^T \frac{\partial\,[K]}{\partial\,x_1}\tilde{\mathbf{y}}\;. \tag{6.11}$$

The adjoint vector $\boldsymbol{\psi_1}$ is obtained from solving Equation (6.8) with an adjoint load ($\frac{\partial f}{\partial \mathbf{y}}$) equal to a null vector of dimension $6n_{fe}$, except for the entry correspondent to the vertical displacement of the last node ($u_{z,n_{fe}+1}$) which is equal to one.



(a) Sensitivity of the vertical tip displacement with respect to the bending stiffnesses EI_{yy} of each finite element.

(b) Relative error of the sensitivity of the vertical tip displacement with respect to the bending stiffnesses EI_{yy} of each finite element.

Figure 6.2: Sensitivities and relative errors of the sensitivities of the vertical tip displacement with respect to the element bending stiffness EI_{yy}

The sensitivities obtained with the adjoint method, as well as a benchmark with the forward finite differences to validate the computed values, are presented in Figure 6.2a . For the finite differences approximation, a relative step size of 10^{-4} was used for each finite element structural property. It can be observed that the closer the finite element to the root, the stronger influence of the bending stiffness EI_{yy} on the vertical tip displacement. The negative value of $\frac{\mathrm{d}\delta_{tip}}{\mathrm{d}EI_{yy}}$ indicates that an increase in the bending stiffness EI_{yy}, leads to a decrease in the vertical tip displacement δ_{tip}. As one moves along from the root towards the wing tip, the absolute value of the sensitivity decreases, reaching the value of approximately zero at the last finite element. In Figure 6.2b it is presented the relative error of the sensitivity $\frac{\mathrm{d}\delta_{tip}}{\mathrm{d}EI_{yy}}$, when using the finite differences to estimate the sensitivities and using the adjoint method values as reference. Note that the order of magnitude of the error tends to increase for finite elements located closer to the wing tip. This coincides with the fact that the calculated sensitivities become closer to zero. Nevertheless, a good agreement between both methods is verified, thus validating the implementation of the adjoint method.

The sensitivities $\frac{\mathrm{d}\delta_{tip}}{\mathrm{d}EA}$,$\frac{\mathrm{d}\delta_{tip}}{\mathrm{d}EI_{zz}}$ and $\frac{\mathrm{d}\delta_{tip}}{\mathrm{d}GJ}$ are zero, meaning that the properties EA, EI_{zz} and GJ will not influence the vertical displacement and, therefore, are not represented graphically.

6.3.2 Tip Rotation

The second interest function dependent on the structural solution is the tip rotation. Since there are no explicit terms related to the subset of variables x in the tip rotation expression (Equation (5.3)), the term $\frac{\partial \theta_{tip}}{\partial x}$ is equal to zero. In this case, however, the external force vector will depend on the subset of variables x, namely, the shear center location. By changing the shear center's location, the external applied torsional moment varies, even though the aerodynamic force distribution remains the same. Thus, the problem may be separated in two components: the first expressing the dependence of the tip rotation with the subset of structural properties ($\frac{\mathrm{d}\theta_{tip}}{\mathrm{d}x_1}$) and the second related with the influence of the variation of the tip rotation with the shear center variables ($\frac{\mathrm{d}\theta_{tip}}{\mathrm{d}x_2}$). These two equations are given respectively by

$$\frac{\mathrm{d}\,\theta_{tip}}{\mathrm{d}\,x_1} = -\boldsymbol{\psi_2}^T \frac{\partial\,[K]}{\partial\,x_1} \tilde{\mathbf{y}} \tag{6.12a}$$

and

$$\frac{\mathrm{d}\,\theta_{tip}}{\mathrm{d}\,x_2} = -\boldsymbol{\psi_2}^T \frac{\partial \mathbf{f}}{\partial\,x_2} \tag{6.12b}$$

The adjoint vector $\boldsymbol{\psi_2}$ is calculated from solving Equation (6.8) with an adjoint load ($\frac{\partial f}{\partial \mathbf{y}}$) equal to a null vector of dimension $6n_{fe}$, except for the entry correspondent to the tip rotation of the last node ($\theta_{x,n_{fe}+1}$) which is equal to one.

The solution of Equation (6.12a) is represented in Figure 6.3a. Only the torsional rigidity terms (GJ_j) will influence the tip rotation value. The closer the finite element is to the wing tip, the weaker the reduction on tip rotation with the increase in the torsional stiffness (GJ). The sensitivity values appear in pairs of equal sensitivities since the torsional moments are only applied at the end nodes of the odd finite elements, according to the adopted load transfer model. These values were validated with a forward finite difference approximation method. The observed relative errors lie on the order of magnitude of 10^{-4}. The only exception is observed in the last finite element in which subtractive canceling occurs with the forward finite differences, leading to a relative error of $\varepsilon = 1$. The sensitivities $\frac{\mathrm{d}\theta_{tip}}{\mathrm{d}EA}$,$\frac{\mathrm{d}\theta_{tip}}{\mathrm{d}EI_{zz}}$ and $\frac{\mathrm{d}\theta_{tip}}{\mathrm{d}EI_{yy}}$ were omitted since they are equal to zero.

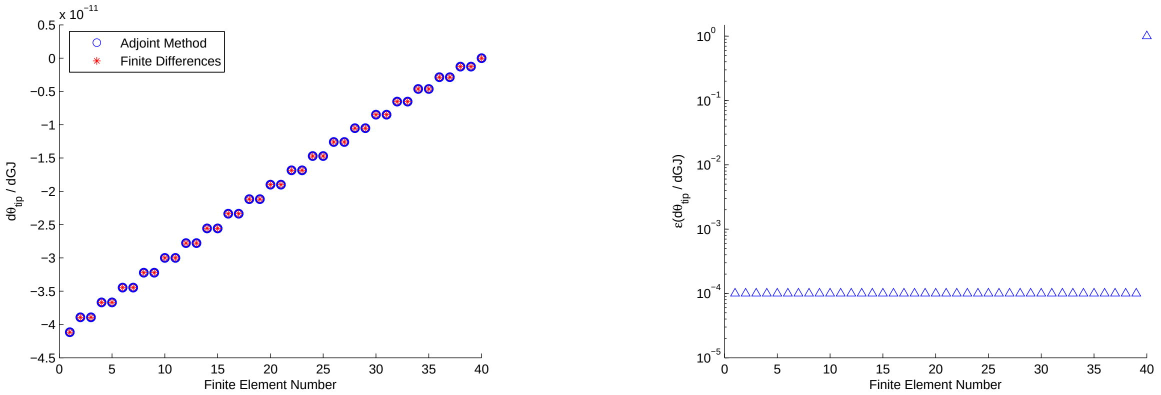
The solution of Equation (6.12b) is shown in Figure 6.3a. Regard that the explicit influence of the shear center location on the cross-sections at the end of the even finite elements is zero, since no external torsional moments are applied at those points. The benchmark with the forward finite element method was also performed, similarly to the previous cases. The correspondent sensitivity error values are graphically represented in Figure 6.3b. The sensitivities at the end nodes of the even finite elements are not represented since their value is equal to zero, leading to a division by zero problem.

Similarly to Subsection 6.3.1, the agreement among the sensitivities obtained from the adjoint method and the finite differences is very good, proving the correct implementation of the former method.

In Table 6.5 it is represented the time intervals necessary to compute the sensitivities $\frac{\mathrm{d}\theta_{tip}}{\mathrm{d}x}$ using the `tic` and `toc` functions of MATLAB®. Notice that the time interval to compute the sensitivities is greater with the finite differences method when compared to the adjoint method by a factor of 98.94. For the former method, 41 runs of the structural third module had to be performed to estimate these values. For the latter case, only one run of the structural module had to be performed, plus a two runs on the adjoint module (one for each interest function).

Method	Finite Differences (FD)	Adjoint method (AM)	Ratio FD/AM
Δt (s)	1.713513	0.017318	98.94

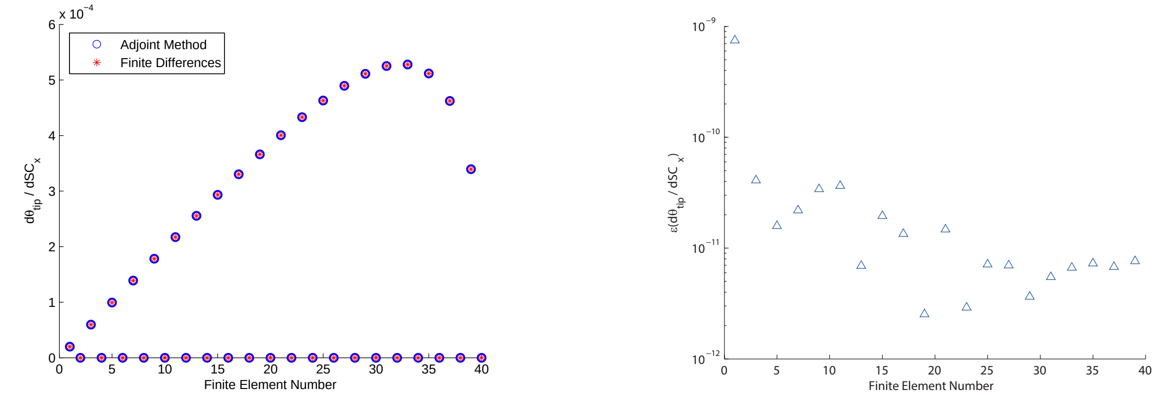
Table 6.5: CPU time (Δt) for calculating all the third module's sensitivities using an Intel(R) Core(TM) i5-4200U CPU @ 2.30Ghz.



(a) Sensitivity of the tip rotation with respect to the torsional stiffness GJ of each finite element.

(b) Relative error of the sensitivity of the tip rotation with respect to the torsional stiffness GJ of each finite element.

Figure 6.3: Sensitivities and relative errors of the sensitivities of the tip rotation with respect to the torsional stiffness GJ



(a) Sensitivity of the tip rotation with respect to the shear center location SC_x of each finite element.

(b) Relative error of the sensitivity of the tip rotation with respect to the shear center location SC_x of each finite element.

Figure 6.4: Sensitivities and relative errors of the sensitivities of the tip rotation with respect to the shear center location SC_x

6.4 Total Sensitivity

The last part on the sensitivity analysis framework is to multiply the assembled matrices computed in each of the previous modules in order to obtain the total sensitivity vector for each interest function, according to Equation (6.4).

