INSTITUTO SUPERIOR TÉCNICO
Universidade Técnica de Lisboa

Multidisciplinary Optimisation Strategies Using Evolutionary Algorithms with Application to Aircraft Design

Artur Jaime Barceló Caldeira Fouto

Dissertação para obtenção do Grau de Mestre em

Engenharia Aeroespacial

Júri

Presidente:	Professor Fernando José Parracho Lau
Orientadora:	Professora Maria Alexandra dos Santos Gonçalves de Aguiar Gomes
Co-Orientador:	Professor Afzal Suleman
Vogal:	Professor Hélder Carriço Rodrigues

Julho de 2009

Abstract

This document describes the development of a Preliminary Aircraft Design application employing the methodology of Multidisciplinary Design Optimisation (MDO) and the concept of Evolutionary Algorithms.

In the work presented in this thesis, aerodynamics, structural analysis and flight performance are the main disciplines considered for preliminary aircraft design. Aerodynamic analysis is performed through a 3D panel method and structural analysis through finite element method, both by external software packages handled by the developed application.

Novel concepts are also explored: a Particle Swarm Optimiser was developed to handle the MDO problem with a large number of design variables and an Artificial Neural Network (ANN) was integrated to predict the Pareto Front (in a context of Multiobjective Optimisation) and also investigated as an accelerator for the whole optimisation process.

The goal of developing an application that is fully independent from user input during the optimisation process and is able to interact with external analysis tools was reached and several aircraft design optimisation problems are solved, in order to demonstrate the advantages of the MDO concept, the developed optimiser and the use of the ANN.

Keywords: Multidisciplinary Design Optimisation, Particle Swarm Optimisation, Artificial Neural Network, 3D Panel Method, Finite Element Method, Pareto Front

Resumo

Este documento descreve o desenvolvimento de uma aplicação para Projecto Aeronáutico Preliminar, incorporando a metodologia de Projecto Óptimo Multidisciplinar (MDO) e o conceito de Algoritmos Evolucionários.

No trabalho apresentado nesta tese, desempenho, análises aerodinâmica e estrutural são as principais disciplinas consideradas para projecto aeronáutico preliminar. A análise aerodinâmica é realizada através do método de painéis 3D e a análise estrutural através do método de elementos finitos, ambas implementadas por ferramentas de análise externas, cuja interacção é feita pela aplicação desenvolvida.

Novos conceitos são também explorados: um optimizador do tipo *Particle Swarm* foi desenvolvido para abordar o problema MDO com um número elevado de variáveis de projecto e uma rede neuronal (ANN) foi integrada de forma a prever a Frente de Pareto (no contexto de optimização multiobjectivo) e investigada também como acelerador do processo de optimização.

O objectivo de desenvolver uma aplicação que fosse completamente independente do utilizador e capaz de interagir com ferramentas de análise externas foi atingido e vários problemas de optimização no âmbito de projecto aeronáutico foram resolvidos, de forma a demonstrar as vantagens do conceito MDO, do optimizador desenvolvido e do uso da ANN.

Palavras-Chave: Projecto Óptimo Multidisciplinar, Optimização *Particle Swarm*, Redes Neuronais, Método de Painéis Tridimensional, Método de Elementos Finitos, Frente de Pareto

Acknowledgments

I would like to thank my supervisors, Professors Alexandra Gomes and Afzal Suleman, for suggesting this theme to me but most importantly for their support during the course of this work. Without their critical judgment and views this work would not be possible.

My acknowledgment also goes to *Fundação para a Ciência e Tecnologia (FCT)* by supporting this work under Grant *ICDT 05-S3T-FP012-Smorph*.

I also have to thank my friends and colleagues, with whom I shared time, thoughts and, sometimes, setbacks throughout these last months.

Finally, I would like to thank my family, who have always supported and encouraged me to be the best that I can aspire to, in every aspect of my life.

Contents

List of Figures

List of Symbols

x_i	optimisation position vector
v_i	optimisation position vector
ω	inertia of particle
C_P	particle confidence factor
C_G	group confidence factor
X_i	optimisation position vector amplitude
V_i	optimisation position vector amplitude
r	random number in $[0, 1]$
ϕ	potential
U_∞	free stream velocity, longitudinal component
V_∞	free stream velocity, lateral component
W_∞	free stream velocity, vertical component
C_p	pressure coefficient
C_f	friction coefficient
p	pressure
p_∞	static pressure
ρ	density
μ	dynamic fluid viscocity
α	angle of attack (AOA)
β	side-slip angle
C_l	airfoil lift coefficient
C_d	airfoil drag coefficient
C_m	airfoil pitching moment coefficient
C_L	aircraft lift coefficient
C_D	aircraft drag coefficient
C_M	aircraft pitching moment coefficient
δ_{tip}	wing tip displacement
σ_{max}	maximum stress intensity in the structure
$\sigma_{6061-T6}$	maximum allowable stress for the material

Abbreviations

ANN	Artificial Neural Network
AOF	Aggregate Objective Function
CFD	Computational Fluid Dynamics
DV	Design Variable
EA	Evolutionary Algorithm
FE	Finite Element
FEA	Finite Element Analysis
FEM	Finite Element Method
MDF	Multidisciplinary Feasible
MDO	Multidisciplinary Design Optimisation
PSO	Particle Swarm Optimisation
SO	Sequential Optimisation
SQP	Sequential Quadratic Programming

Chapter 1

Introduction

1.1 Thesis Objective

One of the biggest challenges that MDO tools have to overcome is flexibility to adapt to different engineering scenarios and are usually bound to solving a predetermined set of design variables. Furthermore, the analysis fidelity level is typically low, relying on methods that are often too much simplified to deliver the much needed accuracy.

The main objective of this work is therefore to create an MDO tool that moves towards higher fidelity tools, in the context of aircraft design and integrating the emerging concept of evolutionary algorithms. In order to fulfill this requirement, a suitable optimiser has to be chosen or developed. Analysis tools that meet the desired depth level must also be chosen taking in account the balance between accuracy and computational cost. This application must be developed to interact with the analysis tools considering them as independent blocks, functioning as external modules, so that more accurate tools can be easily used in the future, simply by swapping them. Finally, the computational cost of running an optimisation problem should be reasonable.

The developed application should be validated by several optimisation problems, both singlediscipline but, most importantly, multidisciplinary ones, particularly in the aircraft design field.

1.2 Thesis Layout

This thesis is divided in eight chapters.

In chapter 2, the concept of Multidisciplinary Design Optimisation is defined, as well as its applications to engineering in general and the aerospace segment in particular. In this chapter, the MDO statement and the optimisation problems to be analysed are also defined.

In chapter 3, an historical perspective on Evolutionary Algorithms is made followed by the choice and development of the optimiser algorithm to be used (Particle Swarm Optimisation) in the MDO application. A method for accelerating the optimisation process and determining the Pareto Front based on a biologic inspired process (an Artificial Neural Network) is also discussed.

In chapter 4, the choice of a suitable aerodynamic analysis method is discussed (3D Panel Method). This chapter includes a description of the definition of the aerodynamic model. Important data to be extracted from the aerodynamic solution and how this module is managed by the main application is also described.

In chapter 5, a similar process as in the previous chapter is done by choosing an adequate structural analysis (Finite Element Method) method and describing the structural model, both from an aircraft structures as well as a FEM points of view.

In chapter 6, a thorough description of the application development process is made. All of the relevant programming choices and problems that arose during the implementation of this work are addressed and the interaction between all of the developed and external modules is described.

In chapters 7 and 8, a series of optimisation problems are solved in order to validate the developed MDO application. Singlediscipline problems are first solved to validate the optimisation concept in each discipline and, finally, a multiobjective multidisciplinary problem is solved, to show the full potential of the application in finding optimal solutions.

Finally, in chapter 9, conclusions are drawn from the results presented in the previous chapters and the fulfillment of the proposed objectives is discussed. Additionally, a discussion of the issues that were left unapproached and that may serve the basis for future work is made.

Chapter 2

Multidisciplinary Optimisation

The first section of this chapter provides a brief introduction to Multidisciplinary Design Optimisation (MDO), followed by an historic perspective of the uses of this methodology in engineering. Section 2.2 describes the state-of-the-art of MDO in aeronautics and aerospace engineering.

2.1 Introduction

As the complexity of engineering systems grows and, with it, the complexity of the design process, it becomes useful to explore the interaction of the various disciplines involved from the early stages of the design process in order to guarantee that an optimal solution is found.

Traditionally, engineering design consists of a sequence of steps, beginning with a conceptual solution to a certain mission that is to be performed. This conceptual phase continues on to a preliminary stage until a configuration can be frozen. Only then are detailed analyses performed, corresponding to each discipline involved in the "product" to be developed, as can be observed in figure 2.1.

However, this design methodology leads to a successive bottlenecking in design freedom as the analyses and design detail is increased (figure 2.2), a fact that has been formally demonstrated [5] and that may lead to a suboptimal design, furthermore emphasizing the advantages of an MDO approach at an early stage.

For the purpose of this work, the following definition for MDO is considered: *"A methodology for the design of complex engineering systems and subsystems that coherently exploits the synergism of mutually interacting phenomena"* [1]. Multiple conflicting requirements have always had to be taken into account and, therefore, it can be considered that the multidisciplinary process has always been used. The key word in the definition, however, is methodology [2]. MDO provides a collection of tools and methods that per-

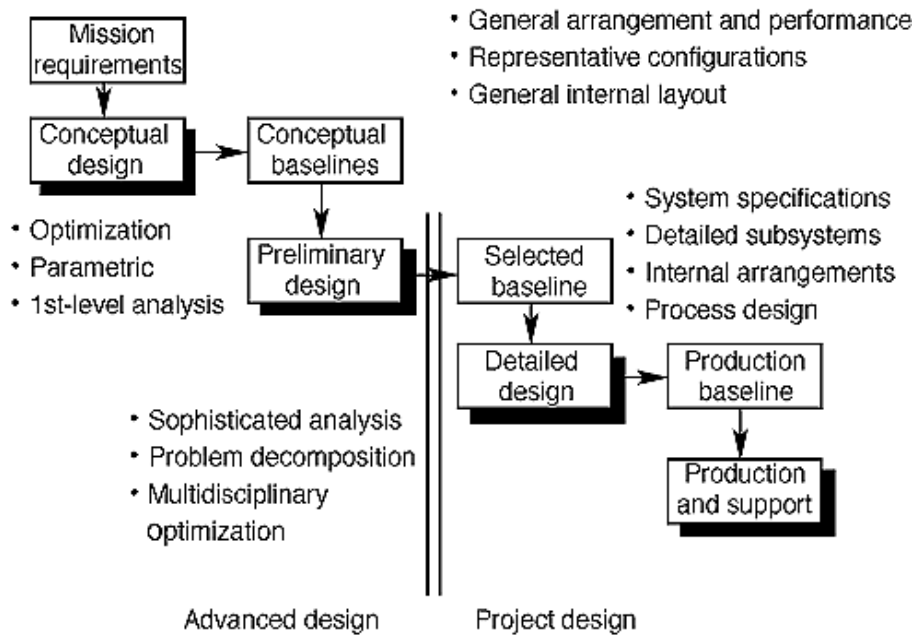

Figure 2.1: Division of a product development into phases [1].

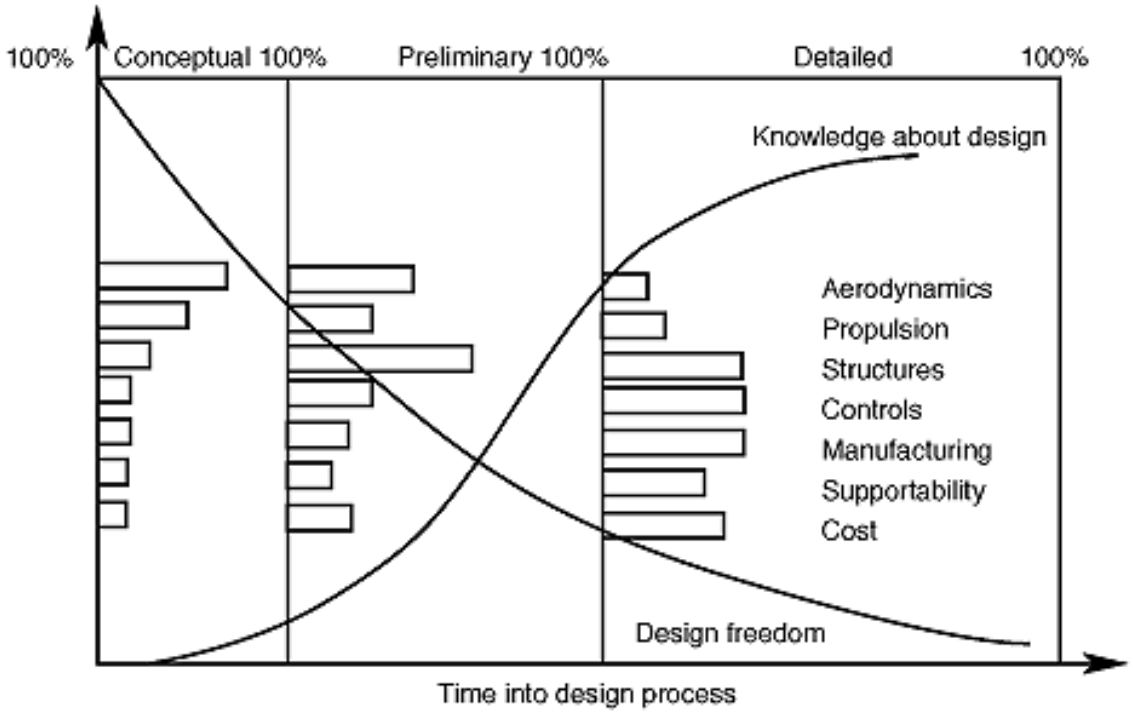

Figure 2.2: Traditional approach to product development [1].

mit the trade-off between different disciplines involved in the design process. *"MDO is not design but enables it"* [1].

Ideally, the MDO environment should be interactive and flexible enough to allow the problem definition, constraints to be applied and simulation depth to be fully specified by the design team, rather than the individual disciplines' teams.

In order to facilitate information exchange between the various disciplines and respective teams (or for that matter, analysis tools), a single global parametric model of the whole system should be used, from which discipline specific models can be generated [6, 7, 8]. This consistency has been shown to offer advantages, both when it comes to communication between disciplines and eventual redefinition of the global parametric model. [9, 10, 7]

This environment should be transparent, in the sense that it should allow the design team to monitor the evolution of variables, verifying whether these are dependent or independent with relation to the problem. This enforces the notion that the top design team should have full control of the process flow.

Taking in account that modern engineering systems are extremely complex, it is only natural to distribute the various disciplines over their respective groups, all interconnected by the MDO environment. Although process distribution may present some management challenges, it truly allows for the distribution to be a physical resource distribution, rather than just a process division. This enables groups to be able to be in different sites, often worldwide; it also enables the use of computational resources and data storage spread over a vast number of nodes [11].

2.2 MDO Strategies Applied to Aircraft Design

MDO has been used as a design methodology in the past, both in conceptual studies as well as in real-life applications.

First of all, it is convenient to establish a definition of fidelity levels. Bartholomew [12] has defined a set of analysis fidelity levels as follows:

- Level 1: empirical equations;
- Level 2: intermediate level models (e.g., beam theory, panel method, etc.);
- Level 3: state-of-the-art, high fidelity models (e.g., CFD, FEA),

and has observed that, at the industry level MDO, the tendency is to move towards the highest level, as each discipline's experts tend to use the highest fidelity models.

MDO methodology has been used in the *WingMOD* process, in order to minimise the weight of the *Boeing's Blended Wing Body* [6]. In this case," *the process is fully multi-disciplinary and includes design variables for planform shape/size, mission, aerodynamic, structural sizing/topology, fuel/payload, and trim schedule (134 in all). WingMOD uses a close-coupled approach using intermediate fidelity disciplinary analyses for high aspect ratio wing aircraft. (...) The aerodynamic analyses include the vortex lattice method and quasi two-dimensional compressibility corrections. The structural sizing and constraints are based on aeroelastic loads and deflection analysis, simplified buckling, and stress analysis of simple beams. The weight is based on the structural analysis corrected by some statistical data. A wide breadth of practical constraints are considered (705 in all) along with 20 design flight conditions that cover most of the critical design considerations*" [6]. In this work, the main need was to be able to include a CFD approach, without excessive computational resource consumption, as the number of runs of the analysis routine was in the order of thousands. This lead to the choices presented above, regarding structural and aerodynamic analysis methods.

In the *GM* IVDA (Integrated Vehicle Design Analysis) system [9], the goal was to achieve automobile fuel efficiency. Even though apparently far from aircraft design, the problem was dealt in such a way that the included disciplines were the same as the ones typically found in the latter. The system is composed of both commercial codes (*ODYSSEY, NASTRAN, LPM, DYNA3D, CAL3D, ADAMS*) and *GM* in-house developed codes (aerodynamics, solar load, fuel economy, and others) but only one design variable was optimised at a time. The main purpose of this project was to investigate the pros and cons of an MDO methodology, having as main conclusion that, in an industrial environment, there is the need for automated interaction between analysis tools and that off-the-shelf specialised solutions should be always adopted, when possible.

In [10], the authors describe the process for a detailed aeroelastic optimisation of the *F-22* fighter. A structural configuration was fixed, being the optimisation goal the minimisation of weight, while satisfying stress safety margins, flutter margins and fatigue life requirements. Active aeroservoelastic methods were also considered, by modifying active controls for load alleviation and introducing filtering control laws to discard unfavourable aeroelastic interactions that could result in the onsetting of flutter. A very high fidelity FE model was used, limited only by the high memory limit (10 TBytes). The optimisation strategy that was implemented allowed for different disciplines to be at different optimisation levels at a given stage of the global optimisation process. One of the main conclusions of this work was that the inconsistency between disciplines resultant from the adopted strategy did not seem to affect convergence severely.

A *Collaborative Environment for Turbine Engine Development* is discussed in [13]. In this work, it is shown that one of the major challenges to performing MDO is to be able

to do *Multidisciplinary Design Analysis*. Feasibility (regarding fatigue life and distortion tolerance requirements) and minimum weight (both for structural as well as cost factors) were the main goals to be achieved. The authors of this work noted that the complex analysis capability (resulting from the integration of the individual simulations required) is not as smooth as desired, and that large step size finite differences are required to obtain robust derivatives. This MDO process was implemented in *iSIGHT*, and a *Collaborative Optimisation* strategy was attempted (an optimisation strategy based on distributed design, in which individual disciplinary teams are charged with satisfying local constraints), although without great success due to the nature of the problem and high fidelity requirement. Currently, most of the efforts of this project are on developing specific disciplinary tools and respective process automation.

In [8], the design of a "better", more agile version of the *F-16* fighter was studied. Maneuverability, controllability, weight, and producibility were weighted in determining the quality of the attained solution. After a first optimisation process, it was verified that only with aeroelastic tailoring of the structure was the new version able to exceed the maneuverability of the original version. One of the most interesting points of this study was a comment of the author regarding the following: " *... the approach to achieve integration would probably be the same today (1998) as in 1988-89. The differences in the overall process would be in the tool selection... and the amount of data generated.*" This reinforces the idea that the conceptual idea behind MDO is self contained and needs not to suffer any changes in spite of the evolution of the analysis tools.

Another example where MDO was used in a real-life situation, was the development of the *F/A-18 E/F* [14]. The initial version of the *F/A-18* was not intended for aircraft carrier operation, therefore a re-design was made regarding carrier suitability (landing weight), strike mission (payload), fighter mission (range), increased survivability, maneuverability, growth potential, and others. The goal was to reach a feasible design at an acceptable cost. As main conclusions of this project, the authors emphasize the building of an aerodynamic database made of a combination of CFD results and wind tunnel data (therefore, a very high quality one) which proved to be critical to achieve a proper aeroelastic optimisation.

These cases are summarised in fig. 2.3, regarding the possible trade off between high fidelity analysis methods and multidisciplinarity.

More recently, some strategies have been studied in order to verify the validity of the MDO approach in environments where uncertainty is present. In [15], the preliminary design of a *Supersonic Business Jet* is based on the parallel use of MDO and uncertainty management (necessary when design uncertainties are present) leading to robust design. The result was a tool that allows to robustly design an aircraft which revealed to be fundamental to ensure the feasibility and the credibility of a project with high uncertainties

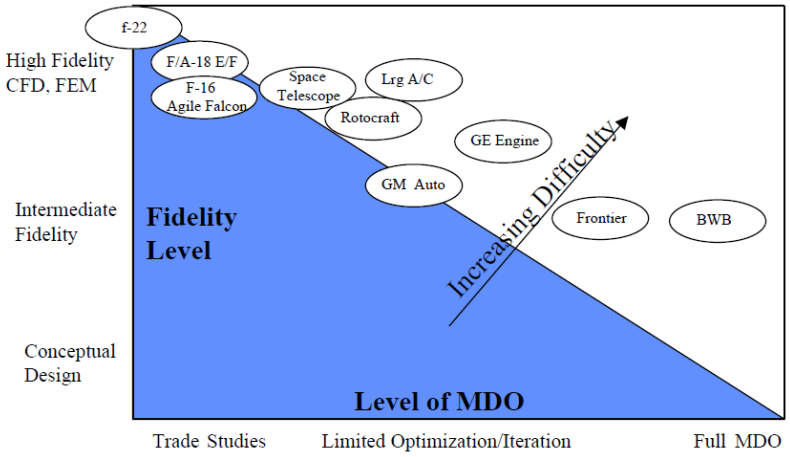

Figure 2.3: Distribution of Design Process Fidelity and Level of MDO [2].

such as a supersonic business jet. This approach is however complementary to the classical design approach using MDO, which remains the backbone of the preliminary design activities.

In the space segment, the study of launcher platforms has been done in several occasions [16, 17], taking in account the integration of propulsion, orbital insertion, mass/dimensions and cost disciplines. In the case of [17], a commercial MDO software was used (*modeFRONTIER*) in conjunction with other off-the-shelf solutions for each discipline, in a straightforward approach. The main point of this work was to demonstrate that tools are available and have evolved to a point where they can be used in an industrial environment and real-life applications with enconomic advantages (particularly in the space segment, where R&D but also operational costs are already very high).

Still in the space segment, using the example of a linear aerospike engine [18], clear advantages to using MDO are shown. In this work, a linear aerospike nozzle is designed based on aerodynamics and structural modules. First, a Sequential Optimisation strategy is implemented, i.e., the best aerodynamic nozzle shape is found, followed by finding the best possible structure for that shape. This solution is then considered to be the baseline and an MDO strategy is applied to it. Improvement in net thrust was verified, meaning that performing MDO truly explores the synergisms between the considered disciplines and is able to reach a truly optimal solution for a given problem.

In [19], a user-friendly application and MDO framework were developed, for the problem of preliminary aircraft wing design. In this work, different optimisation strategies were also compared, namely Multidisciplinary Feasible and Sequential Optimisation and the chosen optimiser was a gradient based algorithm (Sequential Quadratic Programming). In-house aerodynamic, structural and aeroelastic modules were developed, in order to support the application. The main conclusions that can be had from this work were regarding the eventual non-suitability of gradient based methods for a very large number of design

variables, as large-scale, direct search methods could have an edge in that situation.

