2. MULTIDISCIPLINARY DESIGN OPTIMISATION

Modern engineering problems need to be solved with a systematic approach. These problems also require investigations into multiple physics branches, each with its own objective, to accurately describe the overall system response. A solution to this problem is through the use of Multidisciplinary Design Optimisation (MDO). This method has matured in recent years and is now classified as a separate stream within the optimisation field and can be found in journals, specialised technical briefings, conferences and other publications.

The benefits of using an MDO scheme are evident given that a small increase in vehicle performance can have far reaching implications for the operation and life cycle of the vehicle. This is evident when an MDO scheme is able to reduce a vehicle's structural mass without altering the vehicles performance or maintenance lifecycle. This reduction in structural mass can be re-added to the vehicle as additional payload, thus increasing the usefulness of the vehicle. Furthermore, an evaluation of the constraints on multiple disciplines can be achieved effectively through the use of MDO and can reduce the cost of system re-design if required.

Sobieski [18], performed a comprehensive survey of MDO methods detailing their development and limitations. Of interest was the highlighting of certain needs namely multi-platform operation, parallel computation and space visualisation. Parallel computation would reduce the computational time, and space visualisation allows the user the ability to understand the performance of the generated design in the region near, and not just on, the design point. Both points address the challenge of computational cost from one side or the other, while the problem of organisational complexity associated with Evolutionary Optimisers is still being investigated. The optimisation algorithms surveyed by Sobieski mostly made use of traditional methods, the limitations of which have been discussed in the previous chapter.

The work of Giesing and Barthelem [19] extends Sobieski's survey into the field of industrial optimisation applications. In these applications, more interest is usually placed on the detailed design of components and systems instead of the conceptual or preliminary MDO approaches investigated by Sobieski. Giesing and Barthelem go on to state that MDO frameworks should be flexible allowing them to adapt to the user requirements and that ways need to be found to increase the computational efficiency without compromising the optimisation fidelity. Giesing and Barthelem also discuss their support for design space search methods such as EAs in

MDO frameworks. In their view this is because EAs are easy to implement and don't suffer from the same limitations of traditional optimisation methods.

A problem associated with MDO studies becomes evident when commercial high-fidelity processes are included in the solution sequence. Most software packages require complex user input at each code iteration making the packages unsuitable for automated operation. This user input requirement can be further exacerbated if the code is not robust. Common software packages which exhibit such tendencies are CFD and FEA packages. It then may become necessary to use other approximation techniques such as Response Surface Modelling (RSM) [6] or Design and Analysis of Computer Experiments (DACE) [20] in such cases as these techniques can reduce the overall computational expense. They can also be used when the analysis packages can not be directly integrated within the MDO architecture.

As most current MDO architectures are developed in university environments where access to large commercial codes is restricted, there is a tendency for the MDO codes to be specialised and hence difficult to configure for other applications. This is in direct contrast to current trends in requirements for higher flexibility, robustness and efficiency.

An example of a multi-objective MDO problem is the problem of wing conceptual design investigated in this thesis. The two objective functions are the minimisation of wing mass and the maximisation of the Lift to Drag (L/D) ratio. Each of the objective functions is from a different discipline namely structures and aerodynamics respectively. The design variables are the aerofoil shapes at different locations along the wing and the internal geometry describing the wing box. The optimiser makes use of an integrated approach where the variables are passed to an integral set of equations that are solved for the aerodynamic and structural properties. The optimiser then iterates improving the design subject to the two objective functions.

The overall approach is simple. Once the two disciplines are coupled into a single analysis module, the system can be treated in the same manner as a single discipline optimisation and can be solved using a parallel cluster of computers. The disadvantage of this approach becomes apparent when there is a requirement to solve the components making up each discipline individually on different machines.

2.1 Limitations of Current Optimisation Techniques for MDO

As the survey of Sobieski [18] showed, traditional deterministic optimisation techniques are the most common method used for MDO. These techniques are effective when the solution landscape is smooth and approaching uni-modal. Once ridges, local minima or noise is introduced into the solution landscape, traditional methods exhibit a decreased effectiveness. Guinta [6] showed that the presence of noise causes the deterministic optimiser to incorrectly calculate the gradients of the objective functions and this inaccuracy can lead to divergence in the solution.

In the application of MDO to the problem of conceptual wing design explored in this thesis, numerical noise is of concern. Both the solutions to the CFD and FEA simulations have noise as the solution methods are iterative and are only seen to converge once the residuals within each program have decreased below a program set minimum. This minimum residual value is normally many orders of magnitude greater than the machine zero (*eps*) value for the computer upon which the simulation is running to limit the computation expense of the simulation. Therefore, the statistical error on a portion of the solution might be large, as in the calculation of induced drag with PanAir, but still be within engineering limits. To a deterministic optimiser this error would be seen as destabilising noise and would slow or even hinder the calculation of the global minima.

It is important to note that EAs will be restricted to conceptual MDO studies within this thesis as the number of variables being investigated are less than 100. Once the number of variables increases beyond 100, EAs alone may not be of benefit but may need to be coupled with other methods such as Design of Experiments (DOE).

2.2 Formulation of the MDO Problem

The general MDO problem is formulated to seek a vector X of design variables that minimises an objective function $F(X)$ subject to disciplinary, interdisciplinary and side constraints. Furthermore $(X_i)_L \leq X_i \leq (X_i)_U$, $i = 1,\ldots,N$ where N is the number of design variables and $(X_i)_L$ and $(X_i)_U$ are the lower and upper bounds placed upon the design variable X_i.

Two different classes of design variables are specified:

- Aerodynamics
- Structures

This specific MDO application makes use of 67 variables with upper and lower bounds. These design variables describe both the internal and external wing geometry along with specific flight conditions such as angle of attack. For a complete description of the variables and constraints please refer to Chapters 6 and 7.

2.2.1 METHOD

The method couples the proven Evolutionary Optimiser (HAPMOEA) and an Aero-Structural solver. The Aero-Structural solver integrates two commercial high fidelity analysis tools for FEA and CFD analysis using MATLAB as the controlling operating system.

The optimisation process consists of the eight main steps as illustrated in Figure 2. In the first step, the design variables, design constraints, flow conditions and fitness functions are defined. The second step consists of generating an initial population of wing geometries at random. While the stopping condition has not been reached, the optimiser generates new candidate geometries in steps three and four respectively. In step five, the aero-structural solver evaluates each candidate geometry. These analyses provide the necessary information to compute the fitness function (or functions) in step six. If the problem is multi-objective the optimiser computes the Pareto fronts in step seven. The optimisation terminates if the stopping condition has been met in step eight.

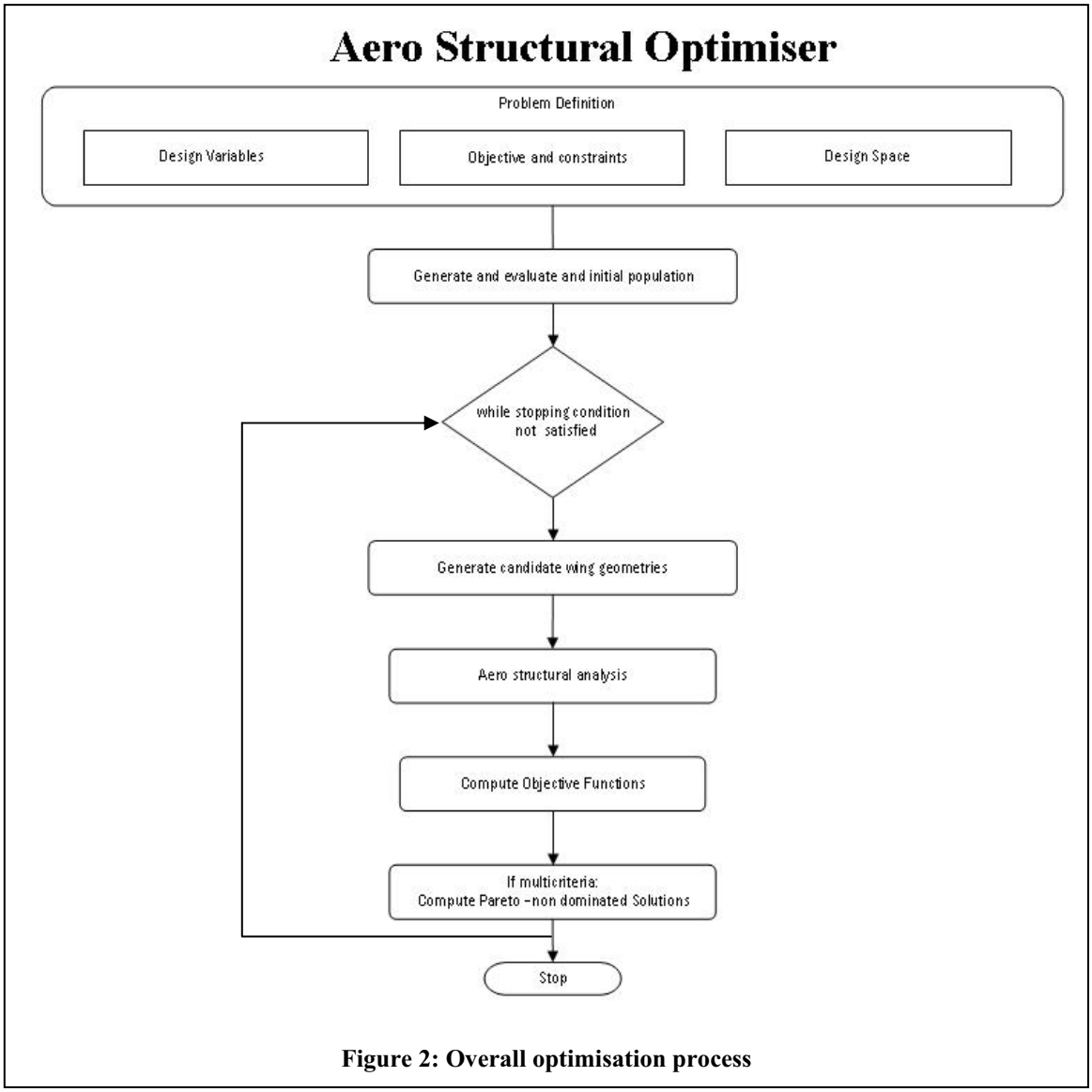


Figure 2: Overall optimisation process

The MDO process can be modularised as shown in Figure 2 allowing purpose built packages to interact and solve certain portions of each design step. This interaction gives the program the ability to consider different disciplines simultaneously reducing the number of design iterations required, and hence lowering the overall cost of the project.

3. EVOLUTIONARY OPTIMISER

Evolutionary optimisers are relatively new optimisation algorithms. This chapter describes the fundamentals behind the algorithm, the development of the process and finally the HAPMOEA code is explained.

3.1 EA FUNDAMENTALS

Evolutionary optimisers are relatively new design and optimisation algorithms that model the Darwinian theory of 'survival of the fittest' within a generation of candidate solutions. In more general terms, EAs function through the iterated mapping of solutions in a generational framework. Traditional deterministic methods such as the simplex or conjugate gradient method work by improving the calculated candidate solution from one point in the design search space to the next following the objective function's gradient. In contrast, EAs are not deterministic and do not calculate any gradients within the algorithm but are rather stochastic and follow a random path from a sub-optimal point within the design space to the global minima. As the process is random if an EA is run twice with the same initialisation commands and environment the two optimisations will not follow the same routes through the design space to the global minima. EAs are not the only stochastic optimisation algorithms available but sit alongside directed random walk, simulated annealing and Monte-Carlo approaches to name but a few.

The evolution of candidate solutions within EAs are not totally random otherwise the performance of the algorithm would be no better than taking guesses at the solution. EAs exploit population statistics where certain offspring from a parent population will exhibit superior characteristics over their parents, and some inferior. The main EA operation of iterated mapping to find a subsequent population between generations is simplified to the generation and evaluation of a candidate offspring population from its parents and then the removal of a preset number of inferior individuals. Goldberg [21] summarised the above procedure as the application of a variation and then a selection operator. The first operator produces a generation of offspring and the second mimics the 'survival of the fittest'.

Evolutionary Algorithms seem to have appeared in two separate fields, Genetic Algorithms (GAs) and Evolutionary Strategies (ESs). GAs were established as early as 1975 by Holland [22] but did not gain wide acceptance in the optimisation fraternity. The research of Goldberg provided the turning point for GAs which are now firmly accepted into the optimisation

community and as a consequence, the GA has been extensively researched and successfully applied to modern problems. The technique established by Holland worked by the binary coding of variables making up the genetic material in the same manner as chromosomes operate, namely through the use of a base two string. The variation operator identified by Goldberg can be then broken into two operations, crossover and mutation. Mutation influences local sections of the chromosome by randomly swapping letters in the chromosome. Crossover operates globally over the chromosome, switching portions of chromosomes between parents to generate offspring. The most simple selection operator identified by Goldberg is the roulette wheel otherwise known as the stochastic fitness-proportionate method. Many other selection operators exist and are used.

Evolutionary Strategies (ESs) are a far younger area of research having only emerged through the research of Back and Rudolph from the Technical University of Berlin [23]. This first ES worked with only one parent and one offspring. The individuals in the optimisation were real coded where each problem variable was assigned a floating-point value in the chromosome. A candidate offspring was generated through the application of the variation operator that applied random alterations to each problem variable describing the parent's chromosome. The selection operator then simply determined which individual, the parent or offspring, had better genetic information and that individual was used as the parent for the next iteration. This method is known as the (1+1) ES where one parent and one offspring make the working population. The '+' indicates that the selection operator is evaluated over both the parent and offspring.

The (1+1) ES was extended by Back and Schwefel [24] to the (μ+1) ES. This development introduced populations of solutions in each generation. In its current form, the variation operator is applied to μ parents to produce one offspring. The selection operator then determines whether or not the new offspring has a higher fitness than the worst member currently residing in μ^2 and if so, replaces that member. Deterministic control of the amount of mutation applied in the variation operation is used in both the (1+1) and (μ+1) ES.

Figure 3 illustrates a canonical evolutionary strategy. In this form, the population is initialised and evaluated. While the stopping criteria are not met, the code iterates the variation (mutation and recombination) and selection operators over the parents and offspring comprising each calculated generation. The stopping criteria can be the maximum number of iterations or the specification of a target objective function value.

Nowadays the distinction between GAs and ESs has diminished due to the modifications of their respective variation and selection operators. This has progressed to a point where a fine line cannot be drawn between GAs and ESs anymore. The final main difference between GAs and ESs in modern algorithms is the predominance of adaptive mutation used in ESs though GAs have produced related methods [25-27]. This mutation method makes ESs very appealing for real coded optimisation problems.

Initialise: $init(\mu_o)$

Evaluate: $f(\mu_o)$

g=0

while stopping condition not met,

Recombine: $\lambda_R^{g+1} = reco(\mu^g)$

Mutate: $\lambda_M^{g+1} = mut(\lambda_R^{g+1})$

Evaluate: $\lambda^{g+1} = f(\lambda_M^{g+1})$

Select: $\mu^{g+1} = sel(\mu + \lambda)$ (Plus strategy) or,

$\mu^{g+1} = sel(\lambda)$ (Comma strategy)

g=g+1

loop

Figure 3: Canonical Evolution Strategy

Numerous studies have looked at the application and capabilities of EAs at finding global optima when applied to aeronautical problems. EAs have been applied to the design of full aircraft, wings in isolation and rotor blades in both single discipline and MDO frameworks to name but a few studies [13, 17, 28, 29].

Other techniques for evolutionary optimisation are in existence besides GAs and ESs such as Evolutionary Programming (EP) and Genetic Programming (GP) but have not found widespread use in real coded applications. The work performed by Koza [30] highlights where EPs can be used in a real coded applications and how GPs are used mainly in the generation of computer programs as they do not make use of number sets as solutions to the objective function. For a detailed comparison between ESs and EPs the reader is directed to Back [23].

3.2 The Development of Evolutionary Algorithms for Design and Optimisation in Aeronautics

The potential benefits associated with using EAs over traditional optimisation techniques when solving engineering problems has been known for some time. Studies have been performed on wing optimisation problems and the potential benefits of EAs have been identified [13, 14, 31].

Obayashi [32] applied EAs to several wing planform problems including subsonic and transonic designs. Different variance and selection methods were evaluated in a Multi Objective Genetic Algorithm (MOGA) with the final goal being the calculation of non dominated Pareto fronts. In the transonic wing application, three objective functions were constructed and evaluated. These were the minimisation of aerodynamic drag and aircraft structural weight and the maximisation of the amount of fuel stored in the wing. Two constraints were imposed on the optimisation: the lift generated by the wings must be greater than the structural mass of the vehicle, and the structural strength must be greater than the applied aerodynamic loads.

Anderson [33], Takahashi [34] and Gonzalez [35] have also applied unique EAs for wing design. Gonzalez, for example, applied EAs to study the effect of a hierarchical topology of low fidelity analysis tools on conceptual wing design. This research was also performed in a multi-objective MDO framework and the results show the benefits of using variable fidelity solvers in problems with two or more objective functions.

Conceptual wing design has also been undertaken with GAs. Crispin [36] applied GAs to aircraft conceptual design while Crossley [37] applied GAs to helicopters. In both cases the use of GAs indicated the effectiveness of their use at finding globally optimal and feasible solutions. Crossley also showed that the use of parametric variations within GAs can significantly reduce the overall cost of a project.

Work on aircraft conceptual design has also been performed by Ruben [38] and Ali and Behdinan [39]. In both cases, the use of evolutionary optimisers helped to reduce the expected design time. Ruben was able to show that though the use of an evolutionary optimiser on the specified aircraft design yielded a weight reduction of five percent from the initial starting configuration. Ali and Behdinan investigated the use of evolutionary optimisers on medium-sized transport aircraft.

Parmee and Watson [15] made use of co-evolutionary multi-objective GAs to perform a preliminary airframe design. The results from their study indicate that through the use of smaller population sizes, quicker and less detailed optimisations can be achieved. Furthermore, as the number of objective functions increase, so the role of on-line sensitivity analysis increases.

Traditional transport and commercial aircraft have been investigated and optimised using EAs along with Unmanned Aerial Vehicles (UAVs) [35, 40] and Micro Aerial Vehicles (MAC) [41]. These studies included the investigation of large design spaces, novel and non-notional configurations. Cvetkovic [42] and Raymer [16, 43] applied EAs to other aircraft applications.

The above work indicates a growing interest in EAs for complex design and optimisation. Not only are EAs increasing in robustness, but new and exciting techniques to reduce the computational cost of each candidate solution are being researched and implemented.

Little research has been conducted in comparing the application of EAs against other methods in an MDO framework. Raymer and Crossley [16, 43] compared different optimisation methods; Monte Carlo, random walk, simulated annealing and evolutionary algorithms along with a steepest descent search in an attempt to enhance an aircraft conceptual design. The different methods were applied to four different classes of aircraft; a fighter, a commercial airliner, an asymmetric light twin and a tactical UAV. The algorithms were limited to the minimisation of one objective function and only used a single fidelity model. The number of variables investigated was set at seven. The results of Raymer's work show that all the methods computed solutions that improved the initial designs of the aircraft. Interestingly the steepest descent search method produced slightly better results over the same time frame in comparison with the other optimisation methods. This is due to the small number of variables investigated and hence if this value was increased, the other methods investigated would show better performance.

3.3 Advantages and Limitations of Traditional EAs for Aeronautical Problems

Although the advantages for using EAs over traditional deterministic optimisation methods when applied to complex problems are plain to see, EAs have not found extensive use in the

field of engineering. The reason for this is that most of the engineering problems investigated to date have had smooth objective functions allowing deterministic optimisation methods the ability to accurately calculate gradients and hence require fewer objective function evaluations.

This issue defined above has necessitated new efficient and robust techniques for the application of EAs. One such algorithm to address this issue is the Hierarchical Asynchronous Parallel Multi-objective Evolutionary Algorithm (HAPMOEA) described in the next section.

3.4 HIERARCHICAL ASYNCHRONOUS PARALLEL MULTI-OBJECTIVE EVOLUTIONARY ALGORITHMS

An advantage of standard EAs over deterministic optimisers is that they are highly efficient at finding the location of the global optima within the design space. Once found though, deterministic optimisers are able to converge on the solution in far fewer objective function evaluations. Numerous researchers have identified this disadvantage associated with EAs and hence many schemes have been developed to overcome this disadvantage. One such method which reduces the time taken for an EA to converge on global optima is the Hierarchical Asynchronous Parallel Multi Objective Evolutionary Algorithm (HAPMOEA) [17, 35, 44]. This algorithm has been shown to be robust and highly efficient.

The algorithm is built upon traditional evolutionary strategies, namely covariance matrix adaptation (CMA) [45], asynchronous candidate evaluation, parallel implementation, a hierarchical topology of fidelity methods, Pareto tournament selection, constraint handling mechanisms and single or multi-objective optimisation.

3.4.1 MULTI-CRITERIA OPTIMISATION

Aeronautical optimisation problems usually involve objective functions which have multiple conflicting objective functions, an example of which is solved in this thesis namely the trade-off between aerodynamics and structures. As the solution then becomes multi-dimensional, the solution to the problem is a non-dominated Pareto front. An example of a two dimensional (two objective function) problem is shown in Figure 4.

Here the problem can be formulated as

$$\textit{Maximise / Minimise}$$
$$f_i(x)_i = 1,.....,N \tag{1}$$

Subject to constraints:

$$g_j(x) = 0 \quad j = 1,.....,M \tag{2}$$
$$h_k(x) \leq 0 \quad k = 1,.....,M$$

Where f_i are the objective functions in Eqn 1, N is the number of objectives; x is an n – dimensional vector where its arguments are the decision variables. For a minimisation problem, a vector x_1 is said partially less than vector x_2 if according to Eqn 3:

$$\forall_i f_i(x_1) \leq f_i(x_2) \text{ and } \exists_i f_i(x_1) \leq f_i(x_2) \tag{3}$$

In this case the solution x_1 dominates the solution x_2.

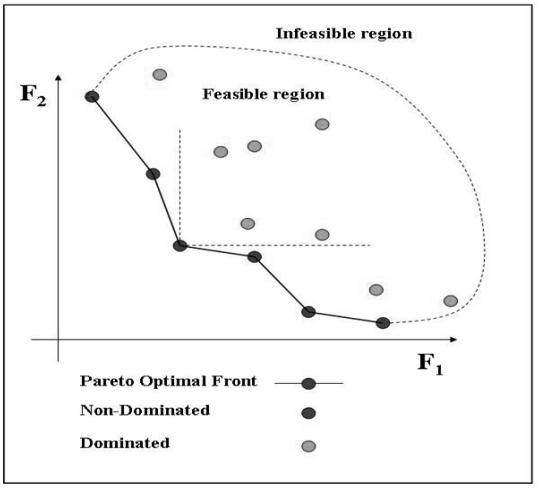


Figure 4: Pareto Optimality

The results shown in Figure 4 represent a trade-off between the two objective functions. Pareto selection is used to determine which candidate solutions lie along the non-dominated Pareto optimal front. In Figure 4 the user has the ability to select a solution along the Pareto front which can favour an objective function, a candidate solution which lies closest to the $\mathbf{F_1}$

axis favours the $\mathbf{F_2}$ objective function and *visa versa*, or both as is the case of a solution equidistant from both axes.

Extensive research has been performed and reported on Multi-Objective Evolutionary Algorithms (MOEAs). Texts include the works of Deb [8] and Coello-Coello [7].

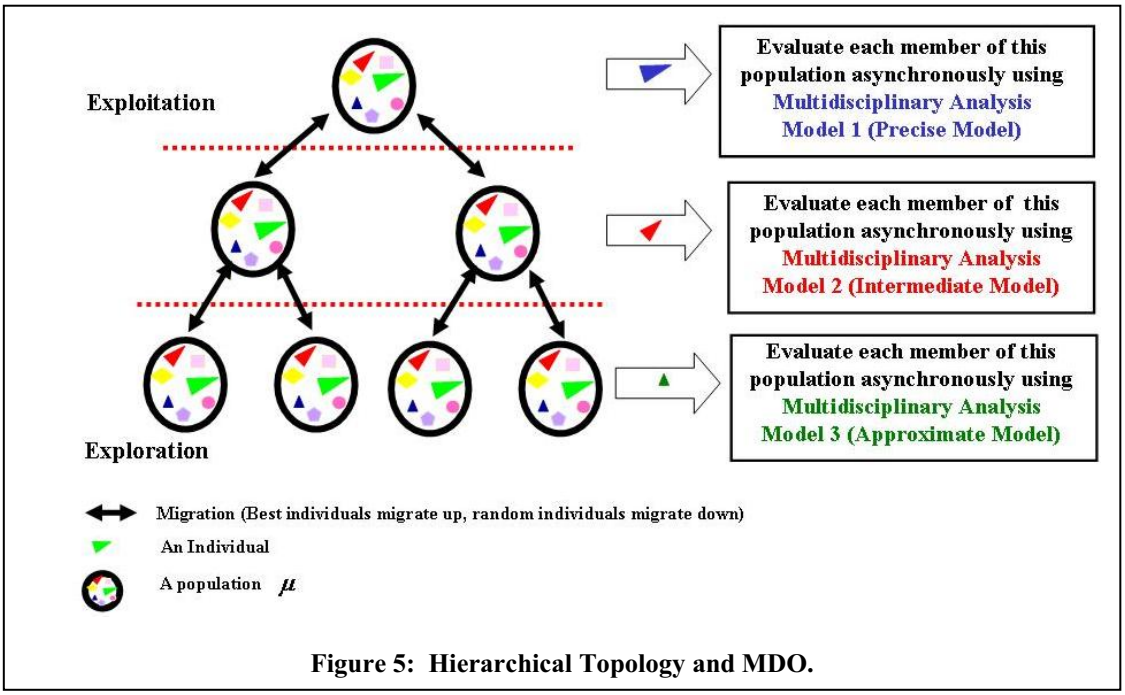


Figure 5: Hierarchical Topology and MDO.

3.4.2 HIERARCHICAL POPULATION TOPOLOGY

A typical hierarchical population topology is shown in Figure 5. In this Figure populations are given set objectives along with the overall minimization of the objective function. This concept was proposed by Sefrioui [46] where of the seven populations shown and evolved in Figure 5, the lowest four populations are dedicated to the efficient exploration of the design space. This is accomplished by using low-fidelity models and large variation operators on the parents within each population. The second level makes use of a higher fidelity model with smaller variation operators and refines the best solutions determined by the lowest level. The top level only consists of one population. This population is created from the best individuals making the middle populations. The models used within this top level have the highest fidelity and the smallest variation operators are used to finely tweak the candidate parameters.

This division of populations into set levels assists in the efficient exploration of the design space, and the optimisation of promising solutions. This is in contrast to many non-hierarchical methods that employ a one-population-cure-all approach.

3.4.3 Parallelisation Of The Algorithm

The HAPMOEA algorithm is similar in concept to the hybrid parallel Multi-Objective Evolutionary Algorithms (pMOEA) used by Cantu-Paz [47] and Veldhuizen [48]. This concept uses a master-slave arrangement on a cluster of personal computers. The master node focuses on the solution of the optimisation problem and passes the candidate solution variables off to remote nodes for evaluation. This passing of information is facilitated through the use of Parallel-Virtual-Machine (PVM) [49] and allows the remote nodes the ability to return the calculated objective function values to the master node for evaluation. A binary tree structure (similar in form to the hierarchical topology shown in Figure 5) is used to identify within which level each population exists and isolates the different simulation parameters.

The algorithm has been tested on a large cluster of machines at the School of Aerospace, Mechanical and Mechatronic Engineering within the University of Sydney. The machines had different hardware configurations including varying RAM modules, CPU speeds and communications attributes. The maximum number of nodes that can be evaluated is 18.

3.4.4 Asynchronous Evaluation

Evolutionary algorithms have traditionally used a generational approach when evaluating populations. This has meant that all the candidate individuals generated during an EA program iteration and passed to slave nodes need to be evaluated before the optimiser can progress to the next iteration. When the computation expense required to evaluate each candidate solution is roughly constant, a generational approach can be used. A problem arises when the computational expense is related to internal influences within the objective function such as in CFD or FEA solutions. In CFD solutions for example, the time required to solve a problem can not be calculated *a priori*. Therefore if CFD methods are included in the analysis, geometries that produce simple flow fields will solve quicker than those with more complex flows. This means that the processors that were assigned the quick solutions sit idle until all the nodes in the population set have been evaluated. This wasted computation power can be put to use by rather using asynchronous evaluation.

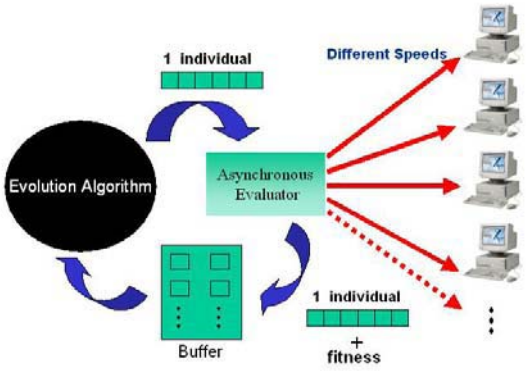


Figure 6: Parallel Computing and Asynchronous Evaluation

The approached taken by the HAPMOEA code is similar to the work conducted by Wakunda and Zell [50] which uses an asynchronous method of algorithm update. This approach rather evaluates each candidate solution individually against the function objectives. The implementation is shown in Figure 6 where instead of offspring being generated and evaluated at the slave nodes during each iteration as a complete set, the HAPMOEA code generates each offspring individually and passes that offspring to an idle slave node for evaluation. When a fitness value has been assigned to the candidate solution by the slave node, the offspring is passed back to the master node where the selection operator is used to determine whether the solution is inserted into the main population or rejected.

As the optimisation is progressing with time, the newly evaluated candidate solution can not be compared to any other candidate solution or the main population directly. Solutions are compared to a previously calculated rolling benchmark, calculated against past accepted and rejected individuals. If a solution is inserted into the population, this is done by replacing an existing individual in the population according to some user set rule. For accurate computations the length of this benchmark buffer need not be too long and twice the size of a generation's population is more than ample.

3.4.5 Pareto Tournament Selection

As the HAPMOEA code can be used for both single and multiple objective optimisations, the selection operator constructed must be able to handle both situations efficiently. Many researchers have extended the standard tournament operator [21, 51] or made use of non-dominated sorting methods [8, 51]. These non-dominated sorting routines construct the non-

dominated Pareto front for all the candidate points within the population. This first Pareto front is given a rank of one and the points making the front ignored and a new Pareto front constructed. This new front is given a rank of two and so on until all the candidate points within the population have been ranked. A problem arises when a fitness value has to be assigned to the different Pareto fronts as a Pareto front of rank two might not have half the fitness of a rank one Pareto front. A way around such a problem is to introduce problem specific information into the optimiser. This is undesirable as the 'black-box' nature of the code is no longer kept.

The HAPMOEA code makes use of an on-the-fly selection operator that does not require any problem specific information and works seamlessly with the asynchronous selection buffer *B*. The selection operator works by constructing a small subset of the selection buffer called the tournament $Q = [q_1, q_2, \ldots, q_n]$, shown in Figure 7. The tournament buffer is constructed from random entries from the selection buffer. If the candidate solution *x* is non-dominated when inserted into the tournament, the solution is inserted into the main population. The only information the HAPMOEA code needs to make efficient use of the tournament selector is the initial size of this subset. This size can vary from *B/2* to *B/6*. The larger the tournament size, the stronger the selective pressure placed on the candidate. The size of the tournament has been found to not overly influence the optimisation as long as the selection pressure is not too large. As the occupants of the tournament selector are taken from the asynchronous buffer at random, no niching or forced separation of variables was found necessary.

If the problem investigated is a single objective optimisation, the tournament selector reduces to the standard tournament selector [21, 51].

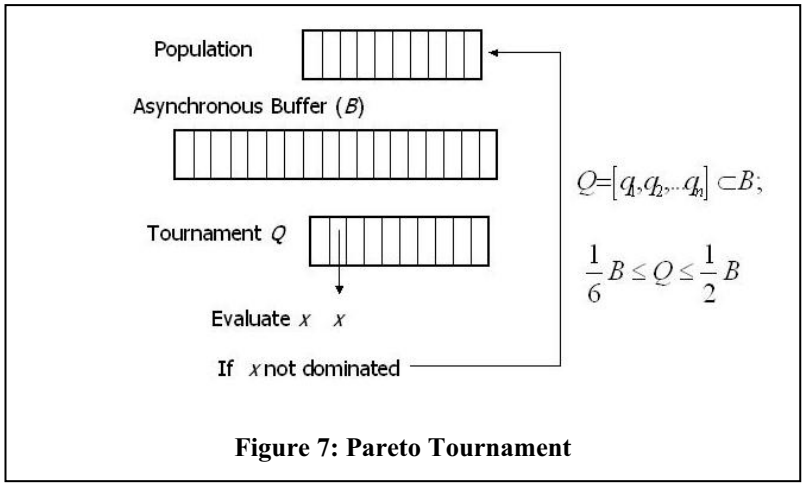


Figure 7: Pareto Tournament

The HAPMOEA code addresses the computation expense problem associated with standard evolutionary optimisers along with usability and code portability. The code greatly reduces the number of iterations required before the globally optimum solution can be found through the use of a hierarchical structure along with asynchronous execution.

4. AERO-STRUCTURAL ANALYSIS

This Chapter describes the aero-structural solver in detail. Section 5.1 describes the aero-structural program layout; section 5.2 details the validation of the aerodynamic solver while section 5.3 describes the structural model and analysis. Section 5.4 details the aero-structural analysis and results for two baseline geometries.

4.1 AERO-STRUCTURAL PROGRAM LAYOUT

A program to perform the aero-structural optimisation was written in Matlab®. The aero structural solver program integrates two commercial analysis tools for Finite Element Analysis (FEA) and Computational Fluid Dynamics (CFD) namely MSC.Nastran®, developed by MSC.Software®, PanAir, a high order Morino class panel method developed by Boeing. FRICTION, a low fidelity algorithm, is used to calculate the form and frictional drag about a candidate wing. The entire aero-structural program is controlled through a Matlab® script file. This allows for an easy coupling of the different required programs as one continuous Matlab® data structure can be utilized to define all the information passed between programs. A flow diagram of this process is illustrated in Figure 8. A description of the aerodynamic and structural analysis tools is given, followed by a general description of the validation cases.

The aero-structural program works in a very structured way as shown in Figure 8. On execution of the code, the workspace is cleared of any existing variables and any output files from previous program runs which may have existed in the working directory. The program loads in any user or optimiser specified inputs through the calling of the user generated input files.

Once loaded, the Aero-Structural program performs a number of geometry checks on the candidate wing and then creates the wing shape from the user specified inputs. From this output the FRICTION, PanAir and MSC.Nastran® input files are created and executed. After running the PanAir simulation, the aerodynamic loads are read in from the PanAir output files and mapped to the MSC.Nastran® simulation.

Once again the candidate wing is tested but this time it is to see whether or not the wing violates any penalty function to do with the aerodynamics or structures. Thereafter an output file containing the simulation results is produced.

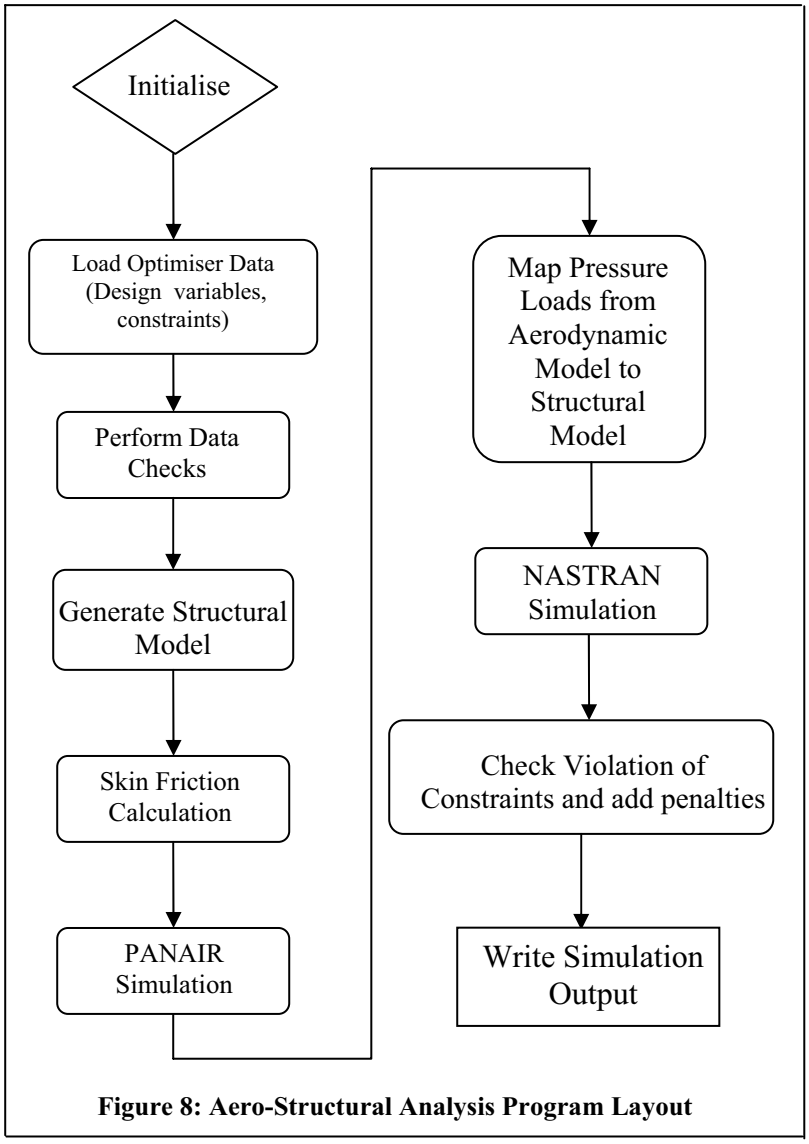


Figure 8: Aero-Structural Analysis Program Layout

If an unrecoverable error is encountered anywhere in the solution sequence, the aero-structural solver is able to recover and generate a simulation output file which indicates to the evolutionary optimiser that the solution failed.

4.1.1 INPUT FILES

The Aero-Structural solver loads in information describing the wing from two user defined sources namely '*UserWing.m*' and "*UserAuxiliaryData.m*" files (see Appendix D). This is the most common and simple method used when only performing a single run of the Aero-Structural solver. When coupled to the Evolutionary Optimiser, a different method is used.

The loading in of Evolutionary Optimiser input variables to PanAir is performed by a number of functions written in Matlab®. The functions have the ability to load in the variables. Four output files are used depending on the types of variables being passed to the aero-structural solver. Each file contains different information relating to the wing. The four different types of input files are used by the Aero-Structural Solver are:

- Planform data (*planformVariables.XX*)
- Auxiliary data (*auxiliaryVariables.XX*)
- Aerodynamics (*aerofoilsData.XX*)
- Structural data (*structuralVariables.XX*)

Examples of the above files can be found in the Appendix E.

