

1-1-2011

Preliminary Sailplane Design Using MDO And Multi-Fidelity Analysis

Chris V. Pilcher
Ryerson University

Follow this and additional works at: <http://digitalcommons.ryerson.ca/dissertations>



Part of the [Aerospace Engineering Commons](#)

Recommended Citation

Pilcher, Chris V., "Preliminary Sailplane Design Using MDO And Multi-Fidelity Analysis" (2011). *Theses and dissertations*. Paper 1614.

This Thesis is brought to you for free and open access by Digital Commons @ Ryerson. It has been accepted for inclusion in Theses and dissertations by an authorized administrator of Digital Commons @ Ryerson. For more information, please contact bcameron@ryerson.ca.

PRELIMINARY SAILPLANE DESIGN USING MDO AND MULTI-FIDELITY ANALYSIS

by

Chris V. Pilcher
B.Eng. (Aerospace Engineering)
Ryerson University (Canada), 2011

A thesis
presented to Ryerson University
in partial fulfilment of the
requirement for the degree of
Master of Applied Science
in the Program of
Aerospace Engineering.

Toronto, Ontario, Canada, 2011

© Chris V. Pilcher, 2011

Declaration

I hereby declare that I am the sole author of this thesis.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

Signature

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Signature

Abstract
**PRELIMINARY SAILPLANE DESIGN USING MDO AND MULTI-
FIDELITY ANALYSIS**

Chris V. Pilcher
Master of Applied Science
Graduate Department of Aerospace Engineering
Ryerson University
2011

A multidisciplinary design optimization (MDO) strategy for the preliminary design of a sailplane has been developed. The proposed approach applies MDO techniques and multi-fidelity analysis methods which have seen successful use in many aerospace design applications. A customized genetic algorithm (GA) was developed to control the sailplane optimization that included aerodynamics/stability, structures/weights and balance and, performance/airworthiness disciplinary analysis modules. An adaptive meshing routine was developed to allow for accurate modeling of the aero-structural coupling involved in wing design, which included a finite element method (FEM) structural solver along with a vortex lattice aerodynamics solver. Empirical equations were used to evaluate basic sailplane performance and airworthiness requirements. This research yielded an optimum design that correlated well with an existing high performance sailplane. The results of this thesis suggest that preliminary sailplane design is a well suited application for modern optimization techniques when coupled with, multi-fidelity analysis methods.

Acknowledgements

I would like to thank Dr. Joon Chung for his guidance and support.

And my family and friends for their support and patience.

Contents

Declaration.....	ii
Abstract.....	iii
Acknowledgements.....	iv
List of Figures	vii
List of Tables	ix
Nomenclature	x
Chapter 1.....	1
Introduction	1
1.1 Research Motivation & Objective	2
1.2 Literature Review	2
Chapter 2.....	8
Design Optimization.....	8
2.1 Optimization Algorithms.....	9
2.2 Multidisciplinary Design Optimization	11
Chapter 3.....	14
Analysis Modules and Functions.....	14
3.1 Custom Genetic Algorithm.....	14
3.2 Variable and Parameter Updating	22
3.3 Aerodynamics and Stability Analysis.....	23
3.4 Structures & Weights and Balance	27
3.5 Performance & Airworthiness	36
Chapter 4.....	39
Results.....	39
4.1 Problem Definition	39
4.2 Optimization Results	42
Chapter 5.....	49
Conclusion.....	49
5.1 Future Work	50
Bibliography	52
Appendix A.....	56

A.1	Genetic Algorithm Selection Method Case Study	56
A.2	Modified Gaussian Mutation Function	58
A.3	Structures Model Convergence Study	59
A.4	Stringer Description	60
A.5	ASW 27 B Information Sheet	61

List of Figures

Figure 1.1 Lilienthal Glider [7].....	3
Figure 1.2 Vampyr Glider [7].....	4
Figure 1.3 Ka 6 (left) Phoenix (right) [7].....	5
Figure 1.4 eta [7]	5
Figure 2.1 Genetic Algorithm Block Diagram.....	10
Figure 2.2 Multi-disciplinary Analysis package	12
Figure 2.3 Multi-Disciplinary Feasible Method	13
Figure 3.1 Roulette and Stochastic Uniform Selection Methods.....	16
Figure 3.2 Rosenbrock function	19
Figure 3.3 Rosenbrock Function MATLAB Vs Custom GA Results Comparison	21
Figure 3.4 Structured Variables System	22
Figure 3.5 AVL Aerodynamic Model	25
Figure 3.6 Normalized AVL Pressure Distribution.....	26
Figure 3.7 ANSYS Flow Diagram	27
Figure 3.8 ANSYS Element Descriptions [39]	29
Figure 3.9 Key Point and Rib Configuration	29
Figure 3.10 Test ANSYS Wing box Mesh	31
Figure 3.11 Average Element Aspect Ratio VS Element Edge length	32
Figure 3.12 Aerodynamic Force Mapping.....	33
Figure 3.13 ANSYS Model with Forces Applied.....	35
Figure 3.14 Stages of ANSYS Wing box Creation and Analysis	36
Figure 3.15 Idealized cross-country flight profile [44]	38
Figure 4.1 Diagram showing many of the design variables	39
Figure 4.2 Penalized and non Penalized Objective Function Convergence History Including Top Views for Optimum Designs.....	43
Figure 4.3 Sailplane layout comparison between the ASW 27 B [45] and the optimized design.....	46
Figure 4.4 Von Mises Stress Distribution of Optimized Design (Top View)	47
Figure 4.5 Von Mises Stress Distribution of Optimized Design (Bottom View)	48
Figure A.1 Comparison of Stochastic and Roulette Selection Methods	57
Figure A.2 Comparison of Tournament and Roulette Selection Methods	57

Figure A.3 Pool Size Comparison For The Tournament Selection Method.....	58
Figure A.4 Mesh Convergence Test Model	59
Figure A.5 Mesh Convergence Results.....	60
Figure A.6 Z Stringer.....	61
Figure A.7 ASW 27 B Data Sheet [47]	62

List of Tables

Table 3.1 Genetic Algorithm Test Case Settings	20
Table 3.2 Rosenbrock Function Optimization Results for MATLAB GA	21
Table 3.3 Rosenbrock Function Optimization Results for Custom GA.....	21
Table 3.4 AVL Case File Input Parameters	26
Table 3.5 Material Properties [38]	28
Table 3.6 Average Element Aspect Ratios for Tapered Sections	31
Table 4.1 Design Variables and Constraints.....	40
Table 4.2 AVL Initial Case Parameters	41
Table 4.3 Genetic Algorithm Options for Sailplane Optimization.....	41
Table 4.4 Design Variable and Constraint Values for Selected Designs.....	44
Table 4.5 Sailplane Data comparison.....	46

Nomenclature

Abbreviations

AVL	Athena Vortex Lattice program
BF	Bottom Flange
CFD	Computational Fluid Dynamics
CFRP	Carbon Fibre Reinforced Polymer
CG	Center of gravity
DOF	Degree of Freedom
FEA	Finite Element Analysis
FEM	Finite Element Method
FS	Front Spar
GA	Genetic Algorithm
LE	Leading Edge
TE	Trailing Edge
MDA	Multi-Disciplinary Analysis
MDF	Multi-Discipline Feasible
MDO	Multidisciplinary-Design-Optimization
NACA	National Advisory Committee for Aeronautics
RS	Rear Spar
SQP	Sequential Quadratic Programming
TF	Top Flange

Greek Symbols

α	Sailplane lift curve slope
----------	----------------------------

δ	Step size
δ_{Atrim}	Aileron trim angle
δ_{Rtrim}	Rudder trim angle
δ_{Etrim}	Elevator trim angle
Δl	Line length
Δp	Pressure differential
Δx_e	AVL element x dimension
Δy_s	AVL element strip width
Λ	Element line divisions
λ	Taper ratio
μ	Non dimensional mass ratio
ρ	Air density
σ_{max}	Maximum Von Mises stress
ϕ	Bank angle

Other Symbols

A	Element area
AR_h	Horizontal stabilizer aspect ratio
AR_{Wing}	Wing aspect ratio
b	Wing span
b_v	Vertical stabilizer span
$\frac{1}{B}$	Prandtl-Glauert factor
c	Chord length
C_{Dmin}	Minimum drag coefficient
Cg_{fuse}	Fuselage center of gravity
C_L	Lift coefficient

C_{LMax}	Maximum lift coefficient
C_m	Pitching moment coefficient
C_p	Pressure coefficient
d	Thickness decay factor
\bar{e}	Average element size
e_0	Initial element size
f	Fitness function
\bar{f}	Normalized fitness function
f_0	Non-penalized fitness value
f_s	Stringer flange length
g	Constraint function
g	Acceleration due to gravity
h_s	Stringer height
I_{xx}	Area moment of inertia about the x-axis
I_{yy}	Area moment of inertia about the y-axis
k	Gust alleviation factor
K	Static margin
L/D	Lift VS drag ratio
L_β	Dihedral effect derivative
L_e	Element lift force
\tilde{L}_{FS}	Lift force on front spar
\tilde{L}_{RS}	Lift force on rear spar
l_m	Average geometric wing chord length
L_r	Roll moment due to yaw rate derivative
m	Sailplane gross mass

M	Mach number
m_{wing}	Wing mass
n_1	Limit load factor for utility class sailplanes
N_β	Directional stability derivative
n_g	Gust load factor
n_{pop}	Number of individuals in a population
n_{par}	Number of parents
N_r	Yaw rate damping derivative
P	Penalty function
p_∞	Ambient air pressure
q	Dynamic pressure
r	Radius from center of thermal
S	Wing planform area
S_v	Vertical stabilizer area
t_{BF}	Thickness of bottom flange
t_{FS}	Thickness of front spar
t_0	Wing box component root thickness
t_{orib}	Wing box root rib thickness
t_{rib}	Wing box rib thickness
t_{RS}	Thickness of rear spar
t_s	Stringer thickness
t_{TF}	Thickness of top flange
U	Gust velocity
V	Airspeed
V_A	Manoeuvre speed

V_{Avg}	Average cross country speed
V_C	Rate of climb in a thermal
V_D	Dive speed
V_S	Stall speed
V_{S1}	Estimated stall speed at maximum design weight with flaps and air breaks retracted
V_{Sg}	Gliding sink rate
V_{St}	Rate of sink in a thermal
V_T	Upward thermal velocity
W	Weight
w_{deflect}	Wingtip deflection
WL_h	Winglet height
WL_λ	Winglet taper ratio
WL_s	Winglet sweep
x	Local variable vector
x_{FS}	X position of front spar
x_{ht}	Horizontal tail apex location
x_{RS}	X position of rear spar
y	Coupling variable vector
z	Global variable vector

Chapter 1

Introduction

Aircraft design is a complex process involving many disciplines and design criteria. The traditional design approach uses teams of experts from each discipline to collaboratively develop a design which meets the required goals [1]. Typically designers rely on years of past experience as well as empirical or statistical methods developed from previous projects. This process has been successful in many aircraft designs however; it can be a time consuming and costly endeavour. Multidisciplinary-design-optimization (MDO) has been successfully shown as an alternative design approach when dealing with extensive disciplinary coupled and complex problems [2] [3]. Sailplane design is an example of a complex design process despite the absence of a propulsion system. Instead they rely on a highly efficient design to maintain flight. Sailplane efficiency is usually based on the ability of a sailplane to glide as long and as far as possible [4]. These two design goals are most influenced by the aerodynamic efficiency and the weight of the sailplane and therefore a sailplane with a very high lift VS drag ratio and a low mass is desired. Consequently the most influential disciplines in sailplane design become the aerodynamic and structures disciplines. The strong dependence on both of these disciplines presents a challenge as they are also highly coupled and often exhibit conflicting impacts on a design. This can lead to problems when conducting disciplinary analysis using low fidelity methods which, are often defined by relatively few variables [5]. Since low fidelity conceptual design approaches are defined by only a few key quantities, it becomes increasingly difficult to model highly coupled disciplines that themselves are linked by many variables. For this reason higher fidelity analysis methods are adopted into MDO problems to better handle disciplinary coupling [6]. However, the use of high fidelity analysis methods can significantly increase the computational requirements and convergence time. Thus it is beneficial to utilize higher fidelity methods in disciplines that have a sensitive impact on the objective function of the optimization problem, while sticking to lower fidelity methods in less influential disciplines. Aside from analysis approaches, another consideration in the formulation of an MDO problem is the choice of a suitable optimization method. Several methods exist each with their own merits, but their effectiveness depends on the type of problem. Two main classes of optimizers are deterministic and stochastic methods which are further explained in Chapter 2.

1.1 Research Motivation & Objective

Though modern optimization techniques have been used in several aerospace applications they have not been widely used for sailplane preliminary design. As will be seen later in the literature review section of this thesis, the majority of work considering sailplane design optimization has been focused on low fidelity conceptual designs, or more detailed work on specific components of sailplane design. Therefore the objective of this work was to apply modern MDO along with multi-fidelity analysis methods to a preliminary sailplane design case. This work aims to demonstrate the validity of modern MDO techniques and multi-fidelity contributing analysis methods for preliminary sailplane design by achieving optimum designs with comparable performance to existing high performance sailplanes.

1.2 Literature Review

This section will first give a brief history on the advancements of sailplane design and describe some of the designs that have helped create the modern sailplane. This will be followed by a summary of recent work in aircraft and sailplane design optimization.

In the late 19th century the first successful gliding flights were performed by Otto Lilienthal. Lilienthal developed a glider from which the pilot hung and was able to launch from hill tops and glide several meters seen in Figure 1.1. The glider was an extremely light design which used external bracing to support a very thin wing. This design provided a slow forward speed and a minimum sink rate of about 1 m/s with a maximum lift VS drag ratio of approximately 6 [7].

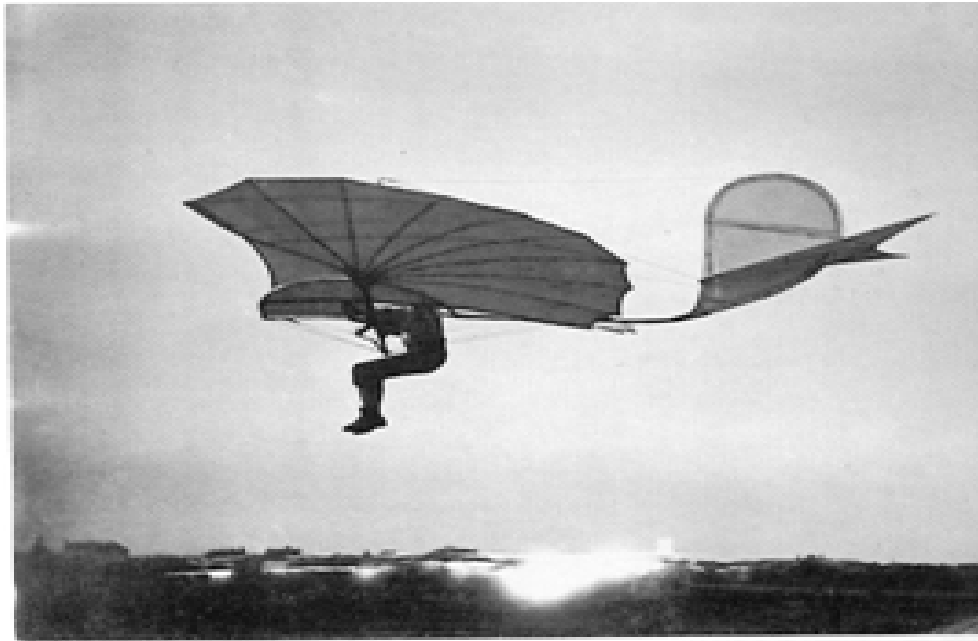


Figure 1.1 Lilienthal Glider [7]

The next fundamental advancement in gliding flight came with the preliminary testing conducted by the Wright brothers in the early 1900's. The Wright brothers developed the concept of control surfaces on early non powered prototypes of their famous Wright flyer. This was the first time that an aircraft was controlled by changing the geometry of the sailplane rather than shifting the center of gravity as done with hang gliders. Since the Wright brothers also started the movement of powered flight around this time sailplanes were not focused on for over two decades until the 1920s. It was in this time that post First World War Germany began developing sailplanes after they were forbidden to develop powered aircraft as a result of the Treaty of Versailles. It was during this period that aerodynamics was identified as the main influence over sailplane performance. Additionally, the concept of thermal soaring was explored when it was found that sailplanes could benefit from rising air currents. The *Vampyr*, seen in Figure 1.2 was a design from this era that started to push sailplane flight endurance from minutes into hours.

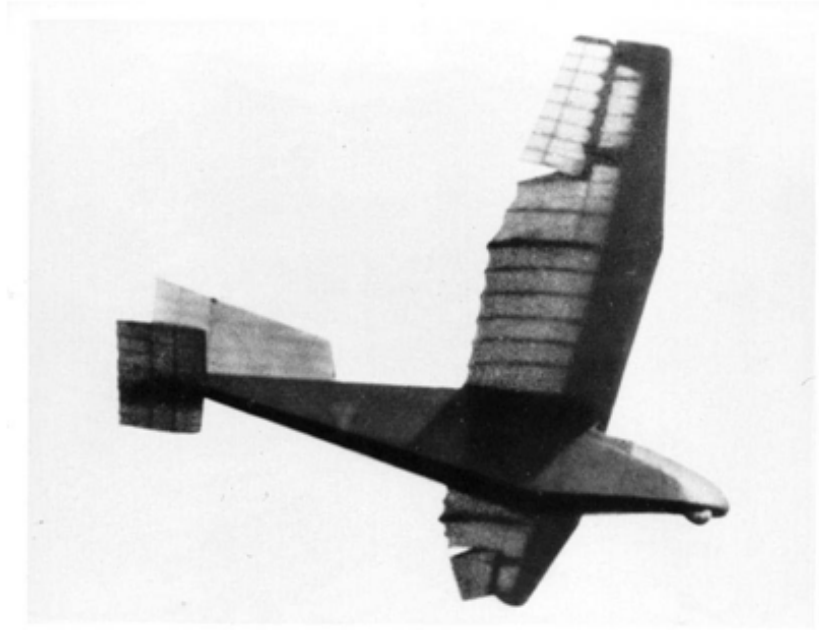


Figure 1.2 Vampyr Glider [7]

Further progress in sailplane development was sparked by annual sailplane competitions in the Rhön Mountains that by the outbreak of the Second World War saw sailplane ranges approaching 500km and their glide ratios nearing 30 [7]. Once again sailplane development slowed during the Second World War, but the huge leap in aerospace technology that came from the war showed great potential in sailplane development. The introduction of laminar flow airfoils such as the NACA-6 series fuelled a movement towards smooth surfaced construction materials in sailplane design. In the late 50's and early 60's the *Ka 6* seen on the left in Figure 1.3, was the most successful sailplane design which, was still primarily wooden but was able to partially take advantage of laminar flow. The next big leap in sailplane design came with the introduction of the Eppler airfoil sections. These were developed specially for sailplanes and could benefit from the advancements in fibreglass construction also occurring at that time. It became possible to construct strong lightweight structures with sufficiently smooth lifting surfaces that could take advantage of laminar airfoil sections. The *Phoenix* sailplane on the right of Figure 1.3 was the first to take advantage of these advancements and, in 1957 and was able to achieve a glide ratio of 40:1 [8]. Furthermore, in the 1950's sailplane classes were introduced to handle the wide variety of designs used in competitions. The classes included: an open class, 15 m wingspan class, 18 m wingspan class and a standard class. The standard class is essentially the same as the 15 meter wingspan class however, lift enhancing devices are prohibited [9].

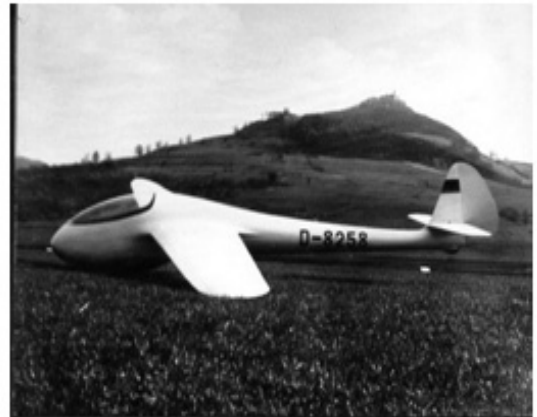


Figure 1.3 Ka 6 (left) Phoenix (right) [7]

By the early 70s carbon fibre composite materials were entering into sailplane designs, the first being the *SB-10*. It was this advance in materials that allowed the massive 29 m wingspan *SB-10* to be the first sailplane to reach a glide ratio over 50 [8]. The most modern advances in sailplane design have come with the development of composite materials allowing for larger wingspans and aspect ratios. The highest performance sailplane to date is the *eta* seen in Figure 1.4 which has a wingspan of 31 meters and an aspect ratio of 51 allowing it to reach a maximum glide ratio of 70 [7].



