Global Multi-Disciplinary Design Optimisation of Super-Hypersonic Components and Vehicles

Andrew Cusick

Submitted in fulfilment of the requirements for the
Degree of Doctor of Philosophy

Aerospace Research Sciences Division
School of Engineering
College of Science and Engineering
University of Glasgow

September 2021

Executive Summary

Both high-speed transportation and reusable launch vehicles are once again under constant development, after reaching stagnation points due to immature technologies. This resulted in many programs being shut down prior to launch, while the fortunate few that made it to production, were ultimately cancelled due to their combination of unreliability and high operational costs. Today, national institutions and private organisations alike are competing to revolutionise the super-hypersonic regime, but optimal performance remains as imperative as ever if such concepts are to be realised as long term solutions.

The focus of this research is to develop a framework capable of generating, analysing, and optimising such vehicles; producing high performance designs that comply with real-world constraints from a broad spectrum of potential candidates. This contrasts with the majority of design optimisation work found in aerospace engineering literature, which favours improving existing designs through localised alterations. While this is useful in many engineering applications, it means little in the context of high-speed transport and reusable launch vehicles; since reliable designs do not yet exist, and current concepts vary drastically based on a number of factors. The direction of this research is therefore to define optimality on a larger scale, based on performance metrics that are transcendent among all super-hypersonic vehicles.

Quantifying the performance of such vehicles in this large scale conceptual design context, in terms of both optimality and feasibility, requires the development of versatile analysis and optimisation tools. A number of vehicle generation and integration tools have been developed, combined with low-fidelity aerodynamic methods, particle swarm optimisation, and robust constraint handling techniques, to provide the basis of this research. The justification, necessary background, and implementation of these methods is discussed in detail, alongside their validation in a large design space context. Further analysis modules and optimisation techniques are employed in addition to these primary methods, which are used to provide a deeper understanding of vehicle performance, optimality, and feasibility.

Initial optimisation work is focussed on aerofoils. A number of generative methods are employed and tested for their level of control over such shapes, since fine tuning will be required

to achieve globally optimal results. Following this, aerodynamic optimisations are carried out, demonstrating the combined methods and their ability to consistently produce high performance results in both existing design and global optimisation contexts. This work is then extended to multi-fidelity analysis; where open-source high-fidelity programs are employed efficiently through the implementation of dynamic surrogate modelling techniques. The combination of these methods with the existing framework is used to improve the quality of results.

The remaining work is based on optimisation of three dimensional bodies; with both wing and full vehicle design spaces employed. A low-fidelity wing-box model is implemented in the former case, adding further design variables and constraints to ensure structural integrity within optimal designs. A prior reusable launch vehicle concept is used to benchmark these results, allowing trim and longitudinal stability characteristics to be determined. An improved parametric fuselage definition is then employed, enabling full vehicles to be designed and optimised from an arbitrary starting point. Uncertainty quantification and improved constraint handling methods are utilised to increase confidence in vehicle performance. Finally, a mass estimation routine is implemented to determine weight, payload and trim characteristics; with surface temperatures, lateral stability, and payload integration defining further feasibility characteristics.

Contents

List of Figures

List of Tables

Acknowledgements

This thesis represents four years of research made possible by the support of a number of people in a number of different ways. What it does not capture is the experience it has provided me with: to work freely on something I truly enjoy; to travel, meet and work with some extraordinary people; and shape my perspective in a way no other line of conventional work could. Undertaking this research, although very challenging at times, has been one of the best decisions of my life, and it is important to me that all the people who have been there along the way, both academically and personally, understand how much I appreciate the interactions we've had.

I would like to thank my supervisor, Professor Konstantinos Kontis, for his support and guidance throughout this research; and Professor Melike Nikbay, for becoming actively involved in this work, and hosting me at ITU. Their continued support and technical contribution made this project possible, and I am grateful for all of the advice and discussions that helped shape this research.

Without a few wonderful individuals, this journey would not have been the same. Firstly, thanks goes to Angus Brown, for being a constant source of support and enjoyment throughout our shared experience. To Andrew Russell, who was a tremendous technical sounding board, but more importantly a fantastic source of distraction. Speaking of distractions, this experience was made considerably more enjoyable through the friendship of Caroline Henderson, who added some much needed excitement to daily office life. Finally, to Ipek Karakurt, who turned a research trip into an experience worth so much more, and provided me with constant support throughout what was undoubtedly the toughest year of my research.

Finally, to those who have been with me from the start; my family and friends, who would support me through any adventure or adversity. I am forever grateful for the constant encouragement and joy that you provide, particularly throughout the last year.

This research, and more importantly, the experience I've been fortunate enough to have, would not have been possible without the EPSRC research grant (no. EP/N509668/1), and the University of Glasgow mobility scholarship, along with their continuous support throughout.

Author's Declaration

"I declare that, except where explicit reference is made to the contribution of others, that this dissertation is the result of my own work and has not been submitted for any other degree at the University of Glasgow or any other institution."

Printed Name: Andrew Cusick

Date: 17/05/2021

Nomenclature

Units

All units of measurement throughout conform to the Internation System of Units (SI).

Symbols

The following are standardised definitions used throughout this research, with briefly adopted terms specified at the time of use.

Latin

a = Polynomial coefficients

A = Area

b = Wingspan

B = Bernstein polynomial

c = Chord

C, S = Class, shape functions

C_D = Drag coefficient

C_L = Lift coefficient

C_m = Moment coefficient

C_p = Pressure coefficient

E = Young's modulus

f = Cost function

g = Inequality constraints

h = Equality constraints/Height

I = Area moment of inertia

J = Polar moment of inertia

$\mathbf{K}$ = Stiffness matrix

L = Lift/Length

M = Mach number

$\mathbf{n}$ = Normal vector

n_{var} = Number of optimisation variables

Pr = Prandtl number

r = Radius/Recovery factor

R = Penalty factor

Re = Reynolds number

t = Thickness/Iteration

T = Temperature/Number of iterations

$\mathbf{u}$ = Vector of displacements and rotations

V = Velocity

W = Weight

$\mathbf{x}$ = Design variable vector

x, y, z = Cartesian coordinates

Greek

α = Angle of attack

β = Sideslip/2D Shockwave angle

γ = Ratio of specific heats

ε = Constraint tolerance

η = Efficiency factor

θ = Inclination angle/Mesh rotations

Λ = Wing sweep

λ = Taper ratio

μ = Mean value/Dynamic viscosity

ν = Prandtl-Meyer relation

ρ = Density

σ = Standard deviation/Boltzmann constant

τ = 3D Shockwave angle

ϕ = Dihedral/Violation function

ω = Aitken's relaxation/Inertia parameters

Subscripts

$*$ = Reference conditions

$1, 2$ = Upstream, downstream

∞ = Freestream

e = Boundary layer edge

le, te = Leading and trailing edge

w = Wall

u, l = Upper, lower

Superscripts

$\hat{}$ = Normalised

Abbreviations

CFD	Computational fluid dynamics
DOE	Design of experiments
GA	Genetic algorithm
GPR	Gaussian process regression
HST	High-speed transportation
HTHL	Horizontal take-off horizontal landing
IIM	Inverse parametric mapping
LEO	Low Earth orbit
PSO	Particle swarm optimisation
RANS	Reynolds-averaged Navier-Stokes
RLV	Reusable launch vehicle
RMSE	Root mean square error
SLR	Singly-linked ring
SQP	Sequential quadratic programming
SSTO	Single-stage-to-orbit
TSTO	Two-stage-to-orbit
VTVL	Vertical take-off vertical landing

Chapter 1

Introduction

Until recently, the space race seemed a distant memory, with public interest and funding dwindling following the incredibly successful Apollo program. Considering the rapid advancement of the 1960s: from no human ever reaching space, to walking on the moon (and perhaps more impressively, returning safely to Earth), it is no surprise that we expected astronauts to step foot on other planets long before now. Of course, there are many scientific and engineering successes to celebrate in the time since then; such as the establishment of space stations, reaching outside the solar system with the Voyager program, and the now relatively routine deployment of satellite systems both large and small. Furthermore, the advancements in robotics have allowed samples to be taken from across the solarsystem, removing the huge challenges that come with human transportation. But despite these tremendous achievements, it is understandable that if you could go back to the golden age of human space exploration, people may be surprised to know that no human has ever escaped Earth's orbit. There are of course a variety of reasons for this, but many failed attempts to make space more accessible demonstrate the enormous technical challenges faced by engineers. Maturation of technology has therefore taken considerably longer than expected, which leaves us without the key characteristic required to drive down the cost of spaceflight: reusability.

Originally thought to be commonplace by the turn of the century, no fully reusable launch vehicle (RLV) has ever made it to space, despite numerous lucrative development contracts. Such a vehicle would drastically reduce the cost of space missions, and in turn, exponentially increase market accessibility and size. Not only would orbital systems be considerably cheaper to deploy, but rapid turnaround times could lead to large space based infrastructure, along with the development of highly profitable tourism and transportation ventures. Yet the failed large scale projects of the past have made governmental bodies and private companies weary of undertaking ambitious space ventures, delaying the grand opening of this game changing global market.

However, a rejuvenation in space programs over the last decade has brought about a much more

peaceful space race than its predecessor, with a number of private companies attempting to capitalise on the untapped market. These organisations have learned from the failures of past programs, and can afford to invest the time and money into incrementally developing the necessary capabilities for efficient spaceflight. In particular, modern day programs commonly involve partially reusable launch vehicles to be developed first, allowing the most technically challenging components, such as efficient propulsive methods, to be worked on in the background while a variety of other technologies are tested and proven. Companies such as SpaceX, Virgin Galactic, Blue Origin and Reaction Engines, who are willing to pay the heavy price of research and development, realise that successful demonstration of such technologies will be extremely lucrative in the long run.

Much like launch vehicles, high-speed transport (HST) aircraft also reached a stagnation point, with the second of only two successful supersonic commercial jets retiring from service in 2003. Despite an increasingly interconnected world in which technology only seems to become more streamlined and accessible, flight transportation appears to be the only innovation that has quite literally slowed down. Considering the hindrance this imposes on global accessibility, it is in many ways surprising that flight times have not significantly reduced since the introduction of the Tupolev Tu-144 in 1968, followed by the Concorde in 1969. What it shows then, is the significant technical challenge required in designing and manufacturing a vehicle not only capable of performing at high speeds, but doing so in a cost effective manner.

Unsurprisingly, the private sector has not overlooked this opportunity either, with a number of start-up companies developing radically different high-speed concepts. These variations exist not only at the design level, but within business models; with some seeking to develop private jets for the super-rich, before branching out to commercial airliners. In a similar fashion to the planned iterative integration of reusable technology within RLV concepts, this approach allows the necessary commercial HST technology to mature during the development of smaller vehicles, that can be entered into service in a much shorter timeline. For example, one of the major hurdles in high-speed flight is minimisation of sonic boom signatures, which prevented Concorde from flying supersonic over North America. This problem only becomes more challenging with larger aircraft, providing some technical validation for an initial entry into the business jet market.

1.1 Research Motivation

1.1.1 Historical Perspective

Fully reusable launch vehicles have proved to be a significant, and so far overwhelming technical challenge. Research and design of reusable two-stage-to-orbit (TSTO) vehicles has peaked and troughed since the 1930s, with its single-stage counterpart (SSTO) receiving more attention in the 1960s [1]. The step towards entirely reusable spacecraft provides an abundance of advantages, with the most notable being a drastic decrease in the cost of manufacturing and integrating partially reusable or single use components. Furthermore, a huge reduction in overall turnaround time between missions would vastly increase access to space for not only commercial purposes, but potentially transport and tourism markets also. With improved access being a constant high-priority topic at the time, NASA produced a report in 1994 [2], anticipating that full RLVs would replace the space shuttle by the late 2000s. Despite their early conception and promise to revolutionise the space market as a whole, a fully reusable orbital system has not yet been successfully deployed, with only sub-orbital or partially reusable systems currently in place. Many factors have contributed to this slower than anticipated progress, such as drastically reduced funding and general public interest around the turn of the century. Nevertheless, the various aerothermodynamic, propulsive, structural and mission specific challenges associated with the development of fully reusable launch vehicles cannot be understated.

Throughout research and conceptualisation of SSTO or TSTO RLVs, the over-arching problem has been their inability to deliver a reasonable payload fraction to at least low Earth orbit (LEO) [1,3]. This makes costly development programs unattractive considering the small initial benefits in the best case, and potential of a completely infeasible design in the worst. However, with continued advancements in propulsion, materials, and various other technologies, commercially viable fully reusable launch vehicles with increasing payload fractions appear more feasible and realistic. Considering that no successful configuration exists yet, it is not surprising that organisations who have undertaken the challenge produce radically different designs; varying not only in size and shape, but in propulsive methods, take-off/landing approaches, and mission specifications, to name a few.

Many failed attempts exist in the pursuit of a fully reusable launch system, with a need for maturation of various technologies being a common theme amongst the dissolved projects. In 2003, Bayer [4] compared a number of rocket propelled vertical take-off/vertical landing (VTVL), vertical take-off/horizontal landing (VTHL), and horizontal take-off/horizontal landing (HTHL) SSTO concepts that existed at the time; encompassing ballistic, wing-body, and lifting-body designs. Bayer pointed out the strong design sensitivities and risks that are present in the initial concept selection, emphasising that often performance degradation throughout design iterations

Figure 1.1: X-33 candidate designs. From left to right: Lockheed Martin VentureStar (winning bid), McDonnell Douglas DC-X, and Rockwell wing-body concept

coupled with increasing fidelity of analysis can severely damage or completely negate their practical feasibility. The X-33/VentureStar (VTHL) RLV program [5] proved to be an excellent example of both the aforementioned themes, with candidate designs shown in figure 1.1. During a ground test of the system, a highly complex hydrogen tank design failed, affirming the need for technology maturation. As the X-33 design process matured itself, various alterations were deemed necessary, resulting in a drastically increased take-off weight. A combination of these factors, among others, resulted in the termination of the X-33/VentureStar program, and provided a harsh warning to future SSTO/TSTO concepts.

A second but radically different design involved in the X-33 project was the VTVL Delta Clipper Experimental (DC-X) concept [6]. While not selected as the final design, multiple successful test flights were conducted as part of a separate project, demonstrating the capability of vertical landings and rapid turnaround time. Although the DC-X program was also prematurely discontinued; in this instance following a catastrophic failure on landing during a test flight, its legacy is clear to see today, with various organisations employing a vertical landing approach for their RLV programs.

Completing the wide spectrum of concepts during this period of interest in space accessibility, the wing-body X-34 configuration [7] was developed as a low Earth orbit small payload launcher. It would be deployed from a carrier aircraft, reaching altitudes over 75000m, before

returning to Earth via an unpowered re-entry/descent, and landing on a conventional runway. Much like the X-33/VentureStar project, the X-34 was cancelled prior to flight, but once again its legacy has provided a benchmark for future designs of carrier aircraft launched or HTHL small payload launchers and beyond [8].

Following these ambitious SSTO projects, organisations have tended towards more conservative TSTO developments, with a recent space boom being attributed to the successful reusable rocket capabilities demonstrated by SpaceX [9]. The recent report by Niederstrasser [10] detailed this further, with various organisations deciding to take on the final frontier as of late. While not all of these concepts are fully reusable ones, it is apparent in the current climate that reusability is the long term goal required for the industry to move forward as a whole.

As for high-speed transport vehicles, the Concorde boasted significantly reduced flight times, such as London to New York in under 3 hours. However, carrying only 55 passengers made the supersonic jet unsustainable, as extortionate seat prices were required to make the Concorde financially feasible. Since its retirement, the aviation market unsurprisingly attempted to maximise passenger numbers, which has also proved to be a poor business model; turning companies towards more efficient, two engine long-range aircraft. Nonetheless, the demand for global mobility by companies and tourists alike is only rising in a world that is becoming increasingly wealthy and interconnected. The lessons learned from Concorde and the recent trends of the aviation industry as a whole provide a route map for those wishing to take on supersonic transport, and many companies are taking the leap.

1.1.2 Current Concepts

Much like their predecessors, current concepts for reusable launch vehicles and high-speed transport aircraft vary drastically in their nature. Both concepts face common problems, specifically relating to suitable materials for the high temperatures experienced at super-hypersonic speeds; the selection of an efficient propulsion system for the given mission profile; maintaining low noise levels, particularly due to large boom signatures; and many others. These account for much of the variance between concepts. However, differing experience between institutions, and a general lack of experience with respect to super-hypersonic vehicles, certainly contribute to this large variation. Key technology maturation issues are evidently necessary to achieve the performance gains required to create a feasible, enticing business proposition. Thus, it is not surprising that current designs vary drastically between competing companies, with independent technology development programs driving further system level discrepancies.

Looking specifically at RLVs, figure 1.2 shows a number of current concepts under development, each with radically different design choices. The Indian Space Research Organisation

(ISRO) concept represents a conventional wing-body design, intended to be stacked on a larger rocket before staging close to LEO. Similar wing-body concepts exist with differing launch mechanisms, such as the Orbital 500R, which is intended to be launched from a carrier aircraft. The SpaceX Starship has undergone a number of recent test flights (often ending in dramatic explosions) which demonstrates a particularly nuanced RLV design. This enormous two-stage rocketship is a fully reusable VTVL vehicle, which pitches down at high-altitude re-entry, made possible due to its large cylindrical shape, alleviating thermal loads on the surface. The Reaction Engines Skylon concept also intends to re-enter at higher altitudes due to its large fuselage, while also incorporating large wings to allow for a HTHL launch. Successfully designing an SSTO that can take-off from conventional runways would revolutionise the space industry, and it is no surprise that a revolutionary propulsive system is required to achieve this.

In terms of HSTs, a large passenger aircraft capable of competing with the current subsonic state-of-the-art would be an overly ambitious project, considering the current level of supersonic transportation experience and technology. Even though decreased travel times would make a less efficient aircraft commercially attractive, generations of technology maturation are required before this becomes feasible. Instead, companies such as Aerion and Spike Aerospace are embracing the small number of passengers constraint. Whereas the Concorde tried to compete in the commercial market, these designs are made for private travel; selling time savings to those who can afford to pay the price. Commercial designs are still under development however, with Boom Supersonic seeking to replicate similar passenger numbers to that of Concorde, but at a considerably lower cost. Note that during the write up of this work, Aerion also released their plans to develop a similar Mach 4+ commercial airliner, further confirming the iterative technology maturation business model discussed previously.

Advancements in propulsive technology promise to increase the viability of both RLV and HSTs. Particularly in the RLV case, reducing the reliance on purely rocket based methods is a necessity, due to their inefficiency at low altitudes. While utilisation of air-breathing engines for use at low altitudes en route to space is not a new concept in itself [11], it was often shown to provide only marginal performance gains, if any, due to the increased structural requirements of the air-breathing system [1]. With the projected increase in efficiency of combined rocket/air-breathing propulsion configurations [12], an even more diversified set of RLV concepts have emerged. Regardless of the chosen engine architecture, structural weight and propellent storage remain key concerns, and the selection of propulsive method will inevitably have a large impact on the overall shape of the vehicle as a whole.

This work does not intend to constrain designs by employing a specific solution to a problem such as propulsive methods or launch mechanism, when technology is yet to mature and many

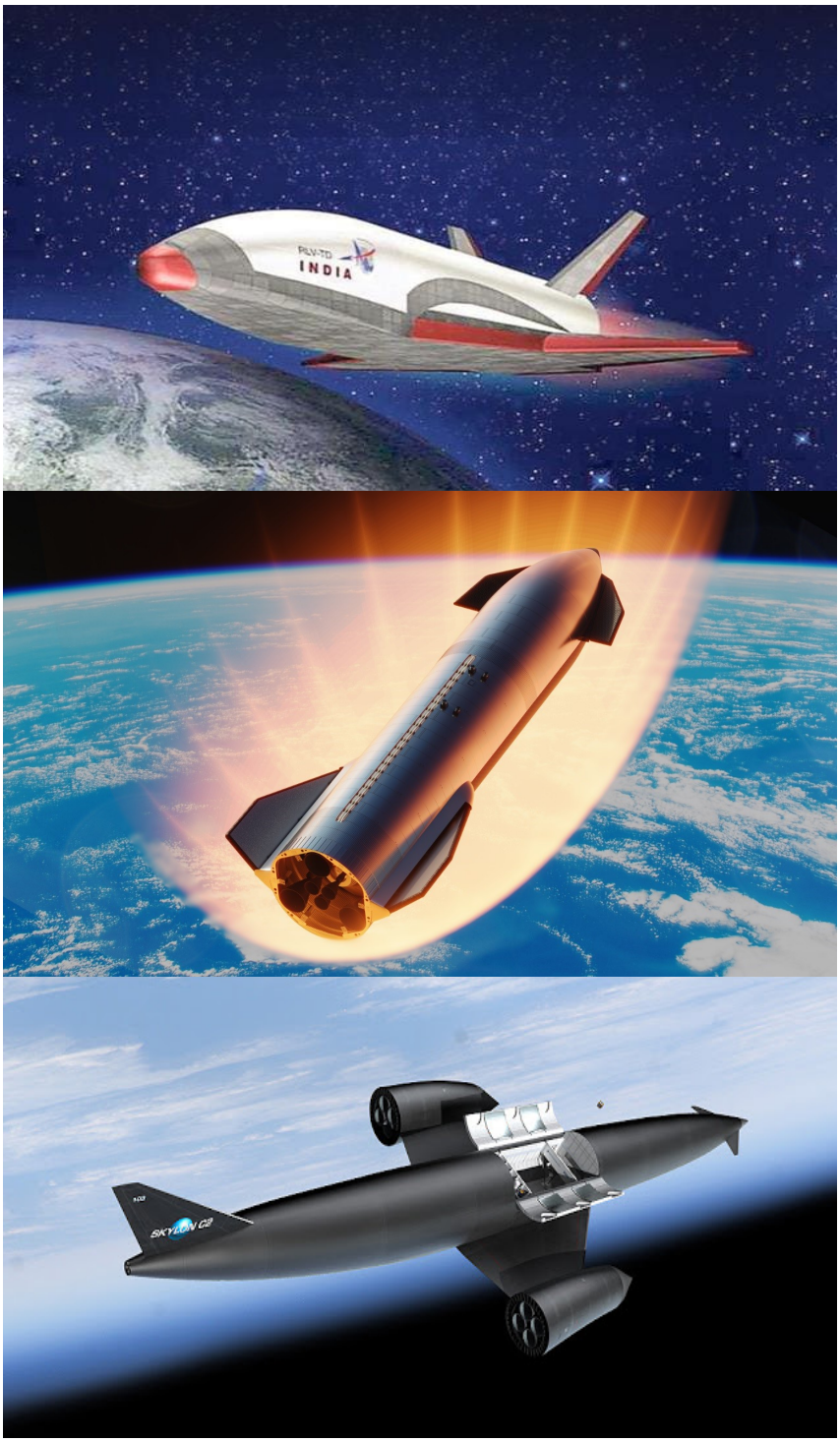


Figure 1.2: Current RLV concepts under development. From top to bottom: ISRO RLV, SpaceX Starship, and Reaction Engines Skylon

candidate solutions are under development. Instead, focus is placed on common quantifiable performance characteristics across the super-hypersonic domain, generating a number of candidate designs independent of these system level decisions that are yet to be realised. By developing techniques capable of producing high-performance designs from this global standpoint, results of equivalent if not higher value can be obtained when such system level constraints are put in place.

This serves as the main motivation for this research, as the book on super-hypersonic vehicle design is far from written; with large disparities between current concepts, and only a handful of short term success stories among a plethora of long term failures. The lack of a go to design for such vehicles in is an area to be exploited, and given the need to maximise performance across all disciplines in order to make HST and RLV designs viable business propositions, conceptual design phases must become more exploratory and automated, leaving behind the typical iterative design cycle. Furthermore, to develop and test such vehicles is a high cost, high risk venture, with enormous engineering effort required to obtain the necessary performance while maintaining cost effective efficiency. Thus, utilising available computational power to explore novel designs, and identify design characteristics with promising performance should be a critical element of the conceptual process.

1.1.3 Application of Optimisation Methods to Aerospace Design

With advancements in computer technology and optimisation methods, it is no longer reasonable or time effective to create an initial design based on engineering experience, and iterate once multi-disciplinary analysis has been carried out. This is particularly pertinent in the case of super-hypersonic vehicles, where reliable designs do not yet exist. Instead, a design optimisation approach should be employed, in which analysis can be conducted on a large scope vehicle configurations, limited only by the versatility and accuracy of the performance models employed.

Optimisation continues to grow in popularity across all forms of engineering, with potential applications at all stages of the design process due to the wide variety of techniques. These are chosen and exploited based on the problem at hand, taking into account factors such as fidelity of analysis and computational resources; limitations placed on the design space due to external factors; and the existence of an initial design, whether it is to be improved upon directly or used to benchmark designs output from a more exploratory search.

In the field of aerospace engineering, much attention is placed on high-fidelity analysis methods, particularly in terms of aerodynamics. Due to the high computational cost of state-of-the-art computational fluid dynamics (CFD) solvers, efficiency is required in the optimisation algo-

rithm, leaving gradient based methods as the only reasonable choice. These optimisers require drastically fewer function evaluations than their gradient-free counterparts, however an additional requirement is necessary to employ such methods effectively: a close to optimal starting point. Often then, pre-existing configurations are used as starting points, with design spaces built around their parameters. As a result, the final output may exhibit an improvement over the initial design, but the likelihood of it being globally optimal for the given optimisation problem is low. This is due to the dependency of the design space on the existing configuration; ensuring that the final result does not deviate too far from the original, while also reducing the number of function evaluations.

Air/spacecraft design optimisation; whether it be that of an aerofoil, wing, or full vehicle, therefore generally focuses on local optimisation: where an initial design is given, and the goal is to improve upon it. In such approaches, the optimisation variables often only capture specific attributes of the design, rather than building it from a fully parametric definition. The main reason for this is due to the fact that while the initial design is to be bettered, it is desirable to maintain similarities between it and the optimised configuration. Furthermore, certain aspects of the design may be fixed due to circumstances out-with that of the optimisers control, for example fuselage dimensions defined by payload bay sizing. In many cases however, optimisation state variables (those which remain constant) are chosen rather arbitrarily to enforce this similarity, and provide a performance enhancement while preserving much of the original design. This desire to preserve characteristics of the starting point lends itself to local, or more specifically, gradient-based optimisation techniques; in which variables are perturbed with respect to an original design, and the resulting performance gradients used to determine iterative design improvements.

As previously mentioned, the underlying assumption in the selection of any initial design is that it is close to optimal. However, without knowledge of the design process that led to this configuration, or if it has not been designed with the input of some prior large design space optimisation work, its optimality cannot be assumed. Thus, in terms of optimisation, many of these starting points can also be considered arbitrary; seeking to improve upon a design simply because it is pre-existing. Of course, robust design processes are employed by industry with many iterations and analyses of varying components that make up a full vehicle. However, even in the best of cases, time and budgetary constraints require these avenues to be prioritised, leaving many potential design choices somewhat or completely unexplored.

While it is important to demonstrate that optimisation frameworks are capable of improving existing designs without major modification, this only scratches the surface of what these algorithms are capable of. By utilising optimisation techniques earlier in the design process, high

performance configurations can be output, with characteristics that may not appear obvious to experienced engineers. From here, engineers can select designs or design characteristics to be verified under higher fidelity analysis, which can then be employed as starting points for localised optimisation, rather than relying on an existing design. Implementation of such a methodology would ensure that the original design is close to optimal for the given problem, and could easily employ existing designs for comparative or constraint purposes.

It is inevitable that the future will be dominated by optimisation and machine learning processes, as these have quickly become the default methods for data analysis of increasingly larger scales. Such algorithms allow correlations to be made between variable parameters within a given dataset or system, and its desired outputs; something that human analysts struggle to determine effectively. In particular, engineers are likely to air on the side of caution, and choose components or designs that are known to work. Of course there may be any number of good reasons to do this, but there is also a degree of conservatism; with companies not willing to spend excessive amounts on research and development of new concepts. However, with maturation of these numerical methods, much of these processes can be automated, allowing cheap exploration of various design choices. This may require a large bank of knowledge and models in order to be employed effectively, but correct implementation ensures that, at a minimum, an understanding of the performance impacts arising from the analysed design choices will be gained.

Of course, as with any optimisation, what is deemed optimal mathematically will differ from that of the real world. Particularly in the field of aerospace vehicle design; disciplines modelled, to what level of fidelity, and flight states considered, will have a large impact on the resulting mathematical model of a much more complex real-world application. These issues present themselves regardless of the optimisation procedure, since even state-of-the-art analysis of a single design across a small portion of the flight envelope requires vast computational resources. Importantly then, it should not be overstated how grand a challenge it is to develop real-world feasible designs from an arbitrary starting point.

1.1.4 Brief Review of Prior Research

Previous research on the topic varies drastically in terms of disciplines considered, fidelity of analysis, optimisation method, objective functions and design variables. An optimisation comprising of a single disciplinary analysis method may result in ideal cost function values, but resulting designs will most likely be infeasible with respect to other disciplines not modelled. Increasing the number of accurately analysed disciplines leads to further constraints placed on the optimiser, resulting in higher cost functions, but ultimately more feasible designs. Nonetheless, no such analysis to date can be considered all-encompassing, and engineering experience remains a dominant factor in vehicle design. Analysis fidelity and number of design variables,

constraints and objectives should be considered together based on computational resources and research scope. High-fidelity models may provide state-of-the-art results, but can also drastically increase computational cost, and decrease design space variability. Couple this with the knowledge that increasing number of design variables has a large effect on the iterations required for convergence, and it is clear that limitations must be enforced from the outset.

If an existing configuration is sought to be optimised, it is reasonable to assume that the design space will be limited compared to that of a random start point optimisation, as this ensures that the optimal design will not vary significantly from that of the original. A close to optimal starting point makes gradient based optimisation techniques coupled with high-fidelity analysis appealing, considering that a small design space may require relatively few iterations to converge. On top of this, in an optimisation where only small alterations may be made to the configuration throughout, it is essential that the physics are captured as accurately as possible. Furthermore, assuming a basis design that is close to optimal exists, such an optimisation may be aimed at a latter stage of the design process, where it is essential that computational resources are dedicated to state-of-the-art prediction methods.

In cases where a starting point is not available, perhaps in the early conceptual or design phases of a relatively new concept, global design optimisation techniques combined with lower fidelity analysis methods are well suited to the problem. Taking full advantage of the optimisation process requires a large design space in which novel, globally optimal configurations can be found that may not be obvious with the current scope of engineering experience. Utilising a population based optimisation algorithm with low-fidelity methods allows the broad search space to be explored at a relatively cheap computational cost. If multiple objective functions are sought to be minimised, this approach also allows the creation of Pareto fronts, in which a number of optimal solutions are found. During the early phases of a conceptual design process, these fronts allow interesting configurations or characteristics to be extracted from the results for further analysis, rather than a single best design resulting from single-objective optimisation.

The majority of research focuses primarily on aerodynamic optimisation. An early example of this is the work done by Landon et al. [13], in which the low-fidelity Supersonic/Hypersonic Arbitrary Body Program [14] was used as the prediction tool, and an initial geometry was optimised through the use of free-form deformation methods [15]. Chiba et al. [16] used evolutionary algorithms along with surrogate models to provide approximate analysis, before feeding the most promising configurations to a Navier-Stokes solver. Here the goal was to optimise a wing-body reusable launch vehicle for flight states around a Mach number of unity. Four objective functions were employed in this research, relating to aerodynamic properties at various flight states. A similar method was employed by Tatsukawa et al. [17], which utilised a genetic algorithm and

a Navier-Stokes solver throughout, in this instance to optimise a lifting-body configuration. As with many hypersonic aerodynamic optimisations, L/D was used as an objective, together with zero-lift drag and internal volume. Here a limited number of function evaluations was set, likely due to the computational requirements of the solver.

Zhang et al. [18] also used genetic algorithms to optimise hypersonic wings using local piston theory. Their method boasts low computational cost, however the force and moment inaccuracies when compared with higher fidelity solvers, along with piston theory being purely inviscid, highlight its limitations. Shen et al. [19] also used a genetic algorithm, along with a simple modified Newtonian method, in optimisation of the European EXPERT re-entry vehicle [20]. Again L/D was used as an objective function, alongside a reduction in optimum angle of attack. Another example of a modified Newtonian based optimisation comes from the work of Sheffer and Dulikravich [21], with drag being minimised for cone and hypersonic plane configurations.

Of course, many disciplines define a feasible hypersonic vehicle. With strong couplings at play, multi-disciplinary optimisation has gained increasing popularity over the years. Bowcutt [22] provides an early example of multi-disciplinary design optimisation (MDO) applied to hypersonic aircraft; combining aerodynamic, propulsive, and trim analysis. Here, a small number of design variables were used to define the vehicle, once more with the goal of maximising L/D. A further study conducted by Bowcutt [23] added trajectory analysis to the optimisation process, setting range as an objective function for a full flight path. A missile configuration was studied, operating between $M = 4.5 - 7$. Being a trajectory driven problem, stability and control also factored heavily in the overall framework. Utilising MDO in this analysis found a 46% increase in range over the initial baseline, showing the methods potential when applied successfully. It should be noted that the few decision variables in these works require many translations to produce full configurations. Generally, a continuous design space that defines the vehicle as physically as possible is desired, since it may lead to improved convergence.

Dirkx and Mooij [24] coupled local-inclination aerothermodynamic methods with trim analysis, for both capsule and winged vehicles. Various realistic constraints were enforced, such as stability characteristics; maximum structural loading; bounded control surface deflections and angle of attack; as well as maximum heating requirements, with three geometric based objective functions being employed. Wuilbercq et al. [25] used reduced-order models to compute aero-thermal, heat protection and propulsive characteristics, with the inclusion of mass estimation for reusable hybrid vehicles in both ascent and re-entry flight regimes. Deng et al. [26] coupled flight characteristics and geometric performance parameters within objective functions, utilising a surrogate model in conjunction with a CFD solver to improve upon an existing design. Recently, Di Giorgio et al. [27] introduced an aerothermodynamic design optimisation

framework for hypersonic vehicles. Aimed at high-speed passenger aircraft, engineering level prediction methods are utilised, with potential future extension to multi-fidelity and surrogate based techniques. Much of the reasoning behind creation of their framework is akin to the presently described research, and further demonstrates the necessity of developing these methods in relation to high-speed vehicles.

While some of these works incorporate global optimisation methods, they often do not employ global design spaces. Not only does this limit the usefulness of these versatile techniques, but it often predetermines the output configurations, with little scope to explore novel design characteristics. Of course, many limitations are imposed with good reason, to avoid vehicle characteristics that are known to be undesirable or infeasible. However, achieving this by restricting the design space should be avoided, as in many cases boundaries are derived arbitrarily or overcautiously. Furthermore, undesirable and infeasible characteristics are not often the result of a single design variable, and thus reducing the search space can eliminate many feasible designs in an attempt to remove some unwanted ones. Therefore, this work emphasises the importance of constraint handling within the design optimisation process, and uses such methods to ensure that only configurations conforming to specified design requirements can be seen as optimal.

The implementation of large design spaces requires flexibility in both the analysis methods and determination of optimality, ensuring performance is adequately quantified to distinguish between designs in terms of objective and constraint functions. Therefore, as with many of the previous works described here, this research seeks to conduct multi-disciplinary analysis on candidate designs throughout the optimisation process.

1.2 Aims & Objectives

The overall goal of this work is to produce globally optimal super-hypersonic vehicle designs with respect to multi-disciplinary analysis, from an arbitrary starting point. Considering the limited experience of successful high-speed applications, and the vast discrepancies between current concepts, it is clear that much work is required in all aspects of the design and testing phases. Here the intention is to develop a framework that quantifies performance common to all super-hypersonic vehicles, abstracting above the yet to be defined system level characteristics. Conducting large design space optimisation work with such performance analysis, including robust definitions of optimality and feasibility, should enable high-quality, novel designs to be output. The broad scope of this research requires a number of key modules to be developed, which are detailed below, along with anticipated applications:

- Implement fully parametric definitions of super-hypersonic vehicle components that allow for a high degree of control.
- Develop and validate rapid aerodynamic analysis tools, capable of quantifying performance for arbitrarily defined components and full vehicle configurations.
- Develop optimisation algorithms that can rapidly analyse highly non-linear cost function spaces in search of globally optimal solutions.
- Implement robust constraint handling techniques that quantify feasibility of individual components and full vehicles, allowing large design spaces to be searched effectively without producing false optima.
- Experiment with additional analysis modules and levels of fidelity not commonly used within global design optimisation, to further quantify component and vehicle performance and feasibility.
- Conduct multi-disciplinary global design optimisation on full super-hypersonic vehicles, demonstrating the overall capability of the developed framework.

Throughout the development of this framework, key focus was placed on modularity within the system as a whole. As previously discussed, no optimisation framework to date is all-encompassing in the aerospace field. Aerothermodynamics, structural mechanics, propulsive and control methods, mission profiling and many more factors play a role in determining the final configuration of such vehicles. Obtaining state-of-the-art accuracy for any one of these disciplines in an automated, global optimisation context remains far out with our grasp. Thus, it was realised in the early phases of this work, that developing a versatile system capable of handling any mixture of flight states, vehicle components, and analysis disciplines, was of the utmost importance. For such a system to remain useful as analysis methods develop and the

design space broadens, it must be able to implement and handle such changes with as little disruption to both the user and the end product as possible. Therefore, it is paramount that the presentation of this work highlights its overall modular approach, along with the various implemented methods that allow the system to provide a desirable end result, regardless of the nature of the specified problem. Of course, presently many problems cannot be dealt with using this framework. Clear lines will be drawn in terms of its limitations, along with its ability to be expanded upon in future research.

1.3 Thesis Outline

The remainder of this work is divided into the following chapters:

- Chapter 2 introduces the methods used for aerodynamic modelling, along with their implementation within the overall framework, and validation against experimental and CFD data found in literature.
- Chapter 3 introduces the optimisation techniques employed to tackle the large design space, highly constrained problems carried out in this research.
- Chapter 4 focusses on aerodynamic aerofoil optimisation, validating the generative techniques employed and providing an initial use case of the developed framework as a whole.
- Chapter 5 extends the low-fidelity aerofoil analysis to multi-fidelity optimisation, utilising surrogate models and open-source high-fidelity methods to enable efficient use of computationally expensive analysis, improving results while maintaining reliability.
- Chapter 6 introduces a low-fidelity wing-box structural model, allowing fluid structure interactions to be modelled in a multi-disciplinary optimisation, which is carried out in comparison to a pre-existing RLV design
- Chapter 7 focusses on global aerodynamic design optimisation of a fully parametric vehicle definition, enabling lifting-body and wing-body design spaces to be optimised with respect to robust feasibility constraints.

1.4 Related Work

Journal Articles

Cusick, A., Kontis, K., Nikbay, M. (TBC). Multidisciplinary Design Optimisation of Reusable Launch Vehicles, AIAA Journal of Spacecraft and Rockets (Under Review)

Cusick, A., Kontis, K., Nikbay, M. (TBC). Large Design Space Optimisation of Super-Hypersonic Vehicles (Under Review)

Cusick, A., Kontis, K., Nikbay, M. (TBC). Multi-Fidelity Global Design Optimisation of High-Speed Aerofoils (Work in Progress)

In Conference Proceedings

Cusick, A., Kontis, K., Nikbay, M. (2020). Multidisciplinary Design Optimization of Reusable Launch Vehicles, 23rd AIAA International Space Planes and Hypersonic Systems and Technologies, Montreal, Quebec, DOI: 10.2514/6.2020-2425

Cusick, A., Kontis, K., Nikbay, M. (2019). Aerostructural Design Optimization Of High Speed Wings For Reusable Launch Vehicles, 10th Ankara International Aerospace Conference, Ankara, Turkey, DOI: AIAC-2019-195

Cusick, A., Kontis, K. (2019). Creation of Design and Analysis Tools for Large Design Space Reusable Launch Vehicle Shape Optimization, AIAA Aviation 2019 Forum, Dallas, Texas, DOI: 10.2514/6.2019-2929

Chapter 2

Aerodynamic Modelling

2.1 Introduction

This chapter describes the methods used to analyse arbitrary vehicle components in terms of aerodynamics. A consortium of techniques have been employed, allowing accurate results to be obtained for different vehicle components, flow orientations, and flight speeds. A number of independent inviscid methods are used, with autonomous switching conditions implemented, enabling their use in tandem across arbitrary surfaces with varying flow properties. Viscous contribution is also calculated on a local scale, allowing full friction coefficient and temperature distributions to be output. The theoretical nature of these methods, and the required conditions allowing results to be calculated for arbitrary surfaces will be discussed. Inviscid and viscous prediction techniques are then validated independently, before utilising the analysis module as a whole to predict the aerodynamic characteristics of a full wing-body vehicle.

Finally, the implementation of these methods in an in-house context that allows arbitrary vehicles to be analysed is described.

2.2 Scope

At the conceptual or early design phases, flexibility and computational efficiency are desirable over high-fidelity accuracy. While computational fluid dynamics (CFD) methods are state-of-the-art in aerodynamic prediction, the time required to analyse the thousands of configurations generated in a typical large design space optimisation is unreasonable. On top of this, CFD is heavily reliant on discretisation with respect to convergence and therefore accuracy. Designing an algorithm that can provide sufficient discretisation not only of arbitrary vehicles, but of the complex flow domain surrounding them, is far beyond the scope of this work.

Low-fidelity methods can provide rapid results based on vehicle geometry and flow properties.

While their accuracy is not up to CFD standards, it is more than acceptable for initial design optimisations covering a diverse population. All aerodynamic analysis carried out in this work is therefore based on panel methods, meaning only the geometry to be analysed is required to be modelled, and not the surrounding flow domain. This is the primary advantage of such methods, as static results can be gathered knowing only the local orientations of a discretised geometry with respect to the freesteam flow, removing the need for iterative numerical solutions. The rapid computational time of these methods, and their flexibility in terms of input geometry, make them ideal for large exploratory design optimisation work.

2.3 Inviscid Methods

The low-fidelity inviscid aerodynamic methods implemented are dependent upon the relationship between geometry inclination and incoming flow direction. Beginning with the simplest case of x-axes alignment between flow, geometry, and therefore reference frame, as shown in figure 2.1, the freestream velocity direction is a unit vector aligned with the x-axis. To determine local surface inclination, the outward facing normal of each panel must be found. As panel methods logically assume forces are applied at a given panel's centre point, this is where the normal is calculated from. In two-dimensions, this is a simple case of finding the vector perpendicular to the line surface, and projecting it outward in the correct direction from the centre point. For three-dimensional bodies, vectors joining triangular corner points, or quadrilateral cross corner points, are required to compute surface normal. Taking the cross product of these vectors outputs the necessary vector, with care once again required to ensure it is pointing away from the surface.

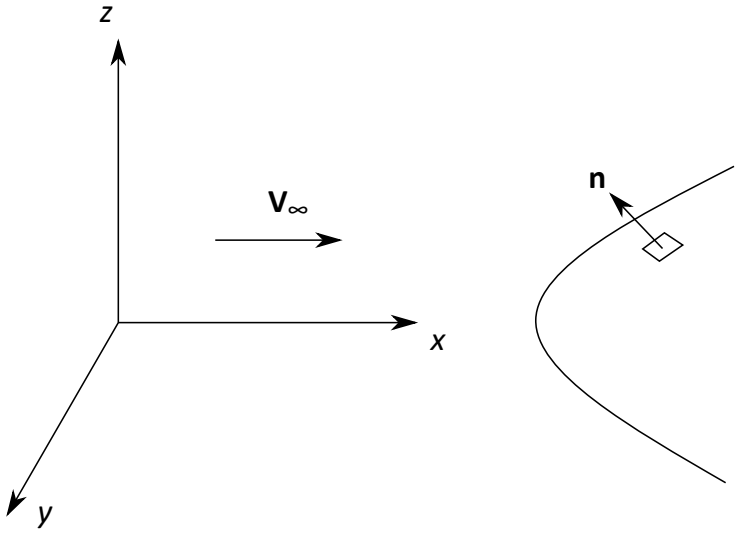


Figure 2.1: Incoming freestream velocity and local surface vectors

Of course, as solutions are highly dependent on this local flow orientation, discretisation that is largely representative of physical components remains of great importance. This will be

discussed further in proceeding chapters, however, it should be noted that the implementation of these methods is capable of handling any unstructured triangular or quadrilateral grid. Assuming an accurately discretised geometry, the relation between outward facing surface normal, $\mathbf{n}$, and the freestream flow direction, $\mathbf{V}_\infty$, is defined as:

$$\theta_i = \sin^{-1}\left(\hat{\mathbf{V}}_\infty \cdot \hat{\mathbf{n}}_i\right) \tag{2.1}$$

Where θ is the inclination angle, i defines the specific panel on the surface, and circumflex denoting both vectors are unit normals. In this sense, a positive angle means that the surface is inclined towards the flow, with a negative angle implying that it faces away from the flow. Note that this convention allows either full surface or velocity vector to be rotated in order to conduct analysis at non-zero angle of attack and sideslip combinations.