Finally, and to cite an example similar to the work developed in this dissertation, in [20], a *Transport Aircraft Wing* is optimised. A simple FE model represents the wing's structure and an approximated aerodynamic model is used. The goal of the optimisation problem was to increase the range. The main purpose of the work by itself was to analyse the robustness of the *Particle Swarm Algorithm*. The main conclusions were that this algorithm "*...is able to reliably find the optimal point, despite the presence of discrete variables and severe numerical noise.*" It is also referred that gradient-based methods are more suitable for the simple problem that was studied. However, it is suggested that, for an increasingly large number of discrete design variables, a tuned Particle Swarm Algorithm can be faster than the gradient-based counterpart, particularly if taken in account that the PSO is a strong candidate for a distributed computation approach (a concern that was already expressed in [19]).

Chapter 3

Evolutionary Algorithms

In section 3.1, a brief historical perspective on this subject is presented. In 3.2, a description of the chosen algorithm will be made. Also important aspects of its implementation are described and some studies are performed to demonstrate the algorithm's capabilities. Finally, in 3.3, the use of an Artificial Neural Network as an optimiser accelerator and a Pareto Front detector are studied.

3.1 Introduction

Being the topic of this work the development of an MDO application, a suitable optimiser needs to be chosen or developed. From what was referred in section 2.2 it would be hard to favour deterministic or heuristic optimisation methods. As it is suggested that, for an increasing number of variables, properly tuned heuristic methods may outperform gradient-based methods and because it would be interesting to explore what these methods have to offer in the optimisation context, a decision was made early on in this work to choose Evolutionary Algorithms (EA's) for the optimiser.

Evolutionary algorithms are a set of a larger group of algorithms, so called meta-heuristic methods. In these methods, the goal is to find the extremes (from this point on assumed to be the minima) of a certain objective function with the advantage that the exact state function needs not to be known, i.e., the evaluation module(s) of a possible solution can be looked at as a "black-box". This is of extreme advantage in the design of complex engineering systems, as it would be difficult to find the function that relates the inputs (the design variables) to the output (the objective function).

For this work, a Particle Swarm Optimisation algorithm was chosen, as this is a reasonably recent method, proposed in 1995 by Kennedy and Eberhart [21], and research in the optimisation field shows this method yields good results, when applied to engineering [19, 20, 22].

3.2 Particle Swarm Optimisation

Particle Swarm Optimisation is a population-based evolutionary algorithm based on the concept of social intelligence. In this algorithm, a group of initial individuals is randomly generated, containing information about their position and velocity within a subspace of the design variables. Each individual is then evaluated by an objective function that defines which individual holds the best position in relation to the problem at hand. On the next iteration, individuals are attracted to that point as well as to their respective best position ever, by changing their velocity. As the optimisation process develops, the whole population further explores the subspace and will eventually converge to the optimum of the objective function in that subspace.

This method holds a number of advantages that makes it a suitable optimiser for the problem at hand: it has advantages over other EA's, regarding efficiency (lower number of iterations needed to attain an optimal solution) and flexibility (independence from the problem to solve) [21, 22]; it is a robust minima finder (for both local and absolute minima), as noise insensitivity is well shown in [15, 23, 24]; it has the ability of finding a minimum outside its initial bounds; there is independence between the dimension of the space in which the particles move and the number of particles in the swarm, regarding the algorithm's ability to find a minimum and it is an obvious choice for a distributed computation environment.

3.2.1 Implementation of the PSO

Although there are many available software packages with several variations on the basic PSO algorithm, a custom version was implemented for this work, as this approach leads to a better control and adaptability to the rest of the application.

According to the heuristics behind PSO, a certain particle is moving in a hyperspace of dimension N, with current position given by x_i and velocity by v_i. Dimension, N corresponds to the number of design variables in the optimisation problem and each component of the vector x_i would be the corresponding design variables' value. The distance d_i between any two points is simply calculated as the difference in position between them.

An implementation of this algorithm was developed in $C++$, according to the following description:

$$x_i^{t+1} = x_i^t + v_i^t \Delta t \tag{3.1a}$$

$$v_i^{t+1} = \omega v_i^t + rC_G \frac{d_{i,Best\ individual}}{\Delta t} + rC_P \frac{d_{i,Best\ position\ of\ i\ ever}}{\Delta t} \tag{3.1b}$$

Parameter r is called the "'craziness"' factor, and gives the swarm some random search capabilities. Typically, this value is a random number between 0 and 1, but a more conservative approach was used in this work, establishing the craziness boundaries between 0.5 and 1.

Parameters C_G and C_P correspond to group and particle confidence factors, respectively. Typically, these parameters are random values that are positive and no greater than a prescribed limit. The ratio between the two limits will determine the behaviour of the swarm. If the ratio favours particle confidence, then it is most likely that an individual particle will move towards its own verified minimum, giving the algorithm good local minima search capability. On the other extreme, where higher group confidence is verified, all of the particles will tend to the global minimum, giving the algorithm a better global minimum search capability.

Parameter ω is a value comparable to the particles' inertia. Again, choosing its value should be done taking in account what is the desired behaviour of the swarm. A lower inertia particle will have a greater sensitivity to found minima, giving the swarm a faster convergence behaviour. Naturally, an inertia approaching a null value will transform the swarm into a group of random search individuals, moving away from the swarm concept that is behind this algorithm.

As can be seen from the above, adjusting confidence factors and particle inertia should be done with care and knowledge of the problem at hand. If this information is not available *a priori*, then conservative values should be used and later on adjusted if necessary.

3.2.2 Detailed Implementation

With exception of external modules (aerodynamic and structural solvers), all of the implementation was done in *C++*. This language has the advantage of allowing the use of classes, a type of structured variable, in which the user can include any type of objects, not only typical variables, but also functions. Using classes allows for a rather straightforward style of programming, as the developed functions that deal with the classes can remain unchanged, regardless of the changes made to the class definition.

For the particle swarm optimiser, a class to define the particle's characteristics was created. This class contains the current position vector x_i, the current velocity vector v_i, the current objective function value, the best position ever vector $x_{i,Best}$ and the best objective function value ever. These last two variables add memory depth to the swarm which allows the swarm to return to a past position that may be better than the current best position. This is indeed a feature that gives this algorithm an advantage in relation to other EA's). The class also contains a call to the objective function, which can be any-

thing, from a small mathematical function to a full in-depth aerodynamic and structural analysis performed by external applications as well as a function to compare whether the current position is better then the best position ever of that particle. Finally, the class contains the prescribed values for ω, C_G and C_P. The swarm then simply consists of a vector of n individuals of the type defined in the class.

As for the optimising process itself, the swarm's particles are randomly generated, regarding their initial position and velocity; confidence factors and inertia are prescribed and an adequate time interval Δt is imposed.

The iterative process then begins, by analysing each particle's objective function value, determining the best particle in the groups and then updating all the particles' new position and velocity vectors. The process ends after a certain amount of time or iterations or if some criterion of convergence is achieved. As the algorithm itself returns the velocity of each particle during the process, this is a natural choice for the criterion of convergence: when the best individual has both low velocity and acceleration, this means that an optimal point has been reached. This was the criterion used in the developed implementation of the particle swarm optimiser.

This process is ilustrated in figure 3.1 (where X_i, V_i are position and velocity amplitudes for each variable i and r is a random number in the interval $[0, 1]$ [1]).

3.2.3 Validation of the Implemented Optimiser

After the implementation of the particle swarm optimiser, three benchmark tests were conducted. The following test functions were used to validate the implementation of the algorithm (in a generic context, outside that of aircraft design): Extended Rosenbrock, Extended Freudenstein and Roth and Extended Beale.

i) Extended Rosenbrock function:

$$f(x) = \sum_{i=1}^{N/2} 100 \left(x_{2i} - x_{2i-1}^2\right)^2 + (1 - x_{2i-1})^2 \tag{3.2}$$

with global minimum:
$x = [1, ..., 1]$, $f(x) = 0$

[1] The pseudo random number generator used was the *'Mother-of-All' Random Number Generator*, a multiply-with-carry type developed by Prof. George Marsaglia [25], in a GNU implementation by Agner Fog.

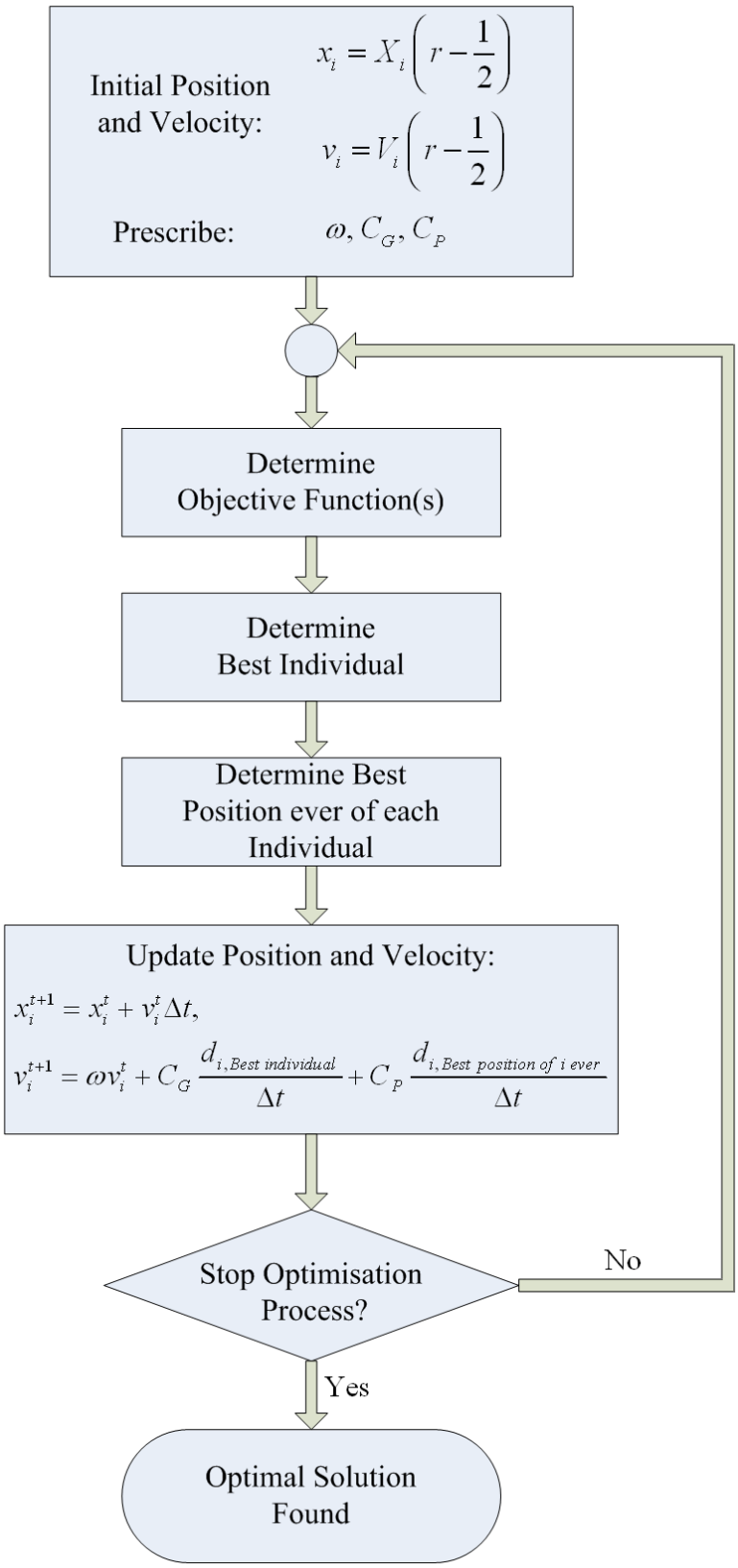

Figure 3.1: PSO flow chart.

ii) Extended Freudenstein and Roth function:

$$f(x) = \sum_{i=1}^{N/2} \{ \; (-13 + x_{2i-1} + ((5 - x_{2i})x_{2i} - 2)\, x_{2i})^2 + \\ (-29 + x_{2i-1} + ((1 + x_{2i})x_{2i} - 14)\, x_{2i})^2 \; \} \tag{3.3}$$

with global minimum:

$x = [5, 4, ..., 5, 4]$, $f(x) = 0$

This function also has a number of local minima, where the pair $\{5, 4\}$ is replaced by $\{11.413, -0.8968\}$ and for a number k of these pairs instead of $\{5, 4\}$ the local minimum is:

$f(x) = k48.98$

iii) Extended Beale function:

$$f(x) = \sum_{i=1}^{n/2} \{ \; (1.5 - x_{2i-1}(1 - x_{2i}))^2 + \\ \left(2.25 - x_{2i-1}(1 - x_{2i}^2)\right)^2 + \\ \left(2.625 - x_{2i-1}(1 - x_{2i}^3)\right)^2 \; \} \tag{3.4}$$

with global minimum:

$x = [3, 0.5, ..., 3, 0.5]$, $f(x) = 0$

This function also has a number of local minima, where the pair $\{3, 0.5\}$ is replaced by $\{-157.52, 1.0063\}$ and for a number k of these pairs instead of $\{3, 0.5\}$ the local minimum is:

$f(x) = k0.4617$

Moreover, the following values

$$\omega = 0.8$$
$$C_G = 2.5$$
$$C_P = 1.0$$

were used.

These favour finding the global minimum, as the group confidence is higher than particle confidence. The inertia starts at the prescribed value and is decreased as the optimi-

sation process advances, ending at $\omega = 0.5$. This strategy favours the initial global search behaviour of the swarm, evolving to a faster local search in the end of the optimisation run.

The initial hypercube where the population is randomly generated was defined in the interval $[-6, 6]^N$. Although this seems a rather small search space, the functions tested have very shallow plateaus near the minima and steep walls around them, being the function value at the extremes of the hypercube in the order of 10^5 for the first two functions and 10^6 for the third.

The following graphs show error vs. number of function evaluations for all of the above test functions and for population sizes $P = 10, 20, 40$ and dimensions $N = 2, 8, 20$. Error here is defined as the modulus of the difference between the found minimum and the known global minimum for each of the tested functions. For high values of N, if a local minimum is found, this may lead to what appears to be high error, but means only that the algorithm failed to find the global minimum. A value of $\Delta t = 0.001$ was used (the choice of this value is furter discussed in 6.4.1).

This first set of graphics (figure 3.2) shows the validation results for the Extended Rosenbrock Function (function 3.2) for $N = 2, 8, 20$ (first three graphics) at the shown swarm size P. The last graphic (bottom right) shows, for $P = 20$, the difference in convergence time for the shown dimension N.

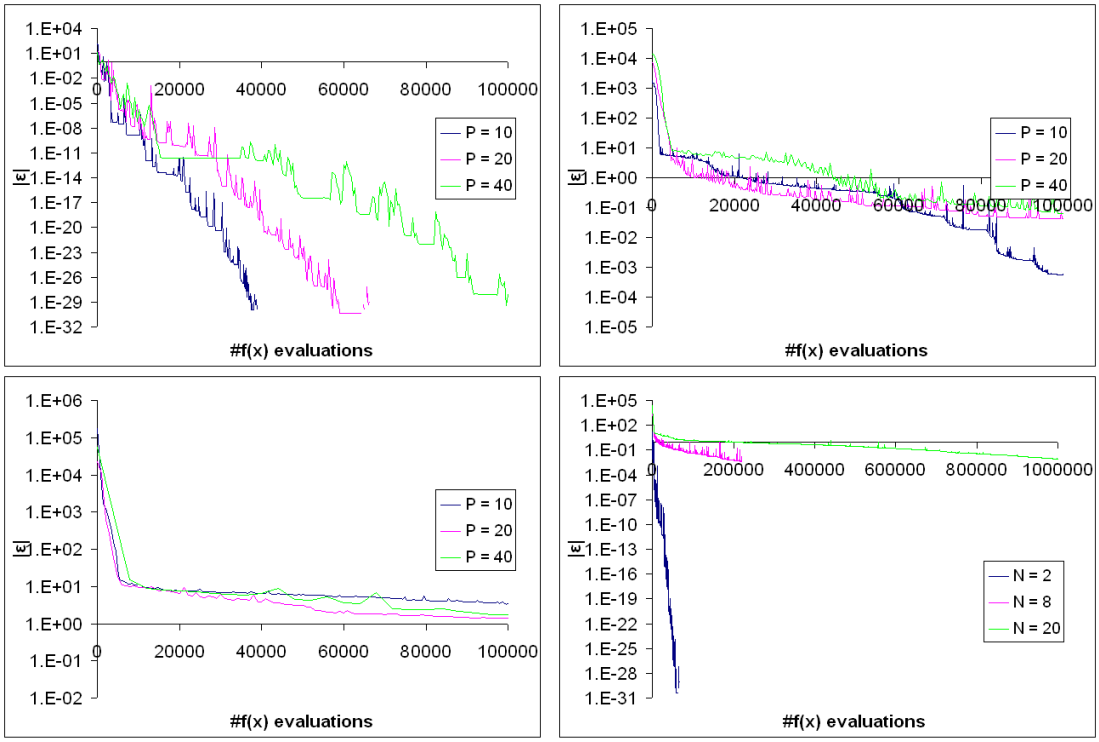

Figure 3.2: Validation for the Extended Rosenbrock Function.

The following set of graphics (figure 3.3) has the same arrangement as the previous figure, but for the Extended Freudenstein and Roth Function (function 3.3).

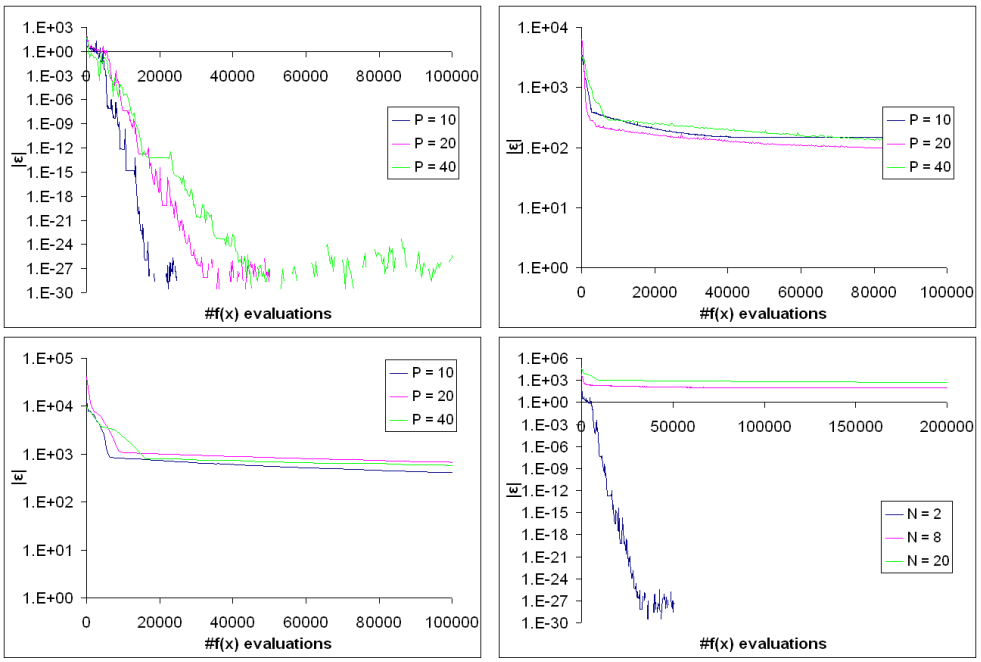

Figure 3.3: Validation for the Extended Freudenstein and Roth Function.

Finally, the last set of graphics (figure 3.4) has the same arrangement, but for the Extended Beale Function (function 3.4).

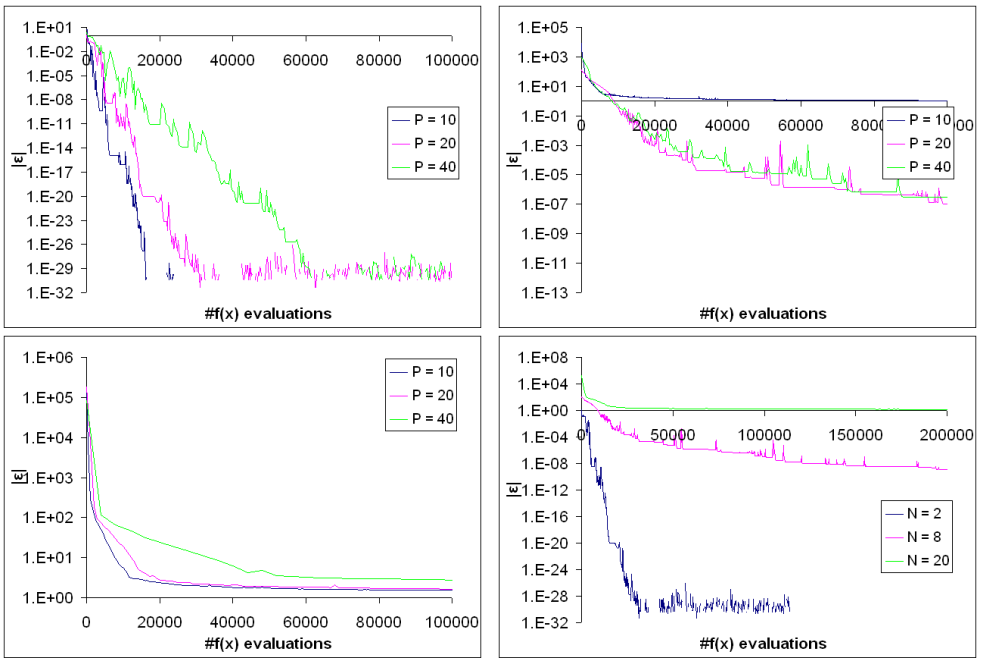

Figure 3.4: Validation for the Extended Beale Function.

From the results shown, one can assume the optimiser is able to find the minima (even if not always the global minimum) of the considered test functions. In the cases where the error does not tend to zero, a local minimum was found, which is a satisfactory behaviour, taking in account that this occurs for higher dimension of the test function. The small amplitude spikes that appear are due to the nature of the algorithm, as the particles' velocities are affected by the "'craziness"' factor. It should be noticed that the algorithm was capable of converging to points outside of the initial search space, which is a positive fact, as, in an engineering problem, the optimal point may be outside of the initial search space that was defined by the designer (that could know nothing of the problem *a priori*).

Also interesting to notice is the fact that, even for an increasing dimension of the problem, the algorithm is capable of finding the minimum even if the number of particles, and therefore, the number of evaluations per iteration, is lower than the dimension. Also to notice is the fact that there seems to be little advantage in having very large swarms. Although this is valid for the tested functions, some care has to be taken in extrapolating this conclusion to physical problems, where the objective functions may be noisier and not as smooth as the ones tested for the validation of the Particle Swarm Optimiser.

3.3 Artificial Neural Network

In the spirit of mimicking biologic processes that this work has followed, the concept of Artificial Neural Network (ANN) is introduced. There is no precise definition upon what an ANN is. Depending on the application, one can look at and define the ANN in the most suitable fashion.

An ANN is a network of simple processing nodes, whose behaviour is based on that of the biological neuron. In these networks, a learning capability can be achieved, larger as the number of nodes and node connection complexity increase. Including an ANN in the process changes the optimiser from being simply an EA, making it also a knowledge based algorithm, as the amount of information the optimiser uses in its process increases.