Figure 1.4 eta [7]

There have been many successful applications of design optimization techniques to aircraft design problems. These applications cover a wide range of approaches depending on the optimization strategies and the fidelity of the analysis methods. The levels of fidelity related to optimization

problems have been defined for structural and fluid analysis by Bartholomew [10]. Bartholomew divides the fidelity of an analysis tool into 3 levels. The first level includes low fidelity analysis based on empirical equations. In the second level, basic structural beam theory, and vortex panel based aerodynamics solvers are covered. Finally the third level encompasses high fidelity finite element analysis and computational fluid dynamics methods for structures and aerodynamic analysis respectively. The majority of aircraft design optimization applications have been regarding conceptual design optimization which use low and sometimes intermediate fidelity analysis such as the work by Neufeld et al. [11]. In this work, a multidisciplinary design optimization for the conceptual design of regional commercial aircraft considering uncertain contributing analysis was conducted. The results of this work showed that in some cases, low fidelity analysis methods can introduce a measure of uncertainty into the optimization results. It was found that the optimum designs seemed feasible until subjected to more detailed analysis. This doesn't mean that low fidelity analysis methods provide inaccurate results; it has instead been described by Giesign and Barthelemy as a lack of assurance that the results properly simulate reality [12].

In the last decade the increases in computational power have driven the use of higher fidelity analysis tools in design optimization [12]. It is now much more practical to embed some forms of higher fidelity analysis within an optimization scheme as done in references [5], [13], and [14]. In these three works, aircraft wing optimizations were conducted using combinations of low, intermediate and high fidelity analysis methods. It is clear from their results that complex tasks such as wing design may require some higher fidelity analysis tools to provide reliable and practical results. This is most prevalent for the aerodynamic and structural coupling involved in wing design which often requires more detail than low fidelity methods can provide. Furthermore, if the designer wishes to aim for a preliminary design optimization, higher fidelity methods must be considered.

With regards to optimizer choice it was found that genetic algorithms (GA) and sequential quadratic programming (SQP) approaches have both been successfully applied to aircraft design optimizations. It was discovered that gradient based strategies such as (SQP) were preferred in some cases [5], [15] as they tend to provide quick solutions if the problem is smooth enough for a gradient optimizer to operate effectively. However, GAs have also proven to be an efficient technique in several previous works [1], [16], [17] and [18] where a more robust strategy was required.

Research revealed that sailplane optimizations have been conducted in the past; although, they have mainly been focused on basic conceptual designs or on specific components such as aero/structural

wing design. In the late 1980s a low fidelity simultaneous optimization for sailplane design and flight trajectory was conducted by Kawamura and Suzuki [19]. This work concluded that sailplane design, as with many complex systems, involves extremely coupled design criteria that cause significant tradeoffs depending on the optimization objective. Also in the late 80s, an integrated aero/structural sailplane wing design was conducted by Grossman et al. [20] in which low and moderate fidelity analysis methods were used to model the aero/structural interactions of a sailplane wing for a given flight mission. In their approach, lifting line theory and structural beam modeling methods were used to determine the deformed wing planform characteristics and their effect on the aerodynamic performance of the wing. This work further demonstrated the need for integrated or MDO analysis in sailplane design. With the growing use of modern analysis techniques, a few more sailplane design problems have been addressed in literature. The optimization of sailplane winglets and sailplane wing/fuselage combinations were conducted by Maughmer and Boermans et al. respectively [21], [22]. These two papers further demonstrated the importance of an efficient aerodynamic design for an unpowered aircraft.

Chapter 2

Design Optimization

The term optimization has been described as a process in which the inputs to a problem are adjusted to determine the minimum or maximum output [23]. The process of optimization is a very old aspect of mathematics dating back to the advent of calculus at which time it was used to determine the extreme values of simple functions [5]. The more modern technique called design optimization uses computational methods to determine the best possible design for a given objective or set of objectives in the most effective and efficient way possible. The objective function of a problem is represented as a mathematical function that is then minimized yielding the optimum result based on contributing design variables and analysis. However, when searching for practical optimum designs, constraints are often required to ensure that the optimum design is feasible. Therefore, constraint functions are added to the problem to keep the optimum design within set limits. Search boundaries are also added to the design variables to avoid impossible or impractical design variable selections, for example, negative material thicknesses. In the event of a constraint violation a penalty function is set in place to penalize the objective function helping keep the optimization out of infeasible design space. A general formulation for constrained optimization problems is shown in by equations 2.1-2.3 where f_0 represents the non-penalized fitness value and P is the penalty function vector. $g_i(x)$ represents the normalized constraint function vector, g_{actual} is the constraint parameters calculated value and, g_{Limit} is the constraint limit.

$$\text{Minimize } f(x) = f_0 + \sum P \quad (2.1)$$

$$\text{If } g_i(x) \leq 0: P_i = 0 \quad \text{else: } P_i = g_i(x) * f_0 \quad (2.2)$$

$$g_i(x) = \left(\frac{g_{actual}}{g_{Limit}} - 1 \right) \quad (2.3)$$

2.1 Optimization Algorithms

The optimizer of a design optimization problem controls the problem by choosing sets of design variables and monitoring their impact on the objective function of the problem. Several optimization strategies exist and use a variety of techniques for searching for optimum designs. The two predominate methods are deterministic and stochastic and of these two methods the most used are the sequential quadratic programming and genetic algorithm approaches. The SQP optimizer method uses a gradient based strategy to move from a starting point in the direction of steepest slope. Similar to Newton's Method, a sequence of one dimensional optimizations involving quadratic approximation functions representing the objective and constraint functions are solved recursively until an optimum solution is found. A drawback of this method is its reliance on gradient information that is typically calculated using finite differencing when analytical means are not possible. As a result, sufficiently smooth objective and constraint functions are required to ensure accuracy when calculating the gradient information. Furthermore, gradient based optimizers are susceptible to finding local optima as they do not explore the entire solution space [6], [24]. Despite some of the fallbacks of SQP it has been widely and successfully used for optimization problems with non-linear constraints [24]. GA's are optimizers based on evolutionary theory and operate by selecting an initial population of randomly selected design points. This type of optimization scheme tends to be less computationally efficient but is generally more robust when dealing with non-smooth functions [5], [25]. In addition, GA optimizers have shown desirable performance in complex aerospace related optimizations from hovercrafts to unmanned aerial vehicles [16], [17]. For this thesis a sailplane optimization with many variables and a non-smooth objective function was conducted and therefore a GA optimizer was chosen over a gradient based method.

The optimizer used for this work was based on the GA available in the commercial technical computing software package MATLAB. However, the default GA available in MATLAB was not designed for the type of optimization problem conducted for this thesis, and therefore was customized to better manage the problem at hand. Evaluations of the default MATLAB GA identified several issues when applied to the sailplane problem that lead to the development of the custom GA. The main issue with the MATLAB GA was the inability to edit the source codes of the GA. This effectively made some functions of the MATLAB GA black box functions that could not be adjusted to better suit the sailplane problem. Another problem with the default GA was its high computation times. Several test cases were conducted using

the default GA and it was found that a sub optimization was conducted within each generation. This meant that for a population of m designs carried over n generations, $(n \times m) \times n$ iterations were required. This led to significantly longer solution times and had no apparent advantage to the customized GA which looped through the population once for each generation. Finally, the output functions for the default GA were limited and therefore in the customized version a post processing code was added to save all the results of the optimization and automatically create graphical results. The block diagram shown in Figure 2.1 is a general formulation of the GA used in this work.

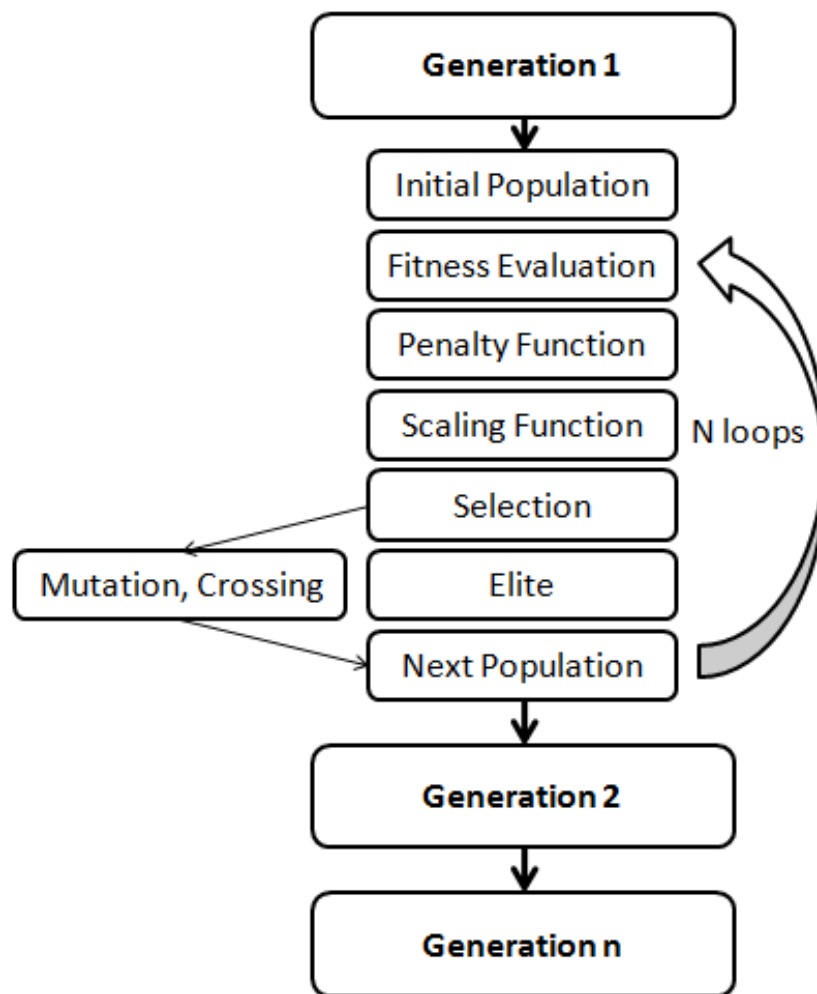


Figure 2.1 Genetic Algorithm Block Diagram

As can be seen in Figure 2.1 a GA mimics the evolution process by subjecting a random initial population to a fitness evaluation to determine the best, or elite designs based on the objective and constraint

functions. Designs that violate constraints are penalized using a penalty function. This used to ensure that poor designs are given poor fitness values and thus less likely to survive in subsequent generations, similar to nature in which the strong survive to pass on their superior genes. The procedure for choosing a subsequent generation of designs is handled through crossover and mutation. Crossover takes the high fitness designs and merges them to form children designs from two or more parents; this is done to mimic the breeding of superior individuals. Mutation is a process in which random changes are made to designs to allow more diversity and search more areas of the design space, giving every point in the design space a chance to be explored [25]. This process is useful as it helps avoid convergence to a local optimum point by pushing some designs away from main search direction. The selection, crossing and mutation processes are then repeated for a specified number of generations until an optimized solution is found. A detailed description of the GA that was used in this work can be found in Chapter 3.

2.2 Multidisciplinary Design Optimization

MDO has been described as a methodology that can be used for the design of complex systems in which coupling between disciplines exists [5]. This makes MDO a valuable tool for aerospace related design tasks as they are generally complex problems incorporating several disciplines and many design criteria. The term disciplinary coupling refers to a condition in which the output of one discipline is required for the input of another. To handle this coupling Multi-Disciplinary Analysis (MDA) packages are developed to pass the required information between disciplines until convergence is achieved. An example showing the MDA package used in this work which includes three disciplines is shown in Figure 2.2.

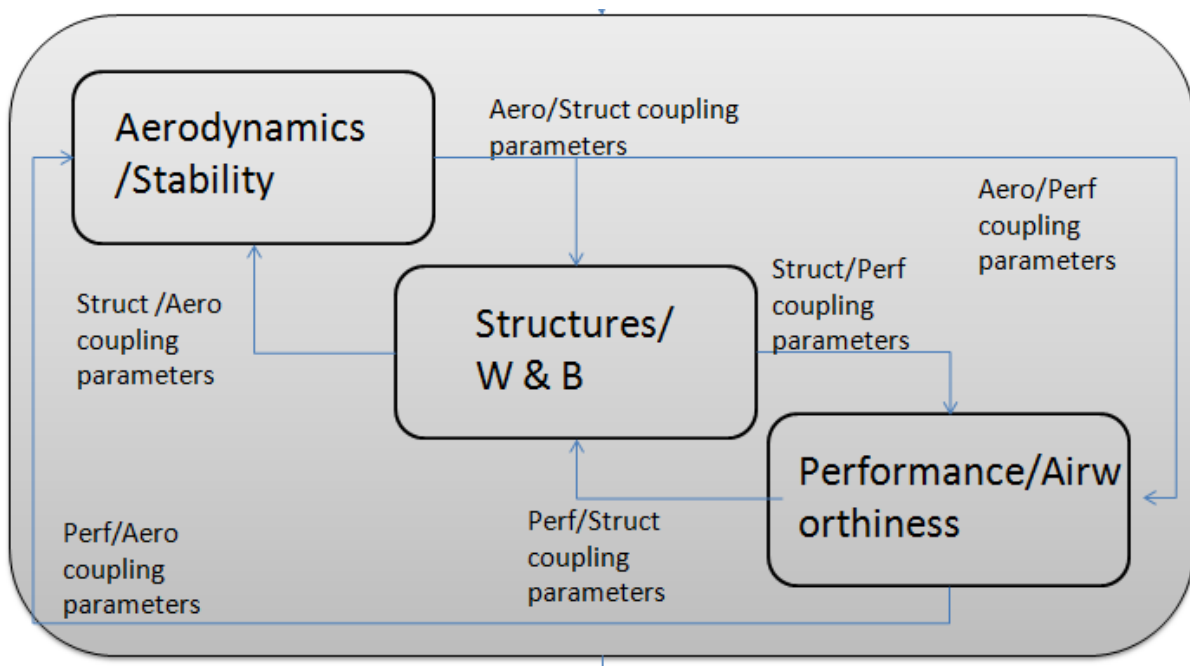


Figure 2.2 Multi-disciplinary Analysis package

Figure 2.2 shows the analysis package for the sailplane optimization in which aerodynamics/stability, structures/Weights and balance and, performance/airworthiness disciplines are considered. The figure also shows how each discipline is connected to the other two by the input and output values. For this reason, MDA packages are organized in a loop that passes the data between the disciplines until the entire analysis package converges within a set tolerance.

Once the optimizer and analysis package have been chosen a suitable optimization strategy can be selected. The MDO structure is responsible for governing the overall optimization process by connecting the optimizer to the MDA package and relaying information between the two. There are several MDO architectures capable of handling coupled optimization problems. The Multi-Discipline Feasible (MDF) method is the simplest and earliest technique. It is also very commonly used for a wide variety of engineering problems. This method was first introduced in 1994 by Cramer et al. [23]. MDF is considered a *black box* approach to an optimization problem in which the optimizer controls only the design variables and the MDA package handles all the analysis disciplines [23]. This method requires the MDA loop to be solved for every change in design variables and often requires several iterations within the loop before convergence between the disciplines is achieved. A block diagram of the MDF method is shown in Figure 2.3. The objective function f and the constraint function g are defined by three types of variables. The global variables denoted as z are used in all the disciplines, the coupling variables are

denoted as y , and the local variables only used in a single discipline are denoted as x . Since this method can require several iterations of the MDA package for each change in the design variables it is considered a more time consuming method than other techniques [26]. However, this more time consuming approach has proven to provide accurate results in many cases [27]. Furthermore, the MDF method has been widely and confidently used in many design optimization applications. Therefore the MDF method was chosen for this work as it served as a reliable and simple platform for evaluating the effectiveness of using multi-fidelity analysis methods to handle heavily coupled disciplinary analysis modules.

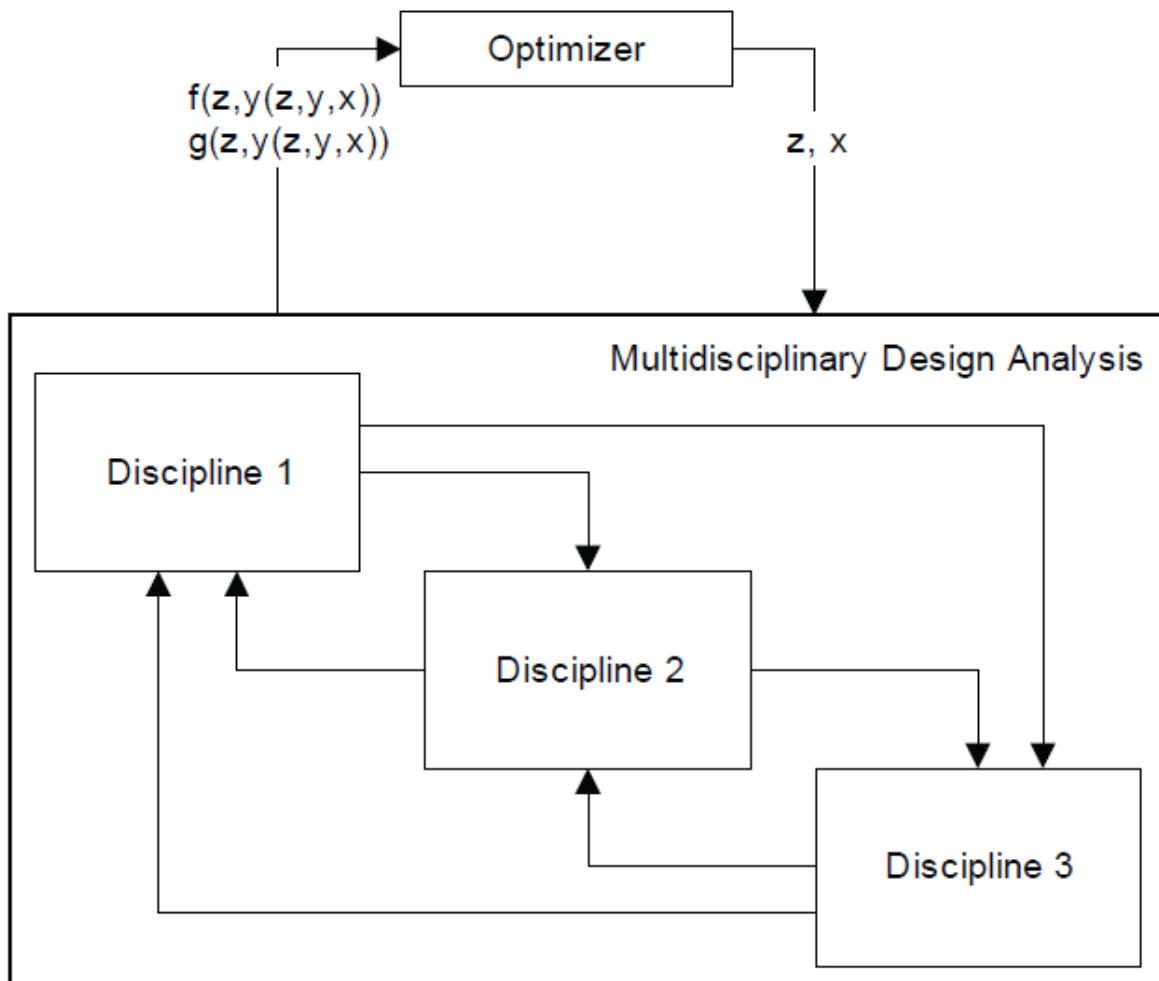


Figure 2.3 Multi-Disciplinary Feasible Method

Chapter 3

Analysis Modules and Functions

This chapter will explain the structure of the MDO framework developed for this thesis by breaking down the various functions and analysis packages. As stated in Chapter 2, a customized version of the MATLAB GA was implemented along with the MDF architecture for the sailplane optimization. Also one of the objectives of this work was to demonstrate the benefits of using multi-fidelity analysis methods within MDO problems. Therefore, a combination of high, moderate and, low fidelity methods were used for the disciplinary analysis modules and will be explained in this section of the thesis.