The inclination angle of a given surface to the flow is utilised across all of the methods in the proceeding section. Importantly, a distinction should be made between surfaces facing the flow (+ve θ), and those facing away from the flow (-ve θ). Positive inclination angles are those that can be "seen" by the flow, which is equivalent to the frontal area observed upstream in the freestream flow direction. The assumption is that at very high-speeds, the flow can only impact this frontal area, with any negative inclination surfaces having pressure equal to that of the freestream. These areas are therefore referred to as being in the shadow of the incoming flow.

At realistic continuum flight speeds, there is much more continuity between impact and shadow regions, and more blended techniques are required to obtain accurate pressure distributions across a given surface. Nonetheless, the distinction between impact and shadow regions is used across the super-hypersonic Mach range, with most panel methods being applicable in only one domain. The description of implemented methods has therefore been separated in the same fashion, with impact flow techniques being discussed first.

2.3.1 Impact Flow

Impact methods are employed for local aerodynamic prediction with positively inclined surfaces. Physically, these techniques attempt to model flow conditions aft of an impinging shockwave, which allow surface pressures to be determined. The combination of high speeds and arbitrary geometry means predictive methods for both attached and detached shockwaves must be implemented, to ensure a realistic flow regime is represented by the aerodynamic model.

Newtonian

Newtonian flow is derived from the assumption that a particle in hypersonic flow will impinge on a surface and immediately change its direction to move parallel with the surface. From this

assumption, a striking particle will lose all of its momentum normal to the surface, while maintaining a constant tangential velocity as it travels downstream. Thus, the pressure coefficient on a body inclined to the freestream flow is computed as follows:

$$C_p = 2\sin^2\theta \tag{2.2}$$

The physical concept of a given particle impinging on a surface, losing its normal velocity and travelling downstream without loss of tangential velocity, is logical at high Mach numbers; where the resulting shockwaves remain close to the surface and thus, have a similar inclination to the freestream flow. This results in the Newtonian flow model increasing in accuracy as $M \to \infty$. However, at reasonable hypersonic speeds this will not be the case, and since equation (2.2) has no dependency on Mach number, a more accurate method is required.

Lees [28] introduced such a method, now well known as modified Newtonian, which looks to correct standard Newtonian theory based on physical properties of the flow. Here, the coefficient in equation (2.2) is replaced by the maximum value of the pressure coefficient, defined as the C_p at the stagnation point behind a normal shock, as shown below:

$$C_p = C_{p_{max}}\sin^2\theta \tag{2.3}$$

$$C_{p_{max}} = \frac{2}{\gamma M_\infty^2}\left\{\left[\frac{(\gamma+1)^2 M_\infty^2}{4\gamma M_\infty^2 - 2(\gamma-1)}\right]^{\frac{\gamma}{\gamma-1}}\left[\frac{1-\gamma+2\gamma M_\infty^2}{\gamma+1}\right]-1\right\} \tag{2.4}$$

Note as $M \to \infty$ and $\gamma \to 1$, $C_{p_{max}} \to 2$, and thus the original Newtonian formulation is reproduced. This allows for greater accuracy at Mach numbers below high hypersonic, which would otherwise assume equal pressure coefficient at a given inclination angle, regardless of Mach number. Even with this modification, the assumption of Newtonian flow is generally only used for high hypersonic regimes, or at regions with high surface inclination to the flow, where a detached shockwave is present. At lower super-hypersonic flight speeds, and in particular those surfaces with attached shockwave solutions, a number of methods can be utilised to predict shock angle, and therefore surface flow properties.

Tangent Wedge/Cone

While the modified Newtonian method is applicable in high Mach number or high inclination cases, improved accuracy is needed at the lower end of the super-hypersonic scale. One such approach is the tangent wedge or tangent cone method. Although not derived from physical fluid mechanics, these techniques have been demonstrated to provide accurate results for various applications and Mach numbers [29]. Here, drawing a line downstream in the direction of the local surface deflection θ, consistent with the definition in equation (2.1) and shown in figure 2.2,

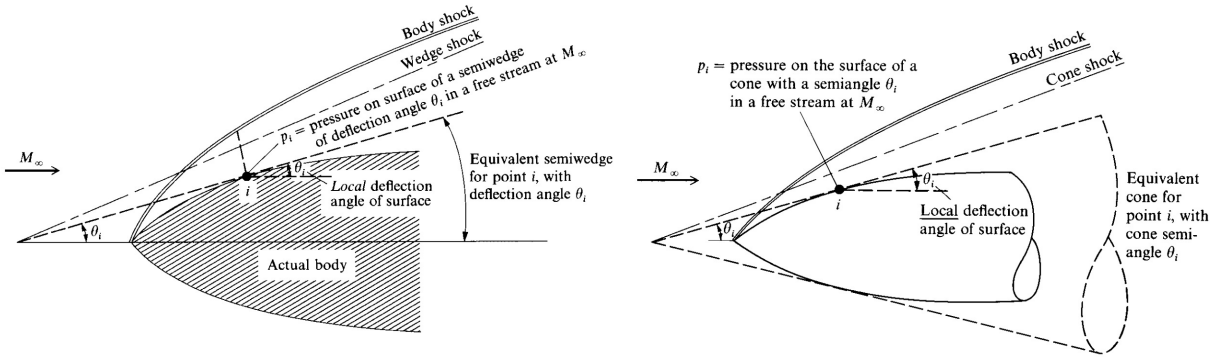


Figure 2.2: 2D tangent wedge (left) and 3D tangent cone (right) definitions [29]

represents the assumed wedge made between the body and freestream. This deflection angle can therefore be used to compute the equivalent shockwave that would exist having turned the flow by θ, allowing properties on the surface to be derived from the exact oblique shock relations. Equation (2.5) relates incoming flow Mach number and deflection angle with the resulting shock angle, commonly referred to as the θ-β-M relation:

$$tan\theta = 2\cot\beta \left[\frac{M_1^2 \sin^2\beta - 1}{M_1^2(\gamma + \cos 2\beta) + 2} \right] \tag{2.5}$$

Where β is the attached shockwave angle and γ is the heat capacity ratio of air. In order to solve this equation, a lookup table or numerical method must be employed, as discussed is section 2.7. Once the equivalent shockwave angle has been found, the flow properties aft of the shockwave, and therefore on the local surface, are calculated via:

$$\frac{p_2}{p_\infty} = 1 + \frac{2\gamma}{\gamma+1}\left(K_\beta^2 - 1\right) \tag{2.6}$$

$$\frac{\rho_2}{\rho_\infty} = \frac{(\gamma+1)K_\beta^2}{(\gamma-1)K_\beta^2 + 2} \tag{2.7}$$

$$\frac{T_2}{T_\infty} = \frac{p_2}{p_\infty}\frac{\rho_\infty}{\rho_2} \tag{2.8}$$

$$M_2 = \frac{1}{\sin(\beta-\theta)}\sqrt{\frac{1 + \frac{\gamma-1}{2}K_\beta^2}{\gamma K_\beta^2 - \frac{\gamma-1}{2}}} \tag{2.9}$$

$$K_\beta = M_\infty \sin\beta \tag{2.10}$$

Similarly for three dimensional flow, The tangent cone method is employed. Here, the two dimensional shock β is replaced with its 3D equivalent, τ.

$$\sin\tau = \sin\theta\sqrt{\frac{\gamma+1}{2}+\frac{1}{K_\theta^2}} \tag{2.11}$$

$$K_\theta = M_\infty \sin\theta \tag{2.12}$$

This allows downstream Mach number to be determined from pre-processed lookup tables or numerical methods, and pressure coefficient from equations provided by Rasmussen [30]:

$$\frac{C_{p_2}}{\theta^2} = 1+\frac{(\gamma+1)K_\theta^2+2}{(\gamma-1)K_\theta^2+2}\ln\left(\frac{\gamma+1}{2}+\frac{1}{K_\theta^2}\right) \tag{2.13}$$

Of course in either case, the combination of local surface inclination and freestream Mach number must allow an attached shockwave to be present, otherwise they are not applicable. In such cases, surface flow conditions are automatically solved using the modified Newtonian approach.

Shock-Expansion

The shock-expansion method provides a more robust, at least in terms of theory, attached shockwave method for calculating surface properties. In contrast to the tangent wedge/cone technique, only the initial surface causes a shockwave as a result of turning the freestream flow, which is once again calculated through the oblique shock relations in equations (2.5)-(2.9). These initial properties aft of the impinging shock are used as the starting point for a series of expansion waves (assuming the surface is convex), defined by the Prandtl-Meyer relation, $\nu(M)$:

$$\theta_{i+1}-\theta_i = \nu(M_{i+1})-\nu(M_i) \tag{2.14}$$

Where the Prandtl-Meyer function for a given Mach number at panel i, which is increasing as the flow travels downstream, is determined from:

$$\nu(M_i) \equiv \sqrt{\frac{\gamma+1}{\gamma-1}}\,tan^{-1}\sqrt{\frac{\gamma-1}{\gamma+1}(M_i^2-1)} - tan^{-1}\sqrt{M_i^2-1} \tag{2.15}$$

Since both surface inclinations and upstream Mach number, M_i, will always be known, equation (2.14) can be solved for downstream Mach number, M_{i+1}. With the expansion process being isentropic, downstream flow properties can be computed from the isentropic flow relations:

$$\frac{T_{i+1}}{T_i} = \left[\frac{1 + \left(\frac{\gamma-1}{2}\right) M_i}{1 + \left(\frac{\gamma-1}{2}\right) M_{i+1}} \right] \tag{2.16}$$

$$\frac{p_{i+1}}{p_i} = \left[\frac{T_{i+1}}{T_i} \right]^{\frac{\gamma}{\gamma-1}} \tag{2.17}$$

$$\frac{\rho_{i+1}}{\rho_i} = \left[\frac{T_{i+1}}{T_i} \right]^{\frac{1}{\gamma-1}} \tag{2.18}$$

This process can then be repeated by incrementing the counter, and continuing to solve for Mach number downstream until an edge is reached. The only unknown in this process is the direction of downstream relative to a specific point on the surface, which for arbitrary three-dimensional cases is not trivial. Determination of this direction will be discussed further in section 2.5.

Shielding

An additional note in terms of impact flow is shielding; where a surface with positive inclination angle is not directly seen by the flow due to a surface upstream. This is particularly relevant at hypersonic speeds, where Newtonian theory more accurately depicts the physics, meaning the flow may not reattach to the surface once it turns into the shadow region. This retains the original definition of impact flow, since surfaces that are blocked by others upstream cannot be seen by an observer standing in the direction of the incoming flow, regardless of inclination angle.

To correct for this, flow properties of shielded panels with positive inclination towards the flow must be solved with shadow methods. In order to determine shielded surfaces, initial shadow panels create a convex hull in the xz-plane around the geometry. Any surfaces within this hull and downstream of line joining the closest containing points, found through panel centre point and normal direction, are therefore shielded. Note that this method is not exact, and assumes relatively well behaved geometries (ie. those with limited concavity). Nonetheless, convex geometries are generally expected in an external aerodynamics context, particularly in the chordwise direction. Thus, in the majority of cases, the incorporation of a shielding routine will not redefine any impact panels, since any concavity within shadow regions is generally not enough to turn the surface back towards the flow. An exception to this is vertical tail surfaces, which may be shielded by the fuselage at high angles of attack. This presents a more general problem outwith the scope of this work, since the flow properties in front of the tail leading edge cannot be determined. Therefore, panel shielding in this work is employed in a component build-up approach, meaning only panels of the same surface can shield one another. This guarantees shielded panels will always have flow inputs from panels upstream.

2.3.2 Shadow Flow

Surfaces facing away from the incoming flow are in the shadow region. These areas are not therefore directly impacted by the freestream. Instead, their flow properties and resulting surface pressures are determined from the turning angle of the upstream flow, which may come from an impacted panel or the freestream itself, depending on surface inclination of the leading edge. Importantly then, since impacted panel flow properties may be inputs to shadow surfaces further downstream, all impact calculations must be carried out before shadow properties are determined.

Newtonian

Following on from the Newtonian assumption that flow continues along its current direction until impacting a surface, it is clear that particles cannot strike any surface that is not directly impacted by the incoming flow. Therefore, it is assumed that surfaces in the shadow region are at freestream conditions, leading to a constant pressure coefficient, $C_p = 0$. Once again this assumption may hold at high hypersonic flight regimes, but becomes inaccurate toward lower Mach numbers.

Prandtl-Meyer

Even at hypersonic Mach numbers in continuum regimes, some flow will remain attached to the body, contrary to the assumptions made by Newtonian flow. To provide a more realistic pressure coefficient at such flight conditions, Prandtl-Meyer expansion waves are once again used assuming a convex surface. Because of this convexity requirement, the shock-expansion method has been implemented in a manner that can be used in both impact and shadow regions, which will be discussed further in section 2.7. This means that upstream flow properties can be computed using either Prandtl-Meyer or oblique shock relations, based on the sign of inclination angle change across surfaces.

Of course, limits exist in terms of turning angle for both oblique shock and Prandtl-Meyer relations. In oblique shock cases this is handled as discussed in section 2.3.1, assuming a detached shock and therefore Newtonian flow. In the case of high Prandtl-Meyer turning angle, as defined by equation (2.19), the flow can no longer remain attached to the body. Such regions could therefore be treated in a Newtonian flow fashion, however more accurate methods have been developed for such surfaces, known as base regions.

$$\theta_{max} = \nu_{max} - \nu(M_1) \tag{2.19}$$

$$\nu_{max} = \frac{\pi}{2}\left(\sqrt{\frac{\gamma+1}{\gamma-1}} - 1\right) \tag{2.20}$$

Base Pressure

Specific relations for pressure coefficient have been developed for base surfaces, which show an improvement over assuming Newtonian conditions. The most widely used expression is Gaubeaud's formula, which determines surface pressure coefficient from freestream Mach number and ratio of specific heats:

$$C_p = \frac{2}{\gamma M_\infty^2}\left\{\left(\frac{2}{\gamma+1}\right)^{1.4}\left(\frac{1}{M_\infty}\right)^{2.8}\left[\frac{2\gamma M_\infty^2 - (\gamma-1)}{\gamma+1}\right] - 1\right\} \tag{2.21}$$

This method is invoked for surfaces which have a high inclination away from the freestream flow, and/or turn the flow beyond the maximum allowable angle, defined by equation (2.19).

2.4 Viscous Methods

The previously described methods are idealised, removing the effects of viscosity within the flow domain, and specifically the boundary layer surrounding the analysed geometry. In terms of super-hypersonic flows however, the contribution of viscous effects is of paramount importance, particularly in terms of surface temperature. Aerodynamic heating is one of the main concerns in designing a high-speed vehicle, as it will encounter regions of extremely high dynamic pressure, causing a substantial temperature gradient to radiate through the boundary layer to the surface, primarily at the leading edges.

Nonetheless, characterising the boundary layer and accurately predicting the resulting temperature and shear stresses felt on the surface is a highly complex task; requiring hands-on flow domain, boundary layer, and geometry discretisation within a CFD context. As discussed previously, this level of detail is not feasible for a large design space optimisation, necessitating the implementation of more approximate methods.

Once again, flow property and force determination on a local panel basis is preferred over a reduction method based on overall geometric parameters. This means both inviscid and viscous contributions are calculated locally on the same grid, allowing both to be combined and integrated to give overall component and vehicle aerodynamic properties.

2.4.1 Reference Temperature Method

Derived from incompressible flow theory, reference temperature methods at super-hypersonic speeds use compressible corrections to determine an indicative temperature within the boundary layer. This temperature is then used to compute the remaining flow properties inside the boundary layer, which in turn are used to calculate heat transfer to the surface, along with friction force contributions.

The classical reference temperature method was derived by Eckert [31], which utilised the following equation:

$$\frac{T^*}{T_e} = 1 + 0.032M_e^2 + 0.58\left(\frac{T_w}{T_e} - 1\right) \tag{2.22}$$

Where T^* is the reference temperature, T_w is the wall wall temperatures, and T_e, M_e are the temperature and Mach number at the boundary layer edge, equivalent to the properties output by the previously described inviscid methods. Wall temperature, which is an unknown to be calculated, is initially set using a cold wall assumption.

Implemented within the framework is a recent improvement to the original Eckert method, in the form of the Meador-Smart relations [32], which differentiates between laminar and turbulent boundary layers.

Laminar:

$$\frac{T^*}{T_e} = 0.45 + 0.55\frac{T_w}{T_e} + 0.16r\left(\frac{\gamma-1}{2}\right)M_e^2,\ r = \sqrt{Pr^*} \tag{2.23}$$

Turbulent:

$$\frac{T^*}{T_e} = 0.5\left(1 + \frac{T_w}{T_e}\right) + 0.16r\left(\frac{\gamma-1}{2}\right)M_e^2,\ r = Pr^{*\frac{1}{3}} \tag{2.24}$$

Where r is the recovery factor, based on reference Prandtl number, Pr^*. The determination of laminar or turbulent boundary layer is dependent on local Reynold's number, which is yet to be determined. Therefore in the initial instance, panels use (2.24). In proceeding iterations, an estimation of transition Reynold's number is made following the advice of Dicristina [33].

Using the reference temperature, along with boundary layer edge properties calculated from

the previously described inviscid methods, reference density ρ^*, and dynamic viscosity, μ^* are computed from the ideal gas relation and Sutherland's law respectively. The Reynold's number is then calculated from (2.25), where V_e is the boundary layer edge velocity, and x is the distance downstream, defined from the leading edge to the centre of each panel. The local skin friction, c_f^* is then derived from the Schultz-Grunow relation in equation (2.26).

$$Re^* = \frac{\rho^* V_e x}{\mu^*} \tag{2.25}$$

$$c_f^* = \frac{0.37}{log_{10}(Re^*)^{2.584}} \tag{2.26}$$

Since this approximate method is based on two-dimensional flat plate flow, it is important that special treatment is given to conical surfaces, as viscous forces are increased due to decreased shock effects and therefore thinner boundary layers in three dimensions. Once again specific treatment varies depending on the flow regime, with the constant Mangler fraction being applied in laminar flow, and van Driest's turbulent cone rule utilised otherwise [34].

In order to determine aerodynamic heating, the Stanton number C_H can be found through the Colburn-Reynold's analogy in equation (2.27), and the net heat transfer at the surface, q_w, derived by equation (2.28):

$$C_H = \frac{c_f^*}{2(Pr^*)^{\frac{2}{3}}} \tag{2.27}$$

$$q_w = \rho_e V_e C_H c_p (T_{aw} - T_w) \tag{2.28}$$

$$T_{aw} = r(T_0 + T_e) + T_e \tag{2.29}$$

Where c_p is the ratio of the heat capacity in air at constant pressure, and T_{aw} is the adiabatic wall temperature. Surface temperature can then be obtained via Boltzmann's law (2.30), and the process repeated until a converged temperature distribution is output across the entire surface.

$$T_w = \left(\frac{q_w}{\sigma\varepsilon} + T_\infty^4\right)^{\frac{1}{4}} \tag{2.30}$$

2.5 Surface Flow Direction

As illuded to the preceding sections, one of the key requirements in determining both inviscid and viscous effects is the determination of streamlines. Necessary for both shock-expansion and reference temperature methods, two approaches have been experimented with in this research.

The first uses a strip theory approach similar to that implemented by Jazra and Smart [35], with the second employing a streamline integration technique. A component build-up method is used, which is particularly necessary in the utilisation of streamlines, as large discontinuities across components may result in inaccurate surface velocity interpolation. The goal of these methods is to determine strips of panels, normally beginning at the leading edges, where flow will run until it either detaches, or an edge is reached. This determines the flow property inputs to panels utilising shock-expansion techniques, and running or characteristic lengths for local Reynold's number computation. The latter is therefore determined either by cumulative distance from leading edge to given panel centre point in the strip theory approach, or total distance travelled by a streamline at exit point of the panel in the integrated method.

2.5.1 Streamline Integration

Streamline tracing attempts to produce a reasonable approximation of flow velocity over a given surface. Given an inviscid flow field over the entirety of a given surface, streamlines are traced from the leading edge downstream until an edge or stagnation point is reached. This process is repeated until it is known for every panel, which panel or edge it expands or contracts into. To ensure that the streamlines remain constrained to the surface, tracing is performed in two-dimensions, with the third being defined by the corresponding value that lies on the panel plane.

Since inviscid inclination methods do not provide information on the direction of velocity on a panel's surface, a simple yet widely used assumption is employed [36, 37]:

$$\mathbf{V}_{surf} = \hat{\mathbf{n}} \cdot \mathbf{V}_{\infty} \cdot \hat{\mathbf{n}} \tag{2.31}$$

Where $\mathbf{V}_{surf}$ is the tangency velocity vector on the surface, and the remaining terms are consistent with equation 2.1. Note the earlier caveat of a "reasonable approximation" of surface velocity was made due to this simplified approach which, as it is pre-requisite to the reference temperature method in section 2.4, does not account for boundary layer effects.

This surface velocity vector is calculated at the centre of each panel, and thus a method to produce a velocity field encompassing all edges of the surface is required. An accurate representation of the geometry to be analysed will provide a large amount of panels and therefore data points at which the surface velocity has been calculated. Therefore, linearly inter/extrapolating out to panel edges should maintain a well defined velocity field. However, this is limited by surface definition, with discontinuities in geometry manifesting themselves in discontinuous velocity fields.

Once the flow field has been defined, a fourth order runge-kutta method is used to trace stream-

lines, with velocity being interpolated within individual panels. Tracing continues bound to the current panel surface until an edge is reached, where the new panel is determined and the process repeats. To ensure all panels are planar, a triangular surface is required, meaning that quadrilateral definitions will be automatically subdivided. In this form, barycentric interpolation can be utilised, allowing an accurate surface velocity to be calculated anywhere within a given panel.

2.5.2 Strip Theory

Strip theory allows viscous effects to be computed without streamline tracing, which decreases computational load, and eliminates the need to ensure smooth geometries are input. To implement however, a regular quadrilateral grid is required, which will be discussed further in section 2.7. Such a grid defines strips of panels from leading to trailing edge, which therefore infers surface streamlines. This is a reasonable assumption for well defined geometries and low angle of attack. Of course, a certain degree of accuracy may be lost, especially at high inclination angles.

2.5.3 Discussion

Although streamline tracing is the preferred panel method approach, issues persist in applying it arbitrarily for the large design spaces sought in this research. Lack of continuity in geometry definition will lead to discontinuities in the velocity field. Guaranteeing continuous, well behaved surfaces requires a large development effort, as well as increased computational load due to increased number of required panels. Such methods are also limited by the simplified definition of surface velocity field, which may negate any potential benefits.

Furthermore, the relative gain of employing streamline tracing methods is limited within the scope of this work. In terms of viscous effects, rough approximations are an accepted limitation in the conceptual phase, with inviscid force contributions driving performance quantification. Many previous works employ empirical relations based on vehicle characteristics to approximate viscous effects, thus making either streamline or strip local methods a relative improvement. Importantly, determination of viscous effects at leading edges, where maximum temperature is normally found, is relatively insensitive to choice of method, since streamlines have not yet propagated downstream.

The main difference arising from the choice of streamline method therefore stems from disparity in downstream shock-expansion direction, since it affects inviscid aerodynamic computations. Nonetheless, avoiding the use of such a method on windward surfaces localises this disparity to shadow regions, which have a much smaller contribution to overall vehicle performance.

Thus, this research will utilise the strip theory approach going forward, as the relative accuracy gains are small compared to the required implementation and validation efforts. Future work should look to improve upon geometry generation methods; to consistently produce smooth, continuous surfaces, at which point streamline methods can be properly implemented. An intermediate approach to be considered is tracing streamlines from each panel centre point until it intersects with one of its edges, therefore determining each panels successor, rather than integrating full surface streamlines from leading to trailing edges.

2.6 Validation

Before the aerodynamic module can be integrated into an optimisation framework, it is critical that it has been shown to produce accurate results, and that any limitations are known. To do this, comparative analysis should be carried out on existing data found in literature. Here, viscous methods are first validated independently, using a variety of geometries found in previous research, to demonstrate their applicability in terms of thermal prediction. Following this, both inviscid and viscous methods are combined to produce results for a number of independent components and a wing-body vehicle, showing the overall module's ability to compute aerodynamic characteristics of full configurations.

2.6.1 Viscous Effects

To demonstrate that the viscous methods described previously have been implemented correctly, and are fit for the purpose of predicting thermal properties on the surface of a given body, a number of CFD test cases found in literature have been replicated. Primarily from the works of Higgins [40] and McNamara et al. [41], a variety of configurations and freestream flow properties demonstrate the versatility and reliability required for use in a large design space framework.

Flat plate

Beginning from a simple flat plate case, the total rate of heat transfer per unit width is calculated at a variety of hypersonic Mach numbers, for both laminar and turbulent boundary layers, and compared to RANS data gathered by Higgins [40]. The total heat transfer rate is calculated from the local rate and streamwise distance from leading edge as follows:

$$\dot{Q}(x) = \int_0^x \dot{q}(x)dx \tag{2.32}$$

The plate is tested at Mach numbers in the range $5 \leq M \leq 10$, with the remaining flow properties analogous to an altitude of 35km, and a constant wall temperature of 300K is specified. Note that for a flat plate parallel to the freestream flow, the boundary layer edge Mach number

is identical to that of the freestream.

The comparison is plotted in figure 2.3. For both laminar and turbulent cases, the low-fidelity method provides a close approximation of the CFD results, particularly at lower Mach numbers. A slight under-prediction is observed throughout the laminar Mach range, whereas the turbulent case tends to somewhat over-predict the total heating rate until Mach 9, where it drops below the high-fidelity case. This under-prediction at high Mach numbers is likely due to the gas dissociation experienced at such high speeds, which is not encapsulated by the low-fidelity method. Regardless, the root mean squared percentage errors for the laminar and turbulent cases are 7.38% and 2.46% respectively, which is more than acceptable for a low-fidelity approximation.

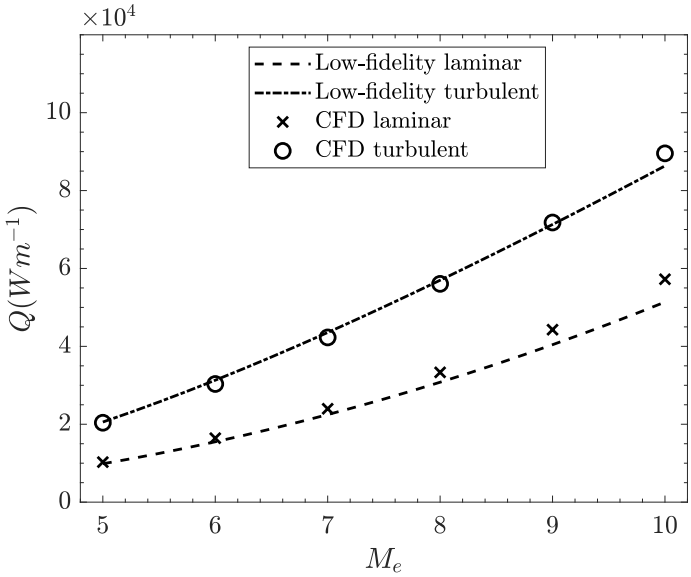


Figure 2.3: Flat plate total heat transfer rates for various Mach numbers at 35km

Cone and Wedge

Heating rates at a local level are also provided by Higgins [40] for wedge and cone shapes at Mach 8 and altitude 35km, with once again a prescribed wall temperature of 300K. Both geometries are defined by 10°semi-angles, with their chord parallel to the freestream flow. Figure 2.4 shows the streamwise local heat transfer rates for the wedge and cone respectively. The results are almost identical to a similar comparison produced by Higgins, showing that the low-fidelity methods have been implemented correctly for three-dimensional cases. The local heating rate in both laminar and turbulent cases is under-predicted by the method at the leading edge, before showing good agreement at $x \geq 0.4$ for the laminar case, and an over-prediction of the heating rate at $x \geq 0.15$ for a turbulent boundary layer.

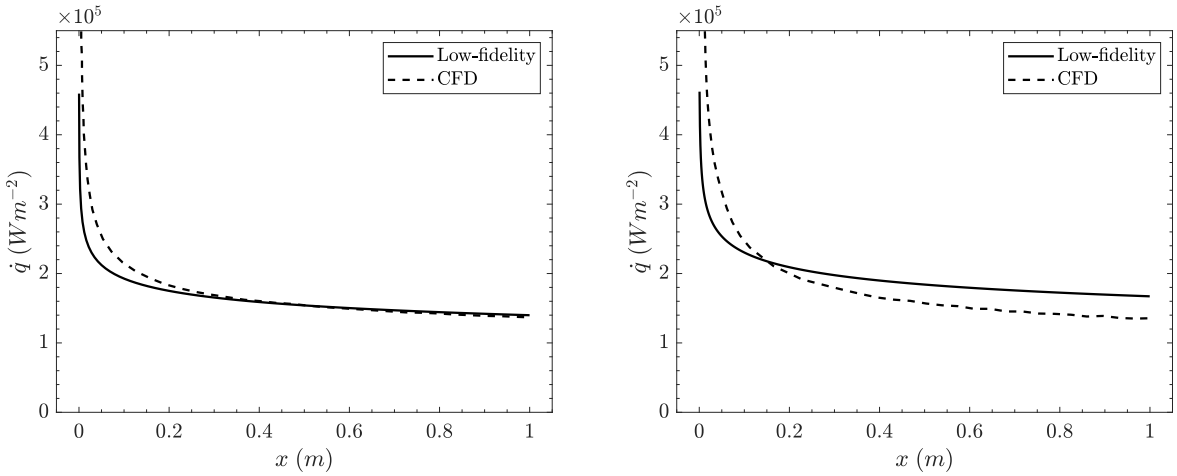


Figure 2.4: Local heat transfer rates at Mach 8, 35km for 10° wedge (a) and cone (b)

Swept Wing

The research carried out by McNamara [42] provides a swept wing test case, in which RANS simulations were carried out using the CFD package CFL3D to predict surface temperature. Importantly then, this allows the iterative section of the low-fidelity method to be validated, producing a converged surface temperature across the entire wing, rather than a prescribed input. The swept diamond wing geometry is shown in figure 2.5, with a comparison between the high and low fidelity methods shown in figure 2.6. Direct comparison for specific locations are also shown in table 2.1.

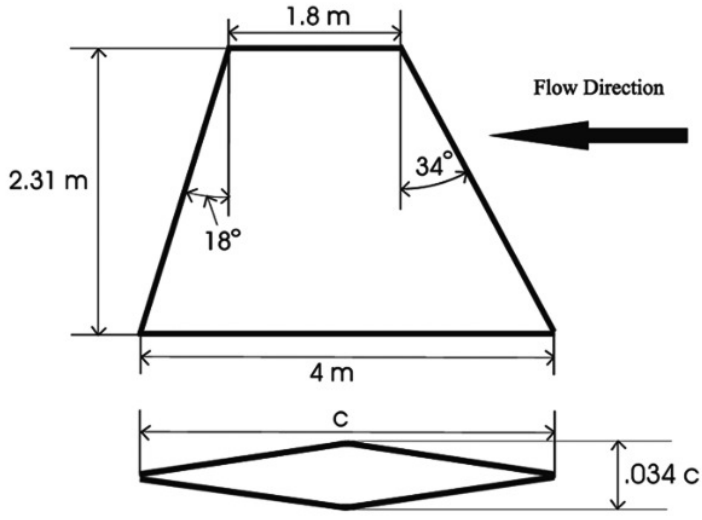


Figure 2.5: Schematic of diamond shape wing used for surface temperature validation [41]

As expected from a low-fidelity panel based method, the three-dimensional effects are not fully captured, and the temperature distribution shows a higher degree of similarity over the wingspan

than the high-fidelity simulation. Similar results have been observed by Wan et al. [43], who also utilised this data to validate their low-fidelity viscous model. However, as shown in table 2.1, only a small error is present over the wing chord. Of particular note is the smallest error occurring at the leading edge, which is where most accuracy is desired, due to this being the area in which temperature based constraints are most likely to be violated in a hypersonic optimisation.

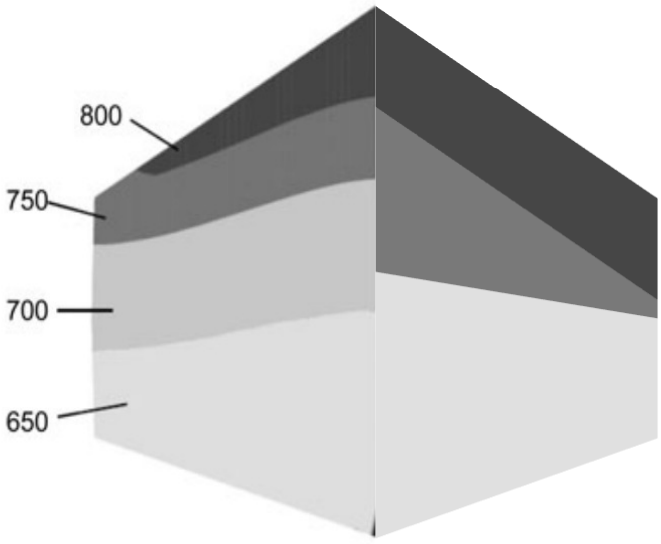


Figure 2.6: Contoured temperature distribution over diamond wing for RANS (left) and low-fidelity method (right)

Table 2.1: Wall temperature comparison for various streamwise locations at 75% span

Location (x/c)	RANS (K)	Low-fidelity (K)	Error (%)
0	861.47	841.26	-2.35
0.5	743.11	717.86	-3.40
1	696.74	664.95	-4.56

2.6.2 Force Coefficients

Both inviscid and viscous aerodynamic methods are combined and used in comparison with experimental and CFD data for body, lifting surface, and full wing-body geometries. This is intended to show the methods applicability to both individual components and overall configurations.

Cylindrical Bodies

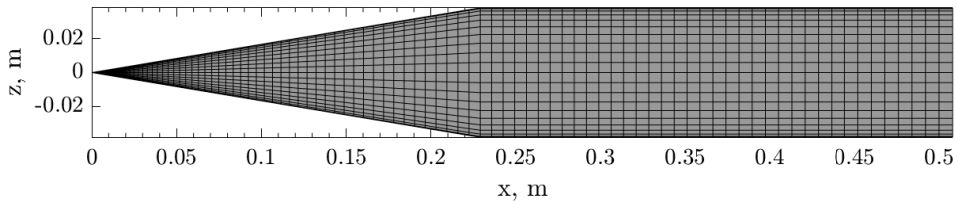
Validation data is presented for six analytically defined body shapes, for which wind tunnel experiments were carried out by Landrum & Babb [38]. Generated models of the body shapes are shown in figure 2.7. Here, a total of 2000 panels make up each configuration. Note that a half-cosine spacing method has been employed, ensuring that the areas with highest curvature (nose) are properly captured. Usually, the aerodynamic module defines atmospheric values are defined by a set altitude. However, in cases where wind tunnel data is used for validation, the actual test case flow conditions provided in the references are input.

Two inviscid methods were utilised in these validation cases: modified Newtonian and tangent cone on impact surfaces, with Prandtl-Meyer expansion used for shaded regions in both circumstances, and the previously discussed viscous approach used throughout. Figures 2.8 - 2.9 show a comparison of results for Mach 4.63 cases.

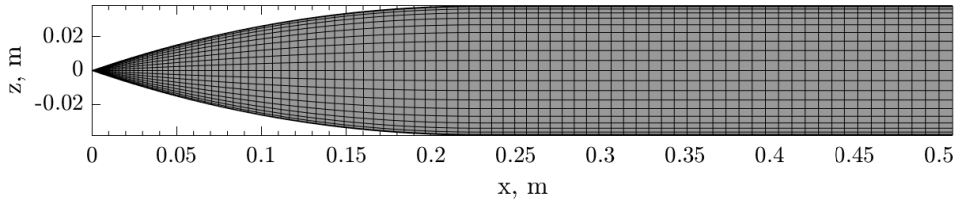
In the majority of cases, tangent cone/Prantl-Meyer provides an accurate comparison for normal and axial forces. For all configurations, there is a tendency for the normal component to be under-predicted as angle of attack increases. This is due to flow separation occurring on the upper surface during experiments, producing an area of suction and therefore increased lift. Since the panel methods in this work do not model the surrounding flow physics, this is not replicated within the numerical results. At lower angles however, where much of this work is focussed, the numerical methods match experimental data accurately.

For axial coefficient, the combination of inviscid and viscous methods produce similar trends to the experimental data, although the discrepancies are not consistent across all cases. The tangent cone methods shows a clear improvement over the modified Newtonian approach in this case, which is expected below high-hypersonic Mach numbers. Once again under-predictions can be observed at high angles of attack, which is again due to flow separation and trailing wake produced on the upper surfaces. At low angles the tangent cone method shows improved accuracy in 4 of the 6 cases, tending to over-predict the axial coefficient below 4 degrees.

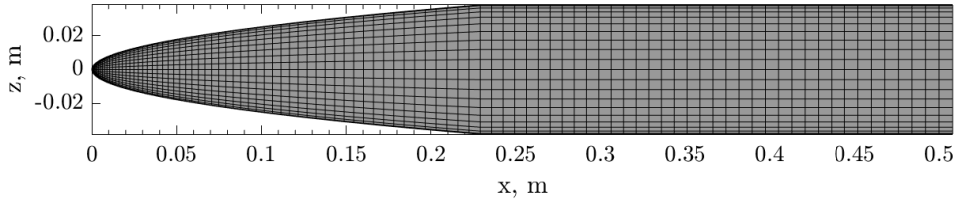
In terms of pitching moment coefficient (measured from the nose), the modified Newtonian method fits experimental data very closely, with tangent cone tending to over-predict the downward trend. Once again both methods perform well at low positive angles of attack, where this work is focussed.



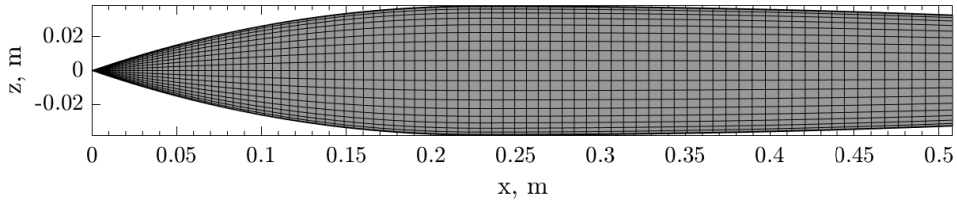
(a) Cone - cylinder



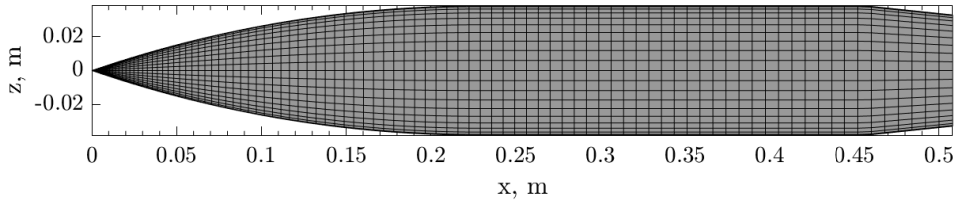
(b) Circular arc - cylinder



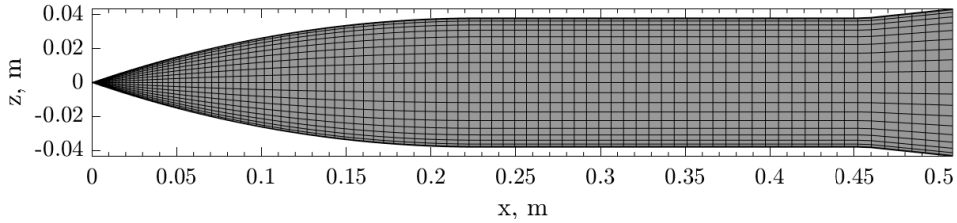
(c) Blunt nose - cylinder



(d) Circular arc - circular arc



(e) Circular arc - cylinder boattail



(f) Circular arc - cylinder flare

Figure 2.7: Analytical nose-body shapes modelled to compare with wind tunnel data

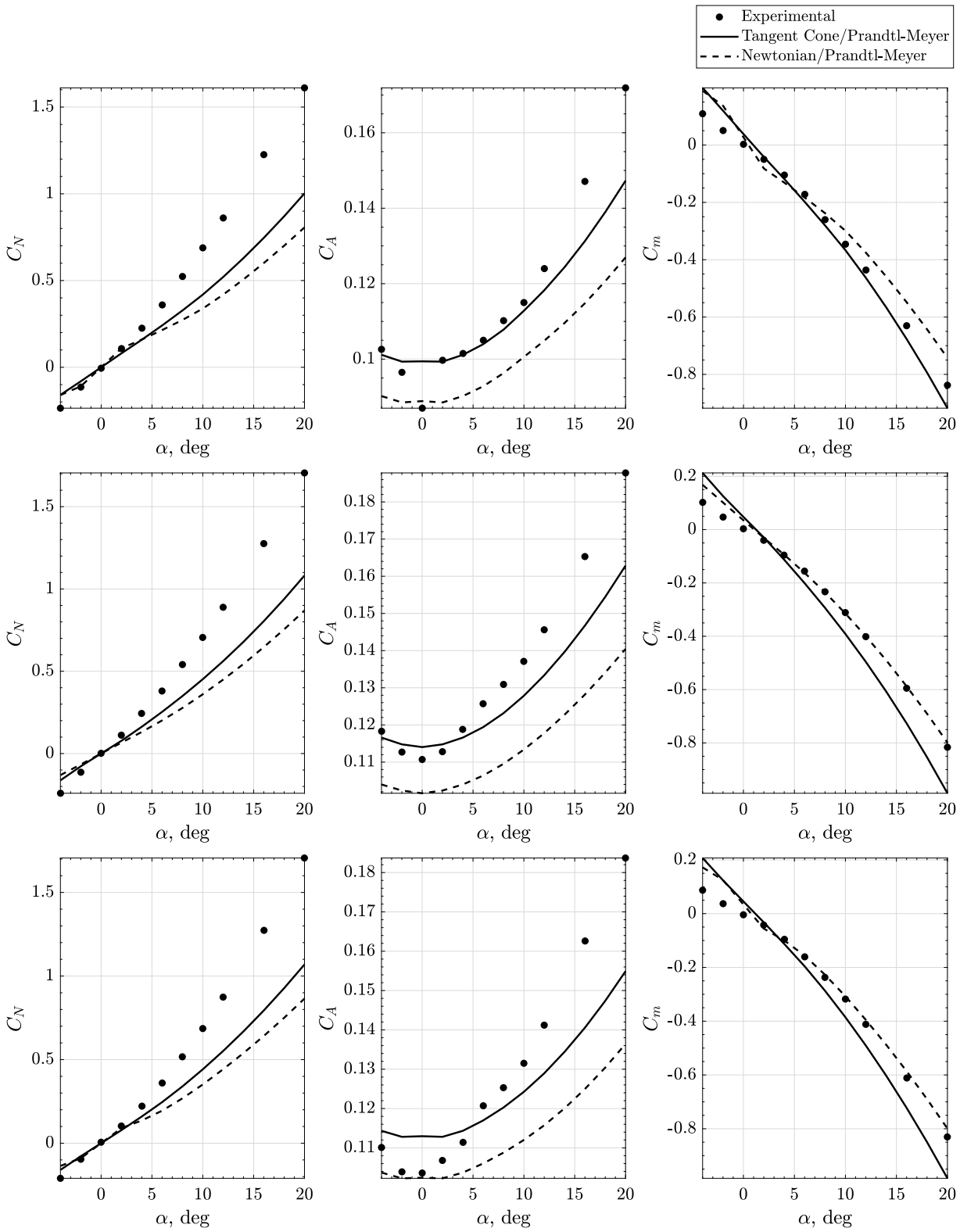


Figure 2.8: Numerical results compared to experimental data for body shapes (a)-(c) at Mach 4.63

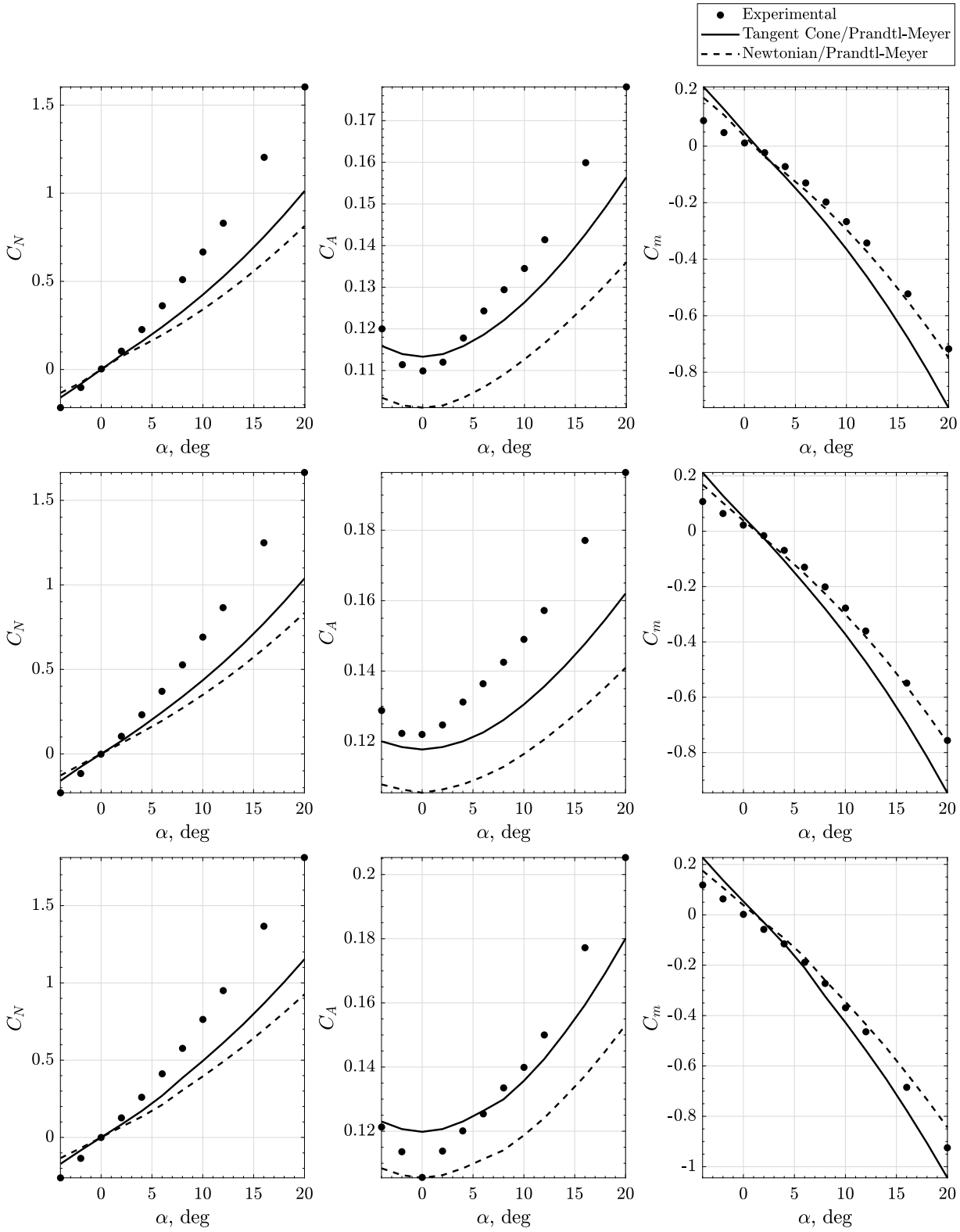


Figure 2.9: Numerical results compared to experimental data for body shapes (d)-(f) at Mach 4.63

Table 2.2: Supersonic flight state for NACA66-206 aerofoil validation

Parameter	Value
Mach	$2, 2.5, 3$
Angle of attack, deg	6
Altitude, ft	35000

Aerofoil

With good agreement between experimental and the implemented low-fidelity methods for a variety of body shapes, it is imperative to test the framework's ability to calculate the aerodynamic properties of lifting surfaces also. While this is a requirement of any aerodynamic framework, it is particularly important here, as independent methods are used for body and wing components.