For this work, a simple feedforward neural network is used, the *multi layer perceptron*. In this model, the network is composed of a layer of input nodes, a single hidden layer (that one can perceive as being the processing layer) and linear output nodes. After appropriate learning, the network can then be understood as being a universal approximator [26] (as long as the dimension of the hidden layer is large enough) for any continuous function.

A typical model for the multilayer perceptron is shown in figure 3.5.

Looking at the processing nodes, the ones in the hidden layer of the ANN, the relation between the i inputs, x_i, and output of node j, y_j, the following:

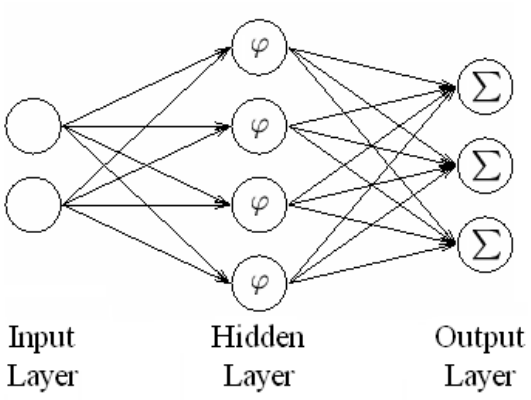

Figure 3.5: Multilayer Perceptron model (in this example with two input and three output nodes).

$$y_j = \varphi\left(\sum_i \omega_{i,j} x_i - \theta_j\right) \tag{3.5}$$

being φ the chosen activation function for the network, $\omega_{i,j}$ the weight associated with input i at node j and θ_j the threshold value for the activation function of node j.

And, naturally, being the output node a linear combination of the outputs of the hidden layer nodes, the function that describes the whole network is given by:

$$\overline{F}(x_i) = \sum_j a_j \varphi\left(\sum_i \omega_{i,j} x_i - \theta_j\right) \tag{3.6}$$

The universal approximation theorem applied to the ANN formulation guarantees that the standard multilayer feed-forward network with a single hidden layer (the multilayer perceptron) that contains a finite number of hidden neurons and sigmoid activation function is a universal approximator in $C(\mathbb{R}^n)$ [26]. Later [27], it was demonstrated that this property is maintained regardless of the chosen activation function, as long as it was a non-constant, bounded and monotone function. Unfortunately, nothing is formally demonstrated regarding the number of hidden neurons needed in order to satisfy a maximum approximation error.

3.3.1 Accelerator for the PSO

As demonstrated in section 3.2, the Particle Swarm Optimiser is a very capable algorithm to determine minima of the tested functions in a short amount of time. Time wise and in the overall application, one can expect the impact of the optimiser's operations to be close to none. The most time will be spent in determining the value of the objective

function(s), as this will involve some sort of aerodynamic and structural analyses which, at the depth level attempted in this work, will take a reasonable amount of time per particle.

Having this factor in mind, an acceleration process was investigated. The solution would have to be able to reduce the most time consuming process, i.e., determining the value of the objective function(s). Therefore, the accelerator process would have to be an interpolator, sufficiently robust to handle any type of continuous function. Such strategy has been adopted in the past, using meta-modeling, surrogate models, Kriging models, neural networks [28, 29, 30, 31, 32], to name a few. Using an ANN was the adopted choice in this work, as it is capable of interpolating any continuous function (as shown in 3.3). This only has one consequence: it forces to having objective functions that are continuous, a task that is simple to implement for the computational profits it is expected to provide.

There are many possibilities to implement a learning strategy on the ANN, being this process itself an optimisation problem. The learning process involves feeding the network with a training set, which is none other than the set of points to be interpolated. The training process should then determine the needed weights $\omega_{i,j}$ and threshold values θ_j for all of the network, minimizing the error between the output of the ANN and the known value for the interpolated points, i.e., minimizing the following objective function:

$$f_{objective} = \sum_{k} \left(\overline{F}(x^k) - F(x^k)\right)^2 \tag{3.7}$$

where k is the number of points in the training set, $\overline{F}$ is the output of the ANN (or approximate objective function) and F is the original objective function (the one to be approximated).

The PSO implemented to solve the MDO problem was a natural choice to perform the ANN training.

In order to use the ANN as an accelerator of the PSO algorithm, the following procedure was adopted: after a certain number of exact determination of the objective function(s) have been performed and the training set is sufficiently large (in order to have confidence in its capability to approximate the objective function(s) value), the ANN is trained and becomes an interpolator for the objective function(s); from this point on, each particle's objective function(s) is evaluated first by the ANN, and only the best candidates out of the whole swarm are evaluated exactly by the original objective function. This means that the worst particles in the swarm will not be truly evaluated, only approximately, which leads to potential time savings.

3.3.2 Pareto Front Detection

As already referred in section 2.1, often the optimisation problem is a multiobjective problem. The proposed approach in this work is to introduce the concept of Aggregate Objective Function (AOF) [33]. This concept allows to turn a multiobjective problem into a single objective problem through the operator:

$$f_{AOF} = \sum_{n} a_n f_{n,singleobjective}, a_n > 0 \tag{3.8}$$

However, prescribing the values for a_n will make the optimisation find an optimal point, not necessarily the global optimum. Actually, defining what is a global optimum solution may not be possible at all, for complex engineering problems such as aircraft design.

This new objective function guarantees that the optimal design point is a Pareto Optimal point, a point for which no single objective can be further optimised without optimisation losses in other single objectives. The collection of optimal points for any a_n weight combination is the called Pareto Front [33], which corresponds to a hypersurface in the search space. Knowing this surface is extremely useful, as tradeoffs between conflicting objectives can be performed *a posteriori* of the optimisation process, guaranteeing that whatever direction a solution may move from found optimal point, as long as it is on this surface, it will be an optimal point.

Again the ANN can be used successfully in order to determine the Pareto Front. Determining this surface using exact evaluations of the single objective functions would become impractical (or at least extremely demanding, from a computational cost point of view), as this would require re-running the optimisation process for a sufficiently large a_n weight combination set, increasing by orders of magnitude the time needed to resolve the Pareto Front.

The use of ANN's should allow a reduction in the computation of this important hypersurface by using its learning capabilities. After a large number of possible solutions have been analysed, these will constitute the training set for an ANN with a outputs equal in number to that of the single objective functions. Then, the exact same optimisation problem is run, but now using the ANN to determine the approximate value of the several objective functions, this for a large set of a_n weight combinations. The collection of optimal points found for each combination of a_n weights will result in an approximate Pareto Front.

Results of the integration of the ANN are not shown in this document, as some issues arose that prevent showing the gains that a finely tuned ANN could potentially demonstrate. Even though the advantage of using an ANN as an accelerator in optimisation

problems has been shown *per si* with significant reduction in computational cost [34], when comparing neural networks with other well established approximation schemes (such as Kriging models) [35], ANN's might not be competitive without fine tuning them. The lack of *a priori* knowledge, already discussed above, regarding the dimension and complexity of the network itself means that an investment must be made in determining the best topology of the network for the problem at hand and in a proper training mechanism, to guarantee minimum error and reliability of the solution away from the training points, while still offering a significant time saving.

Chapter 4

Aerodynamic Analysis

In the first section of this chapter, the main methods for aerodynamic analysis are reviewed in order to chose the most suitable one to MDO applications. In section 4.2, a description of the chosen method, the 3D Panel Method, and its computational implementation is explained. Then, in 4.3, the parameterisation for the panel method is explained and in section 4.4, the analysis options are explored and useful post-processing data is described.

4.1 Introduction

Presently, the best option to attain aerodynamic data, regarding results precision, is Computational Fluid Dynamics (CFD), as equations are solved in their least simplified form [36, 37]. However, a number of reasons discarded the use of this method in the present work. The biggest disadvantage of CFD is the computational cost of attaining a solution. Considering the number of evaluations needed before an optimal solution is produced as well as the memory and processing needs to solve a mesh for a full aircraft, this method is impracticable, with the available resources (personal computer). Furthermore, the automatic meshing tools typically provided with CFD codes rely mostly on excessive refinement in order to achieve quality solutions, with the associated computational cost.

The next method considered was the 3D Panel Method. Panel methods are techniques for solving potential flow. Therefore, their applicability would be reduced to incompressible flow and high Reynolds number and would fail to calculate the viscous component of the flow over the 3D body. However, after applying boundary layer corrections, calculated along streamlines of the potential flow and compressibility corrections, it is possible to achieve good accuracy outside its original bounds. Computationally, its cost is much lower than that of a CFD approach, and the time per analysis allows this method to be used with the optimisation algorithm that is considered in this work [38, 39]. Furthermore, it

also allows an easy way to interact with an FEM application, as both meshes can have a common surface where aerodynamic pressure is applied to the FE model.

More simplified methods are available, such as 2D Panel Method and Lifting Line Theory [39]. However, these methods present too much simplification and the associated inaccuracy, particularly when analysing non-lifting surfaces, such as the fuselage, which have a significant contribution to drag.

For its balance between precision and computational cost, the 3D Panel Method was chosen as the aerodynamic solver for this work.

4.2 3D Panel Method - *CMARC*

The incompressible potential flow model provides reliable flowfield predictions over a wide range of conditions. For the potential flow assumption to be valid for aerodynamics calculations, the primary requirement is that viscous effects are small in the flowfield, and that the flowfield must be subsonic everywhere. As said before, viscous effects can be included afterwards, by calculating the boundary layer effects along a sufficiently large number of streamlines. This approach is valid as long as these effects do not affect the previous flowfield too much, which is a valid assumption in the flow around an aircraft [38, 39], with exception to the interaction of wing elements and fuselage. In this work, such interactions are considered small enough to be negligible.

To solve the potential flow, the Laplace equation needs to be solved:

$$\nabla^2 \phi = 0 \tag{4.1}$$

in which ϕ is the velocity potential such that:

$$\phi = Ux + Vy + Wz \tag{4.2}$$

in which U, V, W are the longitudinal, lateral and vertical components of the flow velocity.

There is a number of ways to solve this equation. In panel methods, the adopted solution is to use singularities, functions which satisfy the Laplace equation to solve the perturbation velocity potential ϕ^*:

$$\begin{aligned} &\phi^* = ux + vy + wz\,, \\ &U = u + U_\infty, V = v + V_\infty, W = w + W_\infty \end{aligned} \tag{4.3a}$$

in which the subscript ∞ refers to the free-stream conditions.

As the equation is linear, superposition of solutions can be used. In this method, the

used singularities are typically sources σ and doublets μ, placed over the boundary S_{Body} of the body (see figures 4.1 and 4.2):

$$\phi^*(x,y,z) = \phi_\infty - \frac{1}{4\pi}\int_{S_{Body}} \left[\sigma\left(\frac{1}{r}\right) - \mu\mathbf{n}\cdot\nabla\left(\frac{1}{r}\right)\right] dS \tag{4.4}$$

in which the vector $\mathbf{n}$ points in the direction of the potential jump μ, which is normal to S_{Body} and positive outside of it, r the distance from point of coordinates (x,y,z) to the considered source σ, ϕ_i is the internal potential and ϕ_∞ is the free-stream potential, written as:

$$\phi_\infty = U_\infty x + V_\infty y + W_\infty z \tag{4.5}$$

Calculating the perturbation velocity potential in order to satisfy boundary conditions allows resolving the velocity potential through (4.3).

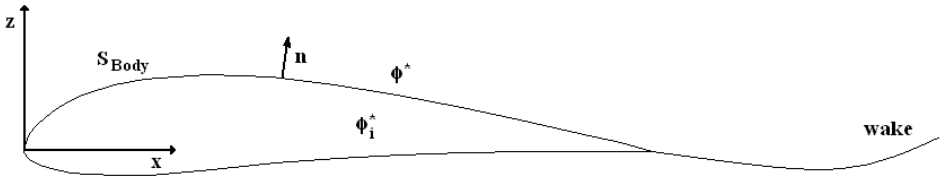

Figure 4.1: Potential flow over a closed body.

This formulation does not uniquely describe a solution since a large number of source and doublet distributions will satisfy a given set of boundary conditions. To uniquely define the solution of this problem, first the condition of zero flow normal to the body's surface must be applied. In the general case of three-dimensional flows, specifying the boundary conditions will not immediately yield a unique solution. Some physical considerations need to be introduced to fix the amount of circulation around the surface S_{Body}. These considerations deal mainly with the proper modeling of the wake (the three-dimensional equivalent of the Kutta condition [38]). Taking into account that sources are used to simulate the effects of thickness and doublets for lift generation (inducing circulation) problems, it is only natural to model the wake only by means of doublets and therefore (4.4) can be rewritten as:

$$\phi^*(x,y,z) = \phi_\infty - \frac{1}{4\pi}\int_{S_{Body}} \left[\sigma\left(\frac{1}{r}\right)\right] dS + \int_{S_{Body}+Wake} \left[\mu\mathbf{n}\cdot\nabla\left(\frac{1}{r}\right)\right] dS \tag{4.6}$$

In order to solve the problem, a suitable discretisation must be done, dividing the

body's surface into N panels and adding N_W wake panels (see figure 4.2). Collocation points are the points where the singularities are placed. Their positioning depends on several factors, such as the chosen model for the singularities themselves and the degree of the method.

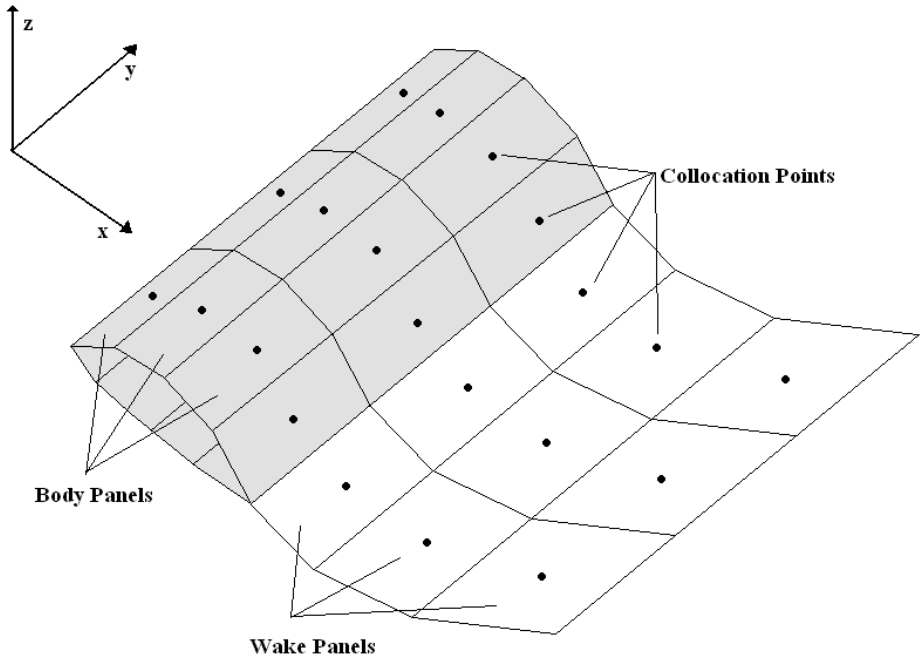

Figure 4.2: Approximation of the body surface and wake by panel elements.

Equation (4.6) can then be applied accordingly to each panel, in order to satisfy (4.1) and the prescribed boundary conditions, resulting in an algebraic system that can be solved.

After obtaining the flow velocity at each panel, the pressure coefficient C_p can be determined according to:

$$C_p = 1 - \left(\frac{|V|}{V_\infty}\right)^2 \tag{4.7a}$$

$$C_p = \frac{p - p_\infty}{\frac{1}{2}\rho V_\infty^2} \tag{4.7b}$$

win which $|V|$ is the flow velocity on a given panel, V_∞ is the free-stream velocity, p andρ are local pressure and density and p_∞ is static pressure.

If in incompressible flow regime, (4.7a) and (4.7b) become equivalent: as ρ the Bernoulli equation can be used to transform (4.7a) into (4.7b) and reverse.

Having selected the 3D Panel Method as the aerodynamic solver for this work, an adequate implementation of it had to be chosen. Developing a custom version of the

code would have been an excessive effort, taking into account the complexity of the task. Furthermore, a well established panel method code was available for use.

CMARC is an enhanced version of the *Panel Method Ames Research Center* code (*PMARC*) developed at NASA Ames Research Center, as a low order panel method that supports complex geometries. Having been developed in the late 80's and being derived from previously developed codes (*VSAERO* mainly), *PMARC* and its successor used in this work, *CMARC*, are considered to be robust and accurate, within the constraints imposed by the method's own formulation [40, 41, 42, 43]. For that reason, *CMARC* was chosen for the aerodynamic analysis.

4.3 Aircraft Parameterisation

In order to interact with *CMARC*, a custom preprocessing solution was created. Typically, a geometry is first created and only then is the discretization done, defining global and local refinements in order to generate the panels and wake lines. A different approach was taken here, being a parameterisation created for typical aircraft macro-components, such as the wings, stabilizers and fuselage. For a certain parameterisation, i.e., a certain aircraft shape, all of the panels are then declared in a file format accepted by *CMARC*. Some care had to be taken in the declaration of the input file, as the orientation of the panels defines which side of the panel corresponds to the external flow.

For wing-like elements, parameteres span, chord, dihedral angle, incidence angle, sweep and thickness are design variables (see figure 4.3). Span is a one-dimensional design variable, whereas all the other are given by a function:

$$DV_i = \sum_{k}^{p} a_k f_k(\bar{s}), \;\; \bar{s} = \frac{s}{Span} \tag{4.8}$$

in which f_k are polynomial functions of degree k, with p as the maximum polynomial degree, and dependent on the nondimensionalized span,$\bar{s}$ (s is the local span value and $Span$ is the full span that the element will have). The higher the degree p, the higher the variation the parametric model can suffer. a_k are the optimisation parameters, the ones that will be handled by the PSO. This function can be thought of as an amplitude modulation of a base value, modulation which depends on the set of values a_k for a given design variable.

Functions f_k are given by:

$$f_1 = \bar{s} \tag{4.9a}$$
$$f_2 = -4\left(\bar{s}\right)^2 + 4\left(\bar{s}\right) \tag{4.9b}$$
$$f_3 = 16\left(\bar{s}\right)^3 - 24\left(\bar{s}\right)^2 + 8\left(\bar{s}\right) \tag{4.9c}$$
$$\vdots \tag{4.9d}$$

These are based on Chebyshev polynomials, for $k > 1$, and are constructed in the interval $[0, 1]$, in such a way that $f_k = 0$ at $\bar{s} = 0$ and $\bar{s} = 1$.

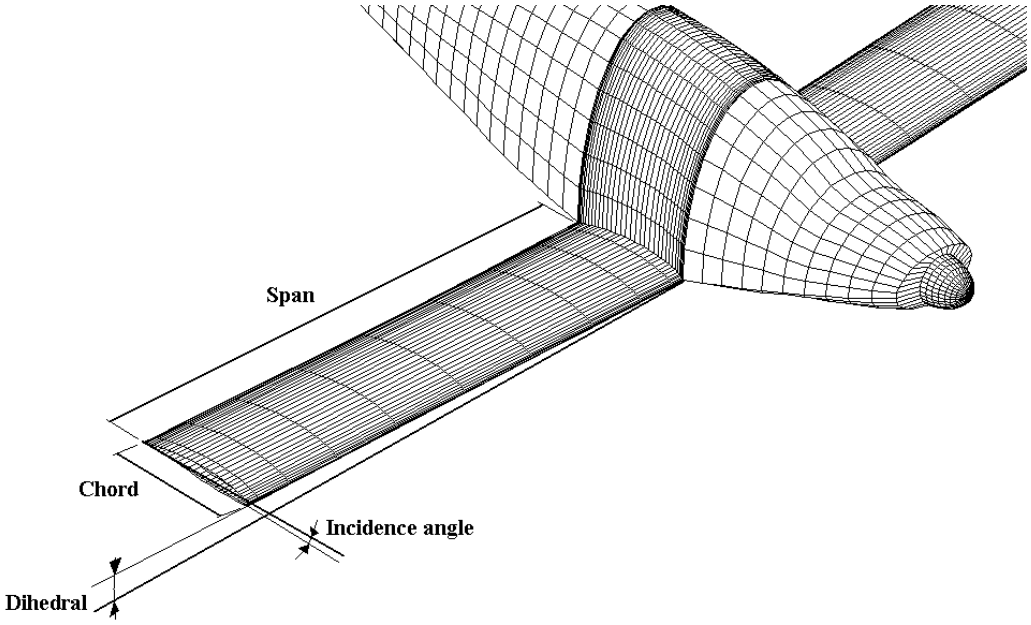

Figure 4.3: Design Variables for wing elements.

This approach can be extended to any other element of an aircraft, provided that a suitable parameter is chosen. In the case of a fuselage, for example, the longitudinal distribution of cabin diameter would be an example of this. This method also presents some advantages, as it makes possible for the design variables to assume different values in any point of the element using the same number of parameters a_k, regardless of the refinement of the discretization, i.e., number of panels, in the case of the aerodynamic solver.

All the parameters a_k for all the design variables will correspond to the values in the optimisation vector x_i in (3.1a).

In following subsections, a thorough description of the parameterisation implemented

is described.

4.3.1 Wings and Stabilizing Surfaces

Regarding the wing (and for that matter, any wing like element, such as stabilizers or winglets), the parameterisation starts with span and the airfoil, which, in this work, is not a design variable, but imposed *a priori* for each lifting surface, for simplicity. The airfoil is read from a file and its i coordinates stored in the ix2 $airfoil$ matrix, being this a nondimensionalised airfoil, with unit chord. Then, the other design variables are calculated from (4.8) replacing a_k with x_k:

$$Chord_j = Chord_0 + \sum_{k=1}^{p} x_k f_k(\bar{s}) \tag{4.10a}$$

$$Dihedral_j = Dihedral_0 + \sum_{k=1}^{p} x_{k+p} f_k(\bar{s}) \tag{4.10b}$$

$$Incidence_j = Incidence_0 + \sum_{k=1}^{p} x_{k+2p} f_k(\bar{s}) \tag{4.10c}$$

$$\vdots \tag{4.10d}$$

The values $Chord_0$, $Dihedral_0$, $Incidence_0$, etc., are needed for a correct parameterisation, as they define the design variables' values at span $\bar{s} = 0$, i.e., these are the base values that suffer the referred amplitude modulation. These parameters are also themselves optimisable values to be included in vector x_i in (3.1a).

In figure 4.4, an example is given for the parameterisation, where sweep (corresponding to the coordinates of the leading edge) and chord determined in function of span are shown and the result, in terms of panels for the aerodynamic solution. A similar process is also done for incidence, dihedral and the airfoil thickness (for a value of 1.0 the airfoil suffers no modification, other values will thin or thicken the airfoil).

Naturally, bounds can and should be applied to any of the design variables, so that any physical imposed constraints are transported onto the aerodynamic model. Even if no physical constraints are to be added, it is a good practice to apply them, in order to avoid generating a model that would have severe geometric distortion to the point where numerical convergence issues of the solution could appear.