3.1 Custom Genetic Algorithm

As mentioned in Chapter 2, there were several aspects of the default MATLAB GA that were not desirable for the problem in this thesis. However, some of the sub-functions of the default GA were used in their original form. Therefore, the customized GA was developed to maintain compatibility with some of the existing MATLAB functions while, literature was used as a references to determine the various options to best suit the sailplane optimization [28]. As was shown in Figure 2.2, the GA is made up of several sub-functions that are used to evolve one generation into the next. The first task of a GA is to generate the initial population which can be conducted in different ways. Two common techniques include, a randomly distributed method which takes points randomly across the entire design space or, a Gaussian distribution method that is taken from one or more starting points. The second method was used in this work as it was found to provide a faster solution if a good starting design was chosen. The next step of a GA is to perform a fitness evaluation on the population. Since the sailplane problem was a constrained optimization, the constraint functions were also calculated at this step. The objective function for each design is first calculated in absence of the constraint functions to determine the non-penalized value and then adjusted based on any constraint violations. Research regarding penalty functions was undertaken to determine a suitable method for this problem and several techniques were discovered. The fundamental technique is to eliminate any design that violates a constraint called the “death penalty”. This approach guarantees that only feasible designs survive, yet, its strict approach can

lead to problems. For instance some constraints may be of less importance, and promising designs that would be considered worth keeping could be slightly violating one of these constraints. Another approach is based on the total number of constraints violated regardless of the magnitude of the violation. This approach was avoided as it could allow a design to survive if it only violated a single constraint even if it was a critical constraint or a large violation. Finally, a more effective method was identified in which penalties are determined based on the extent of a constraint violation. This type of penalty function is preferred as it incorporates a dynamic aspect in which the penalty increases proportionally with the magnitude of constraint violation [29]. This was the method chosen for this thesis, as was shown in equations 2.2 and 2.3. In the event of a constraint violation the objective function is increased based on the percentage of the violation ensuring that slight violations are only slightly penalized while large violations receive heavy penalties.

The next portion of the GA is the scaling function which is used to rank the designs before a selection process determines the designs that will survive into the next generation. The main ways to control the scaling are based on absolute and relative performance. The absolute method scales the individuals based on the objective function values directly; whereas the relative method determines the overall performance of a design relative to the average of all the designs. The default MATLAB GA has an existing function for scaling the designs based on their relative performance, which was maintained in the customized GA.

For the selection process a few methods available as sub-functions of the default MATLAB GA were considered. The responsibility of the selection stage is to determine the parent designs for the next generation. The three methods that were considered are as follows: the Roulette method, Stochastic Uniform selection and, the Tournament method. In the Roulette and Stochastic Uniform approaches, a uniform random number generator is used to select individuals and the probability that an individual will be selected is proportional to its objective or (fitness) value. The fitness values are normalized such that the sum of the overall fitness value is one, as shown by equations 3.1 and 3.2. The designs are then arranged in a line adding up to a total length of one, such that designs with higher fitness values span a larger portion of the line.

$$f_{sum} = \sum_{i=1}^{n_{pop}} f_i \quad (3.1)$$

$$\bar{f}_i = \frac{f_i}{f_{sum}} \quad (3.2)$$

Where f_{sum} , n_{pop} , f_i and, \bar{f}_i represent the sum of all fitness values, the number of individuals in the population, the fitness value on an individual and, the normalized individual fitness value respectively.

Up until this point the Roulette and Stochastic Uniform methods follow the same process however, the two methods are different with regards to the final selections process. In the Roulette selection process a random number between 0 and 1 is selected and located on the line of designs in turn indicating the design that will be selected, shown in Figure 3.1(a). This process is then repeated for the total number of parents denoted by n_{par} . In the Stochastic Uniform method a starting point is randomly selected and the step size δ is calculated using equation 3.3. The line is then uniformly distributed based on the step size, shown in Figure 3.1(b).

$$\delta = \frac{1}{(n_{par} + 1)} \quad (3.3)$$

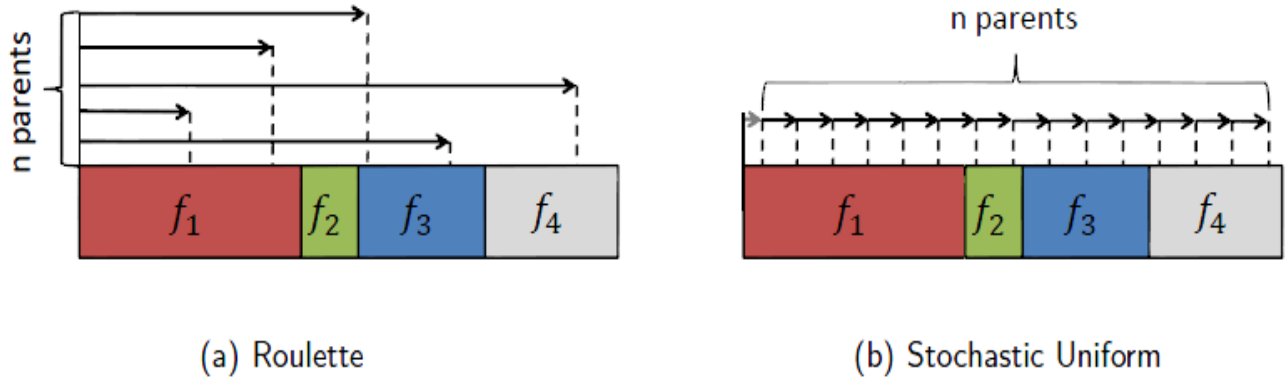


Figure 3.1 Roulette and Stochastic Uniform Selection Methods

Seen in Figure 3.1(a) the randomness of the parent selection for the Roulette method could lead to certain individuals being skipped and in rare cases high performing designs could be neglected. This problem can be mostly avoided by using nonlinear scaling on the fitness values to ensure good designs span a much larger portion of the selection line, or by adjusting the number of parents to be selected.

On the other hand, Stochastic Uniform selection will always select designs that have a normalized fitness value that is greater than the step size.

The Tournament method operates by selecting subsets of the population and then selecting the best individuals from each of the subsets. This method does not require the scaling process that the two previous methods rely on; but, selecting an appropriate size for the subsets can be critical to achieving the best results. A survey of literature regarding this issue identified 4 individuals in each subset as the most common choice, although, larger subsets provide a better likelihood that high performing individuals are chosen. This method has been shown effective when a large population is being used, as it avoids the time consuming sorting and scaling step, though, this advantage is less noticed as problems increase in complexity [28]. A comparison case was conducted to test the three selection methods against each other and the Tournament method was identified as the best choice for this work. The results of this comparison are shown in Appendix A.

The next step of the GA involves three separate processes: Elite selection, Crossover and, Mutation. Elite selection simply selects a predetermined amount of the best performing individuals from the population and passes them through to the next generation. This is done to guarantee that the best individual of a generation will at least be equal to the previous generation and not worsen as the optimization moves forward. The Crossover process is performed to mimic breeding by combining the genes from two parents to generate children. A few Crossover techniques are available within the MATLAB environment. The most common method is the single point method in which the variable vectors of the parents are split at a single point and the portions of the design variable vectors are swapped creating two new children designs. MATLAB includes a double point method as a slightly more advanced technique in which the same process as the single point method is conducted however, a second split is made in each vector and in turn more children designs can be formed. A third method called scattered crossover, is operated based on a randomly generated binary vector with the same length as the design variable vectors. A child is then formed by replacing the ones in the binary vector with the respective design variable from one parent, and replacing the zeros with the corresponding design variables from the other parent. This method was chosen for the thesis as it provides the most diversity of the three methods considered.

The Mutation step of a GA is used to provide genetic diversity by means of altering one or more of the design variables of an individual. This can be done in a variety of ways depending on the distribution function used. A common technique is to use a normal distribution to create children near an existing design. The variance can then be adjusted to widen or tighten the search radius. This technique can lead to problems if the variance is not carefully chosen as was found to be a problem with the MATLAB mutation function. It was observed that if a design variable was close to one of its boundaries the mutation function in MATLAB would occasionally push the variable outside the boundary leading to infeasible designs. In certain cases this problem caused material thicknesses to be forced negative, which in turn caused the structural solver to crash. Adding to the problem the MATLAB GA would also occasionally select the infeasible designs and continue them on to the next generation leading to slow and inaccurate convergence. For this reason a condition was added to the default mutation function that checks the design variable boundaries and in the event of a violation forces the variable back into the feasible region. The modified Gaussian mutation function used in this work can be found in Appendix A. Two parameters are also required for the mutation step, the scale and shrink factors. The scale factor is used to control the standard deviation of the mutation in the first generation, which is equated by multiplying the scale factor by the range of the initial population. The shrink factor is used to control the rate at which the average amount of mutation decreases from generation to generation. This is added to reduce the scatter of individuals as the solution nears the minimum value.

The default MATLAB GA includes several options for displaying results as the optimization is carried out, unfortunately, there lacked an efficient way to track the population data of each generation. Especially when debugging, it is useful to monitor each individual of a population and observe the various selection and genetic processes as they are applied to the designs. To assist in this, a function was created that stored the data from each generation so that it could be reviewed afterwards and used in the post processing of the results. This was also used to help visualize the progress of the designs and identify the most influential design variable changes with regards to achieving a better fitness value.

In summary it should be noted that although some aspects of the MATLAB GA have been altered to better suit this thesis, the customized GA is still heavily dependent on many of the original functions available in the MATLAB environment. Therefore the development of the custom GA should not be regarded as a major contribution of this work, rather just a necessary step to better manage the problem at hand.

Although the customized GA used in this work is still very dependent on the MATLAB GA, validation was conducted to ensure the custom approach operated properly. This was also conducted to determine if the changes made to the default GA provided any benefit to the results. To accomplish this, the Rosenbrock function shown by equation 3.4 was used as a comparison test function.

$$h_1(x, y) = (1 - x)^2 + 100(y - x^2)^2 \quad (3.4)$$

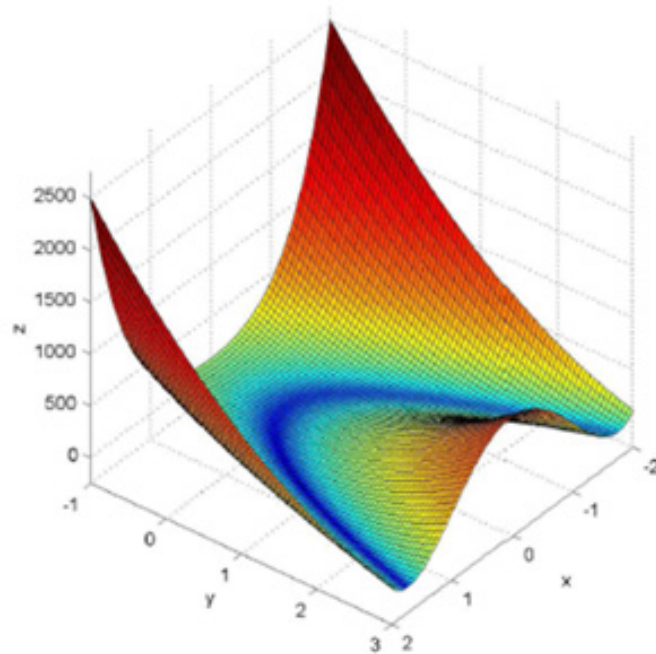


Figure 3.2 Rosenbrock function

The Rosenbrock function has its global minimum at (1, 1), surrounded by an area with a very low gradient. This function is useful for demonstrating the advantage of a GA over gradient based optimizers. While a gradient based optimization algorithm will converge in areas with low gradients it is very likely that it would take considerably more iterations than a GA. A summary of the options and conditions, used for the test, are given in Table 3.1. The MATLAB GA with the default Gaussian mutation function was compared to the customized GA and its modified Gaussian mutation function using the Rosenbrock function as a test problem.

Table 3.1 Genetic Algorithm Test Case Settings

Criteria	Rosenbrock with MATLAB GA	Rosenbrock with Custom GA
Generations	20	20
Population	100	100
Selection	Tournament	Tournament
Tournament pool size	4	4
Mutation	Gaussian	Modified Gaussian
Elite count	2	2
Crossover fraction	0.5	0.5
Scaling	0.2	0.2
Shrink	0.9	0.9
x variable boundaries	[0,2]	[0,2]
y variable boundaries	[-1, 3]	[-1, 3]

The results for the MATLAB GA are shown in Table 3.2 and Figure 3.3(a) and the custom GA results are shown in Table 3.3 and Figure 3.3(b). Comparing the results shows that the custom GA took a more gradual approach towards the solution, taking 10 steps until the converged solution was found in the 18th generation. On the other hand the MATLAB GA took only 3 steps as seen in Table 3.2, meaning that it found a solution very close to the optimum point in the first generation. This however does not mean the MATLAB GA preformed better, in fact its optimum solution had an error term an order of magnitude higher than that of the custom GA. The difference in steps taken by the two optimizers was simply a factor of the random selection of points in the first generation. A more critical difference between the two approaches can be seen in the graphical results. The white dots on each of the plots represent the 1st generation of points, the purple dots show the 10th and, the final generation is shown by the black dots. The custom GA follows a very expected pattern in which the dots gradually cluster towards the global minimum and are all within the feasible region or on the constraint boundaries. This trend is not as gracefully followed by the MATLAB GA, which is due to its mutation method. It can clearly be seen that the MATLAB GA stayed within the specified variable boundaries for the initial population. In contrast, the 10th generation shows that there are several individuals beyond the boundaries which seem to be diverging from the optimal solution. This is a result of the Gaussian mutation algorithm neglecting the design variable limits when creating the distributions for the mutations. Adding to this problem, the optimizer occasionally selects one or more of these outlying points to progress into the next generation leading to a less accurate solution. This was identified as an issue with the default GA when applied to the sailplane problem and was the main reason for developing the customized version.

Table 3.2 Rosenbrock Function Optimization Results for MATLAB GA

Generation	x	y	$h_1(x,y)$
1	1.0418	1.09582	0.01290
16	1.0431	1.0868	0.00202
18	1.0416	1.0868	0.00199

Table 3.3 Rosenbrock Function Optimization Results for Custom GA

Generation	x	y	$h_1(x,y)$
1	1.1057	1.1879	0.13076
2	0.9753	0.9170	0.11107
3	0.9632	0.9170	0.01076
9	0.9660	0.9421	0.00907
12	0.9622	0.9178	0.00746
13	0.9942	0.9946	0.00381
14	1.0277	1.0529	0.00182
15	0.9969	0.9905	0.00114
16	1.0085	1.0201	0.00096
18	1.0086	1.0165	0.00013

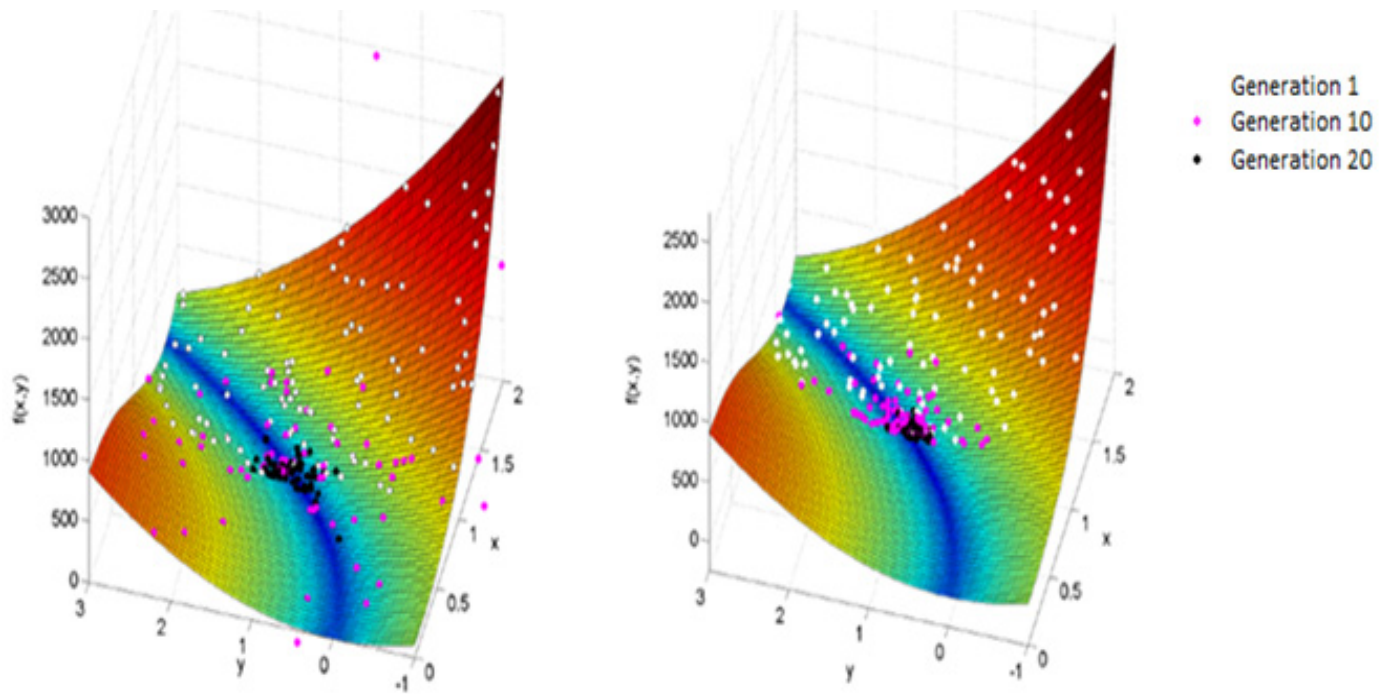


Fig 3.3(a) MATLAB GA

Fig 3.3(b) Custom GA

Figure 3.3 Rosenbrock Function MATLAB Vs Custom GA Results Comparison

3.2 Variable and Parameter Updating

The sailplane optimization conducted for this thesis was defined by a large number of parameters and design variables. To help manage this, a system of structured variables was created that organized the parameters and variables in a hierarchal way similar to a file path system, as shown in Figure 3.4. Arranging the data in this format allowed for entire sets of defining information to be passed between functions as a single variable.

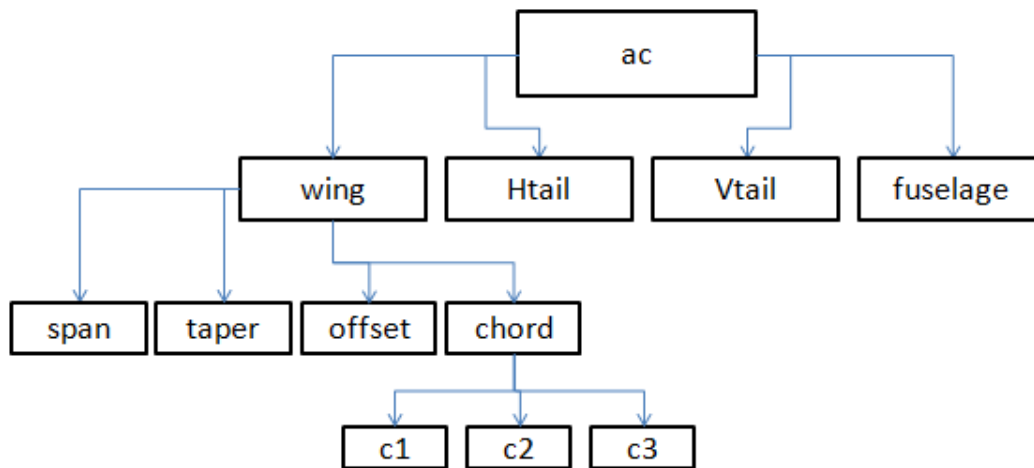


Figure 3.4 Structured Variables System

Figure 3.4 shows a small fraction of the sailplane definition variable denoted as “*ac*” which can easily be managed despite it consisting of many variables and parameters. As an example, to access the wing chord length at the root of the wing, the variable would be: *ac.wing.chord.c1*.

Not only does this portion of the optimization organize and store all of the variables, it also is responsible for updating them. When the GA chooses a set of design variables for an individual design, the vector is sent into the update function and the rest of the data needed to generate the model is calculated. Furthermore, as a design is processed in the MDA loop the update function makes the changes to the designs that occur during the iterations until convergence is achieved. After the evaluation of an individual, the data is cleared to a default case and the process is repeated for the next design.

To help reduce the total number of design variables needed to define the sailplane problem, specifically the wing box thickness variables, two decay factors were introduced. The roll of the decay factors was to define the rate of exponential decay for the wing box component material thicknesses. The first decay factor was responsible for controlling the rib thickness as a function of the wing span b and the thickness of the rib at the root of the wing box. The rib thickness was then defined as:

$$t_{rib}(b) = t_{0rib}e^{(-b/d_1)} \quad (3.6)$$

Where t_{0rib} and d_1 are the wing box root rib thickness and the decay factor variable respectively. The second decay factor was used to control the decay of the spars, flanges and stringers:

$$t(b) = t_0e^{(-b/d_2)} \quad (3.7)$$

Where t_0 represents the root thickness of the respective wing box component and d_2 is the decay factor. With this formulation the rate of exponential decay is increased with decreasing values of the decay factor. A condition was added to limit the thickness values to a reasonable minimum value to avoid unrealistically thin material thicknesses. This approach allowed for several parameters of the wing box to be expressed using only two variables, the respective root thickness and a decay factor.