Planform data includes crank locations, sweep angles, etc. Auxiliary data handles the different flow conditions such as Mach number and angle of attack. Aerodynamics simply contains the different aerofoil shapes as *x* and *y* columns corresponding to the aerofoil sections at the root, tip and crank positions. The structural data has the thickness of the different elements making up the aircraft wing.

4.1.2 OUTPUT FILES

The Aero-Structural Solver creates two output files. The first file contains a simple indication as to whether or not the Aero-Structural Analysis Program simulation had terminated successfully, the second, if the termination was successful, contains the aerodynamic and structural results from the simulation.

To indicate whether or not the simulation had terminated successfully (there were no critical errors requiring Matlab® to unexpectedly return command to the command prompt), a convergence file '*convergence.txt*' is created with either a zero (0), (no errors and hence the results file was created) or a one (1) to indicate errors within the program.

If the aero-structural solver is able to run all the required simulations on a candidate wing, a results file is written '*simresults.txt*'. This results file has a simple header that is a legend into the remaining lines in the file. The legend describes what the value in the file corresponds to, be it the induced drag calculated by PanAir, friction drag calculated by FRICTION, penalties the wing has been penalized with or the structural deflections. If the aero-structural is unable

to calculate any value due to a recoverable error such as the MSC.Nastran® simulation failing, the affected results are indicated by a hash symbol (#).

An example of the '*simresults.txt*' file can be found in the Appendix E.

4.1.3 Automatic Wing Geometry Generation

The aerodynamic and structural wing models are created though an automatic geometry modeller developed for this task. The aerodynamic model is created first and provided the body from which the structural model is created.

After all the input data checks are performed, the aerodynamic model is built in sections. Each section is made up of the area between cranks as is shown in Figure 9. For the wing shown in Figure 9, the first section constructed is between the root and intermediate crank and the second section is constructed between the intermediate and tip crank.

Matrix operations are used to define the initial layout of the section and take into account sweep and taper ratios. All sweep angles are defined relative to the global *XY* frame defined according to the normal aircraft convention where *X* is positive from the wing root leading edge towards the trailing edge, and the *Y*-axis is positive from the root crank towards the tip. The aerofoil shapes at the intermediate rib locations are determined through a linear interpolation of the aerofoils defined at each section boundary.

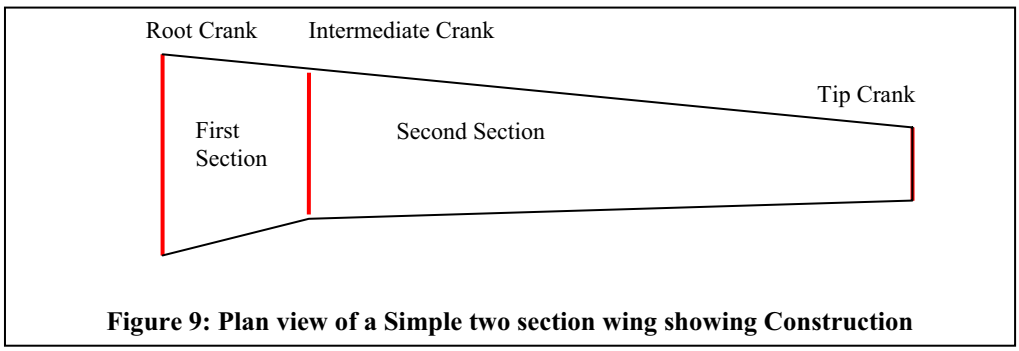


Figure 9: Plan view of a Simple two section wing showing Construction

Once all the sections have been created, the entire wing is defined through the amalgamation of all the computed sections.

As mentioned above, the structural model is extracted from the aerodynamic model. As all the cranks are defined according to the internal structure, this operation is easily performed.

Along with the external definition, when constructing the structural model, the automatic geometry modeller computes the location of all the Spar and Rib Caps.

4.1.4 Automatic Certification Of The Wing Geometry

Sometimes a random combination of number of cranks, chord ratios, sweep angles, etc or material properties can result in a physically unfeasible geometry. The Aero-Structural Analysis Program therefore has a number of internal checks to validate the data that is passed to it.

Structures

The material data defined in the '*MaterialProperties.m*' file is analysed to check if enough information is provided. Furthermore, the program determines which variable is missing data for certain crank locations and either replicates data, or if many variables are missing data, remove data from variables such that a common denominator is found.

Geometry

Two simple geometry checks are performed on the wing before any aerodynamic or structural analysis is performed. The first check ascertains whether or not the wing upper and lower surfaces intersect each other (Figure 10 a-Crossing), and the second check ascertains if the leading and trailing edges intersect (Figure 10 b–Switching). If either of these checks fails, the simulation is aborted.

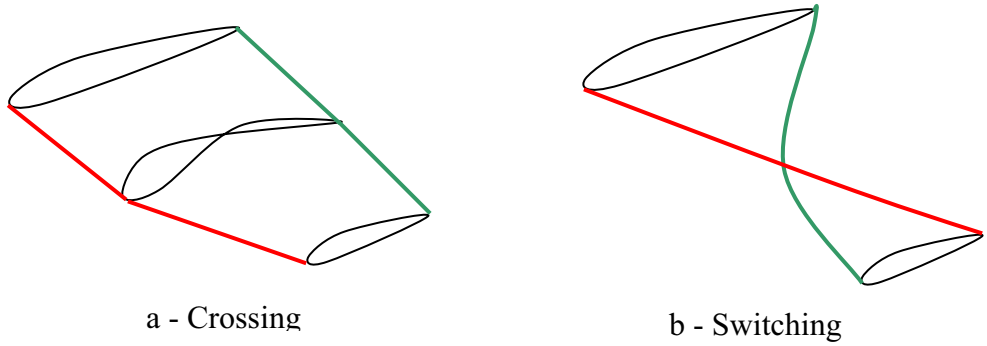


Figure 10: Geometry Checks performed by the Aero-Structural Analysis Program

Once both the data checks and basic geometry checks have been performed, The Aero-Structural Analysis Program checks if any of the data defining the internal layout of the wing will cause errors later in the simulation steps. A wing rib being referenced by two crank locations can cause errors. This can be caused by the method in which the ribs are linearly spaced throughout the wing and the crank locations are rounded up or down to the nearest rib as in Figure 11. This above mentioned case can cause modelling problems and hence one of the sets of data defining one of the crank locations would have to be removed.

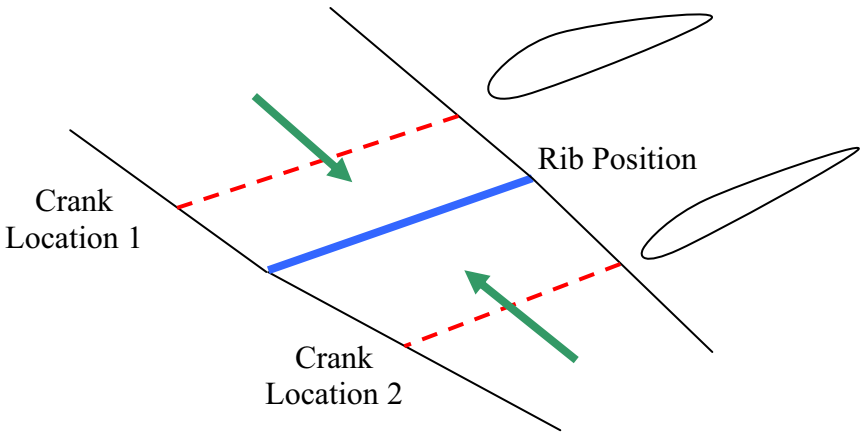


Figure 11: Errors relating to Rib Internal Structure

This removal process is carried out by deleting the data defining the more outboard of the two or more data sets and retaining the inboard.

4.2 AERODYNAMIC ANALYSIS

A proper selection and validation of analysis tools is required before coupling it with an optimisation process. A flow solver should meet some essential requirements such as: result accuracy, computational expense and robustness.

It is always desirable to use a high fidelity solver that can account for the flow complexities. The problem associated with using a full Navier-Stokes solver is the computational expense of the solution as one computation on a full 3-dimensional wing might take several hours on a supercomputer. Therefore, to reduce this expense it was necessary to find another high fidelity method that could yield accurate results at a minimum computational cost. It was for the above reason why it was chosen to use a high-order Morino class panel method code called PanAir to solve for the flow properties about the arbitrary vehicle wing. PanAir

calculates the Induced drag component produced by the wing as it travels through the air. Hence to correctly predict the total drag produced by the wing, both the Form and Friction components need to be calculated; for this task FRICTION [52] was used.

4.2.1 HIGH ORDER PANEL METHOD (PANAIR)

The higher order panel method PanAir was developed by Boeing as A502 to predict subsonic and supersonic linearised potential flows about an arbitrary geometry [53]. PanAir differs from earlier panel methods by employing a "higher order" panel method; that is, the singularity strengths are not constant on each panel. PanAir has the ability to analyse arbitrary geometry, and this ability, along with the ease with which models could be created and results read through input and output files, greatly influenced the decision to select it as the flow solver to utilise in the simulations and coupling with the optimiser. To simplify the construction of PanAir input files, certain bundled tools such as MAKEWGS converts the set of data points describing the external shape of the wing into the Langley Wire Frame Geometry Standard (WGS) [54] and PANIN, combines the MAKEWGS file with an auxiliary file to generate the complete PanAir input file. The flow conditions contained in the auxiliary file are the altitude, angle of attack and reference data for the wing.

The output from PanAir contains the calculated pressures and corresponding forces and moments about the simulated geometry. The outputted second order pressure coefficients were selected as they are the preferred ones for further analysis [53].

PanAir seeks to solve the potential flow equation. This equation is a combination of the continuity equation, momentum and speed of sound equations, subject to certain conditions such as the flow being irrotational. After considerable mathematical rearrangement, these equations can be written as:

$$\left(1-M_\infty^2\right)\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}=M_\infty^2\left(1+\frac{\gamma-1}{2}M_\infty^2\right)\frac{2u'}{U_\infty}\frac{\partial u}{\partial x} \tag{4}$$

This equation can be written in terms of the perturbation velocities, (Eqn 5) .The right hand side term is often neglected as it is of second order in the perturbation components.

$$\left(1-M_\infty^2\right)\frac{\partial u'}{\partial x}+\frac{\partial v'}{\partial y}+\frac{\partial w'}{\partial z}=0 \tag{5}$$

By bringing in the velocity potential we have:

$$u' = \frac{\partial \phi}{\partial x}$$
$$v' = \frac{\partial \phi}{\partial y} \tag{6}$$
$$w' = \frac{\partial \phi}{\partial z}$$

Consequently,

$$\left(1 - M_{\infty}^{2}\right)\phi_{xx} + \phi_{yy} + \phi_{zz} = 0 \tag{7}$$

PanAir solves these equations and uses a fully continuous quadratic doublet, and linear source distribution along both the span and chord-wise directions.

A complete derivation of Eqn 7 can be found in Bertin [55].

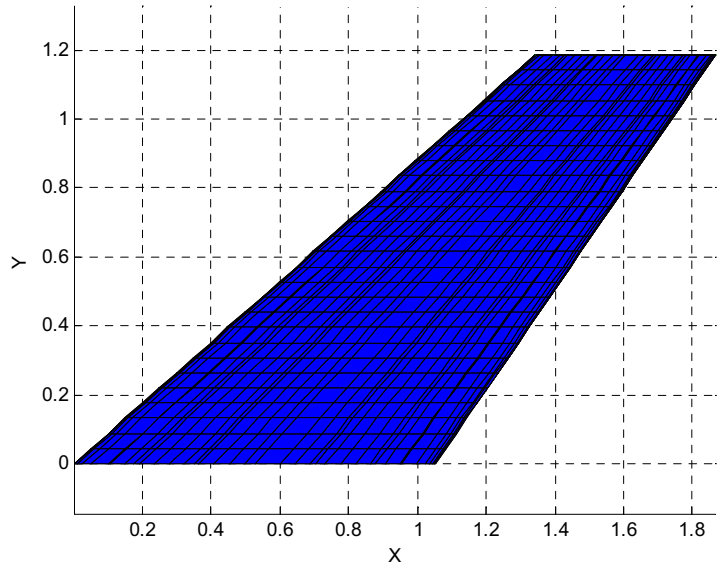


Figure 12: External Surface Elements Generated for PANAIR Calculations

In PanAir, for a panel solution to be found about an arbitrary configuration the wing or similar object must be closed, i.e. there are no gaps in the model. This means that for the simulations performed, the wing described has to be mirrored about the XZ plane such that it

made a complete 'flying wing'. This assumption of only modelling the wing in isolation holds as the wings under investigation have a high aspect ratio and hence the fuselage effects would only affect a small portion of the flow over the wing root.

Figure 12 shows the upper surface of the aerodynamic model of the wing. For all the simulations performed, the authors modelled the wings reflected about the *XZ* plane with a full outer skin along the upper and lower surfaces of the wing, and a straight wing tip. The wing tip panels are plotted in Figure 13.

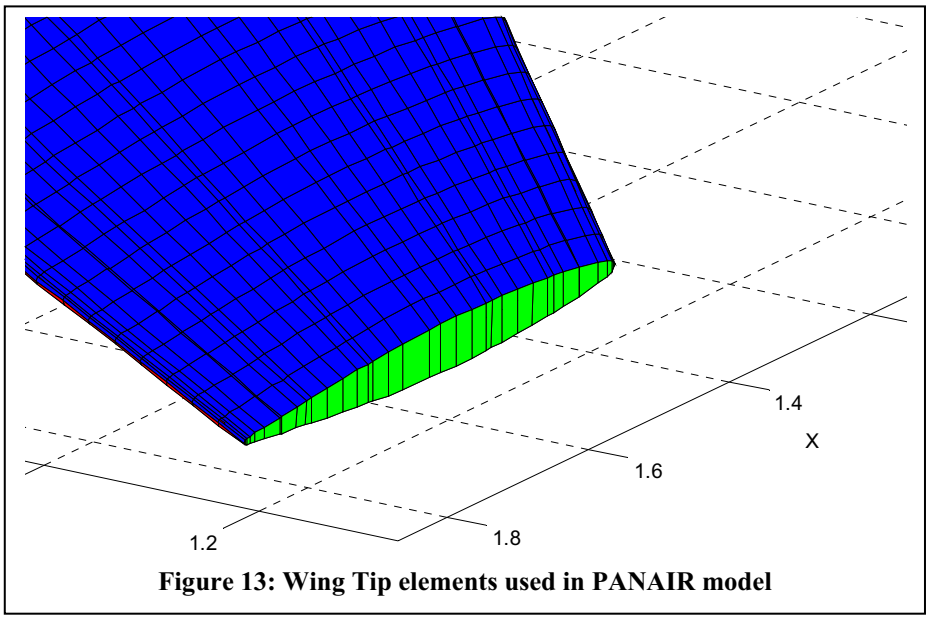


Figure 13: Wing Tip elements used in PANAIR model

A wake is automatically attached to the trailing edge of the wing by PanAir.

4.2.2 VALIDATION OF PANAIR

Before coupling PanAir with the optimiser, the program was validated against experimental wind tunnel data and results published by Margason [56]. The test case considers the 45° swept back wing operating at $M_\infty = 0.6$ and at an angle of attack $\alpha = 6.3$ deg.

Along the leading and trailing edges, the panel sizes increased exponentially for the first ten percent of the chord, and then remained at a constant size for the remaining eighty percent. Through experimentation, this was found to yield the best result for the lowest computational cost.

Results

The CPU time for this aerodynamic computation was 12 minutes on a single Pentium 4 2.4 GHz processor.

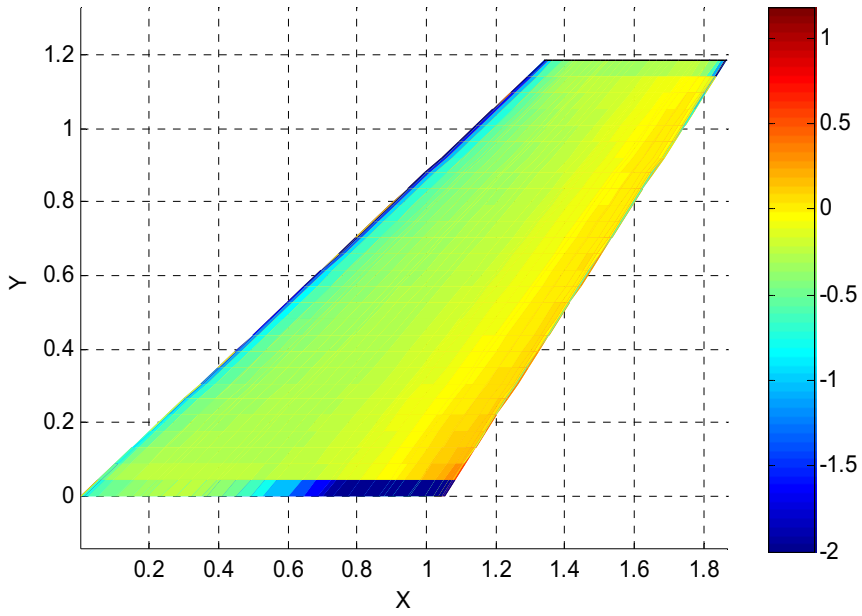


Figure 14: C_P distribution about the Lower Surface of the Test Section

The graded coefficients of pressure distribution for the upper and lower surfaces are shown in Figures 14 and 15

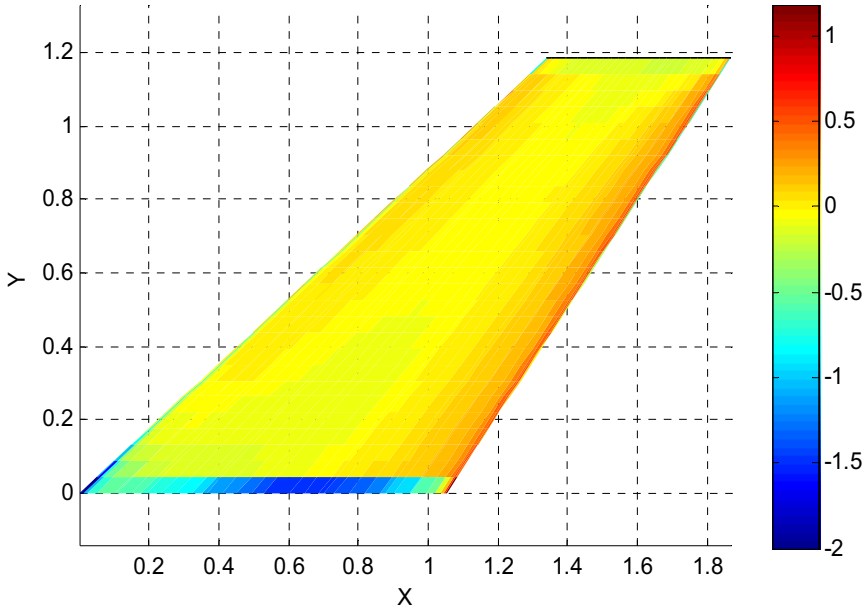


Figure 15: C_P distribution about the Upper Surface of the Test Section

Figures 16 and 17 show the second order pressure coefficient distributions at two span-wise stations as calculated by PanAir. The results from the fifty five percent span locations are shown in Figure 16. As indicated in this figure, the results attained are in close agreement with the experimental results reported in Kolbe [57] and Margason [56].

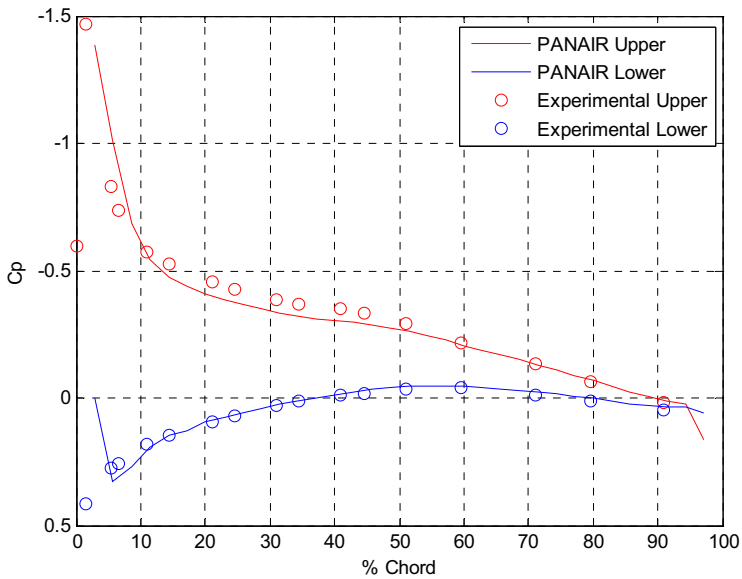


Figure 16: C_P distribution at 55% semi-span.

Figure 17 shows the results at 8 % span. The discrepancy between experimental and PanAir results close to the wing root is due to a mismatch in the Kutta condition. This is due to the set up and execution of PanAir where the Kutta condition is automatically imposed on the boundary with the trailing edge and wing wake. This resulted in an incorrect Kutta condition at the root chord and hence the condition of the upper pressures equalling the lower pressures, as should be found on the leading and trailing edges, was not correctly realised. Further along the wing, the Kutta condition is met as can be seen from Figure 16.

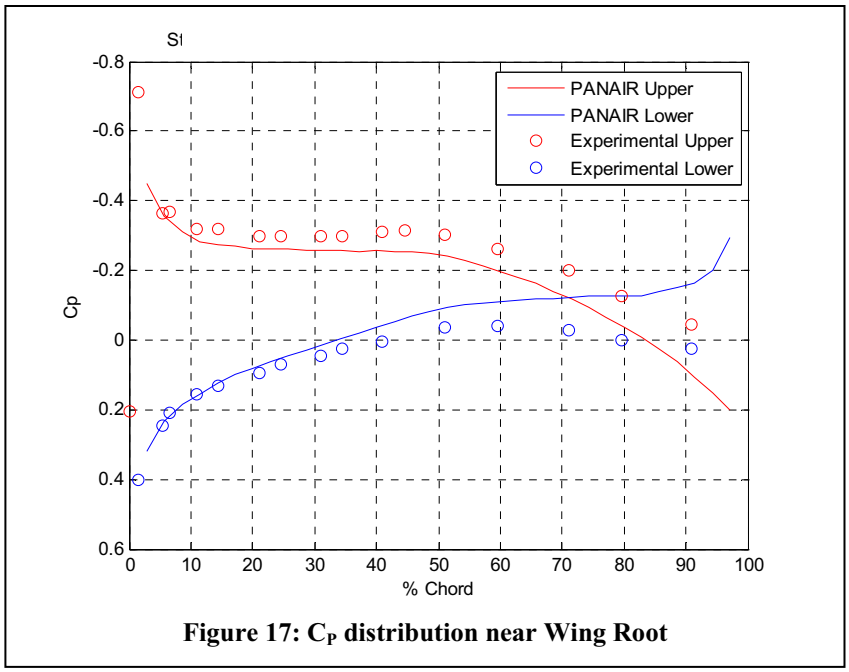


Figure 17: C_P distribution near Wing Root

Concluding the validation study, it is shown that the results obtained by PanAir are in good agreement with experimental data. PanAir has capabilities to provide accurate results for different arbitrary geometries and solve the aerodynamic characteristics on 3D wings. PanAir provides some advantages over more complex Navier-Stokes solvers:

- Good accuracy considering the linearilised flow assumption
- A reduction in computational expense.

The computational time for a single PanAir calculation varied between two and twelve minutes for an arbitrary wing on a Pentium 4, 2.4 GHz machine. The time taken was dependant on the number of discritised surface panels used to model each wing. In contrast, a full Navier Stokes solution takes in the order of several hours. Therefore, the aerodynamic solver has the capability to accurately model the flow about arbitrary geometries and is included in the aero structural algorithm and overall program coupling with the optimiser.

4.2.3 FRICTION

As mentioned in previous section, the program FRICTION is used to estimate the Form and Friction drag produced by the wing at the simulation conditions.

Form drag is drag due to the geometry of the wing being analysed and is influenced by the frontal area of the wing known as the thickness to chord (*t/c*) ratio.

Friction drag is drag due to the movement of air molecules along the wing surface which lead to the creation of boundary layers. If the boundary layer is laminar, momentum and energy are only mixed between neighbouring layers in the flow and on a microscopic level; hence the friction drag is small. As the boundary layer becomes turbulent and the mixing between sections of the boundary layer occurs on a macroscopic level, the friction drag can be many times larger than that for a laminar boundary layer.

FRICTION calculates the laminar boundary layer friction coefficient through the use of the standard Blasius formula (Eqn 8) with the inclusion of the Chapman-Rubesin constant C^*.

$$C_f = \frac{0.664\sqrt{C^*}}{\sqrt{\mathrm{Re}_x}} \tag{8}$$

The turbulent boundary layer is calculated through the use of the van Driest II method. For a full details on the method employed when calculating the turbulent boundary layer, refer to Bertin [55].

Assumptions

In the simulation process it is assumed that fifty percent of the wing was turbulent, and the remainder laminar. This was set as an assumption and kept for all simulations as smooth graphite epoxy has a very low roughness value, even though this value would vary between simulated wings in real life. The value of fifty percent chord was selected after consulting pictures of ribbon experiments performed on numerous aircraft wings [55].

4.3 STRUCTURAL ANALYSIS

The structural analysis is conducted using MSC.Nastran®, a modular Finite Element Analysis (FEA) software package written by MSC.Software®. MSC.Nastran® has modules for heat transfer, dynamics, spot welding, aero-elasticity, and non-linear analyses.

In this work, the operation of MSC.Nastran® is done through the use of the Bulk Data File (BDF) input method. For the simulations SOL 106 is used which defines the Non-linear Static analysis. This method was chosen over SOL 101, Linear Static as it was expected that some of the wing geometries defined would undergo large deflections and or large strains. It was therefore necessary to correctly determine structural forces formed and hence the Non-linear Static solution method, as described in Section 5.3.3, was chosen.

4.3.1 STRUCTURAL MODEL AND CONSTRAINTS

For the structural analysis, a simplified finite element model constructed from a varying number of ribs and spars was used. The majority of the structural model was assembled using shell elements with the spar and rib caps modelled as rod elements. The number of nodes and elements and hence degrees of freedom (DOF) varied depending on the internal wing structure. A sample finite element model is illustrated in Figures 18 and 19.

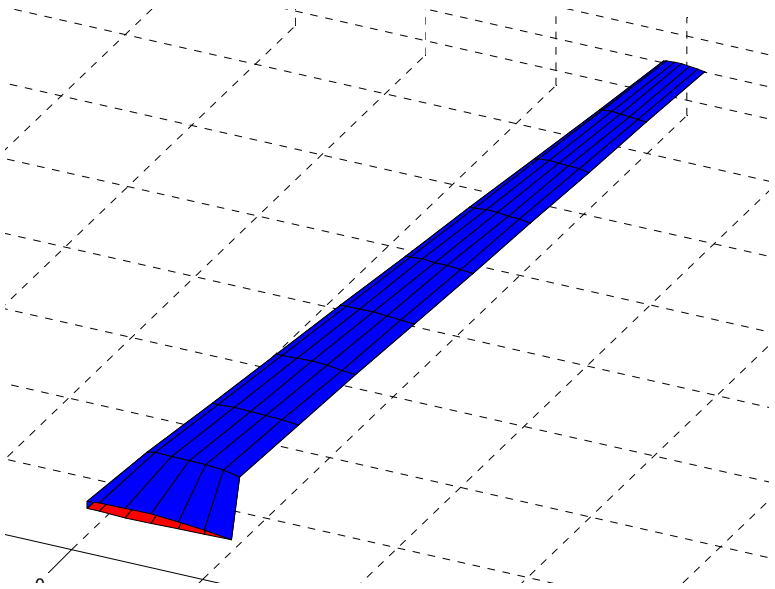


Figure 18: Structural finite element model used in MSC.Nastran (Upper surface panels – blue, Lower surface panels– red)

The different components making up each candidate wing, namely spars, ribs, wing skin and spar and rib caps, are described in more detail below.

Spars

The Spars are linearly distributed over the chord depending on their number. This greatly reduced the number of variables in the simulation, but does not necessarily yield the optimal result for any given number of spars.

The thicknesses of the Spars are modelled to decrease according to a parabolic function as a function of the span. This modelling method allows the optimiser the ability to define thicknesses that vary at a constant rate down the span to Spar thickness which vary parabolically, increasing the scope of the optimisation solution.

Each spar is broken down into sections. The sections are defined as the distance between consecutive ribs and this allows for a minimum of MSC.Nastran® structural elements to be used when defining the overall structure as shown in Figure 19.

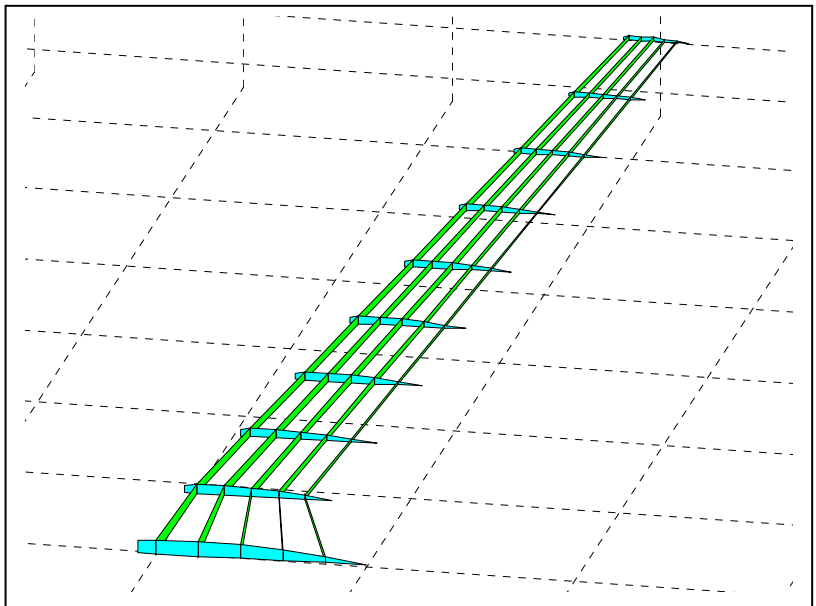


Figure 19: Wing Spars (Green) and Ribs (Light Blue)

As per recommendations by Marisarla [58], the number of spar element sections never decreased below 8. This allows for the correct modelling of non-linear element bending with sufficient accuracy.

Ribs

As with the Spars, the Ribs are also broken down into sections. The sections are defined from the leading edge to the first spar, between consecutive spars if they existed and then from the trailing edge spar to the wing trailing edge.

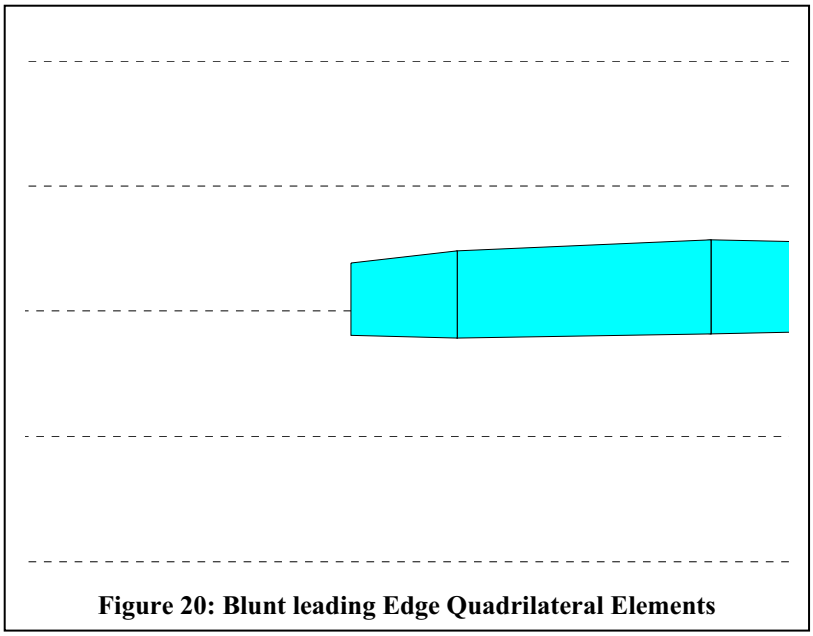


Figure 20: Blunt leading Edge Quadrilateral Elements

The Rib thickness is also varied according to a parabolic function in a similar manner as to the Spar thicknesses.

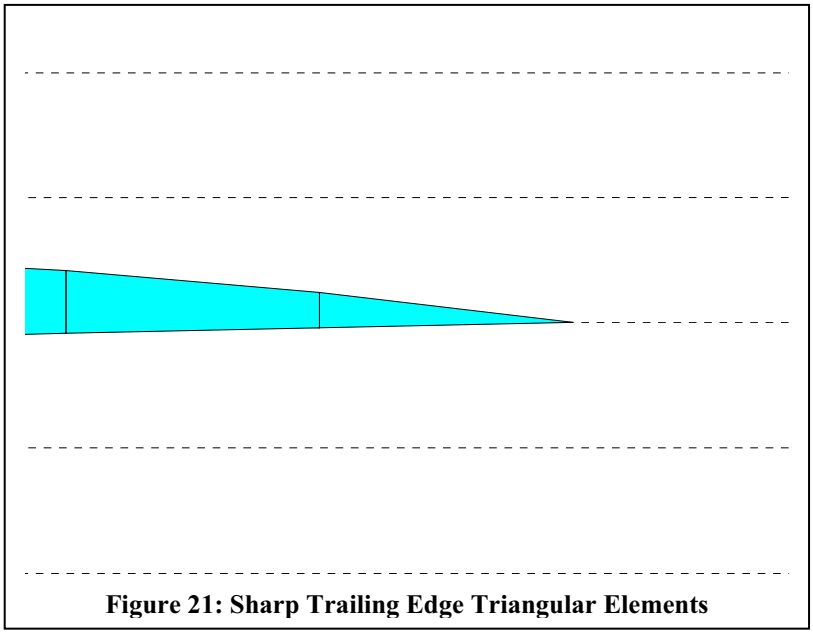


Figure 21: Sharp Trailing Edge Triangular Elements

As the rounded shape of the leading edge provides additional torsional support in the structure, the leading edge section of each rib is modelled using a quadrilateral element as in Figure 20. Furthermore, the trailing edge section is modelled using a triangular element to retain the correct structural modelling as shown in Figure 21.

Wing skin

The wing skin is defined by quadrilateral element sections. These sections covered both the upper and lower surface of the wing and are primarily used as a means of accurately placing the pressure loads onto the internal wing structure. Each section therefore extended between neighbouring spars and ribs, and in the case of the blunt wing leading edge, between rib points as in Figure 22.

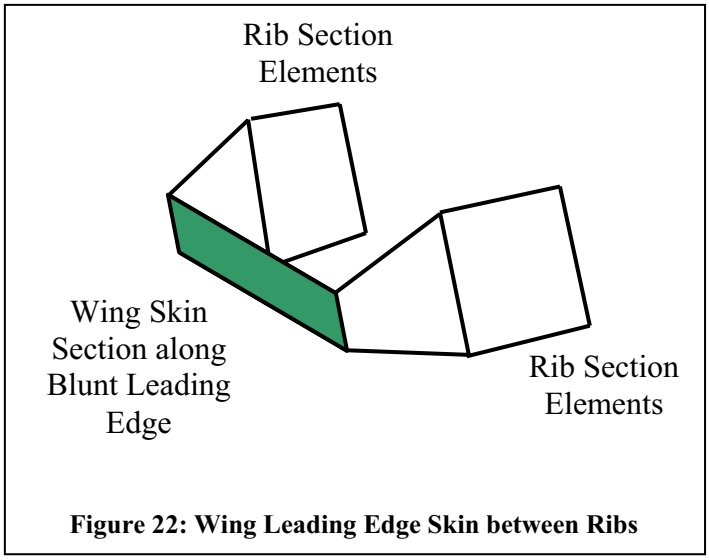


Figure 22: Wing Leading Edge Skin between Ribs

As the wing skin thickness varies from wing root to wing tip and from the wing leading edge to the wing trailing edge, so the wing skin thickness in the MSC.Nastran® structural model has the ability to have a varying wing skin thickness as well. This variation is a simple linear function for both the variation along the span and chord.

Spar and Rib Caps

Spar and rib caps significantly add to the structural rigidity of the wing and hence are modelled as lumped thicknesses at the junction between spars and ribs with the wing skin as

rod elements. As with the rib and spar thicknesses, the rib and spar caps thicknesses have the ability to taper from the root to the tip.

4.3.2 INPUT FILE FROM STRUCTURAL MODEL

MSC.Nastran® works using a Bulk Data input file which defines the type of solver and structural model. Some details and considerations for a typical BDF file in the simulation are:

An example of an abridged BDF file can be found in the Appendix F.

Bulk Data

Before the definition of any of the Element Properties, etc, a number of parameters are set up for each run. The most important are those affecting nodal constraint and solution definition.

As the simulation being performed is a non-linear static one, the requirement to constrain nodes which underwent large rotations was relaxed.

The Non-linear Static solution method was selected. MSC.Nastran® simulations are run with fifty load steps. This allows the program ample chance to find a simulation solution within a minimum time frame. If a larger number were chosen, MSC.Nastran® would find more complexly loaded simulation solutions, but the cost would be in the increase in computational time required to perform all the structural simulations at this small load step size.

The other feature of interest in the non-linear parameter setup is the convergence criteria used. As the solutions being found are for coarse structural models defining rough design analysis, the convergence criteria was relaxed. This is backed up in ‘Quick Reference Guide’ for MSC.Nastran® [59].

Element Properties

The Spars, Ribs and Wing Skins are modelled using shell elements. Considerable research was conducted into different modelling techniques which have been used when defining the above mentioned structural elements [58], [60]. It was decided that shell elements would be used to define the structural members through the PSHELL command. For these structural elements this would not only allow bending and shear within each of the elements, but would

also account for the coupling properties such as Poisson's Ratio which would not be accounted for if only shear elements were used.

The drawback of using rod elements to describe the lumped rib and spar caps, described in later sections, is that rod elements have a constant cross sectional area. Furthermore, each rod element with a different cross sectional area needs to have a separate entry describing the properties attributed to that element. This was done through the PROD input deck.

Quadrilateral Elements

CQUAD4 input decks are used to define skin panels, wing spars and wing ribs which have varying thickness at all four corners of the quadrilateral elements. This allowed for better structural modelling compared to the use of a constant thickness throughout all the defined element sections.

Furthermore, CQUAD4 elements define plane strain plate elements and hence this feature is beneficial to the structural simulation.

One example of a CQUAD element definition is as follows:

CQUAD4 31 1 498 816 812 494
0.00058 0.00044 0.0004 0.00052

This method of CQUAD description allows for the definition of edge thicknesses at node points along the element.

For a full breakdown of the CQUAD4 and subsequent input decks, please refer to the MSC.Nastran® reference manuals available from MSC.Software®.