For the wing mass, the total sensitivity with respect to the design vector is calculated as

$$\frac{\mathrm{d}\, m_{wing}}{\mathrm{d}\,\mathbf{d}} = \frac{\mathrm{d}\, m_{wing}}{\mathrm{d}\,\mathbf{c_3}} \frac{\mathrm{d}\,\mathbf{c_3}}{\mathrm{d}\,\mathbf{d}} \ . \tag{6.13}$$

The total sensitivity for the tip vertical displacement is given by

$$\frac{\mathrm{d}\,\delta_{tip}}{\mathrm{d}\,\mathbf{d}} = \frac{\mathrm{d}\,\delta_{tip}}{\mathrm{d}\,\mathbf{x_1}} \frac{\mathrm{d}\,\mathbf{x_1}}{\mathrm{d}\,\mathbf{c_1}} \frac{\mathrm{d}\,\mathbf{c_1}}{\mathrm{d}\,\mathbf{d}} \ . \tag{6.14}$$

For the tip rotation case, the sensitivity vector expression is obtain using the following expression:

$$\frac{\mathrm{d}\,\theta_{tip}}{\mathrm{d}\,\mathbf{d}} = \frac{\mathrm{d}\,\theta_{tip}}{\mathrm{d}\,\mathbf{x_1}} \frac{\mathrm{d}\,\mathbf{x_1}}{\mathrm{d}\,\mathbf{c_1}} \frac{\mathrm{d}\,\mathbf{c_1}}{\mathrm{d}\,\mathbf{d}} + \frac{\mathrm{d}\,\theta_{tip}}{\mathrm{d}\,\mathbf{c_2}} \frac{\mathrm{d}\,\mathbf{c_2}}{\mathrm{d}\,\mathbf{d}} \ . \tag{6.15}$$

Since the bending moments M_z and M_y are independent of the design vector $\mathbf{d}$, and assuming that the external axial loads N_x are zero, the root stress sensitivity is computed as follows:

$$\frac{\mathrm{d}\sigma_x}{\mathrm{d}\mathbf{d}} = M_z \frac{\frac{\mathrm{d}d_y}{\mathrm{d}\mathbf{d}} I_{zz} - \frac{\mathrm{d}I_{zz}}{\mathrm{d}\mathbf{d}} d_y}{I_{zz}^2} + M_y \frac{\frac{\mathrm{d}d_z}{\mathrm{d}\mathbf{d}} I_{yy} - \frac{\mathrm{d}I_{yy}}{\mathrm{d}\mathbf{d}} d_z}{I_{yy}^2} \ . \tag{6.16}$$

Note that the previous expression only has to be evaluated for the cross-section at the root location, since there is no influence of the other cross-section parameters on the root cross-section configuration, being the other sensitivities equal to zero.

If the cross-sectional geometric configuration are forced to be equal at every cross-section, the total sensitivity is obtained by summing the contributions of each individual cross-section. Thus a new design vector ($\mathbf{v}$) will be defined as

$$\mathbf{v} = [x_1/c \ \ x_2/c \ \ t_A \ \ t_{BD} \ \ t_C]^T \ . \tag{6.17}$$

The total sensitivity results are presented in Table 6.6. The signs of the sensitivities may be confirmed with the qualitative analysis performed in Table 5.2, in the previous chapter. Note that a negative sensitivity corresponds to a decrease in the output function value, when the considered variable is increasing. Conversely, a positive sensitivity value represents an increase in the output function. Thus, all the signs of sensitivities match the qualitative behavior previously observed. This may also be confirmed by analyzing the slopes of the lines tangent to the curves at the baseline points represented in the parametric studies figures. For instance, in Figure 5.3a, the slope of the tangent to the curve representing total wing mass as a function of the front spar position is negative, which is coherent with the negative value of $\frac{\mathrm{d}m_{wing}}{\mathrm{d}x_1/c} = -171.40$ kg. As for the rear spar case, the tangent to the curve is positive, corresponding to a positive value of the derivative $\frac{\mathrm{d}m_{wing}}{\mathrm{d}x_1/c} = 90.828$ kg.

Another observable aspect is related to the magnitude of the sensitivity values. The higher the magnitude of the derivative, the steeper the slope of the line tangent to the curve at the baseline point. This is clearly observed in Figure 5.8a when comparing the influence of the front and rear spars in the variation of the tip displacement. For the front spar case, the sensitivity's magnitude ($\left|\frac{\mathrm{d}\delta_{tip}}{\mathrm{d}t_A}\right| = 0.1476$) is greater than for the rear spar case ($\left|\frac{\mathrm{d}\delta_{tip}}{\mathrm{d}t_C}\right| = 0.0789$), meaning that the correspondent slope of the line tangent to the former curve is steeper than the latter's.

Considering the influence of the spars' location in the output functions, the following is noticeable for the baseline case:

- the front spar location has a higher influence in the total wing mass and the tip rotation;
- the rear spar location has a higher influence in the vertical tip displacement and the maximum stress.

For the influence of the thicknesses in the output functions, it is possible to state that:

- the skins thickness have the largest influence on the total wing mass and the tip rotation;
- the front spar thickness have the largest influence on the vertical tip displacement and the maximum stress at the root.

v	$\frac{\mathrm{d}m_{wing}}{\mathrm{d}v}$	$\frac{\mathrm{d}\delta_{tip}}{\mathrm{d}v}$	$\frac{\mathrm{d}\theta_{tip}}{\mathrm{d}v}$	$\frac{\mathrm{d}\sigma_{max}}{\mathrm{d}v}$
x_1/c	-1.7140E+02	-1.0187E-03	5.7000E-03	4.7191E+05
x_2/c	9.0828E+01	5.3706E-03	2.1334E-03	2.4783E+06
t_A	2.8943E+03	-1.4764E-01	-2.3479E-02	-7.1689E+07
t_{BD}	1.9646E+04	-4.3310E-02	-2.4537E-01	-2.8343E+07
t_C	1.5475E+03	-7.8937E-02	1.4222E-02	-3.7224E+07

Table 6.6: Sensitivity of the interest functions with respect to the geometrical wing parameters.

The respective validation of the total sensitivity analysis framework was performed with the forward finite difference method, similarly to the previous sections. The results of the relative errors obtained for the finite difference case, using the total sensitivity calculated with this framework as reference, shown differences smaller than 0.0143%, proving the correct implementation of the framework.

v	$\varepsilon\left(\frac{\mathrm{d}m_{wing}}{\mathrm{d}v}\right)$	$\varepsilon\left(\frac{\mathrm{d}\delta_{tip}}{\mathrm{d}v}\right)$	$\varepsilon\left(\frac{\mathrm{d}\theta_{tip}}{\mathrm{d}v}\right)$	$\varepsilon\left(\frac{\mathrm{d}\sigma_{max}}{\mathrm{d}v}\right)$
x_1/c	2.7824E-07	1.0911E-05	4.8650E-05	1.6062E-05
x_2/c	7.2510E-07	9.2557E-05	1.4254E-04	9.8900E-05
t_A	1.6135E-11	5.8820E-05	4.0720E-05	5.9260E-05
t_{BD}	1.9261E-11	9.8803E-06	9.5253E-05	2.2814E-05
t_C	1.2711E-11	3.3743E-05	1.0699E-05	3.3430E-05

Table 6.7: Relative sensitivity errors of the interest functions with respect to the geometrical wing parameters, when using the forward finite difference method with a relative stepsize of 10^{-4}

With a validated framework that computes exact total sensitivity values, it is now possible to run optimization cases, which will be proved to be efficient in the following chapter.

Chapter 7

Wing Structural Optimization

In this chapter, structural optimization examples will be presented using the structural and sensitivity modules developed in the previous chapters. The internal structure of the wing will be optimized, maintaining the same external shape, so that the aerodynamic forces are assumed to be constant. The total sensitivities computed with the sensitivity analysis framework will be provided to the optimizer, as represented in Figure 7.1, avoiding the need to estimate these values through approximation methods. Firstly, a bound constrained optimization is presented in order to evaluate what occurs when only boundaries on the design vector are imposed to the problem. Secondly, a fully constrained problem is formulated by adding nonlinear constraints. The optimization results are presented and discussed, and a comparison with finite difference methods to estimate the gradient information is performed, in terms of running time, number of iterations and number of function evaluations.

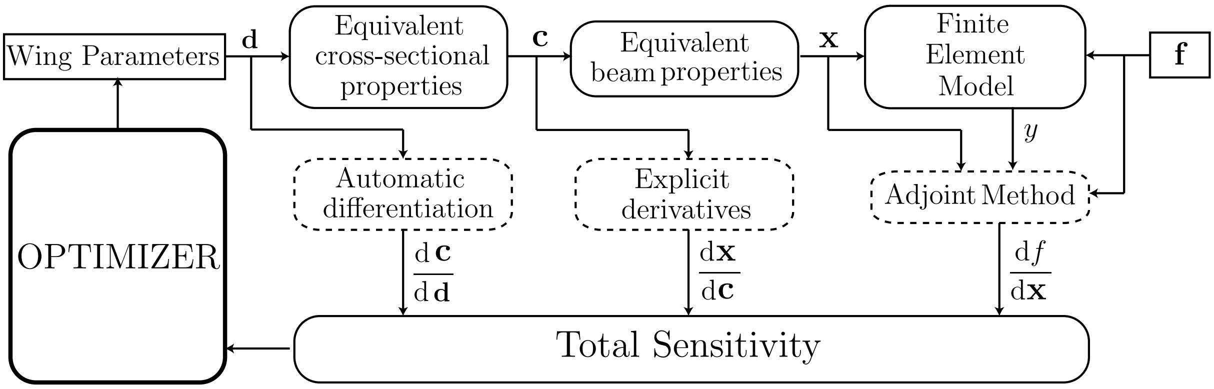

Wing Parameters
d
Equivalent cross-sectional properties
c
Equivalent beam properties
x
Finite Element Model
f
y
OPTIMIZER
Automatic differentiation
Explicit derivatives
Adjoint Method
$\frac{d\mathbf{c}}{d\mathbf{d}}$
$\frac{d\mathbf{x}}{d\mathbf{c}}$
$\frac{df}{d\mathbf{x}}$
Total Sensitivity

Figure 7.1: Structural optimization framework.