3.3 Aerodynamics and Stability Analysis

The aerodynamics solver used for this thesis is a vortex-lattice solver called the Athena Vortex Lattice program (AVL) along with a component breakdown parasite drag approximation based on methods established in [30], [31]. The vortex lattice program was chosen because it is a faster and easier method to implement into design optimization than methods such as, Computational Fluid Dynamics (CFD), and still provides a suitable degree of accuracy and flexibility for the early stages of design. Additionally, this type of approach has seen successful use in many recent aerospace optimization problems for both conceptual and preliminary designs [11], [32], and [33]. The AVL program extends the capabilities of 2D aerodynamic solvers as it has the ability to calculate the lift distributions of finite lifting surfaces. This is achieved by employing an extended vortex lattice model for the lifting surfaces and a slender body model for the fuselage. Furthermore, a dynamic flight analysis can be modeled using a full linearization of the aerodynamic model combined with user specified mass properties [34]. The aerodynamic analysis

is conducted by applying horseshoe vortexes on each of the panels that make up a lifting surface. The Biot-Savart law is then used to determine the fluid velocities along with a boundary condition enforcing that the flow field may not flow through the lifting surfaces [35], [36]. AVL is considered a moderate fidelity analysis package as it is more complex than simple empirical aerodynamic approaches but, lacks the finer detailed analysis available in CFD packages. AVL assumes quasi-steady flow, meaning that unsteady vorticity shedding is neglected. This translates to the assumption that any oscillatory motion modeled must be slow enough such that the period of oscillation is much longer than the time it takes the flow to transverse the airfoil chord [34]. However, this is not a problem for the relatively simple sailplane manoeuvres considered in this research, namely a steady glide condition and a maximum gust loading case simulated by a “pull up” manoeuvre.

AVL uses the classical Prandtl-Glauert transformation to treat compressibility effects by converting the Prandtl-Glauert equation into the Laplace equation, which is then solved using the basic incompressible method. The Kutta-Joukowski relation is then applied to each vortex to calculate the aerodynamic forces under the assumption of irrotationality and linearization about the freestream [34]. The validity of this approach is gauged based on the Prandtl-Glauert factor $1/B$ as shown by equation 3.8.

$$\frac{1}{B} = \frac{1}{\sqrt{(1 - M^2)}} \quad 3.8$$

The Prandtl-Glauert method is considered valid for a Prandtl-Glauert factor ranging from 1-1.25 or, a freestream Mach number $M \leq 0.6$ after which transonic flow may exist and the solution becomes suspect [37]. For the sailplane optimization this process is a valid approach as the freestream Mach number remains far below the limit.

The stability component of the AVL software provides static and dynamic stability coefficients that are used to formulate constraint functions that ensure acceptable handling and flight stability is maintained. It is also used to calculate the aerodynamic center of the sailplane along with the trim requirements for the elevator, rudder and ailerons.

To run the AVL software three files are needed as inputs, a geometry file, a mass file and, a case file. These three files are automatically created from the data generated in the variable and parameter

The drawing consists of three main parts: a plan view at the top, an elevation view at the bottom, and a section view on the right. The plan view shows the hull's footprint with a central circular structure and a horizontal axis with markings at 5.0, 10.0, and 15.0. The elevation view shows the hull's profile with a vertical axis and markings at 1.0, 2.0, 3.0, 4.0, 5.0, and 6.0. The section view on the right shows a cross-section of the hull with a horizontal axis and markings at 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, and 7.0. The section view also includes a vertical axis with a marking at 1.0. The hull is shown with a series of horizontal lines representing the plating.

The mass file is required for the dynamic flight analysis aspect of the AVL program. Mass and inertial properties of each of the components of the model can be entered in separately or, the total values can be specified. For this thesis the mass, inertial properties and center of gravity of the entire sailplane are calculated during the data updating process and entered directly.

25

Table 3.4 AVL Case File Input Parameters

Variable	Parameter
ϕ	Bank angle (deg)
C_L	Lift Coefficient
V	Velocity (m/s)
m	Mass (kg)
ρ	Air Density (kg/m ³)
g	Acceleration due to Gravity (m/s ²)
X	X Position of CG (m)
Y	Y Position of CG (m)
Z	Z Position of CG (m)

Since the aerodynamic loading of the wing is needed for the structural analysis module, the mesh of the wing was kept as consistent as possible between the aerodynamics and structures disciplines. This allowed for a much easier force mapping process which was done by solving the strip and element forces using the AVL program, and then generating normalized pressure distribution splines in the chord and span directions. Figure 3.6 shows an example of a normalized pressure distribution that was calculated using the splines created from the AVL results. Further transformation of the normalized pressure distribution into the lift forces is conducted in the structures module.

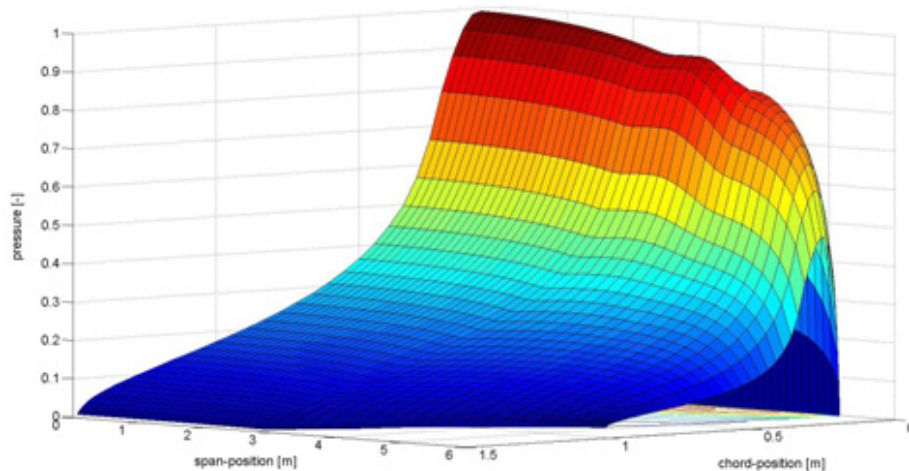


Figure 3.6 Normalized AVL Pressure Distribution

3.4 Structures & Weights and Balance

The structures/weights & balance discipline consists of a Finite Element Analysis (FEA) of a wing box structure using the commercial software ANSYS and, a simple function that manages the weights and locations of each of the sailplane components to determine their center of gravity and inertial properties. Since the ANSYS software used in this discipline was operated in batch mode, several MATLAB functions were created to control the processes involved in generating and solving the FEA models. The MATLAB functions that control the structures discipline were responsible for writing the files that define and generate the wing box model as well as giving the solution commands and collecting the end results. The general components of an input file for the ANSYS model are shown in Figure 3.7. As shown in Figure 3.7 the ANSYS file starts with a header in which the job name, data path, units and, ANSYS execution options are set. Following that, the element and material properties are

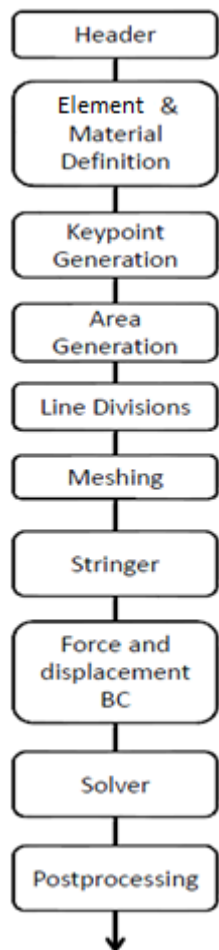


Figure 3.7 ANSYS Flow Diagram

specified. These two initial steps remain constant for the entire optimization; however, the remaining portions of the ANSYS input file were updated for each individual design. The key point generation in the next step is used to define the wing box geometry by setting the corner points of each rib from which the rest of the components are defined. The key point coordinates are a function of the design variables of the optimization and therefore are repositioned and updated for each design. From the key points the areas for each of the wing box components are then defined by systematically creating each area based on 4 corner key points. The line divisions section is responsible for setting the element sizing parameter for each of the lines that served as edge boundaries to the areas. To control this, a dynamic meshing algorithm was created to maintain the best possible mesh quality for the changing wing box geometry. Once the element sizing is selected the wing box is meshed with the ANSYS mapped meshing option. Stringers are defined after the meshing of the areas is complete, since they require nodes to be defined upon. The force and displacement information is then defined consisting of: the aerodynamic loading calculated in the aerodynamics module and, the degree of freedom (DOF) constraints used to fix the root of the wing box. The solution commands are then given and the static FEA model is solved. Finally the post processing stage of the structures module is used to save the required information for the

aerodynamics discipline.

The materials selected for the wing box are commercially available composites found in reference [38]. The majority of the wing box structure is constructed of a standard carbon fibre reinforced polymer (CFRP) however, the stringers are modeled as high modulus unidirectional CFRP. Table 3.6 summarizes the material properties of the two materials.

Table 3.5 Material Properties [38]

Property	Std carbon fibre fabric weave 50% vol (fabric)	M55 UD (modeled in fibre direction)
Density (kg/m ³)	1600	1650
Poisson's ratio	0.1	0.3
Elastic modulus 0° (Gpa)	70	300
Elastic modulus 90° (Gpa)	70	12
Ultimate tensile stress (Mpa)	600	1600
Ultimate compressive stress (Mpa)	570	1300
In plane shear modulus (Gpa)	5	5

The standard (CFRP) composite material was not modeled as a layered laminate due to the complexity and added computational expense it would bring. For this reason it was specified as an orthotropic material using the ANSYS SHELL181 elements. The SHELL181 element is a simple four node element with six DOF at each node consisting of: translations in the x, y and, z directions and rotations about the x, y and z axes. This element type was chosen for its ability to model twist and deflection which are the essential deformations needed for the aerodynamics model. The stringers were positioned such that the fibre direction aligned with the wing span to increase the stiffness of the wing box when subject to bending. To model the stringers, BEAM4 elements were used in ANSYS. The BEAM4 element has the same DOFs as the SHELL181 element and depending on the moments of inertia specified it can be modeled as a number of cross sectional shapes [39]. For this work 'Z' shaped stringers were used, more information on the stringers can be found in Appendix A. Figure 3.8 shows the two elements used in this thesis.

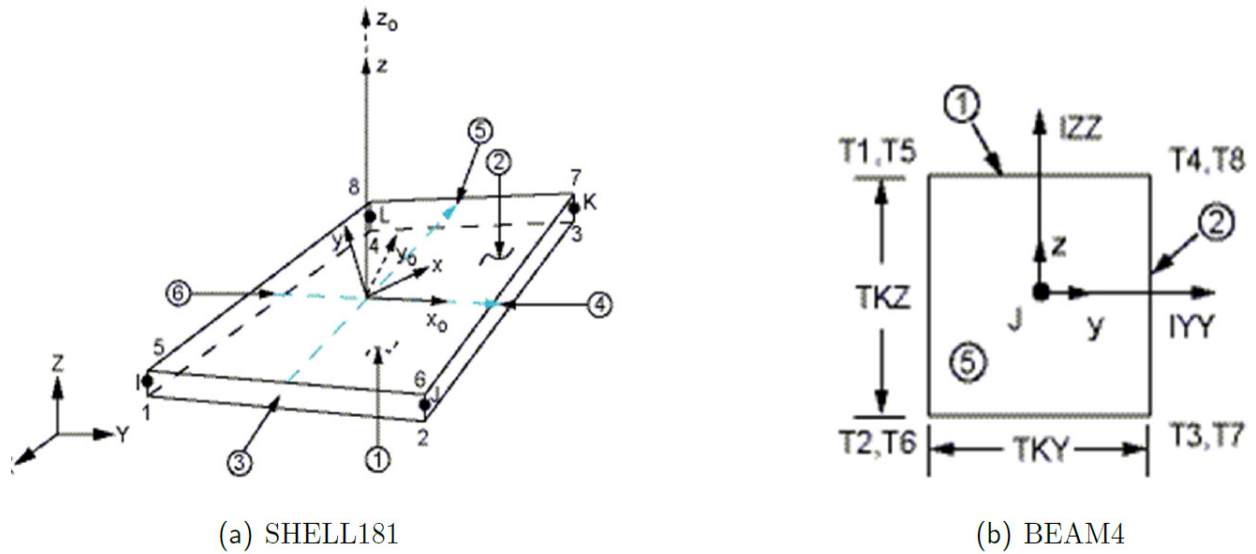


Figure 3.8 ANSYS Element Descriptions [39]

The geometry generation as stated above is based on a set of key points that define the corners of each wing box rib as shown in Figure 3.9. The key points are defined by their x , y and, z coordinates which are constrained based on the airfoil shape and chord length at each spanwise location. The process was designed such that any airfoil profile can be read in to the program and used for the model however, in this work the airfoil was held constant. After the rib areas are created at each spanwise section of the wing, the areas for the flanges and spars are generated by connecting the corresponding points of adjacent rib areas.

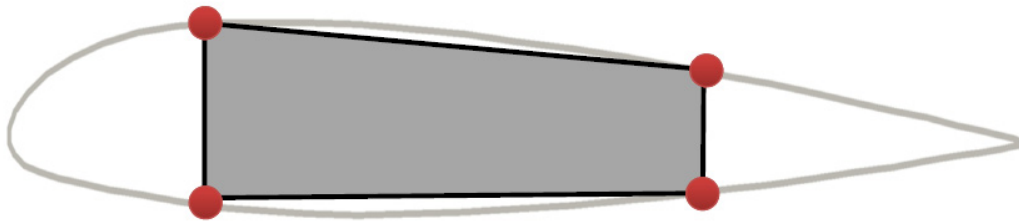


Figure 3.9 Key Point and Rib Configuration

For accurate results in a structural analysis, element aspect ratios close to one are desired [40]. This eliminates problems such as locking that lead to unrealistically stiff element properties. Unfortunately, since the size of the wing box was able to change in both the x - and y -directions, a static mesh was not appropriate. To best address this problem a mapped mesh was applied. However, two problems arise when creating a mapped mesh for a wing box that tapers towards the tip. As the wing box area decreases toward the wing tip, the element sizes also tend to decrease. This is not efficient for the structural analysis of a wing which experiences less applied load toward the wing tip and thus does not

require a refined mesh in that area. On the contrary, a finer mesh at the wing tip is recommended for the vortex lattice aerodynamic solver, therefore a compromise was required. A second issue is that as the wing box changes in chord length, the number of elements and thus the number of nodes must also change. Since the stringers are defined by the chord-wise nodes, the number of stringers was also allowed to change with the changing width of the wing box. Considering this, a dynamic meshing routine was developed to calculate the element sizing for each design and find a compromise to best suit both the structures and aerodynamic meshes. The first step of the mesh sizing is to determine the number of line divisions in the wing box chord and thickness directions or the x and, z directions. This is done using equations 3.9 and 3.10 in which the number of line divisions, Λ is found by rounding the average number of divisions for opposed surfaces to find a compromise to best suit both surfaces.

$$\Lambda_x \approx \frac{\Delta l_{TF}/e_0 + \Delta l_{BF}/e_0}{2} = \frac{\Delta l_{TF} + \Delta l_{BF}}{2e_0} \quad (3.9)$$

$$\Lambda_z \approx \frac{\Delta l_{LE}/e_0 + \Delta l_{TE}/e_0}{2} = \frac{\Delta l_{LE} + \Delta l_{TE}}{2e_0} \quad (3.10)$$

Where e_0 and, Δl denote the specified initial element size for the root rib and, the line length for which the subscripts TF, BF, LE and, TE are the top flange, bottom flange, leading edge spar and, trailing edge spar respectively. The average real element size \bar{e} can then be calculated, which may differ from the desired element size \tilde{e} especially if there are a small number of elements (a large element size or small wing box).

$$\bar{e} = \frac{\Delta l_{LE}/\Lambda_z + \Delta l_{TE}/\Lambda_z + \Delta l_{TF}/\Lambda_x + \Delta l_{BF}/\Lambda_x}{4} \quad (3.11)$$

The setup for equation 3.11 gives equal weighting for each area to have an element size as close to 1 as possible. However, this could be changed to put more emphasis on certain areas of the wing box if desired.

Finally the number of divisions in the y, or wingspan direction $\Lambda_{y,i}$ is set using equation 3.12. This parameter changes for every segment of the wing box and is the parameter used to control the element aspect ratio. It is set to be as close as possible to the real element size \bar{e} that was calculated using equation 3.11. There must be an equal number of y-divisions for the flanges and spars or the mapped

mesh would be lost. Therefore the average real element size is scaled with the chord length of the two ribs that border the segment. The distance between two adjacent ribs is divided by the average of the two scaled element sizes. The result is a mesh that best fits the x and z element dimensions of the bordering ribs.

$$\Lambda_{y,i} = \frac{b_i}{\frac{1}{2} \bar{e} \left(\frac{c_i}{c_0} + \frac{c_{i+1}}{c_0} \right)} \quad (3.12)$$

Where b_i and c_i represent the span and chord at each segment i and, c_0 represents the initial wing box root chord length. Figure 3.10 shows a test mesh that was checked for the average element aspect ratios in the tapered sections of the wing box, the results of the test are shown in Table 3.7

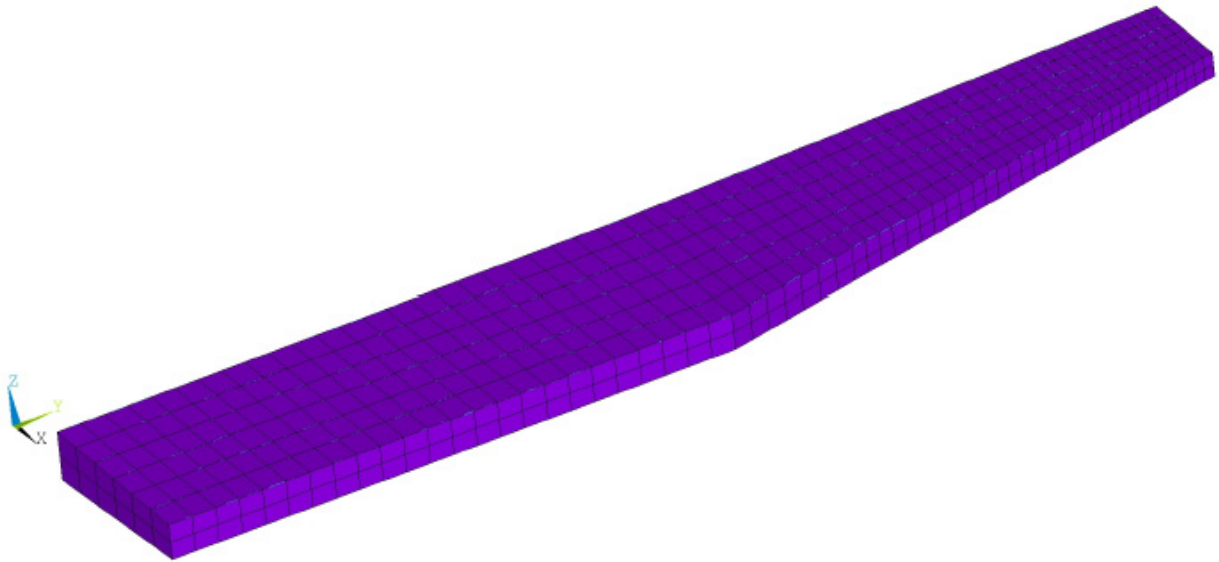


Figure 3.10 Test ANSYS Wing box Mesh

Table 3.6 Average Element Aspect Ratios for Tapered Sections

Segment	Avg. AR on leading edge spar	Avg. AR on trailing edge spar	Avg. AR on upper flange	Avg. AR on lower flange
10	1.2573	0.9057	1.1392	1.1338
11	1.1406	0.8216	1.0335	1.0286
12	1.0561	0.7608	0.9570	0.9524
13	1.2145	0.8749	1.1005	1.0953
14	1.1089	0.7988	1.0048	1.0001
15	1.2040	0.8673	1.0909	1.0858
16	1.0772	0.7760	0.9761	0.9715
17	1.1089	0.7988	1.0048	1.0001

The results in Table 3.7 show that the dynamic meshing routine was able to maintain reasonable average element aspect ratios by automatically adjusting the element sizing to accommodate the wing box taper. Another test was conducted in which the average element aspect ratios of the entire upper and lower flange were monitored as the element line size was varied from 0.001m to 0.01m. The results are shown in Figure 3.11 where the points on the plot refer to increments of 0.001m. As can be expected the mesh is able to maintain an average aspect ratio of very close to one for small element sizes and starts to oscillate around the target value as the element size is increases. The results demonstrate the meshes ability to adapt to changing element sizes maintaining an average aspect ratio within ± 0.1 of the desired value. Further mesh validation can be found in Appendix A.