Triangular Elements

CTRI3 elements are used to model the trailing edge elements defining the wing ribs. As with the CQUAD4 entries, the use of CTRI3 elements allows for the specification of different element thicknesses at each of the node points. This allows for refined structural calculations.

An example of a CTRI3 element definition is as follows:

CTRIA3 252 1 1379 3340 1909
0.00132 0.00132 0.00132

The same definition type is used with the CQUAD elements as is used with the CTRI3 elements in that it allows for the ability to define the edge thicknesses at each of the node points.

Rod Elements

CROD elements are used to model lumped spar and rib caps; the desirable feature being that they are only able to carry tension and compression loads and hence do not alter the torsional properties of the wing section.

An example of a CROD element definition is as follows:

CROD,266,4,266,478

This entry simply defines the rod element according to the nodes located at each end of the element.

Materials

For all the simulations performed, only one material is currently used. Due to time constraints the material is defined as an isotropic graphite epoxy material that had constant material properties under tension, compression and shear. This resulted in a material with a Poisson's Ratio of 0.3, Young's Modulus of 1.53E11 and a density of 1310 kg/m^3. The material is defined through the MAT1 input deck and referenced by all elements.

An example of MAT1 describing the Graphite Epoxy composite used with both the High and Medium Altitude Long Endurance vehicles is as follows:

MAT1 1 153000000000.0 0.30 *M1*
**M1 1310.01*

Due to the large values used when describing the Young's Modulus of the material, the long field format was used.

Nodes

This section referenced the node number to the corresponding *x*, *y* and *z* location. The node numbers are used when defining the CQUAD4, CTRI3 and CROD elements.

Loads

Two loads are applied to each candidate wing structure. The second order pressure coefficients calculated by PANAIR and converted to pressures values are applied to the upper and lower wing skin panels, and a gravity load is added to model the deflection and strains due to the mass of the structure.

The gravity load is required as the mass of the structure will create a downward acting force, and hence a moment about the X-axis, which will then counteract the lifting force and moment generated by the applied pressures. Therefore the gravity load will decrease the deflection of the wing and give a more realistic indication of the initial deflection under the applied aerodynamic loads.

The pressure load is defined through the use of the PLOAD2 input deck. This input deck defines the magnitude of the pressures acting perpendicular to the surface defining a wing skin panel. Due to some pressures being larger than eight characters in length; it was decided that the large field format would be used.

```
PLOAD2* 2       -2917.363     54              *P54
*P54
```

From the above example code, one can see that the long field format was used as the applied pressures on each of the wing skin panels in column three could become very large about the leading edges.

Node Constraints

For simplicity it was decided that all the nodes along the wing root would be constrained. Although this would not happen in the case of a wing installed into the fuselage of an aircraft, for simulation purposes and to ease the comparisons made between results, the entire wing root is fixed in all displacements and rotations.

If the fuselage was included in the simulation environment, the wing spars would simply be extended to the fuselage centre line and the nodes defining the ends of the spars fixed in both displacements and rotations. This would be the correct method of simulating the node fixities for the wing as the fuselage would counter some of the forces and moments generated by the wing.

4.3.3 USE OF MSC.NASTRAN® FOR ANALYSING HIGH ASPECT RATIO WINGS

As it was unknown to what amount each candidate wing would deflect under the calculated aerodynamic loading, it was necessary to use a generic and robust method when solving for wing deflection. Furthermore, the deflection solution would need to be accurate if it were to be included in determining the overall Pareto front members and if the variables used to define the wing were to be passed to a higher level in the optimiser structure. It was therefore decided that a linear static solution (Eqn 9), calculated by MSC.Nastran® would not provide accurate results as this method would only solve the candidate wing using small deflection theory. The choice of solution method was also made after conclusions drawn by Marisarla [58].

$$KQ = F \tag{9}$$

As some of the candidate wings could deflect an amount greater than a linear theory could accurately model, the non-linear solution method was chosen. The limit to linear theory is roughly taken in the case of a cantilever beam to be deflections of the same order of magnitude as the beam thickness. The change to a non-linear solution method only required changes to the MSC.Nastran® input file.

For the Non-linear Static solution, the applied loads are broken down into a number of sub loads. These loads are then applied in steps and the solution energy and deflection iterated until convergence for that load step, and the solution sequence continued until the entire load has been applied to the structure. MSC.Nastran® has the ability to bisect the current load step if it is unable to find a solution with the current step size. If, even after a number of user specified load step bisections, MSC.Nastran® is unable to find a solution, the program terminates with a failure message.

Further information on the non-linear solution method used by MSC.Nastran can be found in the User Manual [61].

A draw back to using this method is that since the load was incrementally increased and iterated for deflection and energy balance, the solution time is longer than the Linear Static solution time, increasing the computational cost per iteration.

4.4 AERO-STRUCTURAL ANALYSIS OF BASELINE DESIGNS

To illustrate the use of the aero - structural solver the method is applied to two test cases related to UAV wing design; a Medium Altitude Long Endurance (MALE) research UAV similar to the Altair UAV wing and a High Altitude Long Endurance (HALE) UAVs similar to the Global Hawk wing.

4.4.1 MEDIUM ALTITUDE LONG ENDURANCE (MALE) UAV

Problem Formulation

A detailed analysis of a wing for a medium altitude research UAV application similar to the Altair UAV is considered. The operating conditions and data are based on details specified by NASA [1]. The aircraft maximum gross weight is approximately 3175 *kg*, has a wingspan of approximately 12.5 *m*, a mean chord of approximately 1.2 *m*, and a planform shape with 2.25° sweep between the root and second crank (Figure 23 and Table 1).

An assumption was made that the aircraft operated at 210 KIAS (108.10 m/s) at a cruise altitude of 41,600*ft*.

It was assumed that the wing uses the NACA4415 aerofoil at all spar locations as information on the actual aerofoils used was unavailable. Table 2 summarises some of the flight properties for the aircraft. These conditions assume an aircraft at mid weight-cruise during an extended cruise phase at intermediate altitude.

Minimum Required Lift Coefficient

The following equation was used to calculate the required coefficient of lift at a number of flight conditions (Eqn 10).

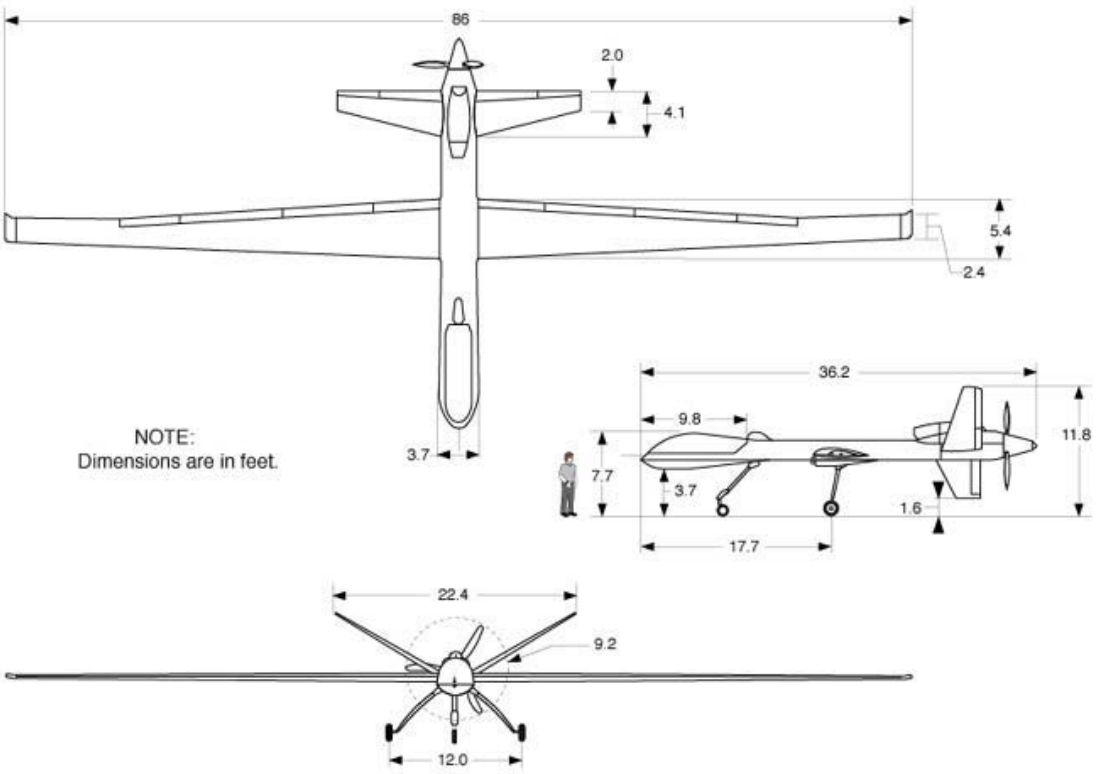


Figure 23: Schematic of Altair MALE [1]

$$C_L = \frac{W \times 9.81}{1/2 \times \rho \times V^2 \times S} \tag{10}$$

The results after using Eqn 10 are summarised in Table 1.

Characteristics, Units	
Maximum Altitude, *(m)*	15850
Cruise Altitude (80%MaxAlti), *(m)*	12680
Density @ Max Altitude, (*kg/m^3*)	0.16492
Density @Cruise Altitude, (*kg/m^3*)	0.27830
Air Speed, *(m/s)*	108.10
Speed of Sound @ Max Altitude, *(m/s)*	295.073
Speed of Sound @Cruise, *(m/s)*	295.073
Mach @Max Altitude	$M_\infty = 0.3663$
Mach @Cruise	$M_\infty = 0.3663$
Gross Weight, (*kg*)	3175

Table 1: Flight Conditions for Altair

Simulations are performed at an angle of attack of 4° which yield a calculated C_L greater than 0.64. This is equivalent to operating the Altair UAV at the beginning of the cruise condition as tabulated in Table 2.

Flight Condition	Required C_L
Max Gross Take Off at Maximum Altitude	1.1066
Beginning Cruise after T/O and Climb (5% fuel consumption)	0.6417
Middle of Cruise (50% fuel left)	0.5152
right before Landing (10% fuel left → reserve)	0.4028

Table 2: Required Altair C_L at different Flight Conditions

Aerodynamics Model

The PanAir and FRICTION codes are used for the aerodynamic analysis and are described in Section 5.2. In total, 1224 aerodynamic panels are used to discretise the Altair MALE model. Figure 24 shows the external planform shape.

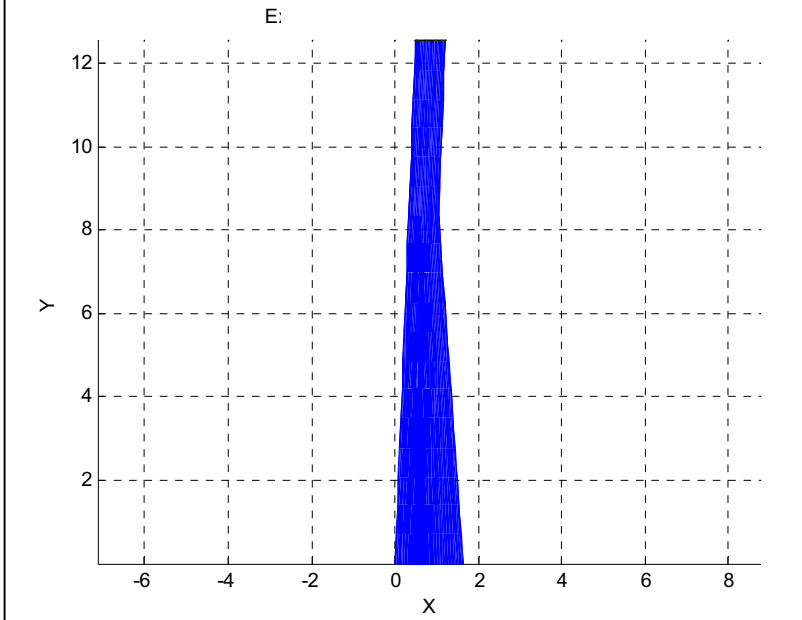


Figure 24: External Planform layout of Altair MALE wing

Figure 25 shows the linear (mid chord section) and exponential (leading and trailing edges) layout of the quadrilateral panels comprising the upper surface of the aerodynamics model.

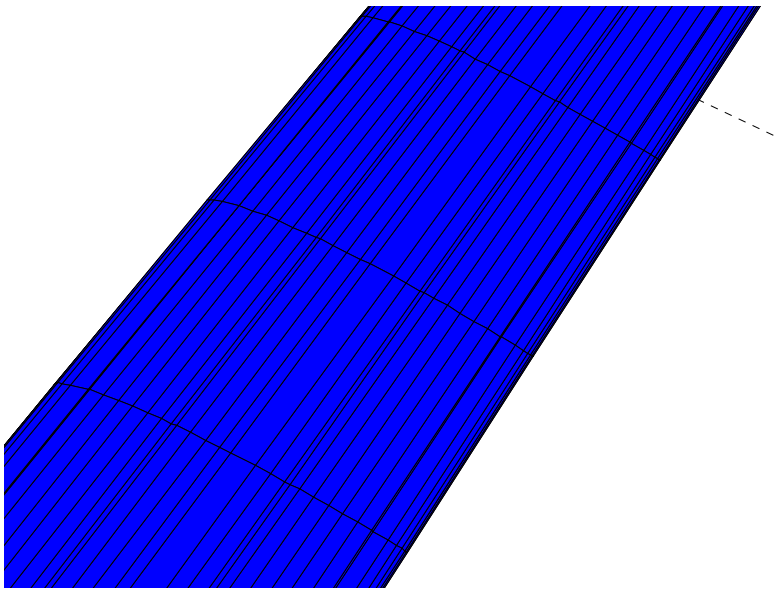


Figure 25: Aerodynamic Mesh for Altair MALE UAV

Structural Model

The structural model is described by the values displayed in Table 3 and shown in Figure 26.

Variable	*Value*
Number of spars	2
Number of ribs	10
Materials	
Wing skin	Graphite/epoxy
Spars	Graphite/epoxy
Ribs	Graphite/epoxy
Thicknesses	
Wing skin (m)	0.0001
Spars (m)	0.08
Ribs (m)	0.005
Thickness Ratios	
Wing Skin (Tip)	0.1
Wing Skin (Trailing Edge)	0.1
Spars	0.05
Ribs	0.25
Areas	
Spar Cap (m^2)	0.00375
Rib Cap (m^2)	0.0015
Material Properties	
Youngs Modulus (Pa)	1.53E11
Poissons Ratio	0.3
Density (kg/m^3)	1310

Table 3: Structural Variable Values for Altair UAV

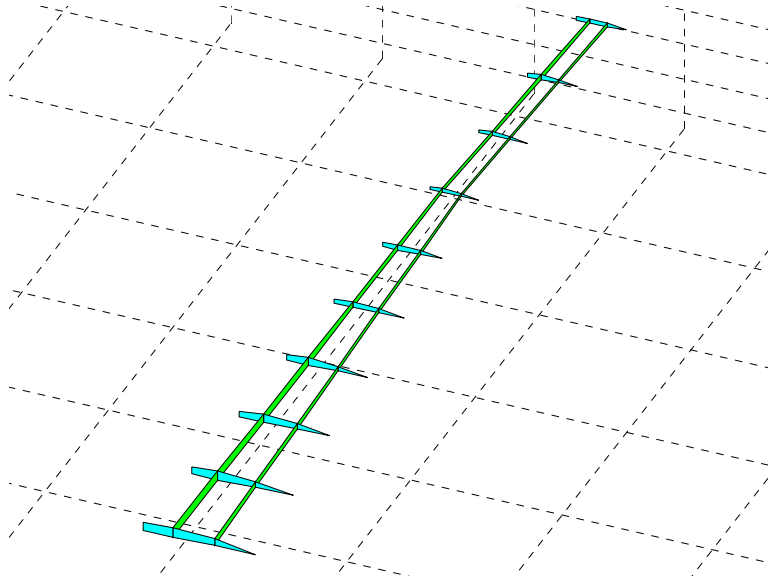


Figure 26: Internal Structural Model for the Altair UAV

Results

Aerodynamic analysis

Figures 27 and 28 show the wing span pressure coefficient distribution. The overall Altair wing aerodynamic characteristics for the above defined configuration are listed in Table 4.

The wing is found to weigh 220.9 *kg* which is in close agreement with the value estimated from Raymer [62].

L/D	23.9
C_L	0.66478
C_D	0.028
C_m	-0.1768
Deflection	12.06%

Table 4: Altair Wing Aerodynamic and Deflection Characteristics

Figure 27 shows the pressure coefficient distribution over the upper wing surface and Figure 28 that of the lower surface.

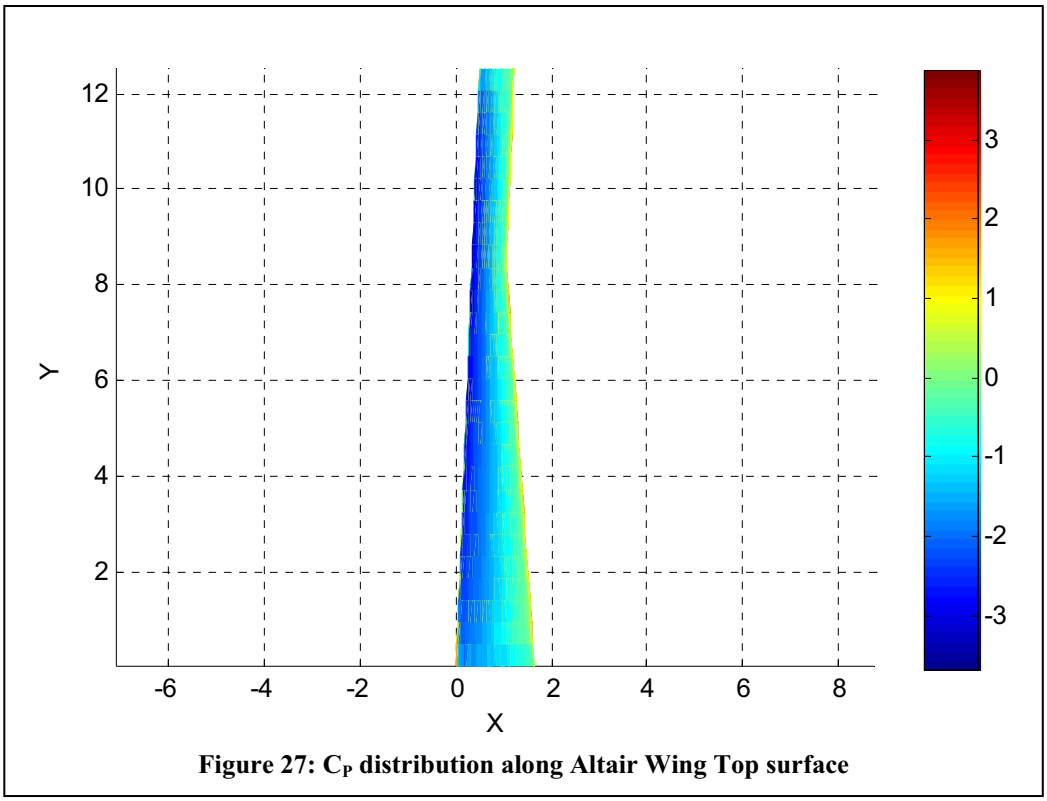


Figure 27: C_P distribution along Altair Wing Top surface

Performing a Mach sweep where the angle of attack for the vehicle was kept constant yields the results in Figure 29 for the calculated induced component of the total C_D value.

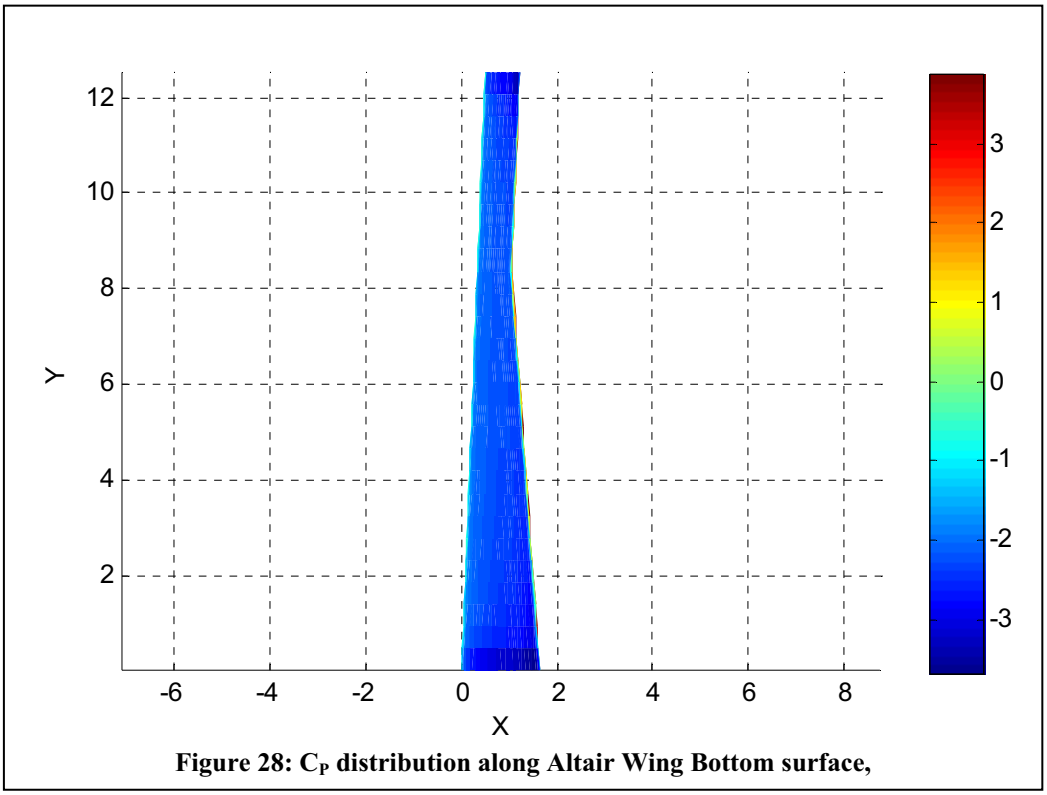


Figure 28: C_P distribution along Altair Wing Bottom surface,

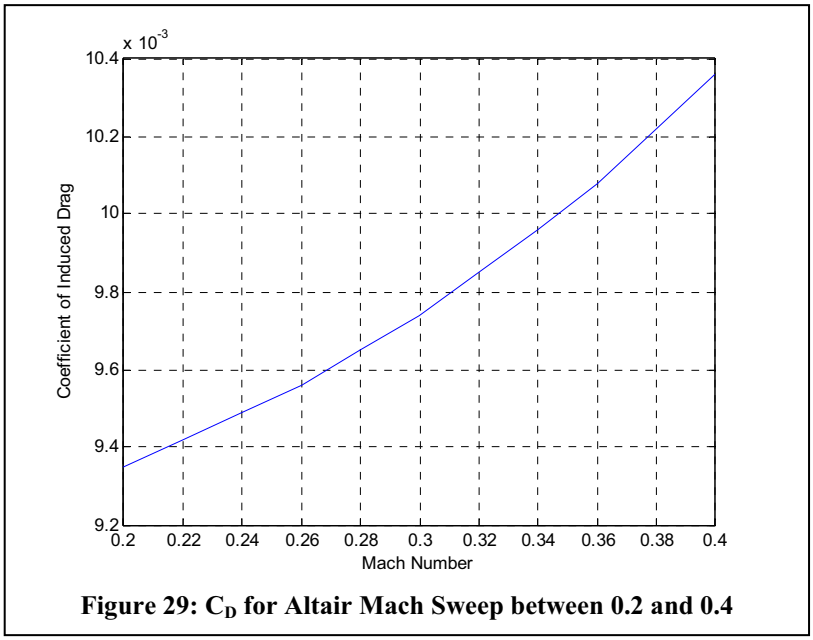


Figure 29: C_D for Altair Mach Sweep between 0.2 and 0.4

Performing the same calculations for the coefficient of lift yields the results shown in Figure 30.

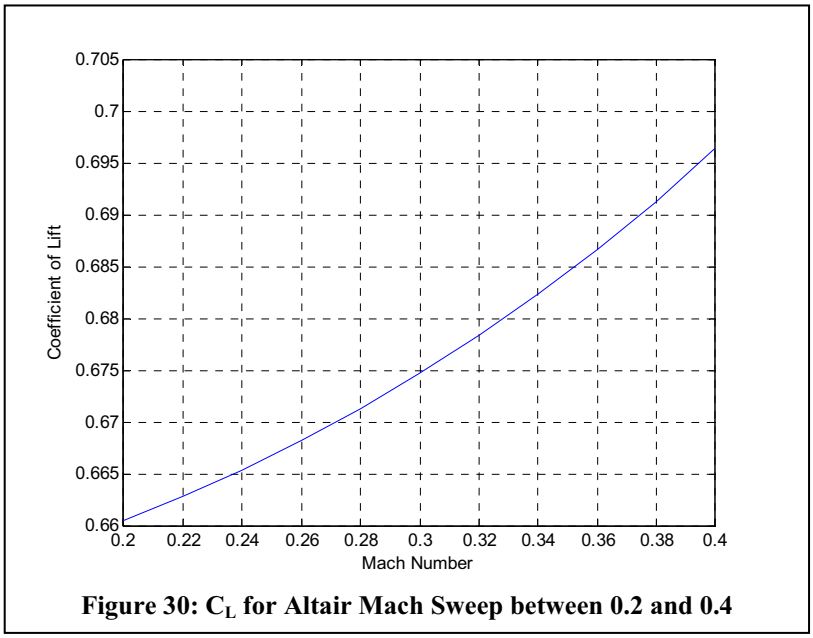


Figure 30: C_L for Altair Mach Sweep between 0.2 and 0.4

Structural Analysis

Figures 31 and 32 show the displacement and strain results respectively after applying the pressure values calculated using PanAir.

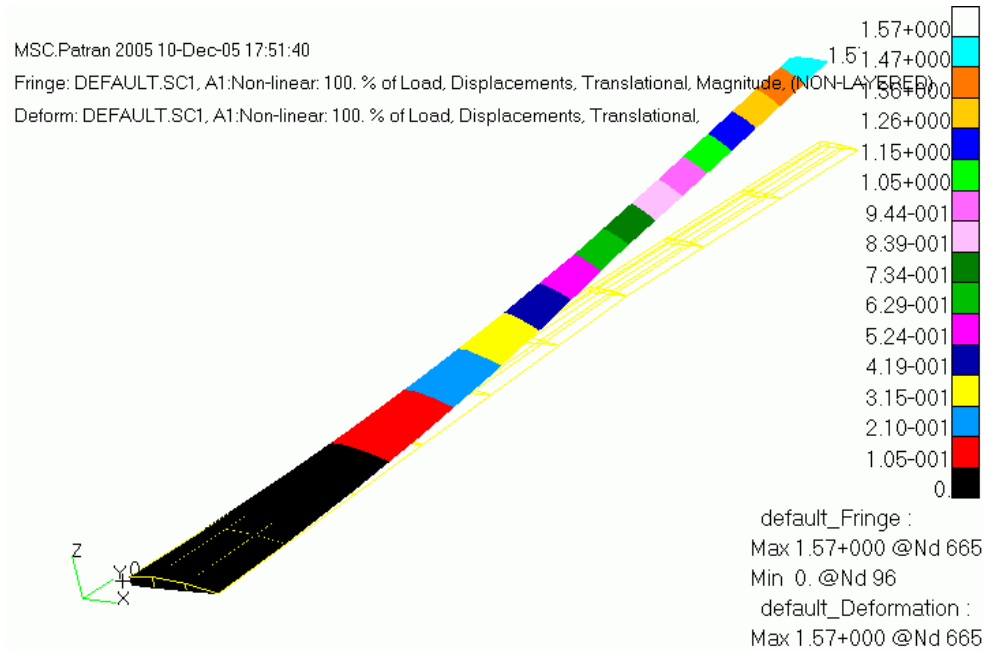


Figure 31: Non-linear Displacement for Altair UAV wing under 2.5g load case

The calculations above resulted in the definition of the benchmark for the Altair MALE UAV redesign to be considered in Section 6.1.

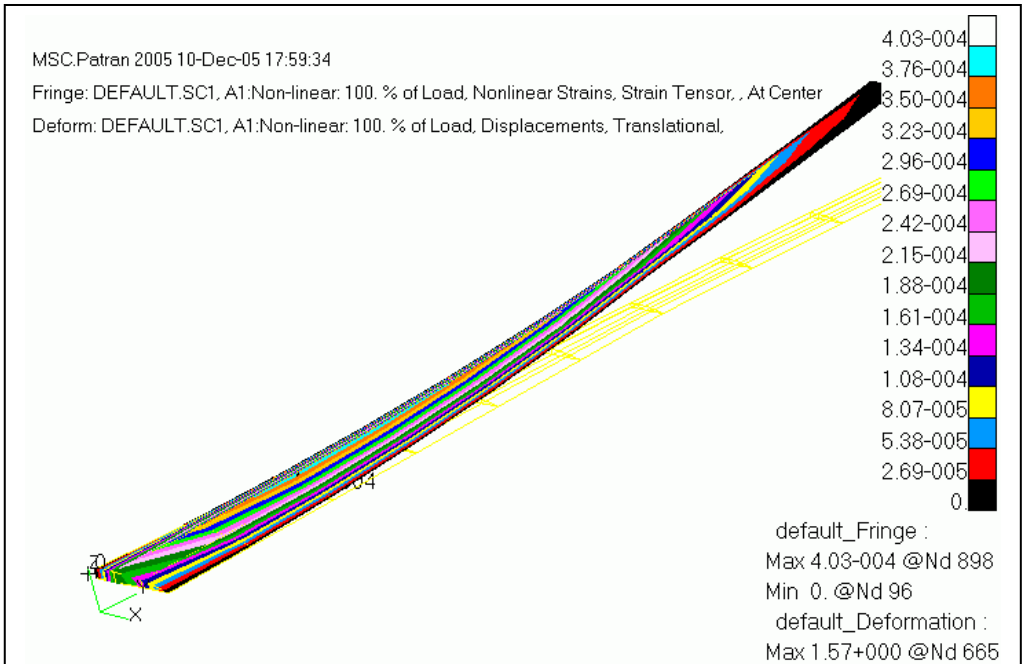


Figure 32: Non-linear Strain for Altair UAV wing under 2.5g load case

5.4.2 HIGH ALTITUDE LONG ENDURANCE (HALE) UAV WING

Problem Formulation

As with the Altair MAVE UAV a detailed analysis was performed on a UAV similar to the Global Hawk HALE. The operating conditions and data are based on the information provided in Air Force Technology [63]. The aircraft maximum gross weight is approximately 12,100 *kg*, has a wingspan of approximately 17.7 *m*, a mean chord of approximately 1.72 *m*, and a planform shape with 5.9° sweep between the root and tip (Figure 33). The assumption was made that the aircraft operates at 334 KIAS (176.56 *m/s*) at a cruise altitude of *15850 m*.

From Selig [64], the Global Hawk makes use of the NASA LRN 1015 aerofoil section throughout the wing.

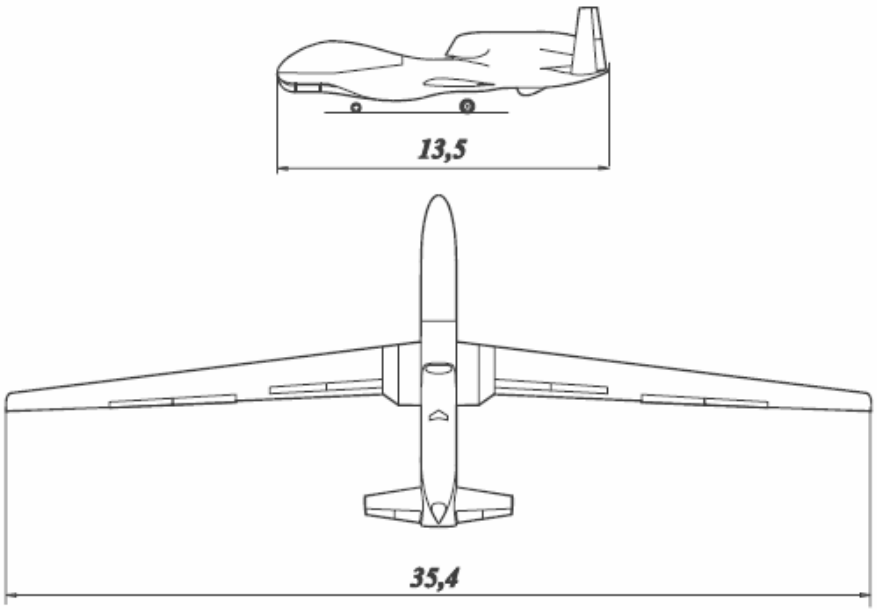


Figure 33: Schematic of Global Hawk HALE [2]

The Global Hawk flight conditions are summarised in Table 5.

Minimum Required Lift Coefficient

Eqn 10 was used to calculate the required coefficient of lift at each flight condition indicated in Table 5. The results after using Eqn 10 are summarised in Table 6.

Characteristics, Units	
Maximum Altitude, *(m)*	19812
Cruise Altitude (80%MaxAlti), *(m)*	15850
Density @ Max Altitude, (*kg/m^3*)	0.08761
Density @Cruise Altitude, (*kg/m^3*)	0.16938
Air Speed, *(m/s)*	176.566
Speed of Sound @ Max Altitude, *(m/s)*	295.073
Speed of Sound @Cruise, *(m/s)*	295.073
Mach @Max Altitude	$M_\infty = 0.5983$
Mach @Cruise	$M_\infty = 0.5983$
Gross Weight, (*kg*)	11,600

Table 5: Flight Conditions for the Global Hawk

Simulations are performed at an angle of attack of 4.75° which yields a calculated C_L greater than 0.8962. This is equivalent to operating the assumed Global Hawk UAV at the beginning of the cruise condition as tabulated in Table 6.

Flight Condition	Required C_L
Max Gross Take Off at Maximum Altitude	1.7341
Beginning Cruise after T/O and Climb (5% fuel consumption)	0.8962
Middle of Cruise (50% fuel left)	0.6718
right before Landing (10% fuel left → reserve)	0.4723

Table 6: Required Altair C_L at different Flight Conditions

Aerodynamic Model

The PanAir and FRICTION codes are used for the aerodynamic analysis and are described in Section 5.2. In total, 1980 aerodynamic panels are used to discretise the Global Hawk HALE benchmark model. Figure 34 shows the linear (mid chord section) and exponential (leading and trailing edges) layout of the quadrilateral panels comprising a portion of the upper surface of the aerodynamics model. This is the same model layout as is used with the Altair benchmark in Section 5.4.1.

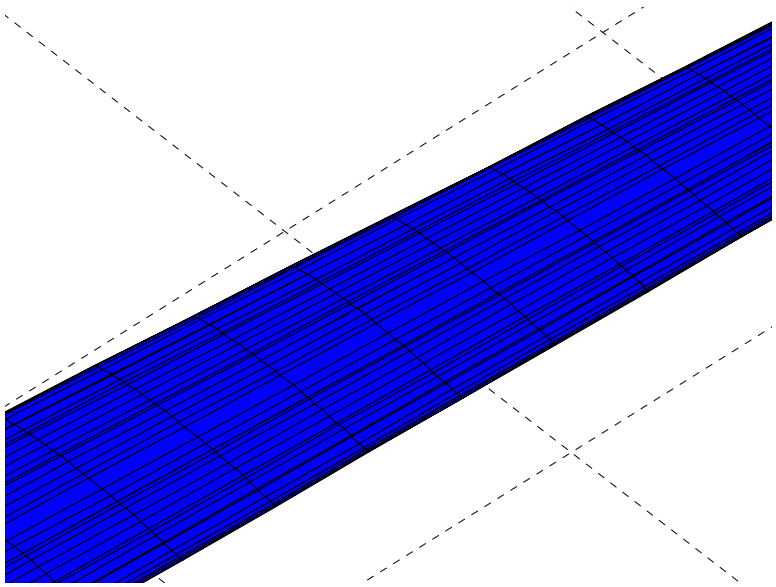


Figure 34: Aerodynamic Mesh for Global Hawk HALE UAV

Structural Model

The structural model is described by the values displayed in Table 7 and shown in Figure 36. Figure 35 shows the internal structure of the wing and Figure 36 the external panels.

Variable	*Value*
Number of spars	5
Number of ribs	10
Materials	
Wing skin	Graphite/epoxy
Spars	Graphite/epoxy
Ribs	Graphite/epoxy
Thicknesses	
Wing skin (m)	0.01
Spars (m)	0.08
Ribs (m)	0.0015
Thickness Ratios	
Wing Skin (Tip)	0.01
Wing Skin (Trailing Edge)	0.01
Spars	0.01
Ribs	0.01
Areas	
Spar Cap (m^2)	0.01
Rib Cap (m^2)	0.0005
Material Properties	
Youngs Modulus (Pa)	1.53E11
Poissons Ratio (ν)	0.3
Density (kg/m^3)	1310

Table 7: Structural Variable Values for Global Hawk UAV

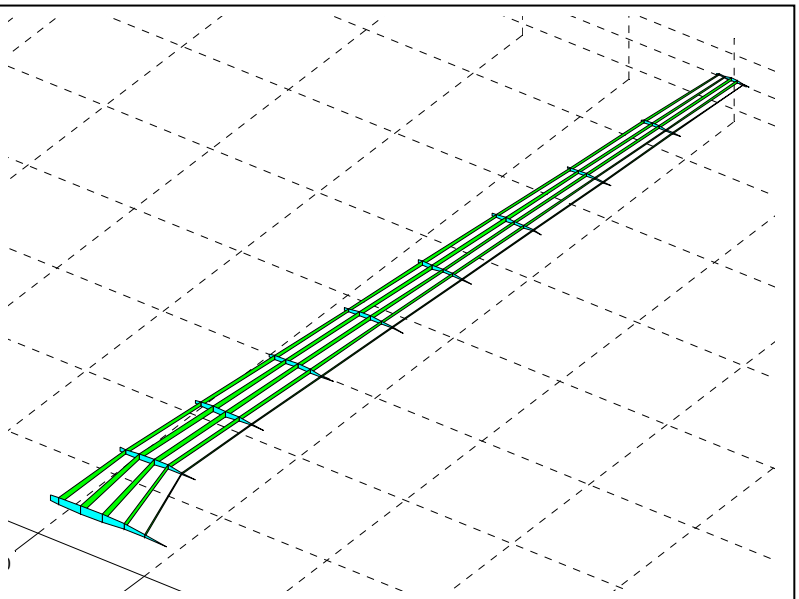


Figure 35: Internal Structural Model for the Global Hawk UAV

Results

Aerodynamic analysis

The overall Global Hawk wing aerodynamic characteristics for the above defined configuration are listed in Table 8. Figures 37 and 38 show the wing span pressure coefficient distribution.