With aerodynamic optimisation of aerofoil sections to be carried out, this presents a simple but crucial starting point for lifting body validation. Implemented predictive methods are compared with RANS data gathered in Giles & Marshall [39], where the NACA66-206 was analysed at the conditions specified in table 2.2.

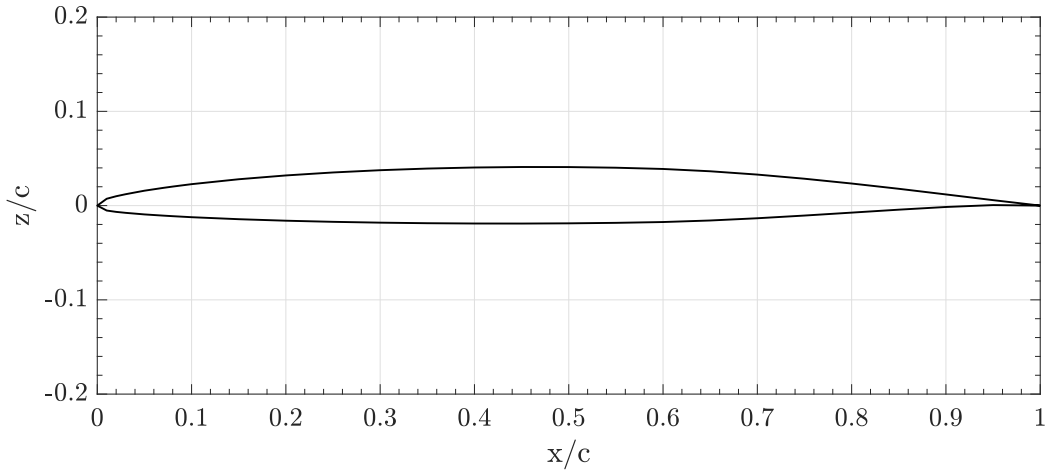


Figure 2.10: Baseline NACA66-206 aerofoil shape

Being a two dimensional problem, the obvious predictive method to employ is tangent wedge, reducing the problem on the impacted surfaces to a series of shocks defined by the local wedge angle created by the geometry, with respect to the freestream flow direction. For shadow surfaces, the Prandtl-Meyer method is again utilised, and the reference temperature approach employed to calculate viscous contributions. The resulting lift to drag ratios for the three Mach numbers are shown in comparison to the baseline case carried out by Giles & Marshall in figure

2.11.

The low-fidelity tangent wedge, Prandtl-Meyer and reference temperature method generally over-predicts the lift to drag ratio across the three Mach numbers. This is due to increased pressure forces near the leading edge, where surface properties must be solved using the Newtonian approach due to shockwave detachment. This over-prediction is therefore lowest at Mach 3, where more of the surface can be resolved using the tangent wedge method. Once again it is noted that such methods improve with increasing Mach number, with Mach 2 generally assumed to be the lower limit of applicability [14].

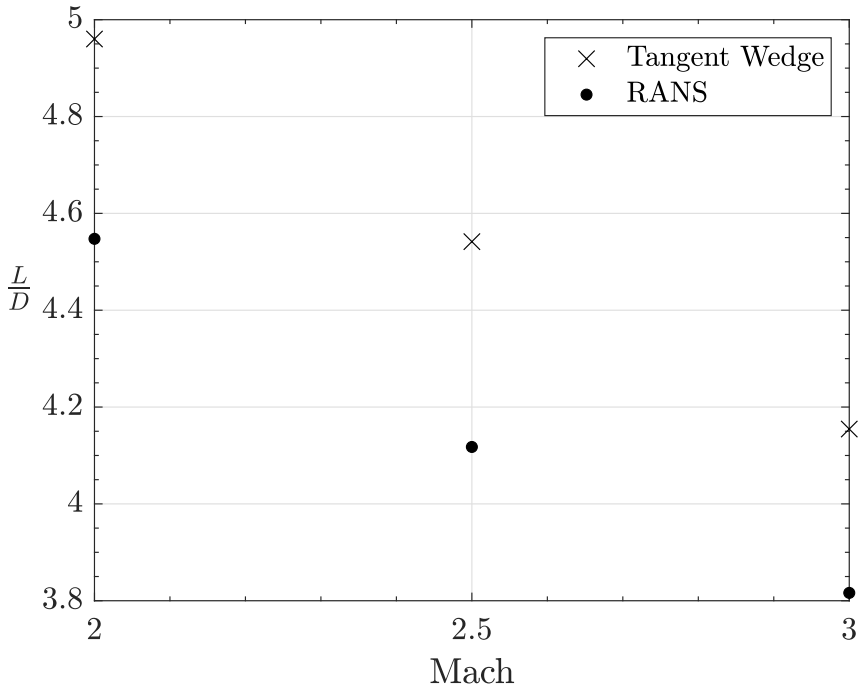


Figure 2.11: Numerical aerodynamic efficiency results compared with RANS data for NACA66-206 aerofoil

2.6.3 Wing-Body Configuration

Having demonstrated the framework's capability in predicting aerodynamic inviscid and viscous characteristics of body and lifting surfaces separately, the final validation step is to ensure accurate results can be obtained from combined geometries. Analysing and optimising wing-body vehicles is the primary goal of this work, meaning this overall validation of the combined predictive methods used across such vehicles is of paramount importance.

The experimental data for high-speed vehicles required to validate such models is severely lacking, primarily due to the highly sensitive classification that nations place upon their performance.

Furthermore, obtaining an accurate geometry of the components is not trivial, generally requiring some estimations to be made from given parameters.

An exception to this is the previously introduced X-34 RLV concept, which has well documented experimental aerodynamics data, as well as geometric characteristics. The aerodynamic and configuration data from Brauckmann [44] and Fuhrmann et al. [45] was therefore used to create a simplified model of the X-34, shown in figure 2.12, and analysed to compare the combined predictive methods against hypersonic experimental data.

The main discrepancies between X-34 and the simplified model are the lack of nozzle, control surfaces, and use of constant cross-section throughout the aft portion of the fuselage. These details were omitted from the model as they only have a small effect on overall aerodynamic characteristics (with undeflected control surfaces). Further discrepancies arise from the lack of available data for the wing and tail aerofoil sections, where estimated profiles were manually extracted from images in literature [44, 45].

Figure 2.13a shows the variation in impact prediction methods across the vehicles surface and components for Mach 6 and 4° angle of attack. Each part of the X-34 has default prediction methods consistent with table 2.3. However, note the use of Newtonian theory on the outboard wing section leading edge, this is due to the inclination of these panels toward the flow, making attached shockwaves impossible. As described in section 2.7, such panels are automatically adjusted to use the default detached shock method, allowing analysis to continue. Note that panels not impacted by the flow are also shown in figure 2.13a, in which case the Prandtl-Meyer relations are generally used across all components, with exception to the base.

The resulting pressure distribution over the entire surface is shown in figure 2.13b. As expected, the areas of maximum pressure are those with the highest inclination towards the flow; mainly the nose and outboard wing leading edge. Pressure then drops off drastically for decreasing inclination angle, where attached shockwave methods are utilised. Further reduction in pressure can be seen as the surface turns away from the flow, particularly on the upper side of the wing.

Finally, a comparison of predicted and experimentally measured force coefficients for Mach 6, across a variety of angle of attack values, can be seen in figure 2.14. These coefficients were obtained using the reference area given in Brauckmann [44]. As can be seen from the figure, the framework predicts both lift and drag coefficients very accurately throughout the whole angle of attack range, capturing both trends observed in experimental results. In terms of lift, the model tends to consistently over-predict the experimental data, with drag coefficient appearing to be a closer match. The model does tend to under-predict drag slightly at low angle of attack, before

over-predicting at $\alpha \geq 10^{\circ}$. Nonetheless, note the prediction also follows the experimental data in terms of minimum drag angle of attack, which for this Mach number is around 4°.

These results are very promising for the use of low-fidelity methods in this framework. The model has previously demonstrated capability in determining the characteristics of individual components. However, this shows its ability to analyse full configurations accurately, through the use of dynamic predictive methods based on component type and inclination angle. In fact, these results are considerable improvement over the cylindrical body results shown previously. One of the reasons for this is due to the increased Mach number, since the inviscid pressure methods become more reliable as Mach number rises. Furthermore, since the X-34 model employed for numerical analysis is a simplified representation, it is assumed that an improved geometry will deviate from the numerical results shown here. This will most likely result in further discrepancies between numerical and experimental data, particularly in the case of drag coefficient, considering the current simplified model matches the experimental data very closely.

This concludes the validation of the aerodynamic model, which has demonstrated accuracy in predicting a range of body, aerofoil, and full wing-body designs. The model can now be applied, with confidence, in an aerodynamic optimisation context.

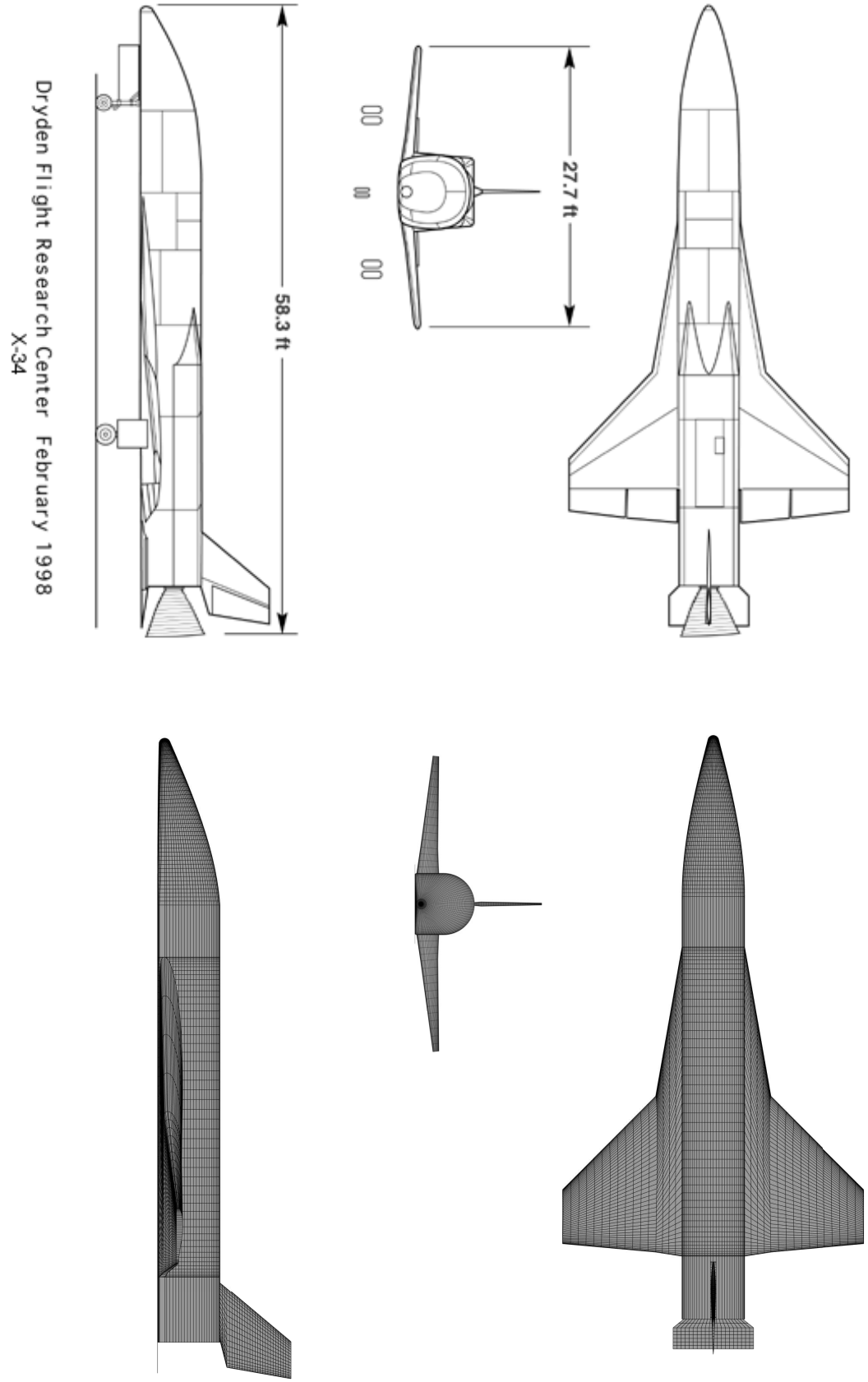
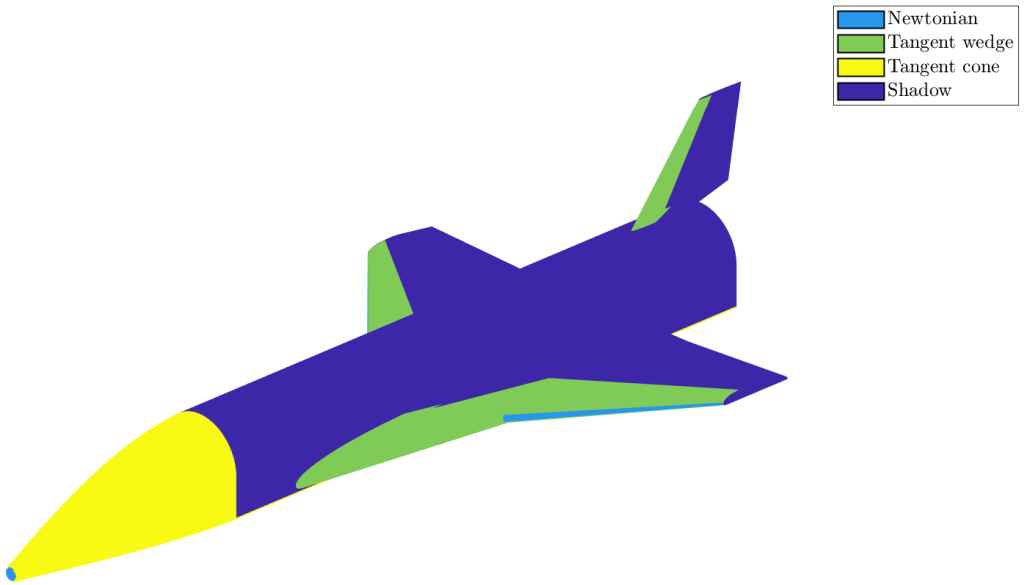
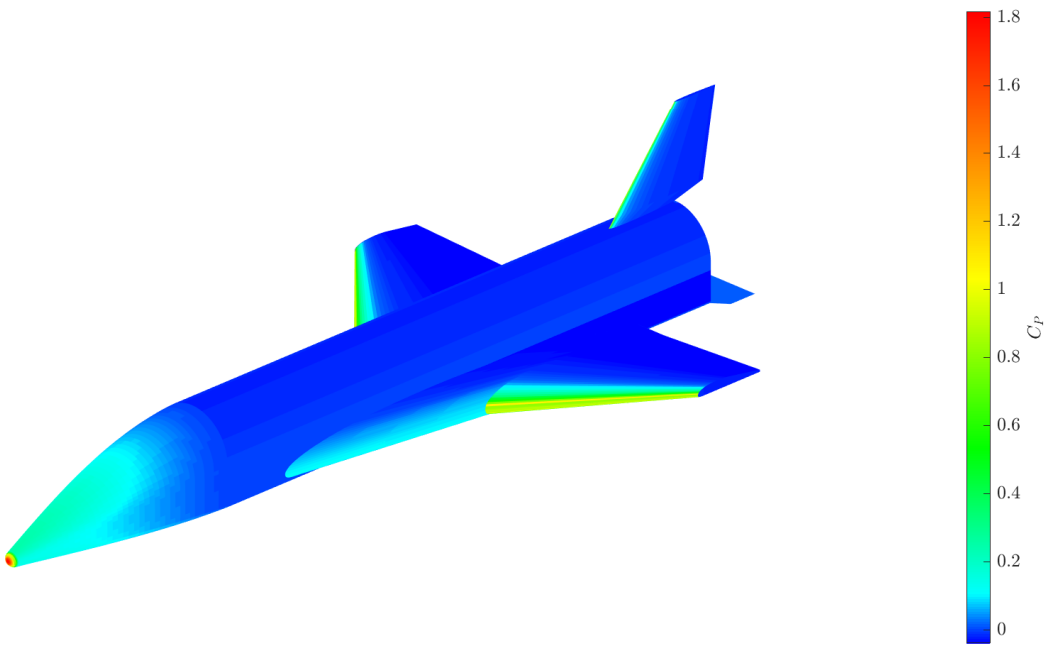


Figure 2.12: X-34 schematic [46] and simplified model used for validation



(a) Impact methods used



(b) Predicted surface pressure coefficient

Figure 2.13: Analysis methods and pressure coefficient for X-34 at Mach 6, 4° angle of attack

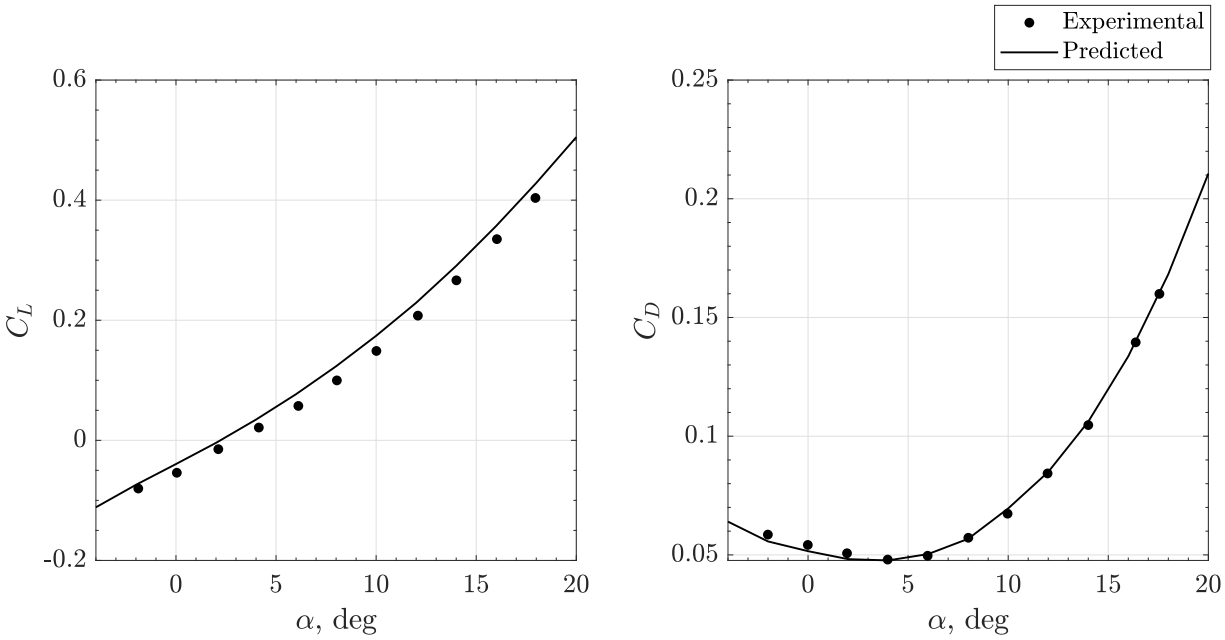


Figure 2.14: Predicted X-34 force coefficients against experimental data at Mach 6

2.7 Implementation in Overall Framework

With aerodynamic methods validated for a variety of use cases, what remains to be discussed is their overall implementation; from receiving an arbitrary geometry, to outputting component and full configuration aerodynamic characteristics. A brief overview of the required steps will be given here, to provide the reader with an understanding of the overall process and its versatility.

2.7.1 Inputs

Two inputs are required to allow aerodynamic analysis to be carried out: the freestream flow properties, and a point grid representing the geometry to be analysed. Only some key geometric properties must be passed to the aerodynamic module, which then derives the remaining inputs.

Freestream Flow Properties

Assuming a static flight state, only two properties are required to create a simple atmospheric model in continuum flow: Mach number and altitude. Using the model presented by Tewari [47], the remaining flow properties, such as air temperature and density, can be calculated.

The aerodynamic module assumes an unrotated geometry is input, therefore a third property to be input is angle of attack. An exception to this is where the geometries trim conditions are to be found, where a sensible starting point is assumed, discussed further in section 2.7.4. Defining angle of attack as a flow property, rather than rotating the geometry prior to aerodynamic

analysis, allows the freestream velocity vector to be rotated instead. This maintains simplicity within the overall process, allowing analysis to be carried out for multiple angle of attack values through the rotation of a single vector. Once the flow direction is rotated, equation (2.1) can be used to recalculate the local surface inclination of the geometry, and the overall process repeated.

Geometry

At a minimum, the geometry input should grid in either (x,y) column format for 2D shapes, or matrix from, with rows and columns representing the streamwise and radial dimensions of three dimensional bodies. A second option is available for 3D geometries specified in triangular format, where a $(n \times 3 \times 3)$ matrix is input, with each row specifying one of n triangles in x,y,z coordinates. Additionally an identifier can be specified, usually defined by the geometries class, which allows the aerodynamic module to select the appropriate prediction method.

With the geometry input, the module derives the data necessary for aerodynamic analysis. Panel centre points where loads will be applied are computed, as are panel areas and outward facing normals, to allow forces and their dimensional components to be resolved at the integration phase. Some panels may collapse into triangles or lines, which must be flagged so that they can be appropriately dealt with during the use of streamline based methods.

Finally, to correctly normalise output force and moment coefficients, reference area and characteristic length are required. If these are not specified directly, but geometries have been passed in a class format, the wing will be projected into the centreline, with reference area computed in the xy-plane, and moment arm defined as mean aerodynamic chord. If no wing is specified, the body xy-plane projection will be used instead.

2.7.2 Pre-processing

Local inclination angles are found using panel outward facing unit normals, and a rotated freestream velocity vector with respect to angle of attack, through equation (2.1). The sign of each panels inclination to the flow therefore determines if it is impacted by the flow (+ve), or in the shadow (-ve).

If an identifier has been specified, it is parsed through the impact and shadow methods list, to determine which approaches should be primary in determining aerodynamic characteristics. A list of all available methods, along with default options for a variety of geometries is shown in table 2.3.

Table 2.3: Available impact and shadow methods for aerodynamic calculation, with default options for input geometries

	Method	Geometry
Impact	Newtonian	Nose
	Tangent Wedge	Aerofoil, wing, tail
	Tangent Cone	Fuselage
	Shock-Expansion	
Shadow	Newtonian	
	Shock-Expansion	Aerofoil, wing, tail, fuselage
	Base Pressure	Base

Lookup tables are initialised for both θ-β-M, and tangent cone relations, allowing fast searching algorithms and interpolation methods to be employed, reducing computational overhead. For surfaces that use attached shock methods, a check must be carried out to ensure that such a shockwave is possible. To do this, the maximum concave deflection angle allowed for a weak attached shock to form on the surface at the given freestream Mach number is found using the θ-β-M table. Any panel inclination angle exceeding this maximum is therefore set to use Newtonian flow for computation of local properties. In a similar fashion, a cut off angle is applied in the shadow regime, with panels at a lower inclination to the flow than this minimum, typically set at -45°, using base pressure to calculate surface properties.

2.7.3 Analysis

Figure 2.15 depicts the overall process within the aerodynamic module, and in particular the logic required to analyse arbitrary input surfaces using the previously described inviscid methods. As a first step, surface flow direction must be calculated on a local basis, specifically defining a series of panels representing a streamline. Whether integrated streamlines or strip theory is utilised here, it is important to note that the direction of surface flow must be defined before any further analysis is calculated, even in purely inviscid cases, since the shock-expansion method requires this information.

For independent panel methods; those that rely solely on a given panels inclination to the flow, properties can be calculated for every panel in one pass through the chosen analysis method. However, when dependent, streamline based methods are used, an iterative scheme is required. Since such methods often propagate from panels further upstream in the impact region, all surfaces utilising independent methods are computed first. If the panel is at the beginning of a streamline, for example the upper surface of an aerofoil at high angle of attack, then freestream flow properties are used as a starting point.

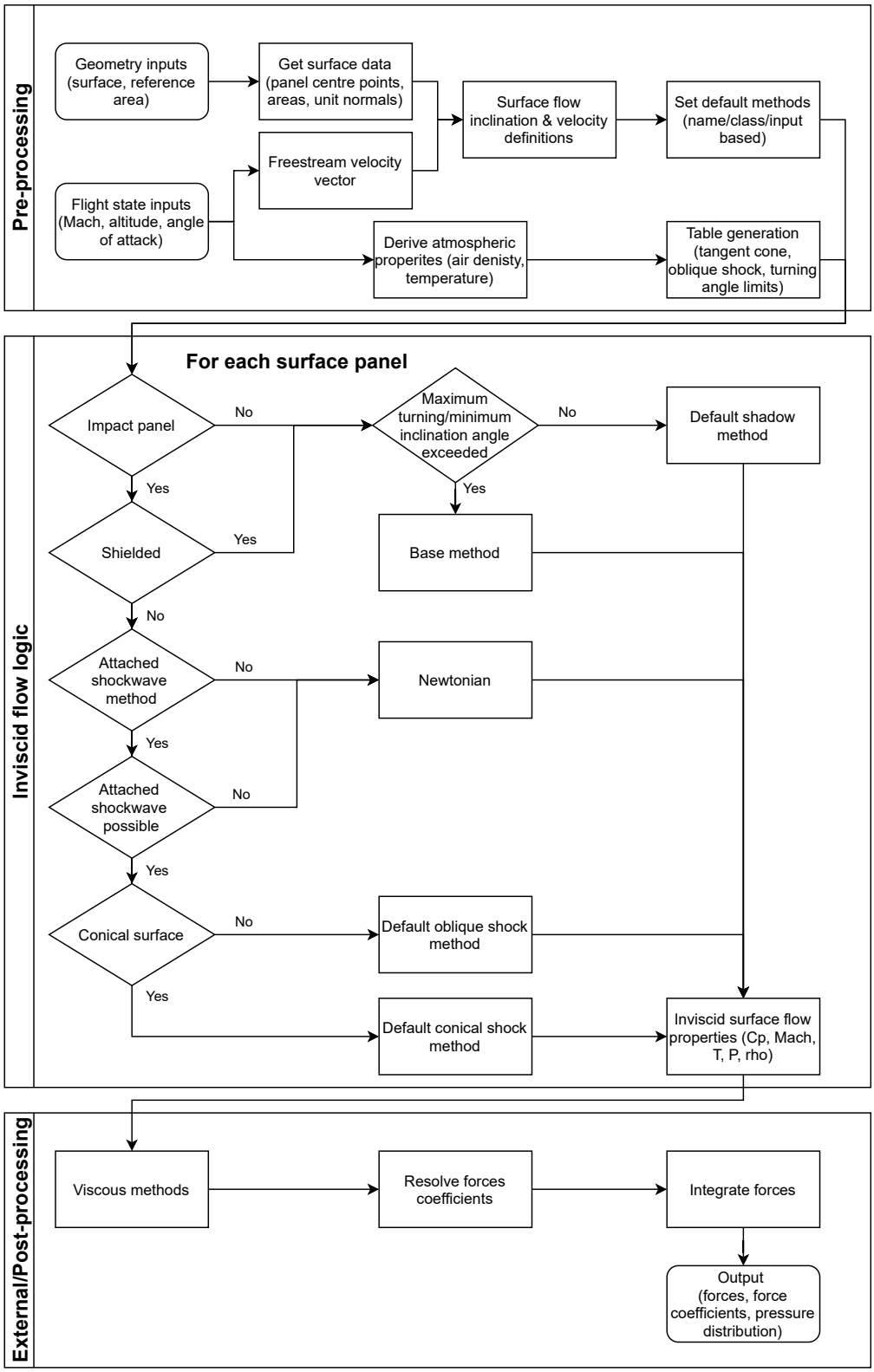


Figure 2.15: Overall aerodynamics methodology flowchart with inviscid flow logic

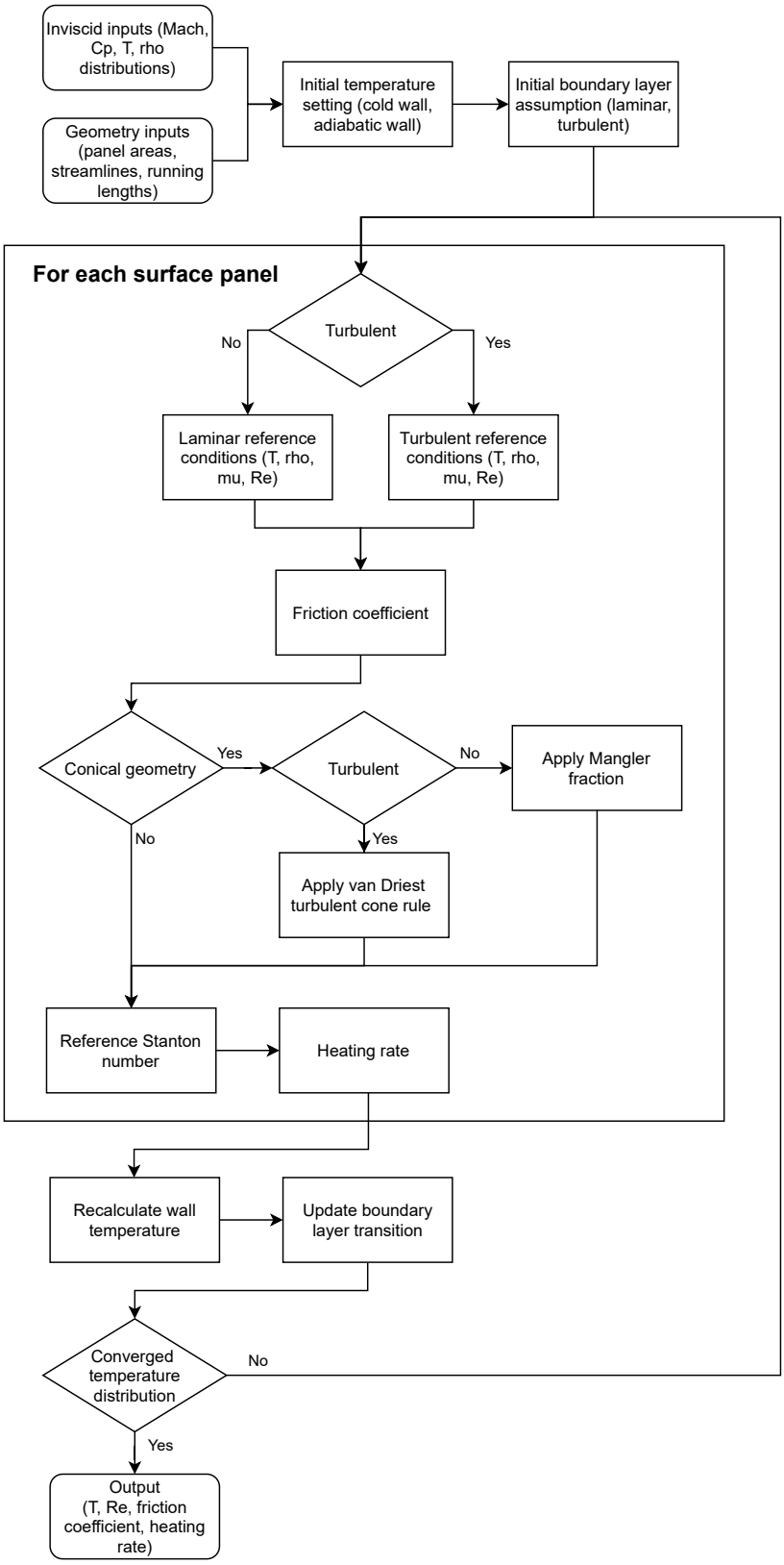


Figure 2.16: Iterative viscous methodology flowchart

Each inviscid method outputs local pressure coefficient, Mach number, temperature, and density of the flow over a given panel. This information, along with the geometry itself, is then passed to the viscous solver, discussed in section 2.4, which iterates until a converged local surface temperature and skin friction coefficient is found. Figure 2.16 shows the implementation of this process in further detail.

Finally, with all outputs received for inviscid and viscous modules, local characteristics can be integrated across the surface to determine overall aerodynamic characteristics of the given geometry. Panel pressure coefficients are resolved to lift and drag components, once again using their outward facing normals and angle of attack, before being normalised by the overall reference area. These coefficients, along with freestream flow properties, are then used to determine lift, drag, and moment forces on the geometry, which completes the analysis phase for the given flight state.

As this model utilises a component build-up approach, each part of the overall geometry is passed through this analysis phase separately. Outputs are therefore on a part by part basis, alongside integrated force and coefficients for the complete vehicle.

2.7.4 Trim

If a component or vehicle is to be trimmed, a further input of mass is required, in order to find the angle of attack at which lift equals weight. The vehicle will be analysed at a nominal starting angle of attack, $\alpha_0 = 5^\circ$, followed by a deviation given by $\alpha_1 = \alpha_0 \pm 5^\circ$ depending on whether the overall lift is less or greater than vehicle weight for the initial case. Thereafter, as the lift curve slope is roughly linear, trim angle of attack is repeatedly estimated through a linear interpolation search, down to three decimal places. A default tolerance of 1% is put in place to satisfy trim convergence. Of course, there are geometry, flight state, and weight combinations that will be impossible to converge. In such cases, default maximum and minimum angle of attack values are invoked, with trim calculation terminated if the system is unable to converge within these boundaries.

2.7.5 Static Stability

Static longitudinal and lateral stability characteristics can also be output from a given aerodynamic analysis. This is particularly useful when combined with trim convergence, where static stability is imperative. Longitudinal stability characteristics neatly falls out of the trim convergence routine, requiring only the previous state to be saved in order for results to be computed. If vehicles are to be analysed at rigid flight states, a small perturbation in angle of attack is applied after the nominal case has concluded, with static longitudinal stability given by:

$$\frac{dC_m}{d\alpha} < 0 \tag{2.33}$$

Determination of lateral stability requires a similar angle perturbation in the xy-plane, otherwise known as the sideslip angle, β. With vehicles being analysed as half-bodies through their longitudinal plane of symmetry for computational efficiency, estimation of static lateral stability requires some additional analysis.

Firstly, to maintain simplicity and avoid the recalculation of half-body geometric properties, the incoming flow velocity vector is rotated, rather than the vehicle itself. The now positive y-component of velocity can be included in equation (2.1) as normal, to determine local surface inclination towards the freestream flow. Computation of aerodynamic coefficients for half-body vehicles is therefore carried out in the same manner as before. However, since the flow is no longer parallel to the symmetry plane, the full vehicle geometry must be analysed. This is done by projecting the original half-body in the negative y direction, and computing geometric surface data for the new half-body. It is important to note that the reference area must be doubled in this case; allowing nominal half-body, and positive sideslip full configuration aerodynamic coefficients to be compared directly. With analysis complete and consistently normalised, static lateral stability can be tested for by the relation:

$$\frac{dC_L}{d\beta} < 0 \tag{2.34}$$

2.8 Conclusions

A variety of low-fidelity methods have been implemented for both inviscid and viscous aerodynamic prediction, providing a versatile set of tools, able to be used in tandem based on local flow properties and surface inclination. Methods such as modified Newtonian, tangent wedge/cone, and shock-expansion have been implemented to determine local inviscid conditions across the surface. Reference temperature methods then use the output inviscid flow characteristics, along with estimated surface streamlines, to determine the contribution of viscous effects. These local properties are then integrated across a given surface to provide overall aerodynamic forces, moments, and coefficients.

With a wide scope of geometries to be analysed throughout this research, a number of validation cases have been presented, in order to compare the models analysis with existing experimental data. Analytical body shapes, the NACA66-206 aerofoil section, and full X-34 wingbody vehicle have been analysed to show the accuracy of individual and combined low-fidelity inviscid methods, while two-dimensional flat plate, and three-dimensional conical and wing configuration data have been used to independently validate the employed viscous methods.

The use of these low-fidelity methods allow rapid computation of arbitrary two and three-dimensional shapes, requiring only a surface mesh and freestream flow properties to compute aerodynamic coefficients and forces. With default methods in place for body and lifting surfaces, along with appropriate checks to ensure the best method is employed at each discretised patch of a given surface, full configurations can be analysed using a component build-up approach.

The development of this aerodynamic prediction module is therefore the key component in terms of analysis within the large design space optimisation work that is to be carried out in this research. Having an algorithm that can handle the highly varied input geometries that a population based framework will produce is of paramount importance; avoiding potential errors that terminate analysis, or produce non-physical results, potentially leading to sub-optimal or infeasible designs being characterised as optimal. Furthermore, the computational efficiency of the module allows for thousands of configuration and flight state combinations to be analysed with relatively little computational cost, which is a necessity when optimising from a randomised starting population.

Chapter 3

Optimisation Methodology

3.1 Introduction

Techniques utilised for optimisation throughout this work are detailed in this chapter. Aligned with the overall theme of this research, a number of optimisation approaches, and optimiser specific tools are employed in a modular fashion, allowing both multi-fidelity and multi-disciplinary problems to be tackled, utilising best practices from a wide scope of prior research.

Defining, implementing, and leveraging appropriate optimisation methods for this research is perhaps the most critical component of the entire framework. While geometric design space, techniques used for analysis, and their versatility in creating arbitrary designs with accurate performance characterisation are absolutely necessary; the relative power of these tools will be under or inappropriately utilised without proper utilisation of optimisation methods that enable the design space to be searched efficiently.

The degree to which a design space can be exploited to achieve optimal results is entirely problem dependent. In a typical existing design optimisation, many of the characteristics will either be pre-determined, or have little scope to change, requiring fine tuning and accurate analysis to squeeze every ounce of performance from the small amount of possible variability. Such problems logically desire incremental changes to be made towards regions of improved performance, as opposed to large scale scattered sampling that is independent of the initial design.

In the absence of an existing model, a larger scale design optimisation approach is desirable; seeking to find areas of the design space with promising performance, but with a higher degree of uncertainty in performance quantification. Trade-offs such as these are often non-negotiable, due to computational power required, and/or inability to automate the use of highly accurate and all-encompassing analysis. In these exploratory optimisations, it therefore makes sense to determine optimality on a spectrum, or employ methods that attempt to reduce uncertainty in

performance quantification, rather than attempting to find the absolute minimum of a cost function that may not be representative of real-world conditions.

Further considerations must be taken into account when working with large design spaces, including determination of feasibility in addition to optimality, since some variable combinations may result in attractive objective functions at the cost of destructive performance elsewhere. This may lead to small and sparse feasible regions within the design space, further demonstrating the need for an efficient, large scale search of designs. These pockets of feasibility, along with the desire to define optimality on a spectrum, combine to highlight the need for diversification within the overall process, thus decreasing the likelihood of premature convergence within a small region of the design space.

The methods implemented here begin from commonly employed gradient based variants, which introduce the fundamental understanding of optimisation processes. While providing the necessary background, these methods are also used to benchmark the more bespoke techniques that are used for exploratory design optimisations, which are the main focus of this research. These heuristic methods are then described in detail; building from their inception to the current state-of-the-art. Single and multi-objective methods are employed, with a focus on agile design space exploration. Methods used to enforce constraints are then discussed, which attempt to strike the balance between finding feasible designs, and ensuring they do not result in premature convergence of the optimisation. The developed population based optimisers are then tested against industry standard variants, demonstrating their fitness for purpose within large scale design optimisation work.

3.2 Local Optimisation

This section focusses on the theory behind gradient based methods, in which a starting point within the design space is iteratively perturbed in each variable dimension, to determine a promising search direction towards an improved cost function. This discussion builds from inefficient entry level algorithms towards more sophisticated techniques that can be readily applied within the scope of this research.

Beginning from first principles, the motivation behind optimisation methods is to take a function, dependent on a number of variable parameters, and minimise the resulting output:

$$\min \ f(\mathbf{x}) \tag{3.1}$$

Where f is the cost or objective function, and $\mathbf{x}$ is a vector of optimisation variables that will be altered throughout the minimisation process. This generalisation clearly has far reaching

applications, throughout engineering and beyond. With such a diverse set of potential problems, a similar set of methods have been developed to tackle them; allowing the user to determine which is the most appropriate to their application.

3.2.1 First-Order Gradient Based

Derivative based search methods rely on two key assumptions about the optimisation problem: that all functions associated with it are at least twice continuously differentiable, and that a reasonable or close to optimal starting point is provided [48]. In such an approach, the goal is to begin with this original design, and iterate the optimisation variables, based on a change in these parameters that improves the cost function, until a minimum is reached. Generally, this is described by the following equation:

$$\begin{aligned} \mathbf{x}_i^{k+1} &= \mathbf{x}_i^k + \Delta\mathbf{x}_i^k \\ i = 1,2,...,n \text{ and } k &= 0,1,2,...,m \end{aligned} \tag{3.2}$$

Where $\mathbf{x}$ is a vector of the design variables, with i being the design variable number. $\Delta\mathbf{x}$ is a vector of the design change, and k is the iteration number. In order to improve upon the initial design, in terms of a give cost function, the change in optimisation variables at a given iteration are calculated by perturbing each in turn, and using the respective cost function values, gradients, and potentially second derivatives of these functions, to find an overall direction that improves the design. Ideally, this process repeats until an optimal value is found, however other criteria, such as maximum number of iterations, or tolerances with respect to the cost function or change in design, can also be used to terminate the optimisation.

Under the assumption that a single starting point is selected, techniques purely based on equation (3.2) can only be guaranteed to converge to a local optimal, due to the fact that the changes in optimisation variables are based on perturbations relative to the initial design.