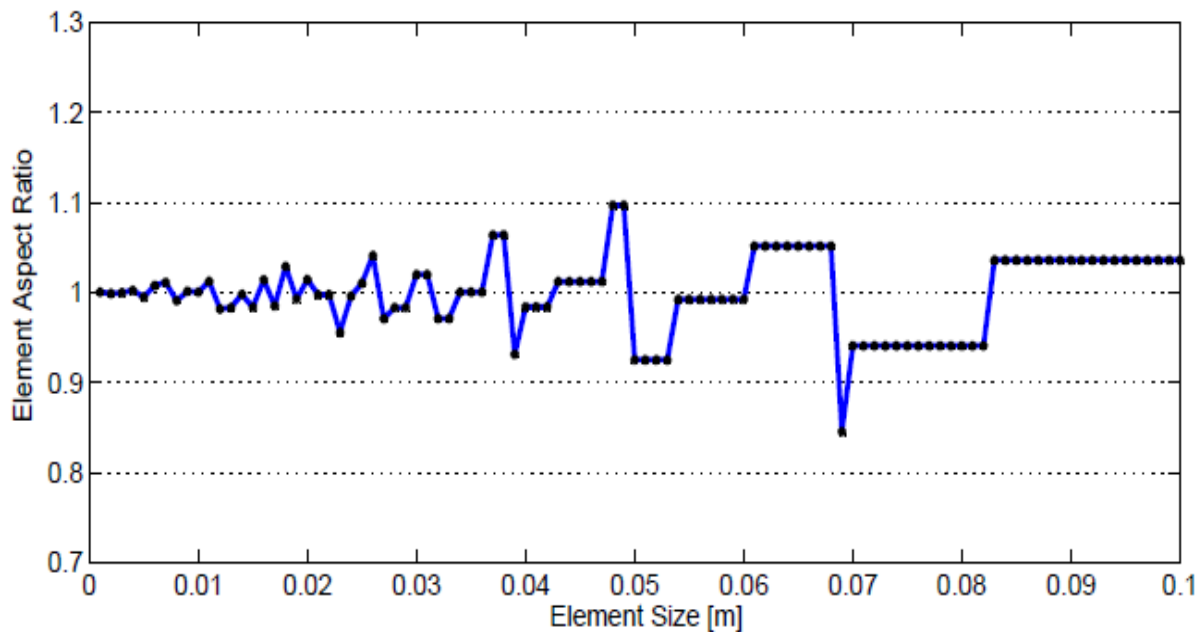


Figure 3.11 Average Element Aspect Ratio VS Element Edge length

The next step in the structures module is the application of the constraints to the nodes at the root of the wing box. However, a problem arises when constraining a model as a cantilever by fixing the root nodes in all their DOFs. The problem is, when completely fixing a row of nodes and leaving the next spanwise row free to move, singularities occur at the joint areas leading to unrealistically high stress values. To alleviate this numerical problem, a buffer zone was added to the root section of the wing box effectively extending the wing box into where a fuselage would exist. This buffer zone is then fixed at its root leaving it to experience the singularities. The stress and weight of this buffer section is then ignored and the more realistic values of the actual wing box root can be monitored.

Next, the aerodynamic loading of the wing box is determined from the pressure distributions provided by the aerodynamics module. Since the aerodynamic model considers the entire wing surface, as opposed to the structures model which only considers the internal wing box, an approximation is made. The leading and trailing edge spars of the wing box do not actually reach the leading and trailing edge of the airfoil and therefore some of the loading must be shifted. This process divides the loading into three sections: the loading forward of the leading edge spar is shifted to the leading edge spar, the loading aft of the trailing edge spar is shifted to the trailing edge spar and, the loading in between the spars is mapped to the corresponding nodes that are consistent between the aerodynamics and structures meshes. Figure 3.12 gives a visualization of this process.

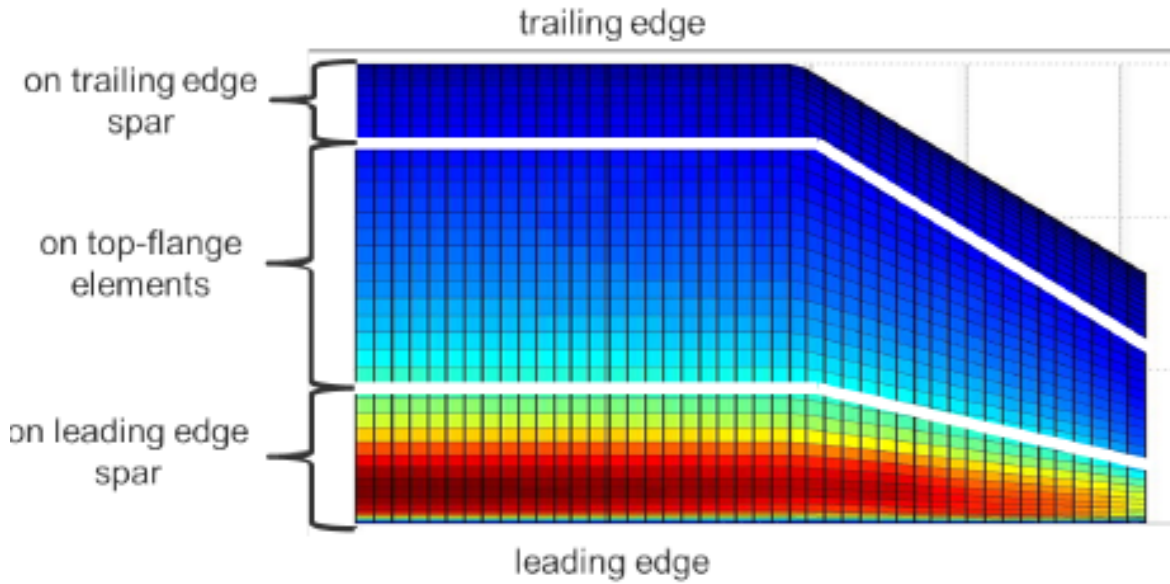


Figure 3.12 Aerodynamic Force Mapping

The aerodynamics module provides the normalized element pressure results and therefore they must be transformed into nodal forces before they can be applied to the structures model. The first step is to de-normalize the pressure coefficient and then calculate the lift force on each element using the fundamental relations for the pressure coefficient C_p found in reference [37]:

$$C_p = \frac{\Delta p}{q} \quad (3.13)$$

$$\Delta p = p - p_\infty \quad (3.14)$$

$$q = \frac{1}{2} \rho_{\infty} V_{\infty}^2 \quad (3.15)$$

$$Lift = \Delta p A \quad (3.16)$$

Where p is the local pressure on the airfoil, p_{∞} the freestream pressure, q the dynamic pressure, ρ_{∞} the freestream air density, V_{∞} the freestream velocity and A the reference area. The lift for an element can then be expressed with the local pressure coefficient.

$$L_e = \int \frac{1}{2} C_p \rho_{\infty} V_{\infty}^2 dx dy \quad (3.17)$$

Equation 3.17 can then be expressed without the integral as the pressure distribution is based on discrete element values:

$$L_e = \frac{1}{2} C_p \rho_{\infty} V_{\infty}^2 \Delta x_e \Delta y_s \quad (3.18)$$

Where Δy_s is the AVL strip dimension in the y-direction given by the dynamic mesh routine and Δx_e is provided in the AVL output file along with the local pressure coefficients, C_p . In the next step, the force per AVL element is mapped onto the ANSYS elements. A spline, representing the integrated pressure along one strip with respect to the x-position, is generated for this purpose. For the FEA model, forces are calculated by the evaluation of the integrated pressure at the element edges; thus the spine function in MATLAB is used. The forces, which are applied on the nodes at the joint of the front spar (FS) and the upper flange denoted by \tilde{L}_{FS} as well as on the rear spar (RS) and upper flange are denoted by \tilde{L}_{RS} are calculated as follows:

$$\tilde{L}_{FS} = \Delta y_s \left[\int_0^{x_{FS}} \Delta p(x) dx \right] \quad (3.19)$$

$$\tilde{L}_{RS} = \Delta y_s \left[\int_{x_{RS}}^c \Delta p(x) dx \right] \quad (3.20)$$

where the pressure distribution $\Delta p(x)$ is constant on each AVL element, and calculated as shown in Equation 3.13. The integral limits are the x-position of the front spar x_{FS} and rear spar, x_{RS} as well as the

trailing edge of the wing, which equals the chord length c . The lift force on the i^{th} FEA element of a strip of width Δy_s on the upper flange can be calculated as:

$$\tilde{L}_{e,i} = \Delta y_s \left[\int_0^{x_{e,i}^+} \Delta p(x) dx - \int_0^{x_{e,i}^-} \Delta p(x) dx \right] \quad (3.21)$$

where the integration boundaries $x_{e,i}^+$ and $x_{e,i}^-$ denote the x position of the front edge of the i^{th} FEA element and the rear edge position respectively, the two boundaries are correlated as follows:

$$x_{e,i}^+ = x_{e,i+1}^- \quad (3.22)$$

With this approach, the discrete element-wise AVL output is integrated using the FEA element boundaries, to create a lift force distribution across the entire top surface of the wing box as shown in Figure 3.13. Figure 3.14 summarizes the various stages involved in the generation, loading and testing of the wing box model.

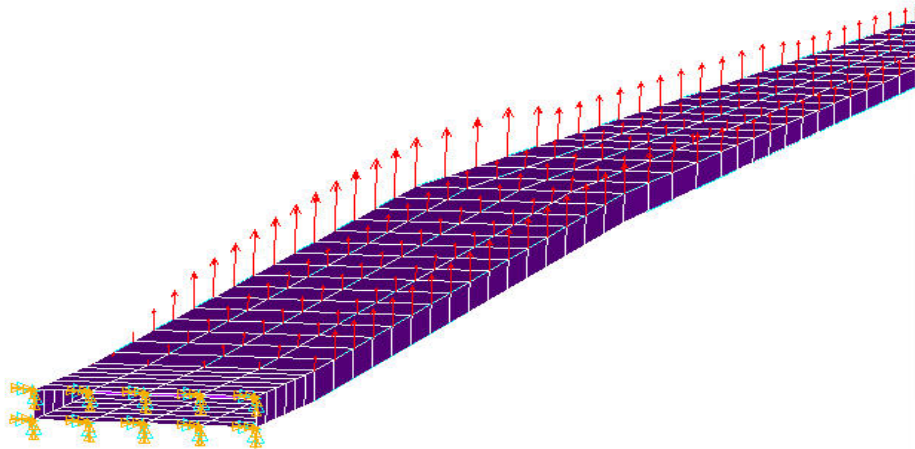


Figure 3.13 ANSYS Model with Forces Applied

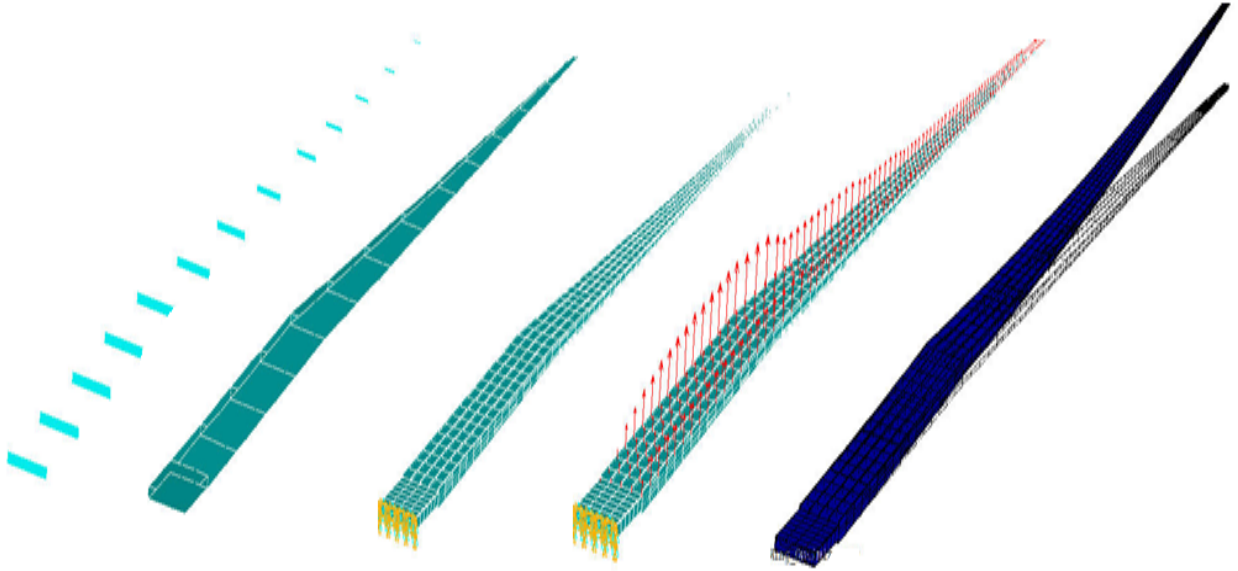


Figure 3.14 Stages of ANSYS Wing box Creation and Analysis

3.5 Performance & Airworthiness

The performance discipline uses empirical formulations taken from the Transport Canada Chapter 522 airworthiness regulations for sailplanes as well as the Federal Aviation Regulations [41], [42]. Chapter 522 is a detailed airworthiness manual used to check compliance with sailplane certification regulations. Many of these regulations require details regarding the performance and handling of the design that can't easily be modeled in early design stages. However, compliance to some of the regulations (mainly flight speeds and loading) can be estimated based on the basic sailplane configuration. This can have a significant effect on the optimization and may help find an optimal design that could prove easier to certify later on in the design process.

To estimate the stall speed of the sailplane a maximum design weight similar to existing sailplanes, W of 500 kg was set and C_{Lmax} was estimated based on existing designs to be 1.5 at the best glide speed.

$$V_s = \sqrt{\frac{2W}{\rho S C_{Lmax}}} \quad (3.23)$$

Where, ρ and S are the density and wing plan form area respectively. The sailplane design manoeuvre speed V_A and the design maximum speed V_D are calculated as follows:

$$V_A = V_{S1}\sqrt{n_1} \quad (3.24)$$

$$V_D = 18 \left(\sqrt[3]{\left(\frac{W}{S}\right) \left(\frac{1}{C_{Dmin}}\right)} \right) \quad (3.25)$$

Where V_{S1} , n_1 , (W/S) and C_{Dmin} are the estimated stall speed at design maximum weight with wing-flaps neutral and air brakes retracted, the limit load factor for utility class sailplanes (+5.3), the wing loading at max design weight in (daN/m^2) and the minimum drag coefficient respectively. The following equations are used to determine the gust loading factor:

$$n_g = 1 \pm \left[\frac{\left(\frac{k}{2} \rho_0 U V \alpha\right)}{\frac{mg}{S}} \right] \quad (3.26)$$

$$k = \frac{0.88\mu}{5.3 + \mu} \quad (3.27)$$

$$\mu = \frac{2 \frac{m}{S}}{\rho l_m a} \quad (3.28)$$

Where ρ_0 is the density at sea level, U is the gust speed in m/s , V is the equivalent flight speed, α is the wing lift curve slope per radian, m is the mass of the sailplane, g is the acceleration due to gravity, k is the gust alleviation factor, μ is the non-dimensional glider mass ratio, ρ is the density at altitude and l_m is the mean geometric chord of the wing.

For the maximum gust loading flight case conducted by the aerodynamics and stability discipline the looping flight condition used to simulate the proper load factor also required the looping flight lift coefficient. To calculate the lift coefficient for this case the load factor equation as shown in [43] is used.

$$n = \frac{\frac{1}{2} \rho V^2 S C_L}{mg} \quad (3.29)$$

Sailplane performance is often governed by cross-country theory which describes a sailplanes basic flight mission as a gliding section shown in Figure 3.15 as (A-B) followed by a thermal climbing circling pattern to regain as much altitude as possible (B-C).

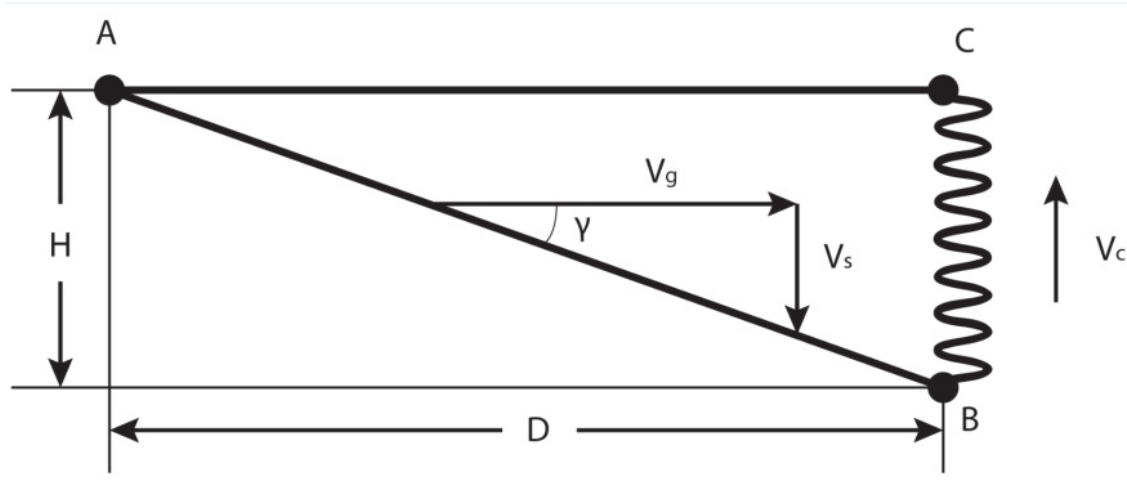


Figure 3.15 Idealized cross-country flight profile [44]

To determine the sink rate during the glide stage the following equation is used:

$$V_{sg} = \frac{C_D}{C_L^{3/2}} (2W/\rho S)^{0.5} \quad (3.30)$$

Then the total rate of climb in a thermal is given by:

$$V_C = V_T(r) - V_{St} \quad (3.31)$$

where $V_T(r)$ is the upward thermal velocity as a function of the radius r from the center of the thermal.

The sink rate V_{St} while in a thermal is given by:

$$V_{St} = \frac{C_{Dc}}{C_{Lc}^{1.5}} \left[1 - \left(\frac{2W}{\rho S C_{Lc} g r} \right)^2 \right]^{-0.75} \left(\frac{2W}{\rho S} \right)^{0.5} \quad (3.32)$$

where C_{Dc} , C_{Lc} , r and g are the drag and lift coefficients during climb, the radius from the center of the thermal and, the acceleration due to gravity respectively. To determine the sailplanes average cross-country flight speed equation 3.33 is used:

$$V_{Avg} = \frac{V V_C}{V_C + V_{sg}} \quad (3.33)$$

Chapter 4

Results

4.1 Problem Definition

The goal of the optimization was to optimize a standard 15-meter wingspan sailplane for the lowest possible sink rate during gliding flight as shown in equation 3.30. The optimization was conducted using 25 design variables including aspects of the sailplane layout, sizing, and wing box structure. The upper and lower boundaries of the design variables were set within reasonable limits based on a survey of existing designs and engineering knowledge. The optimization was also subject to 11 constraints ensuring acceptable handling qualities and compliance with basic airworthiness directives. A summary of the design variables, constraints and their boundaries can be found in Table 4.1 Figure 4.1 is also included to show some of the variables in context.