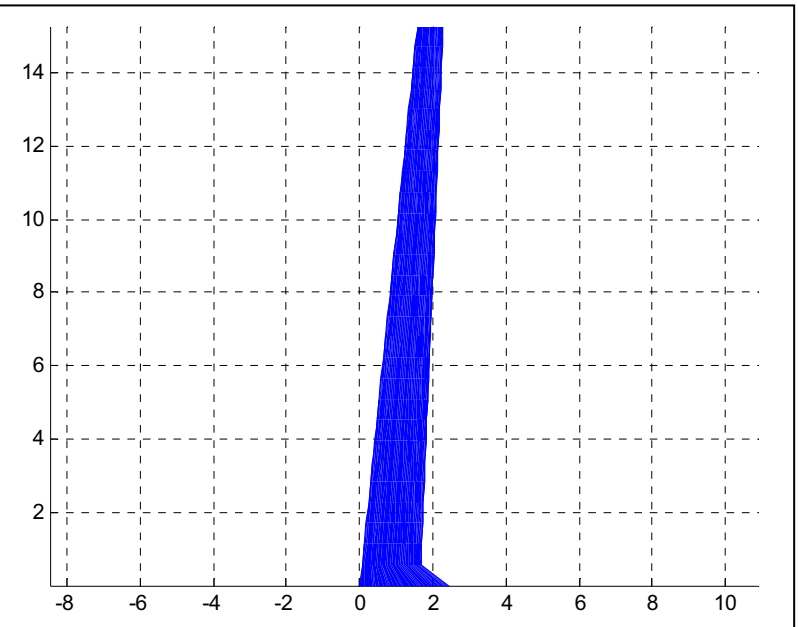


Figure 36: External Planform layout of Global Hawk UAV

L/D	31.729
C_L	0.905
C_D	0.02849
C_m	-0.208
Deflection	16.7%

Table 8: Global Hawk Wing Aerodynamic and Deflection Characteristics

The wing was found to weigh 1138.15 *kg* which is in close agreement with the value calculated through the use of Raymer [62].

Figure 37 shows the pressure coefficient distribution over the upper wing surface and Figure 38 that of the lower surface.

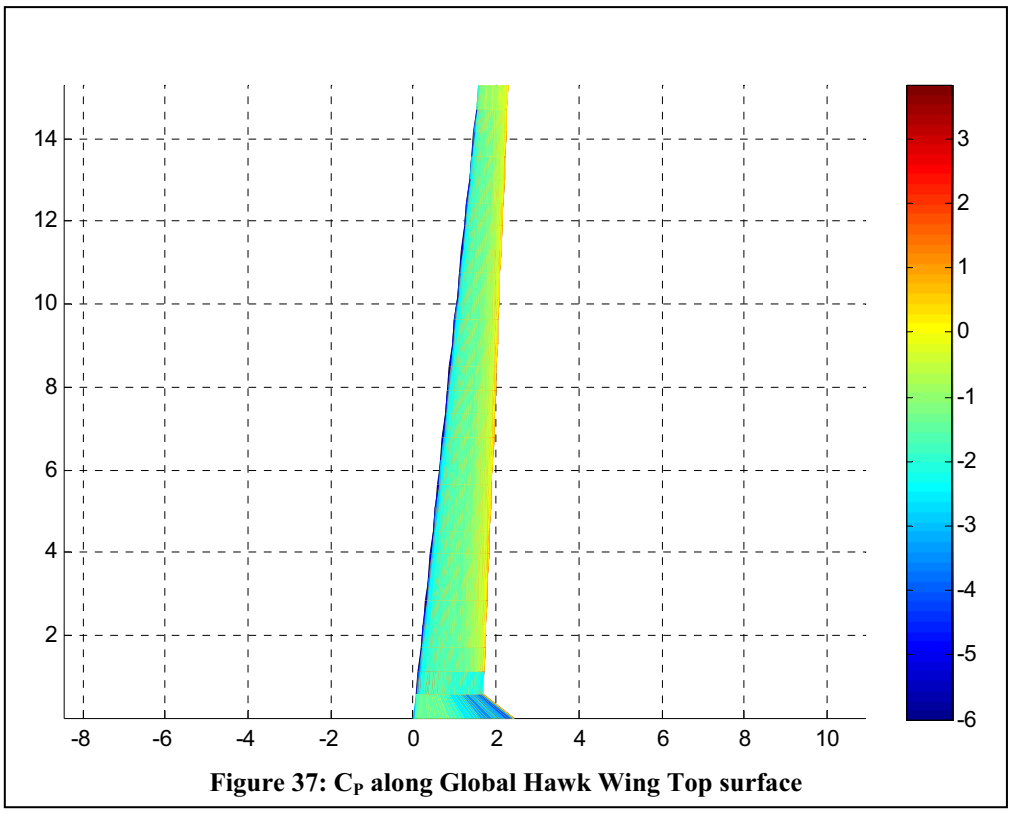


Figure 37: C_P along Global Hawk Wing Top surface

Performing a Mach sweep where the angle of attack for the vehicle was kept constant yielded the results in Figure 39 for the calculated induced component of the total C_D value.

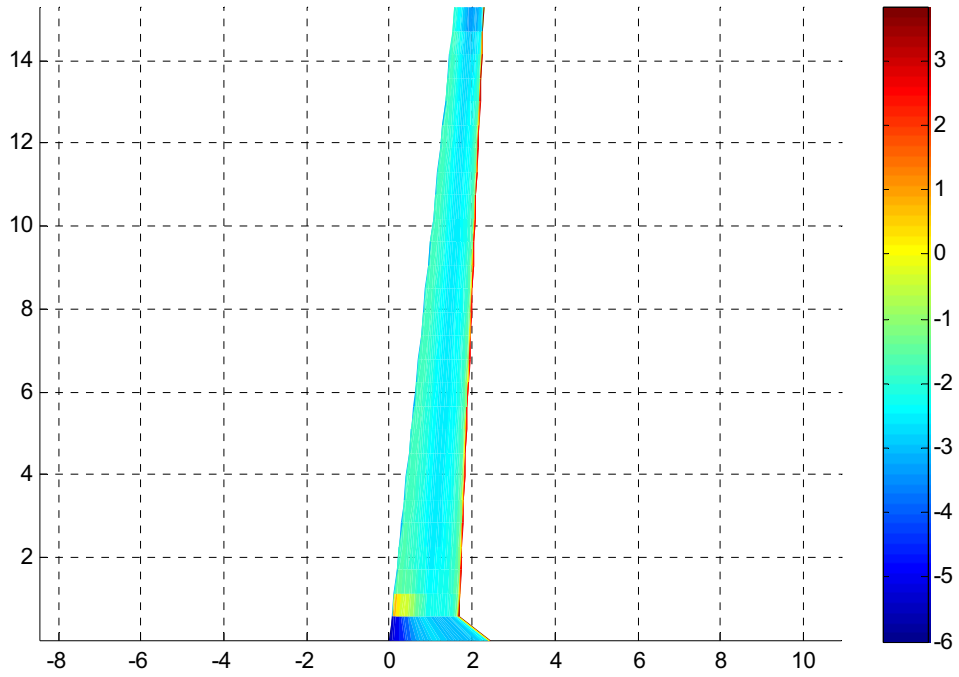


Figure 38: C_P along Global Hawk Wing Bottom surface

Performing the same calculations for the coefficient of lift yields the results shown in Figure 40.

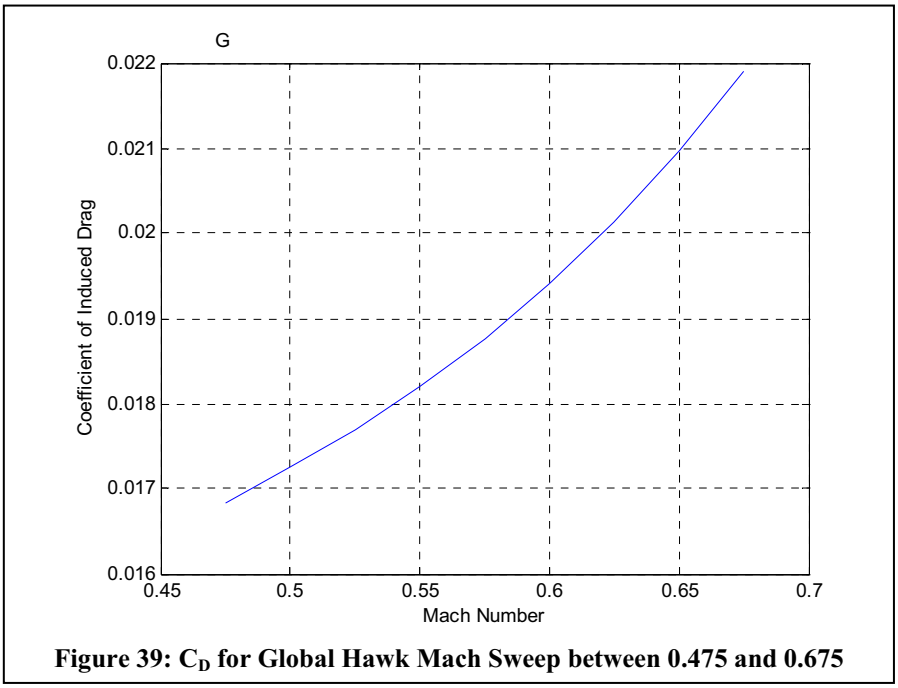


Figure 39: C_D for Global Hawk Mach Sweep between 0.475 and 0.675

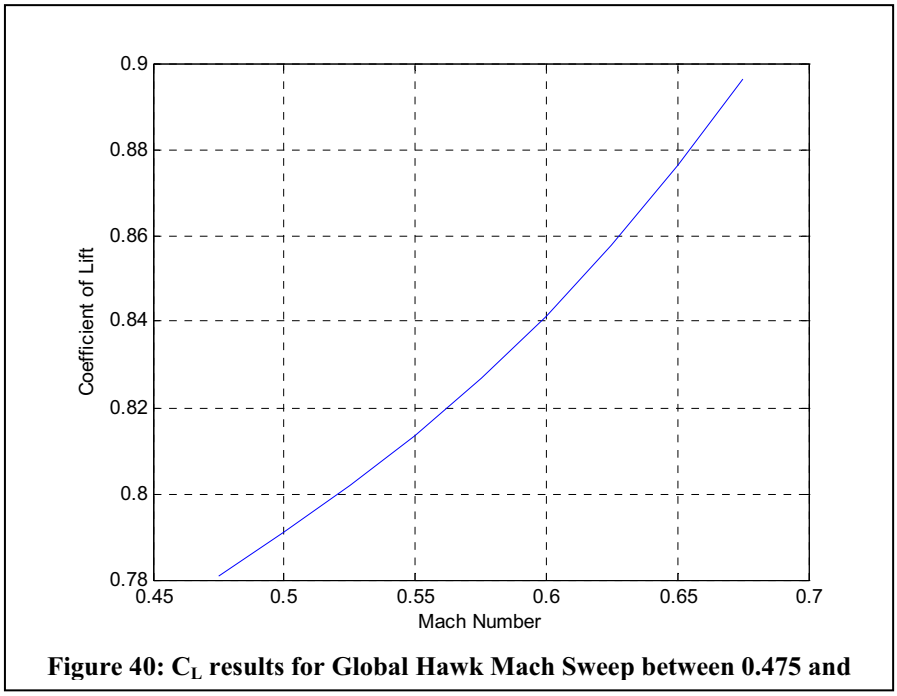


Figure 40: C_L results for Global Hawk Mach Sweep between 0.475 and

Structural Analysis

Figures 41 and 42 show the displacement and strain results respectively after applying the pressure values calculated using PanAir.

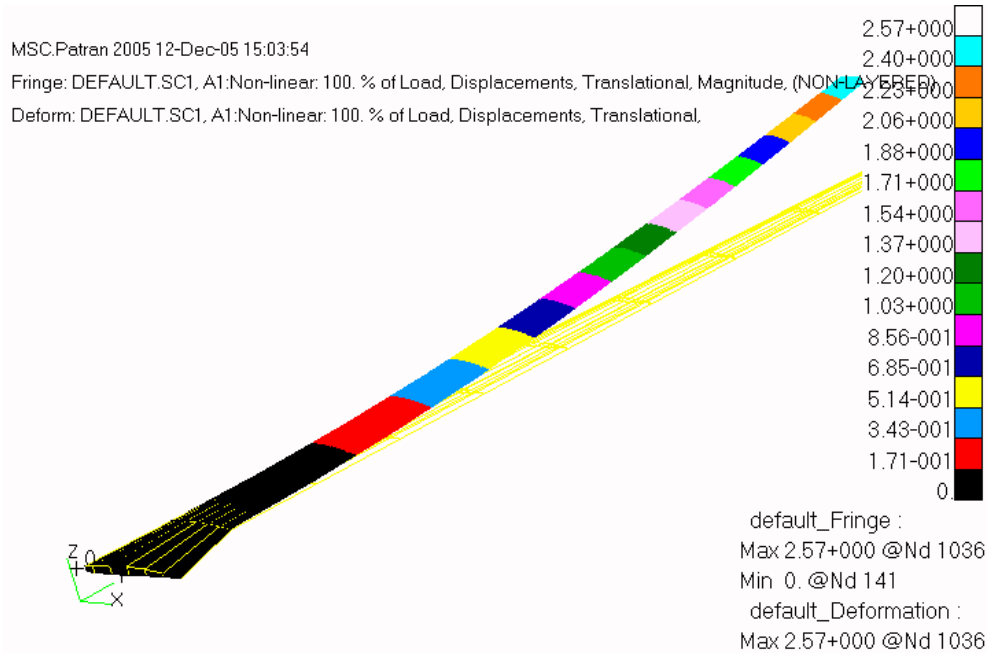


Figure 41: Non-linear Displacement for Global Hawk UAV wing under 2.5g load case.

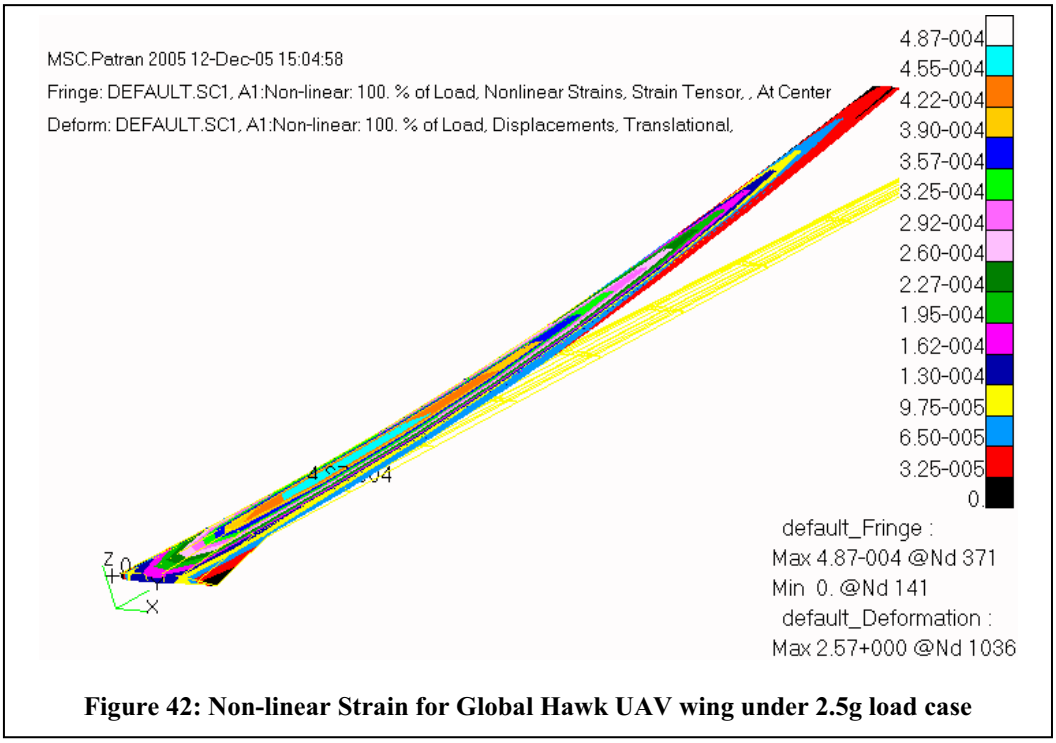


Figure 42: Non-linear Strain for Global Hawk UAV wing under 2.5g load case

The calculations above resulted in the definition of the benchmark for the Global Hawk HALE UAV redesign to be considered in Section 6.2

5. AERO-STRUCTURAL OPTIMISATION

In a general form, the aero-structural optimisation process follows the flow diagram in Figure 43. The Aero-Structural Solver computes the flow dynamics for the planform and structural variables defined by the Evolutionary Optimiser and create the four output files described in Section 5.1.2.

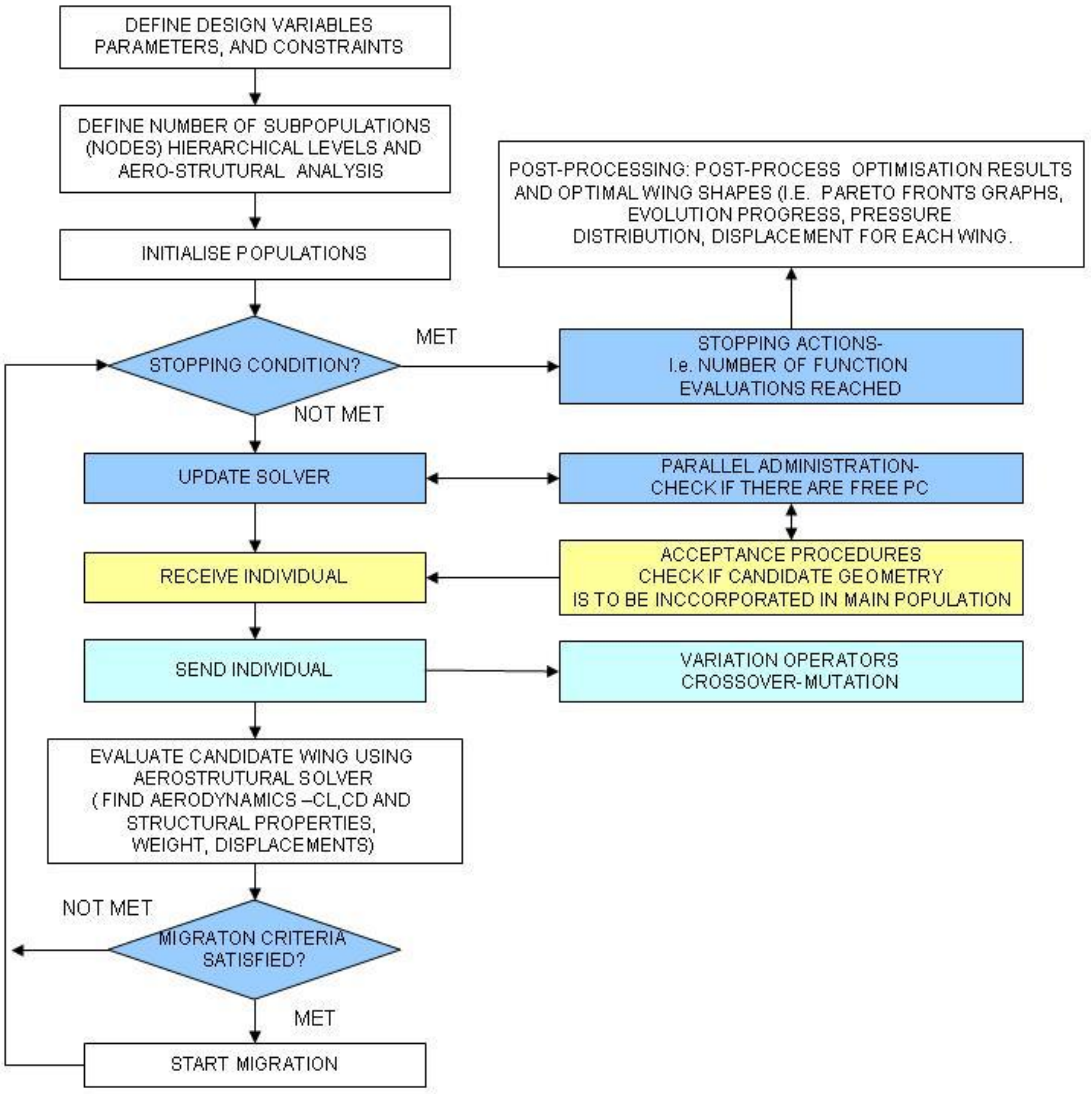


Figure 43: Aero-structural optimiser flow chart

In Figure 43, the Evolutionary Optimiser uses a hierarchical approach with three levels, on the bottom level a coarse analysis is performed to direct the exploration, at the top level a more precise model, that better describes the physics involved, is used and at an intermediate level, a compromised balance between the top and bottom layers is used. Initially the system

specifies the design variables, constraints and parameters and will generate a random sub population of individuals at each layer. It then defines the number of sub populations (nodes) and number of hierarchical levels which for simplicity is equal to the number of analyses being performed. Once these initial populations are generated, the algorithm initialises all the required populations through repeated calls to the Aero-Structural Solver. The scheduler first determines whether the given stopping conditions have been met, and if so, the evolutionary loop is exited and the entire process is stopped. If no stopping conditions are met, the scheduler updates the asynchronous solver so that further progress can be made. The scheduler determines whether or not candidate solutions which have been solved are ready for incorporation into the population. If such solutions exist, the incorporation routine is called and the available candidates, which now have had a fitness assigned (evaluated by the Aero-Structural Solver), are processed; it receives the individual, ages the population and buffer, performs Pareto tournament selection, deletes the oldest individual from the buffer and if the acceptance is true, it is inserted in the population which was subsequently sorted. It then updates the CMA parameters. Finally the scheduler determines whether it is possible to generate more candidates by polling the asynchronous solver. If this is true, the generation routine is called and individuals are generated via the evolutionary operators through recombination and mutation via CMA. The upper and lower bounds are continuously checked throughout the candidate generation routine. During evaluation, the optimiser will take output analyses and parameters to guarantee the satisfaction of constraints and compute the overall fitness function. If the problem is multi - objective, the algorithm will find the non-dominated solutions and will calculate the Pareto fronts associated with those solutions.

As MSC.Nastran® runs on the availability of licenses on a master server, the Aero-Structural Solver was written such that the code checks to see whether or not a license file exists when attempting to analyse a candidate wing with MSC.Nastran®. If a license does not exist, as all the licenses are currently being used by other analyses, the program waits for a license to become available.

This ability to wait for licenses, allows the Evolutionary Optimiser to be run on more machines than there are MSC.Nastran® licenses. This is beneficial as only a fraction of the total time taken to solve each candidate wing solution is used when calculating the structural solution. Therefore, if only the same numbers of computers as MSC.Nastran® licenses are run, there will be instances when MSC.Nastran® licenses are idle. This ability to wait for licenses therefore is beneficial to the user by reducing the overall optimisation time.

5.1 HANDLING OF CONSTRAINTS

Constraints are handled using penalty functions that are broken up into two different categories. The first category penalises the aerodynamics of the wing, and the second the structure. It was found to be beneficial to separate the penalties as it meant that if the optimiser generates a candidate solution which resulted in a light and rigid structure, while the aerodynamics were far from optimum, the structure could still exist within the Pareto set and have its variable values passed onto further generations of candidate solutions. The same consideration applied to the aerodynamics of a candidate wing. If the aerodynamics were found to be near optimal while the wing deflection large with failed structural components due to high strain values, it would slow down the optimiser if the solution were merely discarded.

Therefore, by programming the penalty function to only penalize those components which failed, the overall efficiency of the optimiser is increased.

To further assist the EA when evaluating potential non-dominated solutions, an exponential penalty function is used. This forces penalised wings into regions outside the Pareto front, or off to one extreme favouring only one of the fitness functions. This meant that only wings that were not penalised, or penalised slightly, existed close to the Pareto front. This would not be seen with the use of a linear penalty function as a wing that was penalised for defecting 50% could be seen as a wing with the same weight as a 200% wing.

5.1.1 AERODYNAMICS

Aerodynamic moment

In order to keep the control system of the vehicle relatively unchanged, it was decided to impose penalties on the candidate wing if the aerodynamic moment generated by the wing was greater than the original benchmark tests performed.

The wing is exponentially penalised by increasing the fitness value defined by the inverse of the lift to drag ratio for every percent over the aerodynamic moment benchmark value calculated.

Coefficient of Lift

If the optimiser were to design a wing with aerodynamics such that the inverse of the lift to drag ratio was minimized, and only the aerodynamic moment penalized, the code might evolve a wing which would produce too little lift when placed in the required flight regime. Furthermore, if one were then to alter the angle of attack for the aerofoil to attain the required Coefficient of Lift, the total drag for the aerofoil would change and hence there lies a possibility that the aerofoil would cease to be the optimal configuration.

It was therefore decided to penalise a candidate wing if the lift created was below a set amount. This amount was usually set at ninety percent of the required lift at the start of the cruise phase of the aircraft. As with the Aerodynamic moment penalty, the Coefficient of Lift penalty was added to the fitness function defined by the inverse of the lift to drag ratio at an exponential rate for every percent under the required value. If the Coefficient of Lift calculated for the wing was above the required value, no penalty was imposed.

5.1.2 STRUCTURE

Aerofoil Thickness

The thickness of an aerofoil shape has implications reaching beyond the aerodynamics of the vehicle. If a very thin aerofoil section is designed it might produce excellent aerodynamic results when built and tested, but there would be a very large drawback in that the space between the upper and lower surfaces of the wing would be very small resulting in a cramped and tight working environment. This small working environment would affect the internal layout of the wing and the ease of component inspection.

Therefore the thickness of each aerofoil must exceed a value of 12% ($t/c \geq 0.12$). If the Evolutionary Optimiser produces a candidate aerofoil that has a smaller thickness value, the aerofoil would be penalised by equally penalising both the aerodynamic and structural values via an exponential penalty method. In addition, aerofoil sections generated by the Evolutionary Optimiser that had thicknesses outside some bounds, usually between 10% and 15%, are rejected immediately before any further analysis is performed on them.

Buckling of Wing Skin Panels

Thin sheets of material under compression and or shear loads are subject to buckling. When a sheet of material buckles under the applied load, the magnitude of the strength with which the sheet of material is able to withstand reduces significantly.

An accurate method of predicting whether or not any of the elements within a candidate wing would be subject to buckling would be to perform a buckling analysis in MSC.Nastran®. The additional computational overhead would be rather large increasing the time required to determine the fitness of each candidate wing. This would further extend the time required by HAPMOEA to find a global optimal solution given the simulation bounds. It therefore became appropriate to find another method to determine whether or not any elements within a candidate wing would buckle. As the Spars and Ribs were of considerable thicknesses compared to the wing skin, and also since the loads and moments carried by these two structural elements lay in plane with the elements as in Figure 44*a*, the possibility of any of the sections making up the Spars and Ribs buckling would be very small compared to the possibility of the skin panel sections buckling. This is due to the fact that the applied loads and moments across the Skin sections lie out of plane as in Figure 44*b*.

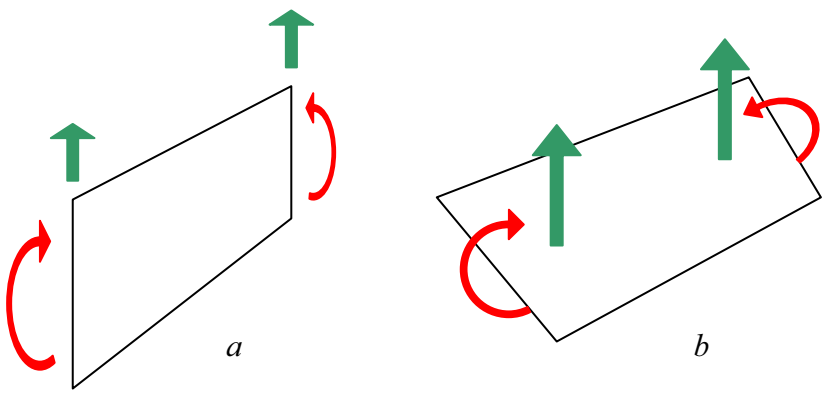


Figure 44: In and Out of Plane Section Loading

Following the method followed by Venter and Sobieszczanski-Sobieski [60], simple analytical expressions are used to test whether or not the stresses in the Wing Skin sections are large enough to cause buckling. These local buckling expressions are described by Eqn. 11.

$$\frac{\sigma_1}{\sigma_{cr}} - 1 \leq 0$$
$$\frac{\tau_1}{\tau_{cr}} - 1 \leq 0 \qquad (11)$$
$$\frac{\sigma_1}{\sigma_{cr}} + \left(\frac{\tau_{12}}{\tau_{cr}}\right)^2 - 1 \leq 0$$

Where:

$$\sigma_1 = \frac{N_1}{2t}, \sigma_{cr} = \frac{3.6kE}{\left(\frac{b}{t}\right)^2}$$
$$\tau_{12} = \frac{N_{12}}{2t}, \tau_{cr} = \frac{4.85kE}{\left(\frac{b}{t}\right)^2}$$

and b is the width of each Wing Skin section and k was an additional safety factor to guard against delamination of the plies making up the carbon graphite composite material. The safety factor was set at four for all the optimisations. The subscript cr denotes the critical value for each panel before the onset of buckling.

Tip Twist

As the wing bends, the pressures generated by the wing would change and hence the forces within the wing would also change. If the wing tip twisted, there is a possibility that the wing would greatly increase the drag created by the wing, and also increase the moment about the *Z*-axis. As a full aero-elastic analysis was not part of the simulation runs, any further increase in the angle at which the wing would make with the freestream flow would be disadvantageous to the computation as the final wing position and internal forces could not be computed.

It was therefore necessary to penalise wings in which after the structural simulation using MSC.Nastran®, the wing tip angle of attack was greater than a user set value from the initial wing setting angle. This penalty took the form of mass added to the structure at the rate for each additional degree.

Deflection

As no full aero-elastic analysis is performed on each candidate wing, the final aerodynamic and structural solution is not known. Therefore the assumption is made that if the initial deflection of the wing was small, the wing would have a smaller chance of aerodynamic divergence if the aerodynamic and structural simulations were performed until convergence between the results is reached. Although this might not be the case for certain wings, it was simple and easy to implement and placed a far smaller overhead on the computational cost compared to the analysis of the mass matrix for each candidate wing.

The wing was therefore allowed to deflect a set percentage of the span for a 2.5g load case before a penalty value was added to the fitness value. As with the Tip Twist condition, the penalty took the form of additional mass and was added per percent deflection over the user set percentage.

Failure of Internal Component Panels

In this aero-structural analysis the testing of structural members for failure is performed through the use of the maximum strain theory. This is done as stress varies between plies and orientations in a composite material, while strain varies linearly through the thickness. This same method was employed in Venter and Sobieszczanski-Sobieski [60].

If any components making up the internal structure within the wing, spars, ribs, skin, etc fails due to excessive strains, the wing is heavily penalised by the addition of extra wing mass. This mass is increased at a user set rate for each panel making up a component that failed.

5.1.3 COMBINED PENALTIES

If a wing is analysed and the penalties are very large against a user set maximum, instead of only penalising one of the fitness functions, both fitness functions would be penalised according to the maximum penalty associated with either of the calculated values. This forces the solution to be dominated and hence the genetic information is not passed on to any further generations of candidate wing solutions.

The above-described code runs in the Matlab environment and is fed inputs and outputs results through formatted text files. The code is robust and easily integrated as a module into the HAPMOEA code architecture.

6. AERO-STRUCTURAL OPTIMISATION TEST CASES

This chapter considers the application of the optimisation methodology for two test cases related to UAV design. The first test case is for a Medium Altitude Long Endurance (MALE) UAV and the second for a High Altitude Long Endurance (HALE) UAV.

6.1 MEDIUM ALTITUDE LONG ENDURANCE UAV DESIGN AND OPTIMISATION

This first test case considers the multidisciplinary, multi-objective design optimisation of a MALE UAV Wing similar to the Altair MALE UAV considered in Section 5.4.1. The two objectives are the maximisation of the lift to drag ratio (L/D), and the minimisation of wing weight. The cruise Mach number and altitude are 0.3663 and 12680 *m* respectively and the wing area is set to 29.21 m^2.

6.1.1 DESIGN VARIABLES AND CONSTRAINTS.

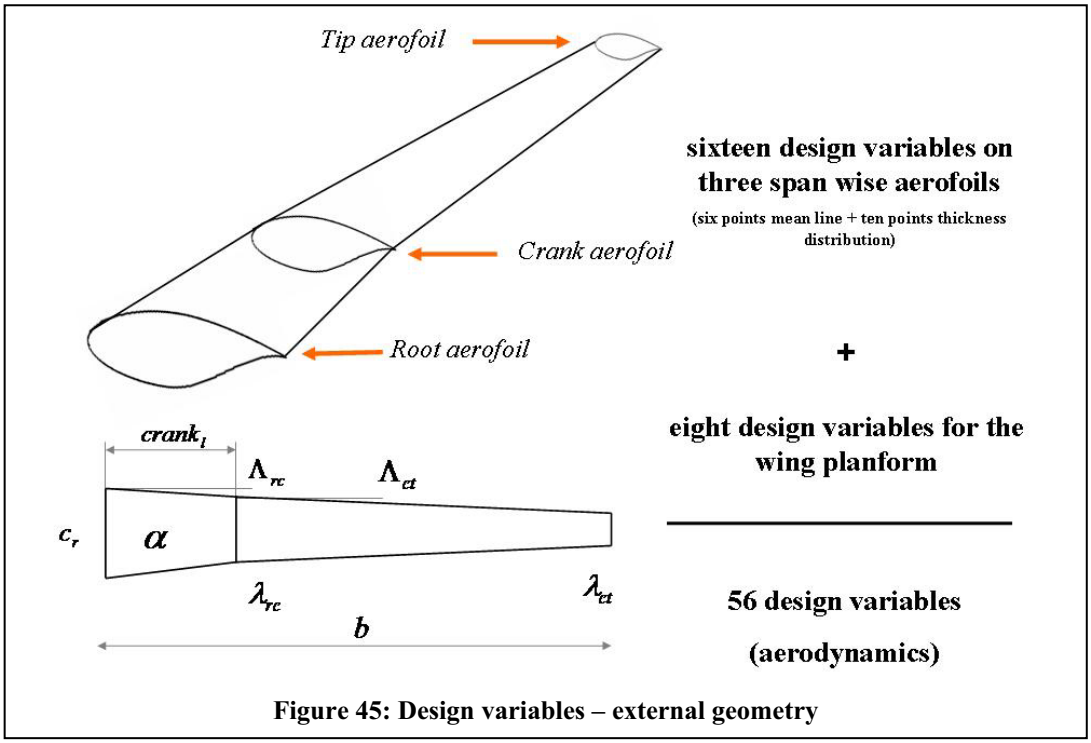


Figure 45: Design variables – external geometry

The wing geometry is represented by three aerofoil sections with sixteen variables for each section, eight variables that describe the wing planform and eleven variables to describe the internal structure.

The aerofoil geometry is represented by the combination of a mean line and thickness distribution, which is a very common procedure in classical aerodynamics [65]. Both lines are represented by Bézier curves with leading and trailing edge points fixed at (0.0,0.0) and (1.0,0.0) respectively, and a variable number of intermediate control points whose x-positions are fixed in advance and whose y-heights form the problem unknowns. In this case 6 free control points on the mean line and 10 free control points on the thickness distribution are taken.

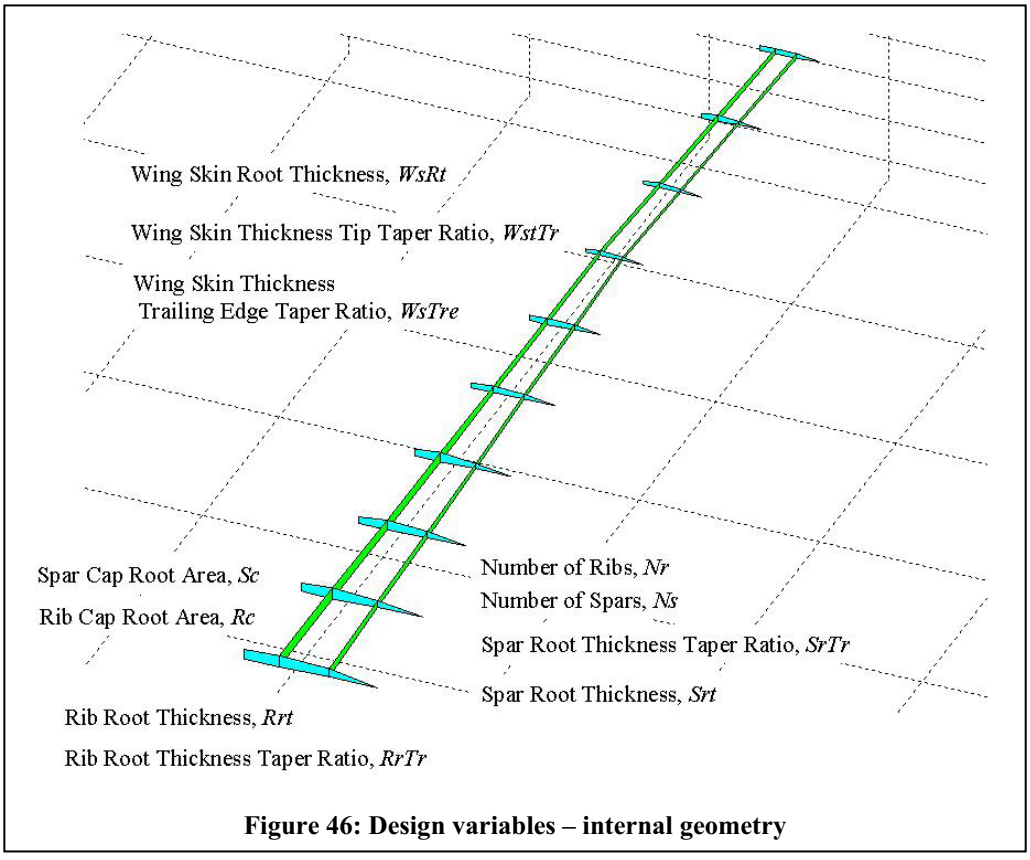


Figure 46: Design variables – internal geometry

The wing planform design variables are indicated in Figures 45 and 46 and their upper and lower bounds are represented in Table 9. In total 67 design variables are used for the MALE UAV wing optimisation.