Beginning with an unconstrained optimisation, the determination of the change in design variables at a given iteration is separated into two sub-problems:

$$\Delta\mathbf{x}^k = \alpha_k \mathbf{d}^k \tag{3.3}$$

That is, the calculation of a positive scalar step size, α_k, and a search direction $\mathbf{d}$, that improve the current design. Many methods have been derived to tackle these two sub-problems, however their limited use in the work justifies only a brief discussion of the methods employed.

In terms of search direction, the conjugate gradient method, an advancement on the typical steepest-descent algorithm is utilised. The determination of gradient is shown in equation 3.4, which points in the direction of cost function increase:

$$\mathbf{c} = \nabla f(\mathbf{x}^k) = \left[\frac{\partial f(\mathbf{x}^k)}{\partial x_1^k} \quad \frac{\partial f(\mathbf{x}^k)}{\partial x_2^k} \quad \dots \quad \frac{\partial f(\mathbf{x}^k)}{\partial x_n^k} \right] \tag{3.4}$$

Which represents the gradient of a cost function $f(\mathbf{x}^k)$, as a set of partial derivatives, with respect to the previously introduced vector of designs design variables $\mathbf{x}$. By this definition, the gradient vector points towards the direction that locally maximises the overall cost function, meaning that the search direction $\mathbf{d}$, must be the negative of this vector. Incorporating this with direction information from the previous iteration, forms the conjugate gradient, as shown in 3.5:

$$\mathbf{d}^k = -\mathbf{c}^k + \beta_k \mathbf{d}^{k-1} \tag{3.5}$$

$$\beta_k = \left(\frac{||\mathbf{c}^k||}{||\mathbf{c}^{k-1}||} \right)^2 \tag{3.6}$$

Advice has also been taken from Arora [48], to alter the conjugate gradient from the Fletcher-Reeves equation (3.5), to the Polak-Ribiére [49] variation, depending on their resulting values at a given iteration. Since this method requires information from the previous iteration, the initial iteration of the optimisation will be identical to that of the steepest-descent algorithm ($\mathbf{d}^k = -\mathbf{c}^k$).

Once the direction of descent has been computed, all that remains as an unknown is the scalar value defining how far in this direction should the design travel, in order to reach the new design state, $\mathbf{x}^{k+1}$. The calculation of this step-size value is thus a simple line search of the function $f(\alpha_k)$, regardless of the number of design variables employed, and its purpose is to find the maximum decrease in the cost function, based on the pre-determined search direction.

This sub-problem can be further divided into two parts: locating the interval in which the minimum value resides, and reducing this interval to the desired tolerance. Once again, many algorithms exists for approaching both of these tasks, and the following implemented algorithms have been chosen with focus on efficiency in terms of cost function evaluations, and accuracy of the computed optimal step-size. Note that this generally results in a trade-off, as increasing the accuracy of optimal step-size may increase cost function evaluations for a given iteration, but may also result in faster convergence and thus, a reduction in total iterations.

Locating the interval in which the minimum value exists is achieved by the variable-interval golden section search algorithm. Here, a small value δ is selected as the step-size, and assuming the resulting cost function is lesser than that of the initial value, that is $f(\mathbf{x}^k + \alpha \mathbf{d}^k) < f(\mathbf{x}^k)$, it

is used as the initial lower boundary of the line search. The upper boundary is then computed, with the aid of the golden ratio, by taking a step in the 1.618δ direction, and computing the resulting cost function. This process repeats, shifting the upper and lower boundaries by this increasing step, until the cost function begins to increase. At this point, the minimum value must lie between the two previous boundaries of the step-size. Within these boundaries, three cost function values have already been computed, and these can now be used to initialise the uncertainty interval reduction phase. This problem can also be approached using the golden ratio, however two other methods have been chosen, based on the reasons specified previously.

The first approach uses the three values of the step-size corresponding to the cost function values that lie on or within the computed initial interval of uncertainty, to produce a quadratic approximation of the overall function $f(\alpha_k)$, within this boundary. Following this, the function can be solved and a minimum value found, which replaces one of the three initial step-sizes depending on its location. This process is repeated until the interval of uncertainty drops below the specified tolerance. In terms of exact line searches, this method is highly efficient. However, any exact line search requires the gradient of the cost function to be calculated at every candidate step-size, which depending on the cost function, can be highly computationally expensive.

In such situations, an inexact line search may be preferable. Armijo's rule has been implemented as an option in this case, which ensures that the step size is not too large, overshooting the minimum point, while simultaneously demanding a reasonable improvement in the cost function. Further details on Armijo's rule, along with the algorithm can be found in Arora [48].

3.2.2 Second-Order Considerations

The rate of convergence of the conjugate gradient method is limited by the use of only first-order derivatives of the cost function with respect to the design variables, and computation of second-order derivatives can improve upon these methods. However, differentiating the already computed gradient vector with respect to the design variables results in a total of $n(n+1)/2$ second-order terms to be determined, in order to form the Hessian matrix, as shown below:

$$\nabla^2 f = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix} \tag{3.7}$$

Note that some of the nomenclature has been dropped here for cleanliness, however the Hessian corresponds directly to a differentiation of the gradient vector in (3.4), with respect once again to the design variable vector.

One such second order approach is Newton's method, which perturbs the variables at a given iteration, and finds the change in cost function by a second-order Taylor expansion, requiring the Hessian to be computed directly [48]. The determination of this matrix at every iteration significantly increases the computational load, and furthermore, optimisations utilising direct Hessian methods run into problems if it is singular or not positive definite.

To overcome these issues, quasi-Newton methods have been implemented, which allow an approximation of the Hessian to be formed and updated using only the current design point and first-order information. Not only this, but their formulation ensures positive definiteness and symmetry throughout the process, meaning that the overall change in design variables, or search direction, will always reduce the cost function.

The BFGS method [50,51] has been utilised in this implementation, which significantly reduces computational time by approximating the Hessian from previous search directions. The search direction for a given iteration is determined by solving the linear system of equations:

$$\mathbf{H}^k \mathbf{d}^k = -\mathbf{c}^k \tag{3.8}$$

Where the Hessian is approximated via:

$$\mathbf{H}^{k+1} = \mathbf{H}^k + \frac{\mathbf{y}^k \mathbf{y}^{k^T}}{\mathbf{y}^k \cdot \mathbf{s}^k} - \frac{\mathbf{c}^k \mathbf{c}^{k^T}}{\mathbf{c}^k \cdot \mathbf{d}^k} \tag{3.9}$$

As observed, an initial Hessian matrix must be provided, due to the incremental nature of equation (3.9). Since the BFGS method guarantees a symmetric positive definite matrix, a similar matrix must be input. To avoid an initial direct calculation of the Hessian, an identity matrix is employed instead. Much like the conjugate gradient method described earlier, this results in the first iteration of the BFGS approach being identical to that of the steepest-descent algorithm.

3.2.3 Bound & Constrained Local Optimisation

So far, the methods discussed assume that the design space is unconstrained, and that variables are allowed to take any conceivable form. In reality, this is rarely the case in engineering design optimisation, with realistic boundaries being placed on the search space, along with constraints on variables and design characteristics or performance. In such cases, equation (3.1) is extended to become:

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & x_{i_{lb}} \leq x_i \leq x_{i_{ub}};\ i = 1:n, \\ & g_j(x) \leq 0;\ j = 1:p, \\ & h_k(x) = 0;\ k = 1:m \end{aligned} \tag{3.10}$$

Where $x_{i_{lb}}$ and $x_{i_{ub}}$ are the design variable lower and upper boundaries, p is the number of g inequality constraints, and m is the number of h equality constraints.

Constraints which are violated may drastically affect the final value of the cost function, imposed by methods that will be discussed in section 3.4. What is important to note here is that since some form of penalty will be added directly to the cost function, constraint function gradients must be also be obtained when determining descent direction.

To utilise the potentially large number of gradients computed at every cost function call to determine search direction in an efficient manner, a sequential quadratic programming routine is employed. This is a sub-problem that must be carried out at every iteration, and is briefly described below:

$$\begin{aligned} \min \quad & \bar{f} = \mathbf{c}^T\mathbf{d} + \frac{1}{2}\mathbf{d}^T\mathbf{d} \\ \text{s.t.} \quad & \mathbf{N}^T\mathbf{d} = \mathbf{e}, \\ & \mathbf{A}^T\mathbf{d} \leq \mathbf{b} \end{aligned} \tag{3.11}$$

Where $\bar{f}$ represents the change in cost function; $\mathbf{N}$ is matrix composed of the change in equality constraint functions, differentiated with respect to the change in design variables; $\mathbf{e}$ is the negative of the equality constraint values; with $\mathbf{A}$ and $\mathbf{b}$ representing the same terms for inequality constraint functions.

Equation (3.11) describes a convex sub-problem, meaning if a minimum exists, it will be unique. Thus solving this sub-problem ensures that if a feasible search direction exists, the cost function value, including penalties produced by violating constraints, will reduce in a given iteration [48].

3.2.4 Drawbacks

While gradient based optimisers are a powerful tool used extensively in engineering literature, the methods described here only guarantee convergence on a local minimum within the cost function space. This may be ideal for pre-existing designs that have scope for fine tuning and thus a more efficient design, but offer little insight when working with large design spaces, or

generally when a close to optimal starting point is not present.

The exploratory optimisation problems sought to be conducted here fall into the latter category, and require the design space to be viewed from a different angle than that of a single initial design to be improved upon. Application of gradient methods within these type of problems would result in a variety of final designs across a number of optimisations and starting points. While some designs may be genuinely promising, many will be grossly misidentified as optimal, with the optimiser converging to a local minimum that is far away from that of the global. Note that as the design space grows and becomes more tightly constrained, the performance of gradient based algorithms can only deteriorate, due to an increasingly large cost function space with many local minima, surrounded by noisy regions of constraint violation. This lack of repeatability and poor convergence characteristics is the antithesis of what optimisation methods are designed to produce, requiring considerably larger coverage of the design space.

3.3 Global Optimisation

The main focus of this work is generating novel designs from an arbitrary starting point. Local optimisation methods work well on the basis of having a starting point that is close to optimal, thus small changes to the design variables can produce gradient values and allow a search direction to be determined. However, in cases where no starting point is available; the starting point is not close to optimal; or its optimality has not been determined, an exploratory search of the design space is a key tool that should be utilised. Proper implementation of this method allows new designs to be produced, and can quantify the optimality of an existing design for the given problem and variable set. It must be emphasised that such work is therefore relevant whether an existing design is present or not, since a design's optimality must be quantified on a global level, in order to determine it a close to optimal starting point.

While gradient-based methods can be adapted to become global optimisers; using multiple start point or hill climbing methods for example, they become inefficient and unreliable as the design space grows and problems become tightly constrained. In such cases, it is more intuitive to think of the search space in a collective manner, rather than adapting one set of variables at a time. Population based techniques have therefore been chosen for global optimisation applications in this research. Such methods can be applied to a broad range of problems, since they only require the cost and constraint function values to progress through iterations. Furthermore, the lack of gradient information allows them to tackle non-smooth or discrete problems, and highly constrained or noisy applications.

Population based optimisers initialise multiple candidate solutions, and update them simultane-

ously within a given iteration. The underlying rationale behind these methods is to find solutions by introducing some randomness, usually within a nature-inspired process. Since these approaches are heuristic, a high number of cost function evaluations are required to ensure that solutions obtained are optimal. Not only is this due to the entire population being evaluated at every iteration, but also due to a potentially high number of iterations required to achieve convergence. Nevertheless, in a large design space, highly constrained problem, optimal solutions can lie in small crevices within a noisy cost function space, meaning a high number of cost evaluations should be considered a necessity, rather than a drawback.

Since these methods are numerical in nature, there is no guarantee that globally optimal solutions will be found. Thus, various methods and adaptations have been proposed through the years, forming a rich research branch that continues to evolve. The probability of a globally optimal solution being found using a numerical method is of course problem dependent. Therefore, it is no surprise that such methods, or adaptations made to them, can simultaneously demonstrate positive and negative effects across optimisation problems. Since there are a large number methods and various nuanced developments within them, this work does not intend to review all the pros and cons for the problems presented. Some methods and their alternatives will be discussed and compared where it is deemed important, otherwise the choice of selected methods will be justified.

At this point, it is imperative to note that finding the absolute global optimal of a given problem is not necessarily the primary goal of the problems discussed in this research. Considering that this work is aimed at the preliminary design phases, a spread of potential designs is much more desirable, due to inaccuracies arising from the low-fidelity cost function. In single-objective optimisation there is no meaningful way to achieve this, since most cost functions will have a unique minimum value. However, utilising multi-objective cost functions allows the formation of performance trade-off curves or Pareto fronts [52], which will be described more in section 3.3.4. Here, a range of designs are simultaneously defined as mathematically optimal, allowing the most realistic to be selected for further analysis.

Clearly then, producing a diverse set of results is of paramount importance, and a variety of techniques have been selected for the purpose of increasing diversity, ensuring that the optimisation does not overly focus on the ideal trade-off point between cost functions. This aligns well with the strengths of population methods, and the purpose of an exploratory search; promoting the exploration of various avenues to find crevices throughout the objective function space.

3.3.1 Candidate Techniques

As mentioned previously, numerical population based methods have gained much attention throughout the years, with various nature inspired techniques being developed. Most prevalent of these is the genetic algorithm (GA) [53], followed by particle swarm optimisation (PSO) [54]. These algorithms have been compared and contrasted across various fields and applications, showing great versatility with little or no modification. Many other techniques have also been developed throughout the years, such as ant colony optimisation and simulated annealing. However, with such a large spread of GA and PSO literature, containing many variants and problem dependent nuances, the choice was quickly narrowed down to these two popular and highly versatile algorithms.

The genetic algorithm was made popular in the 1970s by Holland [55]. As its name suggests, it is inspired by genetics and evolution, mimicking reproduction and survival of the fittest [56]. By analysing the objective and any constraint functions, the algorithm will begin discarding undesirable traits within the population, in terms of the design variables. Similarly, variables or genes that perform well will likely be passed on through generations. The binary nature of the GA allows for both discrete and continuous design variables, and has been demonstrated to work effectively in highly constrained, non-linear cases.

While the genetic algorithm could be utilised effectively in this research, particle swarm optimisation has been shown to be more efficient for purely continuous problems, due to its inherently continuous nature [56]. Many papers have also shown particle swarm to be an improvement over the GA in terms of efficiency and optimal values found [56–59], although as mentioned previously, the ideal choice of optimiser is often problem dependent. On top of this, PSO algorithms have been updated to include characteristics found in evolutionary methods, such as mutation and ageing [60, 61]. For these reasons, it was decided that the particle swarm optimiser was the superior option, with the ability to incorporate most of the techniques found in genetic algorithms. Fully continuous design spaces have therefore been employed, ensuring they are as transparent as possible, and maximising the probability of optimisation convergence. Experiments showing comparisons between the PSO and GA will be discussed in section 3.5. However, the following will present a more detailed PSO introduction, and in depth discussion of the specific implementation within this research.

3.3.2 Single-Objective Particle Swarm Optimisation

Originating in 1995 by Eberhart and Kennedy [62], Particle swarm optimisation has become increasingly popular among researchers, particularly in continuous design space problems. In a broad sense, a population of particles is initialised within the boundaries of the design space,

and proceed to iteratively travel through the hyperspace in search of an improvement in the objective function(s), until some convergence criteria is achieved. This movement of particles is driven by the position or set of positions that have resulted in the best cost function values. Importantly, one of the main distinctions between PSO and other evolutionary algorithms, is that it retains these ideal solutions in memory, and thus particles will continue to be drawn toward their locations in consecutive iterations until an improved position is found. Looking at an individual in a population of particles, its position is updated as follows:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{v}_{k+1} \tag{3.12}$$

Where $\mathbf{x}$ is the vector of design variables defining the particle's position at the current and next iterations k, $k+1$, and $\mathbf{v}$ is the particle velocity, obtained by:

$$\mathbf{v}_{k+1} = \omega \mathbf{v}_k + c_1 r_1 (\mathbf{p}_k^b - \mathbf{x}_k) + c_2 r_2 (\mathbf{p}_k^g - \mathbf{x}_k) \tag{3.13}$$

Here $\mathbf{p}_k^b$ and $\mathbf{p}_k^g$ represent the best observed position of particle d, and the best position observed across the entire population. The constants $c_{1,2}$ define cognitive and social learning factors respectively, while $r_{1,2}$ are random numbers in the range $[0,1]$. The difference in this equation from the original proposed in 1995 [62] is the inclusion of an inertial term ω, introduced by Shi and Eberhart [63]. As its name suggests, this term retains some of the particle's velocity from the previous iteration, to ensure that some consistency is maintained, and particles are not making rapid changes in direction at every update. Equations (3.12)-(3.13) make up the fundamental PSO algorithm, demonstrating its ease of implementation for a multitude of problem spaces.

Applying randomness to the process allows areas surrounding the best observed positions to be explored without the need for gradient information. However, as previously stated, it comes with the price of requiring a large number of cost function evaluations before any confidence in convergence can be drawn. Furthermore, as with all global optimisers, premature convergence to local optima can be observed. Clearly then, further adaptations are required to improve the performance of PSO, and increase the probability of obtaining the global optimal for a given problem. Various concepts have been introduced with varying degree of success in literature, and this work does not look to exhaustively review each one. However, promising techniques that have been implemented in the current research will be detailed.

Research also varies in the recommendation of social and cognitive factors, and treatment of the inertial term, with many citing both factors as problem dependant. Nevertheless, the majority set $c_{1,2}$ as constant in the range $[1,2]$, to occasionally allow particles to "over-fly" the best particle hyperparameters, and further explore the design space. The inertial term is often dynamically reduced throughout the simulation, for example a linear reduction with respect to the iteration

number [63], although initial and final values again vary.

Particle Initialisation

Initialisation of the swarm is an important task that should not be overlooked. Poorly initialised particles can lead to premature convergence, particularly in highly constrained or high dimensional problems. In any case, upper and lower boundaries of design variables are required, so that intermediate values can be distributed to the population.

The simplest way to do this is by assigning variables randomly using a uniform distribution. However, as noted by Richards and Ventura [64], this method fails to provide a consistent even distribution across the entire design space. The approach they recommended to achieve this is centroidal Voronoi tessellations. Here, the hyperspace is divided into n_{var} compartments, with the goal to find the centroid of each. These centroids then become the initial particle positions. Obtaining a precise centroid of each compartment in the hyperspace is computationally expensive, and since the goal is to produce a more evenly spread distribution, an approximation of the centroid will suffice. Ju-Du-Gunzburger algorithm [65] achieves this by choosing a cloud of random points and initial centroids. Points are then grouped based on which centroid they are closest to, and the centroid is moved towards the average distance that the set of points lie away from it. The random points are then re-initialised, and the process repeats until a stopping criteria is met, such as maximum number of iterations.

Along with the Ju-Du-Gunzburger method, Latin hypercube sampling [66] has also been implemented for particle initialisation. This is a generalisation of the Latin square concept; where samples must be selected from unique rows and columns, producing a set of uncorrelated points. Extending this to arbitrary dimensions essentially requires each variable to be split into a number of partitions n, equal to the number of sample points requested. The exact values to be selected are then drawn randomly using a uniform distribution from each partition. This guarantees that each sample will lie within a unique subspace of the overall design space, providing a large spread of coverage in the initial population.

Neighbourhood Topology

The original PSO has a star topology, where every particle is driven by the overall best in the population. Neighbourhood topology looks to replace this single globally dominant solution by splitting swarms into sub-swarms, and replacing the global best with multiple local bests. The local best topology is the antithesis of the star topology, with each particle linked to its nearest K neighbours, where usually $K = 2$, resulting in a circle topology. In a neighbourhood, each particle cost function is compared with that of its neighbours, to define a local best. In this case there as many neighbourhoods as there are particles, since no particle has the same two neighbours.

These local bests are then used to define the velocities of their neighbouring particles, adding a third term to (3.13):

$$\mathbf{v}_{k+1} = \omega \mathbf{v}_k + c_1 r_1 (\mathbf{p}_k^b - \mathbf{x}_k) + c_2 r_2 (\mathbf{p}_k^g - \mathbf{x}_k) + c_3 r_3 (\mathbf{p}_k^l - \mathbf{x}_k) \tag{3.14}$$

Note that the inclusion of a local term, $\mathbf{p}^l$, essentially replaces the original $\mathbf{p}^g$ term, and thus its weighting is equal to that of the cognitive term. The $\mathbf{p}^g$ term remains in the equation, but only with a small weighting, to provide some attraction to the best overall particle.

Defining an efficient neighbourhood topology allows multiple local bests to drive the optimisation, with interconnections ensuring diversity within each neighbourhood. Without this interconnectivity, neighbourhoods would essentially remain independent sub-swarms, converging to multiple local optima.

Much work has been done in comparing these topologies, often with unconstrained mathematical benchmark problems. Kennedy and Mendes [67] recommended a von Neumann topology, where neighbours were connected above, below, and on the side of a two-dimensional lattice. Muñoz Zavala et al. [68] introduced a singly-linked ring (SLR) topology, which they showed to be more efficient for the majority of tested problems than the von Neumann topology. A particle k in the SLR has a minimum of neighbouring particles $k-2, k+1$, and can follow this neighbouring pattern to a maximum neighbourhood size of $N = n_p/2 - 1$, where n_p is the number of particles in the entire population.

More recently, Liu et al. [69] tested 198 regular topologies with 9 different numbers of particles across 90 benchmark functions. These topologies are built from graph theory, where n_p nodes are constructed and connected to r other nodes, where r is known as the degree. This creates a structured topology (as does the SLR approach), that can be easily applied to dynamic neighbourhood techniques, unlike the von Neumann approach. This is an important property, as Lie et al. note, since no specific topology outperforms all others, researchers tend towards using dynamic topologies. Furthermore, using a structured dynamic topology allows an understanding to be drawn over why one may perform better or more consistently than the other [69].

Dynamic topologies come in various forms, such as Liang and Suganthan's [70] small neighbourhood randomised approach. Here, neighbourhood sizes remain constant, but their residents are reset randomly every n iterations. Mohais et al. [71] adopted the same approach, coining the term neighbourhood re-structuring, while comparing it with random edge migration [72]; where a neighbour is detached from a random neighbourhood and re-attached to another. In either case, they reported an improved for the majority of test problems over non-random methods,

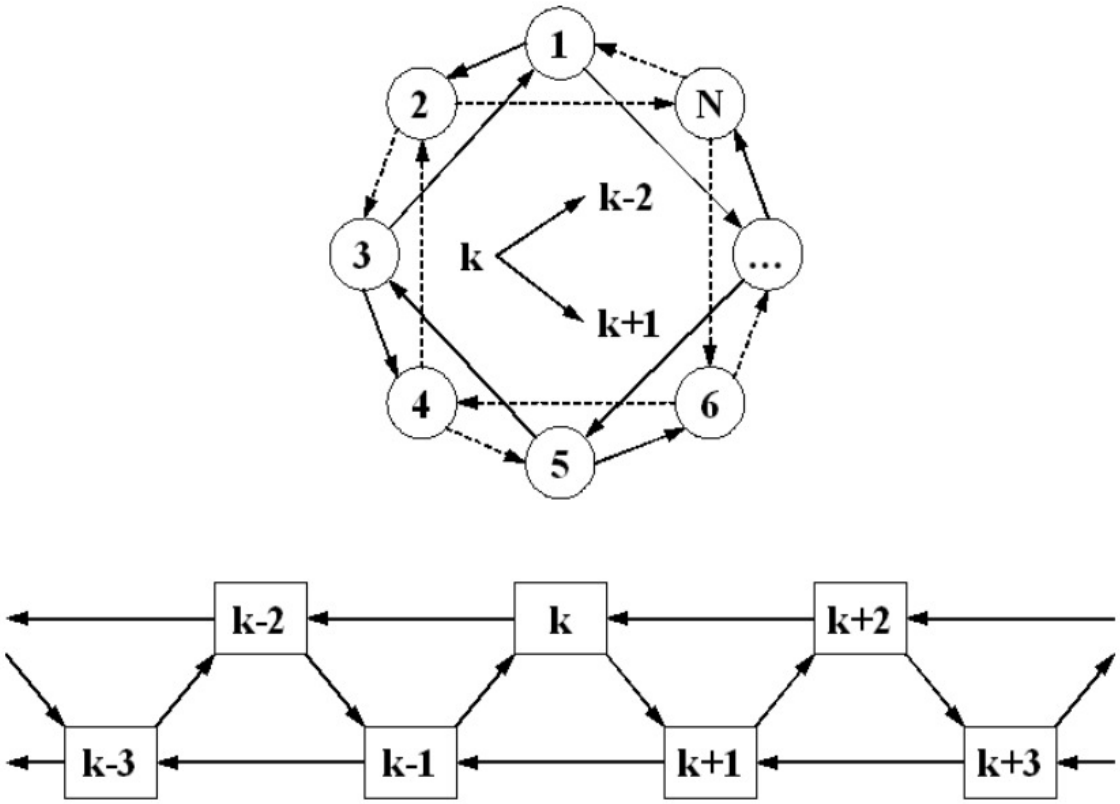


Figure 3.1: PSO singly-linked ring topology [68]

including von Neumann. Other approaches involves dynamically altering the size of a neighbourhood, by increasing the number of neighbours up to n_p until a new global best is found, before returning to the initial neighbourhood size. This method acts as star topology when the optimisation is stalling, and localised sub-swarms when improved values continue to be found. Note also for this method that the $\mathbf{p}^g$ term in equation (3.14) is removed, since during periods of prolonged stall, $\mathbf{p}^l = \mathbf{p}^g$.

In the present work, a collection of these promising topologies has been implemented, allowing problem specific parameters to be chosen. Singly-linked ring (SLR), regular graph, and randomised neighbourhoods can all be utilised in the framework, with options to re-initialise the neighbourhoods every n iterations, and dynamically adjust the neighbourhood size based on an improvement in the global best. The three neighbourhood building techniques chosen lend themselves to simple implementation of a dynamic framework, in which increasing degree r adds additional particles to the existing neighbourhood boundaries.

Mutation

To further encourage diversity in the population, mutation operators found in evolutionary algorithm literature have been employed in the present work. Sierra and Coello Coello [60] recommended utilisation of these techniques in PSO algorithms, since they have been demonstrated to work effectively in genetic algorithm applications [73]. Their approach has been implemented here, and involves the swarm being split into three equal sized sets:

- Set 1 - No mutation applied
- Set 2 - Uniform mutation
- Set 3 - Non-uniform mutation

Note the term set has been used here instead of sub-swarm, since the mutation operators only affect individual particles, not the communications between them. Set 1 acts as a regular PSO algorithm, with design variables affected only by particle, local, and/or global best position vectors. Set 2 seeks to explore the hyperspace further by utilising uniform mutation, which completely discards the current design variable value, and assigns a random value within the variables upper and lower bounds. Exploitation of the search space is achieved through non-uniform mutation in set 3, where the range of variability decreases over time. In this case, the original design variable value can be highly mutated early in the optimisation (equivalent to the uniform mutation operator), but only slightly perturbed towards the end.

The implementation of the uniform mutation operator is trivial and equivalent to that of a random variable initialisation, however the non-uniform operator requires further explanation. Since the original variable value is to be kept, a perturbation function is required that reduces with time while remaining within variables boundaries. Assuming a design variable has been selected for non-uniform mutation, this results in:

$$x_k^m = \begin{cases} x_k + \Delta(t,y),\ y = x_{k_U} - x_k, & \text{if } a > 0.5 \\ x_k + \Delta(t,y),\ y = x_k - x_{k_L}, & \text{otherwise} \end{cases} \tag{3.15}$$

Where m denotes the new mutated variable, x_{k_U} and x_{k_L} are the respective upper and lower variable boundaries, a is a random number in the range $[0,1]$, and t is the pseudo-time iteration counter. Function $\Delta(t,y)$ is then defined as follows:

$$\Delta(t,y) = y\left(1 - r^{\left(1-\frac{t}{T}\right)^b}\right) \tag{3.16}$$

Where r is another uniform random number in the range $[0,1]$, T is the maximum number of iterations, and b defines the degree of non-uniformity [73].

Table 3.1: Typical single-objective particle swarm optimisation parameters

Parameter	Value	Comments
n_p	120	Population size
c_1	1.49	Cognitive factor
c_2	1.49	Social factor ($\mathbf{p}^l$)
c_3	0.1	Social factor ($\mathbf{p}^g$)
N	$2 \rightarrow n_p$	Neighbours per particle
ω_{min}	0.1	Inertia boundaries
ω_{max}	0.8	
n_{iter}	1000	Termination criteria
n_{stall}	100	
P_M	$min(1/n_{var}, 0.05)$	Probability of mutation

Both uniform and non-uniform mutation operators work on an individual design variable per particle basis, not on an entire particle variable set at once. At each iteration, every design variable within sets 2 and 3 are assigned a random number, and any value that falls below the specified probability of mutation parameter P_M, is mutated by the operator for that set. The value of this parameter is intuitively set to $1/n_{var}$, as in Bäck [74], with an upper limit set to ensure small design spaces are not overly mutating.

3.3.3 Overall Architecture

Bringing all of the above together, a brief overview of the single-objective particle swarm optimiser (SOPSO) is presented. Optimisation parameters to be set are listed in table 3.1, along with typical values.

Particles are first initialised through one of the methods discussed previously in this section, with the default being Latin hypercube sampling. Sub-populations are then defined in terms of neighbourhoods and applied mutation operator, with the latter remaining unchanged throughout the optimisation. Each particle is then passed through the cost function to initialise the $\mathbf{p}^b$ population.

With initialisation complete the optimisation then moves into the main loop. Since cost function values have already been calculated, the first step is to select the $\mathbf{p}^g$ and $\mathbf{p}^l$ particles from the $\mathbf{p}^b$ population. Next, equation (3.14) is applied to update the particle velocity, which is then bounded by a maximum, and added to the current population positions. This maximum velocity is defined by half the distance between variable upper and lower boundaries, ensuring particles are not rapidly changing in a single iteration. Mutation methods are then applied, and particle variables are bounded with respect to their defined minimum/maximum values, with infringing particles having their velocity set to zero.

Now that particles have been updated and passed through mutation/bounding phases, they can now be fed into the cost function to obtain new fitness values. Following this, particle bests are updated if the particle has found a reduced cost function, and the $\mathbf{p}^b$ population is then used to update the $\mathbf{p}^g$ particle.

If a new global best has been found, n_{stall} is set to zero, and the inertial counter is increased. Otherwise, $\mathbf{p}^g$ remains the same, n_{stall} is incremented, and the inertial counter is decreased. During periods of stall, particle inertia is decreased based on this counter, allowing a more global search of the design space to be carried out. In a similar manner, neighbourhood populations are increased, providing deeper interconnectivity within sub-swarms. Once a new global optimal is found, particle inertia is iteratively reduced, while neighbourhoods are reset to the minimum number of particles, resulting in a localised search.

The optimisation then reverts to the beginning of the main loop, and the process continues until one of the two main termination criteria are met. The first of which being the typical maximum number of iterations, n_{iter}, and the second is determined by n_{stall}, which refers to the maximum number of iterations that the optimisation will continue for without improving upon the $\mathbf{p}^g$. Whichever of these numbers is exceeded first, the solution will be assumed to have converged and the optimisation will terminate.

3.3.4 Extension to Multi-Objective

Without the need to obtain gradient information, population based techniques can be adapted to directly handle multi-objective problems, whereas gradient based algorithms may prefer a weighted sum approach. This is a key attribute in multi-disciplinary design, as often optimality cannot be determined by a single number, and disciplines may have their own optimisation criteria, which is difficult to accurately weigh against that of another. The most common form of handling multi-objective optimisations is through the use of Pareto dominance and optimality [75]. By treating cost functions independently, a diverse set of optimal results can be output, allowing trade-offs to be visualised, along with commonalities between designs. The developed multi-objective optimiser is similar to that of OMOSPO [60], which was shown to be an improvement over PSO and GA variants for a number of test functions.

Pareto Optimality

Pareto dominance occurs if all elements within a vector are less than or equal the corresponding elements within another, and at least one of the elements is less than that of its counterpart. This is shown below:

$$\begin{aligned} &\mathbf{u} \leq \mathbf{v}, \\ &u_i < v_i, \text{ for at least one value of } i \end{aligned} \tag{3.17}$$

If both vector and element conditions in equation (3.17) are satisfied, $\mathbf{u}$ will dominate $\mathbf{v}$. In terms of optimisation, both vectors represent n cost functions values, $[f_1, f_2, ..., f_n]$, in a given multi-objective optimisation problem, deriving from a set of design variables within the design space.

In a multi-objective optimisation, cost function vectors that are not dominated by any other lie on the Pareto front, and will henceforth be referred to as non-dominated solutions. Note that this front will change during an optimisation, as new solutions are found that dominate previous Pareto front cost function vectors. Importantly, a distinction must be made between Pareto front solutions and Pareto optimality.

A solution is said to be Pareto optimal if there are no other combination of variables within the design space whose solution would dominate it. Since the global optimisation techniques employed here are numerical, finding globally optimal solutions is not guaranteed, meaning that output Pareto fronts cannot be considered Pareto optimal. This remains true regardless of population size, number of iterations, or any other convergence criteria.

Taking this idea a step further, single-objective problems have easily deduced convergence criteria, since only one optimal solution exists in a given iteration. Such optimisations can be terminated if, for example, the cost function has not improved by a specified tolerance after a certain number of iterations. Repeating these optimisations and obtaining the same result increases confidence that the global optimal has been found, but again this cannot be guaranteed. On the other hand, multi-objective problems will have a collection of optimal solutions on the Pareto front any given time during the optimisation, making determination of convergence criteria non-trivial. Presently, there are no reliable methods that can accurately quantify convergence of multi-objective optimisations, meaning that such problems are usually run for as large a population size and set number of iterations that are feasible within the users access to computational resources.

Particle Updating

In order to update particle and global best positions $(\mathbf{p}^b, \mathbf{p}^g)$ in a multi-objective framework, some alterations are required from the single-objective method. Firstly, note that there is no local term to consider, with particles being updated based on their own best position, and the

best overall. Cost function vectors must be compared in terms of Pareto dominance when updating $\mathbf{p}^b$. Thus if the particle best is dominated by the new solution, it will be replaced. In the opposite scenario $\mathbf{p}^b$ of course remains the same. However, if neither dominates the other, literature often suggests selecting one of the two particles at random to move forward as the $\mathbf{p}^b$ [57]. This unveils some potential problems in achieving a converged solution; particularly in terms of non-dominated solutions being improperly handled throughout the optimisation.

Take for example a particle best that is already a non-dominated solution, and the particle finds a solution that is not dominated by the $\mathbf{p}^b$ at iteration i. The new solution may be a non-dominated one, however this is often not the case. In fact the new solution may be dominated by various other particles in the population, but not by its own $\mathbf{p}^b$ that it is being compared to. Random selection would give both solutions an equal chance of being selected as the particle best moving forward.

Similarly, the particle best may be dominated by other members of the population, and the new solution is a non-dominated one. However, if the new solution does not specifically dominate its $\mathbf{p}^b$, it is down to chance whether or not it will be saved in memory.

With respect to the two stated potential scenarios, this could result in a) the non-dominated $\mathbf{p}^b$ is replaced by what is, by definition, an objectively worse solution, or b) a non-dominated solution being discarded if the original $\mathbf{p}^b$ is selected.

If scenario a) is not sought to be rectified, allowing non-dominated positions to be replaced in the $\mathbf{p}^b$ population by dominated ones, steps must be taken to ensure that all non-dominated solutions are retained. A similar scenario also occurs when two non-dominated solutions are competing for the $\mathbf{p}^b$, meaning that the newly found position could be discarded. Therefore, when determining non-dominated solutions from the current particle positions and their respective best positions, they are placed into an external archive [76]. This ensures that all non-dominated solutions found are kept, regardless of the $\mathbf{p}^b$ updating procedure chosen.

Scenario b) is to be avoided at all costs, since non-dominated solutions can be missed entirely, hindering the optimisers convergence. Advice is therefore taken from Deb et al. and Li [77, 78], to combine the current particle positions with the particle best population, and determine non-dominated solutions from a population of $2n_p$. Only after this is the $\mathbf{p}^b$ population updated, which also provides a solution to scenario a), since non-dominated particles have already been identified and thus do not need to be retained in $\mathbf{p}^b$.

Global Best

The selection of global best plays a pivotal role in the algorithms efficiency. Along with the best position that a given particle has experienced in terms of cost function, the global best defines which design variable sets that the swarm will tend towards during the current iteration.

Since optimality is now defined by a front of particles, rather than a single particle, steps must be taken to distinguish the most promising members of the front. There are two main ways to do this: number of particles dominated, and crowding distance. The former can be easily computed when finding non-dominated particles by reversing the Pareto dominance comparison, and counting how many particles are dominated by the Pareto particle. Crowding distance is determining how close a non-dominated particle is to its neighbours on the Pareto front, based on the normalised cost function values, denoted as $\hat{f}$. This is defined below in equation 3.18:

$$\sum_{i=1}^{n} \hat{f}_{i,j-1} - \hat{f}_{i,j+1} \tag{3.18}$$

Where n is the number of cost functions, and j represents neighbouring particles in a sorted array with respect to the non-dimensionalised cost function i.

Both of these methods can be used to select the $\mathbf{p}^g$ particle, by ranking $\mathbf{p}^b$ particles in terms of dominance or crowding distance. Particles with the highest dominating rank are often the most crowded, and selecting them as the global best can localise the search to a small section of the Pareto front. On the other hand, selecting Pareto front particles with the lowest dominating rank (those that lie on the extremities in each dimension), will by definition have infinite crowding distance. Searching around these outer regions ensure the optimisation remains sufficiently diversified, but may lack the convergence characteristics of the former method. Thus in this work, both approaches have been implemented, with either half of the population consistently selecting a high or low dominating global best throughout the optimisation.

In either case, the $\mathbf{p}^g$ selection for a given iteration should not be defined purely by the particle that dominates the most or least number of particles, or that which has the best crowding distance characteristics. Thus, proportionate fitness selection has been implemented, with the best performing particles in the Pareto front having a higher chance of being selected. This probability for each particle to be select as the global best is defined as follows:

$$p_k = \frac{n_k^d}{\sum_{j=1}^{n_{PF}} n_j^d} \tag{3.19}$$

Where n^d is the number of particles dominated, and n_{PF} is the total number of particles in the

Table 3.2: Typical multi-objective particle swarm optimisation parameters

Parameter	Value
n_p	120
c_1	1.49
c_2	1.49
ω	0.3
n_{iter}	1000
n_{PF}	100
P_M	$min(1/n_{var}, 0.05)$

Pareto front. Note that this method cannot be used when crowding distance is the deciding fitness characteristic, since particles at extremities would have infinite values. Thus, number of particles dominated is the default metric for global best selection, with crowding distance utilised to limit the Pareto front population. Once again, this attempts to increase diversity within the population, ensuring that all non-dominated particles have a chance to be selected as the $\mathbf{p}^g$.

3.3.5 Overall Architecture

Much of the multi-objective particle swarm optimiser (MOPSO) process is akin to that of the single-objective variant, whose architecture is described in section 3.3.3. To avoid covering these details again, only the alterations required to allow multi-objective optimisations to be carried out are discussed here. MOPSO parameters to be set are listed in table 3.2, along with typical values.

Note the lack of stall counter and therefore a constantly defined particle inertia, ω. This is due to the Pareto front and global best particle being ever changing throughout a given optimisation, requiring convergence to be defined primarily through number of iterations. Some research views convergence across the entire Pareto front, by terminating the optimisation after a set number of iterations have elapsed with the Pareto front maintaining it's population. However, testing such methods in this work has produced unreliable and inconsistent results, thus preferring maximum number of iterations to define optimisation terminal criteria.

Particles are initialised in the same manner as SOPSO, and updating particle best positions throughout the optimisation is similar, with the requirement of using Pareto dominance as described in section 3.3.4. Once particle bests are updated, they are combined with the archived Pareto front population, to re-define the front and determine particle dominance characteristics in the process. If the Pareto front population exceeds the user defined limit, crowding distance is computed within the front, with the most crowded particles being annexed.

Global bests are then selected using the sub-swarm technique discussed previously, using the roulette wheel probability equation (3.19) to select Pareto front particles with both high and low domination characteristics.

With individual best, Pareto front, and global best particles determined, particle positions can be updated using equation (3.13) for the next iteration, with the loop beginning again from cost function computation.

3.4 Constraint Handling

To this point, the described global algorithms are purely objective function based; attempting to minimise cost functions within variable boundaries by any means necessary. As discussed in local optimisation methods, this is not representative of real-world problems, and defining optimality at all costs without realistic constraints will produce infeasible designs.

Here, particular attention is paid to the methods employed to deal with such constraints. These methods must strike a fine balance between finding non-violating designs, and ensuring they do not result in a premature convergence of the optimisation. Design spaces which are highly constrained may only contain small areas of feasibility, and the prioritisation of constraints over objective functions can lead to the first non-violating design having a profound impact on the remainder of the optimisation. Over-reliance on this initial design will result in a deep search of the surrounding area, potentially missing areas of feasibility, and therefore optimality, in the remaining hyperspace.

3.4.1 Methods for Global Optimisers

Constraint handling is therefore the most crucial aspect of applying a heuristic algorithm to a real-world problem. Evolutionary algorithms (EA) prove powerful tools when finding the minimum in mathematical functions, constrained only by the upper and lower boundaries of the design variables. Such functions are used predominantly as benchmarks to compare EAs, through metrics such as final objective function values, or number of objective function evaluations to reach convergence. However, as mentioned in section 3.2.3, engineering applications of optimisation must adhere to constraints placed on the design variables and metrics related to the specific problem. In gradient-based frameworks, these constraints can be dealt with directly, by computing their respective gradients, and employing an approach such as SQP to solve the system of equations. In gradient-free optimisation, no such information is available, leading to varying methods being developed for heuristic constraint handling.

Generally in research, constraint handling in numerical optimisation is dealt with in two ways: adding a penalty value based on constraint violations to the objective function, or utilising both separately during the various selection processes. Variations of each method have been implemented in the present work, and will be described here. Since a single value is required from an arbitrary number of violation functions, each function is defined as a non-dimensional inequality, scaled by a typical value. To ensure any single violation results in a positive overall value, negative inequalities are set to zero before summation.

Penalty Methods

Adding to the objective function by a constraint derived value is more commonly known as a penalty method. This transforms a constrained optimisation into an unconstrained one, by combining cost and constraint into a single function:

$$f'(\mathbf{x}) = f(\mathbf{x}) + \phi(g(\mathbf{x}), h(\mathbf{x}), R) \tag{3.20}$$

Where f' is the new function to be minimised, and ϕ is a violation function calculated from:

$$\phi(g(\mathbf{x}), h(\mathbf{x}), R) = \sum_{j=1}^{p} r_j \cdot max(0, g_j(\mathbf{x}))^q + \sum_{k=1}^{m} c_k \cdot h_k(\mathbf{x})^q \tag{3.21}$$

With r_j and c_k being positive penalty factor constants, q is a positive number, and the remaining terms consistent with that of equation (3.10). Note that there are various ways of calculating ϕ, with one notable method equating it to the overall maximum constraint violation value, which has been used extensively in engineering applications [79, 80]. Nonetheless, care must be taken in how the constant penalty factors are chosen. As discussed by Runarsson and Yao [81], setting values too small may result in infeasible solutions with low cost function values obtaining lower ϕ values than feasible solutions. On the other hand, high values may increase the chance of finding feasible solutions, but also increases the likelihood that they will be of poor quality. This is due to the fact that even close to feasible solutions will receive drastically higher cost function values than entirely feasible ones, meaning potentially feasible spaces will not be explored due the "walling effect" surrounding them.

An extreme case of this effect is brought forward by the so called "death penalty" method. Here, infeasible solutions are given the highest possible penalty value, or are discarded from the optimisation entirely. Since this adds significant noise to the resulting cost function, the death penalty method should be avoided in the majority of circumstances. However, there are cases where it is necessary to use such an extreme method, such as a variable set that results in an error within the cost function.

Effectively managing these multiple constraints; ensuring one does not dominate the rest, or

close to feasible solutions become hard to differentiate from entirely feasible ones, is clearly a problem dependent, and indeed constraint dependant task [81]. In this work, advice is taken from Arora [48] to scale constraints before passing them to the penalty function, removing the need for individual penalty factors, and thus equation (3.21) becomes:

$$\phi(g(\mathbf{x}),h(\mathbf{x}),R) = R\left\{\sum_{j=1}^{p} max(0,\hat{g}_j)^q + \sum_{k=1}^{m} \hat{h}_k^q\right\} \tag{3.22}$$

Where R is a constant, $\hat{g}_j$ and $\hat{h}_k$ are the normalised constraint values:

$$\hat{g}_j = \frac{g_j(\mathbf{x})}{g_j^e} - 1,\ \hat{h}_k = \frac{h_k(\mathbf{x})}{h_k^e} - 1 \tag{3.23}$$

With g_j^e and h_k^e being some expected value of the constraint. These values require the same care in establishing as the penalty factors, but have a more physical interpretation. Maximum, minimum, or averaged constraint boundary values can be used, as well as user specified expected values. The constant R is assumed to be greater than zero, and serves as a single penalty factor to ensure that infeasible cost function values ($\phi > 0$), will always be greater than feasible ones. The selection of this parameter will therefore depend on the magnitude of the largest feasible cost function.