7.1 Optimizer

In order to solve the optimization problems, the MATLAB® Optimization Toolbox was chosen to be used in this thesis. The appropriate function in the toolbox to solve nonlinear constrained multivariable optimization problems is called `fmincon`. The problems must be formulated as

$$\begin{aligned} &\underset{\mathbf{x}}{\text{minimize}} && f(\mathbf{x}) \\ &\text{subject to} && [A]\,\mathbf{x} \leq \mathbf{b},\ \ [A_{eq}]\,\mathbf{x} = \mathbf{b_{eq}}\ , \\ &\text{and} && c(\mathbf{x}) \leq 0,\ \ c_{eq}(\mathbf{x}) = 0 \\ &\text{and} && \mathbf{lb} \leq \mathbf{x} \leq \mathbf{ub} \end{aligned} \qquad (7.1)$$

where $[A]$ and $[A_{eq}]$ are, respectively, matrices for linear inequality and equality constraints, $\mathbf{b}$ and $\mathbf{b_{eq}}$ are vectors for the linear inequality and equality constraints, and c and c_{eq} are the functions representing the nonlinear inequality and equality constraints, respectively. The vectors $\mathbf{lb}$ and $\mathbf{ub}$ correspond to the lower and upper bounds of the design vector. This information is used as an argument of `fmincon` as

```
fmincon(@f,x0,A,b,Aeq,beq,lb,ub,@nonlcon,options) ,
```

where `@f` corresponds to an implemented function that must output the value of the objective function, and its derivative information (if desired to be used) and `@nonlcon`, corresponds to a function that outputs the values of $c(x)$, $c_{eq}(x)$, as well as the respective sensitivity information.

To set the `fmincon` options, it is necessary to use the function `optimoptions`. For example:

```
optimoptions('fmincon','Algorithm','sqp','Gradobj','on','GradConstr','on') .
```

The first entry in `optimoptions` corresponds to specifying the solver name, which is `'fmincon'`, in this case. Then, other options should be specified such as the type of the algorithm used (`'sqp'`) and whether or not the gradient information of the objective function (`'Gradobj'`) and constraints (`'GradConstr'`) are provided. Note that these options may have different tag specifiers depending on the MATLAB® version. The previous example is defined according to the MATLAB® R2014a version.

`fmincon` supports four different algorithms: `'interior-point'` (default), `'sqp'`, `'active-set'` and `'trust-region-reflective'`. The latter is immediately excluded since is quite restrictive, allowing only specification of design bounds or linear equality constraints (but not both at the same time). The 'interior-point' is the algorithm suggested by MATLAB® to be firstly used. However, in order to obtain more speed on small- to medium-sized problems, MATLAB® suggests the user to try `'sqp'` and `'active-set'`. After some test cases, the `'sqp'` algorithm was found to be the most appropriate, leading to faster and better results. Therefore, this algorithm will be used throughout the optimization cases in the following sections. Note that the remaining default options are unchanged, which include the stopping criterion, where the tolerance of the function value is kept equal to 10^{-6}.

7.2 Bound Constrained Optimization

Once the optimization function is defined and the algorithm is chosen with the options all set, some optimization examples can be studied. One of the simplest optimization cases that may be applied to the developed framework corresponds to a bound constrained mass minimization. This problem may be mathematically described as

$$\begin{array}{ll} \underset{\mathbf{v}}{\text{minimize}} & m_{wing}(\mathbf{v}) \\ \text{subject to} & \mathbf{v}^L \leq \mathbf{v} \leq \mathbf{v}^U \end{array} , \tag{7.2}$$

where m_{wing} represents the wing mass and can be interpreted as a function of the design vector $\mathbf{v}$, which is formed by the design variables x_1/c, x_2/c, t_A, t_{BD} and t_C, as defined in previous chapters. The arbitrarily chosen upper and lower bounds of the design vector, as well as the initial design point, are summarized in Table 7.1.

Design vector ($\mathbf{v}$)	Initial design ($\mathbf{v_0}$)	Lower bound ($\mathbf{v^L}$)	Upper bound ($\mathbf{v^U}$)
x_1/c [-]	0.25	0.20	0.40
x_2/c [-]	0.75	0.60	0.80
t_A [mm]	20.00	1.00	30.00
t_{BD} [mm]	5.00	1.00	8.00
t_C [mm]	20.00	1.00	30.00

Table 7.1: Inital values and bounds of the design variables for the bounded optimization problem.

After defining the problem, the optimization algorithm is run until a converged solution is obtained, which occurs after 3 iterations. The history of the objective function is represented in Figure 7.2.

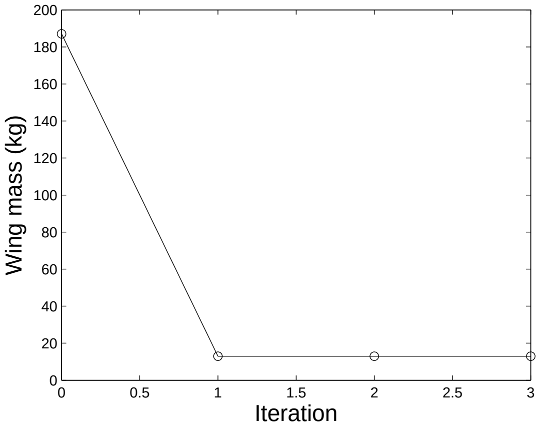


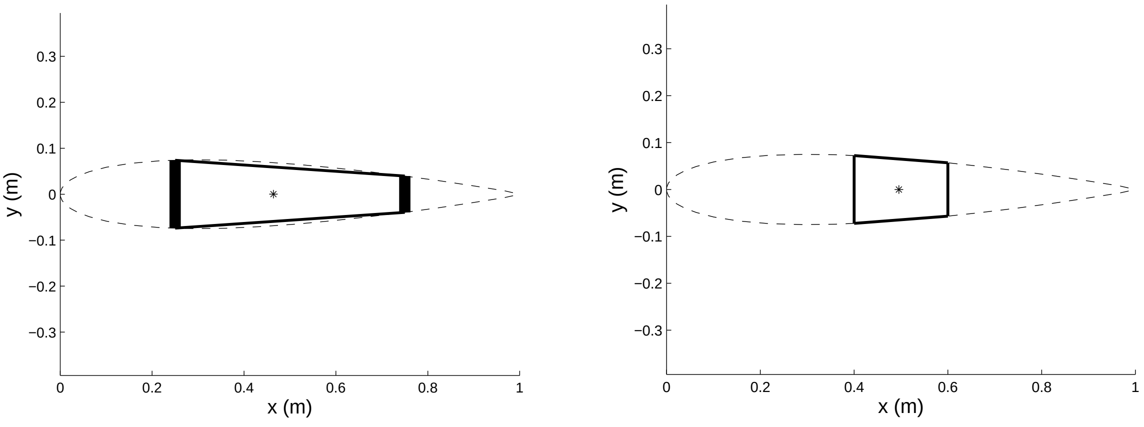
Figure 7.2: Convergence of the objective function (m_{wing}).

The results of the optimization problem are summarized in Table 7.2 and the wing internal structure is represented in Figure 7.3. It is possible to observe that the optimized configuration tends to approximate the spars as close as possible from each other and also decreases thicknesses to the lower bound values. With this optimization, the total wing mass is reduced from 187.006 kg to 12.929 kg. However, the vertical tip displacement (δ_{tip}), tip rotation (θ_{tip}) and maximum stress at the root σ_{max}, all increase at least one order of magnitude.

Design variables	Initial	Optimized	Output functions	Initial	Optimized
x_1/c [-]	0.25	0.40	m_{wing} [kg]	187.066	12.929
x_2/c [-]	0.75	0.60	δ_{tip} [mm]	4.748	80.267
t_A [mm]	20	1	θ_{tip} [º]	0.0809	1.4124
t_{BD} [mm]	5	1	σ_{max} [Pa]	2.320E+06	3.7816E+07
t_C [mm]	20	1			

Table 7.2: Initial and final optimized design variable values and output functions for the bounded constrained problem.

A benchmark regarding the total optimization time, the number of iterations and the number of function evaluations is presented in Table 7.3. The optimum point and the respective output functions are omitted since all methods lead to the same optimum, corresponding to the same output values. The



(a) Before optimization.

(b) After optimization

Figure 7.3: Change in the internal structural configuration resultant from optimization.

number of iterations is the same for the three methods, but the number of function evaluations and the execution time differs on the method chosen to compute gradients. The developed sensitivity framework proves to be the fastest and only requires the computation of the objective function 4 times (one for the initial point, and one for each iteration). The forward finite differences method is less computationally efficient than the sensitivity framework since the running time is approximately 1.7 times higher than the first case for achieving the same results, being the total number of function evaluation 6 times higher than the former method. For the initial point and for each iteration, it is necessary to perform 6 objective function evaluations: one for the current design point and 5 for the design perturbations (one perturbation per design variable). The central differences method is the most costly in terms of computational time, requiring 11 objective function evaluations per iteration: one for the current design point, 5 perturbations in the positive direction and 5 in the negative direction.

Gradient calculation method	Time [s]	No. Iterations	Function Evaluations
Sensitivity Framework	3.81	3	4
Forward Finite Difference	6.42	3	24
Central Finite Difference	12.29	3	44

Table 7.3: Benchmark of excecution time, iteration numbers and function evaluations for three different sensitivity calculation methods for the bounded optimization problem, using an Intel(R) Core(TM) i5-4200U CPU @ 2.30Ghz.

7.3 Fully Constrained Optimization

As a second optimization example, a fully constrained optimization problem shall be considered. This designation was adopted to refer to optimization problems that present not only bound constraints on the design vector but also equality and/or inequality constraint functions dependent on the design vector.