Description	Benchmark	Lower Bound	Upper Bound
Angle of Attack, *(deg)*	4	0	6
Wing Semi Span, *(m)*	12.542	8.2	18.5
Root Chord, *(m)*	1.646	1.6	1.7
Wing Inboard Leading Edge Sweep, *(rad)*	0.039	0	5
Wing Outboard Leading Edge Sweep, *(rad)*	0.039	0	5
Chord Ratio at Break	0.444	0.35	0.65
Chord Ratio at Tip	0.444	0.35	0.65
Crank Location *(m)*	0.731	0.5	0.8
Number of Spars	2	1	3
Number of Ribs	10	8	12
Rib Root Thickness, *(m)*	0.005	0.003	0.0065
Rib Root Thickness Taper Ratio	0.25	0.15	0.3
Spar Root Thickness, *(m)*	0.08	0.05	0.09
Spar Root Thickness Taper Ratio	0.05	0.01	0.2
Wing Skin Root Thickness, *(m)*	0.001	0.0005	0.002
Wing Skin Thickness Tip Taper Ratio	0.1	0.05	0.15
Wing Skin Thickness Trailing Taper Ratio	0.1	0.05	0.15
Spar Cap Root Area, *(m^2)*	0.00375	0.0025	0.04
Rib Cap Root Area, *(m^2)*	0.0015	0.001	0.002

Table 9: Structural and Planform Design Variables for the MALE UAV wing

6.1.2 Design Constraints and Penalties

There are several constraints that must be satisfied for each candidate wing namely:

- The pitching moment must not be greater than -0.1768.
- The calculated coefficient of lift must be greater than 0.64.
- The thickness of each aerofoil must exceed 12% ($t/c \geq 0.12$).
- No internal or external structural sections may buckle or fail due to excessive strains.
- The wing tip may not twist more than one degree.
- The wing may not deflect more than twenty percent of the span.

All the above mentioned constraints are applied by penalising either, or both fitness values via an exponential penalty method as described in Section 6.1. In addition, any aerofoil generated outside the thickness bounds of 10% to 15% is rejected immediately before any analysis is performed on the wing. The above data is summarised in Table 10.

Description	Values
Allowable Strain	0.00333
Allowed wing tip twist in degrees	1
Allowed wing deflection as a percentage span	20
Allowed wing moment (benchmark)	-0.1768
Minimum lift to be generated by the wing	0.64
Wing Mass per degree twist of wing tip	0.1
Wing Mass per deflection percent over 20	0.1
Wing Mass per failed panel	0.1
Additional C_D per over allowable aerodynamic moment	0.001
Additional C_D per less than the required minimum C_L	0.005
Penalise both fitness functions above a Drag ratio of	0.1
Penalise both fitness functions above a Mass ratio of	0.1

Table 10: Constraint and Penalty Values

6.1.3 FITNESS FUNCTIONS

The two fitness functions to be optimised are defined as Eqn 12 and Eqn 13:

$$\min(f_1): \quad f_1 = \frac{1}{(L/D)} + penalty \tag{12}$$

$$\min(f_2): \quad f_2 = W + penalty \tag{13}$$

Eqn 12 relates to the aerodynamics of the candidate wing. The Evolutionary Optimiser seeks to minimise the inverse of the lift to drag ratio with the addition of penalties which are described in Section 6.1.

Eqn 13 relates to the structural aspects of the candidate wing. This function has the units of mass and seeks to minimise the calculated structural mass of the wing. Penalties are also added to Eqn 130 which are described in Section 6.1

For both Eqns 12 and 13, the penalties are calculated based on the constraints specified in Section 7.1.2.

6.1.4 AERO-STRUCTURAL ANALYSIS

The aerodynamic and structural characteristics of the wing configurations are evaluated using the Aero-Structural Solver described in Section 5.1.

6.1.5 IMPLEMENTATION

The solution to this problem has been calculated using two approaches; the first one uses a traditional EA method with a single population model based upon a fine meshed aerodynamic model of each candidate wing. Between 1904 and 2992 aerodynamic panels are used to define the candidate wings using a fine mesh structure. The second method uses a hierarchical topology of mesh resolutions comprising three levels. The same aerodynamic resolution as is used in the single population experiments is used for the top level, a more coarse mesh model used in an intermediate aerodynamic level (between 1428 and 2244 aerodynamic panels), and a coarse mesh (between 952 and 1496 aerodynamic panels) used in the lowest model to explore the design search space.

This results in the coarse mesh solutions requiring roughly 4 minutes to solve for the aerodynamic pressures about a candidate wing, and 15 minutes for a fine mesh. An intermediate wing takes roughly 10 minutes on a single 2.4 GHz machine.

Three P4, 2.4 GHz machines were used in the calculation and the population size for this problem is set to 30 in all levels. This was set for both the single and hierarchical solutions.

For both solution approaches, the single population and the hierarchical method, the structural model panel density was kept as a function of the internal geometry and was not based upon the level of model complexity. For a description of the method used, please refer to Section 5.3.

6.1.6 RESULTS

The algorithm was run for 500 function evaluations and took 36 hours to compute on a cluster of three machines. Figure 47 shows the Pareto fronts obtained by using the two approaches. The Pareto members selected for further investigation are also indicated. By inspecting Figure 47 it can be seen that the use of a multi-fidelity (hierarchical) approach gives an overall lower

front as compared to a single model approach. Both approaches give results that lie to the left of the benchmark value for the MALE wing.

The algorithm progress for the hierarchical approach is shown in Figure 48. The fitness for all levels decreases with time and it can be seen that after 350 evaluations, the fitness of each level reaches a steady state value indicating an optimum solution given the constrains imposed.

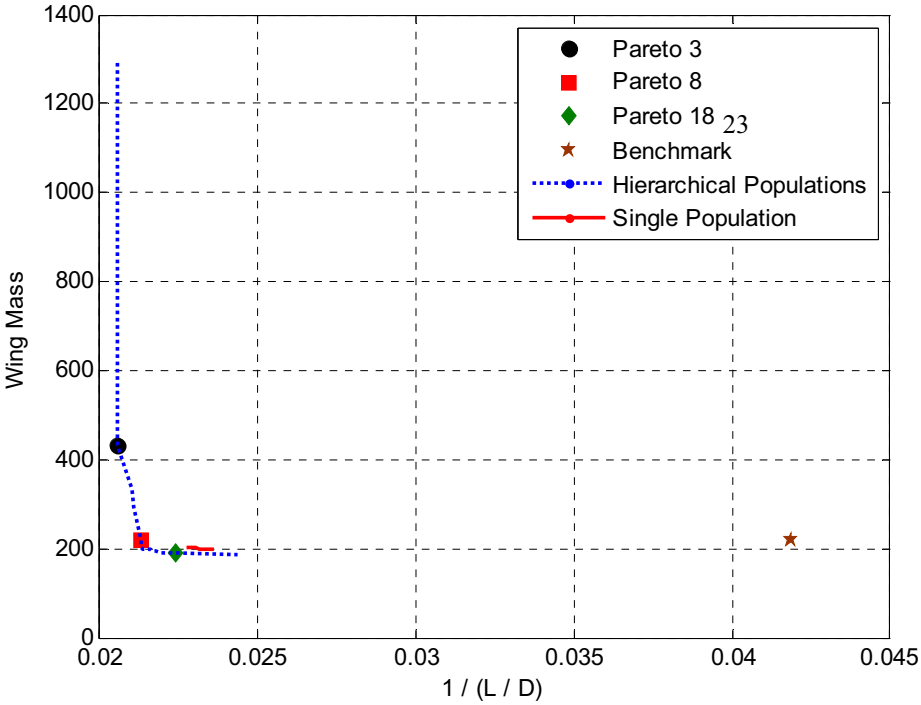


Figure 47: Pareto Fronts for MALE UAV Optimisation

The optimisation was run for a second time but with only one population to test the validity of the hierarchical procedure employed by the HAPMOEA algorithm. The results are shown in Figure 49.

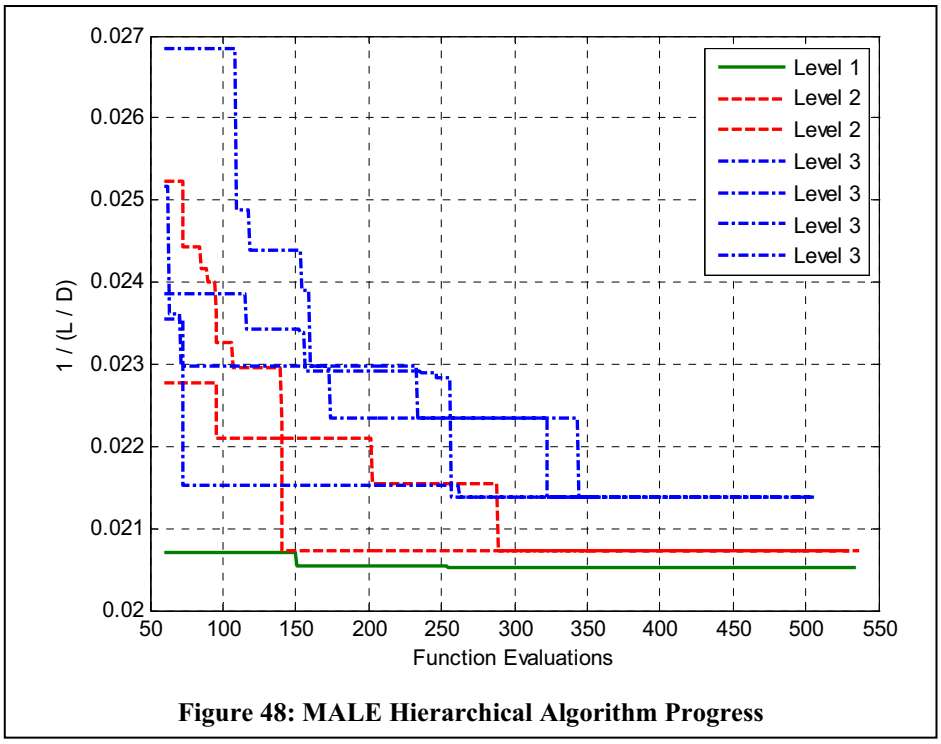


Figure 48: MALE Hierarchical Algorithm Progress

The algorithm progress shown in Figure 49 relates to the Pareto front shown as a solid red line on Figure 47.

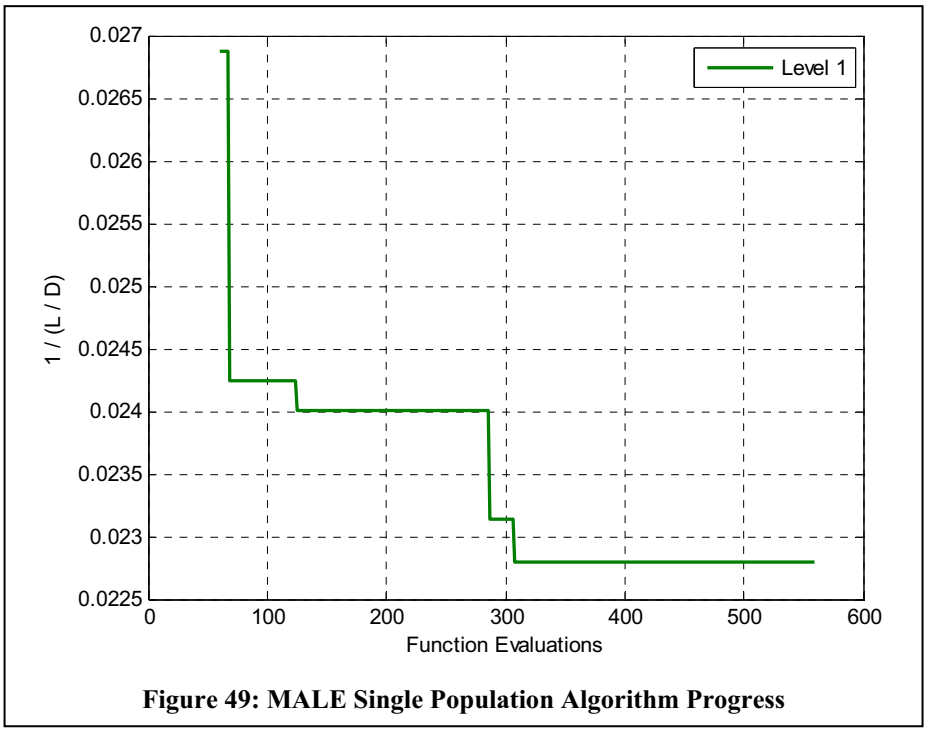


Figure 49: MALE Single Population Algorithm Progress

From Figures 47 to 49 it can be seen that the performance for the Hierarchical scheme is higher than that of the single population scheme over the same number of function evaluations of the Level 1 node. The hierarchical approach therefore yields better solutions for the same number of function evaluations of the Level 1 node.

Three wings are taken for further evaluation (Pareto members 3, 8 and 23) from the hierarchical optimisation strategy results to illustrate the two objective extremes and a compromise between the two results. According to the optimiser, Pareto 3 has the best aerodynamics, Pareto 23 the best structural response to the aerodynamic loading and Pareto 8 lies between the two solutions.

Table 11 summarises and compares the values for the design variables and the objective functions for these three geometries against the benchmark.

Description	**Benchmark**	**PM 3**	**PM 8**	**PM 23**
Angle of Attack, *(deg)*	4	4.055	3.957	5.16798
Wing Semi Span, *(m)*	12.542	13.871	13.385	12.4694
Root Chord, *(m)*	1.646	1.638	1.641	1.66618
Wing Inboard Leading Edge Sweep, *(rad)*	0.039	0.0871	0.0694	0.0801
Wing Outboard Leading Edge Sweep, *(rad)*	0.039	0.0255	0.0486	0.0105
Chord Ratio at Break	0.444	0.422	0.448	0.427
Chord Ratio at Tip	0.444	0.475	0.388	0.477
Crank Location *(m)*	0.731	0.583	0.667	0.787
Number of Spars	2	1	2	2
Number of Ribs	10	11	10	9
Rib Root Thickness, *(m)*	0.005	0.00351	0.00465	0.00435
Rib Root Thickness Taper Ratio	0.25	0.182	0.287	0.243
Spar Root Thickness, *(m)*	0.08	0.0805	0.0576	0.0571
Spar Root Thickness Taper Ratio	0.05	0.131	0.154	0.019
Wing Skin Root Thickness, *(m)*	0.001	0.000885	0.000953	0.00199
Wing Skin Thickness Tip Taper Ratio	0.1	0.122	0.1	0.0541
Wing Skin Thickness Trailing Taper Ratio	0.1	0.106	0.0608	0.108
Spar Cap Root Area, *(m^2)*	0.00375	0.00908	0.00314	0.00314
Rib Cap Root Area, *(m^2)*	0.0015	0.00123	0.00175	0.00105

Table 11: Summary and comparison of design variables for MALE UAV

Figure 50 shows a top view of the wings selected for further analysis and Figures 51 to 53 compare the baseline aerofoils and the aerofoil sections of each Pareto member at the root, crank and tip.

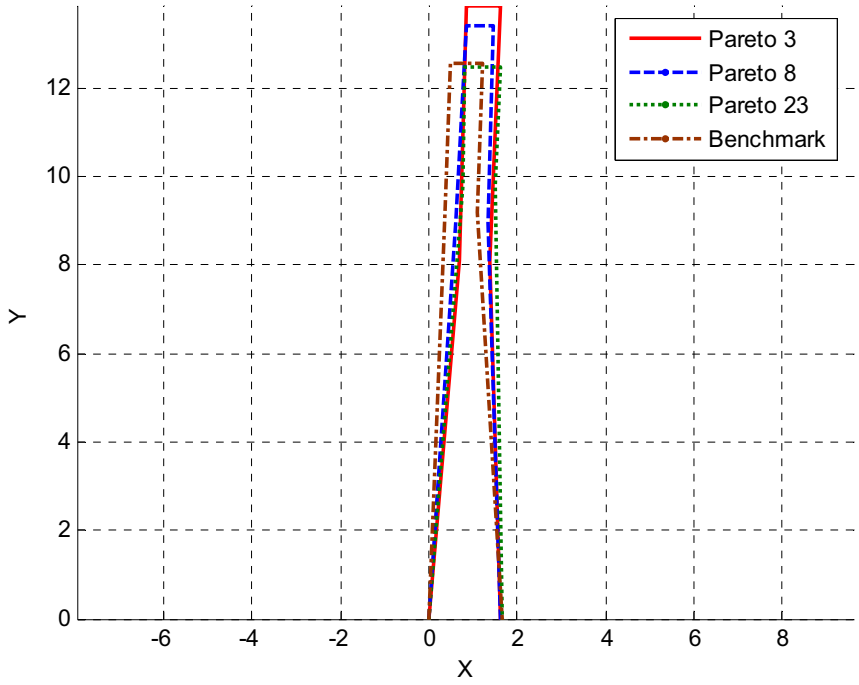


Figure 50: Top Views of MALE Pareto members selected for further analysis

On inspecting Figure 50, the optimiser determined that the original layout of the wing favoured the structural constraints more than the aerodynamics and changed the wingspan and crank location between the different Pareto members to achieve better aerodynamics.

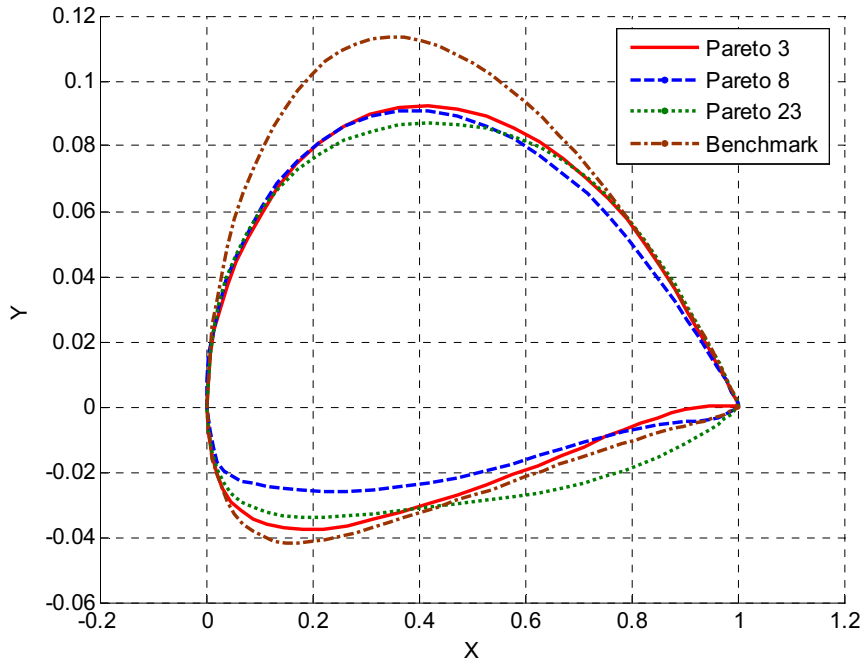


Figure 51: Selected Normalised MALE Pareto member Root Aerofoils

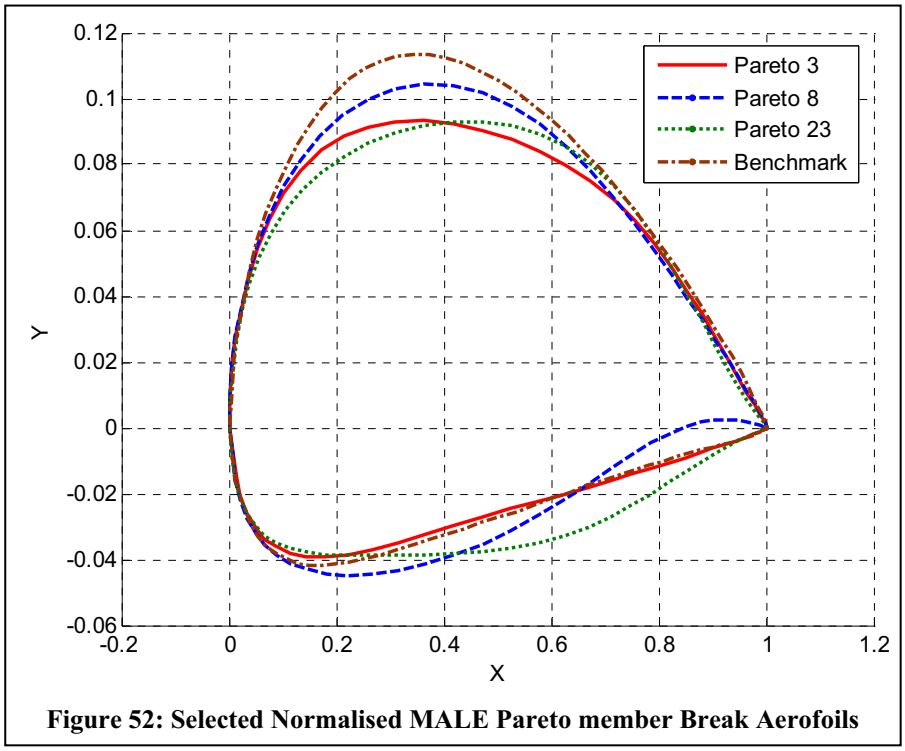


Figure 52: Selected Normalised MALE Pareto member Break Aerofoils

It can be seen that classical aerofoil shapes for the wings have been evolved, even considering that the optimisation was started completely from random and the evolution algorithm had no problem specific knowledge of appropriate solution types.

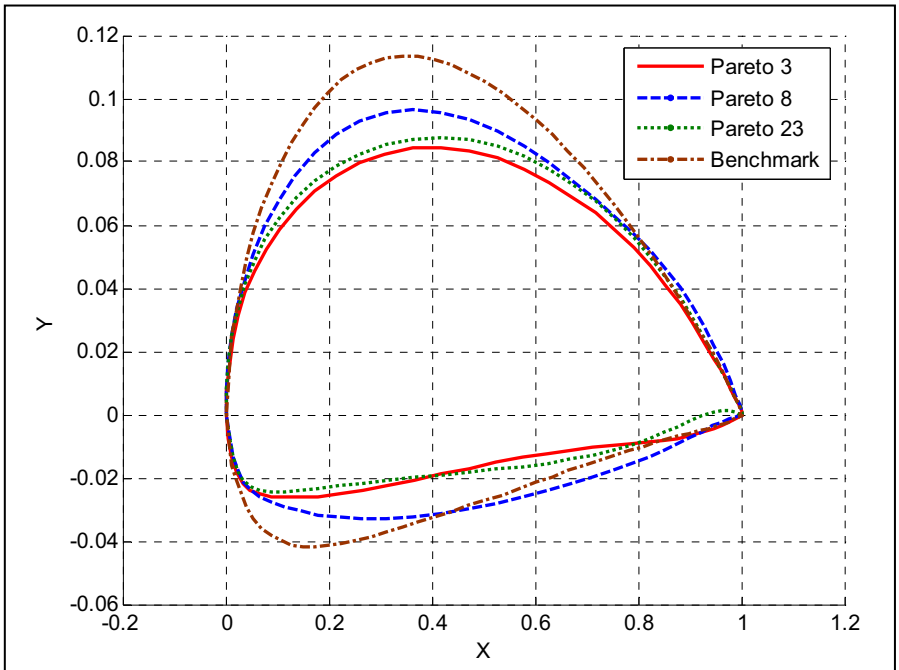


Figure 53: Selected Normalised MALE Pareto member Tip Aerofoils

Of interest is that all the aerofoil sections generated by the HAPMOEA code and shown in Figures 51 to 53 have a lower *t/c* ratio than that of the benchmark aerofoil section. There are only slight differences between the root aerofoil sections describing the Pareto members selected in Figure 51.

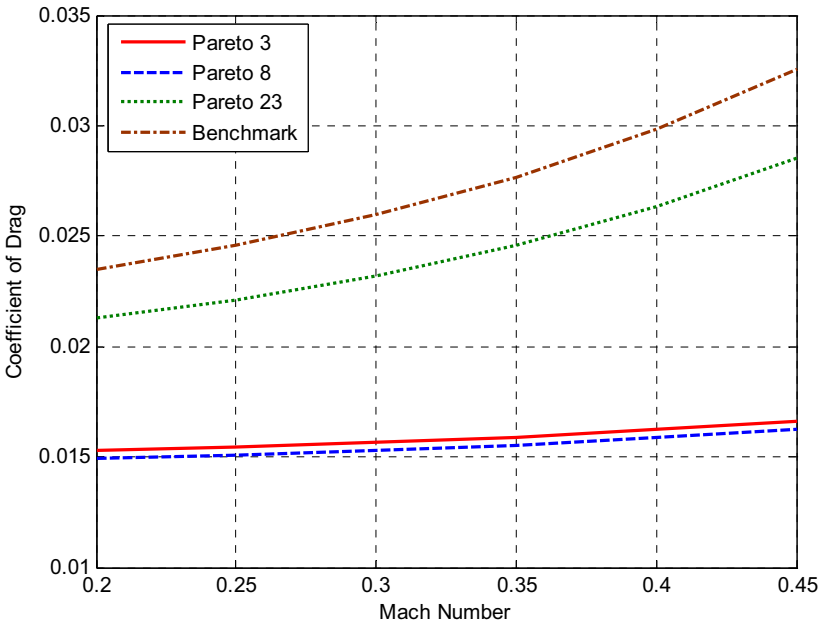


Figure 54: C_D vs. Mach number for selected MALE Pareto members and Benchmark

A Mach number sweep was performed at a constant angle of attack for the selected Pareto members and the results are displayed in Figure 54 for coefficient of drag and Figure 55 for coefficient of lift. The benchmark wing is included for comparison.

Over the range of Mach numbers investigated, the Pareto members have lower drag coefficient than the benchmark wing. The Pareto members three and eight operate at an almost constant drag value over the range of Mach numbers.

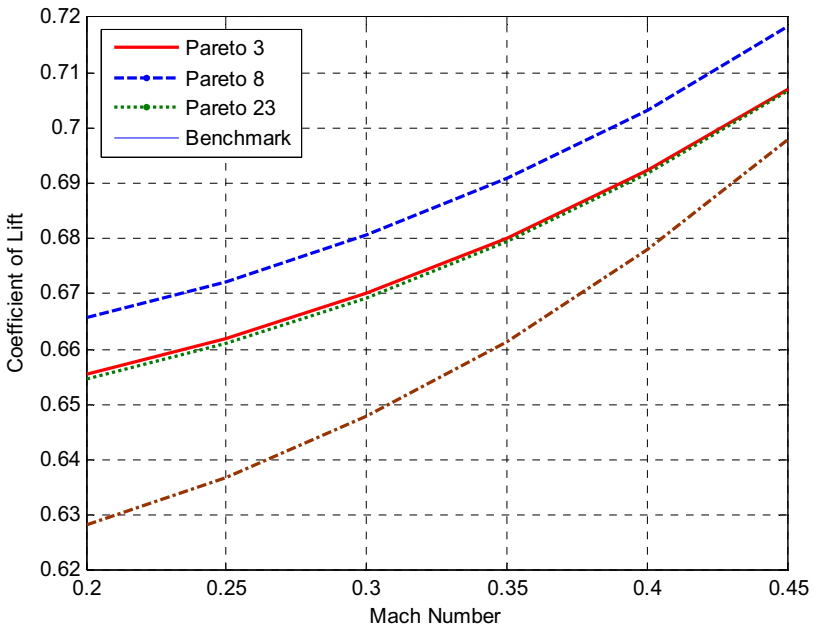


Figure 55: C_L vs. Mach number for selected MALE Pareto members and Benchmark

As the Lift penalty function described in Section 6.1.1 only penalised candidate wing that produced a lift coefficient lower than the required value to maintain the cruise altitude and velocity, the selected Pareto member wings all produce more than the required lift. This extra lift value is small through compared to the overall required force, being roughly 0.02 for Pareto member eight compared to the benchmark.

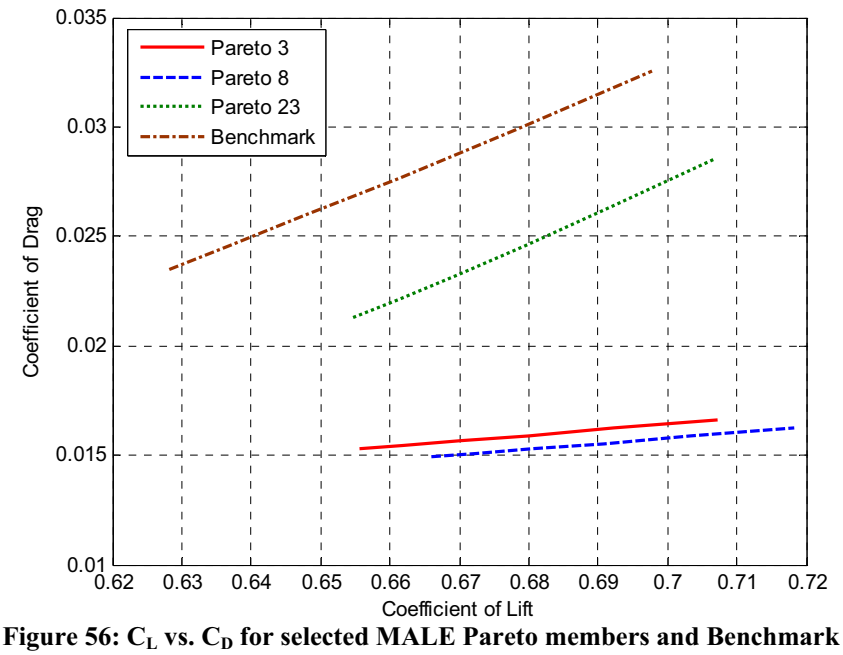


Figure 56: C_L vs. C_D for selected MALE Pareto members and Benchmark

When the Coefficient of Lift is plotted versus the Coefficient of Drag for the selected Pareto members and the benchmark MALE wing in Figure 56, one can see that the performance of all the wings belonging to the Pareto front is higher than that of the benchmark wing. The required Coefficient of Lift of 0.64 is met at a lower Coefficient of Drag with all the selected Pareto member wings.

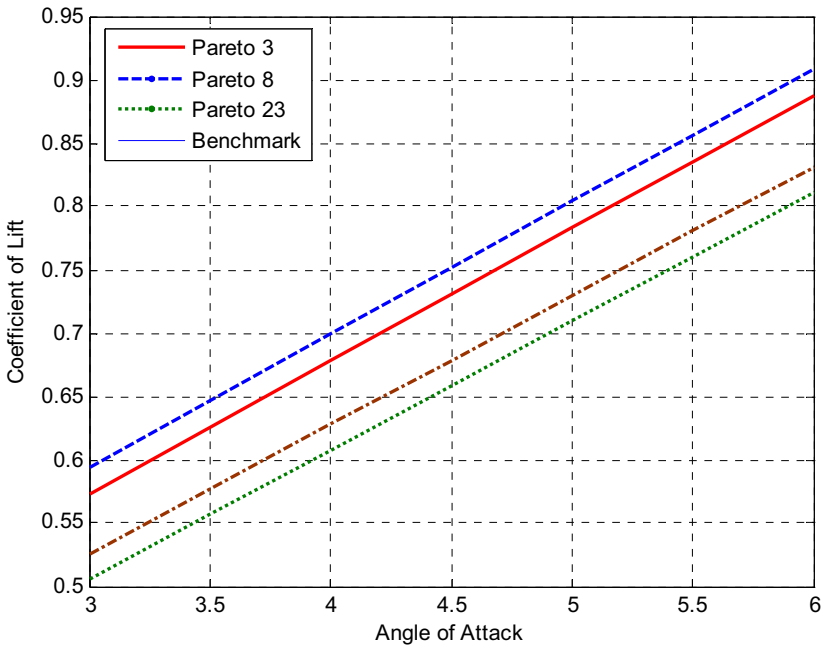


Figure 57: C_L vs. Angle of Attack for selected MALE Pareto members and Benchmark

The selected MALE UAV Pareto member are further analysed by performing an Angle of Attack sweep between three and six degrees and displayed in Figures 57 and 58. For all the analyses performed, the Mach number was kept constant at 0.3663, the assumed operating velocity of the benchmark vehicle.

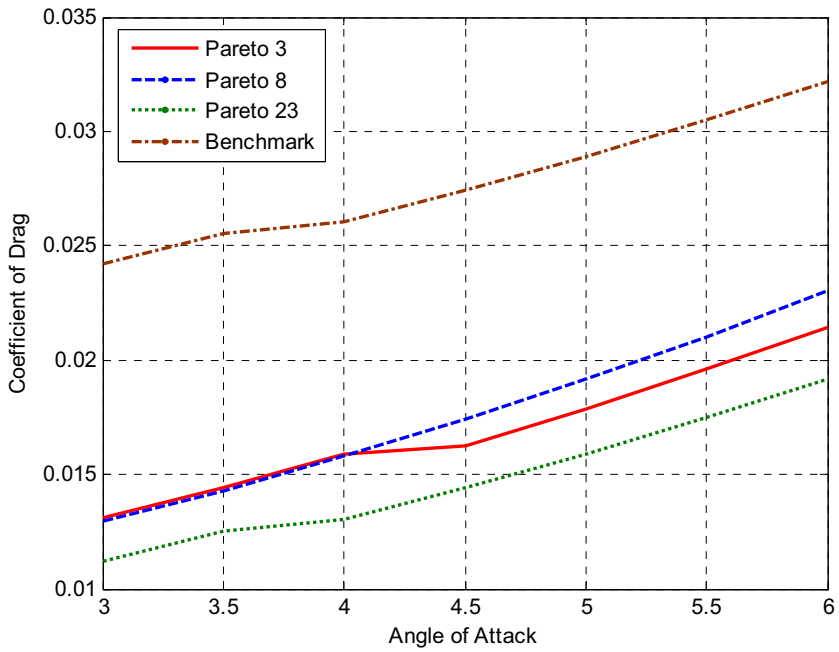


Figure 58: C_D vs. Angle of Attack for selected MALE Pareto members and Benchmark

For all the wings tested, the Coefficient of Lift shown in Figure 57 varied in almost a linear fashion over the angles tested. Furthermore, all the wings behaved in a similar fashion in that the gradients of the Lift slopes are all very similar.

As with the Mach sweep performed and displayed in Figure 54, the selected Pareto members displayed and tested in Figure 58 have a lower Coefficient of Drag over all tested angles of attack. Between the angles of three and four, Pareto 23 and the bench exhibit the same reduction in drag. Pareto eight does not exhibit a similar drop over the angle range covered while Pareto three exhibits the reduction between four and four and a half degrees.

Figures 59 and 60 plot the second order pressure coefficients for the selected Pareto wings.

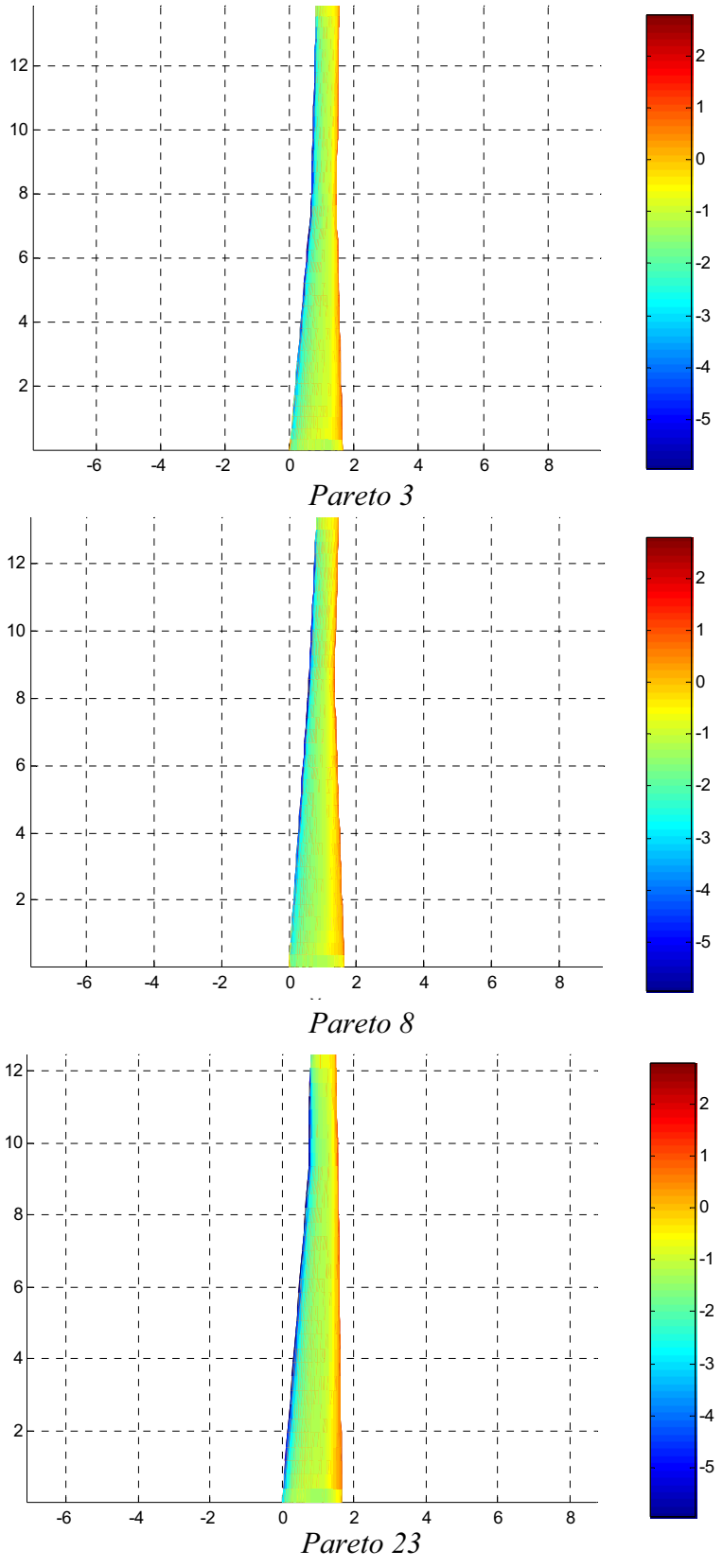
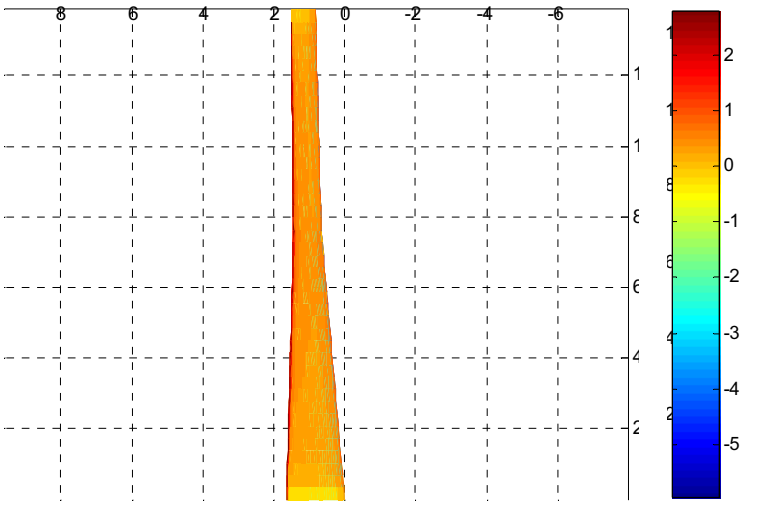
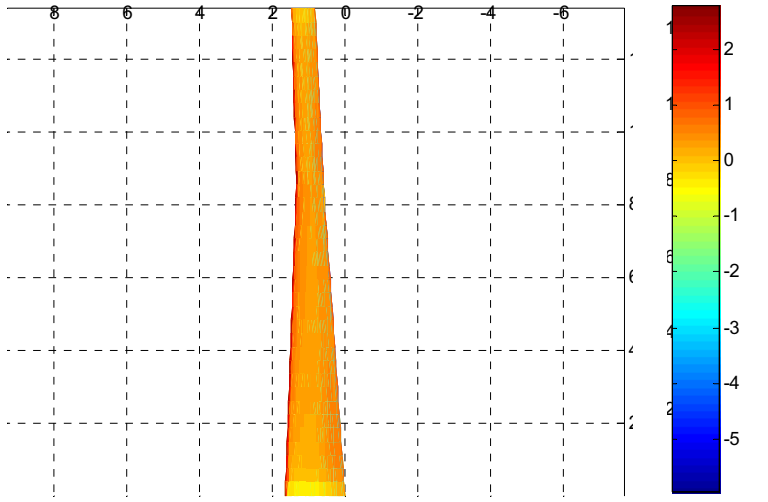


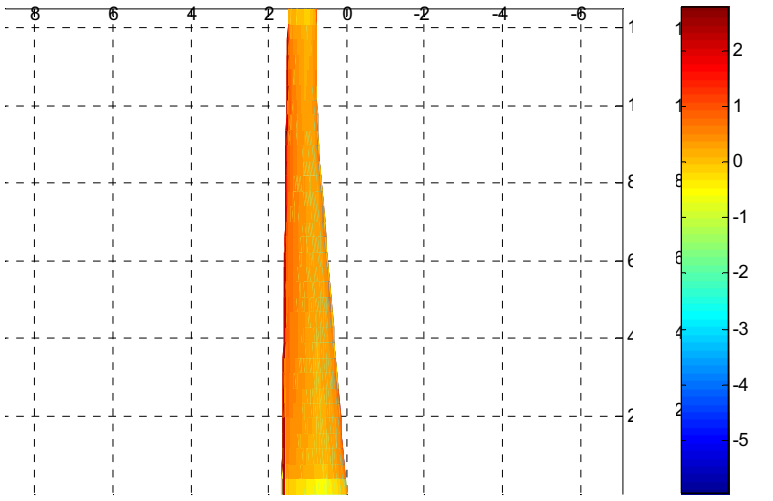
Figure 59: C_P distribution for selected MALE Pareto Members – Upper Surface



Pareto 3



Pareto 8



Pareto 23

Figure 60: C_P distribution for selected MALE Pareto Members – Lower Surface

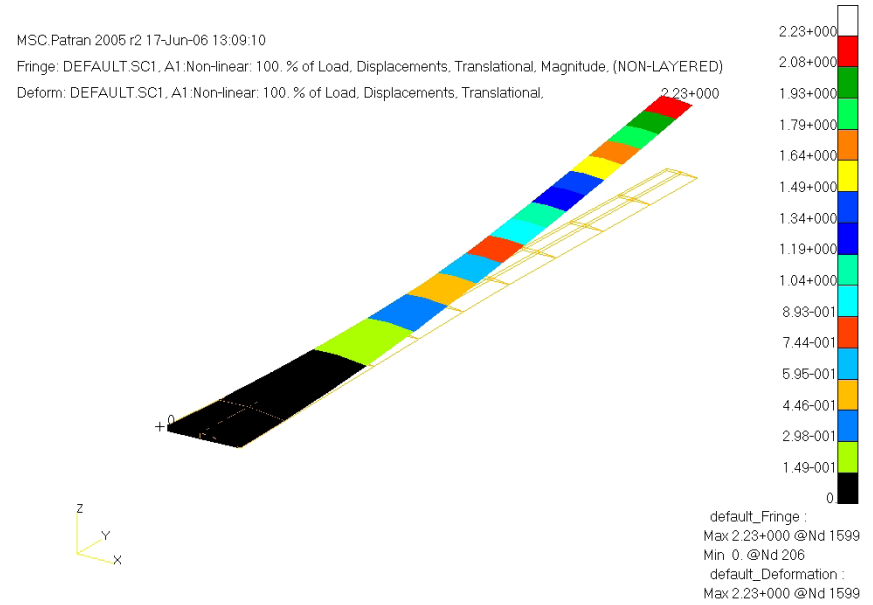


Figure 61: MALE Pareto 3 Displacement under 2.5g load case

In comparison with the MALE UAV benchmark wing pressure coefficients displayed in Figures 27 and 28, the wings selected from the Pareto set produce higher pressures along the leading edge of the wings. The maximum coefficient of pressure for the benchmark wing is negative three while the Pareto members all produce coefficients of pressure as high as negative five. This could be due to the smaller *t/c* ratio of the Pareto aerofoils evident in Figures 51 to 53. The pressures generated along the lower surface of the Pareto wings are also smaller in magnitude than that of the benchmark.