Adaptive Methods

In the penalty method, the resulting violation value must be scaled by a suitable magnitude so as to ensure violating particles do not maintain low enough cost function values to be placed on a Pareto front with non-violating particles. Correct implementation of such a method essentially results in a single-objective optimisation, regardless of the actual number of cost functions, where the violation is sought to be minimised. Only once all constraints have been satisfied will the optimisation begin searching for improved cost function values. In tightly constrained design spaces, this can easily lead to the optimisation being driven by the first non-violating design that is found.

Utilising constraint functions independently to the objective function has been researched more intensely in recent years, avoiding the induced noise and non-linearity that comes with combining them. Various methods have been explored for finding ideal solutions through separate cost and constraint functions, with no one method standing out as superior over the rest. The most promising of these techniques tend to employ adaptive parameters in one sense or another, and two such methods that have been experimented with in the present work will be discussed here.

One technique that has received increased attention is the ε-constrained method, introduced by Takahama and Sakai [82]. Here the constrained optimisation replaces the usual cost function

comparison between solutions with an ε level comparison, which for $\varepsilon > 0$ takes the forms:

$$(f_1, \phi_1) \leq_\varepsilon (f_2, \phi_2) \Leftrightarrow \begin{cases} f_1 \leq f_2, & \text{if } \phi_1, \phi_2 \leq \varepsilon \\ f_1 \leq f_2, & \text{if } \phi_1 = \phi_2 \\ \phi_1 < \phi_2, & \text{otherwise} \end{cases} \tag{3.24}$$

The value of ε therefore dictates whether f_1 is an improvement over f_2 based on the standard cost function comparison, or by a comparison of constraint function values. Considering the extreme forms of ε, it can be seen that for $\varepsilon = \infty$, equation (3.24) is equivalent to an unconstrained optimisation, whereas $\varepsilon = 0$ ensures the lower constraint violating solution is always chosen, regardless of cost function value. The latter setting is equivalent to the penalty method, seeking to minimise the violation function until a non-violating particle is found. However, treating the two functions separately means that ε can be dynamically reduced throughout the optimisation. This allows violating particles below the constraint tolerance to be selected and thus improved upon in the early stages, improving diversity within the population.

Clearly then, the selection of ε is of prime importance, to ensure feasible solutions are found, but are not suboptimal in nature due to infeasible solutions not being properly explored. The value of ε is defined by the violation functions of the initial population, as shown in equation (3.25), and is dynamically reduced until some final iteration t_f, after which violating particles are eliminated from the selection process and Pareto front [82]:

$$\varepsilon_0 = \frac{1}{2}\left(\frac{1}{n}\sum_{j=1}^{n}\phi_j + min(\phi)\right) \tag{3.25}$$

$$\varepsilon_t = \begin{cases} \varepsilon_0 e^{(-\beta t)}, & \text{if } t \leq t_f \\ 0, & \text{if } t > t_f \end{cases} \tag{3.26}$$

Where β is given by introduced a user-specified final ε level value, at some time t_f, after which $\varepsilon = 0$:

$$\beta = \frac{log(\varepsilon_0/\varepsilon_f)}{t_f} \tag{3.27}$$

The specification of t_f and $\varepsilon(t_f)$ is all that now must be discussed. In this work, t_f is generally set to $T/2$, allowing half of the simulation to search through infeasible values with decreasing violation functions, and the latter half focussing solely in obtaining feasible outputs. In order to avoid a sharp discontinuity when t passes through t_f, the final ε level, before being set to zero

for the remainder of the optimisation, is usually $\varepsilon(t_f) \leq 10^{-3}$, depending on the nature of the problem.

Adaptive methods are a clear improvement over penalty approaches, selecting particles based on cost and violation functions independently, rather than a combination of the two which is dominated by the latter. Furthermore, the dynamic reduction of constraint tolerance enables violating particles to be chosen in the various selection processes based on promising cost function values, promoting further design space regions to be searched for feasibility. Nonetheless penalty approaches remain more popular in engineering applications, although this is most likely due to its ease of implementation, rather than improvement in results over adaptive methods.

3.5 Verification

With large design space, highly constrained problems to be solved, it is of paramount importance that the optimiser has been implemented correctly. Furthermore, it must be capable of consistently finding optimal solutions for such problems. Common mathematical optimisation test functions, with known optimal values, have therefore been used to test the developed frameworks performance.

3.5.1 Unconstrained Single-Objective

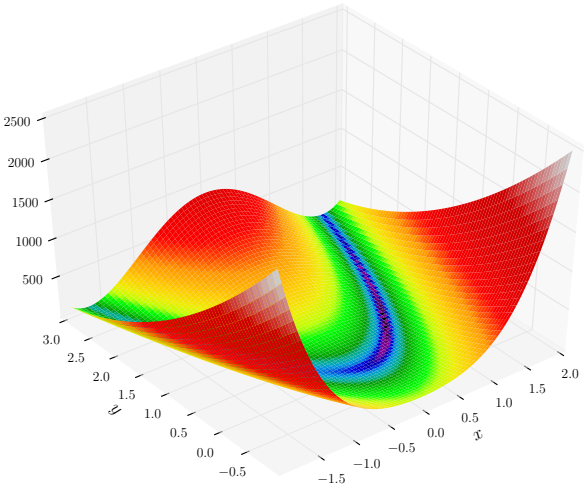
As a starting point, the single-objective PSO is tested on unconstrained problems, where the cost function has to be minimised without any additional requirements. This is the simplest case, and acts as a first check to ensure the algorithm has been correctly implemented. In order to provide a comparison, both the particle swarm and genetic algorithms from the MATLAB Optimization Toolbox [83] have been used to optimise the same mathematical functions. This initial verification is important, since the MATLAB PSO is not capable of handling constrained problems, meaning they can only be compared for unconstrained design spaces. To ensure a fair comparison, a number of runs were carried out with each of the three functions, with each run number employing a consistent random seed value. This means that every run of a given optimisation function is unique, but that the same order of random numbers is observed across the three functions at the same run number. Further properties had to be set commonly among the algorithms, as shown in table 3.1.

The mathematical problems to be optimised were the Rosenbrock [84], Rastrigin [85] and Beale [86] functions, each with variable boundaries and optimal values shown in table 3.3. These functions, the results in terms of best cost function value observed and number of iterations before convergence for MATLAB's GA and PSO functions, along with the described

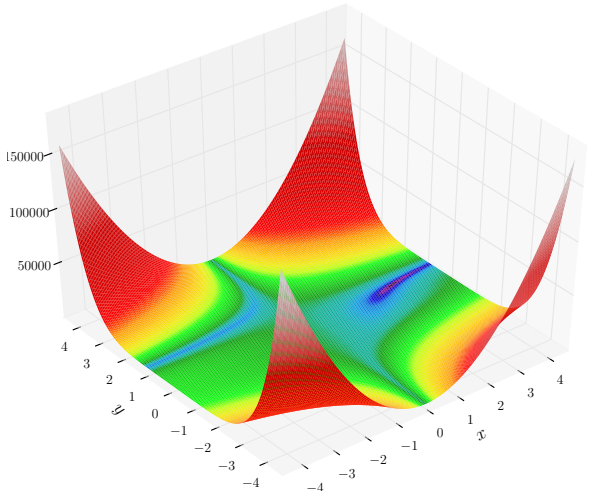
Table 3.3: Single-objective unconstrained and constrained mathematical function properties

Function	Lower Bounds	Upper Bounds	$f(\mathbf{x}_{opt})$
Rosenbrock	[-100 -100]	[100 100]	0
Beale	[-4.5 -4.5]	[4.5 4.5]	0
Rastrigin	[-5.12 -5.12]	[5.12 5.12]	0
Rosenbrock Constrained	[-1.5 -0.5]	[1.5 2.5]	0
Mishra's Bird	[-10 -6.5]	[0 0]	-106.765
Modified Townsend	[-2.25 -2.5]	[2.5 1.75]	-2.024

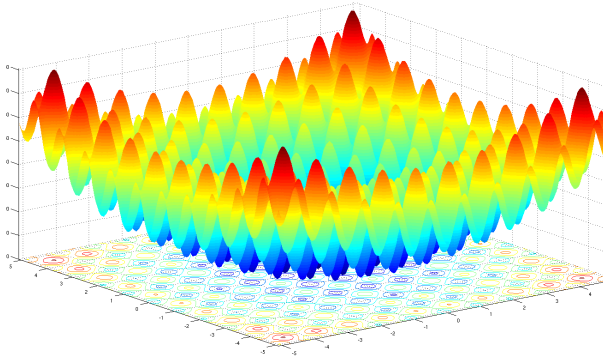
single-linked ring PSO are shown in table 3.4.



(a) Rosenbrock's function [84]



(b) Beale's function [86]



(c) Rastrigin's function [85]

Figure 3.2: Mathematical test functions used for single-objective unconstrained optimisation benchmarking

For every sample of the three mathematical functions, the SOPSO finds the global optimal, giving it a 100% success rate for each function. The MATLAB PSO variant also performs very well, achieving 100% success rate for both the Rastrigin and Beale functions. Only one sample of the Rosenbrock function does not converge on the global optimal, although it is assumed with

Table 3.4: Objective mean and standard deviation of optimisation methods across mathematical test functions

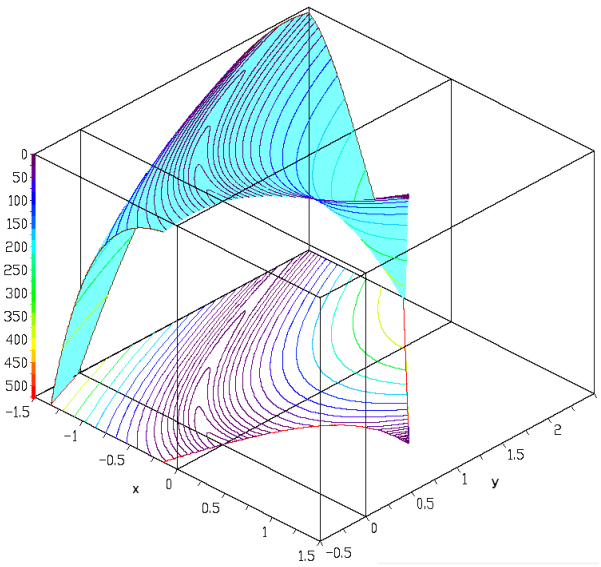
	PSO		GA		MPSO	
Function	$\mu(f)$	$\sigma(f)$	$\mu(f)$	$\sigma(f)$	$\mu(f)$	$\sigma(f)$
Beale	0	0	1.300×10^{-10}	2.956×10^{-10}	0	0
Rosenbrock	0	0	1.664	10.607	1.124×10^{-20}	1.124×10^{-19}
Rastrigin	0	0	0.0298	0.1706	0	0
Rosenbrock Constrained	0.939	0.238	0.739	0.440	N/A	
Mishra's Bird	-106.765	4.709×10^{-14}	-90.662	38.535	N/A	
Townsend	-1.8671	0.1816	-2.020	0.0364	N/A	

an infinitely large sample that both PSO variants would perform similarly. The GA on the other hand does not reach the global optimal for any sample number of the Rosenbrock function, with some runs converging on local optima well above one. Performance is seen to improve for the Rastrigin function, and again for the Beale problem. In both cases, the GA once again does not reach the absolute global optimal for any sample, however the mean and variance of the resulting cost functions are considerably lower in terms of order of magnitude. In terms of iterations to achieve convergence, the SOPSO and MATLAB variant are relatively consistent, with the former taking slightly longer. On the other hand, the genetic algorithm iteration count is highly varied, with some samples prematurely converging, while others cannot converge before the maximum number of iterations is reached. Clearly then, PSO methods are particularly suited to this type of optimisation. Furthermore, direct comparison to an industry standard PSO variant verifies the correct implementation of the SOPSO; which demonstrates slightly improved performance in terms of objective function, with a small increase in required iterations.

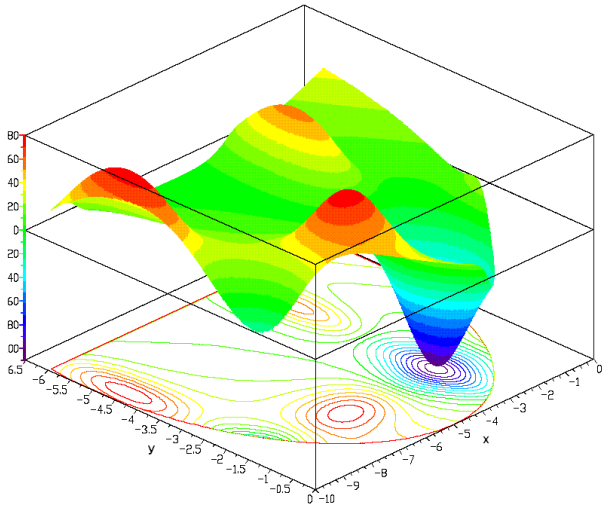
3.5.2 Constrained Single-Objective

With the SOPSO performing well against an established variant, constraints can now be added to the optimisation process, reducing the feasible design space and thus increasing problem complexity. As previously mentioned, the MATLAB PSO can only handle unconstrained problems, meaning only comparisons between the SOPSO and MATLAB GA will be made going forward. The chosen functions are a constrained variant of the Rosenbrock, in which the function is constrained to a cubic and line; Mishra's bird function [87]; and a modified version of Townsend's function [88]. These functions provide a variety of complex constrained problems to be optimised, with properties once again shown in table 3.3, along with function visualisations in figure 3.3.

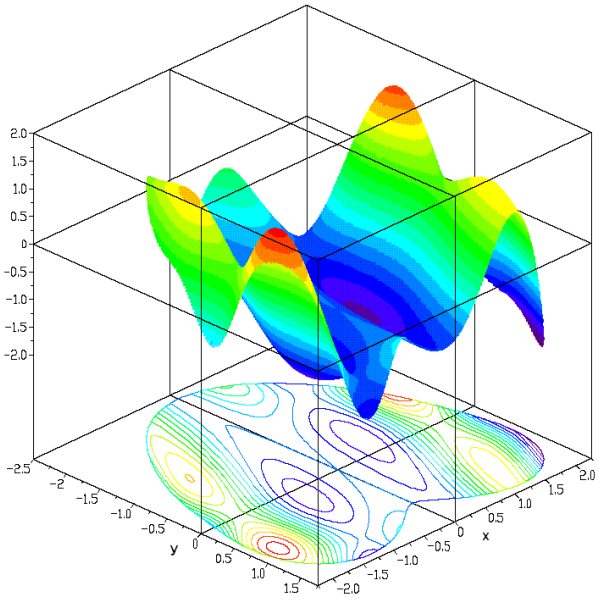
Results for both optimisation methods are again shown in table 3.4. For the Mishra's bird problem, the SOPSO found the global optimal in all cases, while the GA only managed to do so in 82% of samples. The results are very different for the constrained Modified Townsend function however, with the GA converging on the global optimal 98% of the time, while the PSO only did



(a) Rosenbrock's function constrained within a cubic and a line [89]



(b) Mishra's Bird function [87]



(c) Townsend's function [88]

Figure 3.3: Mathematical test functions used for single-objective constrained optimisation benchmarking

so in 20% of cases. Both algorithms struggled with the constrained Rosenbrock variant, with GA and PSO finding the global optimal in only 26% and 8% of cases respectively. Note however that across all runs, the mean value of the PSO solutions is some 30% lower than that of the mean GA solutions, showing that it avoids some of the higher cost function traps that the GA falls into.

These results are appear inconclusive, with neither algorithm performing consistently better than the other. While this is not a particularly satisfying conclusion, it is one that has been widely reported throughout research, showing that the choice of optimiser is problem dependent [56].

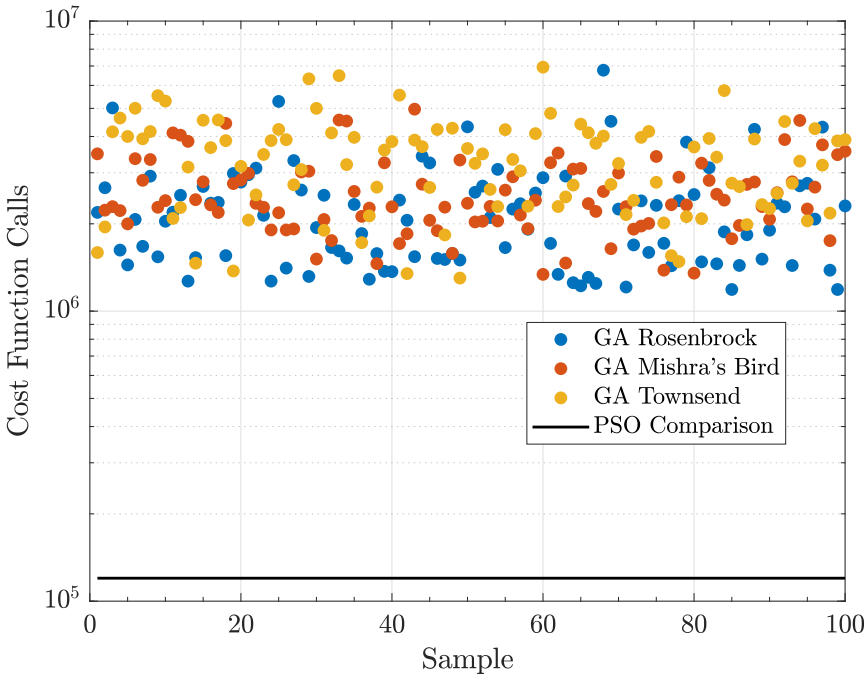


Figure 3.4: Single-objective GA and PSO number of cost function calls comparison

However analysis of a further key metric, number of cost function calls, produces a clear incentive for employing the SOPSO algorithm over a GA within design optimisation work. Figure 3.4 illustrates this, with cost function count for the three constrained optimisation problems shown for the GA, along with a PSO comparison. The cost function count for PSO algorithms scale with population size and iterations only, thus remaining constant regardless of problem or sample. On the other hand, the GA cost function count is variable when constraints are involved, and even in the most efficient case across the three problems, still calls the cost function over 10 times that of the PSO. This is not particularly problematic in the optimisation of mathematical functions, since the time to complete a cost function call is negligible. For intense cost functions however, which will be employed throughout this work, selecting an optimiser that increases computational load by a factor of 10, without producing a significant gain in output results, is clearly counter intuitive.

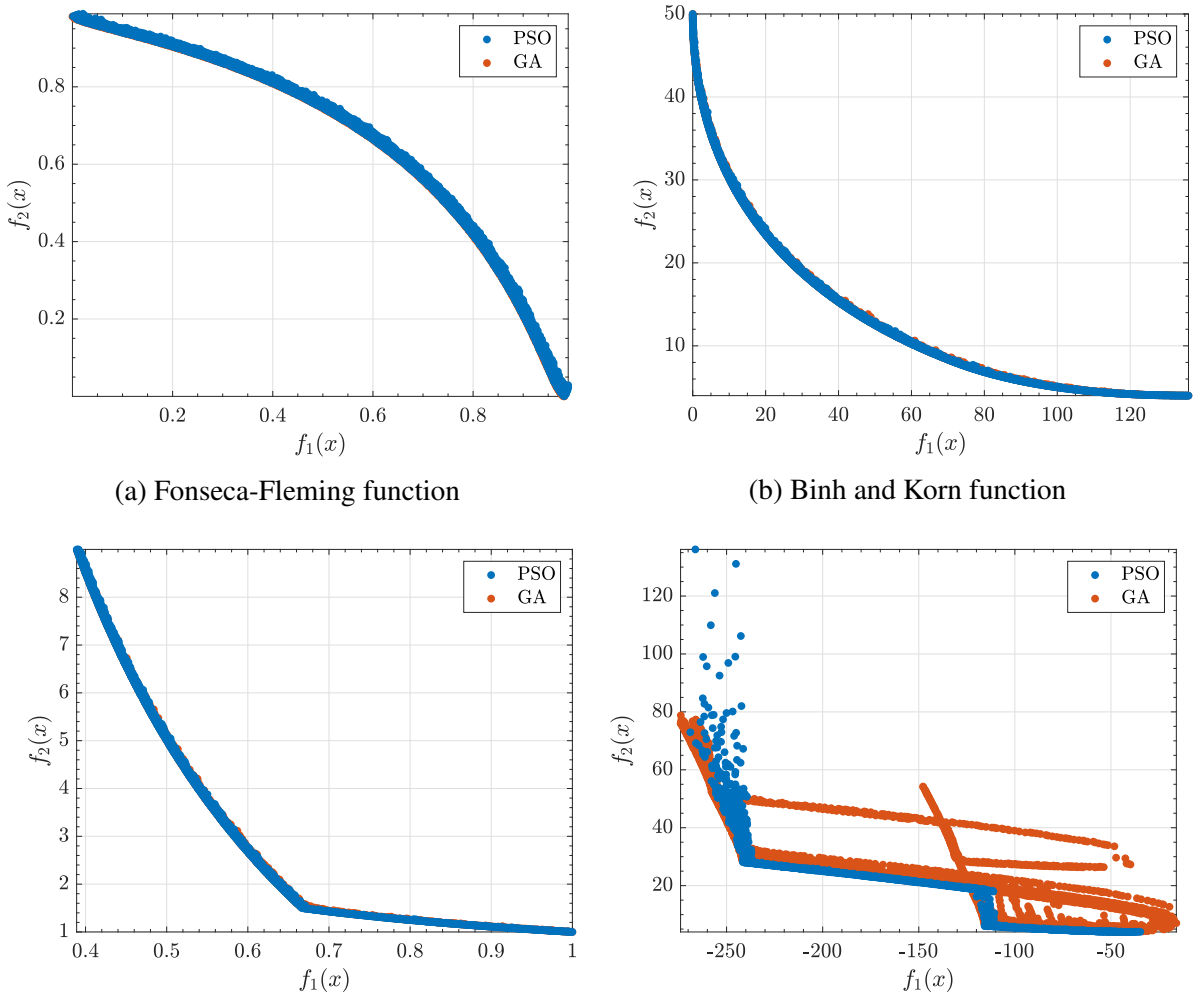
3.5.3 Multi-Objective

Having verified the effectiveness of the SOPSO against industry standard variants for a variety of unconstrained and constrained mathematical functions, the same must be done for the developed multi-objective PSO. This is critical as the single and multi-objective algorithms employ very different operations, meaning they must be verified separately. In multiple cost function cases, the optimiser must be capable of finding an array of optimal solutions, defined on a trade-off basis as a Pareto front. These cost functions remain independent, and thus finding solutions

at the extremes of the front may be just as important as finding those at the Pareto midpoint.

Once more the developed optimiser, a constrained OMOPSO variant, is tested against a MATLAB Optimization Toolbox function; in this case the multi-objective genetic algorithm (MOGA). The mathematical functions used to compare the two algorithms are the Fonseca-Fleming, Binh and Korn, Osyczka and Kundu, and Constr-Ex problems. This set benchmark functions consist of constrained and unconstrained optimisation problems, along with convex, concave, and discontinuous Pareto fronts.

Consistent with prior verification cases, each problem is run through the optimisation methods for 100 samples, in order to highlight their consistency in finding global optima. The amalgamated Pareto fronts for both MOPSO and MOGA are shown in figure 3.5. Note that in all cases, PSO Pareto fronts are plotted on top of GA results, since they are the most critical in this work.



(a) Fonseca-Fleming function

(b) Binh and Korn function

(c) Constr-Ex problem

(d) Osyczka and Kundu function

Figure 3.5: Multi-objective optimisation benchmarking Pareto front results

Across both methods, the optimal Pareto fronts for cases (a)-(c) are consistently replicated, demonstrating their applicability to both unconstrained (Fonseca-Fleming) and constrained multi-objective problems (Binh and Korn, Constr-Ex). Note that unlike the single-objective variant, MATLAB's multi-objective GA function count is consistent with that of MOPSO's. Case (d), Osyczka and Kundu's function, illustrates a tougher problem, with both optimisers showing inconsistencies across samples. Interestingly, the particular discrepancies are not consistent across methods, which may provide some key insights into their applicability to the optimisation problems within this research.

Taking a closer look at case (d), an analogous description would be that the PSO provides higher accuracy in locating the true Pareto front, whereas on the rarer occasions that the GA shows high accuracy, it also shows higher precision. For all 100 samples, MOPSO characterises the large rate of change in $f_1(x)$ portions of the Pareto front reliably, while struggling to precisely identify analogous regions for $f_2(x)$. Nonetheless, the optimiser's ability to avoid locally optimal regions, and consistently characterise the overall Pareto front, demonstrates it's practicality within highly constrained, noisy optimisation problems. Conversely, a number of sub-optimal Pareto front segments can be observed for the GA, some of which completely miss large regions of the true front. The GA does show higher precision in representing the high rate of change $f_2(x)$ regions, but this is only on the occasions in which it identifies the existence of such regions in the first place. This can visualised around the $f_1(x) \approx -100$ mark, where a number of sub-optimal slices can be seen to rise in $f_2(x)$, unable to reliably identify the true optima. The same behaviour can be seen across the board, with levels of particles rising above and inside the true Pareto front.

While neither optimiser's behaviour is ideal in the Osyczka and Kundu case, MOPSO demonstrates much more favourable characteristics in terms of design optimisation. Being able to consistently determine the global front, avoiding local pitfalls, while lacking some precision in finding absolute optima, is much preferred in MDO than large mischaracterisations of the Pareto front itself. Some context is required to drive this point home, which has been discussed previously in this chapter, but its importance validates its repetition here. This is that large, exploratory design optimisations are not all-encompassing in terms of disciplines analysed, or the level of analysis fidelity, and therefore should not seek to find absolute optima at the cost of overall convergence. Focus on such designs is redundant, since the cost function, and therefore Pareto front, will change with increasing fidelity of analysis, and thus improved representation of real-world behaviour. It is much more efficient to ensure that the Pareto front is well-known for the current level of analysis fidelity, without local or unreliable definitions of optimality, allowing close to optimal designs to be analysed at higher fidelity and across increasing number of disciplines.

3.6 Conclusions

Methods used for optimisation throughout this research have been defined in detail, from local single-objective to multi-objective heuristic techniques. The rationale behind these methods have been discussed, along with their relative upsides and drawbacks for the optimisation problems tackled in this research.

Highly efficient local optimisation methods have been implemented, employing second-order estimations to avoid a drastic increase in cost function calls. Constraints are tackled utilising the well known SQP method, which allows feasibility to be determined alongside optimality. Nonetheless, the relative pitfalls of gradient based techniques for global design optimisation have been highlighted, along with general downfalls of assuming starting points without carrying out prior optimisation work.

To conduct the large design space, highly constrained optimisation problems in this research, techniques that actively search the hyperspace in a given iteration are a necessity, if globally optimal results are to be found. Particle swarm optimisation was selected due to its superior performance against other methods for continuous design spaces. Both single and multi-objective variants were developed, employing a variety of techniques found throughout PSO and EA literature. These bespoke versions of a well-known heuristic optimiser promote diversity within the overall process, attempting to maximise the probability of finding the various niches in both feasibility and optimality within a given design space.

Constraint handling has been highlighted as one of the most important, and often overlooked, attributes of a design optimisation approach. This concept of feasibility must be treated as importantly as optimality, in order to avoid poor convergence characteristics. This is intuitively dealt with using gradients of constraint in local optimisation, but some ambiguity exists in how they should be dealt with in gradient free, global methods. Two approaches have been implemented in the current work, the first combining cost and constraints values in a penalty approach, with the second using an adaptive tolerance to allow independent treatment of cost and violation functions. The latter method also allows a mixing of violating and non-violating designs to exist on the Pareto front, providing increased scope for promising but violating areas of the design space to be searched for regions of feasibility.

The developed optimisers have been verified in their implementation and general fitness for purpose, with comparison to industry standard genetic algorithms and particle swarm optimisers across single-objective, multi-objective, unconstrained and constrained benchmark mathematical functions. This comprehensive bank of problems demonstrate the capability of the developed algorithms in handling a versatile set of problems. Moreover, they consistently outperformed

comparative optimisers in terms of finding global optima, reducing computational load, and correctly identifying Pareto fronts. The performance of both SOPSO and MOPSO algorithms lend themselves to the large design space, noisy cost function space problems that are expected is this research. With these optimisers validated, such problems can now be carried out.

Chapter 4

Low-Fidelity Global Aerofoil Optimisation

4.1 Introduction

Throughout this research, one of the most challenging problems has been the optimisation of aerofoil sections. Aerofoils are one of the key components of a vehicles aerodynamic efficiency, and thus a vast array of research has been carried out on their shape generation and optimisation. A number of generative methods have been implemented and compared in the current framework. Regardless of the method employed, appropriate control over generated shapes is required to ensure a feasible design is output. Prior research often applies these constraints on the design space, reducing the variation of potential output shapes. However, for a truly global design optimisation to be carried out, not only is a global optimiser required, but a global design space is also a necessity. This may result in numerous regions of infeasibility within the design space, but these should be characterised mathematically, rather than placing overly stringent boundaries on the hyperspace. Furthermore, this characterisation should be computed through continuous functions rather than a binary designation, to allow the optimiser to iteratively improve upon designs with poor and unwanted qualities.

With a number of shape generation methods available; the need for robust constraints in defining a feasible aerofoil; and the potential for local minima to be seen as globally optimal due to the inaccuracies of low-fidelity methods, a number of studies were carried out to provide insight and higher quality global design optimisation results.

As a first step, the two primary methods for aerofoil generation are tested through a variety of shape optimisations; where existing aerofoils are used a target to be matched as closely as possible. Ensuring the design space is broad enough to create a variety of existing shapes is an important step, and validates that globally optimal results can be produced from the framework. Analysing and attempting to model these shapes also allowed robust geometric constraints to developed, ensuring that the bank of existing, and therefore feasible aerofoils, are not seen as

infeasible by the optimisation framework.

Large design space aerodynamic optimisations are then carried out, using low-fidelity analysis methods. The design space boundaries, along with constraints derived from the shape optimisations are utilised in a performance driven optimisation. The optimisation framework as a whole is first tested by attempting to improve upon a baseline aerofoil, using its geometric and performance properties to constrain the various designs produced throughout. This ensures that comparisons are not made solely on the objective function, and resulting aerofoils maintain similar properties to that of the baseline. Following this, the baseline design and relevant constraints are removed, and a further optimisation carried out, allowing a broader scope of designs to be considered optimal.

4.2 Background

Aerofoil optimisation has been studied since the 1970s, with Hicks et al. [94] being the earliest adopters of such methods to aerofoil design. Despite such early application, aerofoil optimisation has remained popular since, showing the intricacies and challenged faced for a two-dimensional problem. Aerofoils present a standardised case for aerodynamic design optimisation, and thus provide an excellent starting point when three-dimensional optimisation is the end goal. Furthermore, the high degree of sensitivity between aerofoil shape and performance, a key driver in the continuation of such research, allows the framework as a whole to be tested for robustness. In order to achieve optimal, reliable results, the chosen optimisation, analysis, geometry definition, and constraint methodologies need a considerable amount of attention, to ensure that collectively, they provide enough information to properly distinguish optimality from infeasibility.

These topics require discussion to understand the broad variety of research that has been carried out over the years, and in particular the key merits and downfalls of the chosen methods. This will be done in the proceeding section, leading to the justification of the decisions made for the present research, along with detailed explanation of their implementation.

4.2.1 Optimisation and Analysis

Choice of analysis method is often driven by the choice of optimisation method, or vice versa, since high-fidelity analysis does not scale well with number of cost functions required for convergence. Therefore, low-fidelity methods are often paired with global optimisation techniques, whereas high-fidelity analysis usually requires gradient based methods. Moreover, trade-offs can also be expected in terms of design space and global convergence, with low-fidelity approaches

searching for the absolute minimum of a large design space, at the cost of analysis inaccuracies; whereas high-fidelity frameworks employ a much more limited design space, often seeking to improve upon an existing aerofoil.

A number of methods have been employed through the years to bridge the gap between accurate analysis and optimisation scope, which is of particular interest in this research. While global, exploratory design optimisations are the main focus, aerofoils provide an ideal opportunity to experiment with more robust analysis methods, due to their two-dimensional nature and consistent non-dimensionalised chord. Furthermore, lessons learned from these 2D aerodynamic optimisations can be carried forward to more complex 3D problems, which would otherwise exacerbate the flaws found in aerofoil design spaces.

Multi-point Optimisation and Uncertainty

Drela [95] first reported on the shortcomings of aerofoil optimisation, pointing out the exploitation of flow properties at the smallest scale, requiring multi-point optimisation to be carried out. It is suggested the number of points should be increased with the number of design variables, and a solution involving a single-objective weighted sum approach is presented.

Padula and Li [96] discussed options for robust aerofoil design or optimisation under uncertainty, again highlighting that traditional techniques have a tendancy to over-optimise in single-point optimisations. This results in sub-optimal off-design characteristics, often poorer than that of the baseline. Their solution invoked a robust multi-point optimisation, in which randomly perturbed Mach number points in the range $0.7-0.8$ were utilised to form a linear subproblem that minimises the objective function while only slightly altering the design variables. Furthermore, a trust region is employed in an attempt to reduce the number of objective function calls.

Liang et al. [97] conducted robust multi-objective optimisation using the six-sigma approach [98], where the original performance based objective function is recast based on the mean μ_f and variance σ_f^2 of the performance probability density function. In this approach, not only is the sum of the mean and variance to be minimised, but they must also satisfy quality constraints, specified by the user, at $\pm\sigma_f^2$. High lift and low drag cases were utilised as the design points, with angle of attack and Mach number as the respective uncertain variables. Kriging models were created in the range of these parameters, using Latin-hypercubes to create a DOE where the number of samples for each case is equal to the square of the number of design variables.

Buckley et al. [80] tackled the multi-point optimisation problem again using a weighted sum approach. They employed a weight automation technique, so that off-design criteria would be met, but would not have an over-arching influence on the optimisation. Both unconstrained

and constrained techniques were utilised and achieved similar results, with the latter boasting an improvement in computational cost. A further notable aspect of this approach is that the updating weights can clearly highlight the critical and redundant cases in a given optimisation problem.

Constraint Management

Constraint management is by far the least reported of the topics discussed here, with many works choosing to limit the design space rather than incorporate robust constraint methods to define feasibility. The pitfalls of this approach has already been discussed in section 3.4, and therefore will not be repeated here. In terms of aerofoil optimisation however, Arias-Montaño et al. [99] reported similar findings, stating that simplistic approaches to constraint handling have often been utilised, which often discard infeasible or constraint violating solutions entirely.

Martínez et al. [100] provide an example of improved constraint handling, utilising the adaptive ε-constrained tolerance method described in section 3.4 for aerofoil aerodynamic optimisation. This work adapted the original ε-constrained approach for multi-objective algorithms, which is also employed in later chapters of the current research. Nonetheless, inspection of the optimisation problem shows a relatively limited application of such a versatile method. Performance based constraints ensure improved aerodynamic efficiency over a reference aerofoil, but only a single geometric constraint in terms of thickness to chord ratio points to a pre-constrained design space; which is confirmed by upon inspection of the design variables employed.

4.2.2 Implemented Methods

In terms of the overall optimisation and analysis methods, the PSO algorithm has already been selected in combination with low-fidelity aerodynamic methods, which is a necessary trade-off in accuracy for a heuristic approach. Chapter 5 will explore efficient implementation of high-fidelity analysis, using selective sampling to provide an overall improved view of true performance. This chapter focusses on employing methods to exploit a global design space with low-fidelity analysis only.

With global optimiser and aerodynamic methods in place, the remaining issues to be addressed from the above discussion is minimising design point over-optimisation, and employing suitable constraints to ensure feasible designs are output from a large design space. The latter will be discussed throughout the following sections, however it is again stressed that this research is focussed on global design optimisation problems, which requires the implementation of both global design spaces and global optimisation methods. In terms of over-optimisation, a six-sigma uncertainty quantification approach is used, similar to that of Liang et al. [97]. This

replaces the purely performance based cost function with a statistical one, which takes into account multiple flight states. The application of this approach will be discussed further in section 4.5.3.

4.3 Aerofoil Generation

The method employed to generate or adapt aerofoil shapes is crucial if the design space is to be thoroughly explored, considering the large effect that minor alterations in shape have on overall aerodynamic efficiency. Initially, preloaded data files of existing aerofoils were chosen simply by an integer variable. However, this is relatively inflexible and creates discontinuities in the design space, leading to convergence issues and sub-optimal results. With a high level of sensitivity existing between designs and performance, an equal degree of control is required over their shape, meaning the chosen geometry method should be capable of making small, continuous alterations to a given design. Increasing this level of fine tuning increases confidence in the optimisation algorithms ability to find globally optimal shapes. Of course, this will also increase the number of function evaluations required for convergence, but considering that the wing design is of the utmost importance in terms of aerodynamic performance, this high degree of control is not only desirable, but required in an exploratory design optimisation.

Various methods exist in literature for aerofoil shape generation, with varying success in terms of control, versatility, and use of physical design variables. Methods such as Hicks and Henne's analytical bump functions, and free-form deformation are popular in aerodynamic shape [18, 101, 102]. However, in each of these cases, only gradient based optimisation is carried out, due to the fact this method requires a starting point that is to be optimised. In this work, it is imperative that the design variables used to optimise an aerofoil shape are constructive in nature, rather than deformative [103]. Ultimately then, a method is required to create arbitrary aerofoils from scratch, rather than relying on a basis or initial design. Considering this, along with the aforementioned need for a high level of control over the design process, versatile constructive methods must be implemented in order to search for globally optimal designs.

4.3.1 Bézier Techniques

Bézier spline methods are commonly utilised in aerodynamic shape generation [104, 105]. In these techniques, a number of control points are specified in the desired plane, which define the curve shape. This curve will pass through beginning and end points, while using a weighted sum of all the control points to define any intermediate characteristics. Importantly then, intermediate control points are not passed through, and therefore no purely Bézier method allows for direct control over the aerofoils physical properties, since control points affect the full curve

shape, rather than acting as curve points themselves.

The versatile nature of Bézier curves allow for multiple aerofoil design approaches to be implemented, and thus experimentations have been carried out with both upper and lower surface generation, as well as inferring these surfaces from Bézier generated camber curves and thickness distributions.

A Bézier curve of degree n is defined by $n+1$ control points, with its mathematical form given as:

$$[b_0^n(t)]^T = \sum_{i=0}^{n} B_i^n(t)[b_i]^T \tag{4.1}$$

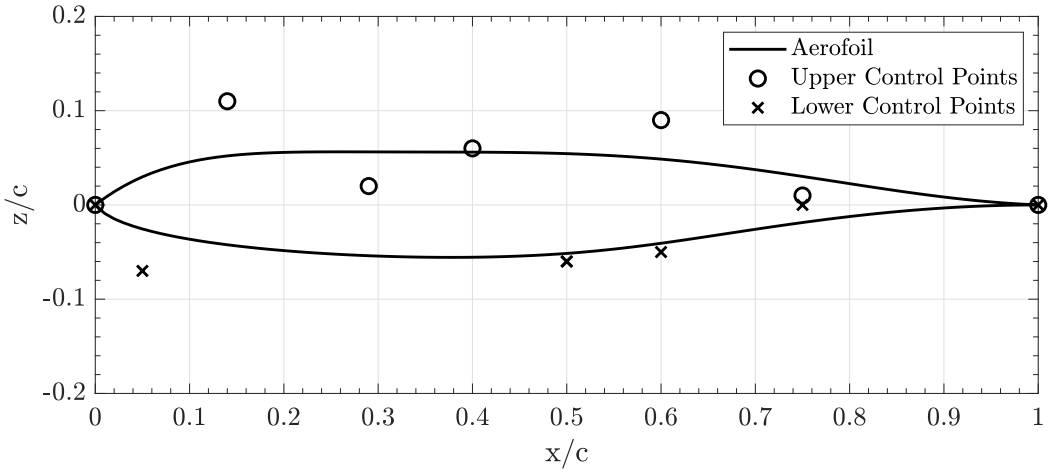
Where $b_0^n(t)$ is a point on the curve, (x,y), for a given $t \in [0\ \ 1]$. b_i is the position of the control points, and B_i^n is the blending function used to generate the shape based on these control point positions, realised through Bernstein polynomials:

$$B_i^n = \binom{n}{i}(1-t)^{n-i}t^i \tag{4.2}$$

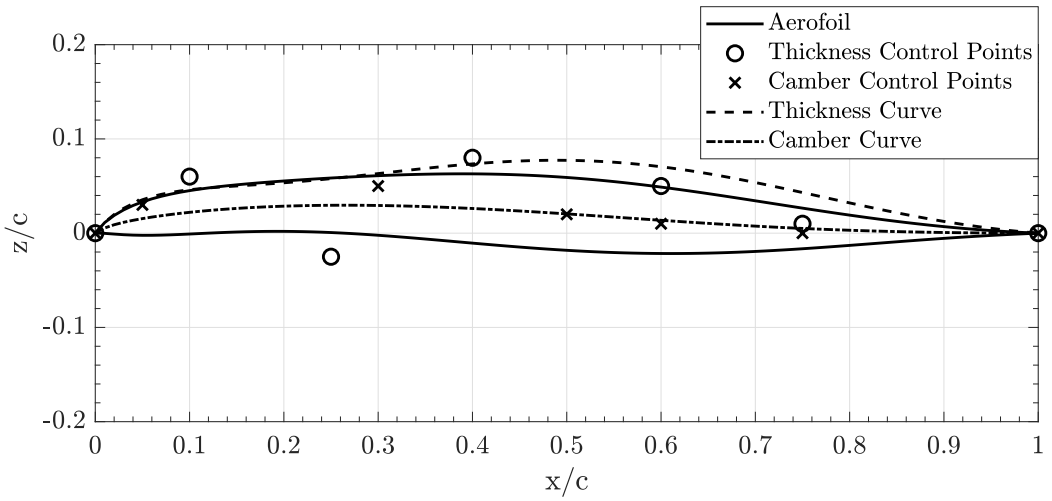
The implementation of these Bézier curves allows an arbitrary number of control points to be defined. Initially, two curves were employed to create upper and lower splines, which define the aerofoil surfaces directly. Since this method, or any purely Bézier method for that matter, is not specific to aerofoils, one of the most important aspects is in controlling the curves to create realistic aerofoil shapes. For this utilisation of the technique, some boundaries are required to be set on the leading edge control points, to avoid sharp edges and ensure that surfaces merge smoothly. Ideally however, considering that the leading edge radius is highly important in high-speed flows, it would be controlled directly.

A second approach was implemented however, in an attempt to define the aerofoil using more representative quantities, where the two curves describe thickness and camber of the aerofoil. In this case, the thickness distribution can be added equally in the normal direction at any given point of the camber line to generate upper and lower surfaces. This results in greater control over the leading edge radius, providing consistent radii on upper and lower surfaces. Both of the Bézier methods are shown in figure 4.1.

Once again this method does not eliminate poor or infeasible characteristics, such as small or negative thickness present across or at some point on the aerofoil. This is again due to the control points not physically representing curve points. For example, having thickness distribution control points lying in negative z may result in negative thickness at some point over the aerofoil. One might be tempted to limit these z-coordinates to a positive minimum value in order



(a) Direct Bézier



(b) Thickness camber Bézier

Figure 4.1: Bézier methods used for aerofoil generation

to avoid this issue. However, this will limit the amount of local control over aerofoil thickness, with such manipulation requiring large Δz between consecutive control points. An example of this is shown in 4.1b, where the negative thickness control point maintains relatively constant thickness between 5-20% chord.

Thus, while imposing such a minimum would reduce the infeasible design space, it would also reduce the feasible design space. In this research, a simple but key distinction is placed on the ability of a given design space to obtain a globally optimal solution; that the feasible design must be as broad as reasonably possible, using constraints to determine feasibility, as imposing design variable boundaries to achieve this results in a design space that cannot be considered global.

Validation of the Design Space

As previously mentioned, Bézier control points are not passed through by resulting curves, instead pulling on them like a gravitational force pulls passing objects. This lack of direct correlation between control points and physical aerofoil properties is not desirable, making design space boundaries difficult to properly quantify.

Importantly then, a discussion is required with regards to controlling Bézier methods, to ensure that undesirable traits, perhaps clear to an engineer but not captured in by the cost function or generation method itself, do not present themselves as feasible or desirable. The Bézier method is a powerful tool that can create an infinite number of potential solutions for a given set of control point boundaries, many of which will be unsuitable for aerofoil design. Nonetheless, particularly in a large design space optimisation, it is imperative not to over constrain these boundaries, as potentially optimal solutions may be lost. In cases such as this, where optimisation variables do not directly define an easily measured characteristic (such as maximum aerofoil thickness or leading edge radius), it is tempting to constrain the variables regardless, until satisfactory shapes are produced. While this may remove undesirable characteristics, it will most likely remove many feasible designs also. Instead, characteristics should be quantified once the aerofoil shape has been generated, and any undesirable traits penalised. This requires additional function evaluations on these unwanted designs. However, computational power is generally not limited in low-fidelity optimisations, making it a worthy trade-off to ensure that the broadest possible design space can be fully explored.