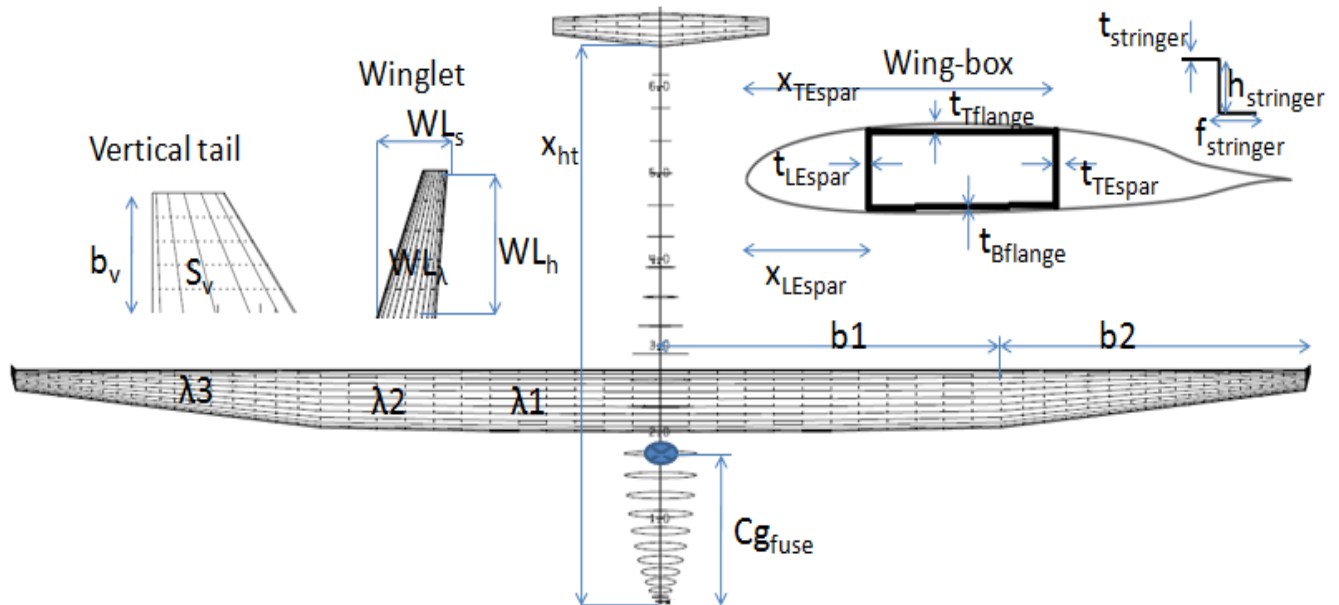


Figure 4.1 Diagram showing many of the design variables

Table 4.1 Design Variables and Constraints

Variable	Symbol	Upper boundary	Lower boundary
Wing span segment length in % of total span	$b1, b2$	0.7	0.3
Taper ratio of wing segments	$\lambda1, \lambda2, \lambda3$	1	0.2
Aspect ratio of wing	AR_{Wing}	30	23
Winglet height (m)	WL_h	0.6	0.1
Winglet taper ratio	WL_λ	0.8	0.1
Winglet tip offset (sweep) (m)	WL_s	0.5	0.1
Leading edge wing spar location in % of chord	x_{FS}	0.25	0.15
Trailing edge wing spar location in % of chord	x_{RS}	0.65	0.5
Leading and trailing edge wing spar root thickness's (m)	t_{FS}, t_{RS}	0.05	0.003
Wing box root flange thickness's (m)	t_{TF}, t_{BF}	0.03	0.003
Stringer height at wing root (m)	h_s	0.02	0.005
Stringer flange width at wing root (m)	f_s	0.01	0.0025
Stringer thickness at wing root (m)	t_s	0.002	0.001
Exponential thickness decay factors	d_1, d_2	5	2
Wing box root rib thickness (m)	t_{Rib}	0.02	0.005
Vertical tail area (m ²)	S_v	1.2	0.8
Vertical tail span (m)	b_v	1.3	1
Horizontal tail aspect ratio	AR_h	6	3
Horizontal tail apex location (longitudinal axis) (m)	x_{ht}	6.85	6
Fuselage center of gravity position (longitudinal axis) (m)	Cg_{fuse}	3.5	2.2
Constraint	symbol	unit	limit
Maximum wing box root stress	σ_{max}	Mpa	< 570/1.5 _{SF}
Static margin	K	-	>5%
Spiral stability ($L_\beta N_r - L_r N_\beta$)	Spiral	-	>0
Control surface trim angles	$\delta_{Atrim}, \delta_{Etrim}, \delta_{Rtrim}$	deg	+/-30
Stall speed	V_s	m/s	<22.2
Wing tip deflection and max g load	$w_{deflect}$	m	<1.5
Pitching moment coefficient	C_m	-	<0
Lift VS drag ratio	L/D	-	>45
Wing weight	m_{wing}	kg	<150

An existing high performance sailplane was chosen as a target for the design goals in the optimization. The ASW 27 B is a high performance 15 meter wingspan sailplane that achieves a minimum sink rate of 0.52 m/s at a mass of 320 kg [45]. More details on the ASW 27 B can be found in Appendix A. Therefore the goal for the optimization was to reach a design with comparable performance to the ASW without violating any of the constraints. Certain aspects of the sailplane were not considered as design variables, including the airfoil. Therefore they were selected to match the comparison design.

As stated in Chapter 3.4, two flight cases were considered in the aerodynamic analysis portion of the optimization: a trimmed gliding condition and a maximum gust load condition. The trimmed gliding condition was used as the condition to assess the sailplanes performance and was used to calculate the objective function. The results used in the comparison to the ASW 27 B sailplane were also gathered from this case. The maximum gust load condition was used to provide the aerodynamic loading for the ANSYS wing box structural analysis, to ensure the wing box structure could support the required loading. Since many of the parameters used to define the two AVL flight conditions are based on the design variables used in the optimization; they were constantly changing and therefore a set of initial parameters were rewritten at the beginning of each iteration of the MDA. Table 4.2 shows the initial AVL input parameters for the two cases.

Table 4.2 AVL Initial Case Parameters

Parameter	Flight condition	
	Trimmed glide	Maximum gust load
Mass[kg]	400	400
Airspeed[m/s]	30	65
Altitude[m]	0	0
Center of gravity coordinates[m]	[2 ,0 ,0]	[2 ,0 ,0]
Acceleration of gravity[m/s ²]	-9.81	-9.81
Load factor	1	5.3
Lift coefficient	-	1
Pitching moment coefficient	0	0
target		
Aileron deflection target[deg]	0	0
Rudder deflection target[deg]	0	0

As discussed in Chapter 3.2 there are a number of parameters required by the GA, the parameters selected for the sailplane problem are summarized in Table 4.3.

Table 4.3 Genetic Algorithm Options for Sailplane Optimization

Parameter	Setting
Generations	100
Population	100
Selection	Tournament
Tournament pool size	10
Mutation	Modified Gaussian
Elite count	10
Crossover fraction	0.5
Scaling	0.3
Shrink	0.9

4.2 Optimization Results

The convergence history of the sailplane optimization is presented in Figure 4.2. The objective function which was the gliding sink rate is plotted on the y-axis and, the generation number is shown along the x-axis. The penalized and non-penalized objective function values are shown by the hashed and solid lines respectively. From the convergence history plot it can be seen that after 4 generations the optimum designs were no longer being penalized and therefore were within the feasible solution space. Shown under the convergence plot are the optimum designs from the 1st, 3rd, 21st and final generation. These designs were selected to highlight the main changes in the design. Table 4.4 is a summary of the design variable and constraint values for each of the highlighted designs. The constraint values are presented as their actual value rather than the normalized value used to represent them in the optimization.

Gliding Sink Rate Vs Generation Number

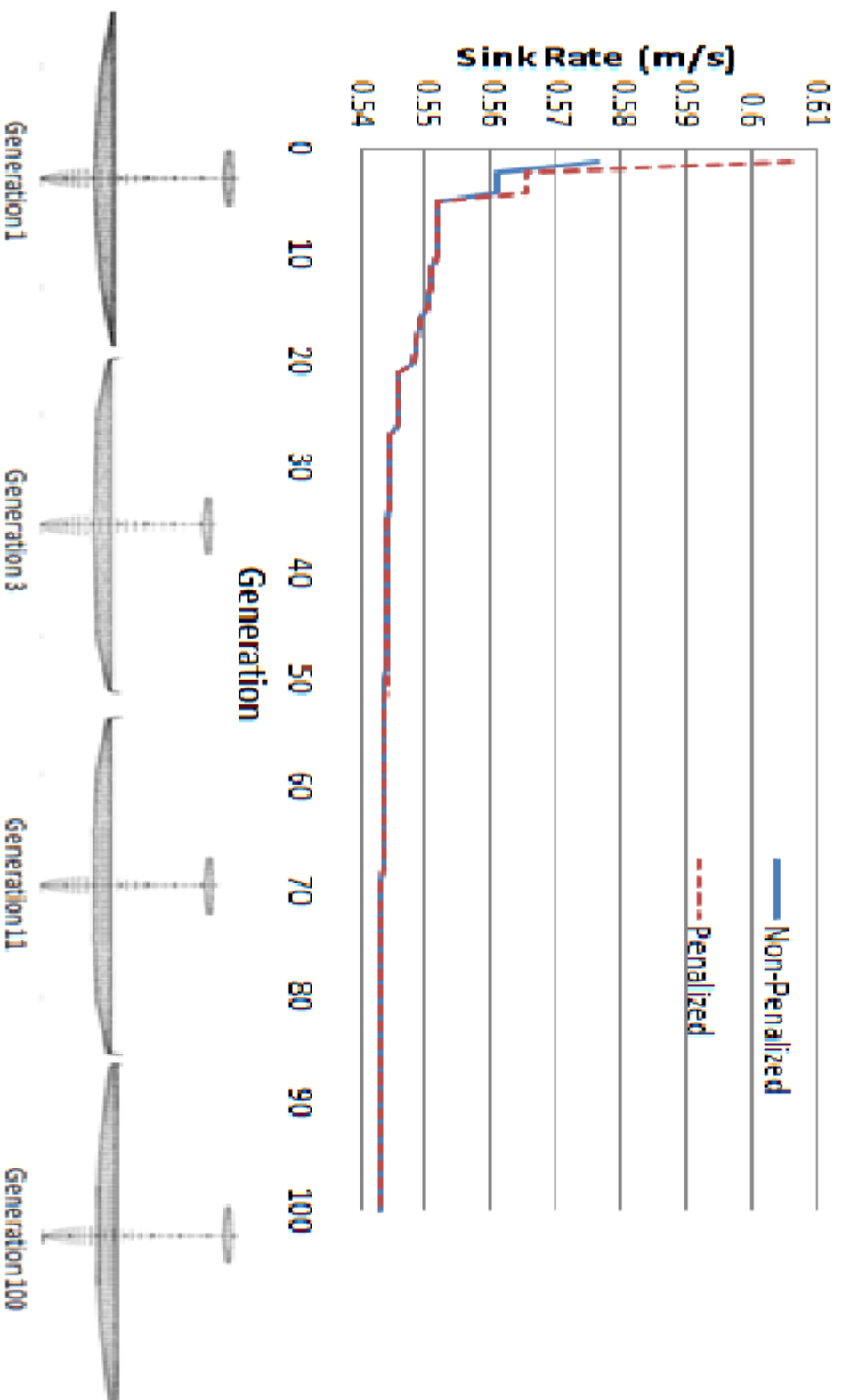


Figure 4.2 Penalized and non Penalized Objective Function Convergence History Including Top Views for Optimum Designs

Table 4.4 Design Variable and Constraint Values for Selected Designs

Variable	Generation 1	Generation 3	Generation 21	Generation 100
b1,b2	0.6415,0.3585	0.6899,0.3101	0.5772,0.4228	0.5772,0.4228
$\lambda_1, \lambda_2, \lambda_3$	0.85,0.78,0.22	0.90,0.88,0.27	0.97,0.85,0.47	0.99,0.85,0.47
AR_{wing}	25.922	25.3386	25.1274	25.0037
WL_h	0.3430	0.3842	0.5967	0.5967
WL_{λ}	0.3076	0.5117	0.5856	0.6934
WL_s	0.1052	0.3472	0.2042	0.2042
x_{FS}	0.2220	0.2183	0.2329	0.2234
x_{RS}	0.5814	0.5101	0.5142	0.5142
t_{FS}, t_{RS}	0.0249,0.0081	0.0284,0.0216	0.0198,0.0145	0.0203,0.0185
t_{TF}, t_{BF}	0.0041,0.0202	0.0104,0.0200	0.0145,0.0165	0.0145,0.0179
h_s	0.0191	0.0172	0.0138	0.0138
f_s	0.0060	0.0062	0.0086	0.0062
t_s	0.0016	0.0015	0.0015	0.0017
d₁, d₂	4.51,3.87	4.69,3.71	4.29,3.95	3.41,3.71
t_{Rib}	0.01315	0.0068	0.0129	0.0133
S_v	0.9026	1.2503	1.0487	0.8318
b_v	1.087	1.1449	1.1973	1.2503
AR_h	3.959	4.624	4.057	4.630
x_{ht}	6.782	6.069	6.317	6.5456
Cg_{fuse}	2.617	2.759	2.646	2.6464
Constraint				
σ_{max}	153	138.7	140.5	143.1
K	34.5	16.67	24.8	22
$(L_{\beta}N_r - L_rN_{\beta})$	8.82	6.53	9.8	9.3
$\delta_{Atrim}, \delta_{Etrim}, \delta_{Rtrim}$	0,-3.5356,0	0,-1.7,0	0,-2.5,0	0,-2.3753,0
V_s	22.7	22.36	22	21.9
w_{deflect}	1.5229	1.4591	1.4375	1.4763
C_m	0	0	0	0
L/D	48.8	49.28	49.7	49.9188
m_{wing}	112.7	119	115.9	119.99

A careful review of Table 4.4 identifies the main changes the optimizer made to the designs during the optimization. It becomes apparent that with a population of 100 designs the optimizer was able to find a reasonably well performing design in the first generation; however, a review of the constraint limits in Table 4.1 reveals that the optimal design in generation 1 was in violation of two constraints. Both the minimum stall speed and the maximum wing tip deflection constraints were violated. This was mainly a result of the designs heavily tapered wing which forced a very thin internal wing box structure as well as a smaller wing area. Over the next two generations the optimizer was able to reduce the wing tip deflection back into the feasible region by making a number of changes. One such change was a shift of

the wing area distribution by moving the joint between the spans b_1 and b_2 inwards by nearly 5%. This effectively shifted wing area away from the wingtip and thus reduced the lift generated by the outer wing section. The wing taper was also relaxed allowing room for a larger wing box structure and a larger wing area. Another change was the forward shift of both the spars which, as shown by Figure 3.6 is where most of the aerodynamic load occurs and thus placed them in a better load bearing position. Lastly the thicknesses of both the spars and the top flange were increased to provide more rigidity to the structure, although the weight of the 3rd generation design did slightly increase as a result. The stall speed constraint violation was still a problem in the 3rd generation design; but the violation was extremely low and therefore did not have much impact in the form of a penalty to the objective function. By the 21st generation the optimal design was free of constraint violations. To eliminate the stall speed constraint violation the wing area was increased slightly and the wing weight was decreased which, based on the definition of the stall speed shown by equation 3.23 results in a lower stall speed. This design also showed a small decrease in the wing weight and a minor increase in the lift VS drag ratio. This can be attributed to the reduction of several of the wing box component thicknesses and very slight changes to the wing geometry and winglets. By the final design the objective function value was only able to increase very slightly to a final gliding sink rate of 0.5434m/s. Improvements beyond this could not be achieved as several of the constraints were on or very near their respective boundaries and the design had converged on its optimum glide speed in which the combination of induced and form drag reached its minimum. The wing weight of the final design increased slightly compared to the 21st generation however, the wing area also increased which negated any negative effect on the sink rate.

Table 4.5 summarizes the data comparison between the ASW and the final design obtained with the proposed optimization. Table 4.5 shows that in many aspects the optimized design correlates very well with the ASW having nearly equal wing geometry and configuration. The wing weights are also very similar however, the mass of internal components such as control linkage and control surface mounting hinges were estimated for the optimized design and could lead to some discrepancy. This likely contributed to the main discrepancy between the two designs which was the total sailplane weight at their best performance conditions. This difference is also due to the fuselage weight estimate used in this work, which was done conservatively due to a lack of reference data. Furthermore, the fuselage was not varied in the optimization and thus its weight was estimated as a fixed value. A review of equation 3.30 shows the impact of sailplane weight on the sink rate helping to explain the slightly higher sink rate achieved by the optimum design. A three principle view comparison of the two sailplanes is shown in Figure 4.3

Table 4.5 Sailplane Data comparison

Criteria	ASW 27 B	Optimized Design
Wing area (m ²)	9	8.9987
Wing aspect ratio	25	25.0037
Airfoil	DU 89-134/14	DU 89-134/14
Wing mass (kg)	116	119.99
Manoeuvre speed (km/h)	215	224.317
Mass for below performance specifications (kg)	320	329.99
Best L/D speed (km/h)	100	96.79
Max glide ratio	48	49.918
Minimum sink rate speed (km/h)	unknown	71.2
Minimum sink rate (m/s)	0.52	0.5434

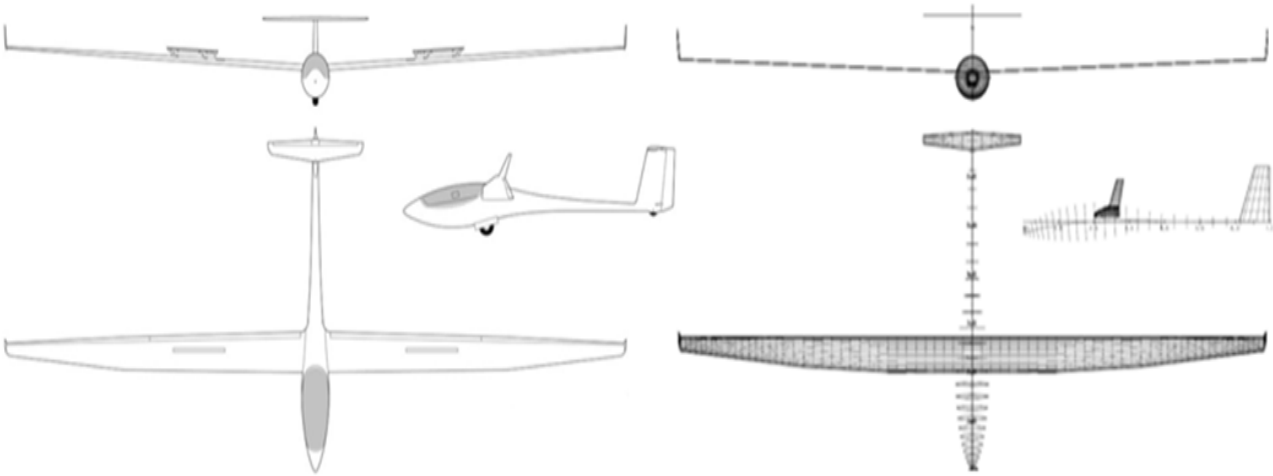


Figure 4.3 Sailplane layout comparison between the ASW 27 B [45] and the optimized design

Figures 4.4 and 4.5 show the Von Mises stress for the top and bottom of the wing box of the final design respectively. It can be observed that the stress is reasonably well distributed as a result of the exponential material thickness formulation. Although, since the objective of the optimization was focused on the gliding sink rate of the sailplane, the resulting wing box structure is not fully optimized. This is mainly identified by the maximum stress constraint which was far from the boundary in the final design. This was partially due to the impact of other constraints namely the wingtip deflection constraint which reached its limit deflection and thus forced the maximum stress to remain quite conservative. Another factor that contributed to this was the competitive nature of aerodynamic and structural objectives. Since this optimization was aimed at mainly aerodynamic efficiency, the optimizer favoured designs with long slender, high aspect ratio wings. This is very opposed to a structurally efficient design

which would have preferred a shorter, lower aspect ratio wing and could have provided a lighter structure with a more optimized stress distribution. This is evidence of the highly coupled disciplinary analysis involved in complex designs such as aircraft.