Having the values for the design variables, the parameterisation of the panel's vertices can be done. Using *CMARC*'s aircraft coordinate system, where x is aligned with the longitudinal axis pointing downwind, y is to right of the aircraft and z is up, and be-

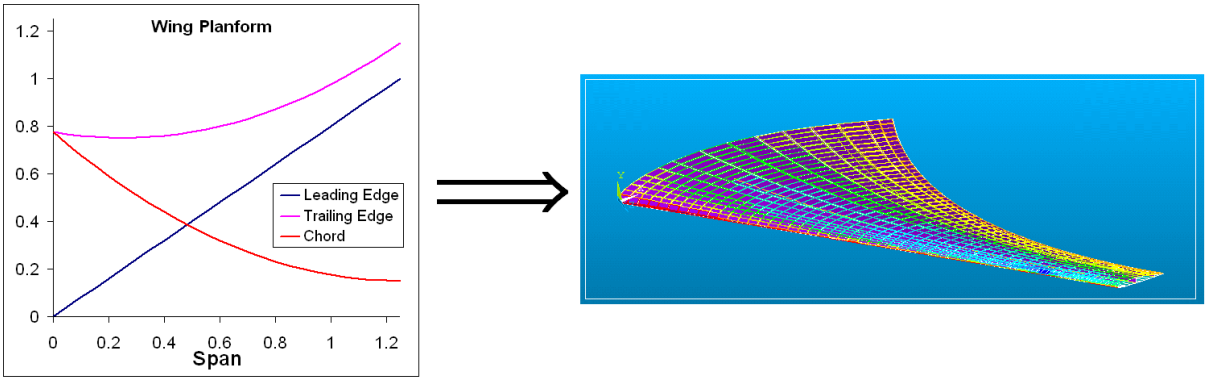

Figure 4.4: Example of Parameterisation.

ing $x_{i,j}$, $y_{i,j}$ and $z_{i,j}$ the matrices that contain the vertices coordinates (i and j correspond to the two directions in the parametrized surface):

$$x_{i,j} = airfoil_{i,1} \cdot Chord_j \cdot Sweep_j + \Delta x \tag{4.11a}$$

$$y_{i,j} = Span_j - airfoil_{i,2} \cdot \sin(\theta_j) \cdot Chord_j \cdot Thickness_j + \Delta y \tag{4.11b}$$

$$z_{i,j} = Dihedral_j + airfoil_{i,2} \cdot \cos(\theta_j) \cdot Chord_j \cdot Thickness_j + \Delta z \tag{4.11c}$$

where Δx, Δy and Δz are specific variables in the *CMARC* input file for each section declaration; the same principle applies for the incidence, that can be defined at the beginning of the declaration of each section, which avoids having to apply a rotation around axis y to all the vertices. Parameter θ_j is the angle of a section j in the plan yz, calculated from the derivative of the dihedral as a function of span.

A panel parameterisation of this sort is done for each wing-like element in the aircraft.

Doing the parameterisation in this fashion also allows to easily append a *wake line* to the correct location in the patch (in these cases, the trailing edge of the wing). Wake lines are needed, as it is from them that the wake, which allows a body to produce lift in 3D Panel Method, is generated (see figure 4.5).

4.3.2 Fuselage and non-Lifting Bodies

An aircraft fuselage is typically an oblong shape, for which a method such as that done in 4.3.1 is not suitable. A parameterisation based on a solid of revolution is the starting point for the process described below.

It is of extreme importance that contiguous panels share common vertices, as this defines the physical boundary of the aircraft, which can have no leaks. Furthermore, it is vital that there are no overlapping or crossing panels, as there has to be a clear definition of what is the inside and the outside of the model. Bearing this is mind, simply creating

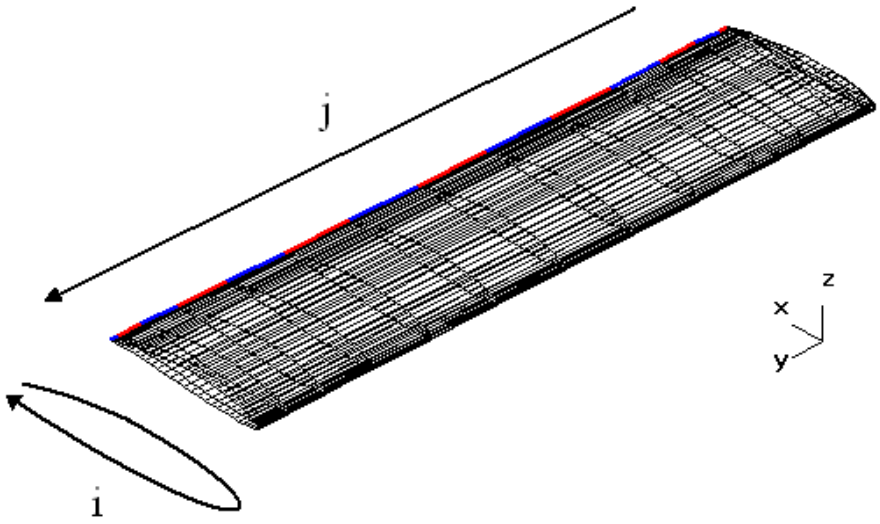

Figure 4.5: Wing parameterisation, with directions i, j and the associated wake line.

an ellipsoid to model this type of bodies is not a valid approach, as there would be an overlap with wing panels.

Therefore, the fuselage and any other non-lifting volume in the aircraft was modeled in several patches, as many as needed to properly cover the whole surface and correctly connect with the panels of wings and stabilizing surfaces. In figure 4.6, a detailed view of the panel connection between wing and fuselage is shown.

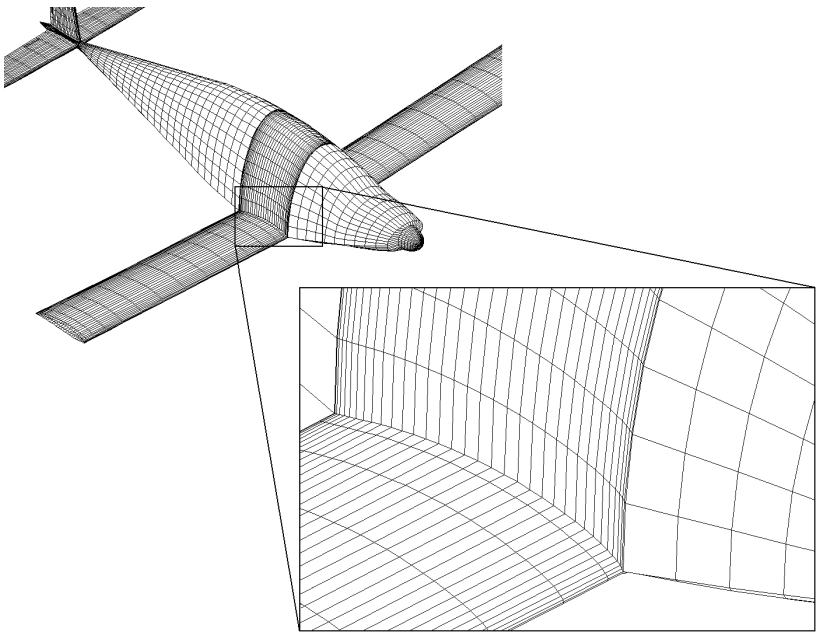

Figure 4.6: Mesh detail: wing-fuselage panel connection.

Two patches were defined to model a semi-fuselage, taking in account the needed deformation of a uniform grid to properly connect to wings and stabilizers. Looking ate fig. 4.7, upper and lower fuselage were created from specific lines, from hereon defined as *Root* and *Tip*. The *Root* line corresponds to a curve that starts at the end of the aircraft, follows along the symmetry plane xz, contours the base of the vertical stabilizer and continues along the symmetry plane. As for *Tip*, it can be compared to a waterline, as it begins at the same starting point as *Root*, continues along a line that goes until the horizontal stabilizer's root trailing edge, contours above the stab's airfoil, continues until the wing's root trailing edge, then above the wing's airfoil and finally reaches the nose of the aircraft.

Naturally, these curves must be constrained at some points: they must coincide with previously defined points, at the stabilizers' and wing's roots and they must share the same starting and ending points, if the end tip and nose are to be some sort of cone or dome.

The root and tip lines will define the fuselage and their shape can therefore be considered as design variables by themselves (the coordinates of control points used to generate splines, for example). The remaining shape consists in, at each station, to connect two points, one in each line, with an ellipse arc (see fig. 4.7).

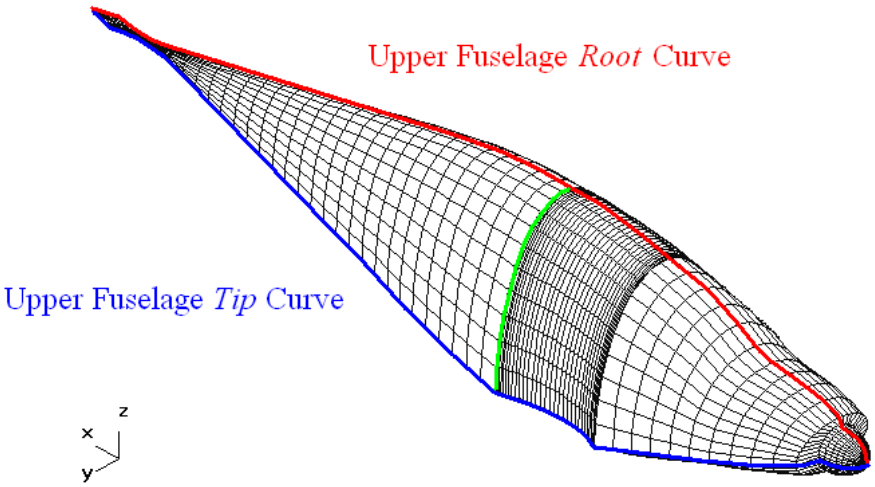

Figure 4.7: Fuselage mesh (ellipse arc highlighted in green).

As for the lower fuselage, a similar approach is taken, having a *Root* line that defines the lower limits of the fuselage in the xz plane, and a *Tip* line that must be coincident with the *Tip* line of the upper fuselage, except at the stabilizers' and wing's roots, where it must coincide with the under camber of the respective airfoils.

A note here must be introduced regarding the option of modeling the fuselage as a

non-lifting body. There is the possibility to include any body as a lifting body in the 3D Panel Method, provided a suitable wake is created. However, little information is available in order to choose a proper starting line for the wake, as it is highly problem dependent. This fact and taking into account that, from an aerodynamic design point of view, the fuselage should generally generate as little lift as possible (ideally, none at all, as the component of induced drag is very high if compared to that of a wing), it was decided not to include a wake in the fuselage, making it a non-lifting body. Aircraft concepts such as *Boeing's Blended Wing Body* or the *Joined Wing Aircraft*, in which the fuselage obviously plays an important role regarding lift generation, should not be treated in this fashion, which is suitable for more traditional aircraft design.

4.4 Analysis Options and Solution Post-Processing

CMARC allows in its input file structure to define all outer flow characteristics such as free-stream velocity V_∞, density ρ, viscosity μ, angle of attack α and sideslip angle β. Although these are prescribed in the file, all options for the analysis can be overridden in the *CMARC* application's own *GUI*. This feature becomes very useful, as the same geometry (the same input file) can easily be analysed in different conditions of α and β, in order to extract information on stability derivatives, for instance.

As for the output file generated, it contains all the information on the geometry, so that the input file is no longer needed for any post-processing. It further contains information on pressure and skin friction coefficients C_P and C_f, velocity components V_x, V_y, V_z, velocity $|V|$ and *Mach* number values on each panel, as well as numerical computations of the forces, moments and respective coefficients on each patch and for the whole geometry, both in wind axis and body axis. This avoids the need to create further routines to calculate these values from panel velocity alone and allowing the direct extraction of this information for use in objective function and for *FEM* loads application.

Chapter 5

Structural Analysis

In this chapter, the structural analysis method is presented in 5.1. Then, in section 5.2, a thorough description of the FE model is made, followed by some aspects of loads application to that model, in 5.3. The post-processing of the solution data is described in section 5.4.

5.1 Introduction

The use of the *Finite Element Method* (*FEM*) is widespread [44] and allows to create structural models from very basic complexity up to a complete aircraft, with all the associated coupled systems.

The structural model for the optimal design of a complete aircraft has to weigh two extremes: it had to be simple enough to allow a small computation time per evaluation, but complex enough so that important information could be extracted from the solution, such as weight, average and maximum stresses and deflection of the structure at key points, to name a few.

Regarding the software to be used,*Ansys*® was chosen, as it is an extremely complete package, from mesh generation to element types available to a fast matrix solver. Furthermore, it can be fully controlled through the command line with the use of an input file declaring all actions to be taken. This is a key feature, as the application described in this work is intended to be fully automatic and independent from external input.

5.2 Structural Model

As noted previously (2), MDO takes advantage of the interaction between different disciplines to eventually achieve a better optimal design. Thus, it is important to establish a system of information exchange between analysis modules that assures the models used

in different modules have as many common features as possible. In this case, it makes sense to use the discretization of the aerodynamic module as a starting point for the FE model. This allows to use the information from the output file of the aerodynamic analysis to generate the outer panels of the aircraft. It also allows to apply the exact pressure obtained from the aerodynamic analysis to the corresponding panel without any modification.

The basic functions of an aircraft's structure are to transmit and resist the applied loads, to provide an aerodynamic shape and to protect passengers, payload and systems, from the environmental conditions encountered in flight. These requirements, in most aircraft, result in thin shell structures where the outer surface or skin of the shell is usually supported by longitudinal stiffening members and transverse frames to enable it to resist bending, compressive and torsional loads without buckling. Such structures are known as semi-monocoque [3].

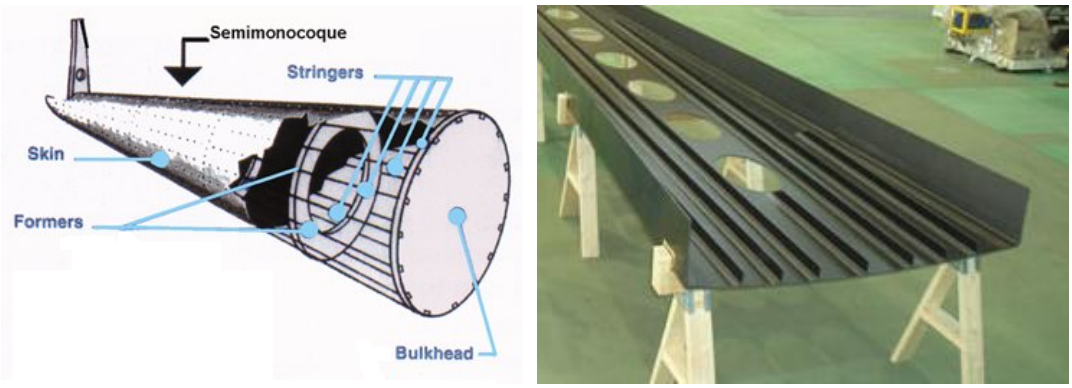

Figure 5.1: Stressed skin construction examples [3].

The method of designing aircraft structures using stressed skin construction will be adopted in this work. Therefore, there is the need to define exactly what structural components need to be modeled in the aircraft structural FE model according to their function. This will be addressed in the following subsections.

The material was assumed to be isotropic and the same all around the aircraft, as the purpose of this work is to demonstrate the advantages of using the MDO concept, and creating more complex material models would not improve much this purpose, unless the cost variable could be accounted for. Although composites would have been interesting to include, optimising fiber orientation and composite thickness would require a database of available fabrics and manufacturing solutions that would increase the number of design variables and therefore the computational cost.

The chosen material for the structure was T6061-T6 Aluminium, a widely used alloy in aircraft structures, for its low cost and good machinability and weldability [3]. Its mechanical properties can be found in table 5.1. For the studies presented in this work, a

factor of safety of 1.2 was used in conjunction with a maximum load factor of 2.5 in the aircraft, which results in the maximum admissible tensile strength shown.

Properties	Value	Unit
Tensile Yield Strength	289	MPa
Modulus of Elasticity	68.9	GPa
Shear Strength	207	MPa
Shear Modulus	26.0	GPa
Poisson Ratio	0.33	
Density	2700	$kg \cdot m^{-3}$
Max. Adm. Tensile Strength	96	MPa

Table 5.1: 6061-T6 mechanical properties [45].

5.2.1 Wing and Fuselage Panels

The stressed skin construction principle relies on having the skin, i.e., the outer panels of the structure, handle most of the stresses arising from aerodynamic loads.

As already referenced, *Ansys*® has a multitude of element types, from which the user should select the ones most suitable to solve the problem at hand. Regarding the wing and fuselage panels, the natural choice was on shell elements. Even though *Ansys*® has shell elements where the formulation allows the modeling of bending capability, it was considered that a more traditional approach would yield correct results, having the panels not withstanding bending by themselves.

Information regarding the geometry had to be available in order to generate the model. The adopted solution was to read the aerodynamic panel's vertices directly from the output file of the aerodynamic solver and use their coordinates as the corner nodes for the shell elements.

From the elements available in *Ansys*® , SHELL43 was chosen as this element is well suited to model linear, warped, moderately-thick shell structures. The element has six degrees of freedom at each node: translations in the nodal x, y, and z directions and rotations about the nodal x, y, and z axes. The deformation shapes are linear in both in-plane directions. For the out-of-plane motion, it uses a mixed interpolation of tensorial components [4].

This element was chosen for its robustness, as it can withstand a moderate amount of warping and shell thickness without compromising the solution quality. As for the used element inputs, the thickness was defined at each corner node.

In order to determine the best skin thickness at every point of the aircraft, without having a large amount of parameters to optimise, the following strategy was employed: for

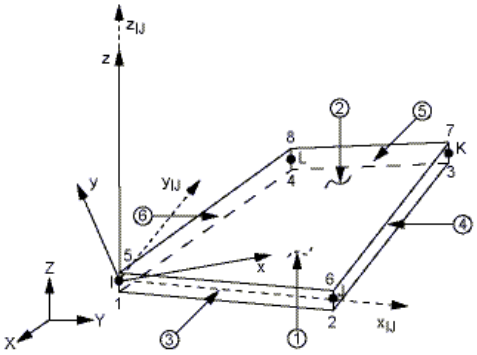

Figure 5.2: SHELL43 element [4].

each component of the aircraft (wings, stabilizers and fuselage) a skin thickness function was created:

$$Thickness(x, y, z) = Thick_0\ Thick(x)\ Thick(y)\ Thick(z) \tag{5.1}$$

in which each of the right hand functions are as in 4.8 and dependent on only one direction. Variable $Thick_0$ is also optimised, yielding a wide range of possible skin thickness distributions, with a relatively low number of optimisable parameters.

Figure 5.3 depicts an image of the obtained panels for the wing alone.

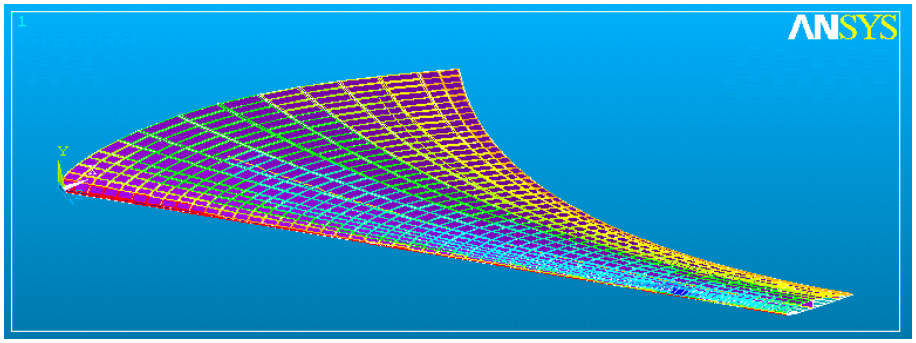

Figure 5.3: Wing panels and respective elements.

5.2.2 Spars

Spars can be considered to have two main components: the spar caps and the shear web. The caps, in a wing, correspond to the wing's outer shell. Therefore, in order to model a spar, all that is needed is to add the shell elements to create the shear web connecting the upper and lower spar caps. This is simply done in a similar fashion as in 5.2.1, as the nodes are already declared and the discretization of the airfoil and the

number of stations spanwise is known.

The spanwise distribution of thickness along the shear web and the chordwise position of the spar(s) are the design variables. Figure 5.4 shows an image of the wing, with a shear web included.

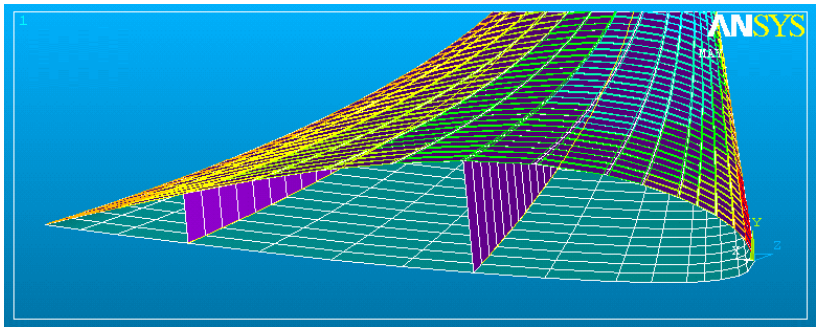

Figure 5.4: Wing panels with main and secondary spars generated.

5.2.3 Ribs, Stringers and Stiffeners

Stringers and stiffeners are needed in shell-based structures in order to avoid buckling of these surfaces and provide overall rigidity to the large panels on the aircraft without a large mass increase. Stiffeners are applied longitudinally to the fuselage and stringers applied spanwise to the wings. One can consider these structural elements to differ only in nomenclature, regarding their position on the aircraft.

Ribs are applied to the wing, in a chordwise arrangement, to avoid buckling or excessive displacement of the wing's upper and lower cambers, as this has obvious aerodynamic implications. They are also applied to the fuselage, to maintain its sectional shape.

Beam elements were used to model these features. The advantage of using beam elements vs. link elements is that these model in a more accurate way stiffeners and alike. As a link element models a simple bar, it does not take into account the area moment of inertia, assuming the bar is only capable of standing traction and compression, not bending. Typical sections for stiffeners have an area moment of inertia that is significant to the solution, increasing the overall stiffness of a given panel, therefore justifying the use of beam elements.

The chosen element for this task was BEAM4. This is a uniaxial element with tension, compression, torsion, and bending capabilities. The element has six degrees of freedom at each node: translations in the nodal x, y, and z directions and rotations about the nodal x, y, and z axes. Stress stiffening and large deflection capabilities are included [4]. This element was chosen as it allows for the input of cross-sectional area, area moment of inertia I_{zz}, area moment of inertia I_{yy}, torsional moment of inertia I_{xx} and thicknesses

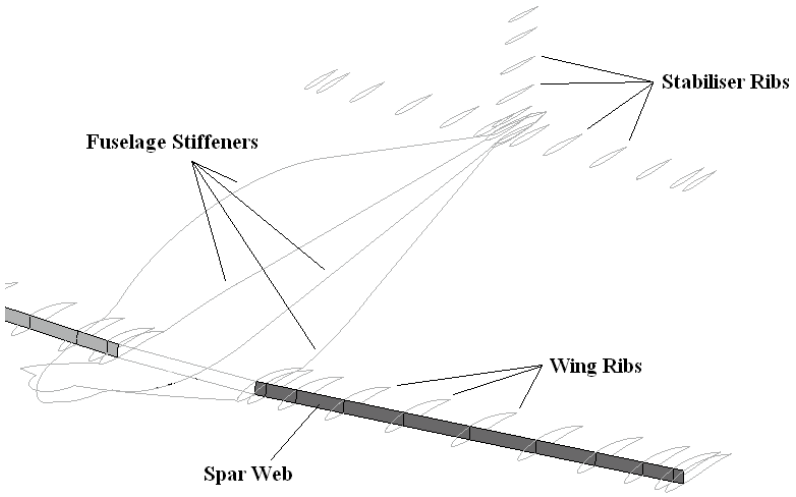

Figure 5.5: Wing spar, fuselage stiffeners and ribs.

along the z axis and y axis.