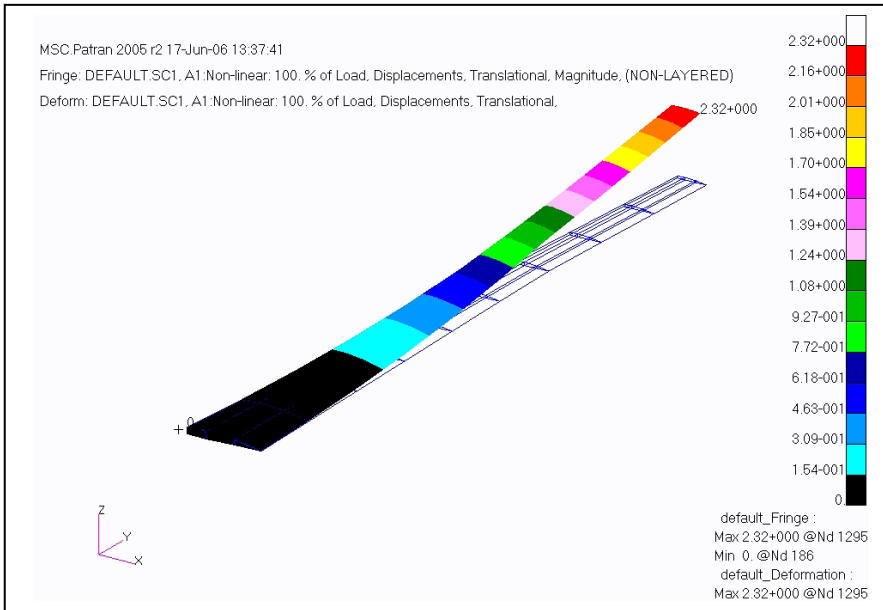


Figure 62: MALE Pareto 8 Displacement under 2.5g load case

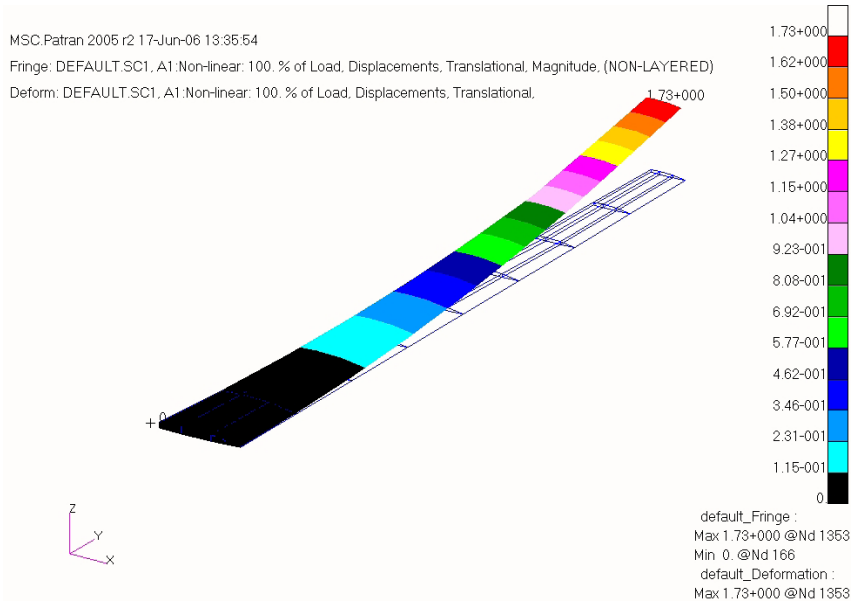


Figure 63: MALE Pareto 23 Displacement under 2.5g load case

A larger resultant lift force is therefore created by the Pareto member wings and this is in good agreement with the high performance of the wings noted when the Coefficients of Lift and Drag were plotted in Figure 56.

Figures 61 to 63 show the deflected Pareto wings. The original wing position and internal structure is shown as a wire frame for comparison.

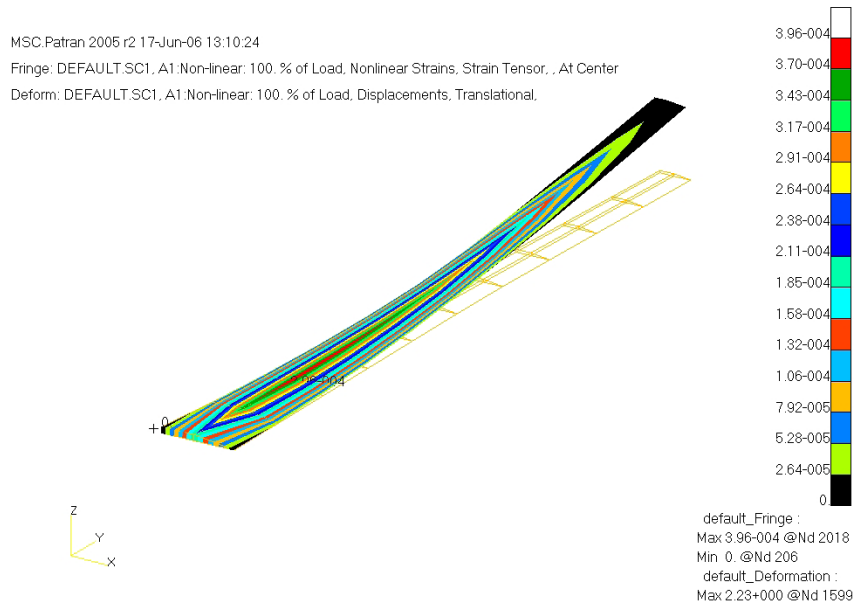


Figure 64: MALE Pareto 3 Strain due to Displacement

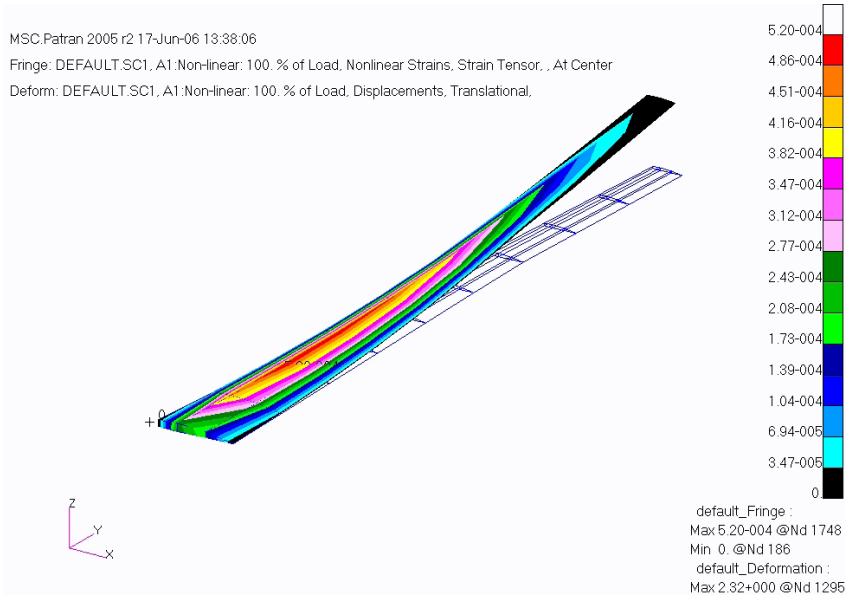


Figure 65: MALE Pareto 8 Strain due to Displacement

All the MALE wings deflect nonlinearly. From the Figures above it can be seen that the deflection at the wing tips is greater than the thickness of the aerofoil sections and hence wing spars. The non-linear analysis solutions, selected for the MSC.Nastran® structural calculations, is therefore necessary as the deflections encountered are too large for a Linear Static analysis and if used, would have yielded incorrect results.

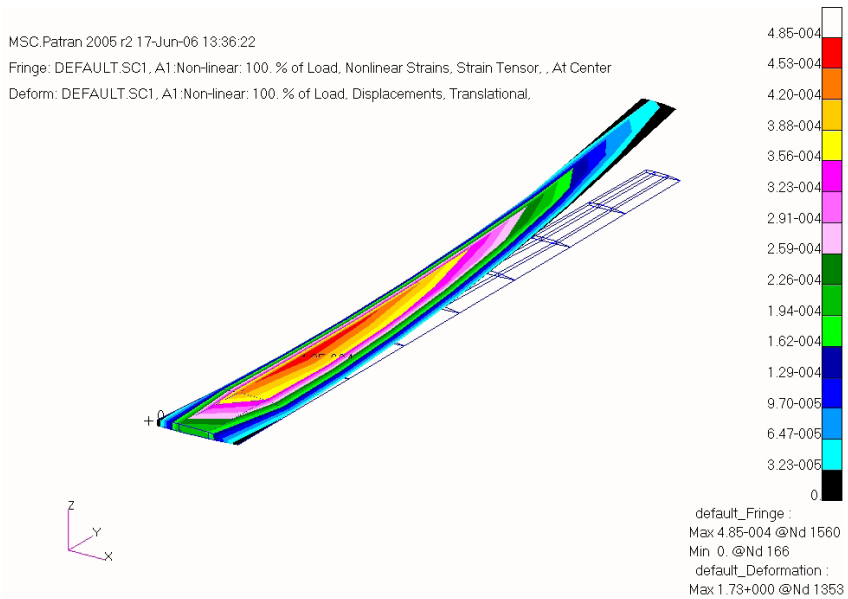


Figure 66: MALE Pareto 23 Strain due to Displacement

An analysis of the strains produced by the aerodynamic forces is shown in Figures 64 to 66.

The aerodynamic load generated by the aerofoil sections cause the candidate wings to want to rotate about the wing root (X-axis). As the wings were fixed in all directions and all rotations at the wing root, the maximum strain does not occur at the wing root but rather a distance from the root, where the rotation occurs.

	Pareto 3	Pareto 8	Pareto 23	Benchmark
Deflection (%)	16	17.3	13.8	12.5
Maximum Strain	0.000396	0.00052	0.000485	0.000403
Mass (kg)	233.735	217.873	185.1	220.9
L/D	43.367	44.4	44.25	23.9
Mass Penalty (kg)	40	0	0	0
Drag Penalty	0.0027	0	0	0

Table 12: Summary of Displacement, Strain and Mass for Selected MALE Pareto members and the Benchmark

A summary of each of the Pareto and benchmark wing overall characteristics is shown in Table 12 for comparison.

It can be noted from Table 12 that all the Pareto member wings out-performed the benchmark wing on L/D ratio with Pareto members 8 and 23 had weights which were lower than that of the benchmark wing when the penalty values were included in the overall wing mass. All the wings had strain values which were below the maximum allowable strain defined in Table 10.

6.2 HIGH ALTITUDE LONG ENDURANCE UAV WING DESIGN AND OPTIMISATION

In the second case, the detailed analysis and optimisation of a wing for a high altitude long endurance (HALE) UAV application is considered. This is a multi-objective problem where the goal is to maximise the lift-to-dag ratio and minimise wing weight. The operating conditions and data for the baseline wing are based on data from Northrop-Grumman [66]. The aircraft has a wingspan of approximately 17 *m*, a mean chord of approximately 2.43 *m,* a wing area of *50.19* m^2 and a plan form shape with 5.9 *deg* sweep. A breakdown of the benchmark wing can be found in Table 13.

The aircraft is assumed to operate at $M_\infty = 0.5983$ at a cruise altitude of 15840 *m*. It is also assumed that this UAV uses a single aerofoil, the LRN1015 aerofoil, as the only wing section aerofoil along the wing span.

6.2.1 DESIGN VARIABLES AND CONSTRAINTS.

Similar to the previous MALE UAV test case, the wing geometry is represented by three aerofoil sections consisting of sixteen variables each, eight variables that describe the wing planform and eleven variables to describe the internal structure. Once again, 6 free control points on the mean line and 10 free control points on the thickness distribution are used to describe each of the aerofoil sections.

The benchmark and upper and lower bounds for the design variables are indicated in Table 14.

Description	**Benchmark**	**Lower Bound**	**Upper Bound**
Angle of Attack, *(deg)*	5.75	0	6
Root Chord, *(m)*	2.432	2.375	2.5
Wing Inboard Leading Edge Sweep, *(rad)*	0.102972	0	0.1745
Wing Outboard Leading Edge Sweep, *(rad)*	0.102972	0	0.1745
Chord Ratio at Break	0.78	0.55	0.9
Chord Ratio at Tip	0.287	0.2	0.8
Crank Location *(m)*	0.0881	0.06	0.6
Number of Spars	5	3	7
Number of Ribs	16	12	18
Rib Root Thickness, *(m)*	0.0015	0.001	0.005
Rib Root Thickness Taper Ratio	0.05	0.01	0.15
Spar Root Thickness, *(m)*	0.09	0.045	0.091
Spar Root Thickness Taper Ratio	0.05	0.01	0.15
Wing Skin Root Thickness, *(m)*	0.001	0.0001	0.05
Wing Skin Thickness Tip Taper Ratio	0.01	0.0095	0.15
Wing Skin Thickness Trailing Taper Ratio	0.01	0.0095	0.15
Spar Cap Root Area, *(m^2)*	0.012	0.004	0.0125
Rib Cap Root Area, *(m^2)*	0.0005	0.00045	0.002

Table 13: Structural and Planform Design Variables for the HALE UAV wing

6.2.2 DESIGN CONSTRAINTS AND PENALTIES

The constraints that must be satisfied for each candidate wing are indicated in Table 14. The rates with which wings which fail the constraints are penalised are also indicated.

Description	Values Allowed
Allowable Strain	0.00333
Allowed wing tip twist in degrees	1
Allowed wing deflection as a percentage span	20
Allowed wing moment (benchmark)	-0.3041
Minimum lift to be generated by the wing	0.89
Wing Mass per degree twist Penalty Values	0.1
Wing Mass per degree over 20 span	0.1
Wing Mass per failed panel	0.1
Additional *Cd* per over allowable	0.001
Additional *Cd* per less than the required minimum	0.005
Penalise both fitness functions above a Drag ratio of	0.1
Penalise both fitness functions above a Mass ratio of	0.1

Table 14: Constraints and Penalty Rates

All the above mentioned constraints are applied by reducing a candidate wings fitness values via a exponential penalty method as described in Section 6.1. In addition, any aerofoil generated outside the thickness bounds of 10% to 15% cases the wing to be rejected immediately before any analysis can be performed.

6.2.3 FITNESS FUNCTIONS

The two fitness functions to be optimised are defined as Eqn 14 and Eqn 15:

$$\min(f_1): \quad f_1 = \frac{1}{(L/D)} + penalty \tag{14}$$

$$\min(f_2): \quad f_2 = W + penalty \tag{15}$$

Eqn 14 relates to the aerodynamics of the candidate wing. The Evolutionary Optimiser seeks to minimise the inverse of the lift to drag ratio with the addition of penalties which are described in Section 6.1.

Eqn 15 relates to the structural aspects of the candidate wing. This function seeks to minimise the calculated mass of the wing. Penalties are also added to Eqn 15 which are described in Section 6.1

For both Eqns 14 and 15, the penalties are calculated based on the constraints specified in Section 6.2.2.

6.2.4 AERO-STRUCTURAL ANALYSIS

The aerodynamic and structural characteristics of the wing configurations are evaluated using the Aero-Structural Solver described in Section 5.1 and represented in Figure 67.

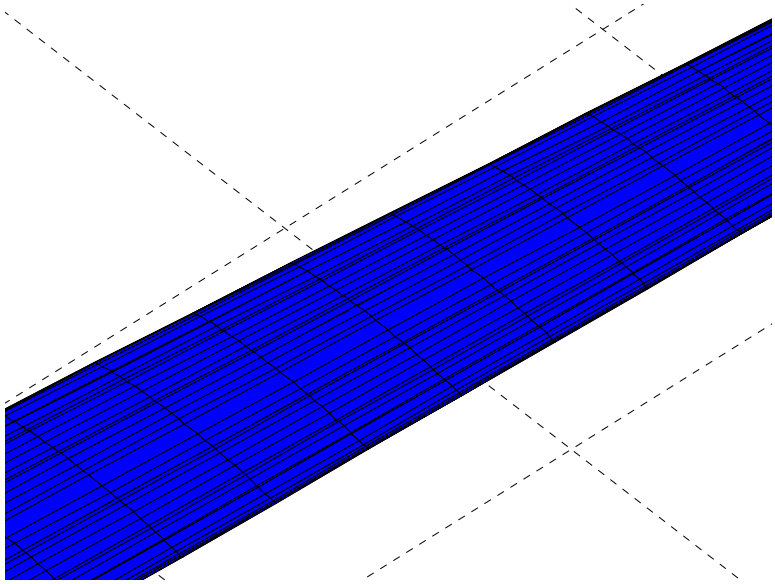


Figure 67: Aerodynamic Mesh for HALE UAV

6.2.5 IMPLEMENTATION

A single optimisation was performed for the HALE UAV wing. This optimisation used a hierarchical approach with three levels and seven populations. The top level (Level 1) used a fine mesh of between 2992 and 4624 aerodynamic panels to calculate the pressures about the candidate wings. A more coarse mesh model was used in the intermediate aerodynamic level (between 2244 and 3468 aerodynamic panels), and a coarse mesh (between 1496 and 2312 aerodynamic panels) was used in the lowest mode to explore the design search space.

Three P4, 2.4 GHz machines were used in the calculation and the population size for this problem was set to 30 in all levels.

The structural model panel density was kept as a function of the internal geometry and was not based upon the level of model complexity.

6.2.6 RESULTS

The algorithm was run for 300 function evaluations and took three days to compute on a cluster of three, 2.4 GHz machines. Figure 68 shows the Pareto front and Benchmark value for the top node of the optimisation process. Figure 70 shows the algorithm progress for objective one where each step in the figure roughly corresponds to information migration from the lower levels to an upper level. Figures 71 to 73 compare the aerofoils at root break and tip for each of the members of the Pareto front and the aerofoil for the baseline geometry. Table 16 indicates the final values of design variables and objective functions for the baseline geometry and some members of the Pareto front.

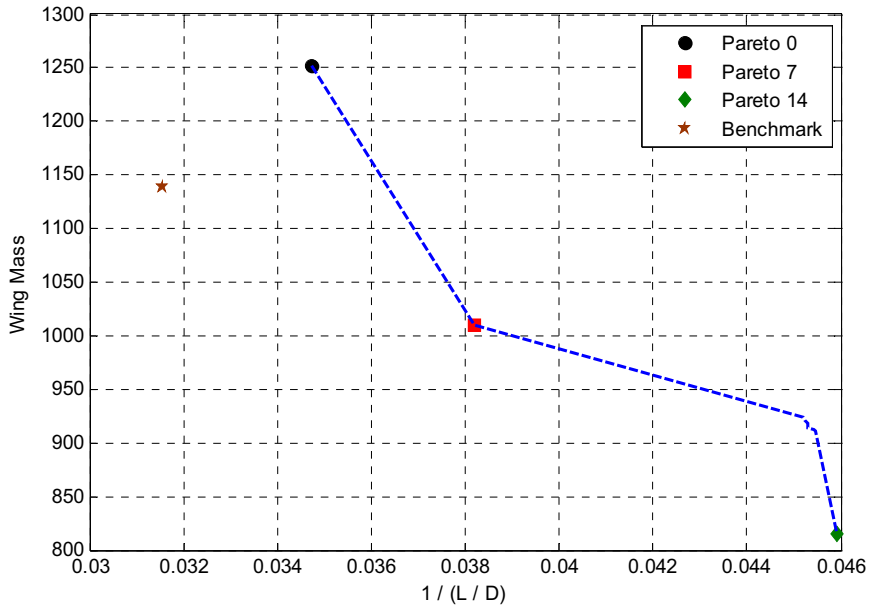


Figure 68: Comparison of Pareto front and Benchmark geometry.

On an initial investigation of Figure 68, it would seem as if the optimisation was not a success in that the evolutionary optimiser was not able to better the aerodynamic properties of the benchmark wing. The difference lies in the representation of the results as a the optimiser was unable to determine a wing which did not fail one or more of the penalty functions and hence the true lift, drag and wing mass are not represented above, but rather the fitness as seen by the optimiser.

Table 15 shows the components of each of the fitness functions for the selected Pareto members, namely members 0, 7 and 14.

Wing	CL	CD	1 / (L / D)	Wing Mass (kg)
Pareto 0	0.88344	0.02165	0.0245	963.383
Pareto 7	0.89168	0.02439	0.0274	774.284
Pareto 14	0.85968	0.02375	0.0276	525.516

Table 15: HALE Pareto Member Data without added Penalties

When the values from Table 15 are plotted in Figure 69, the results indicate that the optimiser produced Pareto members do lie to the left of the Benchmark wing and hence are have better aerodynamic and structural properties.

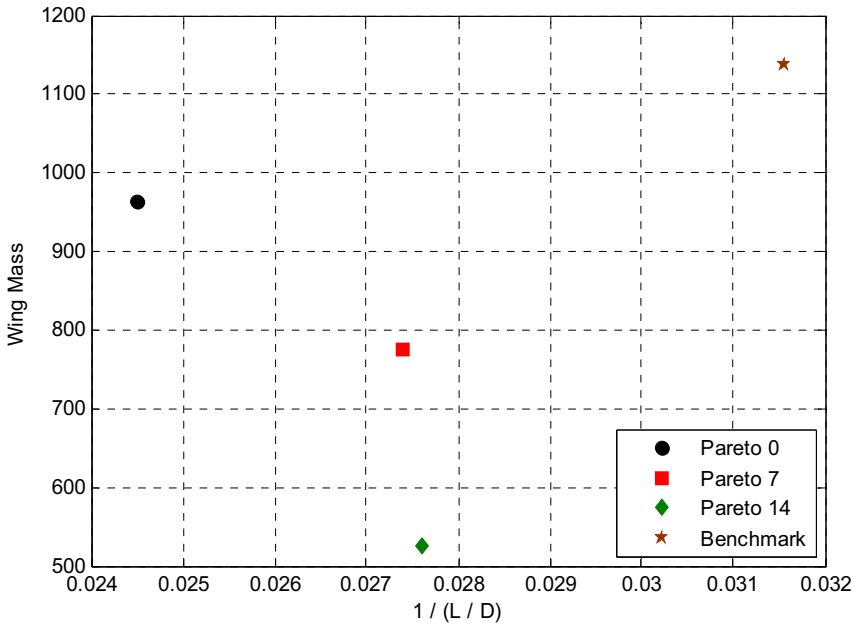


Figure 69: HALE Pareto front and Benchmark geometry with no Penalties

This difference between Figure 68 and Figure 69 is caused by a large deflection penalties added to the Pareto front.

The algorithm progress for the HALE UAV wing optimisation is shown in Figure 70. Although the optimisation was run numerous times with different migration properties, the highest level did not end up with the lowest fitness values as was the case with the MALE UAV optimisation case.

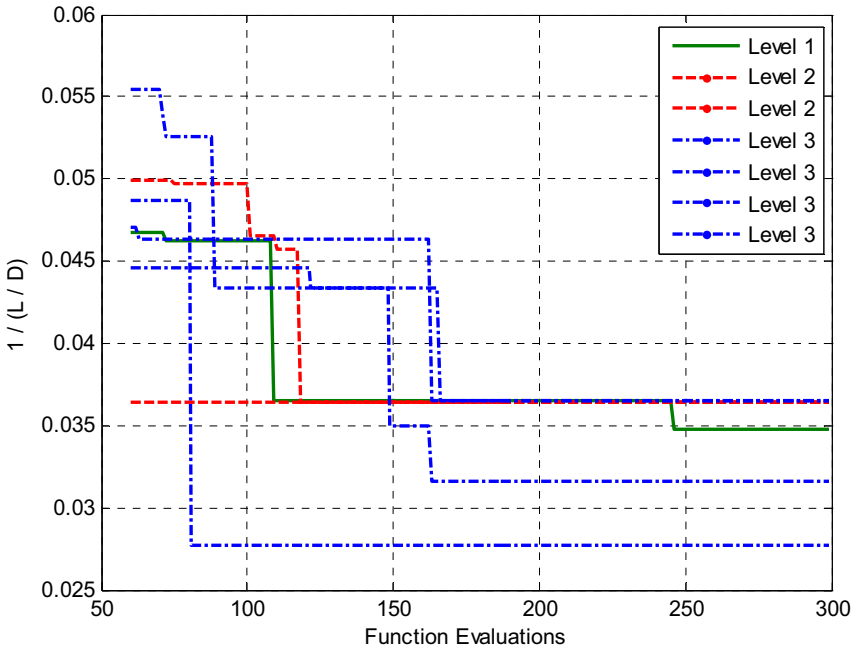


Figure 70: HALE Evolution Optimiser progress and migration steps

At program termination, the population with the lowest fitness is a level three population. A possible cause for this phenomenon is that when a candidate wing is evaluated in one of the exploratory populations, the fitness calculated is low with small penalties. As this design is passed up the optimiser tree, a case can arise where the fitness value stays almost constant while the calculated error increases. This is due to the higher tolerance on the aerodynamics calculation that, in some cases, leads to a higher pressure distribution along a wing which in turn leads to a higher deflection and constraint violations.

Description	**Benchmark**	**PM 0**	**PM 7**	**PM 14**
Angle of Attack, *(deg)*	5.75	5.427	5.716	5.428
Wing Semi Span, *(m)*	16.941	17.768	16.586	16.199
Root Chord, *(m)*	2.432	2.403	2.401	2.389
Wing Inboard Leading Edge Sweep, *(rad)*	0.103	0.111	0.0769	0.0996
Wing Outboard Leading Edge Sweep, *(rad)*	0.103	0.0141	0.0799	0.0165
Chord Ratio at Break	0.78	0.744	0.804	0.845
Chord Ratio at Tip	0.287	0.255	0.283	0.255
Crank Location *(m)*	0.0881	0.124	0.117	0.14
Number of Spars	5	4	3	3
Number of Ribs	16	14	15	15
Rib Root Thickness, *(m)*	0.0015	0.00122	0.0024	0.0013
Rib Root Thickness Taper Ratio	0.05	0.023	0.0215	0.03
Spar Root Thickness, *(m)*	0.09	0.0906	0.091	0.0904
Spar Root Thickness Taper Ratio	0.05	0.128	0.133	0.12
Wing Skin Root Thickness, *(m)*	0.001	0.000975	0.0103	0.00119
Wing Skin Thickness Tip Taper Ratio	0.01	0.0645	0.0962	0.0942
Wing Skin Thickness Trailing Taper Ratio	0.01	0.0569	0.0391	0.0501
Spar Cap Root Area, *(m^2)*	0.012	0.008	0.00655	0.00508
Rib Cap Root Area, *(m^2)*	0.0005	0.000867	0.00131	0.00102

Table 16: Summary and Comparison of HALE UAV selected Pareto members and Benchmark

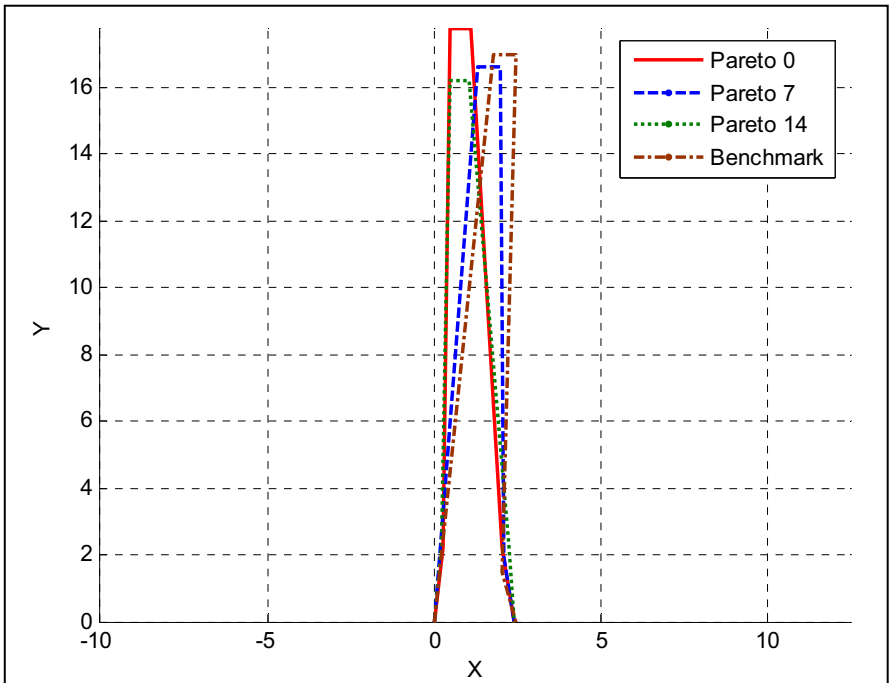


Figure 71: Selected HALE Pareto and Benchmark planform shapes

Planform Shapes

Figure 71 shows the planform shapes for the selected HALE Pareto members along with the Benchmark. From the Figure one is able to see that all the wings have a high aspect ratio and that the reference area was kept constant. The evolutionary optimiser produced wings that seemed to favour wings with a lower leading edge sweep angle. The crank location between the Pareto members and the Benchmark wing are all at a rather consistent point in the wing.

Aerofoil Sections

The root, crank and tip aerofoil sections are shown for Pareto members 0, 7 and 14 along with the Benchmark LRN 1015 aerofoil section. From Figures 72 to 74, one can see the large variation in aerofoil shapes. The variations though seem to have a common trend where the aerofoils follow an almost sinusoidal path along the lower surface. The high variation in aerofoil shapes is consistent with the variation in planform shapes shown in Figure 71, as each variation requires different aerodynamic characteristics in an attempt to meet the requirements of the penalty functions.

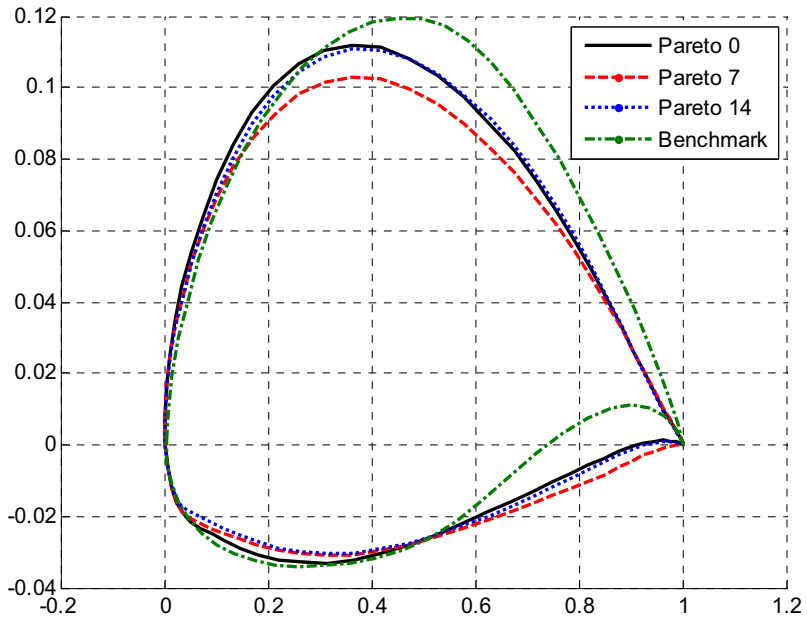


Figure 72: Selected Normalised HALE Pareto member Root Aerofoils and Benchmark

The Pareto member aerofoil sections all have a lower *t/c* ratio compared to that of the benchmark aerofoil shape. This would reduce the skin friction and form drag for each wing.

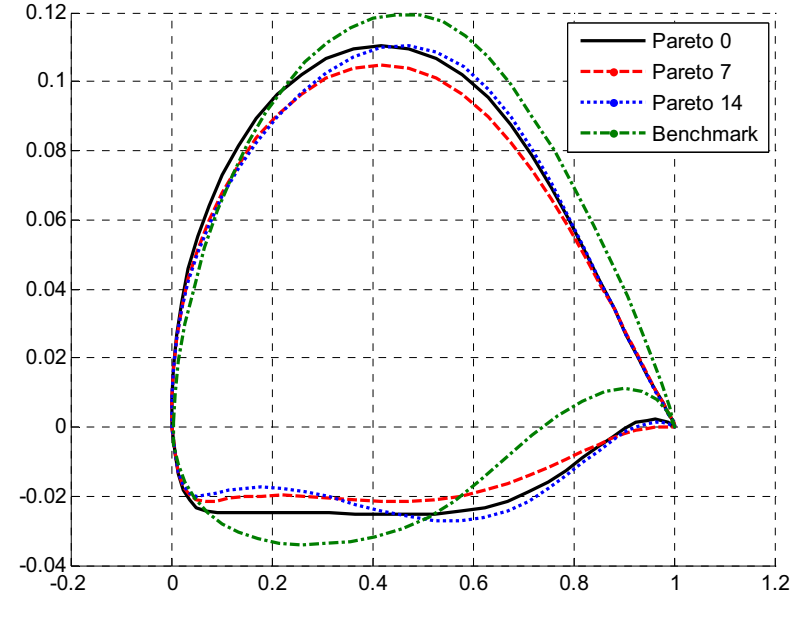


Figure 73: Selected Normalised HALE Pareto member Crank Aerofoils and Benchmark

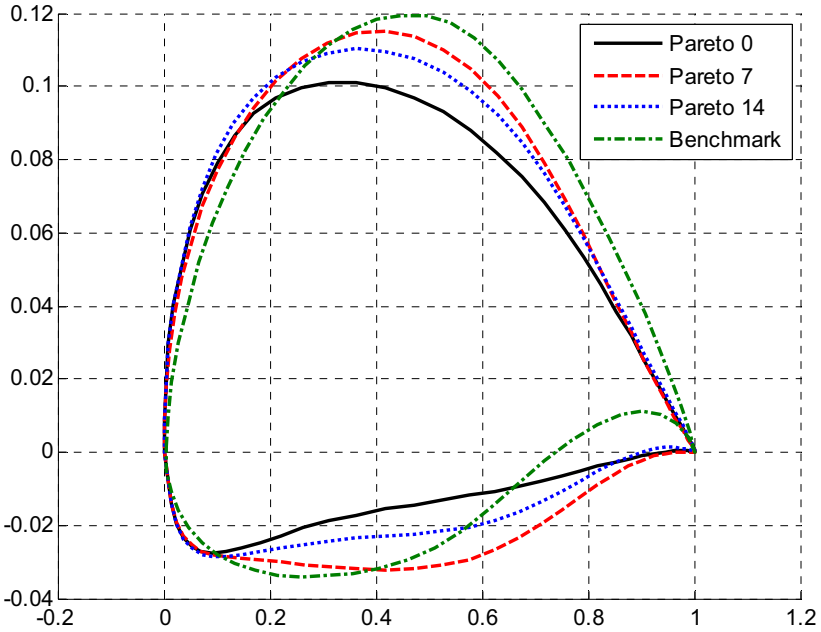


Figure 74: Selected Normalised HALE Pareto member Tip Aerofoils and Benchmark

The diverse range of aerofoil shapes further confirms that the use of a Pareto front when evaluating the HALE UAV wing problem was a success at there are large variations in geometry.