Based on the results obtained for the previous optimization case (recall Table 7.2), some constraints on the output functions will be introduced to build the new optimization example. Suppose that the tip vertical displacement (δ_{tip}) shall not exceed 50 mm and that the tip rotation shall be less or equal than 1^{o}. Also considering the factor of safety ($F.S.$) definition [52],

$$F.S. = \frac{\sigma_{ult}}{\sigma_{al}} \ , \tag{7.3}$$

where σ_{ult} represents the ultimate stress and σ_{al} the allowable stress. According to the FAR Regulations [53], a safety factor of 1.5 is chosen for the wing structure. Since the maximum stress (σ_{max}) computed for the 3D beam corresponds to the axial stresses due to bending, the ultimate tensile strength of Aluminum 2024-T3 [54], which is equal to 483 MPa, will be considered as σ_{ult}. This implies that the allowable stress is equal to 322 MPa and, therefore, this value shall not be exceeded. The formal mathematical formulation of the constrained problem is given by

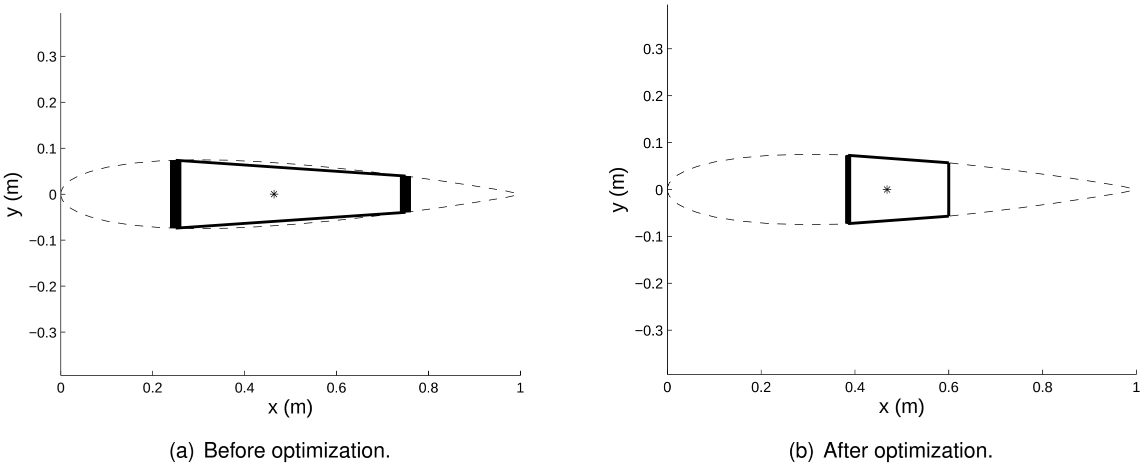
$$\begin{aligned} \underset{\mathbf{v}}{\text{minimize}} \quad & m_{wing}(\mathbf{v}) \\ \text{subject to} \quad & \delta_{tip}(\mathbf{v}) - 0.05 \leq 0 \ , \\ & \theta_{tip}(\mathbf{v}) - \pi/180 \leq 0 \ , \\ & \sigma_{max}(\mathbf{v}) - 322 \times 10^6 \leq 0 \ , \\ & \mathbf{v}^L \leq \mathbf{v} \leq \mathbf{v}^U \end{aligned} \tag{7.4}$$

The bounds are assumed to be equal to the ones in the previous problem, represented in Table 7.1.

The initial and final results in the optimization problems are summarized in Table 7.4. It can be observed that the wing mass decreased from 187.006 kg to 16.795 kg. This value is slightly higher than the result obtained in the bound constrained optimization case, but it is the cost of restricting the design by applying nonlinear equality and inequality constraints. The vertical tip displacement and tip rotation have tended to the upper limits of the constraint, being equal to the defined maximum values. The maximum stress falls below the maximum stress limit. In the new wing configuration, the rear spar was moved to the lower bound and the front spar was shifted towards the trailing edge. All the specified thicknesses decreased and the skin thickness was equal to the lower bound. The resultant configuration is represented in Figure 7.4. The convergence history of the objective function and constraints are represented in Figure 7.5.

Design variables	Initial	Optimized	Output functions	Initial	Optimized
x_1/c [-]	0.25	0.3856	m_{wing} [kg]	187.066	16.795
x_2/c [-]	0.75	0.60	δ_{tip} [mm]	4.748	50.000
t_A [mm]	20	2.1199	θ_{tip} [º]	0.0809	1.000
t_{BD} [mm]	5	1	σ_{max} [Pa]	2.320E+06	2.38634E+07
t_C [mm]	20	1.0324			

Table 7.4: Initial and final optimized design variable values and output functions for the fully constrained problem.



(a) Before optimization.

(b) After optimization.

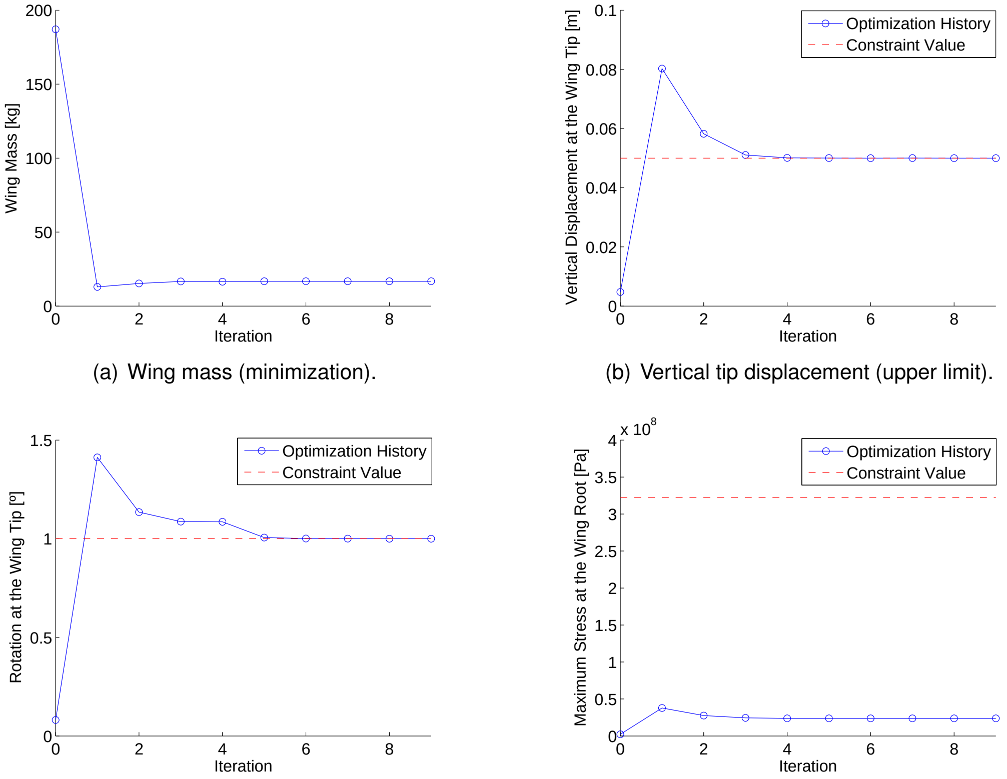
Figure 7.4: Internal structural configurations before and after the fully constrained optimization problem.

Similarly to the previous section, the results obtained using the sensitivity framework were benchmarked with the forward and central finite difference methods (Table 7.5). The optimal configurations were the same in the three cases, however, the sensitivity framework was again proven to be the most efficient, requiring roughly half the time when compared to the forward finite difference and approximately one third of the time when compared to the central finite difference. The number of function evaluations is consistent with what was verified in the previous section.

As a last note, regard that even though the overall number of design variables is not much greater than the number of functions of interest, the sensitivity framework presented more efficient results than the gradient estimation through finite differences. It is expected that, for a larger number of design variables, this framework becomes even more efficient. Besides that, the aerodynamic vector was externally loaded and is not being reevaluated with the aerodynamic panel method at every design perturbation. Thus, if the full aerodynamic code was run for each design perturbation, the advantages in overall computational time would be even more noticeable with the sensitivity framework.

Gradient calculation method	Time [s]	No. iterations	Function evaluations
Sensitivity framework	16.58	9	10
Forward finite difference	28.21	9	60
Central finite difference	46.78	9	110

Table 7.5: Benchmark of excecution time, iteration numbers and function evaluations for three different sensitivity calculation methods, for the fully constrained optimization problem, using an Intel(R) Core(TM) i5-4200U CPU @ 2.30Ghz.



(a) Wing mass (minimization).

(b) Vertical tip displacement (upper limit).

(c) Tip rotation (upper limit).

(d) Maximum stress at the wing root (upper limit).

Figure 7.5: Convergence history of the objective function and constraints.

Chapter 8

Conclusions

8.1 Achievements

The main objective of this thesis was to develop an efficient optimization tool for wing structural design, relying on gradient-based information to find the best solution and to be applied in an existent aero-structural framework.

Firstly, a review on the importance of the application of Multidisciplinary Design Optimization in the aeronautical sector was presented, showing the potential in using gradient-based methods to produce new designs. These methods require the evaluation of sensitivity information, which was found to be one of the most expensive steps in current gradient-based optimization tools, since these are frequently obtained by approximation methods that are highly dependent on the number of design variables. Therefore, a survey on the existent sensitivity analysis methods was conducted to investigate alternative more efficient methods. The methods found were applied to a beam design problem modeled with finite elements, providing valuable insight into the implementation process and advantages of each method, to be latter applied to the wing structural case.

The wing structural model in Almeida's framework [11] was reformulated. The new model is composed by three sequential structural blocks, each having its correspondent exact sensitivity calculation module.

The first structural block computes the structural properties at a given section of the wing based on a set of geometric parameters and material properties and was redeveloped using a simplified wingbox approximation model, in order to be compatible with the automatic differentiation method, which was found to be the most suitable exact sensitivity method. The finite difference method was utilized to benchmark the sensitivities obtained with the algorithmic differentiation method, being verified a relative error of 0.118% for the worst case. Even though the latter method was found to be slightly faster than the former however, the resultant sensitivities were not exact.

The second structural block calculates the average properties of each finite element based on the respective boundary nodal values. For this case, the analytic symbolic differentiation was employed to compute the intermediate sensitivities. Since the explicit expressions were available and trivial, no numerical validation was performed.

The third block generates the equivalent 3D-beam element model, which is necessary to determine the wing's structural response with an applied load condition. The adjoint method was chosen for calculating this module's sensitivities since the number of intermediate design variables (average finite element properties and shear center locations) was greater than the number of interest functions dependent of the structural solution. Hence, being practically independent on the number of intermediate design variables, this method represented a clear advantage in terms of computational efficiency face the other methods. When benchmarked with the forward finite difference method, the adjoint method proved to be 100 times faster in calculating the same intermediate gradient. Both methods presented an overall agreement in sensitivity values, validating the adjoint module.

The total exact sensitivities of the interest functions with respect to the design variables were then computed using the chain rule of differentiation. The qualitative behavior coincided with the results observed in the developed parametric studies and the quantitative validation was performed with forward finite differences, proving the correct implementation of the framework.

Lastly, structural optimization examples demonstrated convergence to better designs, in terms of objective function values, satisfying the imposed constraints and using the total sensitivities calculated with the sensitivity analysis framework. For the bounded constrained optimization case the total wing mass was reduced from 187.066kg to 12.929kg and for the fully constrained case a reduction from 187.066kg to 16.795 was verified. Note that the main objective in this work were not the optimization results *per se*, but the efficiency in applying the developed sensitivity analysis framework, which was clearly noticeable with the reduction of the computational time for almost half, when compared to using the forward finite difference methods for gradient estimation, and for roughly a third, when using the central differences.

8.2 Future Work

During the development of this framework, several ideas that could be implemented and tested in the scope of this project arose.