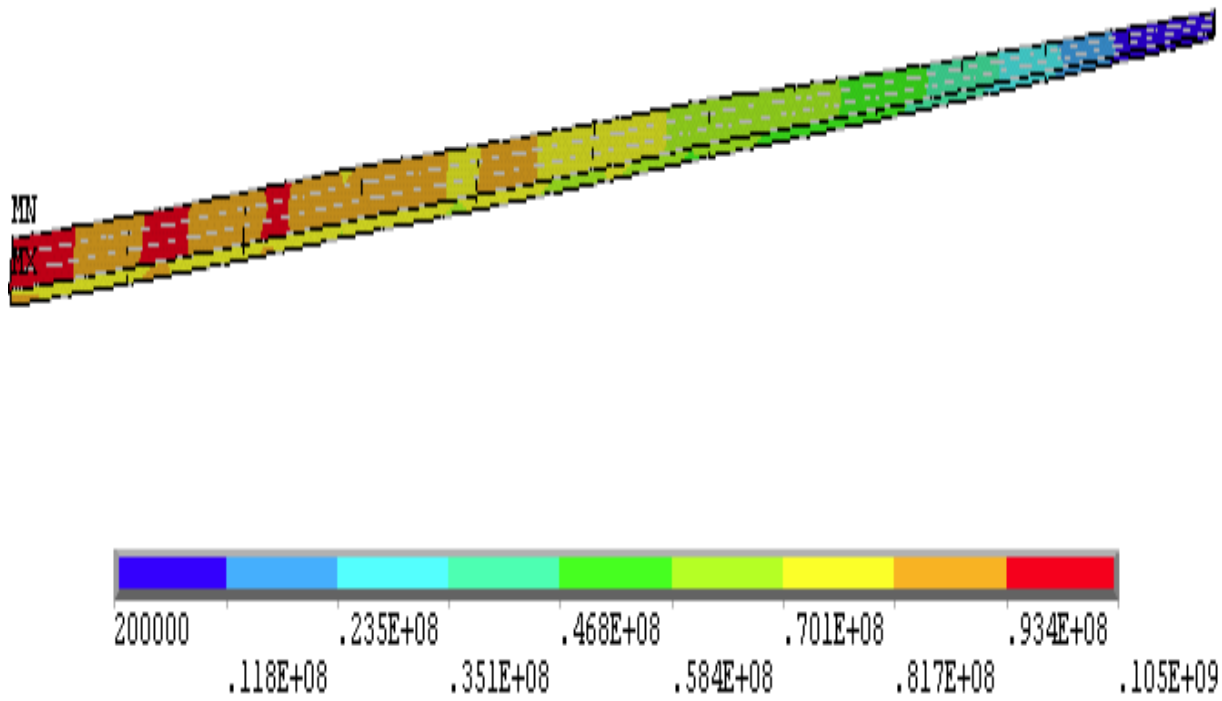


Figure 4.4 Von Mises Stress Distribution of Optimized Design (Top View)

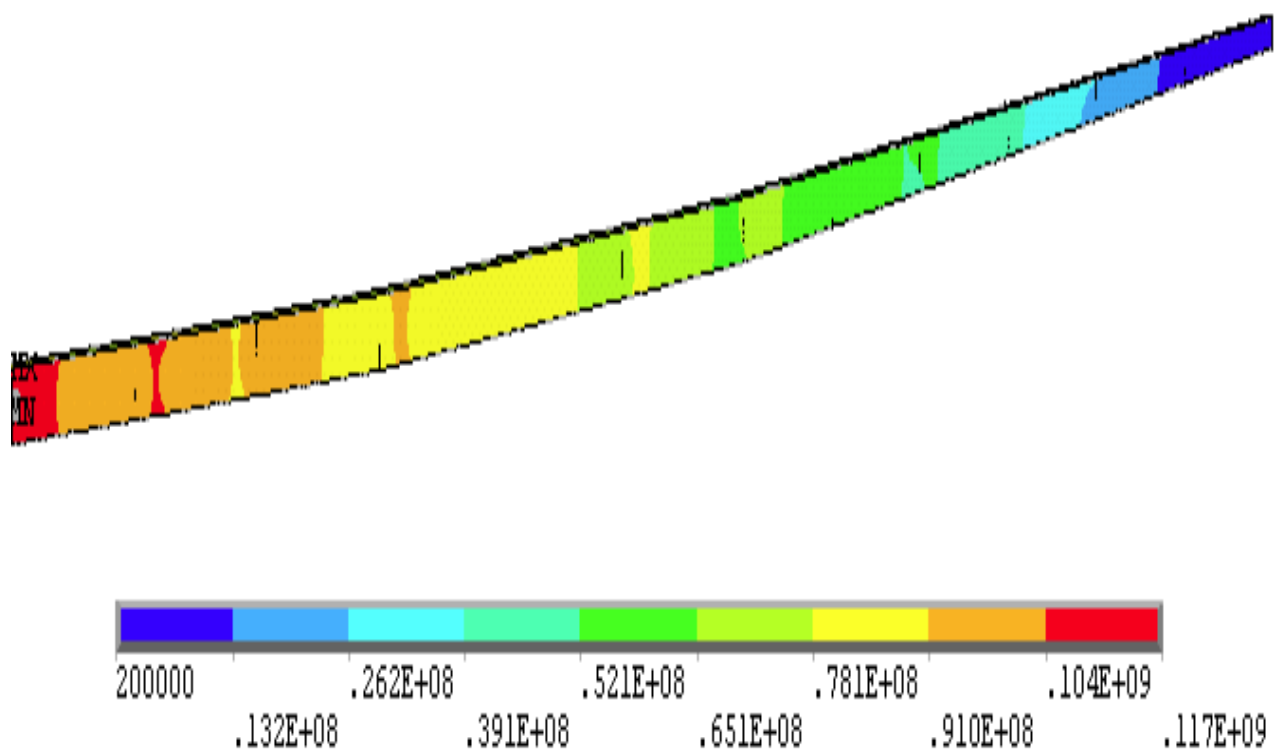


Figure 4.5 Von Mises Stress Distribution of Optimized Design (Bottom View)

Chapter 5

Conclusion

An MDO approach using multi-fidelity analysis methods has been applied to preliminary sailplane design. The proposed work was based on a non-planar vortex lattice solver for the aerodynamics and stability analysis; the commercial FEM solver ANSYS for structural analysis, and, empirical aircraft performance equations were used for performance and airworthiness calculations. Multi-fidelity analysis was used to include more detailed analysis where required, while using simple methods where applicable to manage computational costs. Specifically, this entailed the use of a high and a moderate fidelity method to accommodate for the aero-structural coupling of wing design, and, a low fidelity approach to the sailplane performance and airworthiness calculations. An automatic meshing algorithm was developed to adapt and re mesh the model to accommodate for changes of the wing and wing box configuration. The meshing algorithm was designed to maintain desirable element aspect ratios and thus an acceptable mesh quality for the entire possible range of designs. A modified genetic algorithm was developed based on components of the existing optimizer available in the MATLAB software package. The customized genetic algorithm was validated using well known analytical functions and, showed improved performance over the default MATLAB genetic algorithm. Furthermore, a series of case studies were performed to determine the best optimizer settings for the preliminary sailplane design problem. The objective of the optimization was to minimize the gliding sink rate of the sailplane while maintaining competitive performance in comparison to existing high performance sailplanes. The design was also subject to several airworthiness and stability constraints.

The results of the optimization showed that the application of MDO techniques along with multi-fidelity analysis methods to sailplane preliminary design is a viable and useful approach. When these methods are applied with performance and airworthiness constraints, it has been shown that practical preliminary sailplane designs are obtained. This work was intended to demonstrate that modern optimization techniques can be used with confidence in early sailplane design phases. These techniques have been successful in many applications and have the potential to benefit sailplane designers by providing a non-bias approach that can stretch outside traditional design trends. Finally, MDO methods

can also be used to quickly and affordably explore potentially beneficial designs that would be considered too expensive to investigate with traditional means which, could lead to many innovations in aerospace design.

The main contribution of this research was the expansion of existing work in sailplane optimization which, has predominately been focused on conceptual design and simplified analysis techniques. The existing work as discussed in Chapter 1 of this thesis identified the significant impact disciplinary coupling can have on sailplane design optimization. Furthermore high fidelity analysis methods have been successfully applied to focused aspects of sailplane design in more recent publications. This work aimed to explore the effectiveness of combining aspects of the previous work in sailplane design optimization by applying more detailed analysis to sensitive aspects of sailplane design while still considering the entire sailplane configuration at a preliminary design level. By using this approach the impact on the entire sailplane configuration can be observed when changes are applied to single aspects of the sailplane such as the wing design.

As briefly stated in the results section of this thesis, the wing box of the final design was not completely optimized in a structural point of view, however, this was considered in the separate thesis project mentioned in Chapter 1. A small utility class aircraft wing optimization was carried out using portions of the same software package that was developed for this thesis. This project focused only on the aerostuctural design of the wing and utilized the adaptive meshing routine discussed in Chapter 3 to analyze the aerodynamic and structural coupling involved in wing design. An existing wing design was provided by Found Aircraft Canada Inc, and the optimization showed promising results in reducing the wing box structural weight by optimizing the material thicknesses to better distribute the stress. The success of this separate project has demonstrated the abilities of the software used in this research when applied to different applications with different optimization objectives.

5.1 Future Work

Though the research conducted in this thesis was very successful, there is considerable room for improvements. For instance, the optimization framework could be expanded to include an option for selecting other optimization strategies, or, a variety of optimizer algorithms. This addition may identify a more efficient approach than the MDF and GA framework that is currently applied to the problem.

Another major task in this research was trying to find a compromise between analysis fidelity and computational effectiveness. This is a problem experienced in most complex optimization tasks as the benefits of the approach start to degrade with increasing implementation costs, whether it is from increased computational power requirements, or increased development and convergence times. However, as computational power increases, a future project could expand on this research by implementing high fidelity analysis methods for all disciplines. That being said, future research could push beyond the preliminary design stage and utilize the high fidelity analysis methods to optimize for detailed designs.

Bibliography

- [1] R E Perez, J Chung, and K Behdinan, "Aircraft conceptual design using genetic algorithms," in *AIAA Paper2000-4938, presented at the 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Bellevue, WA, 2000.
- [2] Xiaoqian Chen et al., "Research on Theory and Application of Multidisciplinary Design Optimization of Flight Vehicles," in *47th AIAA / ASME / ASCE / AHS / ASC Structures, Structural Dynamics, and Materials Conference*, Newport, Rhode Island, 2006.
- [3] R E Perez, H H T Liu, and K Behdinan, "Evaluation of multidisciplinary optimization approaches for aircraft conceptual design," in *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Albany, New York, Aug. 30-1, 2004.
- [4] Andy J Keane and Prasanth B Nair, *Computational Approaches for Aerospace Design the pursuit of excellence*. West Sussex, England: Wiley, 2005.
- [5] F Mastroddi and et. al, "On the use of geometry design variables in the MDO analysis of wing structures with aeroelastic constraints on stability and response," *Aerospace Science and Technology*, 2010.
- [6] Adel Younis and Zuomin Dong, "Trends, Features, and Tests of Common and Recently Introduced Global Optimization Methods," *Engineering Optimization*, vol. 42, no. 8, pp. 691-718, 2010.
- [7] Fred Thomas, "100 Years of Sailplane Design and Beyond," in *AIAA/ICAS International Air and Space Symposium and Exposition*, Dayton, Ohio, 2003, pp. 1-24.
- [8] Mark D Maughmer, "THE EVOLUTION OF SAILPLANE WING DESIGN," in *AIAA/ICAS International Air and Space Symposium and Exposition*, Dayton, Ohio, 2003.
- [9] Fédération Aéronautique Internationale. (2011) Fédération Aéronautique Internationale. [Online]. <http://www.fai.org/>
- [10] T. R. of MDO Within Aerospace Design and P. T. an MDO Capability Through European Collaboration, "Multidisciplinary techniques for commercial aircraft systems design," in *Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, St. Louis, MO, 1998.
- [11] Daniel Neufeld, Joon Chung, and Kamran Behdinan, "Aircraft Conceptual Design Optimization with Uncertain Contributing Analyses," in *AIAA Modeling and Simulation Technologies Conference*, Chicago, IL, 2009.
- [12] J P Giesign and J F M Barthelemy, "A Summary of Industry MDO Applications and Needs," in

Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, St. Louis, MO, 1998.

- [13] S Ricci and M Terraneo, "Application of MDO techniques to the preliminary design of morphed aircraft," in *11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Portsmouth, Virginia, 2006.
- [14] G Shi, G Renaud, X Yang, F Zhang, and S Chen, "Integrated Wing Design With Three Disciplines," in *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Atlanta, Georgia, 2002.
- [15] Ryan Peoples and Karen Willcox, "Value-Based Multidisciplinary Optimization for Commercial Aircraft Design and Business Risk Assessment," *Journal of Aircraft*, vol. 43, no. 4, pp. 913-921, 2006.
- [16] Joon Chung and Tea-Cheol Jung, "Optimization of an air cushion vehicle bag and finger skirt using genetic algorithms," *Aerospace Science and Technology*, vol. 8, p. 219-229, 2003.
- [17] Daniel J Neufeld and Chung Joon, "Unmanned Aerial Vehicle Conceptual Design Using a Genetic Algorithm and Data Mining," in *AIAA Infotech@Aerospace 2005 Conference and Exhibit*, Arlington, Virginia, 2005.
- [18] H Chung, S Choi, and J Alonso, "Supersonic business jet design using knowledge based genetic algorithm with adaptive unstructured grid methodology," in *Proceedings of the 21st AIAA Applied Aerodynamics Conference*, Orlando, FL, 2003, pp. 23-26.
- [19] Shinji Suzuki and Norihisa Kawamura, "Simultaneous Optimization of Sailplane Design and Its Flight Trajectory," *Journal of Aircraft*, vol. 33, no. 3, pp. 567-571, 1996.
- [20] B Grossman, Z Gurdal, G J Strauch, M Eppard, and R T Haftka, "Integrated Aerodynamic/Structural Design of a Sailplane Wing," *Journal of Aircraft*, vol. 25, no. 9, pp. 855-860, 1988.
- [21] Mark D Maughmer, "The Design of Winglets for High-Performance Sailplanes," *Journal of aircraft*, vol. 40, no. 6, pp. 1099-1106, 2003.
- [22] L M M Boermans, F Nicolosi, and K Kubrynski, "Aerodynamic design of high-performance sailplane wing-fuselage combinations," in *21st Congress of International Council of the Aeronautical Sciences*, Melbourne, Australia, 1998, pp. 13-18.
- [23] E Cramer, J Dennis, P Frank, R Lewis, and G Shubin, "Problem formulation for multidisciplinary optimization," *Problem formulation for multidisciplinary optimization*, vol. 4, no. 4, pp. 754-776, 1994.
- [24] P T Boggs and J W Tolle, "Sequential quadratic programming," *Acta Numerica*, vol. 4:1-51, 1995.

- [25] F Herrera, M Lozano, and J L Verdegay, "Tackling real-coded genetic algorithms: operators and tools for behavioural analysis," *Artificial Intelligence Review*, vol. 12, no. 4, pp. 265-319, 1998.
- [26] R J Balling and J Sobieszcanski-Sobieski, "Optimization of Coupled Systems: A Critical Overview of Approaches," Institute for Computer Applications in Science and Engineering, Hampton, VA, NASA Contractor Report 94-100, 1994.
- [27] K F Hulme and C L Bloebaum, "A COMPARISON OF SOLUTION STRATEGIES FOR SIMULATION-BASED MULTIDISCIPLINARY DESIGN OPTIMIZATION," in *Seventh AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, St. Louis, Missouri, NY, 1998, pp. 2143-2153.
- [28] Randy L Haupt and Sue Ellen Haupt, *PRACTICAL GENETIC ALGORITHMS*, 2nd ed. Hoboken, New Jersey: JOHN WILEY & SONS, INC., 2004.
- [29] J T Richardson, M R Palmer, G Liepins, and M Hilliard, "Some guidelines for genetic algorithms with penalty functions," in *Proceedings of the Third International Conference on Genetic Algorithms*, 1989, pp. 191-197.
- [30] Daniel Raymer, *Aircraft design: a conceptual approach*, 3rd ed. Reston, VA: American Institute of Aeronautics and Astronautics, 1999.
- [31] J Roskam, *Airplane design.*: DAR corporation, 2000.
- [32] Daniel Neufeld, Joon Chung, and Kamran Behdinan, "Development of a Flexible MDO Architecture for Aircraft Conceptual Design," in *EngOpt 2008 - International Conference on Engineering Optimization*, Rio de Janeiro, Brazil, 2008.
- [33] Daniel Neufeld, Kamran Behdinan, and Joon Chung, "Aircraft wing box optimization considering uncertainty in surrogate models," *Structural and Multidisciplinary Optimization*, vol. 42, pp. 745-753, 2010.
- [34] (2010) Athena vortex lattice 3.30. Software Package.
- [35] J Moran, *An Introduction to Theroetical and Computational Aerodynamics*, 2nd ed.: Dover Publications Inc, 2003.
- [36] J J Bertin and M L Smith, *Aerodynamics for Engineers*, 5th ed.: PrenticeHall, 2008.
- [37] J D Anderson, *Fundamentals of Aerodynamics*, 4th ed. New York: McGrawHill, 2007.
- [38] Performance Composites. (2011) Performance Composites Ltd. [Online]. http://www.performance-composites.com/carbonfibre/mechanicalproperties_2.asp

- [39] ANSYS Inc, *ANSYS Inc Theory Reference*, Peter Kohnke, Ed. Conosburg,PA, USA, 2004.
- [40] Patrick M Knupp, "Achieving finite element mesh quality via optimization of the Jacobian matrix norm and associated quantities Part I—a framework for surface mesh optimization," *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING*, no. 48, p. 401–420, 2000.
- [41] Transport Canada. (2010, June) Part V - Airworthiness Manual Chapter 523 - Normal, Utility, Aerobatic And Commuter Category Aeroplanes. [Online].
http://www.tc.gc.ca/eng/civilaviation/regserv/cars/part5-standards-523-sub-c-2062.htm#523_303
- [42] U.S. DEPARTMENT OF TRANSPORTATION, *Glider Flying Handbook*.: FEDERAL AVIATION ADMINISTRATION, 2003.
- [43] Robert C Nelson, *Flight Stability and Automatic Control*, 2nd ed. Boston: McGrawHill, 1998.
- [44] Fred Thomas, *Fundamentals of Sailplane Design*.: College Park Press, 1999.
- [45] (2011, Mar.) alexander schleicher sailplanes. [Online]. <http://www.alexander-schleicher.de/>
- [46] R C Hibbeler, *Mechanics of Materials*, 6th ed. Upper Saddle River, NJ: Pearson, Prentice Hall, 2005.
- [47] Alexander Schleicher. (2011) Alexander Schleicher Sailplane data. [Online]. <http://www.alexander-schleicher.de/>
- [48] Steven C Chapra, *Applied Numerical Methods with MATLAB for engineers and scientists*, 2nd ed. New York, NY: Mc Graw Hill, 2008.
- [49] Mathworks Inc. (2011, May 5th) Genetic Algorithm Options. [Online].
<http://www.mathworks.com/help/toolbox/gads/f6174dfi10.html>.
- [50] Daryl L Logan, *A first course in Finite Element Method*, 4th ed. Toronto: Thomson, 2007.

Appendix A

A.1 Genetic Algorithm Selection Method Case Study

A test was conducted to determine the best selection method for the sailplane optimization problem. For this test the Stochastic, Roulette and, Tournament selection methods were compared. The behaviour of each selection method was tested based on the results at the end of the first generation with a population of 400 designs. The initial population was generated using the Random Uniform method available in MATLAB, and, the ranking method was used for the scaling function as shown by the following equation:

$$\bar{f}_i = \left[\frac{\sum_{i=1}^{n_{pop}} f_i}{f_i n_{pop}} \right]^2 \quad (A.1)$$

Where \bar{f}_i is the scaled fitness value, also denoted as the expectation, f_i is the un-scaled fitness value which, is already penalized in the event of a boundary conditions violation and, n_{pop} is the size of the population. This scaling function is very well suited for initial generations, in which, the range between the best and worst performing individuals is greater than that of subsequent generations. Figure A.1 is a comparison of the Stochastic and Roulette methods, where, the black bars represent the initial population and the black dashed line indicates the fitted normal distribution for the initial population. The blue bars indicate the expectation of the parents selected for the next generation, and, the blue dashed line indicates its normal distribution. The figure shows that both the selection methods shifted the population slightly toward the higher expectation values, but, still included a large percentage of the poor performing designs as parents for the next generation. Figure A.2 was created from the same test conditions. It includes the Tournament method with a pool size of 4 compared to the Roulette method from Figure A.1. From this figure it can be seen that the Tournament method makes a significant shift toward the higher expectation values, meaning that it included a larger percentage of the best designs, while excluding most of the worst performing individuals. This aspect of the Tournament selection method is desired to increase the convergence speed; however, it consequently decreases the diversity

of the population. Therefore the tournament pool size should be selected carefully depending on the type of optimization problem.