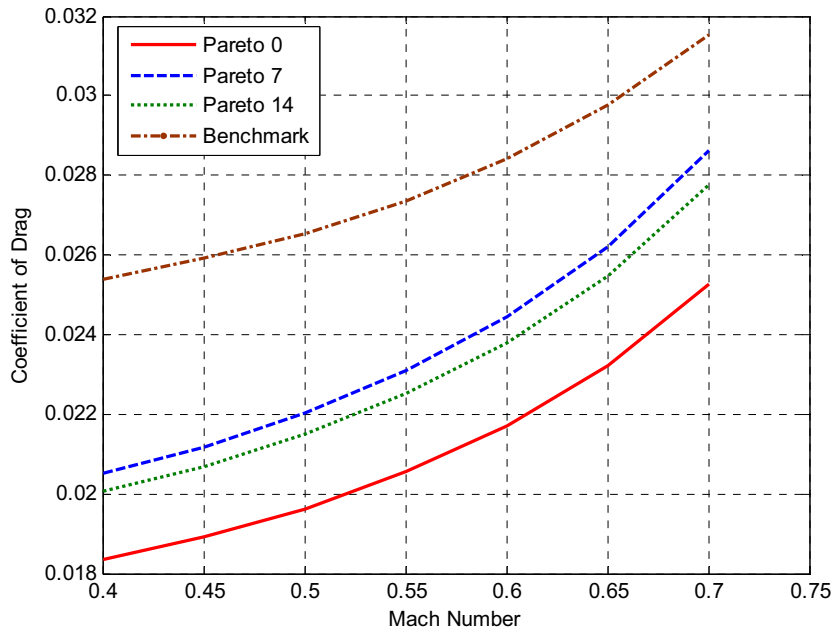


Figure 75: C_D vs. Mach number for selected HALE Pareto memebers and Benchmark

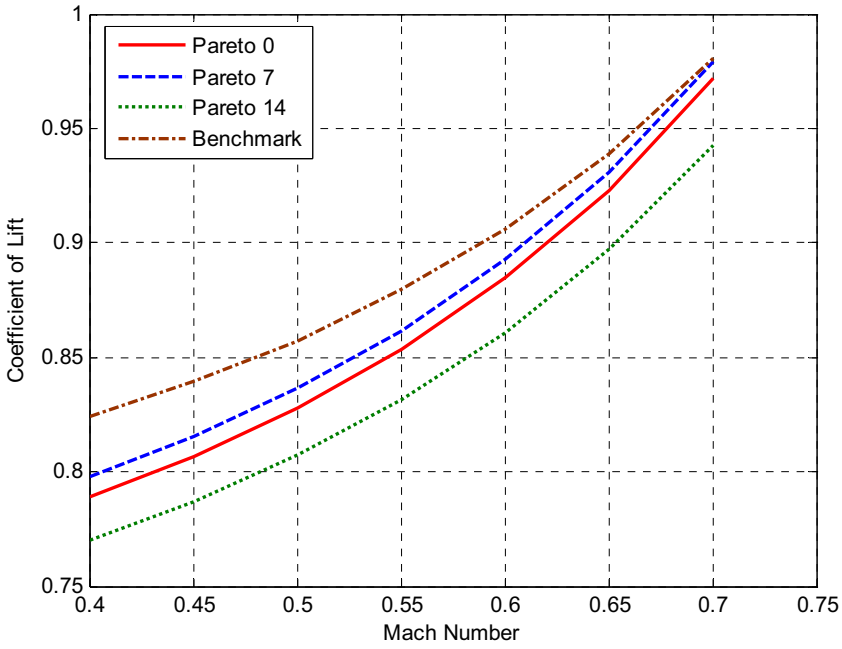


Figure 76: C_L vs. Mach number for selected HALE Pareto members and Benchmark

Mach Sweep

A Mach number sweep was performed for the selected Pareto members and the results are displayed in Figure 75 for coefficient of drag and Figure 76 for coefficient of lift.

The Pareto member wings produce less drag than the benchmark wing over the range of Mach numbers investigated. Pareto member zero produced the lowest drag of 0.02175 at the operational Mach number of 0.59.

For both the Coefficients of Lift and Drag, the trends in the Figures for the Pareto members are very similar.

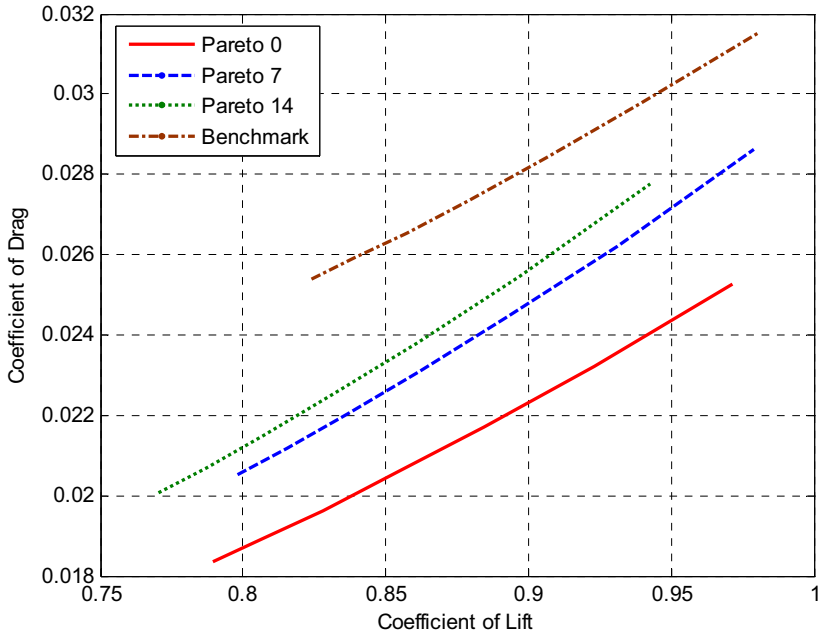


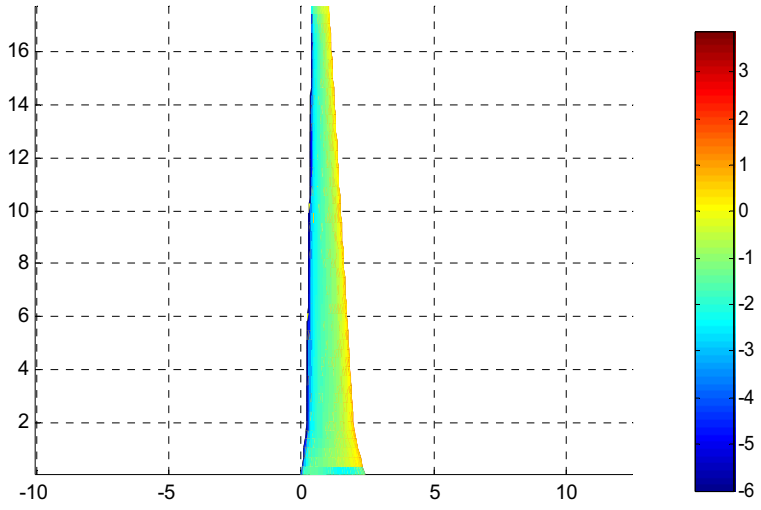
Figure 77: C_L vs. C_D for selected HALE Pareto members and Benchmark

From Figure 77, the performance of the Pareto selected wings can be compared to that of the benchmark HALE UAV wing in the coefficient of lift versus the coefficient of drag plot. As was indicated in the Pareto front shown in Figure 69, Pareto zero has the best L/D ratio and Pareto fourteen the worst. Even though Pareto fourteen has the best structural wing solution, it still has a better aerodynamic performance than the benchmark wing.

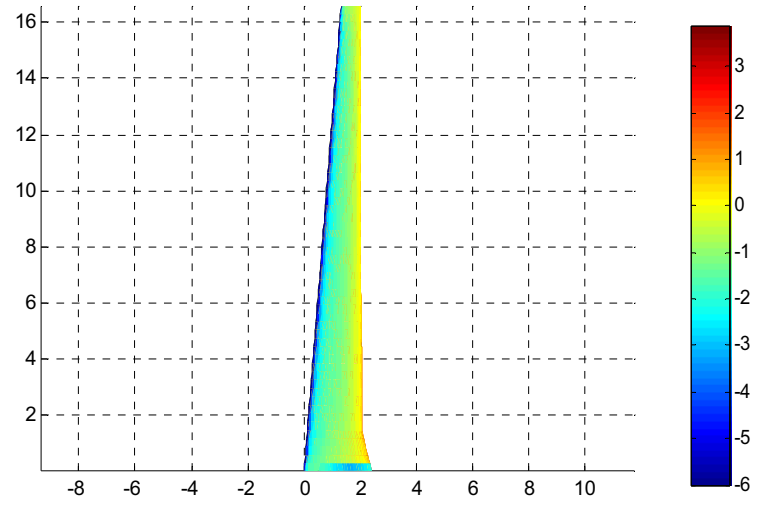
The second order external aerodynamic pressure coefficients are plotted for the upper and lower Pareto wing surfaces in Figures 78 and 79 respectively.

The pressure coefficients calculated by PANAIR about the Pareto wings are of a similar order of magnitude as those computed for the benchmark wing.

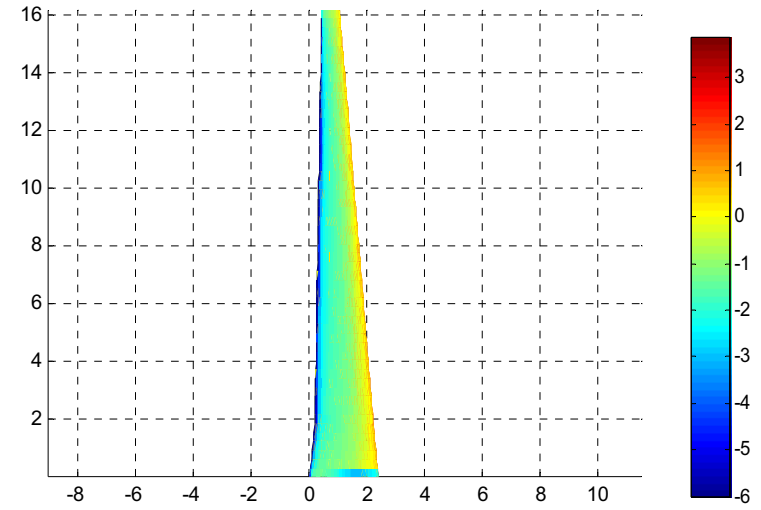
Figures 80 to 82 show the deflected Pareto wings. The original wing position and internal structure is shown as a wire frame for comparison.



Pareto 0

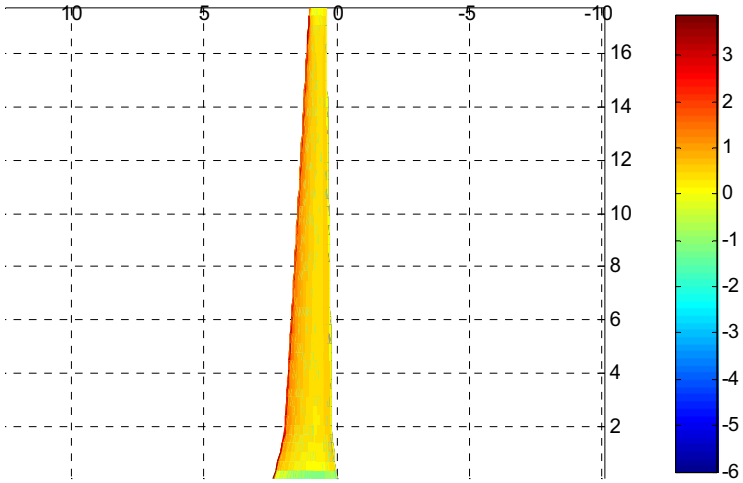


Pareto 7

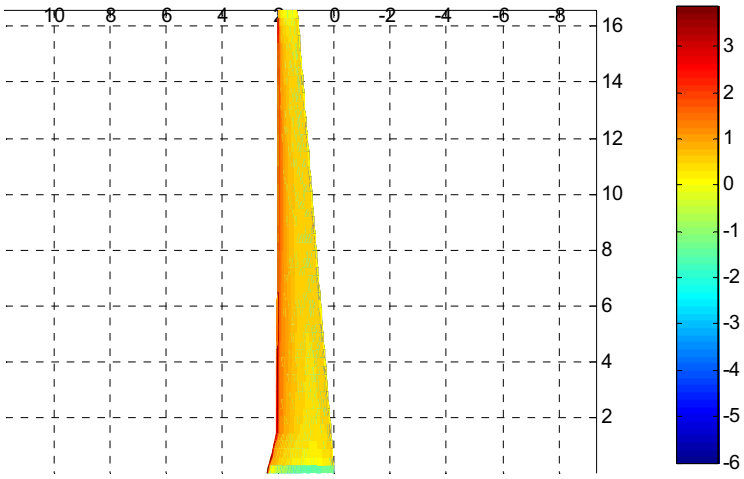


Pareto 14

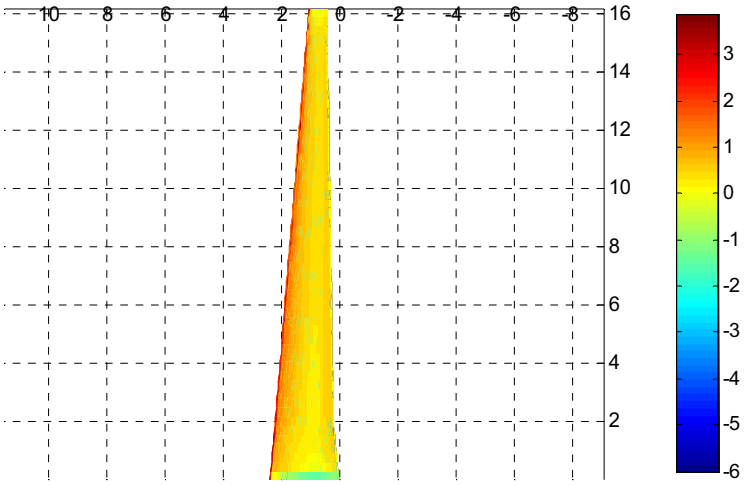
Figure 78: C_P Distribution for selected HALE Pareto Members – Upper Surface



Pareto 0



Pareto 7



Pareto 14

Figure 79: C_P Distribution for selected HALE Pareto Members – Lower Surface

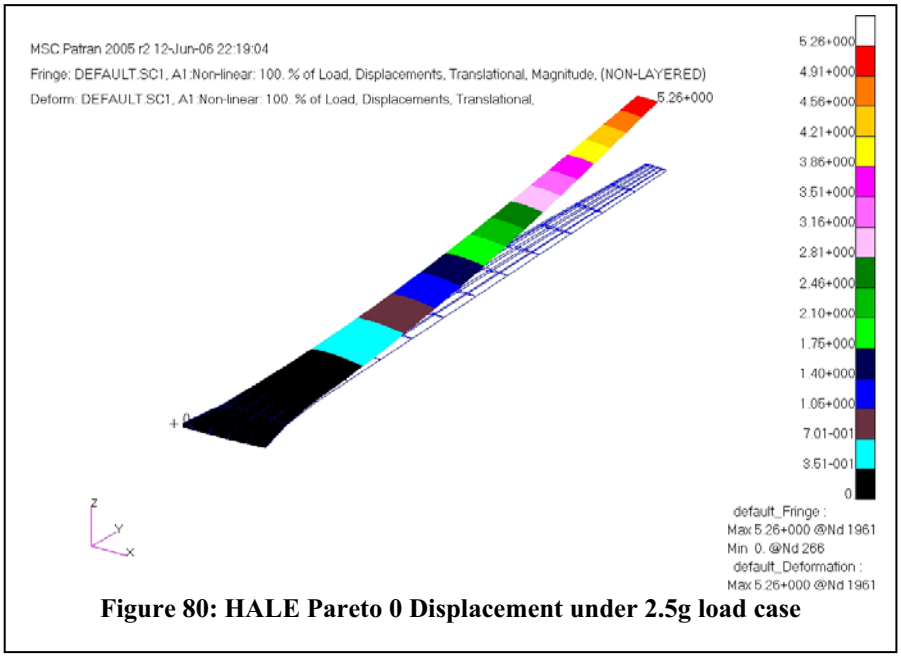


Figure 80: HALE Pareto 0 Displacement under 2.5g load case

From Figures 80 to 82 showing the displacement of the HALE UAV wings, it is evident that the wings deflected more than the allowable twenty percent span value indicated in Table 15. Further testing is needed to analyse the effects of constraining the displacement to within the bounds without over influencing the optimisation process.

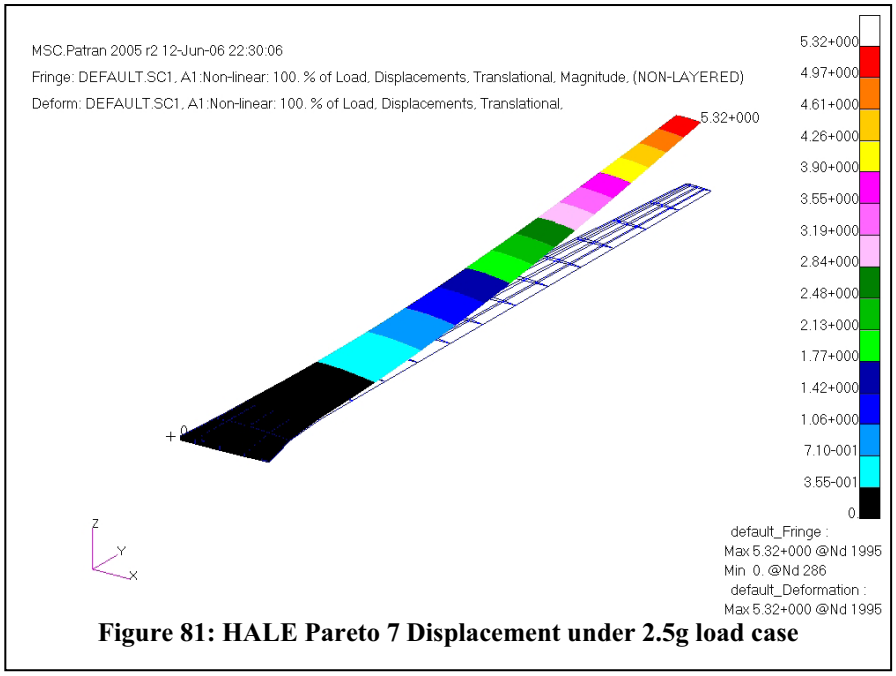


Figure 81: HALE Pareto 7 Displacement under 2.5g load case

An analysis of the strains due to deflections are shown in Figures 83 to 85.

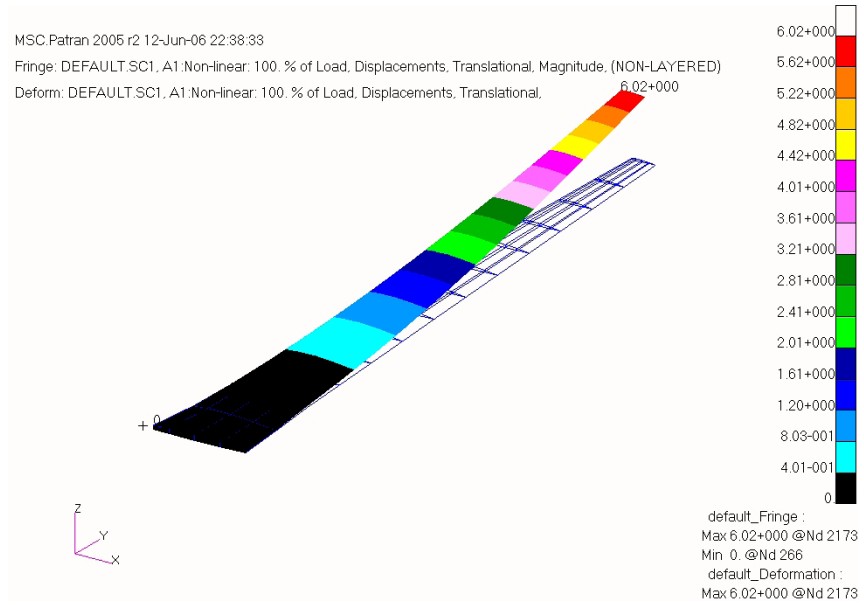


Figure 82: HALE Pareto 23 Displacement under 2.5g load case

Due to the large displacements shown in Figures 83 to 85, the internal strains in the HALE UAV wings are very large and are almost within an order of magnitude of the maximum allowable strain. This is a concern as the wing is an all composite design, these large strain values are undesirable as there could be defections in the manufacture process which would increase the localised strain about the defect. This in turn could cause the wing to fail under load.

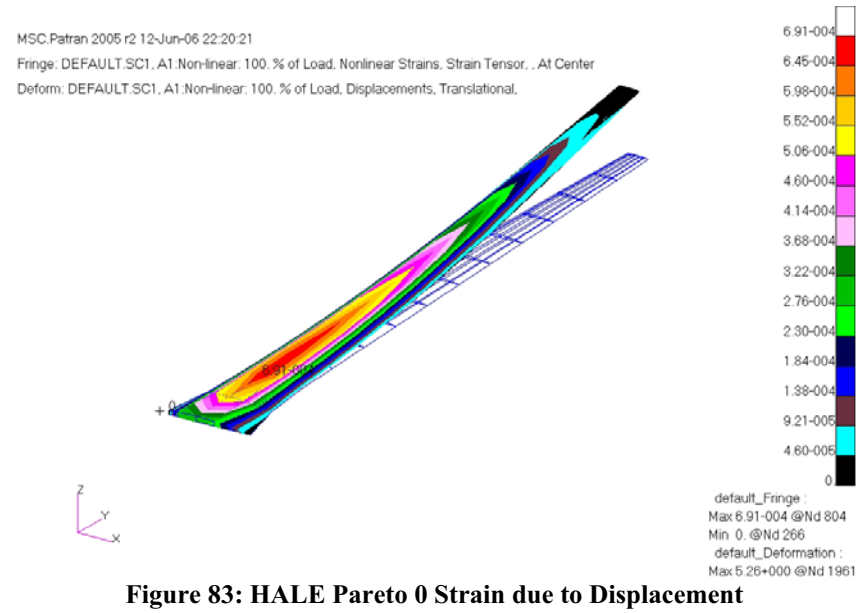


Figure 83: HALE Pareto 0 Strain due to Displacement

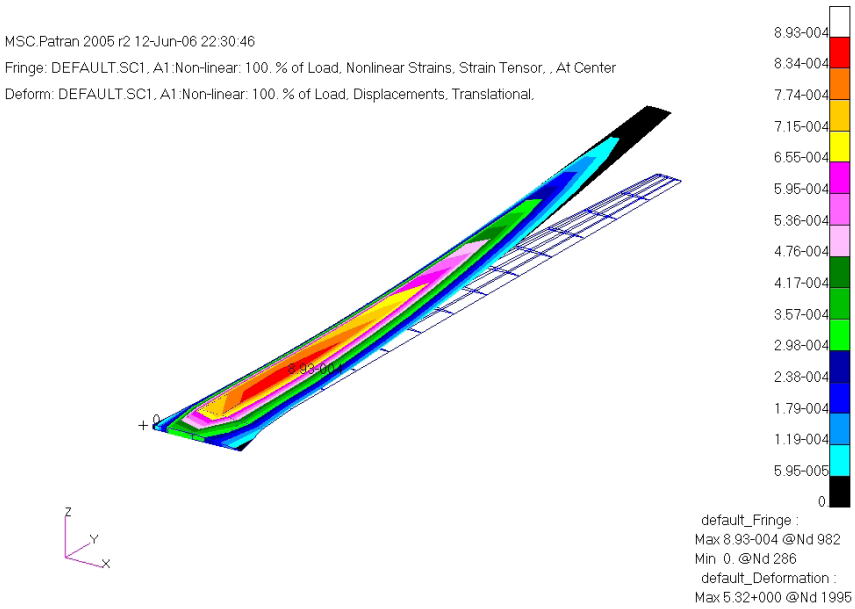


Figure 84: HALE Pareto 7 Strain due to Displacement

All the Pareto members selected from the Pareto front shown in Figure 69 had better lift to drag ratios than the Benchmark wing.

The mass of the simulated Pareto members was lower than that of the benchmark, but the deflection of the wings was larger than the allowable twenty percent. The maximum strain values calculated by MSC.Nastran® were higher than those calculated for the MALE UAV Pareto wings. These large strain values are caused by the large displacements of the wings due to a less rigid wing internal structure.

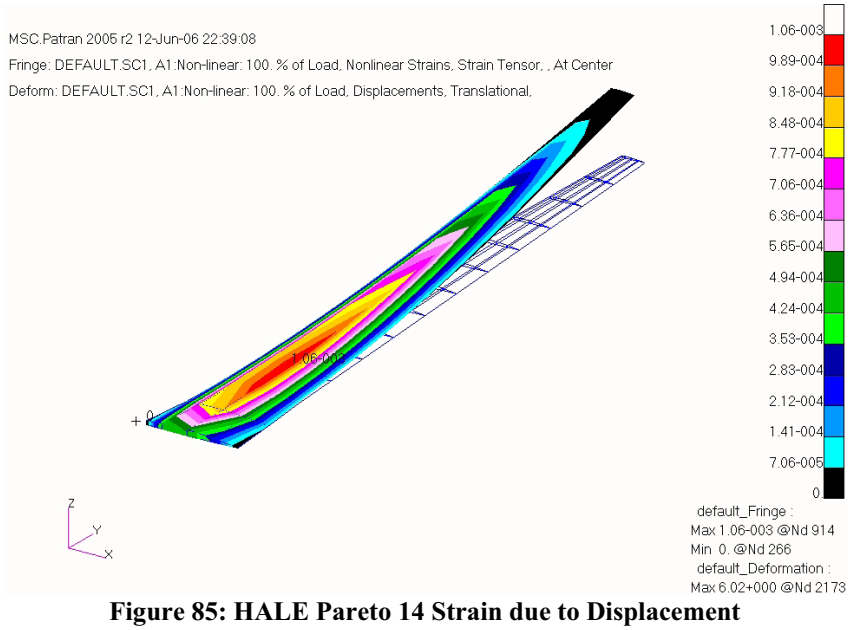


Figure 85: HALE Pareto 14 Strain due to Displacement

	Pareto 0	Pareto 7	Pareto 14	Benchmark
Deflection (%)	29.65	31.08	37.16	16.7
Maximum Strain	0.000691	0.000893	0.00106	0.000487
Mass (kg)	963.38	774.28	525.51	1138.15
L/D	40.805	36.559	36.197	31.729
Mass Penalty (kg)	286.55	235.65	167.69	0
Drag Penalty	0.0051	0.0074	0.0076	0

Table 17: Summary of Displacement, Strain and Mass for Selected HALE Pareto members and Benchmark

From Table 17 it can be seen that all the HALE Pareto wings had penalties. This would have influenced the HAPMOEA code as it calculated new generations of candidate solutions for evaluation.

A method to reduce the penalties and keep the penalty rates and fitness functions constant would be to increase the number of dimensions the optimisation was performed over. The HALE UAV was evaluated as a two dimensional problem (Inverse of Lift to Drag ratio vs. calculated wing mass) but could have been evaluated as a four dimensional problem. The two additional dimensions would be the penalties added to each fitness function. These two additional penalties would be formulated such that the optimiser has the ability to drive them to zero, allowing the first two initial fitness functions describing the inverse of the Lift to Drag ratio vs. the calculated wing mass to dominate the solution.

7 CONCLUSIONS

This work has described the characteristics and implementation details of a framework in which different aeronautical problems can be analysed. The framework is based on a Multi-objective Evolutionary Algorithm. This thesis indicates and addressed some of the limitations of current optimisation techniques and the need for alternative approaches. The thesis provides a description of the main components of the framework, i.e. an Aero-structural solver, a wing mesh generation capability, a robust MOEA optimiser and capabilities for parallel computing and post processing.

The methodology is illustrated on its application to the design optimisation of UAV wings as a multi-objective and MDO problem showing the benefits of the method. The method is found to be capable of identifying the trade-off between the multi-physics involved and provides classical aerodynamic shapes as well as alternative configurations from which the designer can choose. It is observed that there was a computational gain on using asynchronous evaluation and a hierarchical topology of fidelity models as compared to a single model during the optimisation.

Large gains in aerodynamics can be made for both the MALE and HALE UAV test cases through the use of the data calculated by the HAPMOEA code from the assumed benchmark wings. The lift to drag ratio can be extended to regions only seen previously within the flight envelopes of high performance gliders.

Structural gains can also be made, though as the true benchmark structures are more complex and probably include hard points within the wings and fuel tanks. The results obtained through the HAPMOEA code are only indicative of future possibilities given more accurate data.

A more detailed analysis needs to be performed on algorithms to better describe the internal structure of high aspect wings. This analysis could aid the optimisation of the HALE UAV test case and limit the penalties due to deflection and strain. Through the selection of different penalty rates, HALE wings which better fit the design constraints could be found.

Given more accurate benchmark information, a more complex and detailed investigation and optimisation could be performed.

8 FUTURE WORK

A limitation to this study is that the solution found is not a full aero-elastic solution but rather a one iteration coupling of the aerodynamics of the candidate wing with the structural model. Furthermore, a rather low fidelity model is used to calculate the form and skin friction components of the drag produced by the candidate wing.

Further analysis methods and techniques could be incorporated into the Aero-Structural Solver framework, or modifications to the code could be made to perform calculations which would assist in designing more optimal wings based upon other constraints. The work could be divided into two sections namely the aerodynamics and aerodynamic package, and the structural model and structural simulation method.

8.1 AERODYNAMICS

8.1.1 SOLVER

The PANAIR aerodynamic solver could be upgraded to make use of a Navier-Stokes solver. This would increase the computational cost for each candidate wing simulation, but would also increase the accuracy of the solution attained if complex flow geometries were being analysed such as slotted flaps.

If the solver were upgraded, special consideration would have to be given when designing the program to automatically generate the domain meshes about the planned wings. If little consideration was given in defining mesh densities about areas of high macroscopic gradients, loss of accuracy could be significant.

8.1.2 DRAG CALCULATIONS

Currently the Aero-Structural Solver makes use of PANAIR and FRICTION to calculate the aerodynamic loads and aerodynamic fitness values for each candidate wing. As PANAIR is only applicable to linear potential flows, it would be beneficial to upgrade to a nonlinear flow solver such as TRANAIR®. Such an upgrade would allow the user or optimiser the ability to model complex flows which included shock waves at or near transonic speeds. This would also minimise any solution errors due to large Mach angles.

FRICTION is a low fidelity program which was passed a constant value for the percentage chord location where the transition from a laminar to turbulent boundary layer took place. The assumption was a crude estimate made by the author and would not hold true for all the candidate wings analysed. For a more accurate skin friction drag coefficient calculation, a higher fidelity model could be incorporated into the solution sequence. Another simple addition which could be made to FRICTION could be a simple iterative method which could determine the Reynolds number at which the flow would make the transition from laminar to turbulent, and the corresponding chord ratio passed to FRICTION when calculating the required coefficients of drag.

8.2 STRUCTURES

8.2.1 MATERIALS

MSC.Nastran® has the ability to define materials that are made up of a number of plies. Hence graphite epoxy based materials could be correctly defined through the build up and orientation of the plies within the material. This could lead to a more detailed optimization where the material thickness could not only be a variable, but the make up of the material, fibre thickness, orientation and laminate layers, could also be added as design variables in the optimization process.

8.2.2 ELEMENTS

Most of the element input decks used when defining the elements in the structural model had the capacity to make use of material orientation within the element. Future work could entail the use of fibre orientation material properties such as those mentioned above. This would lead to a better understanding and optimisation as the element's true behaviour under loading could be studied.

If the position of the spars and ribs as functions of chord and span respectively were included as design variables within the simulation framework, the optimality of the spars could be increased as spars could be placed such that the bending moments generated by the aerodynamic forces could be minimised, and the variable rib placement could mean that better rib distributions could be achieved where sections of high strain which occur at crank locations could be minimised through the use of two closely spaced ribs.

More complex thickness functions could be devised, such as third or fourth order polynomials, to describe the way in which Spar and Rib thicknesses vary according to span. This could assist in the reduction of high strain concentrations which would occur where the wing geometry changes greatly.

CBAR or CBEAM elements could be used to describe the Rib and Spar Caps. The height and width of the sections could then be added to the design variables optimised by the Evolutionary Optimiser code. This would allow for more accurate deflection calculations and assist in the minimisation of the candidate wing structure.

8.2.3 SOLUTION METHOD

Further analysis could be conducted to incorporate buckling analysis to investigate the natural frequencies of the candidate wings. MSC.Nastran® can incorporate the MSC.Flightloads® package and undertake full aero-elastic analysis of candidate wing geometries. This would greatly increase the solution resolution of the candidate wings as solutions could be penalised against a final wing deflection value instead of the first solution iteration. This is beneficial as many wing solutions would solve to a position which would lie in the predefined constraints, but as the initial iteration indicates that the deflection is greater than this predefined value, the wing is discarded prematurely.

REFERENCES

1. NASA. *Airborne Science Program - Altair Specifications*. 2004 [cited 2005 August]; Available from: http://www.nasa.gov/centers/dryden/research/AirSci/UAV_Specs/altair_specs.html http://www.nasa.gov/centers/dryden/news/FactSheets/FS-073-DFRC.html.
2. Goraj, Z., et al., *High altitude long endurance unmanned aerial vehicle of a new generation – a design challenge for a low cost, reliable and high performance aircraft.* Bulletin of the Polish Academy of Sciences, 2004. **52**(3).
3. Mason, W.H., Knill, D.L., Giunta, A. A., Grossman, B. , Haftka, R. T. and Watson, L. T. . *Getting the Full Benefits of CFD in Conceptual Design*. in *AIAA 16th Applied Aerodynamics Conference,*. 1998. Albuquerque, New Mexico: AIAA
4. Agarwal, R.K., *Computational Fluid Dynamics for Whole Body Aircraft.* Annual Review of Fluid Mechanics, 1999. **31**: p. 125–169.
5. Thomas, Z.a.G., A. . *AIAA Paper 1999-3798-Multidisciplinary Design Optimization Techniques: Implications and Opportunities for Fluid Dynamics Research*. in *30th AIAA Fluid Dynamics Conference* 1999. Norfolk, VA, .
6. Giunta, A.A., Balabanov, V., Haim, D., Grossman, B., Mason, W. H. , Haftka, R. T. and Watson, L. T. , *Multidisciplinary Optimization of a Supersonic Transport Using Design of Experiments Theory and Response Surface Modeling.* The Aeronautical Journal, 1997(October 1997): p. 347–356.
7. Coello, A.C., D.A.V. Veldhuizen, and C.B. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*. 2002, New York: Kluwer Academic Publishers.
8. Deb, K., *Multi-Objective Optimization Using Evolutionary Algorithms*. 2003: Wiley.
9. Kim, H.J.a.R., O. H., *Dual-Point Design of Transonic Airfoils Using the Hybrid Inverse Optimization Method.* . Journal of Aircraft, 1997. **34**(5): p. 612–618.
10. Alexandrov, N.M. and R.M. Lewis, *Analytical and Computational Properties of Distributed Approaches to MDO.* 2000.
11. Sobieszczanski-Sobieski, J. and R.T. Haftka, *Multidisciplinary Aerodpace Design Optimisation: Survey of Recent Developments.* Structural Optimisation, 1997. **14**: p. 23.
12. Bartholomew, P., *The Role of MDO within Aerospace Design and Progress towards an MDO Capability*, in *7th AIAA/USAF/NASA/ISSMO Symposium on Mulidisciplinary Analysis and Optimisation*. 1998, AIAA: St. Louis, MO.
13. Obayashi, S. *Multidisciplinary Design Optimization of Aircraft Wing Planform Based on Evolutionary Algorithms*. in *IEEE International Conference on Systems, Man, and Cybernetics*. 1998. La Jolla, California, IEEE, October . IEEE.
14. Oyama, A., M. S. Liou, and S. Obayashi. *Transonic Axial-Flow Blade Shape Optimization Using Evolutionary Algorithm and Three-DimensionalNavier-Stokes Solver*. in *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*. 2002. Atlanta, Georgia.

15. Parmee, I. and A.H. Watson. *Preliminary Airframe Design Using Co-Evolutionary Multiobjective Genetic Algorithms*. in *The Genetic and Evolutionary Computation Conference*. 1999. Orlando, Florida, USA: Morgan Kaufmann.
16. Raymer, D. *Use of Genetic and Evolutionary Algorithms for Aircraft Conceptual Design Optimization.* . in *41th AIAA Aerospace Sciences Meeting*. 2003. Reno, Nevada.
17. Gonzalez, L.F., *Robust Evolutionary Methods for Multi-objective and Multidisciplinary Design in Aeronautics, Thesis, School of Aerospace, Mechanical and Mechatronic Engineering, J07 , NSW, 2006 Australia,* . in *School of Aerospace, Mechanical and Mechatronic Engineering*. 2005, University of Sydney: Sydney. p. 222.
18. Sobieski, J. and R. Haftka, *Multidisciplinary Aerospace Design Optimization: Survey of Recent Developments.* 1996.
19. Giesing, J.P.a.B., J. F. . *A Summary of Industry MDO Applications and Needs*. in *Seventh AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimisation*. 1998. St. Louis, Missouri: AIAA.
20. Sacks, J., et al., *Design and analysis of computer experiments.* Statistical Science, 1989. **4**: p. 409-435.
21. Goldberg, D., *Genetic Algorithms in Search, Optimization and Machine Learning*. 1989: Addison-Wesley.
22. Holland, J.H., *Adaption in Natural and Artificial Systems*. 1975: University of Michigan Press.
23. Bäck, T., G. Rudolph, and H.P. Schwefel. *Evolutionary programming and evolution strategies: Similarities and differences*. in *Second Annual Conference on Evolutionary Programming*. 1993. San Diego, CA.
24. Bäck, T. and H.P. Schwefel. *Evolution Strategies I: Variants and their computational implementation*. in *Genetic Algorithms in Engineering and Computer Science* 1995: Chichester: Wiley.
25. Collard, P. and C. Escazut. *Genetic Operators in a Dual Genetic Algorithm*. in *Seventh IEEE International Conference on Tools with Artificial Intelligence*. 1995. Virginia, USA: IEE
26. Sefioui, M., J. Periaux, and J.G. Ganascia. *Fast convergence thanks to diversity. Evolutionary Programming V*. in *5th Annual Conference on Evolutionary Programming*. 1996: MIT Press.
27. Thierens. *Adaptive mutation rate control schemes in genetic algorithms*. in *IEEE World Congress on Computational Intelligence: Congress on Evolutionary Computation* 2002: IEEE Press.
28. Mäkinen, R., Neittaanmäki, P., Périaux, P. and Toivanen, J. . *A Genetic Algorithm for Multiobjective Design Optimization in Aerodynamics and Electromagnetics.* . in *ECCOMAS 98 Conference*. 1998. Athens, Greece: Wiley.
29. Sasaki, D., Obayashi, S., , *Adaptive range multi objective genetic algorithms and self –organizing map for multi objective optimisation problem.* in VKI Lecture Series 2004-07 , Optimization methods & tools for multicriteria/multidisciplinary design , Applications to Aeronautics and Turbomachinery , Eds, H. Deconinck, J. Periaux and K. Giannakoglou, 2004, 2004.