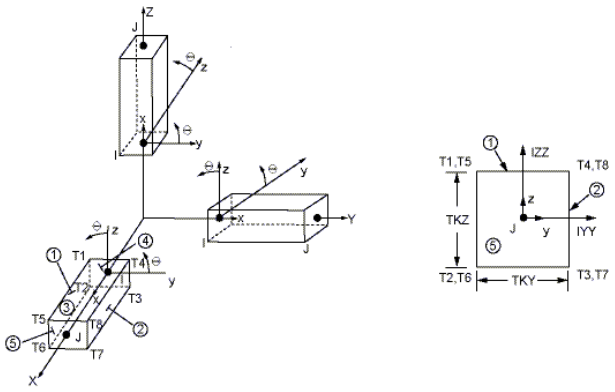

Figure 5.6: BEAM4 element [4].

Optimising the area moment of inertia alone would inevitably lead to a high value for that parameter during the optimisation process and not necessarily a feasible solution. Therefore, a "T"-section beam was considered, allowing to model ribs in a more realistic fashion. Having a fixed section geometry leads to having a fixed relationship between the stiffness of these structural features and their mass contribution.

In order to model these "T"-section beams, the geometry in fig. 5.7 was considered, where flange width w and stem height h were considered to have the same value and thickness t was considered to be 1/10 of the others. This yields only one parameter to be optimised: the beam's width (or height). Both area and area moment of inertia can then be determined.

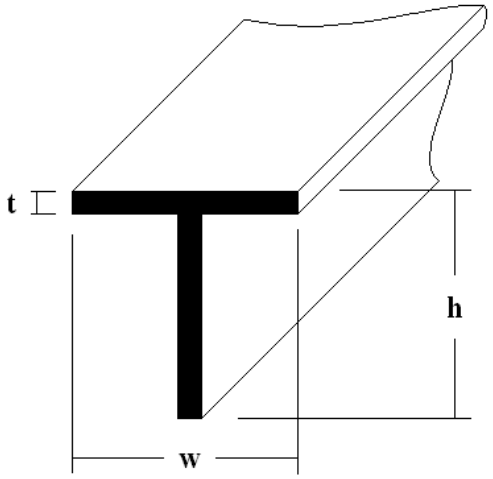

Figure 5.7: "T"-section beam geometry.

5.3 Structural Loads

In this work, the loads to be applied to the FE model are those an aircraft would be subject to during the course of its flight: aerodynamic loads and the structure's weight. As the element geometry and material density are known, structural weight is easily determined in *Ansys*® .

Regarding the aerodynamic loads, pressure is to be applied to the shell elements outward faces. This is a situation where the correspondence between aerodynamic and structural models is of great advantage. As the shell elements coincide, one by one, with the panels of the aerodynamic model, all that needs to be performed is reading the aerodynamic solver output file and apply the correspondent pressure to the panel. Recalling the C_p equation (4.7b):

$$C_p = \frac{p - p_\infty}{\frac{1}{2}\rho V_\infty^2}$$

as the term p_∞ is also present on the inside face of the shell elements, it is only natural to remove it from the loads to be applied to the model, being that the pressure applied only to the outside face of the shell elements will be:

$$p_{Panel} = C_p \frac{1}{2}\rho V_\infty^2 \tag{5.3}$$

5.4 Solution Post-Processing

Regarding the needed information to be extracted from the structural solution, two main aspects were considered: nodal displacement and nodal stress.

Nodal displacement (including rotations) allows to determine if deflections at key locations in an aircraft exceed maximum limits. For example, wing tip deflection may be desired not to exceed a certain value. More importantly, wing torsion should be constrained, as it affects the lift distribution, therefore penalising the aerodynamic quality of the solution. Fuselage deflection is also important, as it will have influence on the tail incidence.

Nodal displacement and rotation at key points therefore become variables in the structural objective function, although their value is only approximated, as a true fluid-structure interaction is not considered in this work.

As for nodal stress, a more generalised approach can be followed. Obviously, maximum allowed stress for the chosen material must not be exceeded in any point of the structure, and this is an objective that is easily fulfilled. However, in order to maintain structural mass as low as possible, all points of the structure would ideally be highly stressed. Therefore, another objective is to have the average stress on all elements be as high as possible. Naturally, an objective function that has the structure's mass as a variable is also determined.

Chapter 6

Framework Tool Implementation

The next sections will present the steps taken in the development of the application, in the same order as they are taken and in the end of this chapter, a flow chart of the application is shown.

6.1 Introduction

The development of the application described in this work was done in *C++* as it allows the use of classes, which offer some advantages which will be described further ahead in this chapter.

The application developed was designed to comply with the following main requisites: it would have to be fully independent from external user input during the optimisation phase; it would have to be able to interact with independent external applications, in particular, the aerodynamic and structural solvers, and it would have to be optimised to run in a reasonable amount of time.

Furthermore, the present tool was designed with a modularity approach in mind, so that, if the solvers were to be changed in the future, more disciplines included in the tool or changes are to be made to the models no major changes would have to be made to the core application, only to the individual blocks.

6.2 Implementation of the Optimisation Algorithm

6.2.1 Definition of the Optimisation Vector

After having defined the design variables in 4.3 and 5.2, it is important to define how these will be optimised in the context of the Particle Swarm Optimiser. As already referred in 4.3, all parameters for all the design variables are assembled in consecutive positions of vector x_i in (3.1a). This can be described in the following manner:

$$\left\{\begin{array}{c} x_1 \\ x_2 \\ \vdots \\ x_k \\ x_{k+1} \\ \vdots \\ x_i \end{array}\right\} = \left\{\begin{array}{c} a_1^{DV_1} \\ a_2^{DV_1} \\ \vdots \\ a_k^{DV_1} \\ a_1^{DV_2} \\ \vdots \\ a_l^{DV_N} \end{array}\right\} \tag{6.1}$$

for N design variables, each with a characteristic number of parameters a_k.

Naturally, the velocity vector v_i in (3.1b) will have the same dimension as x_i and each component of these two vectors will be intimately related.

6.2.2 Swarm Definition

From a programming point of view, in emphC++, the swarm consists of a vector of individuals characterised by a user defined class. This class contains the current position vector x_i, the current velocity vector v_i, the current objective function value, the best position ever vector $x_{i,Best}$ and the best objective function value ever. The class also contains a call to the objective function as well as a function to compare whether the current position is better then the best position ever of that particle. Finally, the class contains the prescribed values for ω, C_G and C_P, which are inertia, group confidence and particle confidence, respectively.

Creating the swarm as a vector has multiple advantages from a programming point of view. Any vector handling routines are trivially used in this context, thus avoiding the development of new routines. Furthermore, even a radical redefinition of the class and the features contained in it have absolutely no impact in the developed code for the optimiser operations.

6.2.3 Scattering of the Initial Swarm

The definition of the initial volume of the design space is of extreme importance. Defining the bounds for each component of (6.1) and defining the number of individuals in the swarm will determine the initial convergence behaviour of the algorithm. Having an initial volume that is very large may present some difficulty for the algorithm to find an interesting region to explore. Narrowing this space may not allow the algorithm to explore such areas or may lead to a high number of iterations before such regions are reached. Gaining sensitivity to this issue was something that happened throughout the work and can be considered part of the learning curve for anyone working in this field.

In order to create the initial population, a high quality random number generator was used [25]. All individuals in the population are assigned a random position within the initial search space and, in order to give the swarm an initial momentum, a random velocity vector is also assigned to each individual. Some care has to be taken assigning the initial velocity, as it should not induce too much of a random behaviour to the swarm (which is a possibility if velocity values are very high) as this would fall outside the behaviour of a pure particle swarm scheme, being closer to an initial random search scheme. Therefore, and only as "rule of thumb", the velocity initial space was defined as having an amplitude between 10% and 20% of the amplitude for the position initial search space, in each direction.

6.3 Individual Class Functions

In this section, a description of the functions included in the class structure is made. The first subsection refers to the determination of the objective function for each individual. In the second, the purpose of the use of the best objective function value ever is discussed.

6.3.1 Objective Function

In this function, the objective function value for each individual is determined. This was one of the most demanding tasks in the development of the application, as interacting with two external applications automatically, particularly when the aerodynamic solver was not developed with such functionality in mind, had a series of challenges that had to be overcome.

Running this function is, by far, the most time consuming process of the whole application (in the order of 5 7 minutes for the problem presented in 8). An overview of this function can be stated as follows: the input file for the aerodynamic solver is generated; the aerodynamic analysis is performed; the resulting output file is analysed, being the significant values used in the calculation of the aerodynamic objective function(s); the input file for the structural solver is generated; the structural analysis is performed; the resulting output file is analysed, being the significant values used in the calculation of the structural objective function(s) and, finally, any other objective function(s) that may require information from both the aerodynamic and structural solutions is calculated. Each of these steps in the application is described in more detail and presented in the flow chart at the end of the chapter, in fig. 6.1.

Generating the Aerodynamic Input File

In section 4.3, the parameterisation of the aircraft was described. In this step, the writing of the input file for *CMARC* is made. This involves the declaration of all analysis options such as angle of attack, free-stream velocity, static pressure and density, Mach number, time stepping interval and number of streamlines for boundary layer analysis.

In this block, the discretization refinement is chosen, regarding how many spanwise panels are used in the wing and stabilising surfaces and number of panels in the fuselage patches. Then, all the station point coordinates are declared for all of the patches, specifying whether the patch has a symmetric correspondent.

Finally, all the information required for wake declaration is written, regarding to which patch a certain wake line is attached, as well as the panel length of such wake. Tipically, 20 panels are used downwind.

Aerodynamic Analysis

CMARC was not developed to interact with other applications. The first versions of this application did not have a GUI, running only from the command line, which would have been an advantage. However, newer versions do have a GUI, which is just a user friendly front end that exchanges information with a DLL containing the actual panel method solver. As there is no information regarding the interaction with this DLL, the only possible option to interact with this application was to develop scripting code emulating user input information, such as input filename, output filename and enabling viscous analysis (in order to use boundary layer correction).

Some aiding applications to this process had to be developed, as there was no way to know when the analysis from *CMARC* was done. In order to do that, and exploring some *C++* options when calling external applications, a secondary application that polls the existence of an output file was developed. The function this block refers to calls this secondary application in waiting mode, which means it only proceeds when the secondary application has ended, therefore guaranteeing that the analysis of the output file is only started when there is a file to be analysed.

Analysing the Aerodynamic Output File

The information that can be found in the output file has already been described, in section 4.4. However, more actions are taken in this block. After reading the needed information, the aerodynamic objective functions are calculated. For instance:

$$f_{L/D} = -\frac{C_L}{C_D} \tag{6.2}$$

and

$$f_{C_M} = |C_M| \tag{6.3}$$

in which (6.2) and (6.3) are examples of typical objective functions to be minimised and C_L, C_D and C_M are lift, drag and pitching moment coefficients for the aircraft.

This block is responsible for writing relevant information in the history file: the values of all of the objective functions that were calculated, as well as all of the variables that are accounted for in their calculation; position and velocity for each individual are also written into the history file. Having such a file allows to have an historical perspective of the evolution of each individual in the swarm, not only the design variables values used, but also the quality of such solution, without having to store the input and output files of all individuals in a large number of time steps. It is easy to see how impracticable such strategy would be for a large swarm evaluated in hundreds or thousands of time steps, regarding the needed storage for the files.

Finally, all files that are not needed in the subsequent blocks of the application are deleted, a feature that will also be used after the analysis of the structural output file.

Generating the Structural Input File

The structural model was already described in section 5.2. This block is responsible for generating the structural input file, having as starting point the aerodynamic output file.

Recall the nodes declared for the structure correspond to the panel's vertices, from the 3D Panel Method. Then, the shell elements are declared, corresponding to the skin of the aircraft elements. Finally, reinforcements are modeled by beam elements, corresponding to ribs, stringers and stiffeners.

As for the declared model loads, a single command line models gravity, therefore including the self-weight as a load. As for the aerodynamic pressure in each panel obtained from the aerodynamic analysis, the four corner nodes of each panel are declared in the correct order to apply pressure on the outward face of the element, being the prescribed value described in (5.3); viscous drag is imposed in a similar way, but using skin friction coefficient, C_f, to calculate a pressure to be applied in the thickness face of the element, i.e., a force tangential to the elements.

Then, all needed constants for the elements are prescribed (see sections 5.2.1 and 5.2.3) and the constraints for the model are declared, regarding what nodes should have null displacements and/or rotations.

The last instructions to be written to this file are those that command a static analysis to be performed and post processing actions, in order to obtain stress, displacement at

important locations in the aircraft and structural mass. Using *Ansys*® own post processing capabilities and being the commands *FORTRAN* based, a custom output file can be easily generated, containing the information described in section 5.4.

Structural Analysis

Ansys® can be run in command line mode, provided that a suitable input file is given. This block is similar to the one for the aerodynamic analysis, being simpler, as it only involves calling the execution of *Ansys*® followed by the name of the input file generated before. This call is made in waiting mode: the execution of the main application waits until the *Ansys*® solution is done and the custom output file is ready for analysis.

Analysis of the Structural Output File

Similarly to the aerodynamic output file, the structural output file is analysed in order to calculate structural objective functions and to write relevant information in the history file. An example of a structural objective function is (6.4). Achieving a maximum displacement objective is done with this penalty function: the aircraft objective function will suffer a penalty if the wing tip deflection is larger than 10% of the wing span; the function is constructed in such a way that it is continuous, in order to guarantee that any aggregate objective function that is used will also be continuous, allowing the use of the ANN described in 3.3.

$$f_{\delta_{Tip}} = \begin{cases} 0, \ |\delta_{Tip}| < 0.1 \ Span \\ |\delta_{Tip}|^2, \ |\delta_{Tip}| > 0.1 \ Span \end{cases} \tag{6.4}$$

Objective Function Determination

As already referred in section 3.3.2, an AOF will be used as the ultimate objective function. Having determined all of the functions contributing to the AOF in the blocks above, and according to equation (3.8), the AOF is simply calculated by doing a weighted sum of all the contributing objective functions.

6.3.2 Best Ever Objective Function

This simple function allows to add memory depth to the individual and, therefore, to the swarm. This is not a inherent need for the basic Particle Swarm Algorithm. However, enabling this feature in the swarm means that a particle will always return to its best position recorded during its motion. For example, if a swarm starts moving cohesively to

a non-optimal region, the fact that one single particle has recorded in the past a better position will attract that particle and, therefore, the whole swarm to that position.

The function is simply defined in the following manner:

$$f_{Objective}\left(x_{i,Current}\right) < f_{Objective}\left(x_{i,Best}\right) \Rightarrow x_{i,Best} = x_{i,Current} \tag{6.5}$$

6.4 Enhancements to the Optimisation Process

The generic optimisation process has already been described in section 3.2.1. However, some specific aspects of the implementation have yet to be explored.

6.4.1 Limiters

In the version of the Particle Swarm Optimiser developed for this work, limiters were implemented after verifying a divergent behaviour of the swarm, when starting at a large distance from the optimal point. This is an understandable behaviour, as, under this scenario and at each time step, the velocity is incremented and may reach such a high value that the whole swarm overshoots the optimal point. This lead to impose limits on the highest value each velocity component may reach, taking in account the used time step:

$$\left|v_i\left(t_{Current}\right)\right| > \left|v_{i,max}\right| : v_i\left(t_{Current}\right) = \left|v_{i,max}\right| \cdot sign\left(v_i\left(t_{Current}\right)\right) \tag{6.6}$$

Some care has to be taken establishing these limits, as they must be loose enough to allow the swarm to move and converge rapidly to the optimal point. These limits should also be established taking in account their physical meaning. For instance, a maximum allowable variation of 1 cm on the wingspan of a commercial airliner obviously has little impact on the solutions, whereas the same variation in the thickness of a shell may be drastic. To reduce this effect, it is preferable to use nondimensional design variables.

These limits in velocity must also be established together with Δt, in order to guarantee both stability and fast convergence. After some tests with the test functions described in 3.2.3, it was determined that $\Delta t \approx 0.2$ and $|v_{i,max}| \approx 1$ (using nondimensional design variables) provides the best results, although these values may be problem dependent. When nothing is known about the problem, Δt should be reduced to avoid stability issues.

6.4.2 Global to Local Search: Inertia and Δt

Another implemented feature was to change the characteristics of the swarm, regarding its search behaviour. As referenced already in 3.2.1, lowering the particles' inertia value gives the swarm a faster convergence behaviour. Therefore, the chosen strategy was to start with an inertia value close to unit and slightly lower the inertia of the particles with time. This means that the particle starts with a good ability of doing a global search, trying to find the region where the global optimum is most likely to be. As the inertia gradually lowers, the swarm converges more rapidly towards the minimum within the region where the global minimum is, i.e., it converges rapidly to the global minimum.

Again, this feature should be used cautiously, as the swarm may fail to find an optimal point in a reasonable time if the inertia is lowered too fast.

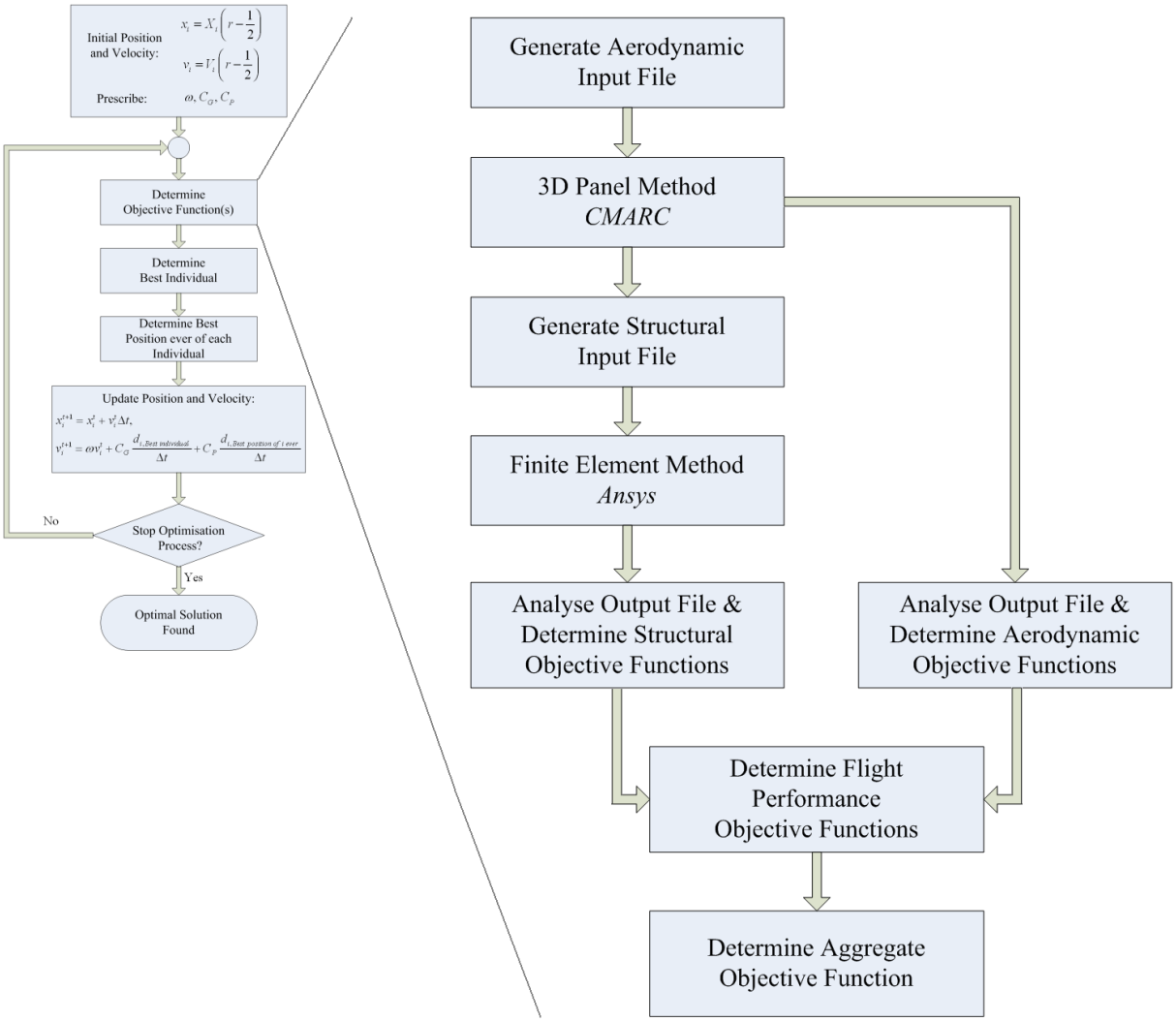

Figure 6.1: Application flow chart.

Chapter 7

Single-discipline Optimisation

In this chapter, four singlediscipline problems are solved. These, two in the aerodynamic and two in the structural domain, were used to test the developed application in the context of real life problems but also increase the knowledge of the behaviour of the optimisation process as a whole.

7.1 Aerodynamic Optimisation

In order to validate the developed application in a singlediscipline environment, two aerodynamic optimisation problems are solved. In the first problem, a simple rectangular wing is optimised with respect to its incidence distribution (section 7.1.1). In the second problem, winglets are added to a baseline aircraft and optimised with respect to chord, incidence and dihedral angles(section 7.1.2).

7.1.1 Rectangular Wing Optimisation

In this first example, a rectangular wing is optimised to yield maximum lift to drag ratio C_L/C_D. The design variable is the wing spanwise incidence distribution. Span, chord and airfoil are predefined and constant throughout the optimization process. The wing semispan is 5 m and chord is 1.25 m. The chosen airfoil was a NACA 63A612, which has maximum C_l/C_d at an angle of attack of 3°. Incidence angle at the wing root was set at 5° to have convergence to a solution other than the predictable optimal elliptical distribution of C_l[38] and the analysis was made at null angle of attack. Bounds were imposed on maximum and minimum local incidence angle to $-6°$ and $+6°$, respectively. The objective function for this problem was given by:

$$f_{Objective} = -\frac{C_L}{C_D} \tag{7.1}$$

In this case, the wing was discretized with 40 panels spanwise, with a cosine spacing (greater refinement near the wing tip). As the purpose was to test the capabilities of the optimizer, the swarm population was set to 6 individuals, 15 iterations were performed and the aerodynamic solution was performed without boundary layer correction, for faster results. As for the PSO specific parameters, the values used throughout this and the following chapter were: $C_P = 2.5$,$C_G = 1.0$,$\Delta t = 0.3$,$\omega = 0.8$ and $|v_{i,max}| = 1$.

Table 7.1 compares a constant incidence wing of 5° throughout the whole span, the best initial solution, i.e., the best random individual in the initial population and the final best solution in the population; the increase in C_L/C_D ratio in relation to the constant incidence wing is shown.

	L/D	to Constant Incidence
Constant Incidence	29.42	–
Best Initial Solution	40.26	+36.8%
Best Solution	45.10	+53.3%

Table 7.1: Gains from the optimisation process.

Spanwise distribution of incidence of the best solution obtained is shown in fig. 7.1. The evolution of the optimisation process is shown in fig. 7.2, with points representing each individual's score at each time step and lines representing the evolution of the average and best value of the objective function.