In turn, this raises the issue of what constitutes a large enough design space, and how to determine whether or not a given set of design variable boundaries is over constrained. This is discussed further in section 4.4, where target aerofoils are used to validate generative methods and design variable boundaries. Assuming a diverse enough set of pre-existing designs are

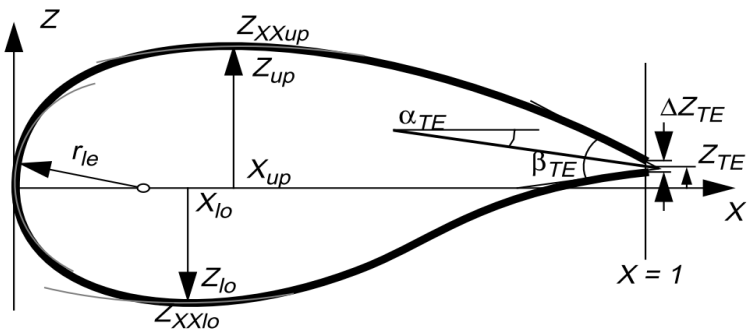


Figure 4.2: PARSEC aerofoil generation method variable definition [106]

implemented, this approach provides a physical representation of the shape generation methods limitations, which can then easily be altered by varying the necessary optimisation variable boundaries. To speed up convergence of future optimisations, this trial and error approach can also be used to ensure that the design space is not unnecessarily large. In such cases, a reduction of design variable boundaries may only reduce infeasible or undesirable designs, resulting in a more efficient overall method.

4.3.2 PARSEC

Initially, Bézier curves were selected for the task of aerofoil generation alone. However, while being a robust and versatile approach, there was something left to be desired in terms of their design variables. Since no direct control can be had over the aerofoils physical properties, the aerofoil specific method PARSEC [106], was also implemented to the overall framework. This approach creates 6^{th} order upper and lower polynomials based on geometric characteristics such as maximum thickness, leading edge radius and trailing wedge angle. The full set of parameters required in defining a PARSEC aerofoil are displayed in figure 4.2, with polynomials calculated below:

$$z = \sum_{i=1}^{6} a_i x^{\left(i-\frac{1}{2}\right)} \tag{4.3}$$

Where z is given vertical location of the curve at prescribed x coordinates. The coefficients a_i, and therefore equation (4.3), are computed for both upper and lower surfaces as a result of solving the linear system of equations (4.4)-(4.5), which contain the various geometric parameters. Upper surface coefficients are defined using:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ x_u^{\frac{1}{2}} & x_u^{\frac{3}{2}} & x_u^{\frac{5}{2}} & x_u^{\frac{7}{2}} & x_u^{\frac{9}{2}} & x_u^{\frac{11}{2}} \\ \frac{1}{2} & \frac{3}{2} & \frac{5}{2} & \frac{7}{2} & \frac{9}{2} & \frac{11}{2} \\ \frac{1}{2}x_u^{-\frac{1}{2}} & \frac{3}{2}x_u^{\frac{1}{2}} & \frac{5}{2}x_u^{\frac{3}{2}} & \frac{7}{2}x_u^{\frac{5}{2}} & \frac{9}{2}x_u^{\frac{7}{2}} & \frac{11}{2}x_u^{\frac{9}{2}} \\ -\frac{1}{4}x_u^{-\frac{3}{2}} & \frac{3}{4}x_u^{-\frac{1}{2}} & \frac{15}{4}x_u^{\frac{1}{2}} & \frac{35}{4}x_u^{\frac{3}{2}} & \frac{63}{4}x_u^{\frac{5}{2}} & \frac{99}{4}x_u^{\frac{7}{2}} \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_{u_1} \\ a_{u_2} \\ a_{u_3} \\ a_{u_4} \\ a_{u_5} \\ a_{u_6} \end{bmatrix} = \begin{bmatrix} z_{te} + \frac{1}{2}\Delta z_{te} \\ z_u \\ tan\left(\alpha_{te} - \frac{1}{2}\beta_{te}\right) \\ 0 \\ z_{xx_u} \\ \sqrt{2r_{le_u}} \end{bmatrix} \tag{4.4}$$

The lower coefficients are computed in a similar manner:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ x_l^{\frac{1}{2}} & x_l^{\frac{3}{2}} & x_l^{\frac{5}{2}} & x_l^{\frac{7}{2}} & x_l^{\frac{9}{2}} & x_l^{\frac{11}{2}} \\ \frac{1}{2} & \frac{3}{2} & \frac{5}{2} & \frac{7}{2} & \frac{9}{2} & \frac{11}{2} \\ \frac{1}{2}x_l^{-\frac{1}{2}} & \frac{3}{2}x_l^{\frac{1}{2}} & \frac{5}{2}x_l^{\frac{3}{2}} & \frac{7}{2}x_l^{\frac{5}{2}} & \frac{9}{2}x_l^{\frac{7}{2}} & \frac{11}{2}x_l^{\frac{9}{2}} \\ -\frac{1}{4}x_l^{-\frac{3}{2}} & \frac{3}{4}x_l^{-\frac{1}{2}} & \frac{15}{4}x_l^{\frac{1}{2}} & \frac{35}{4}x_l^{\frac{3}{2}} & \frac{63}{4}x_l^{\frac{5}{2}} & \frac{99}{4}x_l^{\frac{7}{2}} \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_{l_1} \\ a_{l_2} \\ a_{l_3} \\ a_{l_4} \\ a_{l_5} \\ a_{l_6} \end{bmatrix} = \begin{bmatrix} z_{te} - \frac{1}{2}\Delta z_{te} \\ z_l \\ tan\left(\alpha_{te} + \frac{1}{2}\beta_{te}\right) \\ 0 \\ z_{xx_l} \\ -\sqrt{2r_{le_l}} \end{bmatrix} \tag{4.5}$$

4.4 Existing Aerofoil Shape Optimisation

The aerodynamics of any winged vehicle are highly sensitive to aerofoil shape, thus making modelling of such geometries a highly important task. The implemented methods should provide a high degree of control over the aerofoil shape; ensuring that small changes in design variables result in a small change in geometry. Kulfan [104] defined typical geometric wind tunnel tolerances in comparisons between analytical and approximate aerofoils as follows:

$$|z_{foil} - z_{approx}| < \begin{cases} 4 \times 10^{-4}, & \text{if } x/c \leq 0.2 \\ 8 \times 10^{-4}, & \text{otherwise} \end{cases} \tag{4.6}$$

These tight tolerances of < 1mm highlight the level of control required in an aerofoil design space. Thus, before implementing shape generation software into a performance driven optimisation framework, it is critical to demonstrate that the tools are capable of reproducing existing shapes. If design spaces are incapable of producing such designs that perform well for the given cost function, designs output from optimisations will most likely be suboptimal, since design variable boundaries, and/or level of control of the shape generation tool, are suboptimal themselves.

Validation of the employed methods is carried out here, using a large database of aerofoils as target shapes for the implemented design spaces to mimic. This presents a shape optimisation in which the SOPSO method detailed in the preceding chapter can be employed. From an initial

sampling of the design space, the difference in equation 4.6 at each discretised location of the target aerofoil can be minimised. This allows generation techniques, and their respective design space boundaries to be validated.

4.4.1 Target Aerofoils

To benchmark the employed generation methods, and ensure any constraints developed are relevant to the problem, a large library of existing aerofoil data was used within the framework. The UIUC Airfoil Coordinates Database [107] provides over 1500 aerofoil geometry files, and can therefore be viewed as a robust, diverse representation of such shapes. With a broad spectrum of designs, the methods employed for aerofoil generation can be validated by attempting to match these existing shapes as closely as possible.

Furthermore, variable boundaries, commonly defined arbitrarily or depending on early or unsatisfactory optimisation results, can be proven as fit for purpose. By using a single set of variable boundaries to produce a large number of varying aerofoil shapes, an appropriately sized design space can be found. If the optimal shapes consistently push on a variable boundary, its scope should be extended, since improved designs potentially lie outwith. Similarly, if a boundary and its surrounding area remains untouched throughout the library of target aerofoils, this area can be seen as redundant. Although the design space should be able to produce a broad range of shapes, an overly large space with redundant areas will only add to noise within the optimisation, leading to a reduced probability of convergence.

Another essential reason for analysing such a large bank of designs with the context of the framework, is to verify that any imposed constraints on the design space do not show existing designs as suboptimal. Similar to variable boundaries that do not allow for certain existing shapes to be made, constraints that show such shapes as violating will essentially discard them during the optimisation. This will be discussed further in section 4.4.3.

To maximise confidence in the framework's ability to consistently generate and constrain designs, it is important to utilise as much aerofoil data as possible. However, since the library contains all types of aerofoils, many are not suitable for high-speed applications. Although the majority of these aerofoils can still be recreated by the variables boundaries and constraints employed, any aerofoils that violated constraints specific to high-speed flight were discarded.

Aerofoils with greater than 20% thickness with respect to chord, along with those that had a negligible leading edge radius on either the upper or lower surface, were discarded from the library. Furthermore, some poorly defined aerofoils were also removed, since it would be unnecessary to replicate such noisy geometries. This leaves almost 1300 aerofoils to be matched

by the various generation methods, which should allow for a robust comparison between their performance and limitations.

4.4.2 Optimisation Problem

The bank of collated aerofoils provides a diverse population of target shapes for the Bézier and PARSEC design spaces to recreate. Flexing these generative methods in such a way allows their suitability to the potential problem spaces within the research scope to be determined, along with any limitations or pitfalls. Target aerofoils have been chosen to demonstrate the versatility of these methods, while also ensuring that there are a number of typical design characteristics contained within dataset. In the context of high-speed applications, this means that supercritical aerofoils are particularly present within the target library, while avoiding unnecessary shapes such as overly thick aerofoils. A large number of typically low speed designs are also present within the library, and have been included for two main reasons. The first is that such designs contain desirable characteristics for aerofoil shapes in general, while containing others that are sub-optimal for high-speed flight. Secondly, various RLV or HST concepts utilise aerofoils that are not typical of high-speed designs, meaning diversity outwith these normal characteristics should be encouraged.

Existing shape optimisation of this kind not only validates the use of the generative methods employed, by ensuring common designs can be replicated, but also further demonstrates the capability of the chosen optimiser.

Each aerofoil is therefore to be match as closely as possible, by each generation method, in order to provide an overall view of their comparative performance and limitations. The cost function to be minimised is the root-mean-square error (RMSE) between the generated designs and the target aerofoil, as shown below:

$$\begin{aligned} \min \quad & \sqrt{\frac{\sum_{i=1}^{N}\left(z_i^{foil}-z_i^{approx}\right)^2}{N}} \\ \text{s.t.} \quad & x_j^l \leq x_j \leq x_j^u;\ j = 1:m \end{aligned} \tag{4.7}$$

Where N indicates the total number of sample points where the aerofoils are compared, and m is the number of design variables. Constraints are also implemented to aid in convergence of the optimisation, however the development and validation of these require detailed discussion, which will be given in the proceeding section.

4.4.3 Constraints

As previously mentioned, handling of the aerofoil design space has been one of the most challenging problems faced in this research. Finding the correct balance to allow for a diverse set of feasible, desirable aerofoils to be generated without over-constraining the design space is a far from trivial task. While kinks and imperfections within these geometries can be easily spotted by human eyes, it is often a challenge to find the correct constraint that will eradicate such behaviour, without limiting the design space elsewhere.

In order to derive constraints to be placed upon generated aerofoils, common properties within nominal shapes must be found. Ensuring these commonalties define an aerofoils feasibility is paramount, since without them the optimiser would seek to minimise the cost function without any consideration given to properties such as leading edge radius or thickness. Once again the UIUC library was utilised in order to determine such constraints, along with many trial optimisations in which the optimiser found new and interesting ways to take advantage of the growing number of constraints.

These common properties that exist in the vast majority of aerofoils are as follows:

- Non-zero leading edge radius on both upper and lower surfaces
- Maximum thickness and maximum upper surface point away from leading and trailing edges (typically between 10-70%)
- Minimum peak thickness to chord ratio (typically > 4%)
- Constantly increasing thickness up to maximum ($dt/dx \geq 0$)
- Constantly decreasing thickness aft of maximum ($dt/dx \leq 0$)
- Smooth upper surface up to maximum point ($du^2/d^2x \leq 0$)

Before attempting to recreate the various geometries found in the UIUC database, it is important to validate the implemented constraints. If these target shapes are found to violate the constraints placed upon the generative methods within the shape optimisation, the optimiser will not be able to reproduce them, prioritising non-violating geometries instead. Therefore, any aerofoils that do violate any one of the employed constraints must be removed from the library. On the other hand, since these geometries are commonplace and have applications as lifting surfaces, they should not violate any constraints that are not specific to the application of interest.

By passing the large library of existing aerofoils through these constraints, it allows those outwith the scope of the current work to be identified and removed, along with those that have

been poorly defined. Furthermore, it allows constraints and their respective boundaries to be validated, since existing shapes should be seen as non-violating by the optimiser. Aerofoils removed from the library were therefore those with high thickness to chord ratio, sharp leading edge radius, and those with geometric imperfections that manifest through the described constraints. Following the removal of these unsuitable aerofoils, over 1200 still remain in the library for testing the implemented generation methods. This allows for comprehensive analysis and comparison of the Bézier and PARSEC aerofoil techniques.

4.4.4 Design Space

The design space of each method is established here, which is used to provide upper and lower variable boundaries within which the optimiser can exploit. Direct Bézier, Bézier thickness and camber, and PARSEC design spaces are defined in table 4.1.

With Bézier curve techniques independent of aerofoil parameters, some restrictions are required to produce closed aerofoil surfaces. While Bézier curves do not pass through intermediate control points, they begin and end on outer control points. Thus, to produce non-dimensional aerofoils, these are fixed at $x = 0$ and $x = 1$ respectively. Furthermore, initial control points are fixed at $z = 0$, with final points given some flexibility in the z direction to allow modelling of finite trailing edges. These constraints hold for each curve within both implemented Bézier methods.

The remaining control points are free to move in both dimensions. Four intermediate control points are utilised in both methods, which are defined by half-cosine spacing in the x direction for both minimum and maximum variable boundaries, to allow greater control at the leading edge where maximum curvature is often present. Variation of control point height is then applied consistently depending on the method, as shown in table 4.1.

Although the potential height of these control points is considerably higher than applicable aerofoils in the high-speed domain, it must be reiterated that high control points do not necessarily correspond to thick aerofoils, especially if they are surrounded by lower control points. Maintaining such large variable boundaries, allows high disparity between neighbouring control points, and thus nuanced control over specific regions of the aerofoil.

As for the PARSEC method, variables are based on physical aerofoil properties, such as leading edge radius or curvature at the surface apex. This allows a more intuitive definition of minimum and maximum variable boundaries.

Table 4.1: Design space definition for Bézier and PARSEC aerofoil generation methods

Method	Variable	Minimum	Maximum	Comments
Bézier	$\mathbf{x}_{u,l}$	$1-\cos\frac{i\pi}{2n}-\frac{1}{n}$	$1-\cos\frac{i\pi}{2n}+\frac{1}{n}$	Half-cosine spacing from 0 to 1, where n is number of free control points ($n=4$)
	$z_{u_{2-5}}$	0	0.2	Non-dimensional, control points fixed at extremities
	$z_{l_{2-5}}$	-0.2	0	
Bézier TC	$\mathbf{x}_{t,c}$	As above	As above	
	$z_{t_{2-5}}$	-0.05	0.2	
	$z_{c_{2-5}}$	-0.1	0.2	
PARSEC	r_{le}	0.001	0.1	Variables from figure 4.2
	x_u	0.05	0.8	
	z_u	0.005	0.2	
	z_{xx_u}	-1.2	0	
	x_l	0.05	0.8	
	z_l	-0.1	-0.02	
	z_{xx_l}	0	1.2	
	z_{te}	-0.02	0.02	
	Δz_{te}	0	0.02	
	α_{te}, deg	-25	2	
	β_{te}, deg	3	40	

4.4.5 Results

Overall cost function results are shown in figure 4.3, depicting the RMSE of Béizer, Bézier thickness camber, and PARSEC generation methods for each target aerofoil within the reduced UIUC database. Mean and standard deviation for methods across all database aerofoils are also shown in table 4.2. From this data, it is clear that both Bézier methods consistently outperform the PARSEC technique, showing greater versatility in generating the various nuances present within the library. This is interesting result, considering that the PARSEC method has been developed specifically for aerofoil generation.

Direct and thickness camber Bézier methods show promising performance, with RSME values around the Kulfan tolerance specified in equation 4.6. The direct method appears slightly more consistent than thickness camber approach, perhaps due to the higher correlation between variables and physical design. This is more prevalent in terms of cost function standard deviation between aerofoils, with overall mean values relatively similar. Closer inspection of figure 4.3 show a couple of outlier cases where the thickness camber method struggles to drop below 10^{-2}, with direct Bézier consistently at 10^{-3} or lower. These results validate the use of both Bézier methods in a global design optimisation context, simultaneously verifying the design variable

Table 4.2: Mean and standard deviation of RMSE across UIUC database

	Bézier	Bézier TC	PARSEC
Mean	8.26×10^{-4}	8.87×10^{-4}	2.68×10^{-3}
SD	8.19×10^{-4}	9.65×10^{-4}	3.73×10^{-3}

boundaries employed. Improved cost function values could be obtained by increasing the number of control points. Nonetheless, the 5^{th} order curves employed here strike a good balance in a design optimisation context between number of variables and degree of control.

The PARSEC approach shows a reduction in performance against the Bézier methods, in terms of both mean and deviation. In a handful of circumstances PARSEC demonstrates minimal RMSE, outperforming both Bézier implementations. However, a considerable number of aerofoils are poorly represented by the PARSEC method, highlighting its inefficiency in a global context. Importantly, the PARSEC variable boundaries were repeatedly extended in an attempt to replicate the performance of Bézier methods, with little improvement observed overall. This highlights the downfall of attempting to utilising limited physical properties in a shape generation tool that requires intricate detail across the entire surface.

The PARSEC method appears a good option when well-behaved or a specific subset of designs are preferred, since a number of such subsets can be directly enforced through the physicality of design variables. However, there comes a point where trying to extend these boundaries has the opposite of the desired effect; generating highly unsuitable polynomials for aerofoil surfaces. Including variable sets that exhibit atypical aerofoil properties, such as minimum lower surface point below 5% chord, will result in a number of poorly defined rear aerofoil surfaces, since variables are concentrated near the leading edge. This increases the infeasible regions of the design space, without adding the necessary control, resulting in a nosier design space with minimal increase in versatility.

4.5 Aerodynamic Shape Optimisation

With the implemented aerofoil generation methods tested against diverse database of existing aerofoils, the direct Bézier method was chosen for incorporation within a performance driven optimisation context, as it demonstrated the highest level of control among the implemented methods. Confidence in the results of these performance based optimisations will be increased through a number of methods discussed in the proceeding sections. However, with the generative method, scope of design space, and geometric constraints already validated, it is clear that any difficulties in convergence will be a result of the optimisation or analysis methods.

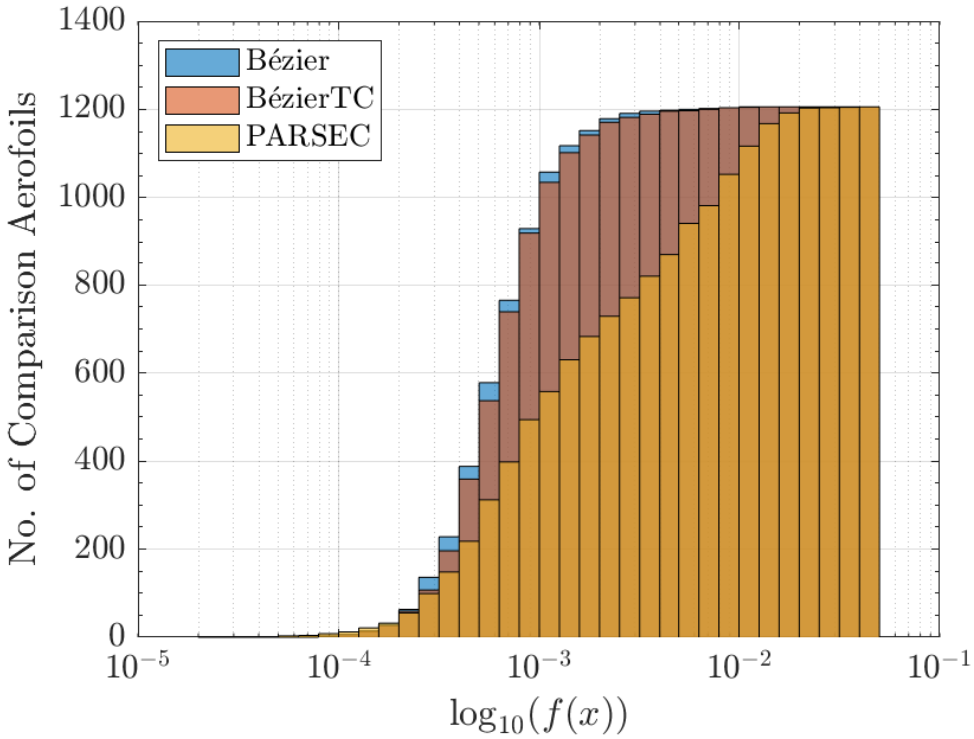


Figure 4.3: RMSE cumulative histogram for implemented aerofoil generation methods compared to target aerofoils, shown on log scale

Aerodynamic aerofoil optimisation allows the developed modules in the previous chapters to be combined for a simple two-dimensional case, as well as verifying their suitability to the problem, before moving on to more complex three-dimensional design spaces. It is important to note that, although the design space and individual analysis of aerofoil is simple, global optimisation of aerofoils under any conditions is far from trivial. Aerodynamic performance is highly sensitive to aerofoil characteristics, requiring the high degree of control provided by the generative methods. This control comes at cost however, drastically increasing the number of possible aerofoil shapes, and in turn making convergence on globally optimal designs a difficult task. Furthermore, a global design space requires implementation of robust constraints, as covered previously.

Two optimisations are carried out: the first attempting to improve upon a baseline shape, with strict geometric and performance based constraints ensuring a fair comparison; and an arbitrary case, where the design space can be searched for the globally optimal aerofoil, subject to more general constraints. The baseline case seeks to validate the usefulness of the overall framework; by improving upon an existing shape, even with the strict constraints placed upon candidate designs. Following this, an arbitrary optimisation can be carried out, allowing the optimiser further freedom in finding ideal designs for the given problem.

4.5.1 Design Space

The design space described in the section 4.4 has already been validated in its ability to recreate a large bank of existing aerofoils. Therefore, the design space should not be restricted in any way, so as to not invalidate these results.

4.5.2 Constraint Management

In a similar manner to the design space, the previous applied geometric constraints have been implemented to ensure only feasible designs are output, and have been validated against the aerofoils within the UIUC database. Unlike the rigidity applied to the design space however, some of the geometric constraints are adapted to quickly remove candidate aerofoils unfit for the application of high-speed flight.

Further constraints can be applied, particularly in the baseline cases, ensuring that output designs compare fairly. This is achieved through requiring physically properties to be maintained, while avoiding sacrifice of performance characteristics in order to produce an improved objective function. The specific constraints employed are detailed in the proceeding sections, within the overall optimisation problem definitions.

4.5.3 Analysis

Aerodynamic analysis is carried out in a similar manner to aerofoil validation in section 2.6, with tangent-wedge being applied on impact surfaces, and Prandtl-Meyer in shadow regions. Analysis is purely inviscid, with viscous results for two-dimensional shapes being low priority for the present work.

To ensure the final design is not a result of over-optimisation on the chosen flight state, uncertainty is applied around the angle of attack, resulting in a statistical sigma cost function defined in equation (4.8). This uncertainty is realised by analysing the aerofoil at an angle of attack half a degree above and below the nominal case, and finding the resulting cost function mean and variance.

$$\min \ \mu(f) + n\sigma(f)^2 \tag{4.8}$$

Where f is the cost function, and n is the sigma level used to weigh the influence of variance on the new statistical cost function, set to $n = 6$ in line with similar works [97, 98].

4.5.4 Baseline Driven Results

As an initial test, the framework is tasked with finding an optimal aerofoil shape within a large design space, that is tightly constrained by various geometric and performance based metrics provided by a baseline design. Importantly, it should be noted that the baseline shape itself is not used as a starting point, with candidate designs being compared on constraint metrics alone. The goal is to demonstrate that the optimisation framework is capable of consistently converging on an improved design, from an arbitrary starting population.

The baseline aerofoil is the NACA66-206, which was used in validation of the overall aerodynamic model. The objective is to minimise drag coefficient at flow conditions previous defined during aerodynamic validation (table 2.2), while maintaining key characteristics of the baseline design. To ensure a fair comparison between candidate and baseline aerofoils, these characteristics are both geometric and performance based, and are outlined along with overall optimisation problem in table 4.3.

Table 4.3: Baseline aerofoil optimisation problem definition

		Comments
Minimise	C_D	
Subject to		
Variable boundaries	$\mathbf{x}_{lb} \leq \mathbf{x} \leq \mathbf{x}_{ub}$	Defined in table 4.1
Baseline Constraints	$C_L \geq C_{L_{base}}$ $t/c \geq (t/c)_{base}$ $r_{le} \geq r_{le_{base}}$ $A \geq A_{base}$	Maintain baseline performance and select geometric properties for fair comparison
General Constraints	$0.1 \leq c_{t_{max}} \leq 0.7$	Non-dimensional chord location of maximum aerofoil thickness
	$\left(\frac{dt}{dc}\right)_{t < t_{max}} \geq 0$	Aerofoil thickness cannot decrease up to maximum thickness
	$\left(\frac{dt}{dc}\right)_{t > t_{max}} \leq 0$	Aerofoil thickness cannot increase after maximum thickness
	$\left(\frac{d^2u}{dx^2}\right)_{u < u_{max}} \leq 0$	Upper surface curvature cannot increase up to apex

Being a single-objective optimisation, the SOPSO variant described in section 3.3.2 is utilised here. To provide a comparison, the gradient based SQP method of section 3.2.3 is also employed. Each optimiser is run multiple times for the same problem, allowing global convergence criteria to be determined. As the SQP is a local, gradient based optimiser, the number of runs is ten

times that of SOPSO, with the best among them being selected for comparison again the global method.

Both optimisers pool from a set of initial designs determined from Latin hypercube sampling. In the SOPSO method these make up the starting population, whereas the multi-point SQP method uses one design per run as a starting point. The PSO is run 10 times, varying the random seed generator at the beginning of each run to ensure each is unique. The SQP optimiser is therefore run 100 times, with the best 10 performers selected for comparison with the PSO.

Defining optimality with the caveat of equal to or increased lift coefficient compared to that of the baseline ensures that minimisation of the objective function does not hinder performance elsewhere. Placing geometric constraints on leading edge radius, maximum thickness, and cross-sectional area, requires the optimiser to find a genuinely improved shape, rather than one which uses minimal area to achieve minimum drag, for example.

Figure 4.4 shows a direct comparison between SQP and SOPSO objective function, along with the baseline. As noted, the SQP results displayed are the best 10% of all runs, whereas for the PSO, only 10 optimisations were carried out in total. Figures 4.5 and 4.6 show the gradient and population based results respectively. The overall optimal C_D design found by each optimiser is shown in comparison to the baseline in both figures, with shading to visualise the relative variance in final designs over the course of different runs.

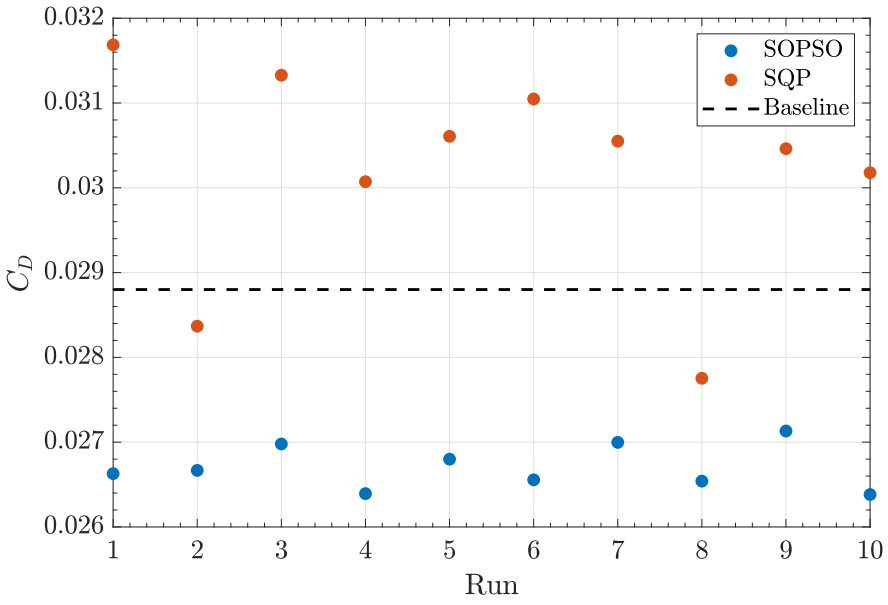


Figure 4.4: SOPSO and SQP objective function comparison with baseline

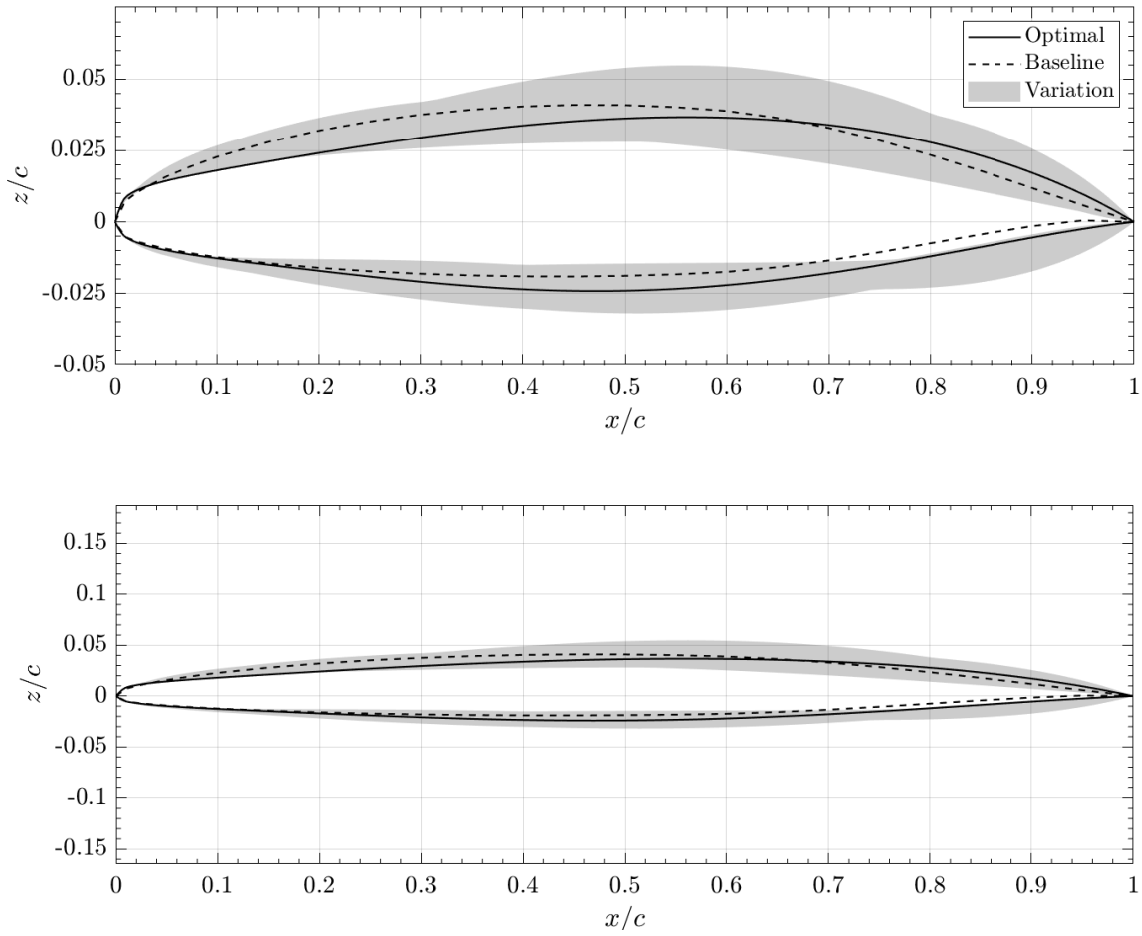


Figure 4.5: SQP optimal design and variation of best 10 runs, compared with baseline shape, shown in expanded view (top) and equal axes (bottom)

In all cases, the SOPSO algorithm outperforms the best overall objective function found from by SQP, while also producing an improved design with respect to the baseline on each occasion. In contrast, out of the 100 multi-point runs carried out by the SQP algorithm, only 2% of the time does it find an improvement over the baseline case. Furthermore, the SOPSO results show considerably improved convergence characteristics, with a standard deviation almost five times smaller than the 10 most promising SQP cases. This is reinforced in figures 4.5 and 4.6, where a large variation can be observed for SQP optimisation, compared to the PSO variance, which can only be properly visualised in the expanded view.

In terms of the optimal designs themselves, they remain very similar to the baseline shape. While this may seem intuitive, it demonstrates some key features about the optimisation process and the original design itself. Importantly, it must be restated that the baseline shape is not used as a starting point in either case; only geometric and performance characteristics are utilised, to ensure candidate designs are reasonable comparisons. Thus, consistent reproduction of the baseline shape shows that it is in fact a close to optimal design for the given problem. Unsur-

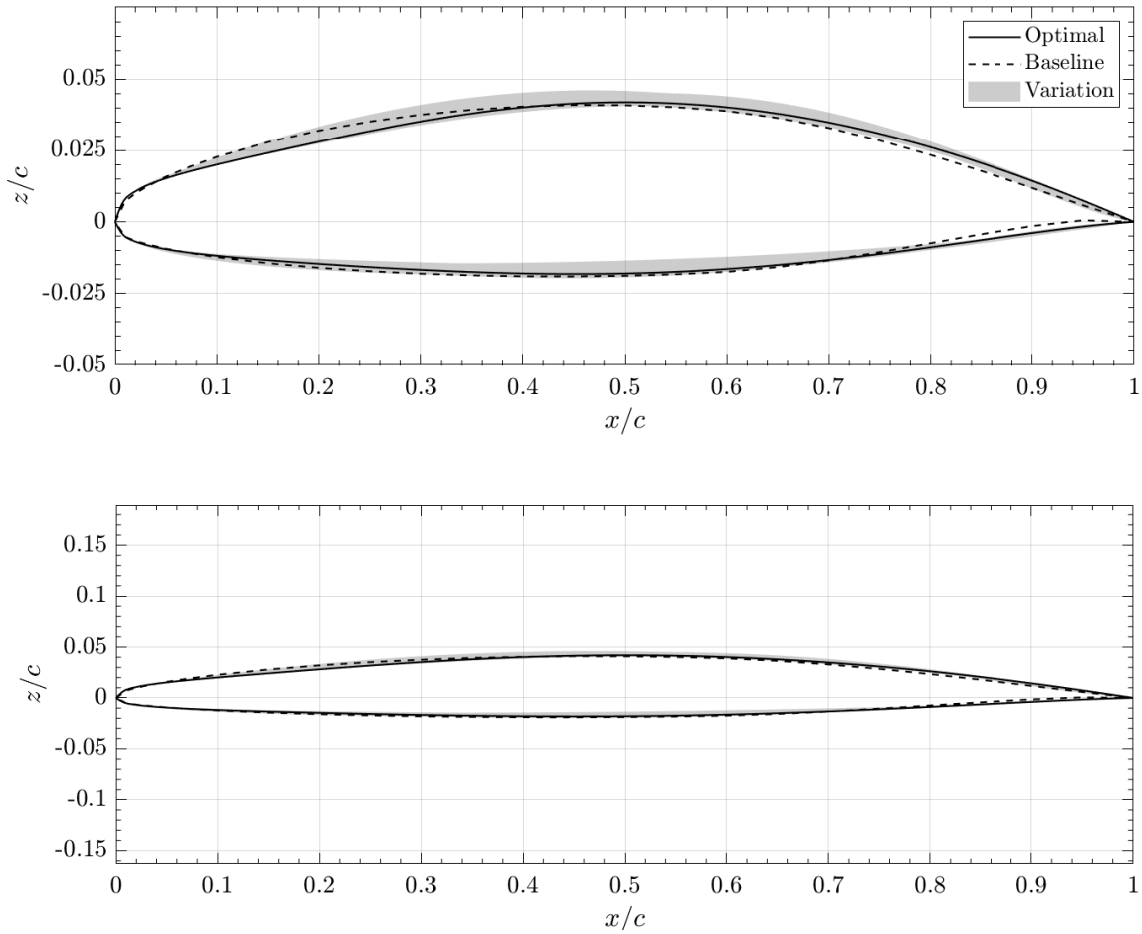


Figure 4.6: SOPSO optimal design and variation across all runs, compared with baseline shape, shown in expanded view (top) and equal axes (bottom)

prisingly, the best candidate shapes exhibit geometric properties that lie close to the baseline boundaries: tending towards minimal thickness, cross-sectional area, and leading edge radius. These characteristics would hold for any drag minimisation problem however, with a large variety of potential aerofoil designs meeting this criteria. Clearly then, performance constraints requiring candidate designs to at least maintain lift coefficient of the baseline have a large influence, driving the optimiser towards the original design.

Observing the optimal overall design in figure 4.6, the main difference that produces a reduction in drag over the baseline case is the shift in area towards the trailing edge. This is clearly seen in the expanded view, with thinner upper and lower surfaces between 10-40% chord, which is then accounted for at 70-100%. This shift also results in the maximum thickness location moving further aft, by approximately 5-10%.

These results present a promising validation of the SOPSO method, which has shown to be capable of consistently finding improved designs versus a baseline case that imposes stringent

constraints on the overall design space. Finding these small feasible and optimal areas of the design space is a challenging task, demonstrated by the SQP algorithm, which only found an improved design in 2% of runs. The next step is to test the optimisers within a design space that is not constrained by a baseline shape; resulting in a much larger feasible space, which could potentially have a large impact on convergence characteristics.

4.5.5 Global Results

Following the optimisers success in repeatedly converging on improved shapes compared to the baseline case, a more open ended optimisation was carried out, allowing a broader spectrum of shapes to be considered optimal. This logical next step allows the optimiser to search for globally optimal shapes, without the need to maintain geometric or performance characteristics of a pre-existing design.

Once again the repeatability of this arbitrary case is of particular interest. With relaxed constraints and therefore a larger region of feasible design space, it is reasonable to assume that convergence characteristics of the optimiser will be reduced compared to the baseline case. Conversely, consistent convergence towards similar final designs provides increased confidence that the globally optimal shape has been found.

In the absence of baseline lifting characteristics for candidate designs to maintain or improve upon, the objective function was changed from minimisation of drag coefficient, to maximisation of aerodynamic efficiency, C_L/C_D. This should ensure that optimal designs exhibit low drag and high lift characteristics. Furthermore, an even wider spectrum of shapes can compete for optimality, since higher lift values allow for increased drag, unlike the baseline case.

Other than the change in objective function, the optimisation problem is almost identical to that of table 4.3. Further differences arise from the removal of baseline performance constraints, with baseline geometric constraints being replaced generalised boundary values. These replacement constraint values ensure feasible thickness and area characteristics are present within optimal designs. To determine appropriate values for these constraints, the UIUC aerofoil database was once again employed. By aggregating thickness and area properties for all aerofoils within the database, reasonable constraint values were chosen.

Again it is important to note that design space and remaining general constraints remain consistent to the original validation in section 4.4, maintaining the validity of the framework to accurately replicate all existing shapes within the reduced UIUC database.

Once more, 10 runs are carried out using SOPSO, with the best 10 runs selected from 100 total

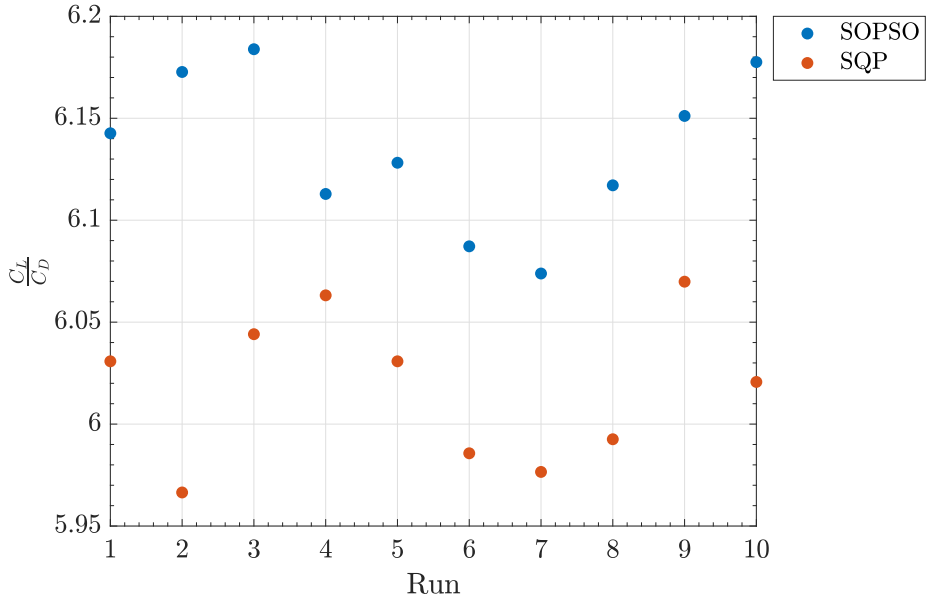


Figure 4.7: SOPSO and SQP objective function comparison

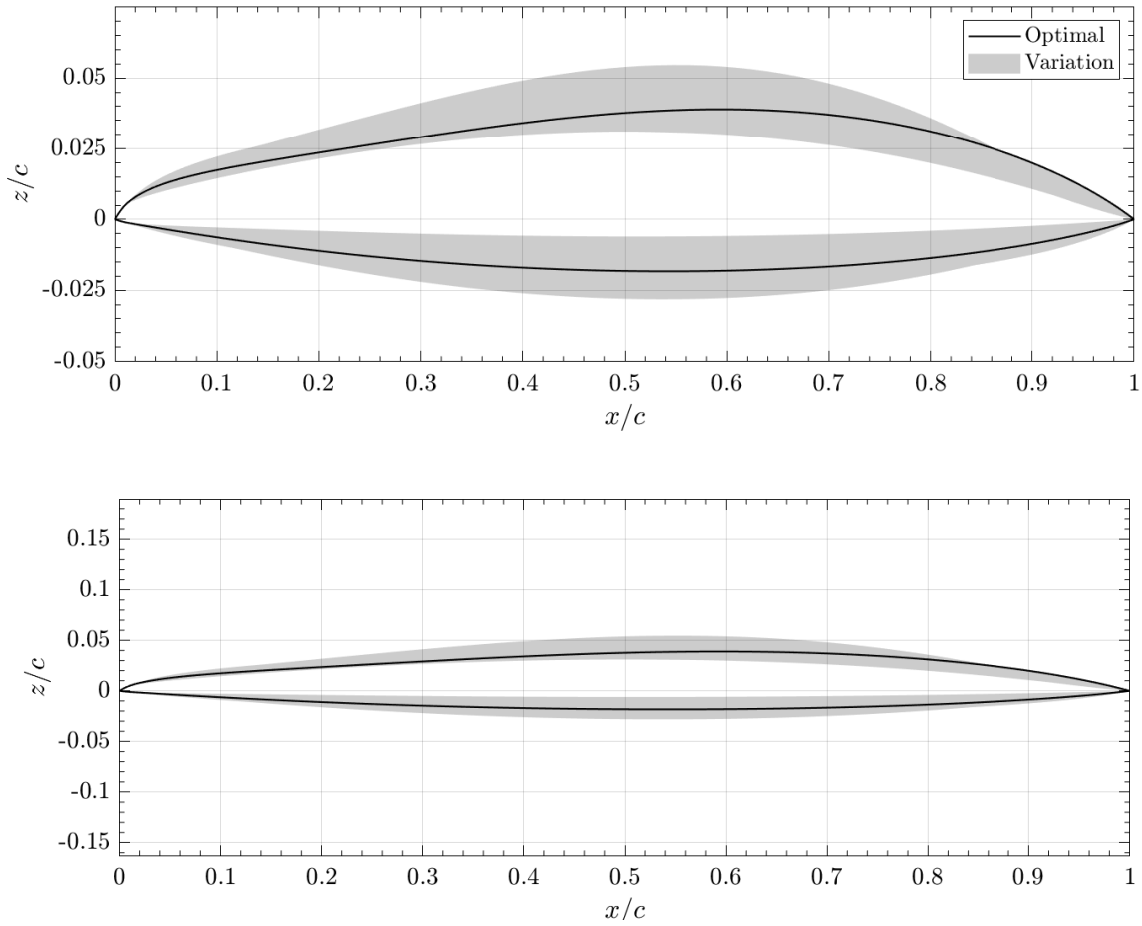


Figure 4.8: SQP optimal design and variation of best 10 runs, shown in expanded view (top) and equal axes (bottom)

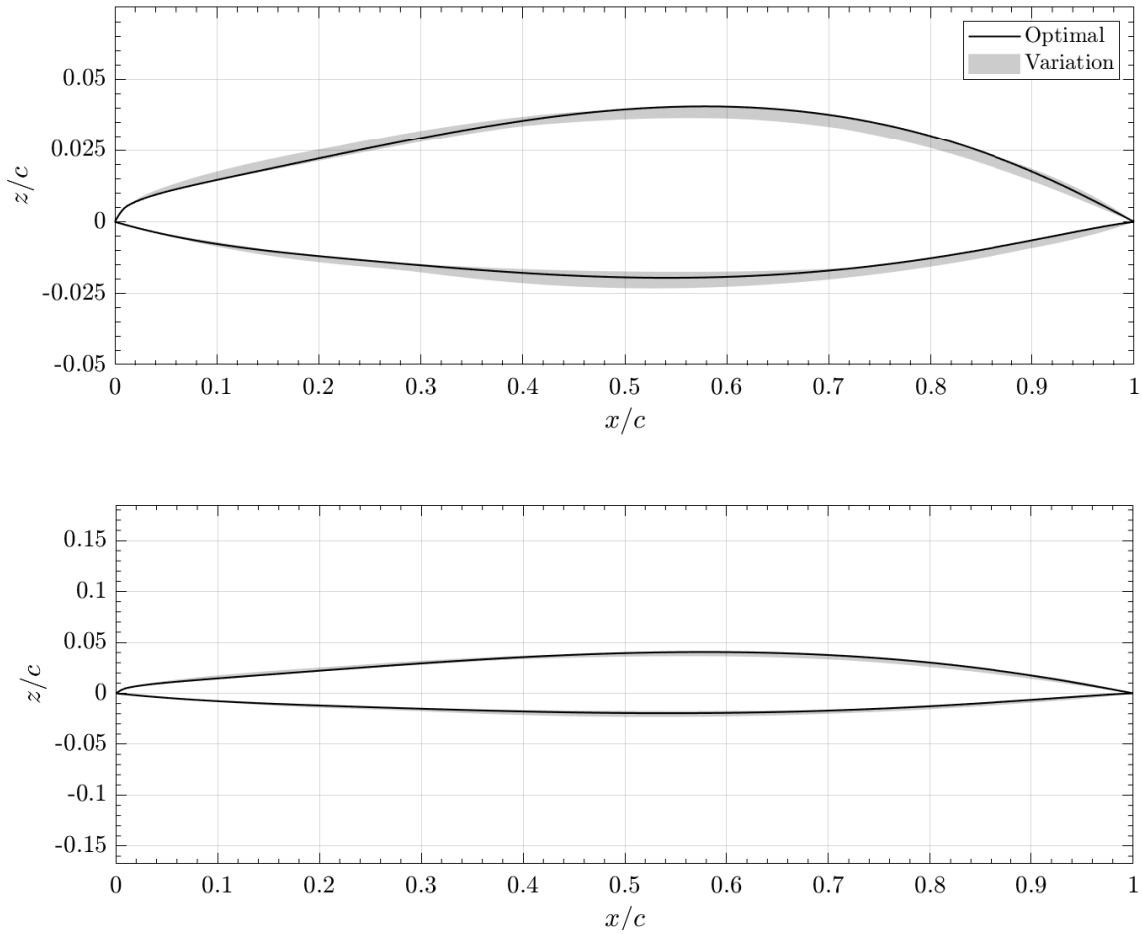


Figure 4.9: SOPSO optimal design and variation of all runs, shown in expanded view (top) and equal axes (bottom)

SQP runs. Comparative results are shown in figure 4.7, with overall best and variance between runs for SQP and SOPSO shown in figures 4.8 and 4.9 respectively.