The first idea concerns the implementation of a similar sensitivity analysis tool for the aerodynamic module of the framework. Since the aerodynamic analysis is performed by using a panel code that solves a linear system of equations, an analogous adjoint formulation may be derived for the aerodynamic module. Furthermore, it will also be necessary to treat the sensitivities of the coupling aero-structural terms, which were not addressed in this thesis.

The second suggestion is related to the complexity of the structural representation of the wing. The current structural representation is obtained with a simplified wingbox model. A further development in the complexity of this model, including taking into account the remaining parts of the wing, may be explored. Another possibility would be to develop a higher fidelity finite element model using, for instance, shell elements to better describe the wing stresses.

Finally, the last idea is to increase the number of design variables in the optimization problem. Instead of maintaining a constant cross-section throughout the wing, different cross-sectional configurations along the span can be studied by adding more design variables. This includes, for instance, a study involving a variation in wingbox thickness distribution or investigating different relative spar positions to simulate the effect of curvilinear spars. The framework is prepared to deal with these variable spanwise cross-sectional structural variations, and the presented ideas may be carried out with a small implementation effort.

Bibliography

[1] Airbus Global Market Forecast (GMF). Flying by numbers 2015 - 2034, August 2015.

[2] O. Fichet. Lean design and development. Presentation from LEAN Ontario, Fujitsu, March 2015.

[3] J. Sobieszczanski-Sobieski and R. T. Haftka. Multidisciplinary aerospace design optimization: survey of recent developments. *Structural Optimization*, 14(1):1–23, 1997. ISSN 0934-4373. doi: 10.1007/BF01197554.

[4] N. Nikolova, R. Safian, E. Soliman, M. Bakr, and J. Bandler. Accelerated gradient based optimization using adjoint sensitivities. *IEEE Trans. Antennas Propagat.*, 52(8):2147–2157, August 2004.

[5] M. Nemec and M. Aftosmis. Adjoint algorithm for CAD-based shape optimization using a cartesian method. In *17th AIAA Computational Fluid Dynamics Conference*. American Institute of Aeronautics and Astronautics (AIAA), June 2005.

[6] M. Akgün, J. H. Garcelon, K. C. Wu, and J. L. Walsh. Efficient structural optimization for multiple load cases using adjoint sensitivities. *AIAA Journal*, 39(3), March 2001.

[7] J. S. Jensen, P. B. Nakshatrala, and D. A. Tortorelli. On the consistency of adjoint sensitivity analysis for structural optimization of linear dynamic problems. *Structural and Multidisciplinary Optimization*, 49(5):831–837, November 2013. doi: 10.1007/s00158-013-1024-4.

[8] G. J. Kennedy and J. S. Hansen. The hybrid-adjoint method: a semi-analytic gradient evaluation technique applied to composite cure cycle optimization. *Optimization and Engineering*, 11(1):23–43, November 2008. doi: 10.1007/s11081-008-9068-9. URL `http://dx.doi.org/10.1007/s11081-008-9068-9`.

[9] J. Luo, J. Xiong, and F. Liu. Aerodynamic design optimization by using a continuous adjoint method. *Science China Physics, Mechanics and Astronomy*, 57(7):1363–1375, May 2014. doi: 10.1007/s11433-014-5479-0. URL `http://dx.doi.org/10.1007/s11433-014-5479-0`.

[10] J. R. R. A. Martins, J. J. Alonso, and J. Reuther. Aero-structural wing design optimization using high-fidelity sensitivity analysis. In *In Proceedings — CEAS Conference on Multidisciplinary Aircraft Design Optimization*, pages 211–226, 2001.

[11] J. Almeida. Structural dynamics for aeroelastic analysis. Master's thesis, Técnico Lisboa, October 2015.

[12] S. S. Rao. *Engineering Optimization: Theory and Practice*. John Wiley Sons, Inc., 2009.

[13] J. Hicken, J. Alonso, and C. Farhat. Introduction to multidisciplinary design optimization. Accessed in October 2015.

[14] A. C. Marta. Multidisciplinary design optimization of aircrafts - class notes, 2015.

[15] D. W. Zingg, M. Nemec, and T. H. Pulliam. A comparative evaluation of genetic and gradient-based algorithms applied to aerodynamic optimization. *Revue européenne de mécanique numérique*, 17 (1-2):103–126, 2008. ISSN 17797179. doi: 10.3166/remn.17.103-126.

[16] A.-L. Cauchy. Méthode générale pour la résolution des systèmes d'équations simultanées [Translated: (2010)]. *Compte Rendu des S'eances de L'Acad'emie des Sciences*, 25(2):536–538, 1847.

[17] R. Fletcher. Function minimization by conjugate gradients. *The Computer Journal*, 7(2):149–154, February 1964.

[18] R. Idema, D. Lahaye, and C. Vuik. On the convergence of inexact newton methods. In *Lecture Notes in Computational Science and Engineering*, pages 355–363. Springer Science Business Media, October 2014.

[19] F. S. Sisser. A modified newton's method for minimizing factorable functions. *Journal of Optimization Theory and Applications*, 38(4):461–482, December 1982.

[20] J. Greenstadt. On the relative efficiencies of gradient methods. *Mathematics of Computation*, 21 (99):360–360, September 1967.

[21] Y. xiang Yuan. Recent advances in trust region algorithms. *Mathematical Programming*, 151(1): 249–281, March 2015. doi: 10.1007/s10107-015-0893-2.

[22] A. D. Belegundu and T. R. Chandrupatla. *Optimization Concepts and Applications in Engineering*. Cambridge University Press, 2^{nd} edition, 2011. ISBN:978-0-521-87846-3.

[23] G. Zoutendijk. *Methods of feasible directions: a study in linear and nonlinear programming*. Elsevier Pub. Co., 1960.

[24] P. E. Gill and E. Wong. Sequential quadratic programming methods. In *Mixed Integer Nonlinear Programming*, pages 147–224. Springer Science Business Media, November 2011. doi: 10.1007/978-1-4614-1927-3_6. URL `http://dx.doi.org/10.1007/978-1-4614-1927-3_6`.

[25] M. Melanie. *An Introduction to Genetic Algorithms*. The MIT Press, 1999.

[26] R. Eberhart and J. Kennedy. A new optimizer using particle swarm theory. *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, pages 39–43, 1995. doi: 10.1109/MHS.1995.494215.

[27] K. Zielinski, D. Peters, and R. Laur. Stopping criteria for single objective optimization. *Proceedings of the Third International Conference on Computational Intelligence, Robotics and Autonomous Systems*, 2005.

[28] M. Dorigo and T. Stützle. *Ant Colony Optimization*. The MIT Press, 2004.

[29] K. K. Choi and N.-H. Kim. *Structural Sensitivity Analysis and Optimization 1 - Linear Systems*. Springer, 2005.

[30] D. M. H. Chaudhry. *Open-Channel Flow*, chapter Finite-Difference Methods, pages 367–405. Springer US, Boston, MA, 2008. ISBN 978-0-387-68648-6.

[31] J. N. Lyness. Numerical algorithms based on the theory of complex variable. In *Proceedings of the 1967 22nd national conference*. Association for Computing Machinery (ACM), 1967. doi: 10.1145/800196.805983. URL `http://dx.doi.org/10.1145/800196.805983`.

[32] W. Squire and G. Trapp. Using complex variables to estimate derivatives of real functions. *SIAM Rev.*, 40(1):110–112, January 1998. doi: 10.1137/s003614459631241x.

[33] K.-L. Lai and J. Crassidis. Generalizations of the complex-step derivative approximation. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*. American Institute of Aeronautics and Astronautics (AIAA), August 2006. doi: 10.2514/6.2006-6348. URL `http://dx.doi.org/10.2514/6.2006-6348`.

[34] J. Martins. Multidisciplinary design optimization: sensitivity analysis handouts, 2000.

[35] J. R. R. A. Martins. *A Coupled-Adjoint Method for High Fidelity Aero-structural Optimization*. PhD thesis, Stanford University, October 2002.

[36] C. Bischof, P. Hovland, and B. Norris. On the implementation of automatic differentiation tools. *Higher-Order and Symbolic Computation*, 21(3):311–331, 2008. ISSN 1388-3690. doi: 10.1007/s10990-008-9034-4.

[37] H. Berland. Automatic differentiation. Department of Mathematical Sciences, NTNU, September 2006.

[38] R. J. Hogan. Fast reverse-mode automatic differentiation using expression templates in c++. *ACM Trans. Math. Softw.*, 40(4):26:1–26:16, 2014. ISSN 0098-3500. doi: 10.1145/2560359.

[39] C. H. Bischof, H. M. Bücker, B. Lang, A. Rasch, and A. Vehreschild. Combining source transformation and operator overloading techniques to compute derivatives for MATLAB programs. In *Proceedings of the Second IEEE International Workshop on Source Code Analysis and Manipulation*, pages 65–72. IEEE Computer Society, 2002. doi: 10.1109/SCAM.2002.1134106.

[40] R. Archer and T. Lardner. *An Introduction to the mechanics of solids*. McGraw-Hill series in engineering sciences. McGraw-Hill, 1978. ISBN 9780070134416.

[41] J. Reddy. *An Introduction to the Finite Element Method*. McGraw-Hill Education (ISE Editions), 3^{rd} edition, 2005. ISBN 0071244735,9780071244732.

[42] C. Larrosa. Thin-walled beams in advanced aerospace structures and materials - lecture notes, 2014.

[43] T. H. G. Megson. *Aircraft Structures for Engineering Students*. Elsevier, 4^{th} edition, 2007.

[44] I. ANSYS. *Theory Reference for the Mechanical APDL and Mechanical Applications*. SAS IP, Inc., 12^{th} edition, 2009.

[45] J. N. Reddy. *Mechanics of Laminated Composite Plates and Shells: Theory and Analysis, Second Edition*. CRC Press, 2^{nd} edition, 2003. ISBN 0849315921,9780849315923.

[46] A. E. Armenàkas. *Advanced Mechanics of Materials and Applied Elasticity*. CRC Press, 2013. ISBN 978-1-4200-5777-5,1420057774.

[47] R. D. Cook, D. S. Malkus, M. E. Plesha, and R. J. Witt. *Concepts and Applications of Finite Element Analysis, 4th Edition*. Wiley, 2001. ISBN 0471356050.

[48] N. Ottosen and H. Petersson. *Introduction to the Finite Element Method*. Prentice Hall, 1992. ISBN 0134738772,9780134738772.

[49] J. W. Stærdahl. Finite element methods: 3D beam element - lecture notes, 2008.