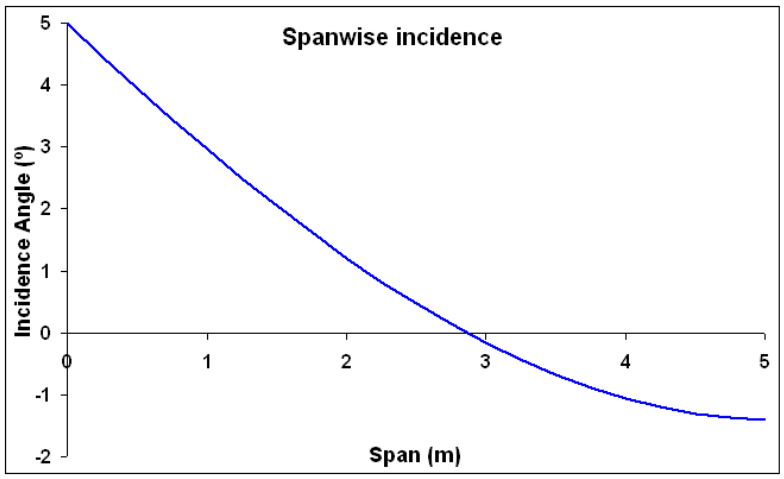

Figure 7.1: Spanwise distribution of incidence.

From the results above, it can be concluded that the optimiser performed well in the context of this problem. As expected, incidence distribution is such that the incidence angle decreases towards the wing tip, down to a negative angle. Being the wing's planform

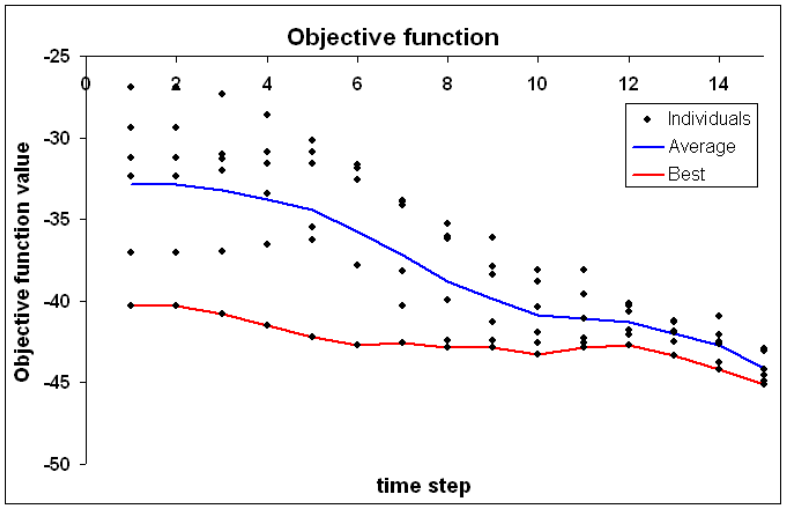

Figure 7.2: Optimisation process evolution.

a rectangular one, this is to be expected, as the induced circulation will lead to an effective angle of attack at the tip higher than that of the free-stream. The solution also presents a high variation of incidence in the root region, also to be expected, as the root incidence was set to 5°, a higher value than the airfoil's optimal C_l/C_d point.

Also from fig. 7.2, the effects of the velocity limiters are clear: not only is the behaviour of the swarm a damped one, without too much randomness, as the slope at the end of the process shows a non-accelerated behaviour, which would be expected without these limiters, avoiding an optimal point overshoot.

7.1.2 Winglets Optimisation

In this second problem, a full aircraft without winglets is modeled and analysed, in order to establish a baseline solution. The strategy is to create a set of winglets that will increase the aircraft's performance, while maintaining the original stability of the aircraft as much as possible. This would correspond to creating an add-on solution for an existing aircraft. Figures 7.3 and 7.4 show the panel discretization (without winglets) and the resulting C_P value for the aircraft without winglets, respectively.

The goal of the optimisation problem is to increase the baseline L/D ratio, without changing the C_M of the aircraft. Maintaining C_M as close to null as possible guarantees that no extra drag will be generated by the horizontal stabiliser, while trying to maintain static stability of the airplane.

Span, chord, dihedral, sweep and incidence are the design variables allowed to vary within the bounds presented in table 7.2. The bounds on span and area correspond to 10% of the semi-span of the original wing and 5% of the main wing area, respectively.

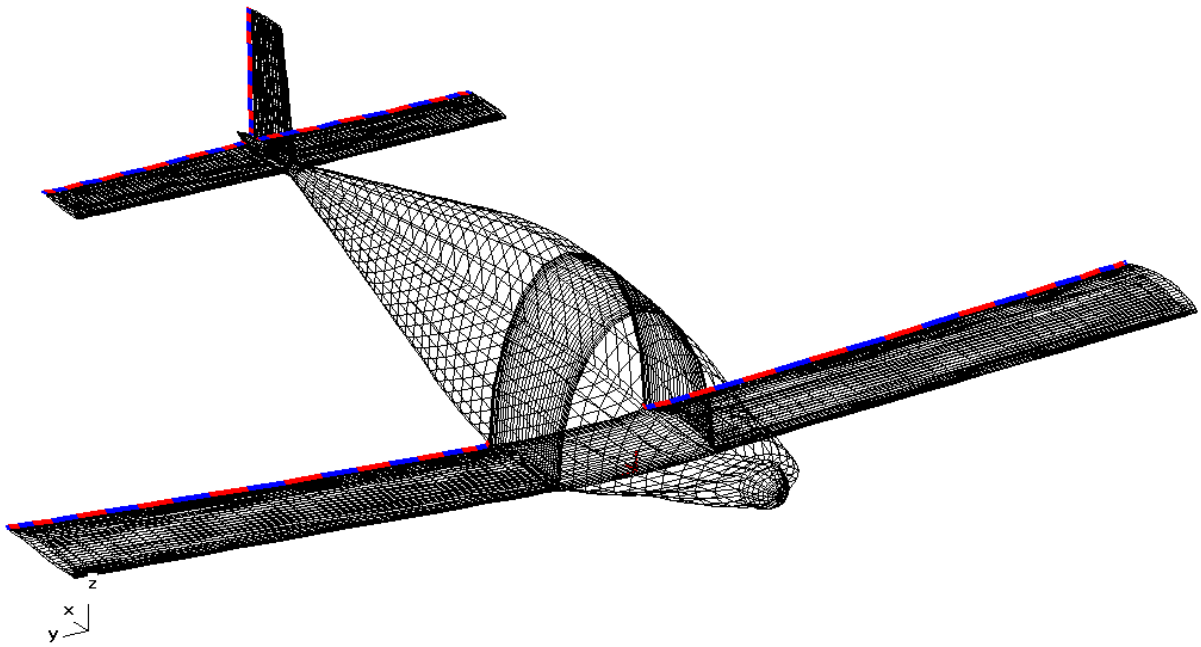

Figure 7.3: Panel Method of the full aircraft with wake lines highlighted.

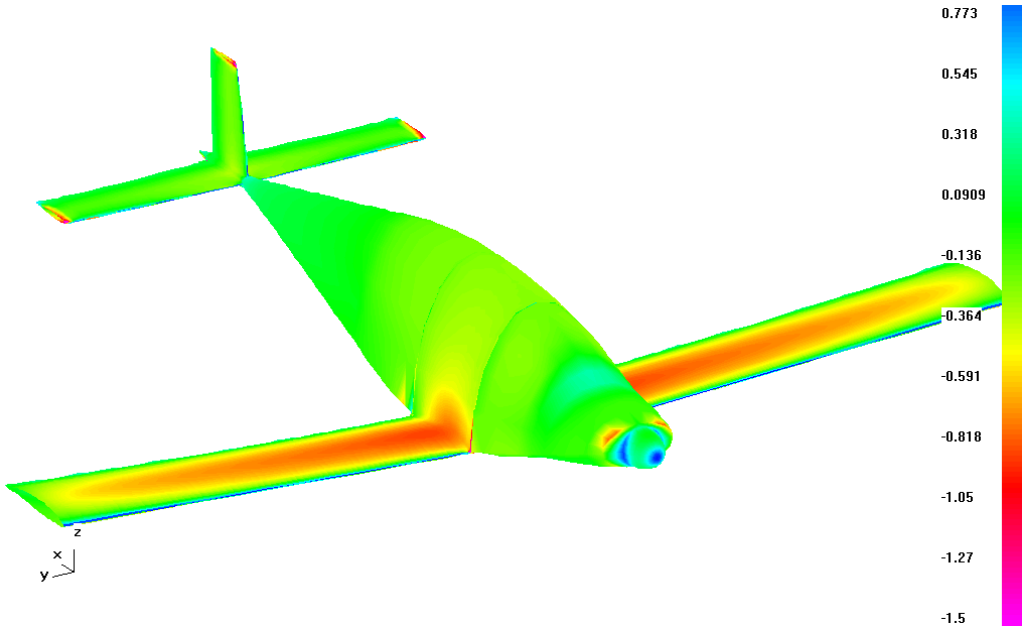

Figure 7.4: C_p distribution on the aircraft: $L/D = 25.62$.

The swarm had 6 individuals and the number of time steps was 8. The discretization had a total of 7164 panels.

Design Variable	Lower bound	Upper Bound
Span (m)	0.00	0.35
Chord (m)	0.10	0.90
Dihedral $(^{o})$	-75.0	$+75.0$
Sweep $(^{o})$	-75.0	$+75.0$
Incidence $(^{o})$	-6.0	$+6.0$
Area (m^2)	0.0	0.17

Table 7.2: Aerodyanmic DV bounds.

In this example, as in the following, and as already described in 3.3.2, a multiobjective optimisation was performed by using an AOF, aggregating the various objectives through the weighted sum of single-objective functions. In this particular case:

$$f_{Objective} = f_{L/D} + f_{C_M} + f_{Winglet\ Area} \tag{7.2}$$

in which

$$f_{L/D} = -\left(\frac{C_L}{C_D} - 25.62\right) ,$$

the term -25.62 simply applies a bias on the function, so that objective function values that are negative are better than the baseline and positive are worse (this is not be strictly necessary and was just used as a simple way to verify the optimiser's behaviour during execution);

$$f_{C_M} = 100|C_M| ,$$

the weight 100 serves the purpose of making this component significant in the AOF, as C_m values are typically 1 to 2 orders of magnitude lower than C_l, for the airfoil alone. This guarantees the original static stability by applying a penalty if the pitching moment of the aircraft is changed;

$$f_{\text{Winglet Area}} = \begin{cases} 0, \text{ Winglet Area} <= 0.17 \\ \left[100\left(\text{Winglet Area} - 0.17\right)\right]^2, \text{ Winglet Area} > 0.17 \end{cases}$$

this function applies a penalty if the winglet area is higher than 5% of the wing area, as prescribed before.

In order to have a fast convergence, the number of iterations was limited and no boundary layer correction was considered, which leads to lower drag than in reality.

	Baseline	With Winglets	Variation
$Lift(N)$	4732	5099	+7.8%
L/D	25.6	28.6	+11.7%
C_M	0	-6.3×10^{-4}	–

Table 7.3: Gains from the optimisation process.

In figure 7.5, the evolution of the optimisation process is shown, in which the objective function value is shown for each of the six individuals in the swarm during the prescribed eight time steps.

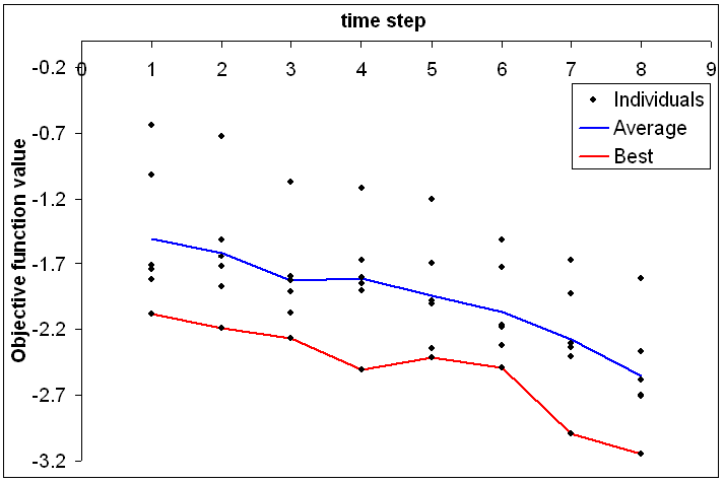

Figure 7.5: Optimisation process evolution.

The obtained solution has a very accentuated forward sweep and the typical high dihedral (see figures 7.6, 7.7 and 7.8), which is a valid aerodynamic solution, as any forward swept lifting surface will induce circulation from its tip to the root. This will reduce the vorticity that was previously generated at the free tip of the main wing, therefore having the effect of a larger apparent span and, consequently, a reduction of induced drag.

Being this an aerodynamic optimisation only, a structural analysis was not included. This inclusion would eventually lead to a more traditional looking winglet, without the accentuated forward sweep that could lead to structural design challenges.

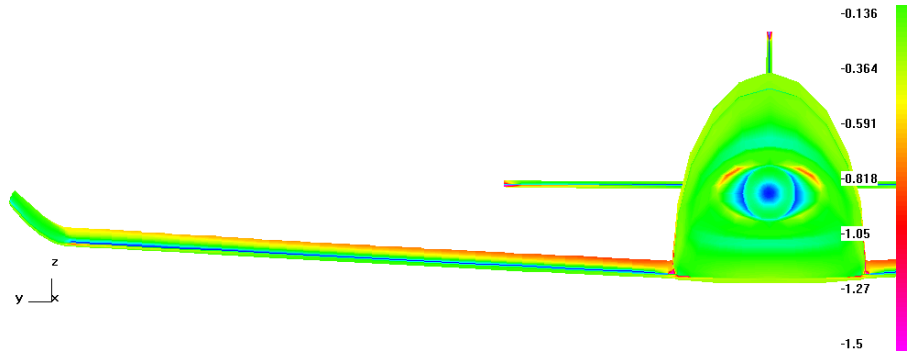

Figure 7.6: Resulting winglet (front view).

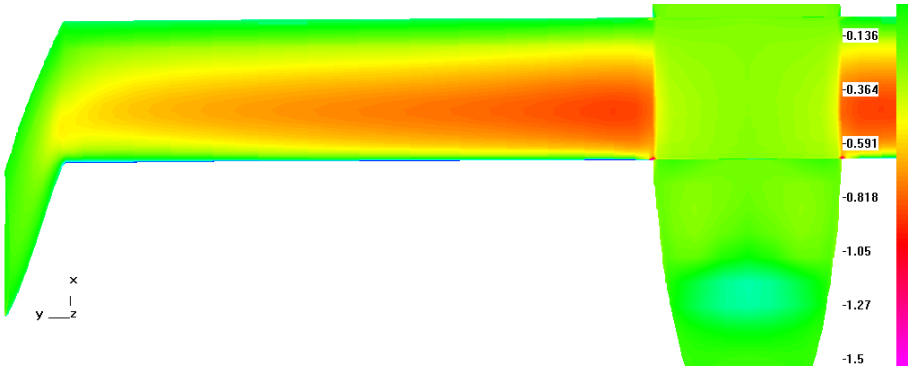

Figure 7.7: Resulting winglet (top view).

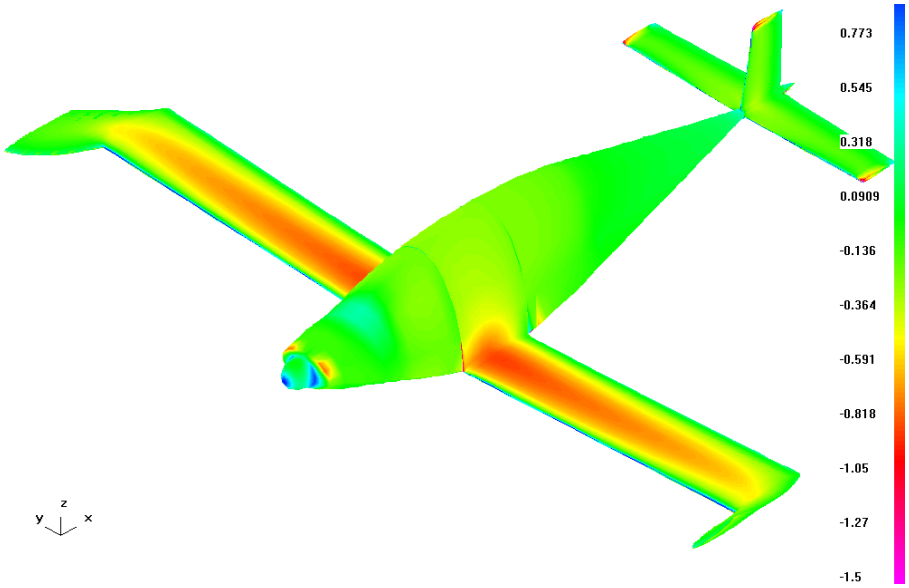

Figure 7.8: Resulting winglet (isometric view).

7.2 Structural Optimisation

As in section 7.1, two singlediscipline optimisation problems are solved, now in the structural domain. In the first problem, a beam's web height is optimised (section 7.2.1). In the second problem, the aerodynamic baseline solution from section 7.1.2 is used and structural parameters are optimised.

7.2.1 Beam Optimisation

In this example, a simple structural optimisation problem is explored. An aluminum I-beam is optimised for mass with respect to the web height along the span. Height is defined along the z direction. The geometry of this beam and its loadings is shown in fig. 7.9 (the shown web height distribution is only for illustrative purposes, as nothing is known about it *a priori* the optimisation process). The beam is fixed at its origin.

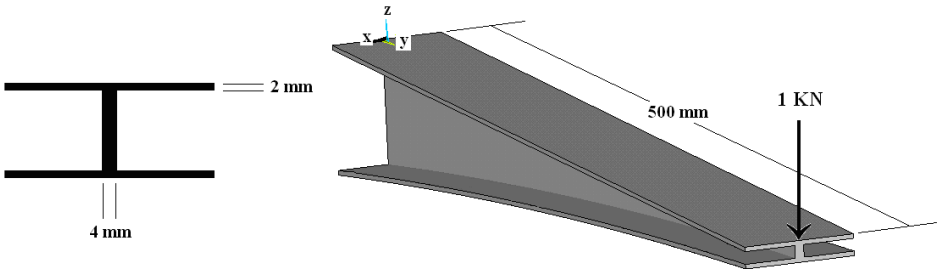

Figure 7.9: Beam section and geometry.

Limit bounds were imposed on maximum and minimum height of the beam's web, 5 mm and 70 mm, respectively. As before, the problem is multiobjective: weight is to be minimised, average stress maximised and maximum stress limit fulfilled. Thus, the objective function was constructed according to (7.3).

$$f_{Objective} = f_{mass} + f_{Average\,Stress} + f_{\sigma_{max}} \tag{7.3}$$

in which,

$$f_{mass} = 25\, m_{beam}\ ,$$

$$f_{Average\,Stress} = \frac{\sigma_{max} - \sigma_{avg}}{10}$$

$$f_{\sigma_{max}} = \begin{cases} 0, \ \sigma_{max} \leq 100\text{MPa} \\ 100 \left[\frac{\sigma_{max}-100}{100}\right]^2, \ \sigma_{max} > 100\text{MPa} \end{cases}$$

Regarding the first objective, both mass and average stress are considered: f_{mass} naturally penalises a solution with high mass, whereas $f_{Average\ Stress}$ will benefit a solution that has equal stress in the beam's cap, where this function is analysed. As for the function regarding maximum stress, this highly penalises solutions with stresses above the prescribed admissible stress, in this case chosen as 100 MPa.

Again the weights seen in these function serve the purpose of making all singleobjective function significant but giving more emphasis, in this case, to mass.

For this problem, the beam was discretised with 960 shell elements (320 for each of the caps and web), the swarm had 8 individuals and 20 iterations were run. Table 7.4 compares the best initial solution and the final best solution in the population regarding their objective function value, mass, maximum stress and average stress.

	Objective Function	Mass (kg)	σ_{max} (MPa)	σ_{avg} (MPa)
Best Initial Solution	16.34	0.523	74.7	42.2
Best Solution	11.04	0.399	102.1	91.9

Table 7.4: Gains from the optimisation process.

Fig. 7.10 shows the optimal web height distribution along the beam span. As expected in a problem of this sort, the solution shows an almost linear variation, reaching, at the beam tip, a value that was naturally determined by the lower bound imposed on this parameter (5 mm).

Figures 7.11 and 7.12 show the stress distribution on the upper spar cap, subject to tension. It shows the effect of having a component in the objective function that benefits solutions in which this element is stressed in a uniform way. It should also be noted that the maximum stress in the structure is higher than wanted by 2%. This can be explained as the objective function is continuous and reflects the care that must be taken when designing AOF functions such as this one, in which penalty functions are present. In this case it can be concluded that not enough weight was given to $f_{\sigma_{max}}$ when compared to f_{mass}.

Finally, figures 7.13 and 7.14 show the evolution of the objective function and mass for this optimisation problem. Looking at both graphs, it should be noted that, initially, a large number of individuals have very high objective function values, mainly due to maximum stress being higher than the allowed. As the optimisation process progresses, all of the individuals start approaching the optimum (iterations 8 to 15). Then, higher

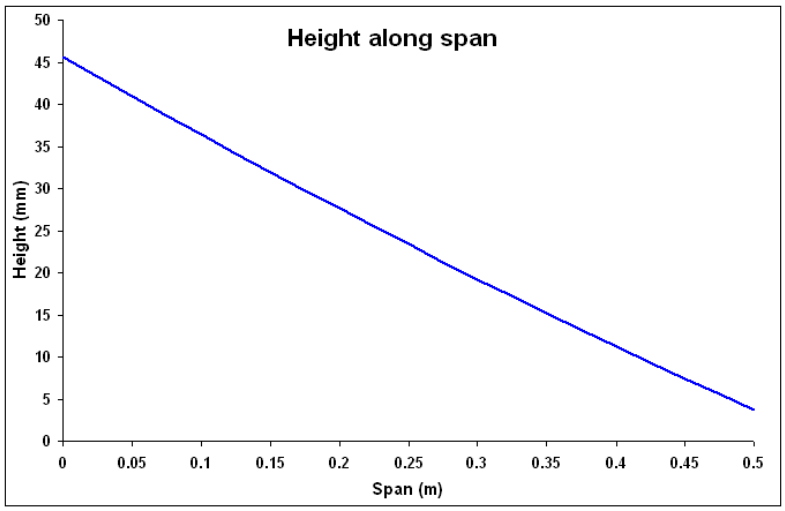

Figure 7.10: Optimal height distribution along the span.

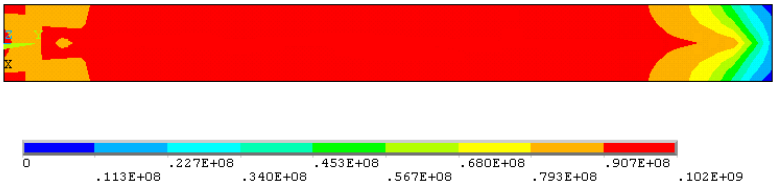

Figure 7.11: Stress in the upper cap, at the optimal configuration (in Pa).