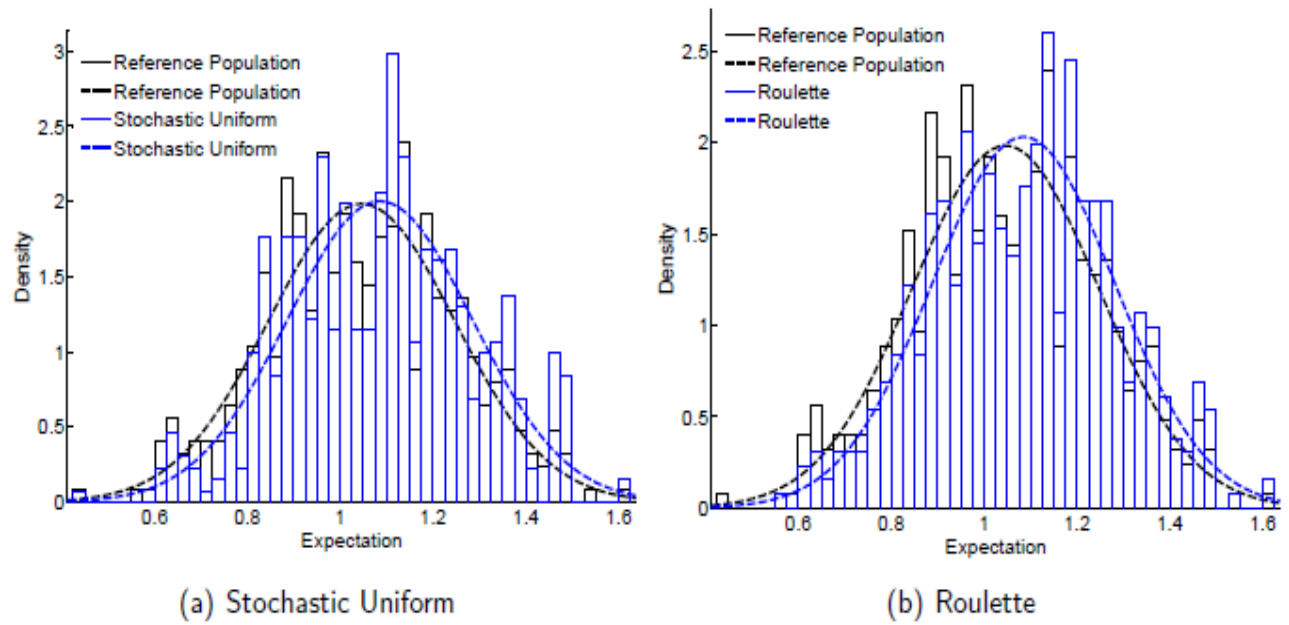


Figure A.1 Comparison of Stochastic and Roulette Selection Methods

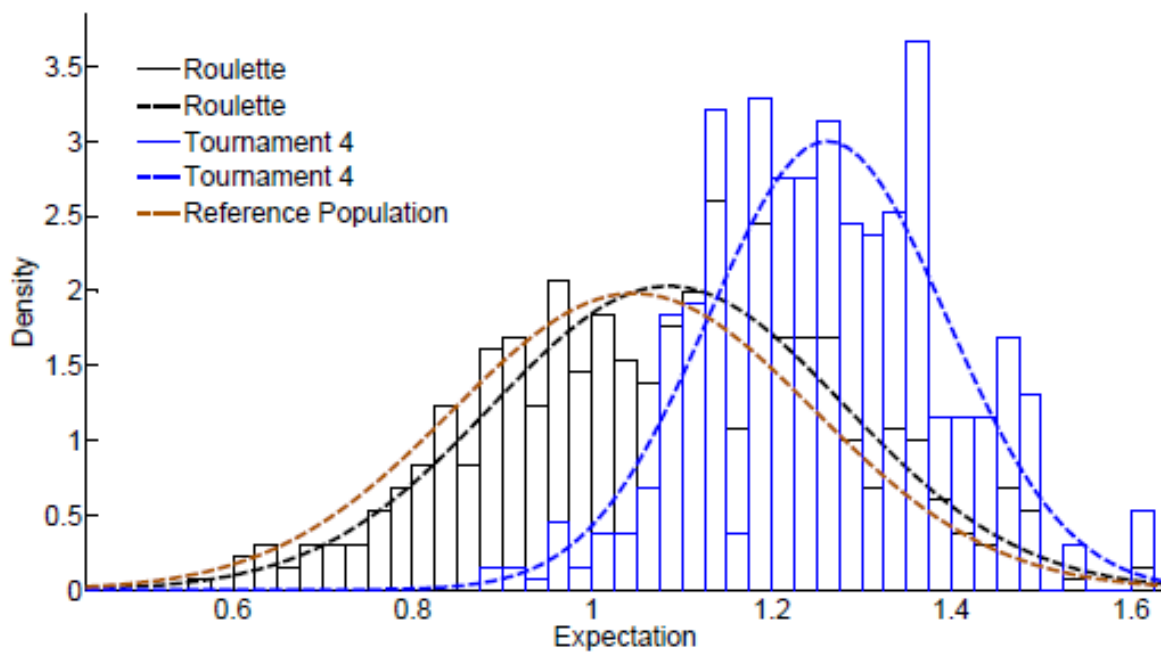


Figure A.2 Comparison of Tournament and Roulette Selection Methods

Figure A.3 shows a comparison of the Tournament method for a pool size of 4, which, is the MATLAB default setting, and, a pool size of 20. The results display the expected trend; with a larger pool size the optimizer has a greater probability of selecting well performing designs, thus eliminating the inferior designs. However, as stated above, increasing the pool size decreases the diversity of the population, which, can lead to premature convergence. For this reason, a compromise was made by setting the pool size to 10 in sailplane optimization, which, when coupled with a large population of 100 designs allowed adequate diversity and a reasonable convergence time.

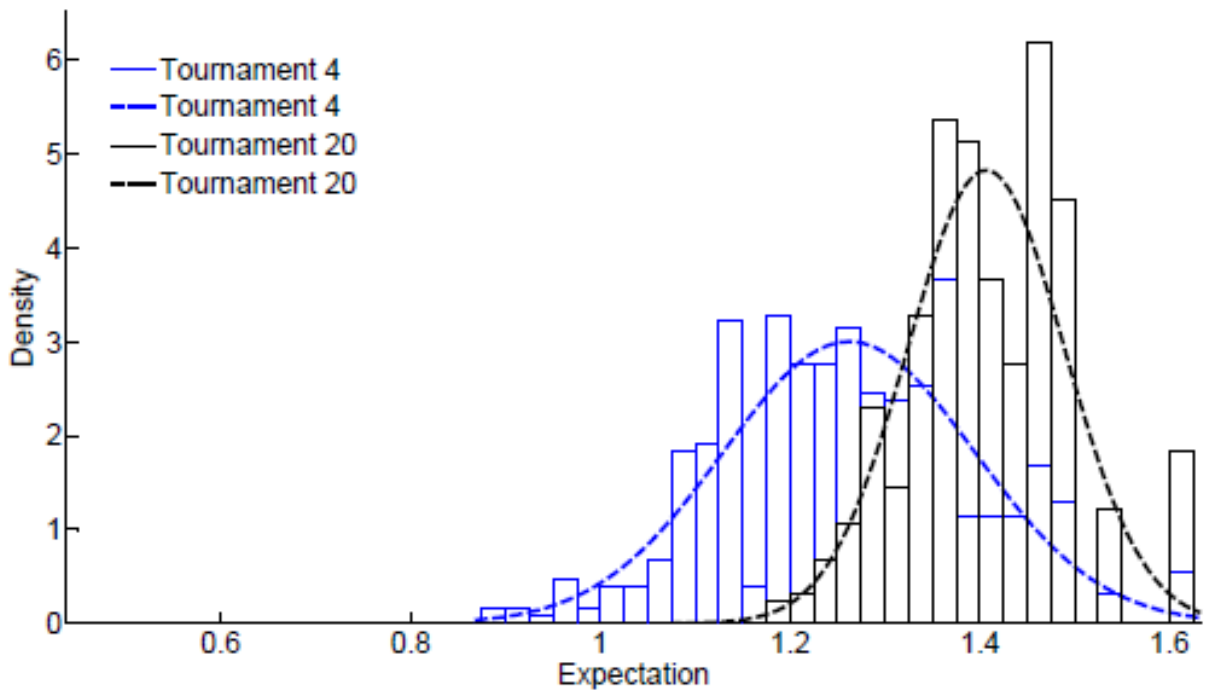


Figure A.3 Pool Size Comparison For The Tournament Selection Method

A.2 Modified Gaussian Mutation Function

```
function mutationChildren =
mutationgaussian1(parents,options,GenomeLength,FitnessFcn,state,thisScore,thi
sPopulation)
    scale=0.3;
    shrink =0.9;
    scale = scale - shrink * scale * state.Generation/options.Generations;
    scale = scale * (UB - LB);
    mutationChildren = zeros(length(parents),GenomeLength);
    for par = 1:length(parents)
        parent = thisPopulation(parents(par),:);
        for j=1:GenomeLength
            repeat = 1;
            while repeat == 1
                if options.discrete_v(j) == 0;
                    change = scale(j)*randn(1,1);
```

```

else
    change = round(scale(j)*randn(1,1));
end

if (LB(j)<=(change+parent(j)) && (change+parent(j))<=UB(j))
    repeat = 0;
end
end
mutationChildren(par,j) = parent(j) + change;
end
end
end

```

A.3 Structures Model Convergence Study

During the optimization the wing box model was free to change in size and shape. This made it impossible to use a consistent mesh. As discussed in Chapter 3, a dynamic meshing algorithm was developed to automatically mesh the models and ensure adequate mesh quality for each design despite the changing geometry. However, as further validation a simple wing box mesh was tested for convergence with increasing mesh density to determine a suitable element size to set as a target for the dynamic meshing routine. Figure A.4 shows the model used for the convergence test. A simple loading case was applied consisting of a nominal load distributed evenly between the nodes on the upper flange at the wingtip. The simple loading configuration was used to limit the computation time and remove the need for the aerodynamics solver. Only the stringers at the corners of the wing box were considered in the test due to the fact that the number of stringers in the structure changes depending on the number of nodes in the chord direction.

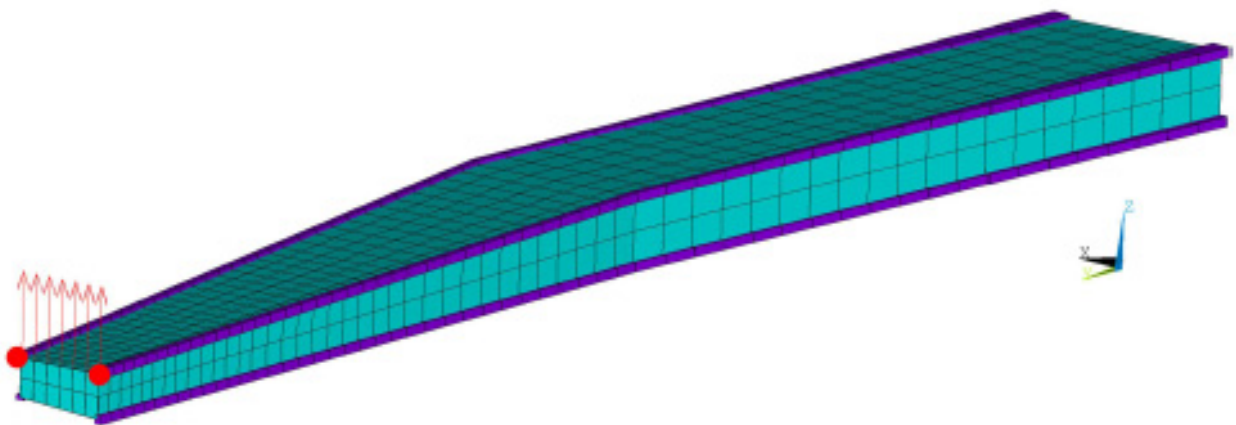


Figure A.4 Mesh Convergence Test Model

The wing tip deflection was monitored in the test by calculating the average of the deflections of the corner nodes on the top flange at the wingtip. Figure A.5 shows the response of the model for a changing total number of elements. It should be noted that the elements at the root rib are specified and the rest of the element sizes are determined based on the process described in Chapter 3.

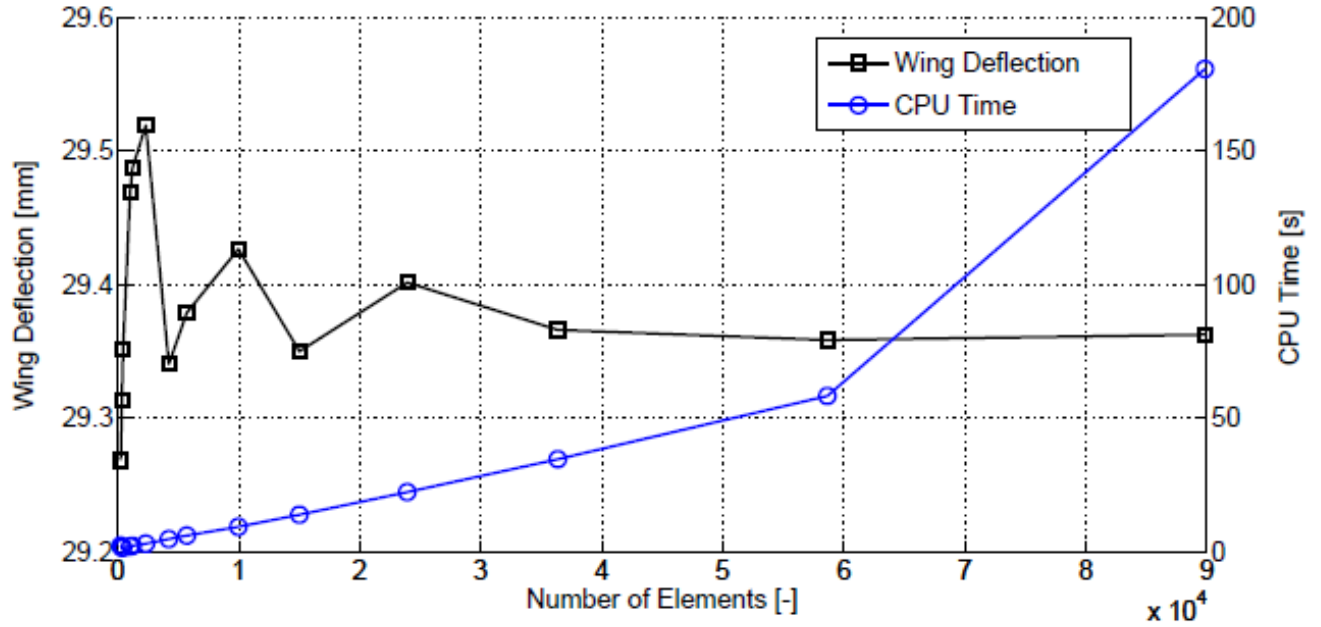


Figure A.5 Mesh Convergence Results

Figure A.5 includes root element sizes ranging from 0.01m to 0.2m and clearly shows convergence toward a wingtip deflection of approximately 29.36mm. For the larger element sizes the deflection has a maximum deviation of only 0.5%, demonstrating the effectiveness of the meshing processes ability to maintain suitable element quality for the tapering structure.

A.4 Stringer Description

As stated in Chapter 3, the ANSYS BEAM4 element was used to generate “Z” stringers to provide extra reinforcement for the wing box structure. Figure A.6 shows a diagram explaining the stringer dimensions, and, equations A.2 and A.3 were used to calculate the moments of inertia for the stringer cross section [46]. The web length is denoted by h , the flange width as f and, the thickness as t .

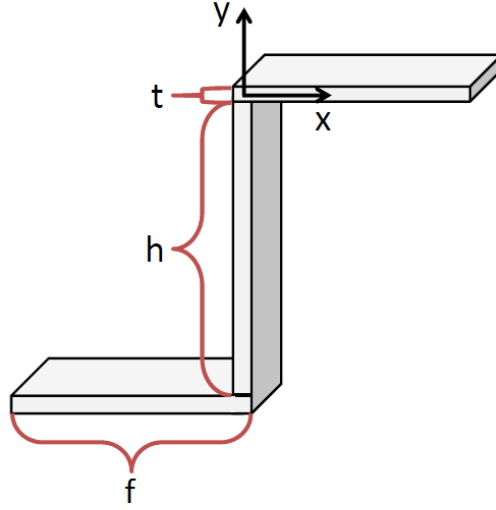


Figure A.6 Z Stringer

$$I_{xx} = \frac{th^3}{12} + 2 \left[\frac{t^3f}{12} + \left(\frac{h+t}{2} \right)^2 ft \right] + \left(\frac{h+t}{2} \right)^2 t(2f+h) \quad (A.2)$$

$$I_{yy} = \frac{ht^3}{12} + 2 \left[\frac{f^3t}{12} + \left(\frac{f-t}{2} \right)^2 ft \right] \quad (A.3)$$

A.5 ASW 27 B Information Sheet

Figure A.7 shows the performance data sheet for the ASW 27 B sailplane which served as the comparison design for this thesis. More information regarding the ASW 27 B can be found on the manufactures website [47].

ASW 27 B

CONTROL CIRCUITS AND FITTINGS

Aileron, elevator, flaps, and airbrakes are actuated by pushrods running in anti-noise ball-bearings, and use automatic connections at the assembly joints. The rudder is actuated by stainless steel cables which run in Polyamide tubing. Infinitely variable trim, lockable by a stick-mounted key. All control surface hinges of the wing and of the horizontal tail unit use needle bearings or low-maintenance plastic bearings. The actuating levers and bellcranks are fitted with ball bearings and precise "uniball"-joints. This provides the lowest possible actuating forces for the pilot and guarantees comfortable, non-fatiguing flying. The fittings are welded steel and milled or turned aluminum alloy respectively.

BOARD EQUIPMENT AND ACCESSORIES

Static pressure vents (for the A.S.I.) in the fuselage tail boom left and right. Pitot, static pressure and TE-compensation through 3-way-nozzle (multi-probe) in the fin. VHF antenna in the fin.

TECHNICAL DATA

Span incl. Winglets	15 m	49,21 ft	Mass of one wing	58 kg	128 lb
Wing area	9 m ²	96,88 sqft	Max. wing loading	55,56 kg/m ²	11,38 lb/sqft
Wing aspect ratio	25		Min. wing loading	≈34 kg/m ²	6,96 lb/sqft
Fuselage length	6,55 m	21,48 ft	Water ballast, max.	190 l	50,25 US gal
Cockpit height	0,80 m	2,62 ft	Useful load, max.	130 kg	286,7 lb
Cockpit width	0,64 m	2,1 ft	Useful load in the pilot seat, max.	115 kg	253,6 lb
Height at tailplane	1,3 m	4,26 ft	Max. speed	285 km/h	154 kts
Winglet height	0,45 m	1,50 ft	Maneuvering speed	215 km/h	116 kts
Wing airfoils root	DU 89-134/14 and				
tip	DU 89-134/14MOD				
Winglet airfoil	DU 94-086 M4				
Empty mass with min. equipment	245 kg	540 lb	For m = 320 kg (705 lb) flight mass:		
Empty mass "SL", min. equipment	230 kg	507 lb	Min. speed	70 km/h	38 kts
Max. take-off mass	500 kg	1102 lb	Min. sink	0,52 m/s	102,4 ft/min
			Best glide ratio, L/D (100 km/h)	48	

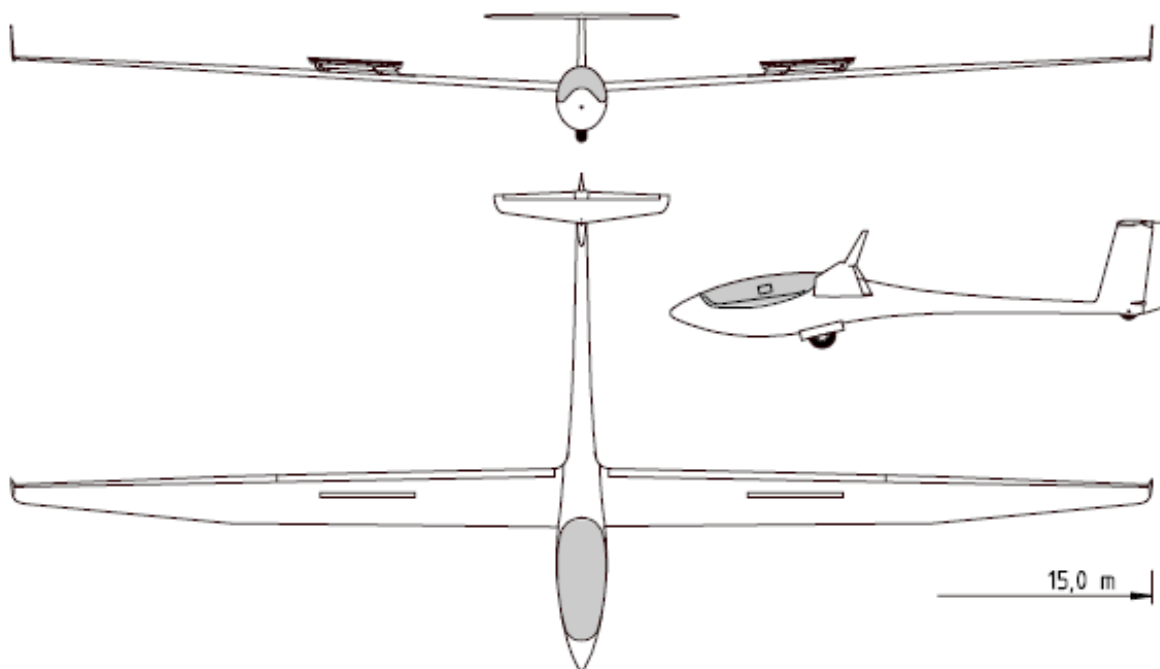


Figure A.7 ASW 27 B Data Sheet [47]