[50] A. Cardeira. Aeroelastic analysis of aircraft wings. Master's thesis, Instituto Superior Técnico (IST), December 2014.

[51] T. Corke. *Design of Aircraft*. Prentice Hall, 2003. ISBN 9780130892348.

[52] F. Beer, E. Jr. Johnston, J. DeWolf, and D. Mazurek. *Mechanics of Materials*. McGraw-Hill Education, 2011. ISBN 9780073380285.

[53] Code of federal regulations, Title 14, PART 25—AIRWORTHINESS STANDARDS: TRANSPORT CATEGORY AIRPLANES, Subpart C — Structure, 25.303 Factor of safety, April 1970.

[54] Aerospace specification metals inc. `http://asm.matweb.com/search/SpecificMaterial.asp?bassnum=MA2024T3`. Accessed: April 2016.

Appendix A

Automatic Sensitivity Analysis Benchmark with Finite Difference Methods

This appendix includes two extra sensitivity analysis benchmarks of the automatic differentiation method with the finite difference methods, using relative step sizes of 10^{-6} and 10^{-8}, respectively:

Var. (x)	$\varepsilon\left(\frac{\partial EA}{\partial x}\right)$	$\varepsilon\left(\frac{\partial EI_{zz}}{\partial x}\right)$	$\varepsilon\left(\frac{\partial EI_{yy}}{\partial x}\right)$	$\varepsilon\left(\frac{\partial GJ}{\partial x}\right)$	$\varepsilon\left(\frac{\partial \rho_l}{\partial x}\right)$	$\varepsilon\left(\frac{\partial SC_x}{\partial x}\right)$	$\varepsilon\left(\frac{\partial SC_y}{\partial x}\right)$
c	5.496E-11	6.614E-07	1.128E-07	1.000E-06	8.307E-11	3.185E-10	4.192E-03
x_1/c	2.904E-09	1.153E-07	4.460E-10	1.664E-07	3.111E-09	3.636E-08	2.231E-04
x_2/c	6.920E-09	4.026E-07	7.860E-08	1.185E-05	7.052E-09	2.953E-07	1.951E-02
t_A	1.798E-10	3.095E-07	4.528E-11	9.670E-07	2.862E-10	3.183E-07	2.437E-02
t_B	2.557E-12	3.586E-09	9.552E-08	5.267E-07	2.480E-11	3.146E-07	5.939E-07
t_C	3.426E-10	1.654E-07	9.093E-11	9.759E-07	7.679E-10	1.690E-07	6.963E-03
t_D	2.635E-13	3.782E-09	9.085E-08	5.267E-07	1.450E-10	1.837E-07	5.961E-07
ρ	N/A	N/A	N/A	N/A	8.307E-11	N/A	N/A
E	5.495E-11	7.750E-12	4.104E-11	N/A	N/A	N/A	N/A
G	N/A	1.000E+00	1.000E+00	5.586E-12	N/A	1.000E+00	1.197E+06

Table A.1: Absolute values of the error of the sensitivities calculated computed for a step size of 10^{-6}

Var. (x)	$\varepsilon\left(\frac{\partial EA}{\partial x}\right)$	$\varepsilon\left(\frac{\partial EI_{zz}}{\partial x}\right)$	$\varepsilon\left(\frac{\partial EI_{yy}}{\partial x}\right)$	$\varepsilon\left(\frac{\partial GJ}{\partial x}\right)$	$\varepsilon\left(\frac{\partial \rho_l}{\partial x}\right)$	$\varepsilon\left(\frac{\partial SC_x}{\partial x}\right)$	$\varepsilon\left(\frac{\partial SC_y}{\partial x}\right)$
c	8.049E-09	3.066E-09	3.366E-09	1.878E-08	5.933E-09	2.216E-08	3.479E-01
x_1/c	4.798E-08	3.486E-08	2.976E-07	6.052E-09	2.110E-08	1.283E-07	1.297E-01
x_2/c	1.973E-08	1.923E-08	1.283E-08	1.316E-07	2.945E-08	6.458E-08	1.245E+00
t_A	5.669E-08	1.166E-09	2.871E-09	4.507E-07	5.729E-09	1.849E-07	2.891E+00
t_B	5.645E-08	4.057E-09	2.677E-07	1.321E-08	3.040E-08	3.069E-06	1.223E-07
t_C	6.577E-08	2.383E-08	9.723E-09	3.292E-07	9.237E-08	2.235E-07	2.080E+00
t_D	5.835E-08	1.081E-08	3.330E-07	1.130E-08	2.850E-08	8.580E-07	7.351E-08
ρ	N/A	N/A	N/A	N/A	5.933E-09	N/A	N/A
E	8.049E-09	2.178E-09	4.104E-11	N/A	N/A	N/A	N/A
G	N/A	1.000E+00	1.000E+00	1.025E-08	N/A	1.000E+00	1.197E+08

Table A.2: Absolute value of the sensitivities errors using the forward finite difference method with a relative step size of 10^{-8}.

Appendix B

Explicit Derivatives of the Element Stiffness Matrices

This appendix includes the explicit derivatives of the element stiffness matrices with respect to the axial rigidity EA, to the torsional rigidity GJ and to the bending rigidities EI_{zz} and EI_{yy}, used in Chapter 6.3. The notation $c_\square$ and $s_\square$ is used to represent the cosine and sine of an angle $\square$, respectively.

$$\frac{\mathrm{d}K}{\mathrm{d}EA} = \begin{bmatrix}
\frac{{c_\Gamma}^2\,{c_\Lambda}^2}{L} & \frac{{c_\Gamma}^2\,c_\Lambda\,s_\Lambda}{L} & \frac{c_\Gamma\,c_\Lambda\,s_\Gamma}{L} & 0 & 0 & 0 & -\frac{{c_\Gamma}^2\,{c_\Lambda}^2}{L} & -\frac{{c_\Gamma}^2\,c_\Lambda\,s_\Lambda}{L} & -\frac{c_\Gamma\,c_\Lambda\,s_\Gamma}{L} & 0 & 0 & 0 \\
\frac{{c_\Gamma}^2\,c_\Lambda\,s_\Lambda}{L} & \frac{{c_\Gamma}^2\,{s_\Lambda}^2}{L} & \frac{c_\Gamma\,s_\Gamma\,s_\Lambda}{L} & 0 & 0 & 0 & -\frac{{c_\Gamma}^2\,c_\Lambda\,s_\Lambda}{L} & -\frac{{c_\Gamma}^2\,{s_\Lambda}^2}{L} & -\frac{c_\Gamma\,s_\Gamma\,s_\Lambda}{L} & 0 & 0 & 0 \\
\frac{c_\Gamma\,c_\Lambda\,s_\Gamma}{L} & \frac{c_\Gamma\,s_\Gamma\,s_\Lambda}{L} & \frac{{s_\Gamma}^2}{L} & 0 & 0 & 0 & -\frac{c_\Gamma\,c_\Lambda\,s_\Gamma}{L} & -\frac{c_\Gamma\,s_\Gamma\,s_\Lambda}{L} & -\frac{{s_\Gamma}^2}{L} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-\frac{{c_\Gamma}^2\,{c_\Lambda}^2}{L} & -\frac{{c_\Gamma}^2\,c_\Lambda\,s_\Lambda}{L} & -\frac{c_\Gamma\,c_\Lambda\,s_\Gamma}{L} & 0 & 0 & 0 & \frac{{c_\Gamma}^2\,{c_\Lambda}^2}{L} & \frac{{c_\Gamma}^2\,c_\Lambda\,s_\Lambda}{L} & \frac{c_\Gamma\,c_\Lambda\,s_\Gamma}{L} & 0 & 0 & 0 \\
-\frac{{c_\Gamma}^2\,c_\Lambda\,s_\Lambda}{L} & -\frac{{c_\Gamma}^2\,{s_\Lambda}^2}{L} & -\frac{c_\Gamma\,s_\Gamma\,s_\Lambda}{L} & 0 & 0 & 0 & \frac{{c_\Gamma}^2\,c_\Lambda\,s_\Lambda}{L} & \frac{{c_\Gamma}^2\,{s_\Lambda}^2}{L} & \frac{c_\Gamma\,s_\Gamma\,s_\Lambda}{L} & 0 & 0 & 0 \\
-\frac{c_\Gamma\,c_\Lambda\,s_\Gamma}{L} & -\frac{c_\Gamma\,s_\Gamma\,s_\Lambda}{L} & -\frac{{s_\Gamma}^2}{L} & 0 & 0 & 0 & \frac{c_\Gamma\,c_\Lambda\,s_\Gamma}{L} & \frac{c_\Gamma\,s_\Gamma\,s_\Lambda}{L} & \frac{{s_\Gamma}^2}{L} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

$$\frac{\mathrm{d}K}{\mathrm{d}GJ} = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{{c_\Gamma}^2\,{c_\Lambda}^2}{L} & \frac{{c_\Gamma}^2\,c_\Lambda\,s_\Lambda}{L} & \frac{c_\Gamma\,c_\Lambda\,s_\Gamma}{L} & 0 & 0 & 0 & -\frac{{c_\Gamma}^2\,{c_\Lambda}^2}{L} & -\frac{{c_\Gamma}^2\,c_\Lambda\,s_\Lambda}{L} & -\frac{c_\Gamma\,c_\Lambda\,s_\Gamma}{L} \\
0 & 0 & 0 & \frac{{c_\Gamma}^2\,c_\Lambda\,s_\Lambda}{L} & \frac{{c_\Gamma}^2\,{s_\Lambda}^2}{L} & \frac{c_\Gamma\,s_\Gamma\,s_\Lambda}{L} & 0 & 0 & 0 & -\frac{{c_\Gamma}^2\,c_\Lambda\,s_\Lambda}{L} & -\frac{{c_\Gamma}^2\,{s_\Lambda}^2}{L} & -\frac{c_\Gamma\,s_\Gamma\,s_\Lambda}{L} \\
0 & 0 & 0 & \frac{c_\Gamma\,c_\Lambda\,s_\Gamma}{L} & \frac{c_\Gamma\,s_\Gamma\,s_\Lambda}{L} & \frac{{s_\Gamma}^2}{L} & 0 & 0 & 0 & -\frac{c_\Gamma\,c_\Lambda\,s_\Gamma}{L} & -\frac{c_\Gamma\,s_\Gamma\,s_\Lambda}{L} & -\frac{{s_\Gamma}^2}{L} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -\frac{{c_\Gamma}^2\,{c_\Lambda}^2}{L} & -\frac{{c_\Gamma}^2\,c_\Lambda\,s_\Lambda}{L} & -\frac{c_\Gamma\,c_\Lambda\,s_\Gamma}{L} & 0 & 0 & 0 & \frac{{c_\Gamma}^2\,{c_\Lambda}^2}{L} & \frac{{c_\Gamma}^2\,c_\Lambda\,s_\Lambda}{L} & \frac{c_\Gamma\,c_\Lambda\,s_\Gamma}{L} \\
0 & 0 & 0 & -\frac{{c_\Gamma}^2\,c_\Lambda\,s_\Lambda}{L} & -\frac{{c_\Gamma}^2\,{s_\Lambda}^2}{L} & -\frac{c_\Gamma\,s_\Gamma\,s_\Lambda}{L} & 0 & 0 & 0 & \frac{{c_\Gamma}^2\,c_\Lambda\,s_\Lambda}{L} & \frac{{c_\Gamma}^2\,{s_\Lambda}^2}{L} & \frac{c_\Gamma\,s_\Gamma\,s_\Lambda}{L} \\
0 & 0 & 0 & -\frac{c_\Gamma\,c_\Lambda\,s_\Gamma}{L} & -\frac{c_\Gamma\,s_\Gamma\,s_\Lambda}{L} & -\frac{{s_\Gamma}^2}{L} & 0 & 0 & 0 & \frac{c_\Gamma\,c_\Lambda\,s_\Gamma}{L} & \frac{c_\Gamma\,s_\Gamma\,s_\Lambda}{L} & \frac{{s_\Gamma}^2}{L}
\end{bmatrix}$$