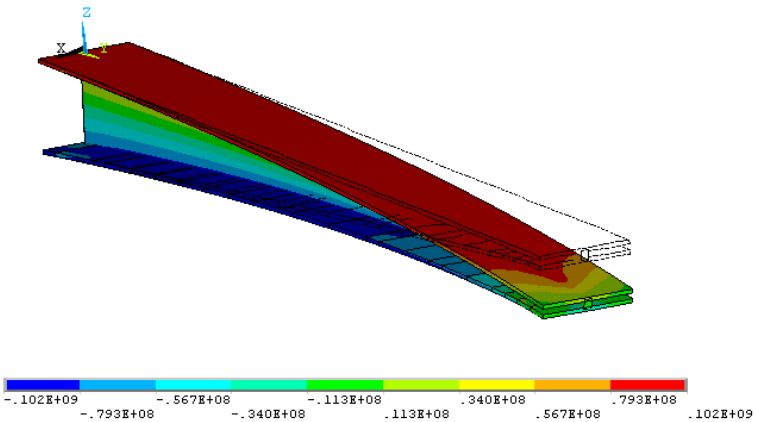

Figure 7.12: Optimal solution (σ_{yy}, in Pa, displacement scale 2:1).

values start appearing again, as all solutions now have a geometry that leads to low mass but, therefore, also high stresses. Naturally, if the optimisation process was left to continue until convergence, individuals would return to lower objective function values.

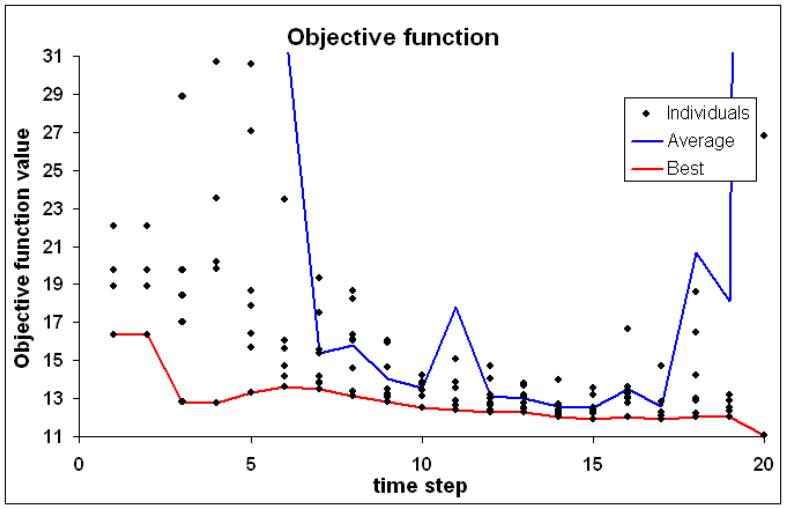

Figure 7.13: Optimisation process evolution.

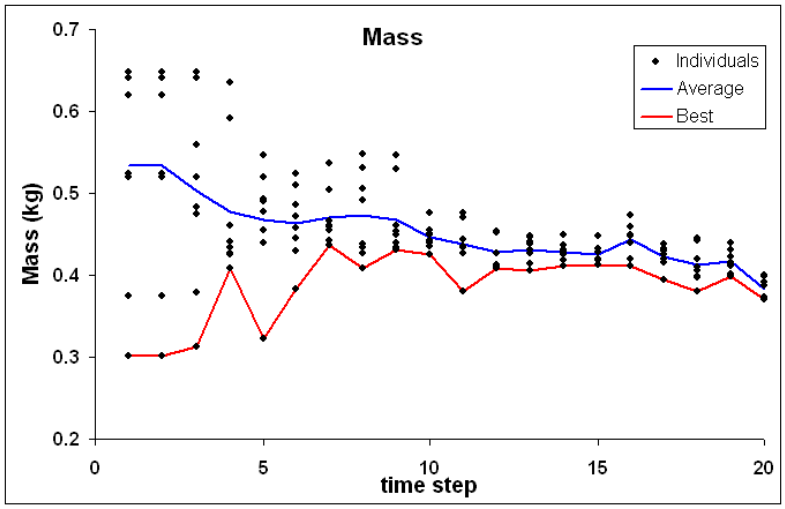

Figure 7.14: Beam mass evolution.

7.2.2 Shell Thickness and Ribs Optimisation

In this second problem, the aerodynamic baseline solution of section 7.1.2 is used and a structural optimisation is performed. Skin thickness and rib geometry parameter (as described in section 5.2) are optimised in the whole aircraft.

In this problem, the aggregate objective function considered mass, wing tip rotation, wing tip deflection and maximum verified stress. Structural mass is obviously a main factor in aircraft design and should be minimised. Wing tip rotation was equally considered in the objective function, as a structural solution that shows significant wing torsion will have its aerodynamic solution invalidated. This is particularly important as an exact fluid-structure interaction was not considered in this work. Wing tip deflection was included so that the solution is penalised if it is larger than 10% of the semispan. The same principle applies for maximum stress in the structure, if this value is higher than the maximum value for the considered material. Finally, we have the following AOF:

$$f_{Objective} = f_{mass} + f_{Wing\ Tip\ Rotation} + f_{Wing\ Tip\ Deflection} + f_{\sigma_{max}} \tag{7.4}$$

in which

$$f_{mass} = \frac{mass}{10} \ ,$$

$$f_{Wing\ Tip\ Rotation} = |Wing\ Tip\ Rotation| * 100 \ ,$$

$$f_{Wing\ Tip\ Deflection} = \begin{cases} 0, \ \delta_{tip} <= 0.2 \\ \left[25\left(\delta_{tip} - 0.2\right)\right]^2, \ \delta_{tip} > 0.2 \end{cases}$$

$$f_{\sigma_{max}} = \begin{cases} 0, \ \sigma_{max} <= \sigma_{6061-T6} \\ 10\left[\left(\sigma_{max} - \sigma_{6061-T6}\right)\right]^2, \ \sigma_{max} > \sigma_{6061-T6} \end{cases}$$

As very thin shells may be unfeasible, a lower bound of 0.635 mm was applied.

The swarm had 8 individuals and the optimisation process ran for 12 time steps. The FE structural model had a total of 7164 shell elements and 3678 beam elements.

The evolution of the objective function value is shown in figure 7.15. The optimisation process evolved as displayed in fig. 7.16, minimising mass – the main contribution to the AOF – while maintaining the optimal solution within the desired bounds, as shown in figure 7.17, in which wing tip displacement for the best individual in each time step is highlighted.

The off bounds value of stress at the vicinity of the wing root, shown in fig. 7.18, serves to show that the construction of the AOF should be done carefully. The optimal point for this particular AOF is one that gave too much value to a low mass, therefore partly sacrificing other objectives, in this case, maximum structural stress.

Again, it should be stressed that a small number of individuals and time steps were

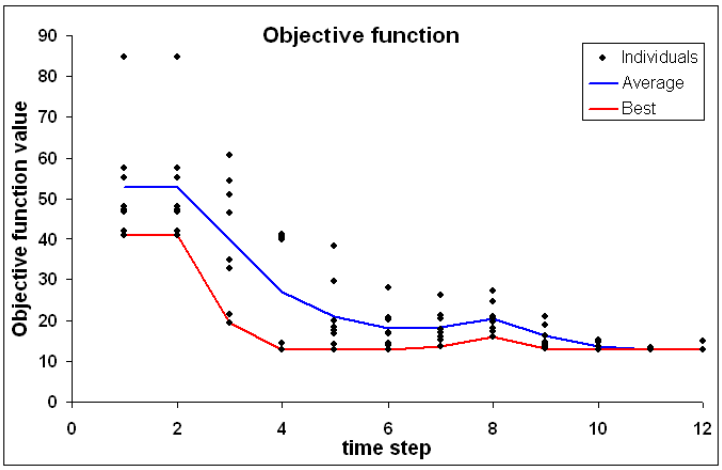

Figure 7.15: Evolution of the AOF.

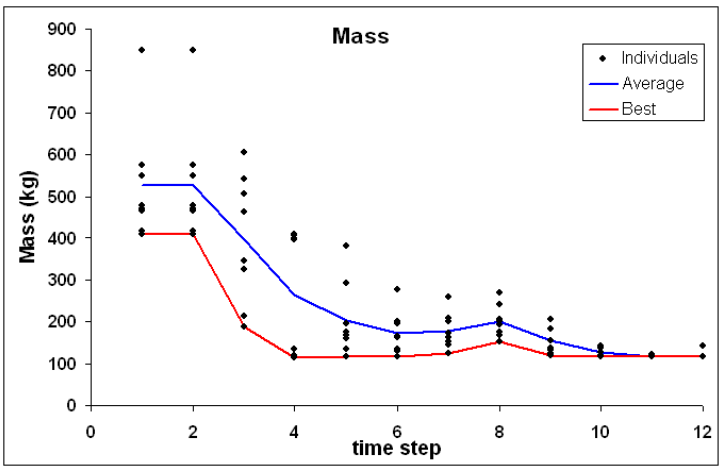

Figure 7.16: Evolution of the mass.

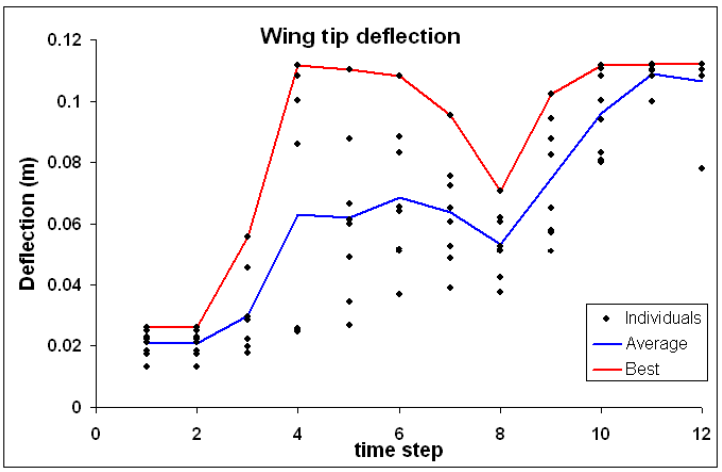

Figure 7.17: Evolution of the wing tip deflection.

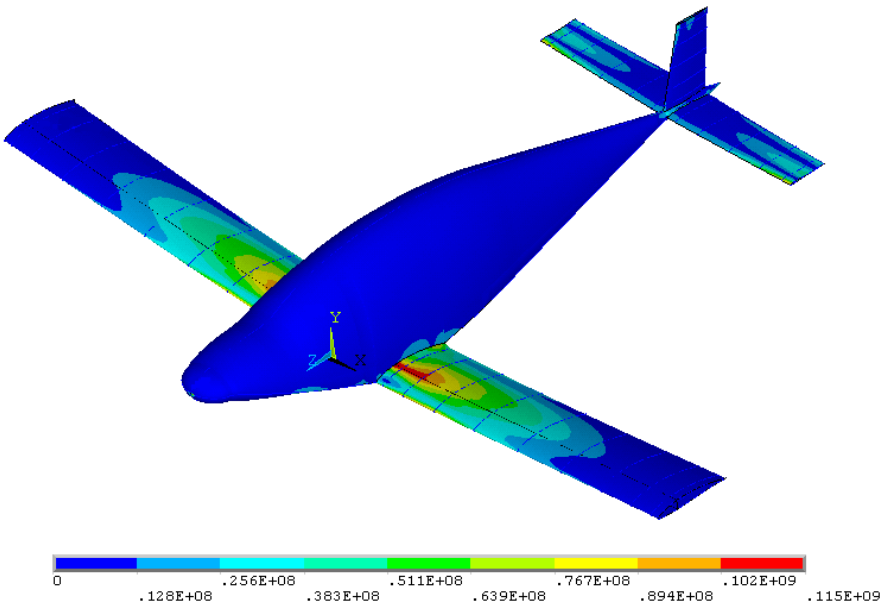

Figure 7.18: Stress Intensity obtained through FEM (in Pa).

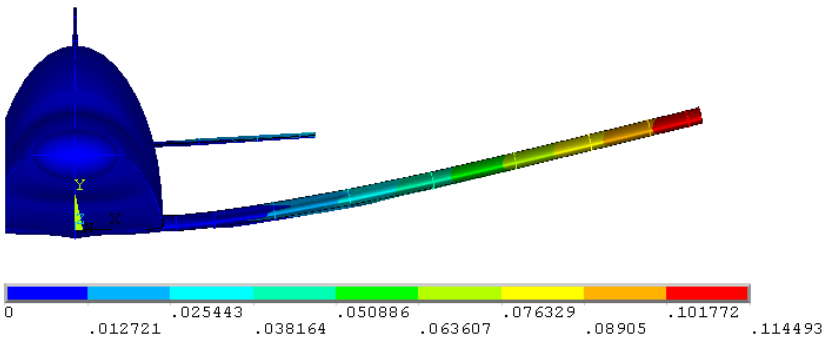

Figure 7.19: Wing deflection (in m; displacement scale 5 : 1).

used to maintain run time within reasonable limits, as the main goal of this work is the development of an MDO tool and these simple test cases serve the purpose of validation and as learning tools. Nonetheless, the algorithm was capable of minimising the objective function to what appears to be the optimal point, as the best individual's mass shows a convergence behaviour. Also to notice is the fact that the average swarm values also show a convergence towards the best individual, a behaviour that also suggests the optimal point was attained.

7.3 Lessons Learnt with Singleobjective Optimisation

In this chapter, some of the tests performed on the tool were described.

These tests proved extremely useful. The parameterisation strategy was tested and proved to be a valid way of having great diversity in the generated models for aerodynamic and structural analyses with a reduced number of optimisation parameters.

The interaction with aerodynamic and structural external analysis tools was extensively tested and allowed to increase the robustness level of the application in case of the analysis returning unfeasible results or in the rare event of a failed analysis.

The application proved capable of performing optimisation on singlediscipline problems, tested in two increasing levels of complexity: first, the simple wing (aerodynamic problem) and beam (structural problem); then, the winglet optimisation problem (aerodynamic) and the structural dimensioning problem. This proved to be a solid starting point to achieve multidisciplinarity.

The lessons learned from these tests were also useful in gaining sensibility for tweaking the PSO's parameters and in understanding the issues related to the construction of proper AOFs, in particular, the weight relation between singleobjective functions and penalty functions.

Chapter 8

Multidisciplinary Design Optimisation

In this chapter, a true multidisciplinary design optimisation problem was solved. To this end, aerodynamics, structure and basic flight performance were considered.

8.1 Statement of the Problem

The scenario was created for a small surveillance unmanned aerial vehicle (UAV): a blended wing body platform, with a central thicker body, designed for long range, flying at an altitude of 5000 m and speed of 70 m/s. The geometry of this aircraft is presented in fig. 8.1. The aircraft has a fixed span and sweep angle, $\Delta = 27^{o}$, and the central body has a fixed geometry. The airfoil, a *Wortmann FX 69-H-098*, was also constant throughout the span, except for thickness variations. This airfoil was chosen for its low C_{m0} (zero lift pitching moment) as the aircraft is tailless. However, this is not a reflex airfoil and therefore is not a natural choice for this planform, regarding the aircraft's static stability. This choice was intended to evaluate the optimiser's capability to create a stable configuration even with this airfoil. The geometry values for this particular aircraft are shown in table 8.1.

	Chord (m)	Incidence (°)	Thickness	Distance from root (m)
Station A	1.80	4.0	20%	0.00
Station B	0.80	3.0	10%	0.60
Station C	*variable*	*variable*	10%	2.50

Table 8.1: Aircraft geometry dimensions.

In terms of aerodynamics, chord and incidence of the wing (between stations B and C) were the design variables to be optimised. Dihedral and sweep angles were not considered, as only a thorough flight stability analysis would be able to solve for these parameters.

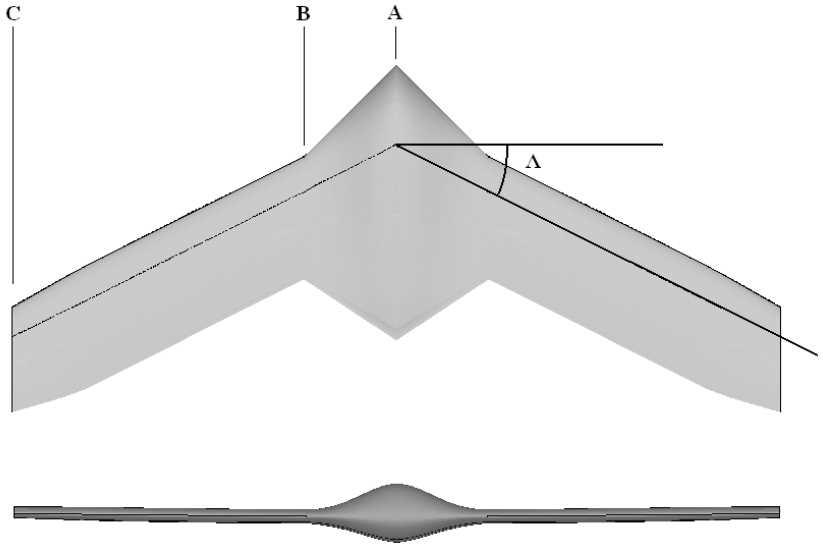

Figure 8.1: MDO problem: aircraft geometry.

As for the structure, beams were simulated by adding their respective beam web at 25% and 75% of the airfoil, being that the wing skin serves as their caps, which approaches a box-wing like construction and aluminum was chosen for material. As in section 7.2.2, skin thickness of the panels was the parameter to be optimized.

Table 8.2 summarises the lower and upper bounds that were applied both to aerodynamic and structural design variables. Recalling (4.8), and taking in account that $p = 2$, this represents a total of 13 parameters a_k that were optimised.

Design Variable	Lower bound	Upper Bound
Chord (m)	0.10	0.80
Incidence $(^{\circ})$	-5.0	$+5.0$
Panel Thickness (mm)	0.635	20.0

Table 8.2: Aircraft DV bounds.

The main objective to be fulfilled by this aircraft is long range. This result was calculated using the Breguet range equation (8.1) [38].

$$R = \frac{\eta}{SFC}\frac{L}{D}\ln\left(\frac{W_i}{W_f}\right) \ , \tag{8.1}$$

in which η is propulsive efficiency (a propeller propulsion was assumed, with $\eta = 0.8$), SFC is specific fuel consumption (here assumed to be $0.35\text{kg} \cdot \text{kW}^{-1} \cdot \text{h}^{-1}$), W_f and W_i are the weight of the aircraft at the final and initial points of its mission. Parameter W_i was calculated from the lift obtained by the aerodynamic solution and W_f was estimated

by assuming a 20% fuel fraction of the non-structural weight, derived from the structural solution:

$$W_i = \frac{L}{g}\,, \quad W_f = W_i - m_{fuel} = 0.8W_i + 0.2m_{structure} \tag{8.2}$$

As before, L is lift and D is drag for the full aircraft.

In this problem, the aggregate objective function was constructed to evaluate each solution taking in account mainly the range, but also included other functions to guarantee that wing tip displacement and rotation, maximum stress in the structure and pitching moment were within limits, in an approach similar to what is expressed in sections 7.1.2 and 7.2.2. The AOF is given by:

$$f_{Objective} = f_{Range} + f_{Wing\,Tip\,Rotation} + f_{Wing\,Tip\,Deflection} + f_{\sigma_{max}} + f_{C_M} \tag{8.3}$$

in which (Range is in km, wing tip rotation in rad, deflection in m, stresses in MPa and pitching moment in Nm):

$$f_{Range} = -\frac{Range}{20}\,,$$

$$f_{Wing\,Tip\,Rotation} = 100\,|Wing\,Tip\,Rotation|\,,$$

A penalty is added to the objective function if wing tip rotation is not null,

$$f_{Wing\,Tip\,Deflection} = \begin{cases} 0,\ \delta_{tip} \leq 0.125 \\ \left[20\left(\delta_{tip} - 0.125\right)\right]^2,\ \delta_{tip} > 0.125 \end{cases}$$

if wing tip deflection is greater than 5% of semispan,

$$f_{\sigma_{max}} = \begin{cases} 0,\ \sigma_{max} \leq \sigma_{adm} \\ 200\left[\frac{\sigma_{max} - \sigma_{adm}}{100}\right]^2,\ \sigma_{max} > \sigma_{adm} = 100\text{MPa} \end{cases}$$

if maximum stress is greater than 100 MPa and

$$f_{C_M} = 2\left(\frac{M_y}{50}\right)^2$$

if pitching moment is not null (choosing that the center of gravity of the aircraft is at 60% of root chord).

To solve this problem, 10 individuals were used, for 38 time steps. The aerodynamic

model had 2552 panels and the FE model had 2668 shell elements (the correspondent to the outer skin shell elements plus the beams' web shell elements).

8.2 Results

The evolution of the objective function value is shown in fig. 8.2. The optimisation process evolved as expected, maximising range (see fig. 8.3) - the main contribution to the AOF - while maintaining the optimal solution within the applied constraints. These are not shown here as the limits are being respected and such graphics would add little to this discussion.

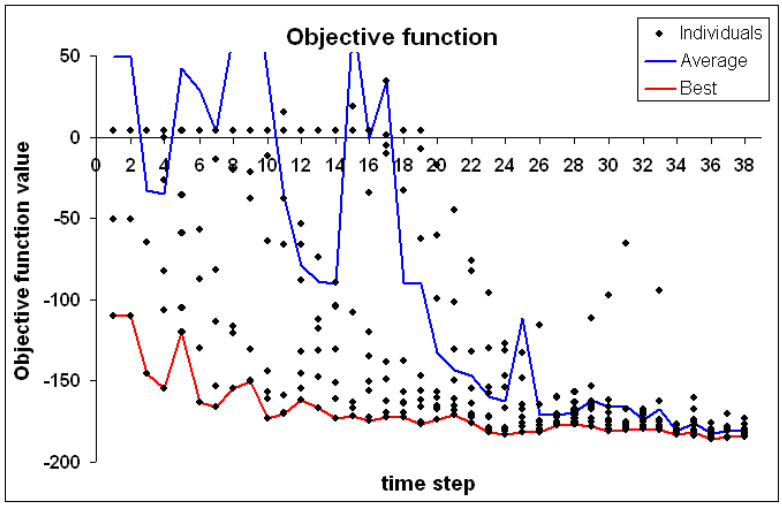

Figure 8.2: Objective function evolution.

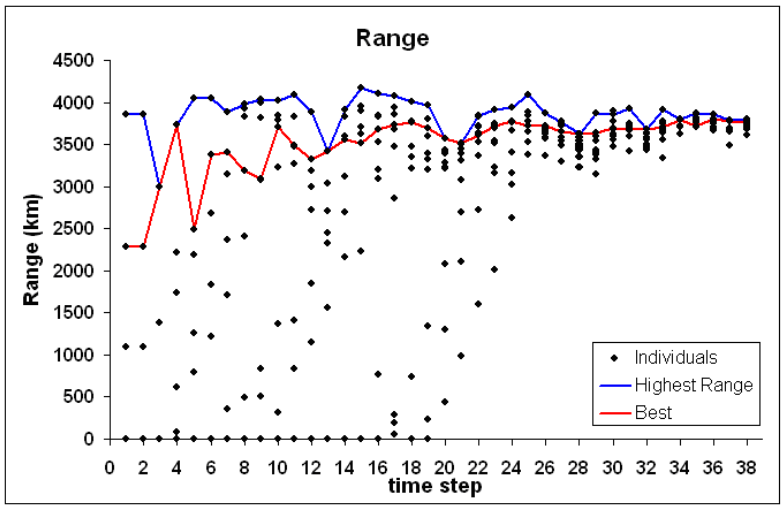

Figure 8.3: Range evolution.