As in the baseline case, the SOPSO algorithm outperforms the best SQP run on every occasion, consistently finding an improved aerodynamic efficiency. Interestingly, variation in the objective function is around the same for both optimisers, despite the drastically increased variation in SQP designs compared to SOPSO. This can be explained by a noisier objective function space when utilising aerodynamic efficiency, allowing multiple unique designs to appear close to optimal. Importantly then, despite only a small improvement in objective function, SOPSO results once again show convergence towards a similar design. This demonstrates the optimisers ability to cut through the noise present in the objective function; consistently discerning the global optimal from a spread of nearby local optima with similar objective values.

The resulting designs are also similar to the baseline case, providing further confirmation that the baseline design is close to optimal for the given problem. Shifting area aft once again

allows thin, almost wedge like frontal areas, with minimal curvature except for the leading edges. Unsurprisingly, a loosening in leading edge radius requirements results in reduced thickness and therefore area here also, with maximum thickness shifted even further aft than in the baseline case. Upper leading edge radius remains relatively high however, allowing this flat upper surface to extend towards maximum thickness. This is due to the Prandtl-Meyer expansion fans used to analyse shadow surfaces, which relieve surface pressure as turning angle increases. Reducing the upper surface pressure so quickly gives a reduction in overall downforce, increasing lift generation at the given angle of attack.

4.6 Conclusions

On the surface, aerofoil optimisation appears to be a simple aerodynamic problem. However, the research that continues to be published on the subject demonstrates the highly nuanced nature of these two dimensional surfaces. This work sought to produce optimal aerofoil designs from an arbitrary starting point, using population based optimisation techniques and robust constraints. The key starting point to carry out this work was determining which method should be employed to generate these shapes. A number of methods were therefore implemented, and tested for their ability to replicate pre-existing aerofoil designs from a large database. This study validated the use of Bézier methods and their implemented design spaces, for use in a global aerodynamic optimisation problem.

One of the most under reported components within research of this nature is the characterisation of infeasible designs, which often result in overly restricted design spaces; the results from which cannot be considered globally optimal. This work imposes a number of constraints that define feasibility, which have once again been derived in conjunction with the aforementioned aerofoil database. This allows a large, continuous design space comprised of feasible and infeasible regions to be implemented, defining optimality as a two variable problem encompassing sum of constraints and cost function minimisation.

Two optimisation problems have been carried out between particle swarm and gradient based optimisers. The first looked at a highly constrained design space based on a close to optimal baseline configuration, with the second allowing a wider scope of designs to be considered optimal, using only generalised constraints to allow for full exploration of the design space. The baseline case showed the developed framework's applicability to typical design iteration optimisation, with appropriate constraints driving the optimiser toward an improved but similar design.

An important distinction is made between this approach and local optimisation techniques which use the baseline design as a starting point. That is the latter method provides little scope to vali-

date the original design, merely improve upon it. The former however, is only guided by baseline constraints, not the design itself. Here, the optimiser is able to find radically different designs that satisfy the geometric and performance characteristics of the baseline, and therefore, consistent convergence towards a similar shape shows that the initial design is in fact close to optimal for the given problem.

The second optimisation problem removed the baseline design and looked to optimise from an arbitrary starting point. Here, aerodynamic efficiency was to be maximised, since there was no baseline lift coefficient to maintain. A generalised minimum area, peak thickness, and leading edge radius were derived for existing aerofoil shapes, with baseline independent constraints imposed as in the previous case.

In both cases, the SOPSO method was a clear improvement over the multi-point SQP approach, with all 10 runs producing improved objective function values over any of the 100 final SQP designs. Furthermore, very little variation in both cost function and design shape was observed across SOPSO runs, showing consistent tendency towards globally optimal results in both highly constrained design spaces and large variance, low objective function spaces.

These results demonstrate the SOPSO algorithms fitness for purpose within this research scope, with repeatable convergence characteristics over a number of optimisations. One aspect of this work that has already been discussed as a limitation is the analysis methods employed; being of low-fidelity and therefore not providing a full picture of high-speed flow physics. With the optimiser validated for single fidelity analysis, one interesting avenue that can also be explored, and continues to be a popular topic in aerospace research, is the employment of a second, higher fidelity analysis method. Aerofoil optimisation such approaches to be implemented with relative ease in comparison to more complex 3D geometries, and has therefore been utilised in the current research.

Chapter 5

Multi-Fidelity Global Aerofoil Optimisation

5.1 Introduction

While low-fidelity optimisation allows for an effective large design space exploration, the resulting optimal solutions may appear sub-optimal with increasing fidelity. Since any optimisation method will exploit any flaws in an ill-constrained design space, the same can be said for a model which does not properly capture the relevant physics. Meaningful results can be drawn from low-fidelity optimisations at the early phases of design, however efficient and reliable implementation of higher fidelity methods can only improve the quality of results.

Even in an idyllic situation with an abundance of computational resources, CFD packages cannot be applied arbitrarily to any problem. This is due to convergence criteria being highly sensitive to the geometry and flow domain meshing. Automated meshing is a continually researched field, with even relatively simple geometries still requiring iterative mesh convergence regimes to be carried out by engineers on a case by case basis. Thus, their inception into a large design space, exploratory optimisation search is far beyond the capabilities of the current state-of-the-art. On the other hand, if the geometry can be reasonably maintained; either by limiting the design space or somehow ensuring that poorly behaved geometries are not passed over to the high-fidelity solver, then multi-fidelity techniques can improve results substantially. The vast majority of research focusses on a limited design space, often utilising deformative methods on an existing aerofoil to be optimised [80, 108]. Recent research by He et al. [109] is an exception to this however, demonstrating almost identical optimised shapes from starting points of random aerofoils as well as that of a circle. Similarly, this research looks to produce optimal designs from a large design space, although through generative methods and no starting point, with poor designs being quantified as such through robust constraints.

Aerofoil sections provide ample opportunity to demonstrate the effectiveness of multi-fidelity techniques in an exploratory design optimisation, since sections can be non-dimensionalised to a constant chord, forming some consistency between designs. Nevertheless, when allowed to explore large design space, aerofoil generation methods can produce poorly behaved, or completely infeasible shapes, such as intersecting upper and lower surfaces. Thus, ensuring only well behaved designs are passed to the CFD solver is of paramount importance, and techniques must be employed to ensure reliable results can be drawn from the highly mesh dependent high-fidelity model. Further methods must be put in place to limit the overall number of designs being analysed by the CFD solver, due to the drastic increase in time taken to complete a single function call compared to the low-fidelity model. Since the heuristic optimisation methods used will make upwards of 10^4 function calls in a given run, high-fidelity calls must be utilised as efficiently as possible.

A Gaussian based surrogate model is employed to achieve this, allowing predictions to be made of the high-fidelity cost function using samples exclusively from non-violating subspaces, significantly reducing the computational power required. To ensure that only non-violating geometries are analysed, along with efficient usage of a computationally heavy analysis method, a two step design of experiments (DOE) approach is employed, resulting in the surrogate model being built from the error between low and high-fidelity cost functions. This allows the optimiser to run primarily in low-fidelity mode, while predicting the error with respect to high-fidelity analysis. To ensure the model is producing accurate results, high-fidelity cost function calls are made selectively throughout the optimisation, and the surrogate model re-fit based on the updated data. This will be discussed further in section 5.3.1.

In a similar manner to the low-fidelity optimisations discussed previously, multi-fidelity baseline and arbitrary optimisations are carried out. Discussion of the high-fidelity analysis methods employed, and their autonomous implementation within the optimisation framework will be provided. Furthermore, the method used to create an initial design of experiments and surrogate model will be introduced, along with a discussion surrounding the selection of high-fidelity cost function calls during an optimisation, and how the surrogate model is updated to reflect this additional data.

The overall goal is to have optimality defined entirely by high-fidelity analysis, with low-fidelity methods and surrogate model working in tandem to select suitable candidates for such analysis. To provide a direct comparison, the optimisation problems remain identical to the single-fidelity baseline and arbitrary cases carried out in sections 4.5.4 and 4.5.5.

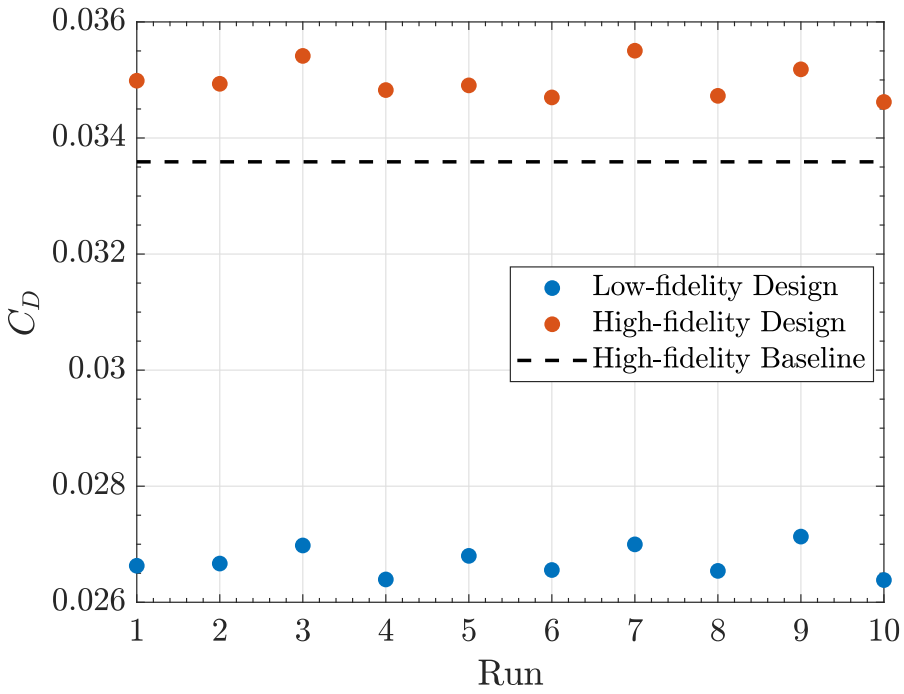


Figure 5.1: Low-fidelity optimisation results analysed at high-fidelity with baseline comparison

5.1.1 Motivation

To provide some context, the optimal aerofoils found in the low-fidelity baseline optimisations (section 4.5.4), were analysed using a high-fidelity solver, along with the baseline design itself. The resulting high-fidelity cost function results are shown in figure 5.1.

These results paint a very different picture to those in chapter 4 (figure 4.4), which showed a consistent improvement over the baseline in terms of drag coefficient. Using high-fidelity CFD analysis, the inverse is in fact true, with inaccuracies from the low-fidelity solver producing misleading results. Clearly then, improvements over the baseline case can only be considered true if they have been validated by high-fidelity analysis. Referring again to the optimal aerofoils and respective cost function values in section 4.5.4, the high convergence characteristics across runs increases confidence that the region of global optimality has been found, in terms of low-fidelity analysis. Thus, it would be counterproductive to continue conducting high-fidelity analysis after a low-fidelity optimisation has been carried out, since it will continue to tend towards this region of inaccuracy. Instead, high-fidelity analysis must be incorporated within the optimisation process, so that these inaccuracies can be quantified, and the optimiser can shift it's focus towards more promising regions of the design space.

5.2 High-fidelity Analysis

The Stanford University open-source SU2 code was selected to conduct CFD analysis within the optimisation process, due to its demonstrated capability in the supersonic domain, and implementation that lends itself to automated use in frameworks such as this [110].

Nonetheless, incorporation of CFD analysis within a global optimisation context is a difficult task, requiring automated meshing between dissimilar shapes. Steps are therefore taken to only allow well-behaved geometries to be analysed, which will be discussed further in section 5.3.1. Even with such checks in place, a wide variety of aerofoil sections can still be selected for high-fidelity analysis, necessitating robust grid generation. The open-source meshing tool GMSH has therefore been chosen for this task, which provides robust automatic mesh generation tools given overall flow domain geometry and target length scales at both body and boundaries.

The interfaces created to enable autonomous mesh generation and CFD analysis will be detailed, providing typical settings and key parameters that allow their unsupervised use both in the design of experiments phase and during a multi-fidelity optimisation.

5.2.1 GMSH Interface

GMSH is an open-source finite element mesh generation tool [111], capable of producing detailed geometry surfaces and flow domains from a small number of user provided parameters. A simple input script, containing known points and the desired mesh refinement at each location, along with utilisation of user-friendly GMSH functions to turn points into closed surfaces, and specification of the meshing techniques to be employed, makes GMSH a powerful and reliable tool to be incorporated in multi-fidelity analysis.

Two sets of points must be transferred to the GMSH input file for the current problem; providing sufficient detail to fully characterise aerofoil geometry and flow domain. Each input point requires four parameters to be defined: x, y, z spacial coordinates, and a characteristic length, which determines the level of discretisation desired when connecting points into a single surface. GMSH then utilises line, spline, and arc functionalities to create and close these surfaces independently, leaving the space between surface and boundary, otherwise known as the flow domain, to be defined. A number of different meshing algorithms can be employed to achieve this. Importantly however, it should be noted that the previously defined characteristic length at each surface point plays a large role at the meshing phase, since the 2D flow domain mesh must conform to the already generated 1D surface mesh. This is of course the requirements of any CFD solver, demanding accurate surface geometry which in turn defines the level of discretisation of the flow domain close to the surface. While an accurate representation of the geometry

to be analysed is imperative, the boundaries of the flow domain require much less attention. These boundaries should theoretically be far enough away from the analysed geometry that the flow remains at freestream conditions, unaffected by the geometries presence. Because of this, and since the scale required should result in large, easily discretised geometries, a considerably higher characteristic length compared to that of the analysed geometry can be specified. Nonetheless, care must be taken to avoid meshes that stretch too quickly away from the surface, which will poorly characterise the flow surrounding the body.

With body and boundaries fully defined, along with respective characteristics lengths, the final step is to mesh the intermediate flow domain region. This is usually a task that requires hands on analysis of the generated meshes on a case by case basis. As this approach attempts to utilise meshing and CFD tools autonomously, some simplifications and handling of poorly geometries or output analysis must be implemented. These will be discussed throughout the proceeding sections, but in terms of meshing, the simplification that allows these methods to be carried out autonomously is the lack of viscous analysis. The decision to analyse aerofoil sections in terms of inviscid aerodynamics exclusively was discussed in section 4.5.3, citing little importance within the overall research scope. In terms of CFD analysis, accounting for viscous forces requires the boundary layer surrounding the body to characterised in great detail. While this is a relatively simple job for single cases, it would take a large effort to develop an algorithm capable of consistently and autonomously capturing the boundary layer for arbitrary aerofoil shapes.

Producing a robust and consistent flow domain across a variety of potential shapes without the need to capture the boundary layer presents a much simpler problem, relying primarily on the characteristic lengths defined on the body and flow boundaries. From here, the meshing algorithms of GMSH take over to define the flow domain. Many of the algorithms have been experimented with, which led to an initial triangulation followed by the blossom recombination algorithm to be chosen [112]. For the given problem, this results in an almost structured like quadrilateral mesh, as shown in figure 5.2. Using this method, a grid independence study was carried out, detailed in section 5.2.4, to ensure that accurate results could be drawn, and increase confidence that such a method could be utilised autonomously within the overall optimisation framework.

5.2.2 SU2 Interface

To carry out aerodynamic analysis on the GMSH generated meshes, the Stanford University open-source CFD code SU2 [110] was employed. The development of SU2 has been intentionally modular, promoting ease of implementation, with one of the key goals being its application to design optimisation.

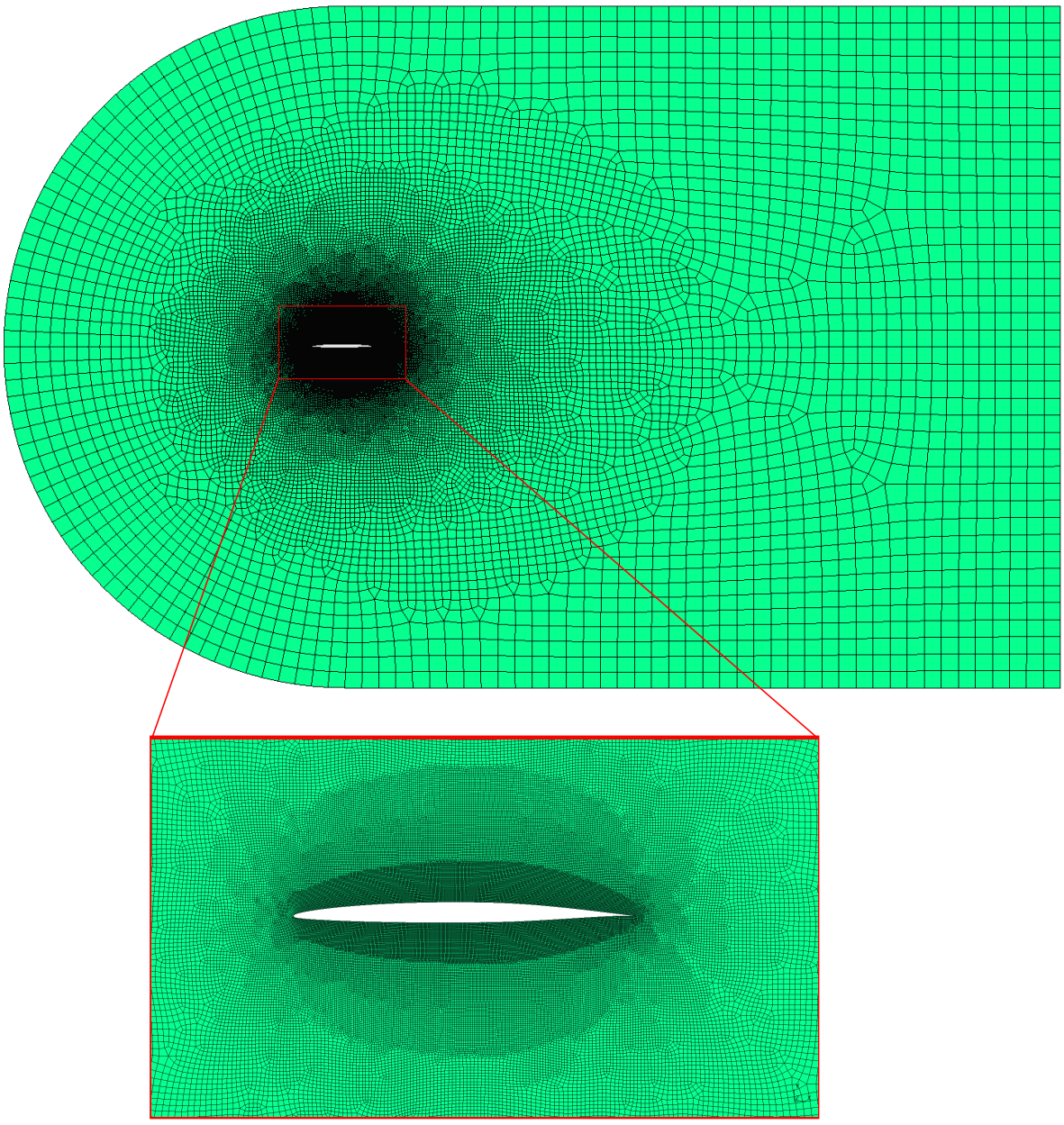


Figure 5.2: NACA66-206 example aerofoil and surrounding domain mesh

Interfacing within SU2 is therefore a relatively simple procedure, requiring only a few parameters to be altered from run to run, which allows multiple flight states and geometries to be analysed with minor input modifications. With the mesh already defined from the GMSH interface, SU2 only requires freestream flow parameters and solver to be defined. The latter is a constant Euler method, which is typically used for high-speed inviscid flows. Freestream flow parameters are consistent with those used in low-fidelity analysis, which can simply be updated within the SU2 input file between runs to represent different flight states.

SU2 outputs a log file of results with respect to number of iterations. The output parameters are solver based, such as residual values and performance characteristics, eg. lift and drag coefficients. The interface reads from the final iteration of this result file, and tabulates results in a consistent manner to low-fidelity analysis. The solutions convergence criteria is also saved, as some geometries or meshes may fail to converge for a given flight state. This will be discussed further in the proceeding section.

5.2.3 Implementation to Overall Framework

During an optimisation, criteria defined in section 5.3 switch the optimiser from default low-fidelity mode to carry out high-fidelity analysis on an aerofoil within the population. A secondary cost function is invoked in which the aerofoil is passed to GMSH and SU2 respectively, before outputting performance characteristics that can be directly compared to the low-fidelity analysis.

Aerofoils are already defined from upper to lower trailing edge in a consistent manner for low-fidelity analysis, meaning they can be directly input into GMSH, which then passes a spline through each point to fully define the surface. A bullet domain surrounds the aerofoil far-field. This allows the body x-axis to be aligned with the x-axis of the flow domain, and angle of attack modelled through a rotation of the freestream velocity vector, similar to that used in low-fidelity analysis. This also allows the aerofoil to be analysed at multiple angle of attack values while using a single mesh throughout. The flow domain size is driven by the bullet radius, r_{dom}, in which the aerofoil geometry lies at the centre. This semi-circle sits upstream of the aerofoil, with downstream flow boundary set at r_{dom} from the trailing edge, providing a large region for the shed wake from the aerofoil to dissipate. The flow domain size with respect to 1m chord aerofoil can be visualised in figure 5.2. Note that a single value for characteristic length is employed for flow domain, and aerofoil surface respectively.

Once generated, the mesh can be directly export from GMSH as a .su2 file, allowing direct usage within the SU2 solver. The freestream flow parameters are then updated within a template aerofoil input script, after which analysis is carried out. Termination criteria is defined by a Cauchy

Table 5.1: NACA66-206 domain size independence study ($L_{body} = 0.001m, L_{dom} = 0.1m$)

r_{dom}	C_L	C_D	C_M	Time, min
$5c$	0.136307	0.033465	0.037227	10
$10c$	0.136312	0.033473	0.037227	21
Δ	0.0036%	0.023%	-0.0008%	110%

series on the drag coefficient, where a tolerance of 10^{-6} is applied across 100 iterations. If the average change in drag coefficient is below this value over the specified number of iterations, analysis will terminate. A secondary termination criteria is applied in the form of maximum iterations, which is set to 2000. In this case, two further steps must be carried out, to ensure results properly interpreted. Firstly, instead of taking performance criteria from the final iteration, they must be averaged over a set number of iterations, which is once again set to 100. Secondly, residuals must be analysed in order to determine overall convergence. This is done in the form of an additional optimisation constraint, in which output residuals must fall below 10^{-5} to be considered accurate in the context of a design optimisation.

5.2.4 Grid Independence Study

Before employing the chosen grid generation and high-fidelity analysis methods within the optimisation framework, a grid independence study was carried out, once again using the NACA66-206 aerofoil. The automation of GMSH based on a few key parameters allows a large variety of meshes to be tested, which can then be compared against similar CFD analysis introduced in section 2.6 [39]. Three parameters are varied in this study: characteristic lengths of both aerofoil surface and flow domain, and radius defining the far-field size. With this being an inviscid Euler study, and mesh optimisation already carried out by GMSH, no further parameters are required to complete this study.

Domain Size

Firstly, the influence of far-field position was tested, using values of characteristic length that were assumed to overestimate the required level of mesh fineness throughout the flow domain - this hypothesis will be confirmed in the characteristic length independence study. The radius of the upstream flow boundary was initially tested at $5c$, allowing typical CFD standards to be met downstream also ($15c$). This radius was then doubled to check for discrepancies between runs, which if present would highlight that the $5c$ case was not capturing enough of the flow domain to produce reliable results. Absolute force coefficients, along with percentage differences are shown in table 5.1. The small change in coefficients across the two runs ($\ll 1\%$) confirm that the $5c$ case models enough of the flow domain to extract meaningful results.

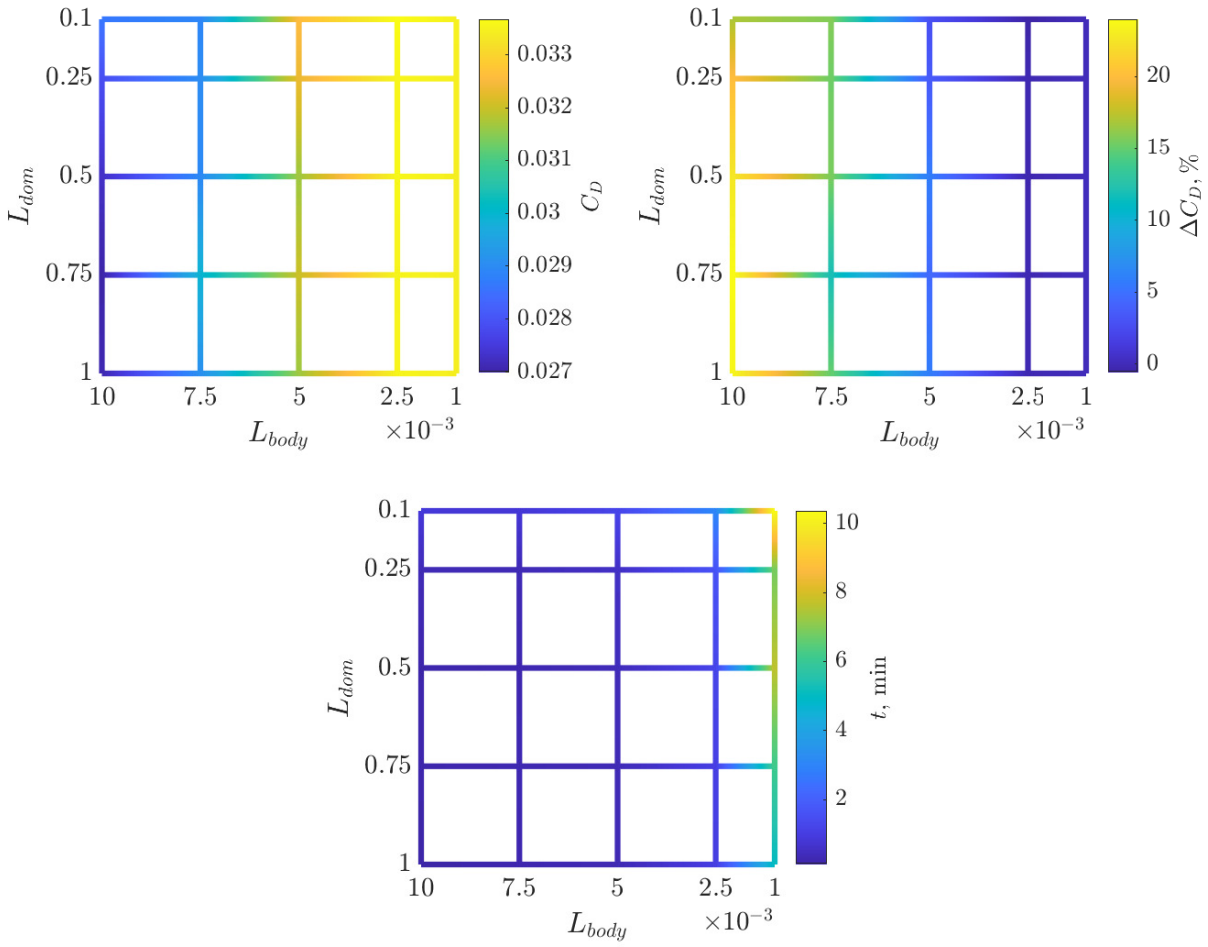


Figure 5.3: NACA66-206 grid independence study shown in terms of absolute drag coefficient (top left), percentage difference with respect to finest mesh (top right), and time to complete analysis (bottom)

Characteristics Lengths

The flow domain radius was then held at $5c$ while characteristic lengths of aerofoil surface and flow domain were independently varied. Five cases were tested for each parameter, resulting in a grid of 25 total runs. Analysis began from the coarsest mesh ($L_{body} = 0.1$m, $L_{dom} = 1$m), with characteristic length being reduced in one of the two dimensions for each consecutive case, to a maximum mesh fineness generated from $L_{body} = 0.001$m, $L_{dom} = 0.1$m. Figure 5.3 shows the resulting grid in terms of absolute and percentage difference drag coefficient, along with time to complete analysis.

As expected, results have a considerably higher sensitivity to mesh characteristic length at the aerofoil than that of the flow boundary. Absolute C_D shows very little change with respect to L_{dom}, with values converging for all $L_{body} \leq 2.5 \times 10^{-3}$ Nonetheless, at coarse body representation, effect of domain discretisation is shown in terms of drag coefficient percentage difference,

with accuracy increasing by over 10% from coarse to fine. This demonstrates the need to avoid overly stretching cells from aerofoil surface to domain boundary, in favour of a more consistent overall discretisation. The convergence of drag coefficient at refined meshes also agrees with the previously discussed CFD data produced by Giles and Marshall [39], providing further confirmation that the solver and mesh techniques are appropriate for the given problem.

Time to complete analysis is of course non-linear, with number of mesh cells being highly sensitive to increasing characteristic length scales. While the relative difference in time to complete analysis is high, even the finest mesh can be analysed in a relatively short time compared to the majority of external aerodynamic CFD applications. It is important to reiterate at this point that the purpose of this grid independence study is not to find the combination of mesh generation parameters that provide optimal computational efficiency, while maintaining high accuracy, for this particular aerofoil. Instead, the goal is to employ mesh parameters that can create accurate flow domains for the wide variety of aerofoils that may be passed to the high-fidelity optimiser.

Thus in this work, a high degree of conservatism must be employed when deciding mesh generation parameters, to maximise the likelihood of accurate, convergent results being output, regardless of the input aerofoil. Onus must therefore be placed on mesh quality and refinement, rather than computational efficiency. This is necessary to ensure the optimisation is not misled by a large amount of noise caused by poor convergence characteristics of the CFD solver. Furthermore, the selective nature of multi-fidelity methods may only require somewhere in the region of 100 high-fidelity cost function calls, including those required to build a design of experiments. This appropriates the use of a conservative approach considering the overall computational cost considerations, with highly refined meshes demanding relatively little time to converge with respect to number of high-fidelity function calls in a given optimisation. Note that no amount of conservativism could guarantee convergence for autonomous implementation with regards to the current state-of-the-art, and many generated aerofoils within the design space would result in the CFD solver diverging. Therefore, as will be discussed in section 5.3, geometric constraints must not be violated in order to allow high-fidelity analysis of a candidate aerofoil.

From the data gathered across far-field size and characteristic length grid independence studies, the chosen mesh generation parameters are: $r_{dom} = 5\text{m}$, $L_{body} = 0.001\text{m}$, and $L_{dom} = 0.2\text{m}$. These parameters result in a mesh that is considerably more refined than necessary for the NACA66-206 case, but have been chosen with the aforementioned degree of conservatism, to increase the probability of convergence across the large aerofoil design space. Figure 5.4 shows the flow field in terms of Mach number using this mesh, clearly showing the resolved detached leading edge and attached trailing edge shockwaves. In the far field, the clarity of these waves reduces due to the increased characteristic length used for the domain boundary. However, the

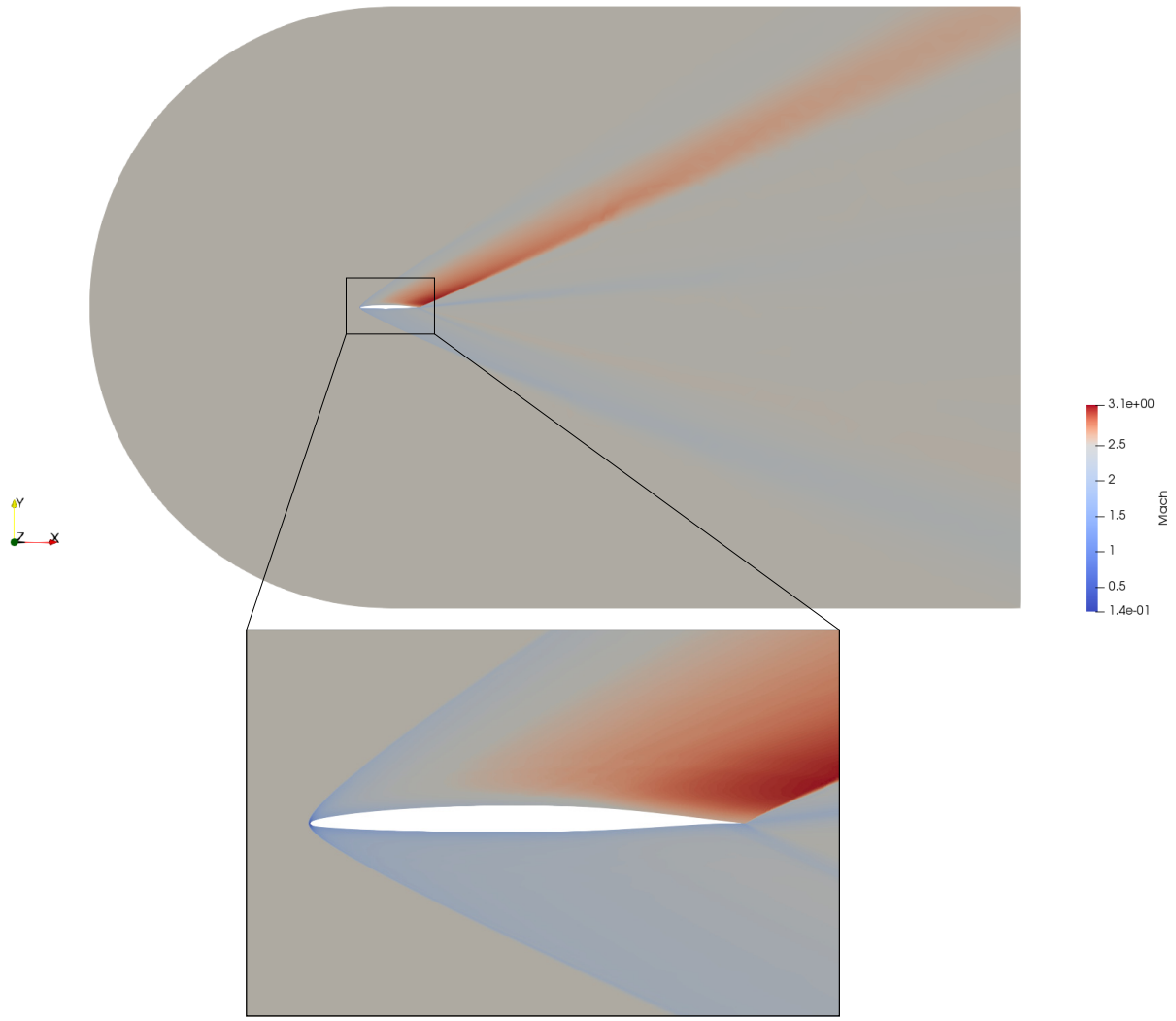


Figure 5.4: Flow visualisation for $M_\infty = 2.5$, showing Mach number across domain and over aerofoil surface

grid independence study shows that this has little to no effect on the pressure forces, and so further refinements of the far field were deemed an unnecessary computational expense.

5.3 Surrogate Modelling

Surrogate models attempt to predict the value of a function at a requested point through a number of pre-analysed data points. These data points represent true function values given a set of inputs, which when combined can be used to estimate the function value given a set of inputs that have not been parsed through the function. This is particularly helpful for functions that require a large amount of processing power to carry out a single analysis. Furthermore, many such functions do not require all combinations of inputs to be analysed, but rather the most promising areas to be found. This is the case in optimisation work, where inputs that result in poor cost functions are discarded in favour sets with improved performance.

Moreover, in terms of the previously described aerodynamic prediction methods, the difference in time to conduct analysis between low-fidelity panel methods and high-fidelity CFD is multiple orders of magnitude apart. With the large design space employed, which contains many undesirable aerofoil shapes; and population based optimisation methods, which analyse hundreds of designs at multiple flight states per iteration, analysing each aerofoil with CFD software would be an enormous waste of computational power. Thus, in order to conduct a global multi-fidelity aerodynamic design optimisation, some form of surrogate model must be utilised to make predictions of the high-fidelity CFD analysis method.

This research employs Gaussian process regression (GPR) to achieve this, which is a generalisation of a Gaussian probability distribution function [113]. Here, a number of finite data points allow predictions to be made anywhere within the function. Of course, being a Gaussian based method, the accuracy of this process will be highly dependent on the number of data points, and their proximity to, the set of input variables for which a prediction of the function has been requested.

The primary reason for employing this method stems from the design space containing regions of infeasibility, defined by the geometric constraints. Some surrogate models require a global view of the search space in order to produce a prediction, however it would not only be unnecessary to analyse these infeasible regions, it would most likely result in a poor overall model. The geometric constraints have been derived to speed up convergence of the algorithm towards feasible designs, but in the case of CFD analysis, which is highly sensitive to mesh quality, they also increase the probability of convergence.

The GPR implementation adopted here only allows non-violating geometries to be analysed by the high-fidelity solver, focussing computational resources on areas of the design space that are genuine areas of interest. Importantly, it must be clarified that this specific modification to the GPR approach cannot result in feasible and high-performing areas of the design space to be overlooked, since these geometric constraints are independent of aerodynamics and therefore level of fidelity. Generally of course, no surrogate model guarantees that the global cost function minimum will be found, with results being highly dependent on the number and spread of sample data points. Methods to increase the likelihood of finding optimal solutions are discussed further in section 5.3.1.

Gaussian process regression is well established technique within machine learning, with a number of open-source and commercial algorithms available for use. MATLAB's fitrgp function [114] was therefore utilised in this research, allowing development efforts to be focussed on

modules not readily available in the wider research community. Moreover, the underlying theory behind this method will not be covered here, with interested readers directed towards MATLAB's toolbox [114], and the dedicated works of Rasmussen and Williams [113].

5.3.1 Design of Experiments

Before a multi-fidelity optimisation can be carried out, a surrogate model must already be in place, to allow the high-fidelity cost function to be predicted throughout. To create this, an initial design of experiments (DOE) must be analysed. The variables used for this DOE, along with output results, can then be regressed to produce a Gaussian model of the high-fidelity cost function. As in population initialisation (section 3.3.2), selection of variable sets to be included in the DOE requires careful consideration. In fact, this is even more crucial here, with a poorly defined DOE causing inaccurate predictions, and potentially large regions of high performing design space to be overlooked entirely. Extra steps are therefore taken in DOE creation, to ensure a diverse and feasible set of aerofoils are used to generate the surrogate model.

Advice is taken from Eldred & Dunlavy [115], and Rajnarayan [116], to conduct the design of experiments across both fidelities, and build the surrogate model from the difference between their resulting cost functions. Furthermore, additional surrogate models are produced in the same manner for any performance based constraints, as these will also vary between low and high-fidelity simulations.

Before selecting samples for high-fidelity analysis, a discussion is required around infeasible regions of the design space (those that violate geometric constraints), and how to avoid their implementation to the high-fidelity module. Such designs may have cost function values that appear attractive to the low-fidelity solver, but their geometric properties make them unsuitable for real-world application. To eradicate the potential of such designs being selected for a high-fidelity simulation, constraints are separated into two subgroups: geometric and performance based. Since the geometric constraints values are independent of analysis fidelity level, they essentially provide an automatic determination of design feasibility. By only allowing designs which fall below a specified geometric constraint tolerance to be analysed by the high-fidelity solver, this is ensures that such computationally expensive simulations are focussed on close to feasible designs. Furthermore, the geometric constraints produce a higher probability of a successful, converging CFD solution.

To create the surrogate model, an initial design of experiments must be carried out on the high-fidelity solver. Typically in optimisation literature, this would be achieved directly via a similar method to population initialisation (section 3.3.2), such as Latin hypercube sampling; which would select a number of design points throughout the hyperspace. There is no set rule as to

how many design points are required to build a surrogate model. However, undoubtedly the number of samples should scale with number of design variables [97]. The resulting locations would be implemented directly into high-fidelity analysis, and in turn used to create the surrogate model. In this case however, where the design space can produce infeasible designs, any form of random sampling cannot be applied. Fortunately, geometries can be built and tested for their validity in rapid fashion. Thus, an initial and considerably larger sampling than that required to create a multi-fidelity design of experiments is extracted from the design space, tested for validity, and feasible solutions selected for high-fidelity analysis.

This essentially creates a two-layered design of experiments, where candidate aerofoils must perform an initial feasibility check before selection. With the number of design variables from low-fidelity optimisations set at $n_{var} = 16$, the number of designs within the initial DOE is limited to $n_{DOE} = 5n$, resulting in 80 aerofoils to be analysed by both low and high-fidelity analysis methods. Selecting 80 non-violating design points could result in an overly biased DOE, where feasible portions of the design space are missed. To combat this, the number of initial feasibility samples is order of magnitudes larger, resulting in $n_{feas} >> n_{DOE}$, where n_{feas} is the total number of samples that do not violate any geometric constraints. Reducing the number of feasible designs to n_{DOE} then becomes a problem of maintaining diversity. While these designs could be randomly sampled again, this does not minimise the risk of having feasible tranches within the design space omitted from the DOE.