$$\frac{\mathrm{d}K}{\mathrm{d}EI_{zz}} \begin{bmatrix}
\frac{12\,A^2}{L^3} & -\frac{12\,A\,B}{L^3} & -\frac{12\,A\,c_\Gamma\,s_\theta}{L^3} & \frac{6\,A\,C}{L^2} & \frac{6\,A\,D}{L^2} & -\frac{6\,A\,c_\Gamma\,c_\theta}{L^2} & -\frac{12\,A^2}{L^3} & \frac{12\,A\,B}{L^3} & \frac{12\,A\,c_\Gamma\,s_\theta}{L^3} & \frac{6\,A\,C}{L^2} & \frac{6\,A\,D}{L^2} & -\frac{6\,A\,c_\Gamma\,c_\theta}{L^2} \\
-\frac{12\,A\,B}{L^3} & \frac{12\,B^2}{L^3} & \frac{12\,B\,c_\Gamma\,s_\theta}{L^3} & -\frac{6\,B\,C}{L^2} & -\frac{6\,B\,D}{L^2} & \frac{6\,B\,c_\Gamma\,c_\theta}{L^2} & \frac{12\,A\,B}{L^3} & -\frac{12\,B^2}{L^3} & -\frac{12\,B\,c_\Gamma\,s_\theta}{L^3} & -\frac{6\,B\,C}{L^2} & -\frac{6\,B\,D}{L^2} & \frac{6\,B\,c_\Gamma\,c_\theta}{L^2} \\
-\frac{12\,A\,c_\Gamma\,s_\theta}{L^3} & \frac{12\,B\,c_\Gamma\,s_\theta}{L^3} & \frac{12\,{c_\Gamma}^2\,{s_\theta}^2}{L^3} & -\frac{6\,C\,c_\Gamma\,s_\theta}{L^2} & -\frac{6\,D\,c_\Gamma\,s_\theta}{L^2} & \frac{6\,{c_\Gamma}^2\,c_\theta\,s_\theta}{L^2} & \frac{12\,A\,c_\Gamma\,s_\theta}{L^3} & -\frac{12\,B\,c_\Gamma\,s_\theta}{L^3} & -\frac{12\,{c_\Gamma}^2\,{s_\theta}^2}{L^3} & -\frac{6\,C\,c_\Gamma\,s_\theta}{L^2} & -\frac{6\,D\,c_\Gamma\,s_\theta}{L^2} & \frac{6\,{c_\Gamma}^2\,c_\theta\,s_\theta}{L^2} \\
\frac{6\,A\,C}{L^2} & -\frac{6\,B\,C}{L^2} & -\frac{6\,C\,c_\Gamma\,s_\theta}{L^2} & \frac{4\,C^2}{L} & \frac{4\,C\,D}{L} & -\frac{4\,C\,c_\Gamma\,c_\theta}{L} & -\frac{6\,A\,C}{L^2} & \frac{6\,B\,C}{L^2} & \frac{6\,C\,c_\Gamma\,s_\theta}{L^2} & \frac{2\,C^2}{L} & \frac{2\,C\,D}{L} & -\frac{2\,C\,c_\Gamma\,c_\theta}{L} \\
\frac{6\,A\,D}{L^2} & -\frac{6\,B\,D}{L^2} & -\frac{6\,D\,c_\Gamma\,s_\theta}{L^2} & \frac{4\,C\,D}{L} & \frac{4\,D^2}{L} & -\frac{4\,D\,c_\Gamma\,c_\theta}{L} & -\frac{6\,A\,D}{L^2} & \frac{6\,B\,D}{L^2} & \frac{6\,D\,c_\Gamma\,s_\theta}{L^2} & \frac{2\,C\,D}{L} & \frac{2\,D^2}{L} & -\frac{2\,D\,c_\Gamma\,c_\theta}{L} \\
-\frac{6\,A\,c_\Gamma\,c_\theta}{L^2} & \frac{6\,B\,c_\Gamma\,c_\theta}{L^2} & \frac{6\,{c_\Gamma}^2\,c_\theta\,s_\theta}{L^2} & -\frac{4\,C\,c_\Gamma\,c_\theta}{L} & -\frac{4\,D\,c_\Gamma\,c_\theta}{L} & \frac{4\,{c_\Gamma}^2\,{c_\theta}^2}{L} & \frac{6\,A\,c_\Gamma\,c_\theta}{L^2} & -\frac{6\,B\,c_\Gamma\,c_\theta}{L^2} & -\frac{6\,{c_\Gamma}^2\,c_\theta\,s_\theta}{L^2} & -\frac{2\,C\,c_\Gamma\,c_\theta}{L} & -\frac{2\,D\,c_\Gamma\,c_\theta}{L} & \frac{2\,{c_\Gamma}^2\,{c_\theta}^2}{L} \\
-\frac{12\,A^2}{L^3} & \frac{12\,A\,B}{L^3} & \frac{12\,A\,c_\Gamma\,s_\theta}{L^3} & -\frac{6\,A\,C}{L^2} & -\frac{6\,A\,D}{L^2} & \frac{6\,A\,c_\Gamma\,c_\theta}{L^2} & \frac{12\,A^2}{L^3} & -\frac{12\,A\,B}{L^3} & -\frac{12\,A\,c_\Gamma\,s_\theta}{L^3} & -\frac{6\,A\,C}{L^2} & -\frac{6\,A\,D}{L^2} & \frac{6\,A\,c_\Gamma\,c_\theta}{L^2} \\
\frac{12\,A\,B}{L^3} & -\frac{12\,B^2}{L^3} & -\frac{12\,B\,c_\Gamma\,s_\theta}{L^3} & \frac{6\,B\,C}{L^2} & \frac{6\,B\,D}{L^2} & -\frac{6\,B\,c_\Gamma\,c_\theta}{L^2} & -\frac{12\,A\,B}{L^3} & \frac{12\,B^2}{L^3} & \frac{12\,B\,c_\Gamma\,s_\theta}{L^3} & \frac{6\,B\,C}{L^2} & \frac{6\,B\,D}{L^2} & -\frac{6\,B\,c_\Gamma\,c_\theta}{L^2} \\
\frac{12\,A\,c_\Gamma\,s_\theta}{L^3} & -\frac{12\,B\,c_\Gamma\,s_\theta}{L^3} & -\frac{12\,{c_\Gamma}^2\,{s_\theta}^2}{L^3} & \frac{6\,C\,c_\Gamma\,s_\theta}{L^2} & \frac{6\,D\,c_\Gamma\,s_\theta}{L^2} & -\frac{6\,{c_\Gamma}^2\,c_\theta\,s_\theta}{L^2} & -\frac{12\,A\,c_\Gamma\,s_\theta}{L^3} & \frac{12\,B\,c_\Gamma\,s_\theta}{L^3} & \frac{12\,{c_\Gamma}^2\,{s_\theta}^2}{L^3} & \frac{6\,C\,c_\Gamma\,s_\theta}{L^2} & \frac{6\,D\,c_\Gamma\,s_\theta}{L^2} & -\frac{6\,{c_\Gamma}^2\,c_\theta\,s_\theta}{L^2} \\
\frac{6\,A\,C}{L^2} & -\frac{6\,B\,C}{L^2} & -\frac{6\,C\,c_\Gamma\,s_\theta}{L^2} & \frac{2\,C^2}{L} & \frac{2\,C\,D}{L} & -\frac{2\,C\,c_\Gamma\,c_\theta}{L} & -\frac{6\,A\,C}{L^2} & \frac{6\,B\,C}{L^2} & \frac{6\,C\,c_\Gamma\,s_\theta}{L^2} & \frac{4\,C^2}{L} & \frac{4\,C\,D}{L} & -\frac{4\,C\,c_\Gamma\,c_\theta}{L} \\
\frac{6\,A\,D}{L^2} & -\frac{6\,B\,D}{L^2} & -\frac{6\,D\,c_\Gamma\,s_\theta}{L^2} & \frac{2\,C\,D}{L} & \frac{2\,D^2}{L} & -\frac{2\,D\,c_\Gamma\,c_\theta}{L} & -\frac{6\,A\,D}{L^2} & \frac{6\,B\,D}{L^2} & \frac{6\,D\,c_\Gamma\,s_\theta}{L^2} & \frac{4\,C\,D}{L} & \frac{4\,D^2}{L} & -\frac{4\,D\,c_\Gamma\,c_\theta}{L} \\
-\frac{6\,A\,c_\Gamma\,c_\theta}{L^2} & \frac{6\,B\,c_\Gamma\,c_\theta}{L^2} & \frac{6\,{c_\Gamma}^2\,c_\theta\,s_\theta}{L^2} & -\frac{2\,C\,c_\Gamma\,c_\theta}{L} & -\frac{2\,D\,c_\Gamma\,c_\theta}{L} & \frac{2\,{c_\Gamma}^2\,{c_\theta}^2}{L} & \frac{6\,A\,c_\Gamma\,c_\theta}{L^2} & -\frac{6\,B\,c_\Gamma\,c_\theta}{L^2} & -\frac{6\,{c_\Gamma}^2\,c_\theta\,s_\theta}{L^2} & -\frac{4\,C\,c_\Gamma\,c_\theta}{L} & -\frac{4\,D\,c_\Gamma\,c_\theta}{L} & \frac{4\,{c_\Gamma}^2\,{c_\theta}^2}{L}
\end{bmatrix}$$