Regarding the analysis of one particular individual in the swarm, it is clear in fig. 8.2 that an individual maintains its objective function as apparently constant, from step 1 to step 19. Because of the way range is calculated, precautions must be made in order to guarantee that fuel fraction is not a negative number, a situation that may occur if lift is lower than structural mass. In these situations, fuel fraction is set to zero, which in turn results in a null range, hence the constant value. From step 26 onwards, the individual has moved in the design space towards a point that allows for a non null fuel fraction and a non null range.

Table 8.3 compares the best initial solution, i.e., the best random individual in the initial population and the final best solution in the population regarding Range, Lift, L/D, structural mass, maximum stress, payload (defined as liftable weight other than structural mass and fuel) and objective function.

	Best Initial Individual	Final Solution	Variation
$Range\,(km)$	2285	3772	65%
$Lift\,(N)$	2465	2267	−8.00%
L/D	24.15	24.57	1.70%
$mstructural\,(kg)$	117.5	38.1	−68%
$Payload\,(kg)$	107.2	154.6	44%
$\sigma_{max}\,(MPa)$	25.5	97.7	283%
$Objective\ Function$	−110.1	−184.5	68%

Table 8.3: Gains from the optimisation process.

From the analysis of these values, it is clear that there was optimisation in both aerodynamic and structural fields, but most importantly, optimisation in a coupled environment. Analysing only aerodynamic performance in lift and L/D ratio shows two similar solutions but their differences arise when the results of the structure are considered. The optimised solution shows a maximum stress value very close to the allowed maximum, guaranteeing that the structure is capable of handling the aerodynamic loads, yet light enough to allow for a long range. The notion of optimisation in a coupled environment becomes even clearer when intermediate results are analysed, as during the optimisation process some solutions had higher lift or L/D or lower mass, but always with a higher value of the aggregate objective function than the found optimal solution.

The other important results of the optimized solution are a wing tip rotation of 0.12º, low enough not to influence the aerodynamic solution (as no coupled aero-structural analysis is performed), a wing tip deflection of 31 mm and a pitch down moment of 70 Nm. Recall that a statically stable solution is the goal and the aircraft should present a null pitching moment. Although the achieved result for this configuration was not null, it is very low for an aircraft of these dimensions and mass, being easily compensated by

control surfaces or the slight shift of the center of gravity.

Figures 8.4 and 8.5 show spanwise chord and incidence distributions. As can be seen in the chord distribution graphic, chord is almost constant throughout the wing. Even though this may not favor the best L/D ratio, it adds area to the wing, having a more significant effect on range (by means of a higher fuel mass) than another distribution. Regarding incidence, the graphic shows a decrease towards the wing tip, which favors not only the L/D ratio by means of a more favorable lift distribution (somewhat overcoming the lower than optimal lift distribution given by the almost constant chord distribution) but also has significant effects in terms stability. As noted before, the chosen airfoil is not a reflex airfoil, which would be a more natural choice for flying wing configuration, due to its lower pitching moment. However, the optimal incidence distribution combined with the accentuated backward sweep of the wing have a balancing effect on the aircraft regarding pitching moment, guaranteeing static stability.

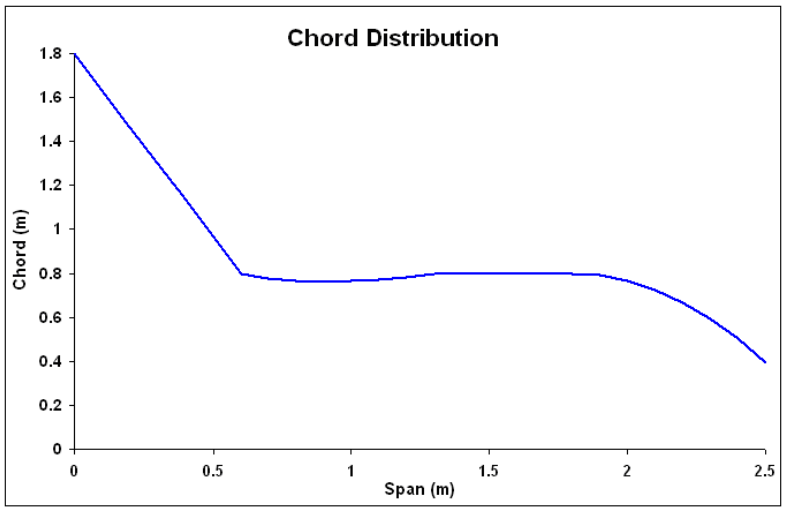

Figure 8.4: Spanwise chord distribution.

The aerodynamic design variables value distributions result in the C_p distribution on the aircraft shown in fig. 8.6. It is clear in this picture that the central body has a significant contribution to lift and it is also clear that the 3D Panel Method is capable of capturing the tridimensional effects of the flow, as the wing root is the section with the lowest C_p value, result of the induced circulation of the body area onto the wing.

Fig. 8.7 shows the stress intensity on the optimised solution. As expected, the wing root area shows higher stresses than the rest of the structure, particularly near the leading edge, as not only upwards bending occurs due to lift, but also backwards bending, result of drag.

Fig. 8.8 shows the thickness distribution from which the FEM solution was obtained, showing a higher value in the areas where stress is higher and thus confirming the opti-

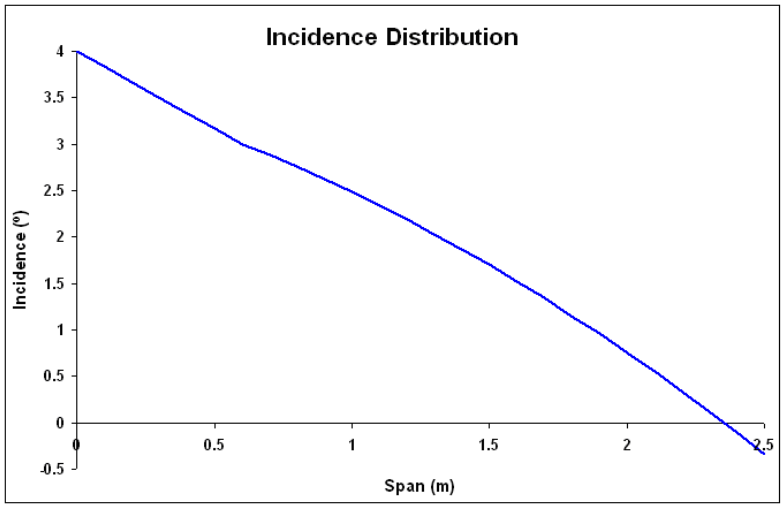

Figure 8.5: Spanwise incidence distribution.

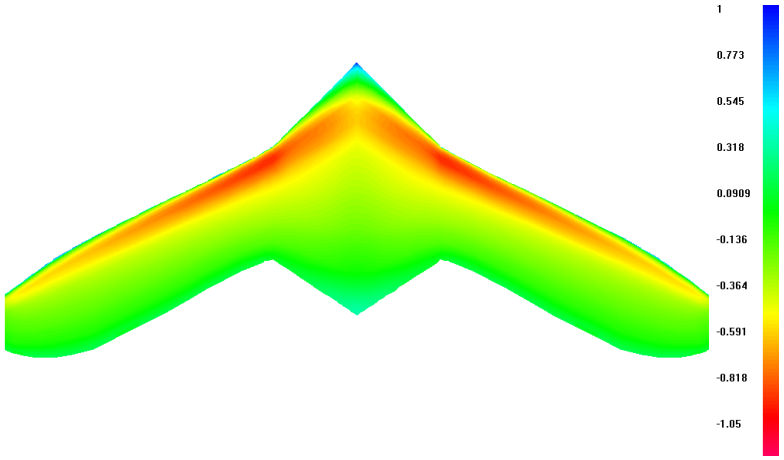

Figure 8.6: C_p distribution in the upper camber of the aircraft.

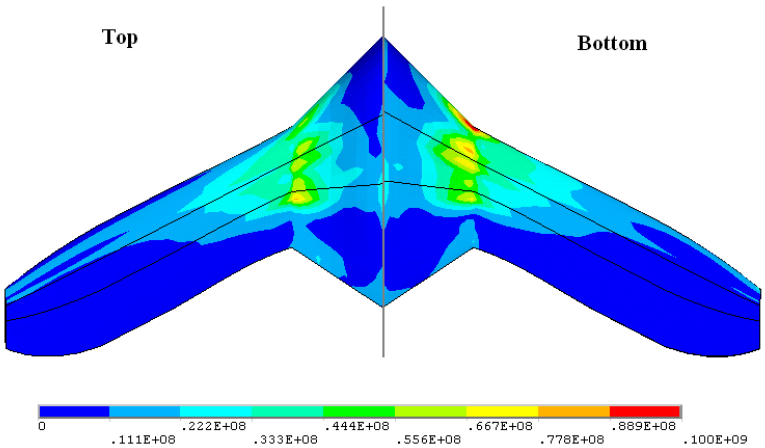

Figure 8.7: Stress intensity (top and bottom views, left and right, respectively; in Pa).

miser's ability to create an structure that is adequate to the aerodynamic loadings.

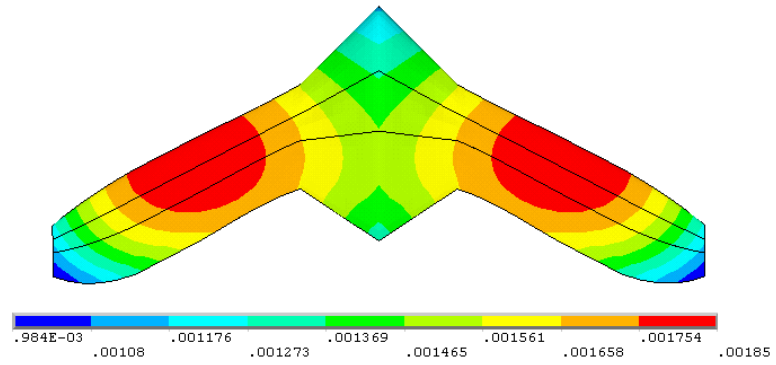

Figure 8.8: Shell thickness distribution (in m).

Solving this problem showed that the application is capable of handling a multidisciplinary design optimisation problem, demonstrating the advantages of this methodology applied to preliminary aircraft design. The modularity approach also proved useful as changing the model's complexity or the analysis tools has no implications in the rest of the application.

As already referred in 7.3, the tests described in chapter 7 proved extremely useful in understanding the issues related to the construction of proper AOFs in the context of multiobjective optimisation, as the results shown in this chapter prove that the tool is able to fulfill all the requirements that are prescribed for the problem.

Chapter 9

Conclusions

Aircraft design is an area in which MDO can offer clear advantages by exploring the interactions between all involved disciplines and taking those into account from the very beginning of the whole design process.

In this work, a novel MDO tool was developed.

A suitable optimisation algorithm was investigated and developed, being the Particle Swarm Optimiser the chosen one. This proved to be a suitable method, particularly for its robustness and noise insensitivity. It should be noted that no comparison with a more traditional, gradient based algorithm, was performed. It did not become clear in this work whether a Particle Swarm Optimiser (tuned or not) is faster than a deterministic counterpart or not (some studies suggest it would actually be slower). However, the Particle Swarm Optimiser has the advantage of not having to deal with derivatives, which can often be confusing, as they may not have a true physical meaning, particularly in the context of aircraft design and for a large number of design variables. Furthermore, any optimisation algorithm that is population based is particularly suited to distributed computation, which is becoming a common reality. In this sense, and if enough computational power is available, the Particle Swarm Optimiser may become a competing alternative to a gradient based algorithm.

Choosing a 3D Panel Method as the aerodynamic solver was based on the compromise between solution quality and computational cost, being that it is not the best possible aerodynamic solver available. However, given that the chosen code (*CMARC*) has been validated and only used within the domain of applicability of the method (incompressible regime), one can assume that the aerodynamic results are reliable. Using an already developed code presented some challenges, regarding the automated interaction with the optimiser, but these were overcome.

Using Finite Element Method for the structural analysis guaranteed the quality of the obtained solution, in the sense that not only is the method widespread and well accepted but FEM also allowed to represent with good fidelity the typical aircraft structural design.

Naturally, for simplicity reasons, the structural finite element model was reduced to its main components (skin panels and framed reinforcements). The finite element model was derived from the output file of the aerodynamic solver, guaranteeing the best possible compatibility between aerodynamic and structural models.

Not performing a numerical convergence study in both aerodynamic and structural analyses, in order to guarantee mesh refinement independence, could lead to numerical errors in the solutions. Even bearing this in mind, the level of refinement of the discretization for both analyses was chosen based on the expected time to analyse a given solution. It must be noted that the main goal of the work described in this document was to develop a framework based on the MDO methodology. Therefore, evaluating the quality of the developed application should not be invalidated by the quality of the analysis tools, as long as they allow a good approach to reality (even if not as good as possible, within the developed theories for solving both structural and aerodynamic problems).

Analysing the obtained results, one can conclude that the optimiser tool is able to do what it is expected to: find the minima of the prescribed cost functions and therefore reach an optimal solution for the problems at hand. The developed application proved to be very flexible, in the sense that it is not limited only to aircraft design, but, with the adequate models and analysis tools, can be applied to any multidisciplinary problem in the engineering field.

It should be noted that the results that are presented for the MDO problem may appear optimistic (in terms of range). A proper static and dynamic stability study, which would have been indispensable to include in a proper preliminary design tool, as well as a number of other disciplines, was left out of the analysis, leading to a rather simplistic model and therefore, to results that might not be feasible for a detailed design point.

As for future developments, possibly one of the most interesting concepts that can be applied to this type of application is distributed computation. The use of evolutionary algorithms is particularly suited to this strategy as it can be implemented on a network of computational resources without the need of actually creating a cluster: each analysis at a given time step of the optimisation process can be independently performed in a node of this network, resulting in an execution time for each time step equal to the time needed for the analysis of a single individual.

In order to use this application in a real life situation, where solution quality is essential, highly detailed models (in the various disciplines to be integrated in the process) should always be used and analysis tools should be the best available. Ideally, aerodynamic analyses should be performed by generating a solid model of the solution and, using CFD methods, evaluating the solution with a highly refined mesh and in a number of situations large enough to cover the whole flight envelope. Along with the aerodynamic solution, a highly detailed structural model should be generated, based on the typical air-

craft structural elements, and the coupled aero-structural analysis performed. Obviously, to be able to do this detailed analysis a preliminary solution should be produced, and that is the primary task of the present application.

Disciplines other than the more traditional structures, aerodynamics and flight performance should also be included: propulsion, aeroelasticity, active control of surfaces, environmental performance (fuel consumption, noise, nocive emissions and such, increasingly important aspects) and operational cost, just to name a few disciplines that matter in the life cycle analysis of an aircraft.

As for the Artificial Neural Network, some integration issues arose that did not allow to present here useful results. The lack of knowledge regarding size and complexity of the network means further investigation is needed in this field. However, once fully functional, this should prove to be a useful tool. Determining the Pareto Front is an important feature, as this truly enables the application to be a Multiobjective Multidisciplinary design tool and gives designers the ability to understand the possible trade-offs that can be done along this surface and therefore chose a suitable optimal design point. Furthermore, doing this with the ANN should allow to approximate this surface in a short time, if compared to obtaining the exact Pareto Front. Integrating the ANN into these applications should also allow a reduction in the computational cost of the solution, as solutions far from optimality are not analysed.

Bibliography

[1] AIAA Technical Committee on MDO. White Paper on Current State of the Art. *http://endo.sandia.gov/AIAA_MDOTC/sponsored/aiaa_paper.html*, January 1991.

[2] Joseph P. Giesing and Jean-François M. Barthelemy. A Summary of Industry MDO Applications and Needs. *At request of the AIAA Technical Committee on MDO*, 1998.

[3] T. H. G. Megson. *Aircraft Structures for Engineering Students.* Butterworth-Heinemann, 2003.

[4] *Ansys*® Inc. *Element Reference Library.* Release 11.0 Documentation for ANSYS.

[5] J. Sobieszczanski-Sobieski, J.-F. M. Barthelemy, and G. L. Giles. Aerospace Engineering Design by Systematic Decomposition and Multilevel Optimization. *14-th Congress of the International Council of the Aeronautical Sciences (ICAS)*, September 1984.

[6] S. Wakayama and I. Kroo. The Challenge and Promise of Blended-Wing-Body Optimization. *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, September 1998.

[7] H. Hoenlinger, J. Krammer, and M. Stettner. MDO Technology Needs in Aeroservoelastic Structural Design. *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, September 1998.

[8] M. H. Love. Multidisciplinary Design Practices from the F-16 Agile Falcon. *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, September 1998.

[9] J. Bennett, P. Fenyes, W. Haering, and M. Neal. Issues in Industrial Multidisciplinary optimization. *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, September 1998.

[10] N. Radovcich and D. Layton. The F-22 Structural Aeroelastic Design Process with MDO Examples. *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, September.

[11] N. M. Alexandrov and R. M. Lewis. Analytical and Computational Properties of Distributed Approaches to MDO. *8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, September 2000.

[12] P. Bartholomew. The Role of MDO within Aerospace Design and Progress Towards an MDO Capability Through European Collaboration. *7th AIAA/USAF/-NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, September 1998.

[13] M. H. Love. Multidisciplinary Design Practices from the F-16 Agile Falcon. *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, September 1998.

[14] J.A. Young, R.D. Anderson, and R.N. Yurkovitch. A Description of the F/A-18E/F Design and Design Process. *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, September 1998.

[15] Y. Deremaux, N. Pietremont, J. Négrier, E. Herbin, and M. Ravachol. Environmental MDO and Uncertainty Hybrid Approach Applied to a Supersonic Business Jet. *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, September 2008.

[16] R.D. Braun, A.A. Moore, and I.M. Kroo. Use of the Collaborative Optimization Architecture for Launch Vehicle Design. *AIAA*, 1996.

[17] Tiago M. Soares. *Optimização Multidisciplinar Aplicada à Concepção de Veículos de Transporte Espacial (MSc. Thesis)*. IST, Lisboa, 2007.

[18] J. Korte, J. Dunn, A. Salas, N. Alexandrov, W. Follett, G. Orient, and A. Hadid. Multidisciplinary Approach to Linear Aerospike Nozzle Optimization. *33rd Joint Propulsion Conference*, July 1997.

[19] Ricardo M. Paiva. *Development of a Modular MDO Framework for Preliminary Wing Design (MSc. Thesis)*. IST, Lisboa, 2007.

[20] G. Venter and J. Sobieszczanski-Sobieski. Multidisciplinary Optimization of a Transport Aircraft Wing Using Particle Swarm Optimization. *9th AIAA/USAF/-NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, September 2002.

[21] J. Kennedy and R. Eberhart. Particle Swarm Optimization. *IEEE International Conference on Neural Networks, Vol. IV, pp. 1942–1948*, 1995.

[22] Ruben E. Perez and Peter W. Jansen. Aero-Structural Optimization of Non-Planar Lifting Surface Configurations. *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, September 2008.

[23] D. Lim, Y.-S. Ong, Y. Jin, B. Sendhoff, and B. S. Lee. Inverse multi-objective robust evolutionary design. *Springer Science & Business Media*, September 2006.

[24] Y. Jin and J. Branke. Evolutionary Optimization in Uncertain Environments – A Survey. *IEEE Transactions on Evolutionary Computation, Vol. 9, No. 3*, June.

[25] George Marsaglia. Random Number Generators. *Journal of Modern Applied Statistical Methods, Vol. 2, No. 1, pp. 2-13*, May 2003.

[26] G. Cybenko. Approximations by superpositions of sigmoidal functions. *Mathematics of Control, Signals, and Systems, Vol. 2, pp. 303-314*, 1989.

[27] Kurt Hornik. Approximation Capabilities of Multilayer Feedforward Networks. *Neural Networks, Vol. 4*, 1991.

[28] D. Chafekar, L. Shi, K. Rasheed, and J. Xuan. Multiobjective GA Optimization Using Reduced Models. *IEEE Transactions on Systems, Man and Cybernetics – Part C: Applications and Reviews, Vol. 35, N. 2*, May.

[29] G. Li, M. Li, S. Azarm, S. al Hashimi, T. al Ameri, and N. al Qasas. Improving multi-objective genetic algorithms with adaptive design of experiments and online metamodeling. *Structural and Multidisciplinary Optimization*, February.

[30] Z. Zhou, Ong. Y.-S., P. B. Nair, A. J. Keane, and K. Y. Lum. Combining Global and Local Surrogate Models to Accelerate Evolutionary Optimization. *IEEE Transactions on Systems, Man and Cybernetics – Part C: Applications and Reviews, Vol. 35, N. 2*, January.

[31] Y. Jin and J. Branke. Evolutionary Optimization in Uncertain Environments – A Survey. *IEEE Transactions on Evolutionary Computation, Vol. 10, No. 1.*

[32] Y. Jin. A comprehensive survey of fitness approximation in evolutionary computation. *Soft Computing, Vol. 9*, 2005.

[33] A. Messac, A. Ismail-Yahaya, and C.A. Mattson. The Normalized Normal Constraint Method for Generating the Pareto Frontier. *Structural and Multidisciplinary Optimization, Vol. 25, No. 2, pp. 86-98*, 2003.

[34] Paulo Caixeta Jr. and Flávio Marques. Aeroelastic Wing MDO Using Metamodel Based on Neural Networks. *8th World Congress on Structural and Multidisciplinary Optimization*, 2009.

[35] Ricardo Paiva, Curran Crawford, Afzal Suleman, and André Carvalho. A Comparison of Surrogate Models in the Framework of an MDO Tool for Wing Design. *5th AIAA Multidisciplinary Design Optimization Specialist Conference*, May 2009.

[36] John. D. Anderson. *Computation Fluid Dynamics – Basics with Applications.* McGraw Hill, 1995.

[37] C. Hirsch. *Numerical Computation of Internal and External Flows, 2nd Edition.* Butterworth-Heinemann, 2007.

[38] Vasco de Brederode. *Fundamentos de Aerodinâmica Incompressível.* Vasco de Brederode, Lisboa, 1997.

[39] Joseph Katz and Allen Plotkin. *Low Speed Aerodynamics - From Wing Theory to Panel Methods.* McGraw-Hill, 1991.

[40] Dale L. Ashby, Michael Dudley, and Steven K. Iguchi. Development and Validation of an Advanced Low-Order Panel Method. *NASA Technical Memorandum 101024*, 1988.

[41] Dale L. Ashby and Lindsey E. Browne. Study of the Integration of Wind Tunnel and Computational Methods for Aerodynamic Configurations. *NASA Technical Memorandum 102196*, 1989.

[42] Farooq Saeed and Jerome Bettinger. Aerodynamic Efficiency of Non-Planar Arc/Elliptic Planforms. *28th Annual ARA Congress*, 2003.

[43] David W. Roberts. *The Aerodynamic Analysis and Aeroelastic Tailoring of a Forward-Swept Wing, (MSc. Thesis).* North Carolina State University, Raleigh, NC, 2006.

[44] J. N. Reddy. *An Introduction to the Finite Element Method.* McGraw-Hill, 2005.

[45] ASM International. *Metals Handbook, Vol.2 - Properties and Selection: Nonferrous Alloys and Special-Purpose Materials, 10th Ed.* ASM Intl., 1990.