30. Koza, J.R., *Genetic programming : on the programming of computers by means of natural selection*. Complex adaptive systems. Vol. xiv. 1992, Cambridge, Mass.: MIT Press. 819.
31. Oyama, A., Obayashi, S. and Nakamura, T. , *Real-Coded Adaptive Range Genetic Algorithm Applied to Transonic Wing Optimization*, in *Lecture notes in Computer Science*, Springer-Verlag, Editor. 2000: Berlin Heidelberg New York. p. 712-721.
32. Obayashi, S., *Multidisciplinary Design Optimisation of Aircraft Wing Planform Based on Evolutionary Algorithms*, in *IEEE International Conference on Systems, Mand and Cybernetics*. 1998, IEEE: La Jolla, California.
33. Anderson, M.B.a.G., G. A. , *Using Pareto Genetic Algorithms for Preliminary Subsonic Wing Design*, in *AIAA Paper 96-4023*, AIAA, Editor. 1996: Washington, D. C.
34. Takahashi, S., Obayashi, S. and Nakahashi. T. . *Transonic Shock-free Wing Design with Multiobjective Genetic Algorithms*. in *International Conference on Fluid Engineering*. 1997. Tokyo, Japan.
35. Gonzalez, L., Whitney, E. W., Srinivas, K., Armfield, S. and Periaux, J., *A Robust evolutionary technique for coupled and multidisciplinary design optimisation problems in aeronautics.* Computational Fluid Dynamics Journal,, 2005: p. 142 – 153.
36. Crispin, Y., *Aircraft Conceptual Optimization Using Simulated Evolution,*, AIAA and P. 94-0092., Editors. 1994
37. Crossley, A.W.a.L., H, *Design of Helicopters via Genetic Algorithm.* Journal of Aircraft, 1996. **3**(6).
38. Ruben, P., Chung, J., and Behdinan, K. and . *Aircraft Conceptual Design Using Genetic Algorithms.* in *8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*. 2000. Long Beach, California.
39. Ali, N.a.B., K. *Conceptual Aircraft Design-A Genetic Search and Optimisation Approach*. in *23rd International Congress of Aeronautical Sciences (ICAS)* 2002. Toronto, Canada.
40. Gonzalez, L.F., et al. *A Robust Evolutionary Technique for Inverse Aerodynamic Design*. in *Design and Control of Aerospace Systems Using Tools from Nature*. 2004 Jyvaskyla, Finland: Computational Methods in Applied Sciences and Engineering.
41. Ng, T.T.H.a.L., G. S. B. , *Application of Genetic Algorithms to Conceptual Design of a Micro-air Vehicle.* . Engineering Applications of Artificial Intelligence, , 2002. **15**(5): p. 439-445, .
42. Cvetkovic, D.a.P., I.C. . *Designer's Preferences and Multi-objective Preliminary Design Processes*. in *Fourth International Conference on Adaptive Computing in Design and Manufacture (ACDM'2000),*. 2000. University of Plymouth, UK: Springer.
43. Raymer, D.a.C., W. . *A Comparative Study of Genetic Algorithm and Orthogonal Steepest Descent for Aircraft Multidisciplinary Optimization-Paper 2002-0514*. in *AIAA Aerospace Sciences Meeting. AIAA*. 2002. Reno, Nevada.
44. Whitney, E., et al., *Advances in Hierarchical, Parallel Evolutionary Algorithms for Aerodynamic Shape Optimisation.* JSME (Japan Society of Mechanical Engineers) International Journal, 2002. **45**(1).

45. Hansen, N. and A. Ostermeier, *Completely Derandomized Self-Adaptation in Evolution Strategies.* Evolutionary Computation, 2001. **9**(2): p. 159 - 195.
46. Sefrioui, M. and J. Périaux. *A Hierarchical Genetic Algorithm Using Multiple Models for Optimization*. in *Problem Solving from Nature*. 2000.
47. Cantu-Paz, E., *Efficient and Accurate Parallel Genetic Algorithms*. 2000: Kluwer Academic Publishers, New York.
48. Veldhuizen, D.A.V., J.B. Zydallis, and G.B. Lamont, *Considerations in Engineering Parallel Multiobjective Evolutionary Algorithms.* IEEE Transactions on Evolutionary Computation, 2003. **7**(2): p. 144--173.
49. Geist, A., et al., *PVM: Parallel Virtual Machine. A User's Guide and Tutorial for Networked Parallel Computing*. 1994: Massachusetts Institute of Technology.
50. Wakunda, J. and J.A. Zell. *Median-selection for parallel steady-state evolution strategies*. in *Parallel Problem Solving from Nature* 2002. Berlin: Springer.
51. Michalewicz, Z., *Genetic algorithms + data structures = evolution programs*. 1992, Berlin ; New York: Springer-Verlag. xiv, 250 p.
52. Mason. *FRICTION*. 2001 [cited 2005 September]; Available from: http://www.aoe.vt.edu/~mason/Mason_f/MRsoft.html.
53. Saaris, G.R., *A502I User's Manual - PAN AIR Technology Program for Solving Problems of Potential Flow about Arbitrary Configurations*. 1992: Boeing.
54. Craidon, C.A., *A Description of the Langley Wireframe Geometry Standard (LaWGS) Format*. 1985, NASA.
55. Bertin, J.J., *Aerodynamics for Engineers*. 4th ed. 2002: Prentice Hall.
56. Margason, R.J., et al. *Subsonic Panel Methods - A Comparison of Several Production Codes*.
57. Kolbe, C.D. and R.W. Boltz, *The Forces and Pressure Distribution at Subsonic Speeds on a Plane Wing Having 45 of Sweepback, an Aspect Ratio of 3, and a Taper Ratio of 0.5*. 1951, Washington: NACA.
58. Marisarla, S., *Structural Analysis of an Equivalent Box-Wing Representation of Sensorcraft Joined-Wing Confuration for High-Altitude, Long-Endurance (HALE) Aircraft*, in *Mechanical Engineering*. 2005, University of Cincinnati: Cincinnati.
59. MSC.Software, *Quick Reference Guide*. 2005.
60. Venter, G. and J. Sobieszczanski-Sobieski, *Multidisciplinary optimisation of a transport aircraft wing using particle swarm optimisaiton.* Structural Mutidisciplinary Optimization, 2004. **26**: p. 121-131.
61. Komzsik, L., *MSC.Nastran 2001 Numerical Methods User's Guide*. 2001.
62. Raymer, D.P., *Aircraft design : a conceptual approach*. 3rd ed. AIAA education series. 1999, Reston, VA.: American Institute of Aeronautics and Astronautics. xviii, 923 p.
63. www.airforce-technology.com. *Specifications - Global Hawk High Altitude, Long Endurance Unmanned Reconnaissance Aircraft, USA*. 2004 [cited 2005 August]; Available from: http://www.airforce-technology.com/projects/global/specs.html.
64. Selig, M. *UIUC Airfoil Coordinates Database*. 2002 [cited 2005 August]; Available from: http://www.ae.uiuc.edu/m-selig/ads/coord_database.html.
65. Abott, I.H. and A.E.V. Doenhoff, *Theory of Wing Sections*. 1980: Dover.

66. NORTHROP-GRUMMAN. *UNMANNED SYSTEMS*. 2006 [cited 2006; Available from: http://www.northropgrumman.com/unmanned/index.html.

APPENDIX

A: AERO-STRUCTURAL PROGRAM EXECUTION

The Aero-Structural program is executed through the *AeroStructural.p* file. This script file loads all the user defined geometry and flight conditions, interprets the data and calls both Panair and MSC.Nastran® as external programs to calculate the aerodynamics and structural response respectively.

Therefore, copies of the following executables must reside in the working directory:

- Panin.exe
- Panair.exe
- Makewgs.exe
- Friction.exe

Along with the above mentioned files, a special file needs to reside in the working directory which is a link to the MSC.Nastran® executable.

The user sets the wing geometry in the *UserWing.m* file along with the aerofoil sections used at the crank locations. The material properties defining the different metals/composites are defined in the *MaterialProperties.m* file. The flight conditions used in the Panair calculations are specified in the *UserAuxilliaryData.m* file.

The rates at which the different penalties are calculated are set in the *penaltyallowrates.m* file. This file also contains the penalty cut off values. In other words, if a calculated penalty fraction is calculated as larger than the user specified value, the evaluation of the wing is ended and the simulation treated as failed.

The different penalty calculations are set in the *penaltycalcs.m* file.

Examples of all the above mentioned files can be found in the Appendix. The user defined values and inputs are extensively commented in the included files.

B: HAPMOEA OPERATION

The HAPMOEA code is run through Linux. The user must configure the loading of the variable limits manually into the *paracostwing.cpp* file. Due to the length of this file it is not included in the Appendix.

The upper and lower bounds for the variables passed into the optimiser, along with the size of each population, buffer size and the number of populations in the optimisation are set in the *optimisation.parameters* file. An example of an *optimisation.parameters* file can be found in the Appendix.

From a Terminal, the code is started by recompiling the program using the included MakeFile. This is done by typing *make Heawing* at the prompt. Once complete the program is executed as *Heawing*. Parallel Virtual Machine must be installed on the machine, and it must not be running.

During the code operation, nodes can be added or removed through the use of PVM. This is done by calling *pvm* in a separate terminal and using the standard *add* and *delete* commands followed by the node name.

OPTIMISATION.PARAMETERS

The *optimisation.parameters* file contains all the data required to run the HAPMOEA code. The file begins with a comment line and then goes on to describe the overall parameters for the optimisation such as the number of run and what the stopping conditions are. The next section details the values specifying the upper and lower bounds for the control points making up the aerofoils, internal structure and external properties. These lines simply contain the values with comments following the end of the input values using the C++ notation of '//'. Note that in this Appendix, some lines may be too long to fit on a single line and hence are continued on the next. For the input file though, this is not possible.

After the specification of the upper and lower bounds there is an optional input field for grid based CFD solutions detailing the size of the domain, Mach number and Reynolds number. This section is not used in this application of the HAPMOEA and can be ignored.

The migration properties are defined for the optimisation in the next section along with the number of populations to be generated and placed in a hierarchical layout.

The population, buffer and tournament size along with mutation operators are set for each level in the final section of the code.

```
PARAMETERS FOR AERO-STRUCTURAL WING OPTIMISATION
1         Number of runs
750     Maximum Function Evaluations
1e-5      Stopping Precision
7               //Mean Line control Points
0.100     0.200     0.400     0.600     0.700     0.800     0.900
//Mean line x-positions.
0.0074882 0.0199926 0.0299902 0.0255458 0.0199890 0.0122106 0.0022120
//Mean line lower bounds ROOT.
0.0274882 0.0399926 0.0499902 0.0455458 0.0399890 0.0322106 0.0222120
//Mean line upper bounds ROOT.
8   //Thickness control Points
0.0 0.100     0.200     0.400     0.600     0.700     0.800     0.900
//Thickness x-positions.
0.0453879  0.0234020 0.0286820 0.0290099 0.0228133 0.0183159
0.0131121 0.0072350    //Thickness line lower bounds ROOT.
0.0479176 0.0936080 0.1147281 0.1160395 0.0912532 0.0732634 0.0524484
0.0289402    //Thickness line upper bounds ROOT.
0.0074882 0.0199926 0.0299902 0.0255458 0.0199890 0.0122106 0.0022120
//Mean line lower bounds CRANK.
0.0274882 0.0399926 0.0499902 0.0455458 0.0399890 0.0322106 0.0222120
//Mean line upper bounds CRANK.
0.0453879  0.0234020 0.0286820 0.0290099 0.0228133 0.0183159
0.0131121 0.0072350    //Thickness line lower bounds CRANK.
0.0479176 0.0936080 0.1147281 0.1160395 0.0912532 0.0732634 0.0524484
0.0289402   // Thickness line upper bounds CRANK
0.0074882 0.0199926 0.0299902 0.0255458 0.0199890 0.0122106 0.0022120
//Mean line lower bounds TIP.
0.0274882 0.0399926 0.0499902 0.0455458 0.0399890 0.0322106 0.0222120
//Mean line upper bounds TIP.
0.0453879  0.0234020 0.0286820 0.0290099 0.0228133 0.0183159
0.0131121 0.0072350    //Thickness line lower bounds TIP.
0.0479176 0.0936080 0.1147281 0.1160395 0.0912532 0.0732634 0.0524484
0.0289402    //Thickness line upper bounds TIP
8 // Number of variables for the planform
5.0 1.6 0.0 0.0 0.35 0.35 0.5 0.0 // lower bounds for Aspect
ratio,rootchord,
sweeptostation1,sweeptostation2,chordratiostation2,chordratiostation3
,cranklocation1,alfa
26.0 1.7 0.1745 0.1745 0.65 0.65 0.8 6  //upper bounds for Aspect
ratio,rootchord,
sweeptostation1,sweeptostation2,chordratiostation2,chordratiostation3
,cranklocation1,alfa
11 // Number of variables for the structure
0.9 7.9 0.003 0.15 0.05 0.01 0.0005 0.05 0.05 0.0025 0.001  // lower
bounds for  numberofspars, numberofribs, ribrootthickness,
ribthicknesstaperratio, sparrootthickness,
sparthicknesstaperratio,skinthickness,skinratioroottip,
skinratioleadingtrailing,sparcaprootarea,ribcaprootarea
2.9 11.9 0.0065 0.3 0.09 0.2 0.002 0.15 0.15 0.04 0.002 // upper
bounds for  numberofspars, numberofribs, ribrootthickness,
```

```
ribthicknesstaperratio, sparrootthickness,
sparthicknesstaperratio,skinthickness,skinratioroottip,
skinratioleadingtrailing,sparcaprootarea,ribcaprootarea
0.684    // #mach
6.3e6  // # Reynols
-2.0   // # boundaryLeft
3.0    // # BoundaryRight
-3.0   // # BoundaryBottom
3.5    // # BoundaryTop
600       Live Update Interval (0 = NONE)
0         Force Synchronous Evaluations (1 == YES)
1.0       * Popsize Evaluations Before Migration
0.33      * Popsize Individuals to Migrate
1         Number of Hiearchical levels
- LEVEL ONE -
3       Level
30        Population Size
2         Parents in Recombination
1         Intermediate Recombination (1 = YES)
60        Buffer Length
2         Parent Selection Pareto-Tournament Size
1.0       Tournament-in-Buffer Ratio
0.1       Initial Mutation Multiplier
- LEVEL TWO -
2         Level
30        Population Size
2         Parents in Recombination
1         Intermediate Recombination (1 = YES)
60        Buffer Length
2         Parent Selection Pareto-Tournament Size
1.0       Tournament-in-Buffer Ratio
0.1       Initial Mutation Multiplier
- LEVEL THREE -
1         Level
30        Population Size
2         Parents in Recombination
0         Intermediate Recombination (1 = YES)
60        Buffer Length
2         Parent Selection Pareto-Tournament Size
1.0       Tournament-in-Buffer Ratio
0.1       Initial Mutation Multiplier
```

C: POST PROCESSOR OPERATION

Three Matlab® files are made available to the user to perform post-processing. All these files require the user defined files (*UserWing.m*, etc) to also reside in the current directory, along with the optimiser generated Pareto data.

PostProcessor.p performs a single full evaluation of a Pareto wing producing outputs of the aerodynamic and structural performance. The user is required to input the pareto wing number of choice.

Alphasweep.p performs an angle of attack sweep. The Pareto numbers for the wing numbers selected and the angles of attack of interest in degrees are specified before the execution of the function. They are array inputs and take the form of:

```
>> pareto = [0 7 14];
>> alphaangles = [0:0.5:4];
```

The user is given the option of including the Bench mark values specified in the User specified files (*UserWing.m*, etc).

MachSweep.p performs a Mach sweep. The input and program dependancies are of the same format as that for the *AlphaSweep.p* function namely the pareto numbers of interest, and the Mach numbers:

```
>> pareto = [0 7 14];
>>machnums = [0.5:0.025:0.6];
```

As with the *AlphaSweep.p* function, the user is given the option of including the Benchmark data.

D: AERO-STRUCTURAL PROGRAM USER DEFINED FILES

The following five files are examples of user defined input files for the Aero-Structural program.

USERWING.M

```
% Geometry Data used to define the aircraft wing
%
% (c) Lloyd Damp and Felipe Gonzalez
% September 2005

% User defined data

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%% GLOBAL HAWK Benchmark %%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% External Geometry
span        =   16.9461;          % Semispan (m)
rootchord   =   2.432;          % Root chord (m)
sweep       =   [0.1030 0.1030]; % Sweep angle (rad)
dihedral    =   0;              % Dihedral angle (rad)
cr          =   [1 0.78 0.287]; % Chord ratio

numcranks   =   1;              % Number of Cranks
crankloc    =   [0.0881];       % Position of Crank as fn
                                %   of span

% Internal Geometry
numspars    =   5;              % Number of internal spars
numribs     =   16;             % Number of internal ribs
mapribs     =   1;              % Number of added divisions in the
                                %   spanwise direction per panel
                                %   for aero calcs

% Rib Root Thickness. Follows the ribthicknessratio to the tip
ribthickness =  0.0015;
ribthicknessratio = 0.05;

% Spar Root Thickness. Follows the sparthicknessratio to the tip
sparthickness = 0.09;
sparthicknessratio = 0.05;

% Wing Skin Thickness. Constant
skinthickness = 0.001;
skinratioroottip = 0.01;
skinratioleadingtrailing = 0.01;

% Sparcap cross-sectional Area at root. Tapers at the same
%   ratio as the spars
sparrodA = 0.012;

% Ribcap cross-sectional Area at root. Tapers at the same
%   ratio as the ribs
ribrodA = 0.0005;
```

```
% Aerofoil Geometry (Define seperate upper and lower
%   surfaces from leading edge to trailing edge). Must define two
%   aerofoil shapes. The first one for the root, the second for the
%   tip. Use cell arrays.
% NASA LRN 1015 at both root and tip

aeroupper{1} = flipud([1.00000  0.00000
  0.99657  0.00146
  0.98712  0.00583
  0.97298  0.01248
  0.95446  0.02023
  0.93125  0.02865
  0.90350  0.03788
  0.87177  0.04788
  0.83659  0.05840
  0.79853  0.06915
  0.75812  0.07981
  0.71589  0.09000
  0.67236  0.09933
  0.62799  0.10732
  0.58299  0.11337
  0.53726  0.11727
  0.49085  0.11922
  0.44416  0.11947
  0.39771  0.11815
  0.35196  0.11535
  0.30741  0.11115
  0.26449  0.10564
  0.22364  0.09892
  0.18527  0.09110
  0.14974  0.08232
  0.11740  0.07273
  0.08855  0.06249
  0.06343  0.05179
  0.04226  0.04085
  0.02521  0.02995
  0.01241  0.01938
  0.00398  0.00957
  0.00008  0.00117]);

 aerolower{1} =  [0.00008  0.00117
     0.00208 -0.00535
  0.01053 -0.01088
  0.02456 -0.01608
  0.04389 -0.02071
  0.06828 -0.02466
  0.09749 -0.02791
  0.13119 -0.03046
  0.16903 -0.03230
  0.21056 -0.03346
  0.25529 -0.03397
  0.30270 -0.03384
  0.35221 -0.03306
  0.40319 -0.03160
  0.45501 -0.02941
  0.50699 -0.02620
  0.55895 -0.02150
  0.61119 -0.01549
  0.66352 -0.00895
  0.71529 -0.00256
```

```
  0.76576  0.00311
  0.81405  0.00755
  0.85913  0.01035
  0.89987  0.01127
  0.93510  0.01032
  0.96363  0.00772
  0.98422  0.00416
  0.99617  0.00114
  1.00000  0.00000];

% Crank aerofoil shape
aeroupper{2} = aeroupper{1};
aerolower{2} = aerolower{1};
```

USERAUXILLARYDATA.M

```
% User specified Auxilliary File constraints. Please check Panin
%   documentation for a list of acceptable commands.
%
% (c) Lloyd Damp and Felipe Gonzalez, 2005

% Global Hawk Test Case

altitude    = 15849.8;      % Altitude (m)
filename    = 'wing.wgs';   % WGS file name
mach        = 0.5983 ;      % Freestream Mach number
alph        = 4.75;          % Angle of Attack (degrees)
xref        = 0.6079;       % Chord reference point
bound       = [1 1 1];      % Surface Boundary descriptors
wake        = [1 1 10];     % Wake data
loadcase    = 2.5;          % Maximum maneuver (g's)

% % Altair Test Case
%
% altitude    = 12679;        % Altitude (m)
% filename    = 'wing.wgs';   % WGS file name
% mach        = 0.3663 ;      % Freestream Mach number
% alph        = 5.98883 ;           % Angle of Attack (degrees)
% xref        = 0.25;         % Chord reference point
% bound       = [1 1 1];      % Surface Boundary descriptors
% wake        = [1 1 10];     % Wake data
% loadcase    = 2.5;          % Maximum maneuver (g's)

% % Onera Data
%
% altitude = 1;
% filename = 'wing.wgs';
% mach = 0.8395 ;
% alph = 3.06;
% xref = 'cbar/4';
% bound = [1 1];
% wake = [1 1 50];
```

PENALTYCALCS.M

```
function [penvals] = penaltycalcs(varargin)

% Function to penalise the aircraft wing according to two
%   things namely aerodyanmics and structure.
%   Varargin structure used such that penalty calcs
%   can be performed in the panaircalcs routine to
%   determine whether or not a simulation passes or
%   fails the aerodynamic calculations prior to also
%   running the structural calcs.
%
% (c) Lloyd Damp
% November 2005

% Unpack Varargin data
if size(varargin,2) == 1,
    aerodynamics = varargin{1};
else
    structure = varargin{1};
    nastran = varargin{2};
    totfail = varargin{3};
    aerodynamics = varargin{4};
end

% Load penalty allowances and rates
penaltyallowrates;

% Load additional Data
newUserWing;

% Unpack data
if exist('structure','var') == 1,
    disp = structure.disp;
    mass = structure.mass;

    totnref = nastran.totnref;
    nref = nastran.nref;
    nu = nastran.nu;
    nl = nastran.nl;

    buck = totfail.buck;
    skin = totfail.skin;
    spars = totfail.spars;
    ribs = totfail.ribs;
    ribcap = totfail.ribcap;
    sparcap = totfail.sparcap;
end

cmy = aerodynamics.moment.my;
cl = aerodynamics.aero.cl;
cdtot = aerodynamics.aero.cd+aerodynamics.aero.cdform+...
    aerodynamics.aero.cdf;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%% Cmy Panelty %%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if (abs(cmy) > abs(cmyallow)) | (sign(cmy) >= 0),
    cmpenalty = exp(1+abs(cmy/cmyallow))*cmypenalty;
```

```
else
    cmpenalty = 0;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%% Lift Panelty %%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if abs(cl) < abs(mincl),
    clpenalty = exp(1+abs(mincl/cl))*clpenaltyrate;
else
    clpenalty = 0;
end

% Only run the next bit of code if all the simulations
%   have been performed, ie. Panair and Nastran
if exist('structure','var') == 1,

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%% Failure of panels Penalty %%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    failpanels = buck + skin + spars + ribs + ribcap ...
        + sparcap;

    if failpanels > 0,
        npenalty = failpanels*mass*numfailedpenalty;
    else
        npenalty = 0;
    end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%% Deflection of the wing tip %%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    % Upper surface deflection
    for a = 1:size(nu,1),
        deflu(a) = disp(find(disp(:,1) == nu(a,end)),4);
    end

    % Lower surface deflection
    for a = 1:size(nl,1),
        defll(a) = disp(find(disp(:,1) == nl(a,end)),4);
    end

    % Mean deflection as function of span
    def = mean((deflu + defll)./2)/span.*100;

    % Calculate the deflection penalty
    if abs(def) > abs(deflectallow),
        dpenalty = exp(1 + abs(def/deflectallow))*...
             mass*deflectpenalty
    else
        dpenalty = 0;
    end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%% Twist of the wing tip %%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    % Calculate tip deflection values
```

```
    leadu = disp(find(disp(:,1) == nu(1,end)),4);
    leadl = disp(find(disp(:,1) == nl(1,end)),4);
    lead = (leadu + leadl)/2;
    trailu = disp(find(disp(:,1) == nu(end,end)),4);
    traill = disp(find(disp(:,1) == nl(end,end)),4);
    trail = (trailu + traill)/2;

    % Calculate the angle of twist for the wing tip
    twist = asin(abs(lead-trail)/rootchord)*180/pi;

    % Calculate Twist Penalty
    if twist > twistallow,
        tpenalty = exp(1 + abs(twist/twistallow))*...
             mass*twistpenalty;
    else
        tpenalty = 0;
    end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%% Balance Penalties  %%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    % If the wing is over penalised through additional mass
    %    the pareto front can be distored due to this error.
    %    Therefore, if the mass penalties exceed 10% of the
    %    total structural mass the drag is equally
    %    penalised by the same percentage.

    if (tpenalty + dpenalty + npenalty) > penbothmass*mass,

        % Calculate new coefficient of drag penalty by
        %     calculating the percentage overshoot of mass
        percmass = (tpenalty+dpenalty+npenalty)/mass;

        % Calculate new cd penalty
        newcdpenalty = cdtot*percmass;
    end

    % If the wing is over penalised through additional drag
    %    the same error in the pareto front can be made.

    if (clpenalty + cmpenalty) > penbothcd*cdtot,

        % Calculate new coefficient of drag penalty by
        %     calculating the percentage overshoot of mass
        perccd = (clpenalty + cmpenalty)/cdtot;

        % Calculate new mass penalty
        newmasspenalty = mass*perccd;
    end

    if exist('newmasspenalty','var') == 1,
        % Check if new cm penalty is larger than the
        %    summation of the existing cd penalties
        if (tpenalty + dpenalty + npenalty) < newmasspenalty,
            tpenalty = newmasspenalty;

            % Reset the other mass penalties
            dpenalty = 0;
            npenalty = 0;
        end
```

```
    end

    if exist('newcdpenalty','var') == 1,
        % Check if new mass penalty is larger than the
        %    summation of the existing mass penalties
        if (clpenalty + cmpenalty) < newcdpenalty,
            cmpenalty = newcdpenalty;

            % Reset the coefficient of lift penalty
            clpenalty = 0;
        end
    end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%% Penalty Percentages %%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    penvals.struct.mass = (tpenalty + dpenalty + npenalty)...
        /mass;
end

penvals.aero.cd = (cmpenalty + clpenalty)/cdtot;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%% Build the output %%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

penvals.aero.cmpenalty = [cmpenalty+clpenalty];

if exist('structure','var') == 1,
    penvals.struct.twistpenalty = [tpenalty];
    penvals.struct.deflectionpenalty = [dpenalty];
    penvals.struct.numfailedpenalty = [npenalty];
end
```

MATERIALPROPERTIES.M

```
% Material Properties for Nastran Input Files
%   User specifies the materials as vectors where each entry
%     in the vector corresponds to a material. Only define the
%     Youngs Modulus and Poissons Ratio for each material.
%
%   The number of materials being investigated MUST be stated!
%
%   The Material Identification Number for each type of element
%     must also be assigned.
%
%   Make sure each entry is less than 8 characters in length!
%
%(c) Lloyd Damp and Felipe Gonzalez 2005

%%%%%%%%%%%% Number of Materials %%%%%%%%%%%%
mat.num = 1;

%%%%%%%%%%%% Data for each Material %%%%%%%%%%

% Graphite-Epoxy (T300 - 1K carbon/M76 epoxy)
% Youngs Modululs (Pa)
mat.E = [1.53e11];
```

```
% Poissons Ratio
mat.NU = [0.3];
% Density (kg.m-3)
mat.rho = [1.31e3];
% Ultimate Tensile Strength (Pa)
mat.tensile = [1532e6];
% Longitudinal Compressive Strength (Pa)
mat.compression = [947e6];
mat.strainallow = 0.00333;

%
http://heseweb.nrl.navy.mil/glast/CALPeerDesignRev/RFAs/RFA%20CAL006-
0305
% 03.pdf

%%%%%%%%%% Material for each Element %%%%%%%%%%
% mat.element = [wing skin, spars, ribs, ribcap, sparcap];
mat.element = [1 1 1 1 1];
```

PENALYALLOWRATES.M

```
% Penalty Allowances and Rates
% (c) Lloyd Damp 2006

% Allowed Values
twistallow = 1;             % Allowed wing tip twist in degrees
deflectallow = 20;          % Allowed wing deflection as a
                            %   percentage span
cmyallow = -0.3041;         % Allowed wing moment (benchmark)
mincl = 0.89;               % Minimum lift to be generated by
                            %   by the wing

% Penalty Values
twistpenalty     = 0.1;     % Wing Mass per degree twist
deflectpenalty   = 0.1;     % Wing Mass per degree over
                            %   20 % span
numfailedpenalty = 0.1;     % Wing Mass per failed
                            % panel
cmypenalty = 0.001;         % Additional Cd per % over
                            %   allowable
clpenaltyrate = 0.005;      % Additional Cd per % less
                            %   than the required minimum

penbothcd = 0.1;
penbothmass = 0.1;

penfailpanair = Inf;
penfailnastran = Inf;
```

E: AERO-STRUCTURAL OPTIMISER DESTINED INPUT AND OUTPUT FILES

The following section contains the *simresults.txt* file outputted by the aero – structural program on termination of the simulation and examples of the *planformVariables.xxx*, *AuxialiaryVariables.xxx*, *aerfoilsData.xxx* and *structuralVariables.xxx* files created by the optimiser to describe a candidate wing for evaluation.

SIMRESULTS.TXT

```
% Simulation performed on 12-6-2006 at 22:36:18
%
% Parameters written in the following order if they exist:
%      Friction convergence (0 = no errors, 1 = errors)
%      Panair convergence (0 = no errors, 1 = errors)
%      Nastran convergence (0 = no errors, 1 = errors)
%      Cl
%      Cd (Induced - Panair)
%      Cd (Form    - Friction)
%      Cd (Viscous - Friction)
%      Cd (Total)
%      Cmx
%      Cmy
%      Cmz
%      Wing Mass (kg)
%      Converged Load Factor
%      Number of failed Skin Quads due to buckling
%      Number of failed Spar Quads due to excessive strains
%      Number of failed Rib Quads due to excessive strains
%      Number of failed Sparcaps due to excessive strains
%      Number of failed Ribcaps due to excessive strains
%      Maximum Displacement in X (m)
%      Maximum Displacement in Y (m)
%      Maximum Displacement in Z (m)
%      Z Displacement as percentage span (%)
%      Cmy Penalty (additional Cd)
%      Wing Tip Twist penalty (additional mass)
%      Wing Tip deflection penalty (additional mass)
%      Number of failed panels penalty (additional mass)

0
0
0
0.85968
0.01936
0.00046
0.00393
0.02375
0
-0.05979
0
524.869
```

```
1
0
0
0
0
0
0.0464668
1.48026
5.84168
36.0624
0.00757909
0
167.496
0
```

PLANFORMVARIABLES.XXX

The *planformVariables.xxx* file is a two line file where the first line identifies the variables being passed from the HAPMOEA optimiser to the Aero-Structural solver in a comma separated list. The second line is an array of the corresponding variable values with whitespace as the delimiter.

```
%Planform Variables: semispan, rootchord,
sweeptostation1,sweeptostation2,chordratiostation2,chordratiostation3
, cranklocation1, additionalspanwisepanels
16.76 2.42226 0.0134707 0.0338693 0.589011 0.521044 0.0777902 2
```

AUXIALIARYVARIABLES.XXX

As the angle of attack was the only design variable in the auxiliary data class which was optimised, the *AuxiliaryVariables.xxx* file only contained that value. The format for this file is the same as that used for *planformVariables.xxx*.

```
%Auxiliary Variables: alfa
5.63689
```

AEROFOILSDATA.XXX

An abbreviated example of an *aerofoilsData.xxx* file is shown below. Comments are inserted as hash symbols at the beginning of a line and the command *#999.0 999.0* informs the parser that the file is about to start describing a new aerofoil section.

```
#Current Aerofoils :T/E to L/E around upper surface back again to T/E
lower surface
1 1e-06
0.998674 0.000133459
0.9947 0.000538285
…
0.00338262 0.00996959
0.00105883 0.00508043
0 0
0.00029532 -0.00518459
0.0020236 -0.0103641
…
0.994688 -0.000391507
0.998671 -9.4675e-05
1 -1e-06
#999.0  999.0
1 1e-06
0.998675 0.000161899
0.994702 0.000632937
0.988096 0.00137853
…
0.00349694 0.0110639
0.0010994 0.00543034
0 0
0.000252948 -0.00553771
0.00190587 -0.0114742
…
0.994686 -0.000460776
0.99867 -0.000116364
1 -1e-06
```

STRUCTURALVARIABLES.XXX

The *structuralVariables.xxx* files follow the same format as that used in both the *AuxiliaryVariables.xxx* and *planformVariables.xxx* files. The file is essentially only two lines long with the variable names in the first line, and the values in the second.

```
%%Structural Variables: numberofspars, numberofribs,
ribrootthickness, ribthicknesstaperratio, sparrootthickness,
sparthicknesstaperratio,skinthickness,skinratioroottip,
skinratioleadingtrailing,sparcaprootarea,ribcaprootarea
3.63124  12.7355  0.00256047  0.133124  0.0637379  0.0885325  0.0245781
0.0656737 0.0738991 0.00595451 0.000733436
```

F: MSC.NASTRAN® INPUT BDF

The file below is an abridged version of a Bulk Data File generated by the Aero – structural program. The file is used by the commercial program MSC.Nastran® to calculate the structural response of a candidate wing to the calculated aerodynamic loading.

```
$ Nastran Bulk Data File automatically generated by
$ Lloyd Damp on 05-Jun-2006
$ at 20:3:17
$$ Executive Control Section $$
$ Linear Static Analysis
SOL 106
CEND
$$ Case Control Section $$
$ Project Title
TITLE = wing
$ Superelement Stuff
SEALL = ALL
SUPER = ALL
$ No printout of Bulk Data
ECHO = NONE
$ Set Subcase and name
SUBCASE 1
      SUBTITLE=Default
$ Set Load
      LOAD = 1
$ Enforce Single Point Constraints
      SPC = 1
$ Displacement Output Data
      DISPLACEMENT(SORT1,REAL)=ALL
$ Set Non-Linear Parameters
      NLPARM = 1
$ Output Element Mass Data
      ELSUM(EIDSUM) = ALL
$ Output Stress data
      STRESS(SORT1,REAL,VONMISES,CENTER)=ALL
$$ Bulk Data Section $$
BEGIN BULK
$ Set output filename
PARAM,POST,0
$ Automatic constraint of singularities,etc
PARAM,AUTOSPC,NO
$ I have no idea what this next line does!
PARAM,PRTMAXIM,YES
$ Displacement Stuff for Non-linear Solution
PARAM,LGDISP,1
$ Non-linear Parameters for solution
NLPARM   1      100              AUTO    1       25      PW      NO
                 .1      .1
$ Element Properties
…
PSHELL,       2,       1,0.0904213,       1
…
$ Rib Cap Rods
…
```

```
PROD*   20              1               0.00333         0
*SC20
*SC20   0               4.3723
PROD*   21              1               0.00294         0
*SC21
*SC21   0               3.8614
…
$ Upper Surface Quadrilateral Element Data
…
CQUAD4        19       1     148     283     281     146
                         0.00078  0.0006 0.00054  0.0007
CQUAD4        20       1     150     285     283     148
                          0.0007 0.00054 0.00048 0.00062
CQUAD4        21       1     152     287     285     150
                         0.00062 0.00048 0.00042 0.00054
…
$ Lower Surface Quadrilateral Element Data
…
CQUAD4        80       1    1218    1407    1409    1220
                         0.00086  0.0006 0.00054  0.0008
CQUAD4        81       1    1220    1409    1411    1222
                          0.0008 0.00054  0.0005 0.00074
…
$ Spar Quadrilateral Element Data
…
CQUAD4       132       1     462    1407    1409     464
                         0.06847 0.05933 0.05933 0.06847
CQUAD4       133       1     464    1409    1411     466
                         0.05933 0.05232 0.05232 0.05933
…
$ Rib Quadrilateral Element Data
…
CQUAD4       171       1     468    1413    1575     630
                          0.0007  0.0007  0.0007  0.0007
CQUAD4       172       1     146    1091    1226     281
                         0.00062 0.00062 0.00062 0.00062
…
$ Rib Triangular Element Data
…
                          0.0007  0.0007  0.0007
CTRIA3       204       1     632    1577     929
                         0.00062 0.00062 0.00062
CTRIA3       205       1     634    1579     931
                         0.00054 0.00054 0.00054
…
$Ribcap Element Data
      $Upper Surface
…
CROD,229,7,144,279
CROD,230,7,144,1089
…
      $Lower Surface
…
CROD,308,10,1232,1421
CROD,309,10,1421,1583
…
$Sparcap Element Data
      $Upper and Lower Surface
…
CROD,354,25,1230,1232
CROD,355,26,287,289
```

```
…
$ Description of Material : Date: 05-Jun-2006
MAT1*   1               153000000000.0                  0.30
*M1
*M1     1310.01
$ Nodes of the Entire Model
…
GRID,460,0,1.195,0.0001,0.10975
GRID,462,0,1.13415,1.24616,0.08953
…
$$ Pressure Distribution Data for Each Surface Panel $$
$ Upper Wing Surface Pressures
…
PLOAD2* 2               -13889.779      9
*P9
*P9
PLOAD2* 2               -14915.589      10
*P10
*P10
…
$ Lower Wing Surface Pressures
…
PLOAD2* 3               3808.320        74
*P74
*P74
PLOAD2* 3               3800.125        75
*P75
*P75
…
$ Gravity Vector
GRAV,4,0,9.81,0.0,0.0,-1.0,-1
$ Combine all Loads (Pressure and Gravity)
LOAD,1,1.0,1.0,2,1.0,3,1.0,4
$$ Node Displacement Constraints $$
SPC1,1,123456,136,271,460,622,919,1081,,
,,1216,1405,1567
ENDDATA
```