with:

$$A = c_\theta\,s_\Lambda + c_\Lambda\,s_\Gamma\,s_\theta$$
$$B = c_\theta\,c_\Lambda - s_\Lambda\,s_\Gamma\,c_\theta$$
$$C = s_\theta\,s_\Lambda + c_\Lambda\,s_\Gamma\,c_\theta$$
$$D = c_\theta\,c_\Lambda + s_\Lambda\,s_\Gamma\,c_\theta$$

$$\frac{\mathrm{d}K}{\mathrm{d}EI_{yy}} = \begin{bmatrix}
\frac{12\,C^2}{L^3} & \frac{12\,C\,D}{L^3} & -\frac{12\,C\,c_\Gamma\,c_\theta}{L^3} & -\frac{6\,A\,C}{L^2} & \frac{6\,B\,C}{L^2} & \frac{6\,C\,c_\Gamma\,s_\theta}{L^2} & -\frac{12\,C^2}{L^3} & -\frac{12\,C\,D}{L^3} & \frac{12\,C\,c_\Gamma\,c_\theta}{L^3} & -\frac{6\,A\,C}{L^2} & \frac{6\,B\,C}{L^2} & \frac{6\,C\,c_\Gamma\,s_\theta}{L^2} \\
\frac{12\,C\,D}{L^3} & \frac{12\,D^2}{L^3} & -\frac{12\,D\,c_\Gamma\,c_\theta}{L^3} & -\frac{6\,A\,D}{L^2} & \frac{6\,B\,D}{L^2} & \frac{6\,D\,c_\Gamma\,s_\theta}{L^2} & -\frac{12\,C\,D}{L^3} & -\frac{12\,D^2}{L^3} & \frac{12\,D\,c_\Gamma\,c_\theta}{L^3} & -\frac{6\,A\,D}{L^2} & \frac{6\,B\,D}{L^2} & \frac{6\,D\,c_\Gamma\,s_\theta}{L^2} \\
-\frac{12\,C\,c_\Gamma\,c_\theta}{L^3} & -\frac{12\,D\,c_\Gamma\,c_\theta}{L^3} & \frac{12\,{c_\Gamma}^2\,{c_\theta}^2}{L^3} & \frac{6\,A\,c_\Gamma\,c_\theta}{L^2} & -\frac{6\,B\,c_\Gamma\,c_\theta}{L^2} & -\frac{6\,{c_\Gamma}^2\,c_\theta\,s_\theta}{L^2} & \frac{12\,C\,c_\Gamma\,c_\theta}{L^3} & \frac{12\,D\,c_\Gamma\,c_\theta}{L^3} & -\frac{12\,{c_\Gamma}^2\,{c_\theta}^2}{L^3} & \frac{6\,A\,c_\Gamma\,c_\theta}{L^2} & -\frac{6\,B\,c_\Gamma\,c_\theta}{L^2} & -\frac{6\,{c_\Gamma}^2\,c_\theta\,s_\theta}{L^2} \\
-\frac{6\,A\,C}{L^2} & -\frac{6\,A\,D}{L^2} & \frac{6\,A\,c_\Gamma\,c_\theta}{L^2} & \frac{4\,A^2}{L} & -\frac{4\,A\,B}{L} & -\frac{4\,A\,c_\Gamma\,s_\theta}{L} & \frac{6\,A\,C}{L^2} & \frac{6\,A\,D}{L^2} & -\frac{6\,A\,c_\Gamma\,c_\theta}{L^2} & \frac{2\,A^2}{L} & -\frac{2\,A\,B}{L} & -\frac{2\,A\,c_\Gamma\,s_\theta}{L} \\
\frac{6\,B\,C}{L^2} & \frac{6\,B\,D}{L^2} & -\frac{6\,B\,c_\Gamma\,c_\theta}{L^2} & -\frac{4\,A\,B}{L} & \frac{4\,B^2}{L} & \frac{4\,B\,c_\Gamma\,s_\theta}{L} & -\frac{6\,B\,C}{L^2} & -\frac{6\,B\,D}{L^2} & \frac{6\,B\,c_\Gamma\,c_\theta}{L^2} & -\frac{2\,A\,B}{L} & \frac{2\,B^2}{L} & \frac{2\,B\,c_\Gamma\,s_\theta}{L} \\
\frac{6\,C\,c_\Gamma\,s_\theta}{L^2} & \frac{6\,D\,c_\Gamma\,s_\theta}{L^2} & -\frac{6\,{c_\Gamma}^2\,c_\theta\,s_\theta}{L^2} & -\frac{4\,A\,c_\Gamma\,s_\theta}{L} & \frac{4\,B\,c_\Gamma\,s_\theta}{L} & \frac{4\,{c_\Gamma}^2\,{s_\theta}^2}{L} & -\frac{6\,C\,c_\Gamma\,s_\theta}{L^2} & -\frac{6\,D\,c_\Gamma\,s_\theta}{L^2} & \frac{6\,{c_\Gamma}^2\,c_\theta\,s_\theta}{L^2} & -\frac{2\,A\,c_\Gamma\,s_\theta}{L} & \frac{2\,B\,c_\Gamma\,s_\theta}{L} & \frac{2\,{c_\Gamma}^2\,{s_\theta}^2}{L} \\
-\frac{12\,C^2}{L^3} & -\frac{12\,C\,D}{L^3} & \frac{12\,C\,c_\Gamma\,c_\theta}{L^3} & \frac{6\,A\,C}{L^2} & -\frac{6\,B\,C}{L^2} & -\frac{6\,C\,c_\Gamma\,s_\theta}{L^2} & \frac{12\,C^2}{L^3} & \frac{12\,C\,D}{L^3} & -\frac{12\,C\,c_\Gamma\,c_\theta}{L^3} & \frac{6\,A\,C}{L^2} & -\frac{6\,B\,C}{L^2} & -\frac{6\,C\,c_\Gamma\,s_\theta}{L^2} \\
-\frac{12\,C\,D}{L^3} & -\frac{12\,D^2}{L^3} & \frac{12\,D\,c_\Gamma\,c_\theta}{L^3} & \frac{6\,A\,D}{L^2} & -\frac{6\,B\,D}{L^2} & -\frac{6\,D\,c_\Gamma\,s_\theta}{L^2} & \frac{12\,C\,D}{L^3} & \frac{12\,D^2}{L^3} & -\frac{12\,D\,c_\Gamma\,c_\theta}{L^3} & \frac{6\,A\,D}{L^2} & -\frac{6\,B\,D}{L^2} & -\frac{6\,D\,c_\Gamma\,s_\theta}{L^2} \\
\frac{12\,C\,c_\Gamma\,c_\theta}{L^3} & \frac{12\,D\,c_\Gamma\,c_\theta}{L^3} & -\frac{12\,{c_\Gamma}^2\,{c_\theta}^2}{L^3} & -\frac{6\,A\,c_\Gamma\,c_\theta}{L^2} & \frac{6\,B\,c_\Gamma\,c_\theta}{L^2} & \frac{6\,{c_\Gamma}^2\,c_\theta\,s_\theta}{L^2} & -\frac{12\,C\,c_\Gamma\,c_\theta}{L^3} & -\frac{12\,D\,c_\Gamma\,c_\theta}{L^3} & \frac{12\,{c_\Gamma}^2\,{c_\theta}^2}{L^3} & -\frac{6\,A\,c_\Gamma\,c_\theta}{L^2} & \frac{6\,B\,c_\Gamma\,c_\theta}{L^2} & \frac{6\,{c_\Gamma}^2\,c_\theta\,s_\theta}{L^2} \\
-\frac{6\,A\,C}{L^2} & -\frac{6\,A\,D}{L^2} & \frac{6\,A\,c_\Gamma\,c_\theta}{L^2} & \frac{2\,A^2}{L} & -\frac{2\,A\,B}{L} & -\frac{2\,A\,c_\Gamma\,s_\theta}{L} & \frac{6\,A\,C}{L^2} & \frac{6\,A\,D}{L^2} & -\frac{6\,A\,c_\Gamma\,c_\theta}{L^2} & \frac{4\,A^2}{L} & -\frac{4\,A\,B}{L} & -\frac{4\,A\,c_\Gamma\,s_\theta}{L} \\
\frac{6\,B\,C}{L^2} & \frac{6\,B\,D}{L^2} & -\frac{6\,B\,c_\Gamma\,c_\theta}{L^2} & -\frac{2\,A\,B}{L} & \frac{2\,B^2}{L} & \frac{2\,B\,c_\Gamma\,s_\theta}{L} & -\frac{6\,B\,C}{L^2} & -\frac{6\,B\,D}{L^2} & \frac{6\,B\,c_\Gamma\,c_\theta}{L^2} & -\frac{4\,A\,B}{L} & \frac{4\,B^2}{L} & \frac{4\,B\,c_\Gamma\,s_\theta}{L} \\
\frac{6\,C\,c_\Gamma\,s_\theta}{L^2} & \frac{6\,D\,c_\Gamma\,s_\theta}{L^2} & -\frac{6\,{c_\Gamma}^2\,c_\theta\,s_\theta}{L^2} & -\frac{2\,A\,c_\Gamma\,s_\theta}{L} & \frac{2\,B\,c_\Gamma\,s_\theta}{L} & \frac{2\,{c_\Gamma}^2\,{s_\theta}^2}{L} & -\frac{6\,C\,c_\Gamma\,s_\theta}{L^2} & -\frac{6\,D\,c_\Gamma\,s_\theta}{L^2} & \frac{6\,{c_\Gamma}^2\,c_\theta\,s_\theta}{L^2} & -\frac{4\,A\,c_\Gamma\,s_\theta}{L} & \frac{4\,B\,c_\Gamma\,s_\theta}{L} & \frac{4\,{c_\Gamma}^2\,{s_\theta}^2}{L}
\end{bmatrix}$$

with:

$$\begin{gathered}
A = c_\theta\,s_\Lambda + c_\Lambda\,s_\Gamma\,s_\theta \\
B = c_\theta\,c_\Lambda - s_\Lambda\,s_\Gamma\,c_\theta \\
C = s_\theta\,s_\Lambda + c_\Lambda\,s_\Gamma\,c_\theta \\
D = c_\theta\,c_\Lambda + s_\Lambda\,s_\Gamma\,c_\theta
\end{gathered}$$