Instead, a crowding distance technique employed, similar to that used when Pareto fronts expand beyond their maximum allowable population, described in section 3.3.4. One key difference however, is that Pareto front crowding distance is computed from the cost function space, whereas DOE samples are chosen through design space crowding distance. The reason for this difference is clear: Pareto fronts are cost function based, whereas the DOE seeks to encapsulate as much of the feasible design space as possible, before any high-fidelity cost function analysis is conducted.

Even with such a large number of samples, the feasibility check requires such little computational power that reduction from thousands of designs to n_{DOE} designs takes around a minute on a single processor core. This is a highly efficient method of sampling the design space for feasibility, and selecting a highly diverse set of samples for DOE analysis. With designs chosen, low and high-fidelity analysis is then conducted, producing a set of cost and performance based constraints for each method. The delta between analyses, along with design variables, is what creates the initial surrogate model.

5.3.2 Optimisation Usage & Dynamic Model Updating

The design of experiments is defined pre-optimisation, as is the resulting surrogate model. The incorporation of such a model enables predictions of the high-fidelity cost function and performance constraints to be made based on design variables and low-fidelity results. The goal is to have the optimisation driven by the highest fidelity analysis, and therefore at each iteration, a prediction is made for each particle in the current population, defining the delta between low and high-fidelity performance metrics. These delta values are then added to particle cost and constraint functions, allowing local best particles to be chosen based on a high-fidelity estimation.

In terms of global best, high-fidelity analysis should be conducted on candidate designs, rather than based purely on prediction. The DOE is not intended to be all-encompassing from the outset, otherwise optimisation would not be required, and a simple query to the model giving the most promising high-fidelity prediction would be all that is necessary. Instead, predictions are made at every iteration for each particle in the population, with a perceived improvement in the global best requiring validation through a high-fidelity cost function call. By carrying out high-fidelity analysis on the predicted global best, this ensures that the optimisation is always pointing towards the best observed design in the high-fidelity cost function space. Regardless of whether this confirms or rejects the newly perceived global best, it provides an additional data point for the surrogate model to predict the true cost function with increased accuracy. In fact, this is particularly useful when a perceived global best is rejected after high-fidelity analysis, highlighting an area of inaccuracy within the model.

If such inaccuracies are not rectified at the time of discovery, the optimisation will continue to focus on the same areas of the design space, when in fact a change of direction is required. Therefore, if the model prediction is not within a given tolerance of the true high-fidelity cost, the surrogate model is re-fit. Accurate predictions are still included when the model is updated, however only doing so when an inaccuracy is found allows a reduced number of fittings for a given optimisation. This dynamic model updating guarantees that the surrogate will reflect the current understanding between low and high-fidelity cost functions during the optimisation, allowing the surrogate to gain further accuracy in areas of interest, and shifting or maintaining focus on them as required.

These inaccuracies can also manifest in under predictions of high performance designs, due to lack of surrounding design points within the DOE. The previous discussion deals with the inverse scenario, since over predictions characterising non-optimal designs as optimal will be analysed using the high-fidelity cost function, which then serves as a new data point within the surrogate model. However, optimal designs that are under predicted may never be analysed using high-fidelity analysis in a given optimisation, since it appears sub-optimal to the surrogate.

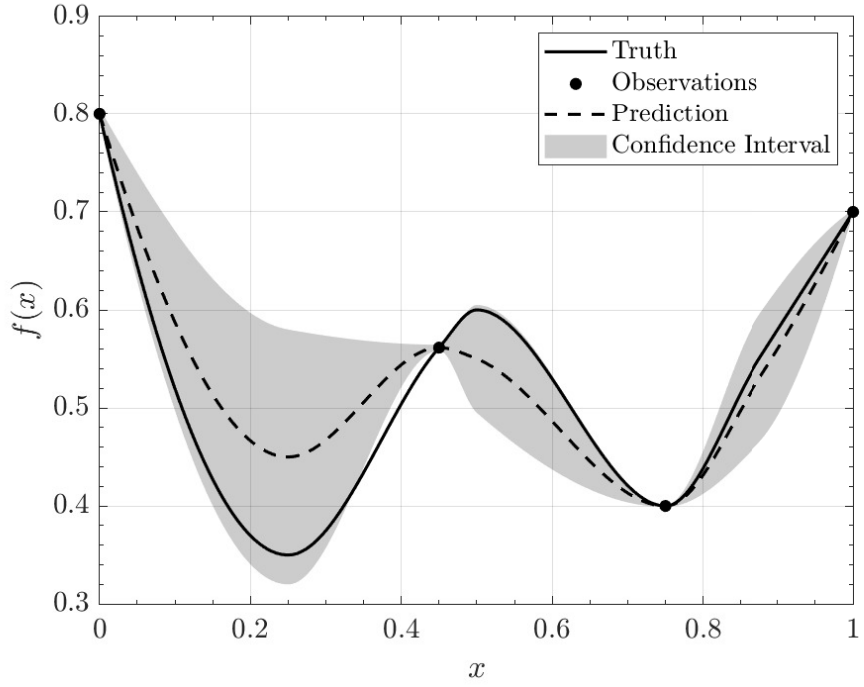


Figure 5.5: Single variable surrogate model illustration, with observations used to predict overall function with confidence boundaries

This inevitably leads to a prematurely converging optimisation, with improved designs unable to be selected for high-fidelity analysis due to lack of surrounding data points, and therefore conservative or poor performance predictions.

To combat this, it is logical to not rely solely on the absolute prediction output by the surrogate, and utilising confidence intervals readily available from Gaussian processes. The lower confidence boundaries, representing the potential best performance of each design, are tended towards once the simulation begins to stall. In a similar fashion to the variable inertia discussed in section 3.3.2, increasing the stall counter will slowly tend predicted costs towards the lower confidence boundary. This allows areas of the design space that are under represented within the surrogate model, and therefore wide confidence intervals, to be selected for analysis. Quantification of true performance from areas of poor prediction allows a deeper understanding of the design space to be gained, and importantly, optimal designs to be characterised as such.

Figure 5.5 shows an illustrative surrogate model, predicting the true cost function from a selection of observations. The resulting predicted function is shown with confidence intervals, with observed points bounding regions of uncertainty. As expected in a Gaussian process, confidence in prediction decreases as distance away from the closest observation increases. Thus, adding samples in these highly uncertain regions will result in a drastically improved model. Figure 5.5 also demonstrates the aforementioned problem of relying solely on the predicted cost, since in

the models current form, the minimum point is over predicted. With a sample already taken at the local minimum around $x = 0.75$, only areas with a lower prediction will be selected for as new observations, unless the lower confidence boundary is incorporated. By tending towards this boundary while the optimisation stalls, it ensures samples will be taken at poorly refined areas, such as the minimum point in this illustration.

Note that the stall counter will continue to rise if a design which appears optimal due to low predictive confidence, returns a true cost function value that is worse than the current global best. In such cases, predicted cost values will continue to tend towards lower confidence boundaries. This is the intended behaviour, resulting in frequent high-fidelity sampling during prolonged periods of stall, until a new global best design is found.

5.4 Multi-Fidelity Optimisation

The optimisation problems described in sections 4.5.4 and 4.5.5 are recast here as multi-fidelity optimisations, utilising the methods described in this chapter. An initial DOE of $5n_{var}$ samples is selected using the two level method described in section 5.3.1, with designs being pick based on design space crowding distance from non-violating members of a considerably larger initial sampling. The surrogate model is then created from these observations, which provides deltas to predict CFD force coefficients from low-fidelity analysis. High-fidelity runs are carried out for every optimisation iteration in which the predicted global best is an improvement over the current best, obtaining the true cost function for verification and incorporating an additional data point within the surrogate. Throughout the optimisation the model is re-fit, ensuring focus is placed on promising regions of the design space. Furthermore, premature convergence is avoided by iteratively reducing the high-fidelity analysis criteria while the optimisation is stalling, allowing unexplored, high cost uncertainty spaces to be properly quantified.

Once again each optimisation is carried out 10 times, to provide a clear view of performance, both in terms of consistency across runs, and to allow appropriate comparisons to be drawn between methods. Two DOEs are created, one for baseline case and one for global optimisation, as initial DOE selection criteria varies between the problems. This means that each run uses the same initial DOE, rather than reinitialising at the beginning of every optimisation.

5.4.1 Baseline Driven Results

As before, overall optimal aerofoil, variation across runs, and baseline comparison are shown in figure 5.6. Multi-fidelity results across runs are shown in comparison to single-fidelity results in chapter 4, along with the baseline design.

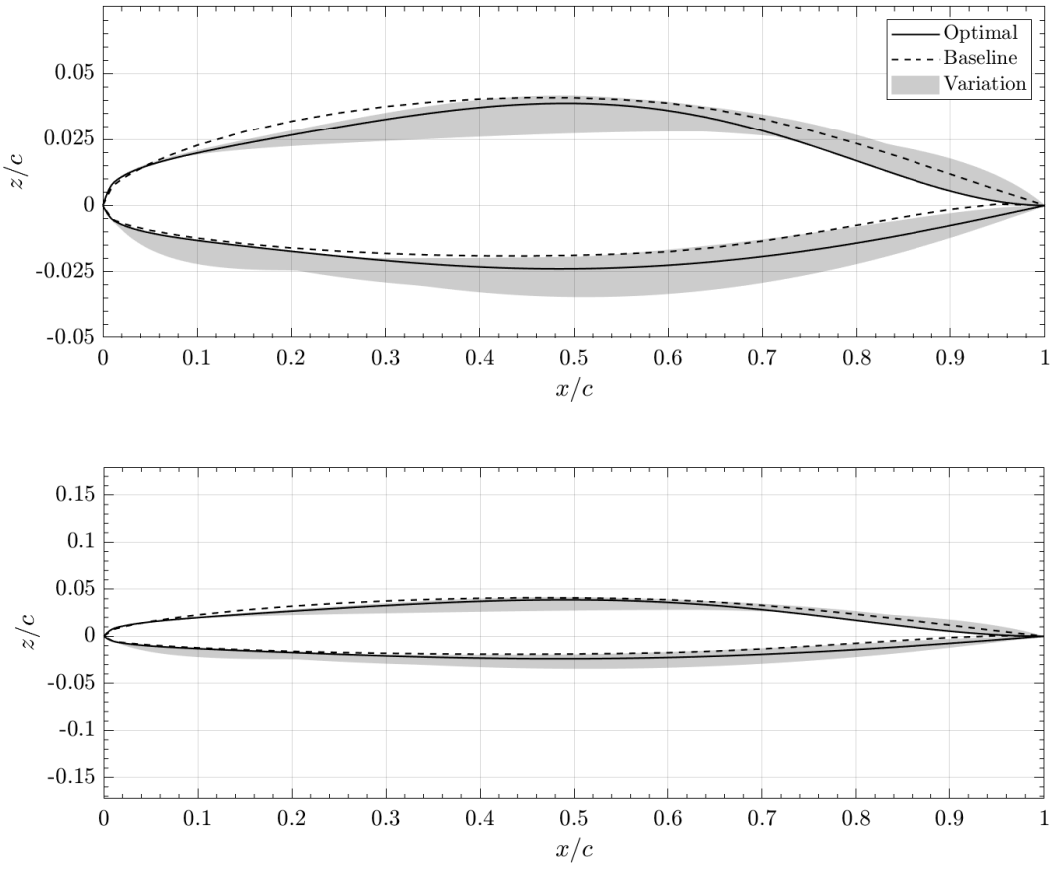


Figure 5.6: Multi-fidelity optimal design and variation across runs with baseline comparison

The minimum C_D aerofoil, found through incorporation of surrogate model lower confidence boundaries, is a considerable improvement upon all other methods and the baseline. Interestingly, the defining feature of this aerofoil is its flicked tail on the upper surface, which lies on the boundary of optimal aerofoils found throughout the 10 runs. The difficulty in consistently finding this particular shape may be explained by two features of the optimisation problem. Firstly, since the aerofoil must at least maintain the cross-sectional area of the baseline, producing a concave rear shape requires additional area in front of this region, which may appear counter-intuitive to the optimiser. Secondly, to achieve this shape, a large counterbalance is required in terms of control points. This is illustrated in figure 5.7, with drastic changes in consecutive upper control point heights required, demonstrating the need for a large design space to achieve the adequate level of control. The pulling nature of Bézier methods require control point boundaries to be considerably wider than maximum allowable surface boundaries, otherwise the potential and benefit of using such a technique is lost. The concave rear surface of the optimal design can only be generated with a large delta between the final two free control points, since the $x = 1$ control point is fixed, highlighting the design space scope required.

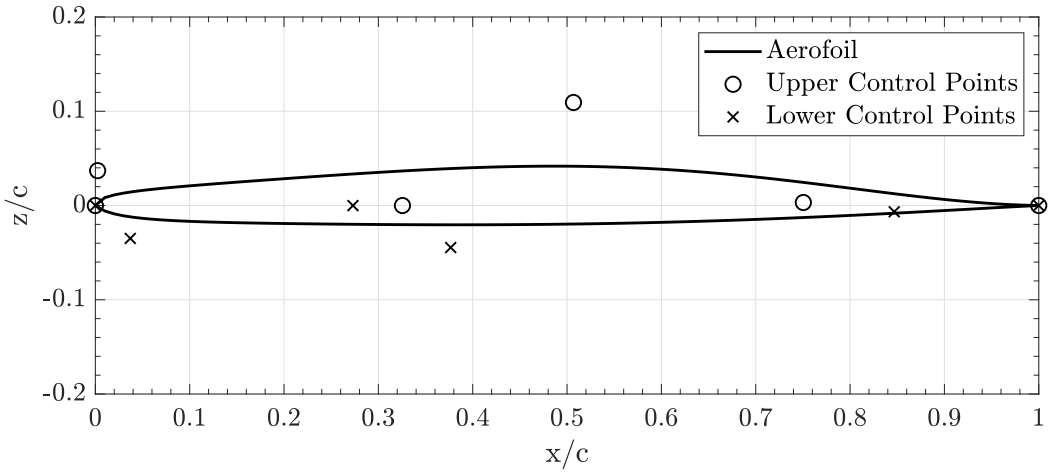


Figure 5.7: Multi-fidelity optimal design with upper and lower Bézier control points

The aerofoils found using the multi-fidelity adaptive cost technique in figure 5.6 are shown in terms of final cost values for each run in figure 5.8, alongside low-fidelity optimisations, baseline results, and multi-fidelity optimisations using exact model predictions. All of the force coefficient values here are derived from high-fidelity analysis, of which a number of clear observations can be made:

- Using only exact predictions from the surrogate model is a surprisingly poor approach
- Incorporating lower confidence boundaries produces a vast improvement
- Each adaptive cost run outperforms all other methods, resulting in a baseline improvement on every occasion

Taking each of these points in turn, failing to incorporate the lower confidence boundaries within the optimisation produces not only poor individual results, but also poor convergence characteristics across runs. This is somewhat the expected result, following the discussion in section 5.3.2. However, the poor performance in comparison to low-fidelity only analysis was not expected, confirming that this approach is not appropriate for use in global optimisation, unless an accurate prediction of the high-fidelity cost function is available from the outset.

With such poor results output from the predicted cost method, doubt was cast on the overall methodology itself, with little confidence that the adaptive cost approach would produce a significant improvement. Nonetheless, tending predictions toward the surrogate model's lower

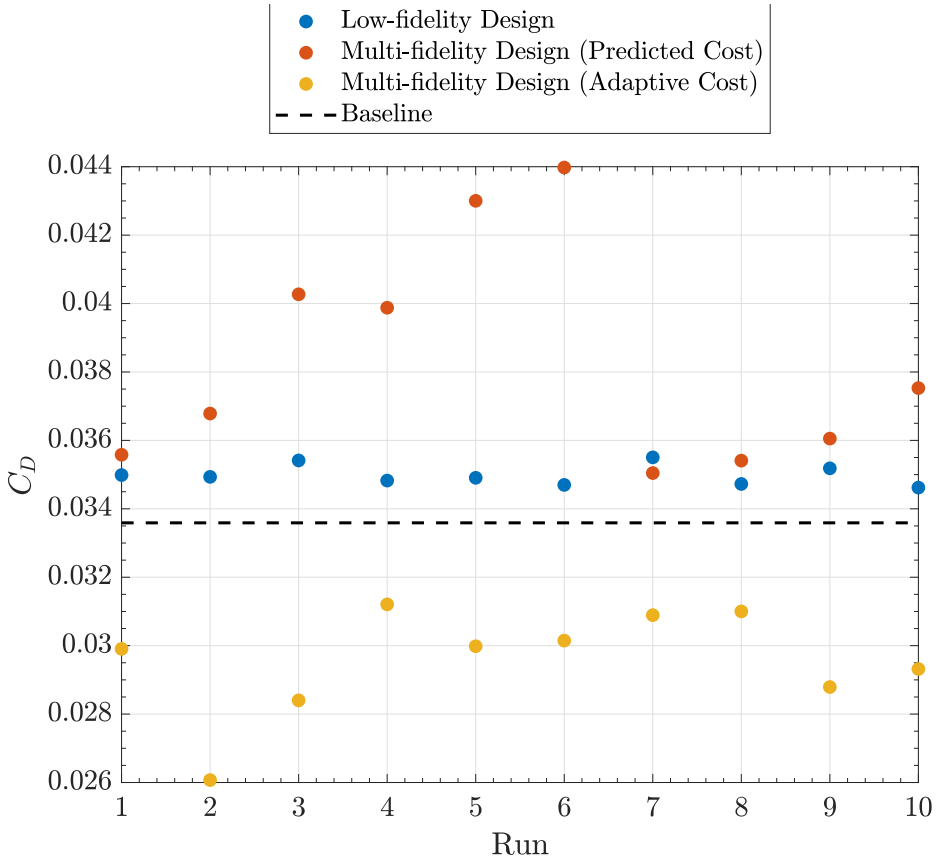


Figure 5.8: Multi-fidelity predicted cost and lower confidence boundary incorporation results compared with high-fidelity analysis of baseline optimisation in chapter 4

confidence boundary during periods of stall consistently produces the desired performance, finding designs that improve upon the baseline design and the previously discussed methods. This shows the necessity of exploring poorly represented areas within the design space, quantifying regions of uncertainty to provide a more global view of the design space.

The worst performing adaptive cost run is still a considerable improvement over the best design found by the other methods, as well as being an improvement over the baseline. It would be reasonable to expect that this significant improvement is reflected in number of high-fidelity cost function calls. However, the average number of calls during the adaptive cost method is 30, whereas the prediction only approach is 24, which is small increase in computation load compared to the vastly improved performance of the adaptive method. Furthermore, the adaptive method is more consistent in terms of function calls with a standard deviation of 5 across runs, while the prediction only method shows a deviation of 8.

One characteristic of the low-fidelity method that is not presented in the multi-fidelity adaptive cost approach is the convergence across runs, which can also be observed in the variation between aerofoils in figure 5.6. However, this is to be expected when working with surrogate

models, since the number of data points defining the high-fidelity cost function (≈ 110), is order of magnitudes lower than the number of function calls carried out by the low-fidelity only optimisation ($n_p \times n_{iter}$).

5.4.2 Global Results

The same methods were used to conduct multi-fidelity optimisation on the global design space, and provide a comparison to the single fidelity results gathered in section 4.5.5. Optimal design and variation throughout runs are displayed in figure 5.9, with method comparison shown in figure 5.10.

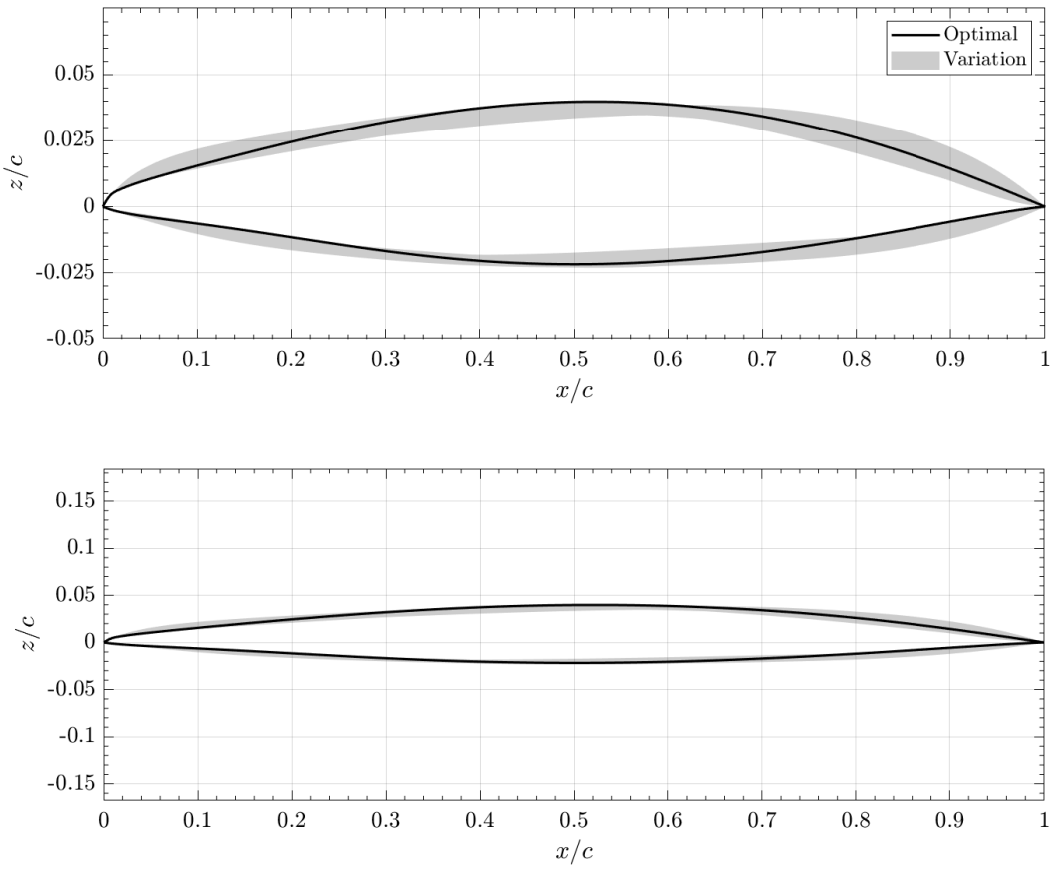


Figure 5.9: Multi-fidelity optimal design and variation across runs for global optimisation

Compared to the single fidelity optimal, the ideal multi-fidelity aerofoil is surprisingly symmetric about the mid-chord, with cross-sectional area and maximum thickness location shifted towards the centre. The variance shown in figure 5.6 covers a larger area than the low-fidelity results, however this is expected with limited high-fidelity cost function calls. Leading edge shapes remain consistent with low-fidelity results; minimising lower radius while maintaining a

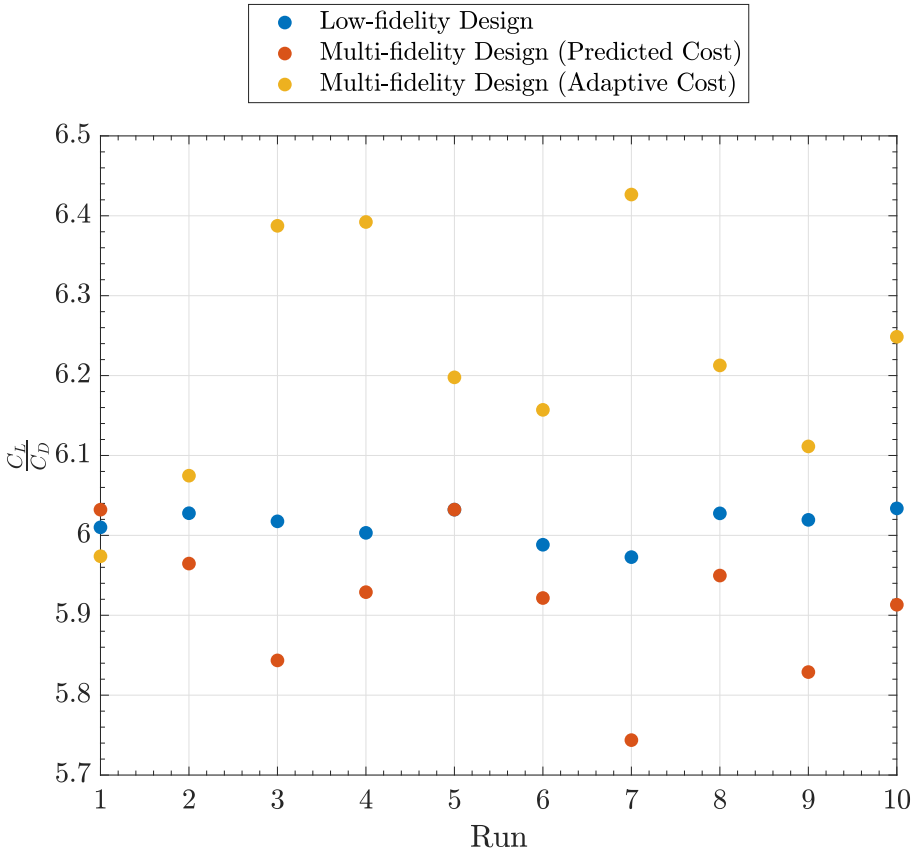


Figure 5.10: Multi-fidelity predicted cost and lower confidence boundary incorporation results compared with high-fidelity analysis of optimisation in chapter 4

larger upper surface radius, before transitioning to relatively wedge like frontal region.

Once again the multi-fidelity prediction only approach performs poorly, consistently producing shapes that perform worse than those found using low-fidelity analysis only. This affirms the need for lower confidence boundary integration within the cost function space; to avoid premature convergence, and encourage quantification of design space regions with high prediction uncertainty.

Consistent with baseline driven case, the best performing aerofoils found in the multi-fidelity adaptive cost method are considerable improvements upon single fidelity and multi-fidelity predicted cost methods. However, the magnitude of variance between runs is increased, with the adaptive cost method performing worst of the 3 approaches in the initial run. While with increased samples this may be reflected in the baseline case, it has already been discussed that the global optimisation problem presents a noisier cost function space. This decrease in likelihood of convergence is therefore magnified in the multi-fidelity methods, where only a small portion of the high-fidelity cost function space is sampled. Further work is therefore required to improve the robustness of multi-fidelity methods for a given run, but the performance benefits are clear

if the computational power is available to carry out multiple samples.

5.5 Conclusions

This work was a direct continuation from the preceding chapter, highlighting the inaccuracies in a single, low-fidelity optimisation that can produce misleading results. To overcome this, an efficient multi-fidelity adaptation of the SOPSO method has been produced, allowing promising candidates to be selected for high-fidelity analysis, and validating their performance against a prediction of the computationally demanding cost function.

To achieve this, a surrogate model was employed, allowing predictions of the high-fidelity cost function to made from a range of initial samples. Gaussian process regression was chosen for this task, due to it only requiring sub-spaces of the design space to be sampled, which is a requirement in a design space that contains infeasible designs. The surrogate model was built using design variables and force coefficient differences between low and high-fidelity analysis, allowing corrections to be applied on the low-fidelity results at each iteration, resulting in a prediction of high-fidelity performance. Interfaces were created to allow autonomous use of open-source programs capable of accurately meshing and analysing aerofoil shapes and their surrounding flow domain. This required a grid independence study to be carried out, in which conservative mesh parameters were eventually chosen, increasing the likelihood of accurate output results across a range of potential input aerofoil designs.

To incorporate such methods within a global optimisation context, a number of factors had to be considered. A two layered design of experiments was introduced, allowing samples to be taken from a larger set of designs that had been filtered in terms of geometric constraints. A crowding distance technique was employed to ensure these sampled feasible designs covered as much of the design space as possible. Allowing only designs that exhibited common aerofoil properties to be analysed by high-fidelity methods once again improved the likelihood that such analysis would be accurate, with CFD residuals implemented as further constraints, ensuring inaccurate data points are not added to the high-fidelity surrogate model.

An adaptive cost method was employed for high-fidelity prediction, which would tend towards the GPR's lower confidence boundaries during prolonged periods of stall within the optimisation. This approach was demonstrated to be a vast improvement over using exact prediction values alone, without incurring a significant increase in computationally expensive high-fidelity cost function calls.

The optimisation problems carried out in chapter 4 were repeated utilising the developed multi-

fidelity capabilities. Once again the optimisations were conducted a number of times, to quantify the convergence characteristics of the employed methods. On each occasion, the multi-fidelity SOPSO variant found an improved design with respect to the baseline case and single fidelity optimiser, demonstrating its ability to consistently find high performance areas of the high-fidelity cost function space, despite a limited number of such samples.

The global optimisation case magnified the issue of selective high-fidelity sampling, demonstrating poorer convergence characteristics due to larger feasible design space, and two variable definition of optimality. Nonetheless, the majority of multi-fidelity adaptive cost runs produced considerable aerodynamic efficiency improvements over other methods, once again demonstrating the usefulness of efficient high-fidelity cost function usage.

Chapter 6

Aero-Structural RLV Wing Optimisation

6.1 Introduction

The logical next step is to extend the optimisation work to three-dimensional models, by increasing the design space to multiple aerofoils, joined as part of a larger wing structure. This requires the addition of planform variables, with aerofoil sections defined at partition boundaries. A multi-partition wing definition is employed, allowing highly swept strakes to be combined with typical wing shapes; a common design characteristic of high-speed vehicles.

To provide a real-world application of high-speed wing design optimisation, a baseline configuration was selected, in the form of the X-34 reusable launch vehicle concept. This design was previously used for aerodynamic model validation, having been originally modelled as part of a joint UKSA/ESA funded project [8]. The simplified model was used to benchmark the various aerodynamic methods in this project, with experimental data introduced previously [44] used to benchmark the design concept; which was a carrier launched wing-body RLV inspired in part by the X-34 vehicle. The continued development of such concepts, along with availability of experimental data, makes the X-34 an appropriate baseline for this exploratory design optimisation work.

A further extension to the work carried out so far is the incorporation of structural dynamics within the optimisation process. This additional analysis discipline ensures that output designs are capable of withstanding the high loads that come with super-hypersonic flight. A parametrically defined wing structure is employed, allowing additional design variables to generate wing-box elements within the aerodynamic wing mesh. These elements are then reduced to a 1D beam model to determine physical displacements and rotations resulting from aerodynamic forces and moments. An interface is employed to converge the combined aero-structural system, iteratively passing information between aerodynamic and structural nodes. Failure criteria are tested after every cycle, adding to aerofoil feasibility constraints discussed in chapter 4, along

with geometric and performance constraints defined by general requirements and the baseline configuration, resulting in a highly constrained multi-disciplinary optimisation problem.

Incorporation of a baseline design and structural analysis also allows an estimation of weight to be output, thus enabling vehicle trim characteristics to be established. This further increases the quality of results through a number of means: by analysing the structural feasibility of design at a range of inclination angles; enforcing static stability constraints at the trim points; and enabling wing weight and trim angle to be included within the definition of optimality. Establishing multiple objective functions through the analysis disciplines modelled allows a diverse set of designs to be output on the resulting Pareto front, highlighting common characteristics among cost functions and trade-off designs. This is the primary goal of these exploratory optimisations, providing a number of high performing low-fidelity results that can be analysed further using higher fidelity analysis methods.

6.2 Vehicle Generation

Before configurations in a given design space can be analysed and optimised, a vehicle generator is required. Such a program must be able convert parametrically defined variables in full wing-body vehicles. Furthermore, it must not only have the capability of reliably producing such configurations given any set of design variables within their respective boundaries, but consistently discretise the various components to a high enough degree that multi-disciplinary analysis can be considered accurate. This process requires combination of the components into a single non-intersecting, air tight surface, in which any inner surfaces are discarded, and a consistent discretisation is enforced at the boundary of any merging parts.

On the other hand, it is important that the variables required for creation are chosen carefully based on the overall research scope, and therefore the desired optimisation criteria. Considering the exploratory nature of this work, the majority of cases allow a high level of flexibility in the choice of optimisation variables, and their respective boundaries. This is desirable in conceptual design, and realised by the utilisation of low-fidelity cost function analysis, which have a considerably small computation cost for a given design. Nevertheless, if optimisation variables do not have a clear and consistent definition, or introduce large non-linearities to the resulting objective functions, the algorithm may fail to consistently converge on globally optimal solutions.

6.2.1 Fuselage

While the fuselage itself is not included in the optimisation, it is important to introduce the variables that define the X-34 configuration. Furthermore, this design space could be included in

future optimisation work that focusses on wing-body vehicles. In this work, it acts as a stepping stone towards a more versatile design space introduced later.

Studying previous high speed wing-body vehicles, such as the space shuttle and X-34 baseline configuration, the main fuselage can usually be approximated with a constant cross-section shape. Considerably fewer design variables are required for such a definition in contrast to a varying shape in the chord-wise direction. Because of this, and since lifting surface optimisation variables will have a much larger effect on the configurations performance as a whole, a constant cross-section shape is deemed to be a worthy trade-off for a significantly less complex design space. Figure 6.1 shows the variables that make up the fuselage definition. Seven parameters are used in defining the main body cross-sectional shape, using a combination of straight edges and elliptical arcs. Five further variables are required, defining lengths of each sub-component and further nose parameters. Note the fore-body, which links the nose and constant cross-section aft-body components, is defined purely by length. In reality however, piecewise continuous polynomial interpolation is utilised to determine its shape, using both nose and aft-body shapes as control points.

The nose of the vehicle is created by a partial sphere, defined by its radius and non-dimensional length. This length is defined as the point at which the partial sphere ends and becomes part of the fore-body, with a value of 1 creating a full semi-sphere. A further variable that was deemed important to include as part of the nose definition is its offset in the z-direction from the aft-body centre. For a variety of reasons, many low and high-speed vehicles have their nose below the centre of the aft-body. In this application, the effect of this offset rotates the nose component around the y-axis, with respect to the distance between the nose and aft-body sections, as shown in figure 6.1.

6.2.2 Wing

An arbitrary number of separate wing partitions can be used to create a single wing, with the most inboard being defined from the vehicle centreline. Each partition is defined by five parameters: dihedral, taper ratio, trailing edge sweep, span, and two-dimensional aerofoil section. Of course an initial root chord is required, which can be an optimisation variable directly, or a ratio of fuselage length. The decision to use trailing edge sweep as a design parameter instead of defining it at the quarter chord or leading edge was to ensure that control surfaces were not made ineffective by high rearward sweep or large sweep discrepancies between partitions. Such a definition may result in undesirable sweep elsewhere on the wing chord, which can be easily calculated and constrained in the analysis phase. An example of the design variables used to define a two partition wing is shown in figure 6.2.

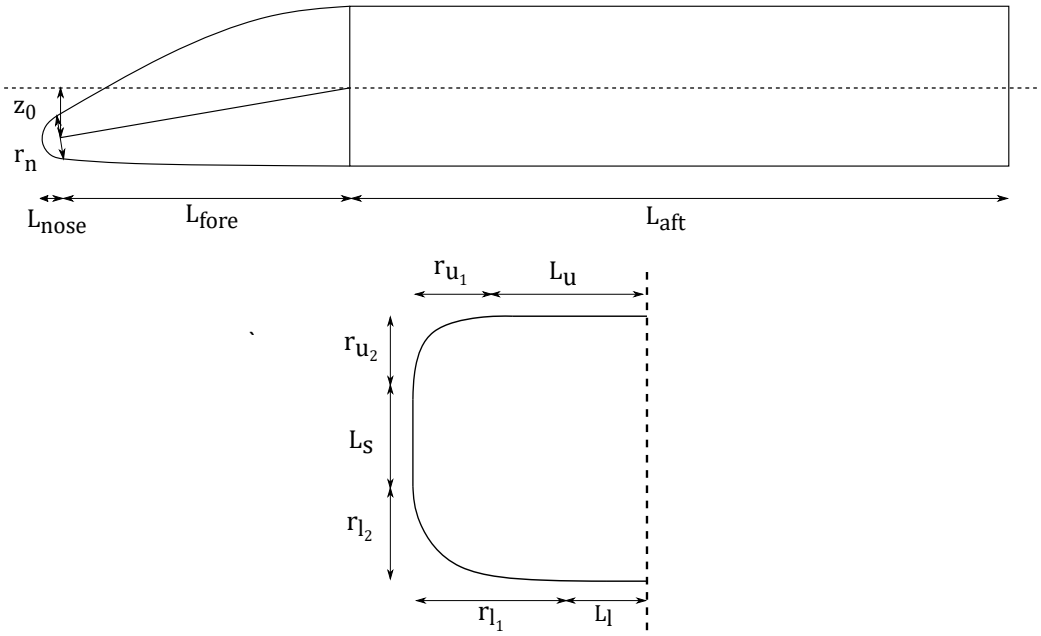


Figure 6.1: Fuselage design variables viewed in chord-wise and cross-sectional planes

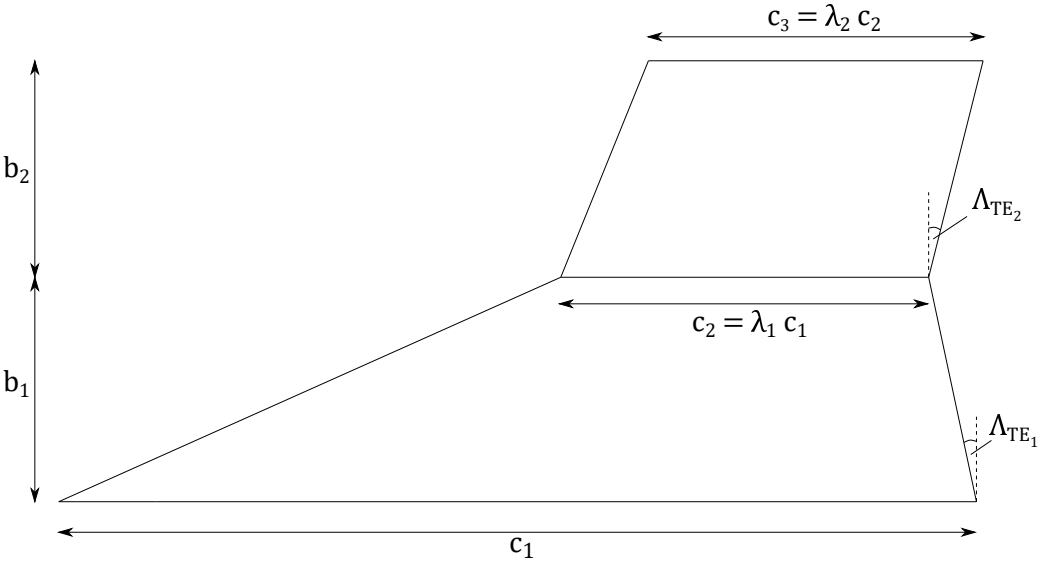


Figure 6.2: Planform design variables for two partition wing, defined from vehicle centreline

6.2.3 Aerofoil

For aerofoil generation, the PARSEC method was chosen due to its ability to directly constrain key characteristics, such as leading edge radius. This is particularly important in ensuring the output designs are capable of withstanding the thermally challenging environment experienced at high speeds. While this method was shown to be less flexible than the Bézier approaches in section 4.4, it should ensure more consistent production of expected aerofoil shapes. In turn, this constrains the production of wing-box elements to structures typically found in wing design. By doing so, the uncertainty produced by utilisation of low-fidelity structural methods is reduced, as is noise within the cost function, due to the more limited search space and consistent design definitions. As mentioned in aerofoil shape optimisation (section 4.4), tighter variable boundaries can be employed without a major loss in potential designs. Note that aerofoil sections are generated non-dimensionally, before being dimensionalised by the chord at their location within the wing during the assembly phase.

6.2.4 Vehicle Assembly

Components are generated independently before being combined into a single, non-intersecting surface. This combination is only required in the case of a wing-body vehicle, but is integral to the overall process in such cases. Figure 6.3 shows the various stages that the generator will go through in order to produce this final output. Here, the initial aft-body is not discretised in the x-direction, due to it having a constant shape with respect to its length, as previously described. The wing on the other hand, has not been discretised in the span-wise direction, due to the fact that it is defined from the centreline of the vehicle, and any part of the wing that is within the body will be discarded. Instead, the only span-wise discretisation that exists in the initial wing shape is that which defines a new partition. To merge these two parts, the radial panel of the fuselage that interacts with each of chord-wise panel edges of the wing must be found. A simple line-plane intersection is used for this, which finds the intersection point assuming an infinitely long line and an infinite plane. If the parts do intersect, a check must be performed to ensure that the points lies within both the fuselage panel, and finite wing panel edge. These intersection points define the new starting position of the wing, with respect to the desired surface that is to be output. Furthermore, they also define the x-direction discretisation of the fuselage, to ensure that an air tight surface with no gaps is created. Once this combination is complete, and the new definition of the wing output, it can now be discretised in the span-wise direction, and the final wing-body configuration is shown in figure 6.3.

Surfaces are discretised into triangular or quadrilateral elements, depending on user specification or choice of analysis methods. A number of techniques are available to create a model that accurately captures the features of a given configuration, thus ensuring a realistic surface is

output. The most consistent approach is to specify target panel dimensions. This means that regardless of two given structure sizes, they will be discretised into similar area panels. For the special case of lifting surfaces, where generally high curvature only occurs at the leading or trailing edges, accurate discretisation can be achieved by cosine or half-cosine methods.

6.2.5 Handling Infeasible or Undesirable Designs

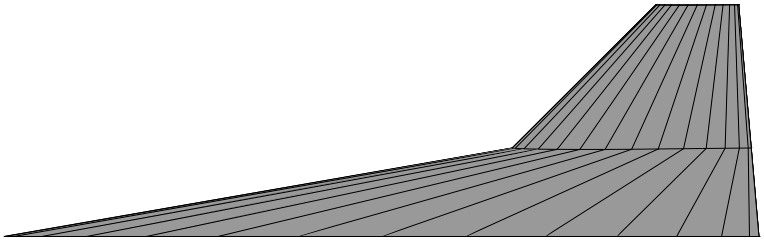
In the majority of global design spaces, there will be instances where the wing and body cannot be successfully merged by the method described above. This can be caused by a number of reasons, since the exact x, y, z joint locations between wing and body cannot be determined a priori. As previously mentioned, such issues could result in undesirable or infeasible configurations, which should ideally be dealt with at this phase, rather than being passed over to the performance analyser or cost function. It is more efficient in these cases to slightly alter the design variables, in order to produce a feasible configuration and resulting cost function, rather than designating the design as infeasible, inducing unnecessary noise. Examples of these errors will be discussed, and the remedies used to produce feasible and logical results will be justified.

Wing lies out-with fuselage

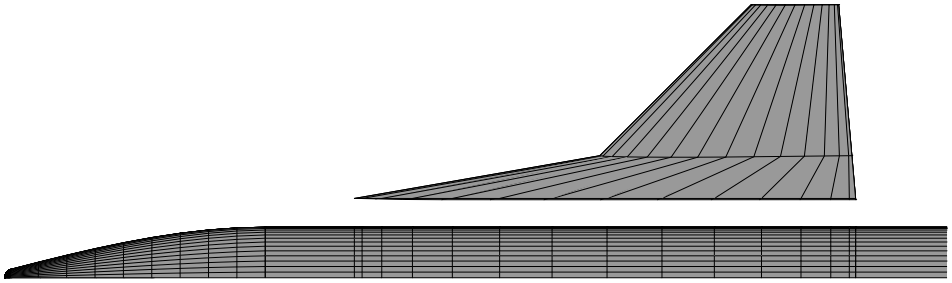
There are a number of reasons why a wing may partially lie outside of the fuselage. In such occurrences, the line-plane intersection will fail, and joining the two components will not be possible. Firstly, assuming both the wing and the fuselage are to be optimised, a large chord, thick root wing section could have points that lie above or below that of the fuselage at the centreline. Unless the dihedral is such that the wing partition falls back inside the fuselage, merging will fail. Moreover, for stability purposes, the leading edge offset of the wing with respect to the centreline of the fuselage can also be altered in the $x-z$ plane, which could also produce a wing that extends beyond that of the fuselage in the x-direction.

To remedy these issues, simple checks are put in place to find in which dimension the wing extends out-with the fuselage, defining whether the wing should be shifted in the x or z direction. This process is repeated until the parts are successfully merged. On some occasions however, the wing may perpetually bounce above and below the fuselage, for example, when using this method. Two fail-safe procedures have been put into place for this. The first reduces the dimensional factor which is attempting to bring the wing inside the fuselage, with respect to the number of attempts that have already been made. The second is a last resort approach, which after a specified number of failed attempts, begins to slowly reduce the overall wing chord. Note that the merging of these parts is based on a number of factors, such as wing dihedral and taper ratio, meaning that an initial root chord that lies outside the body may still result in a successful merge. Therefore, a solution to such problems is not as simple as ensuring the root

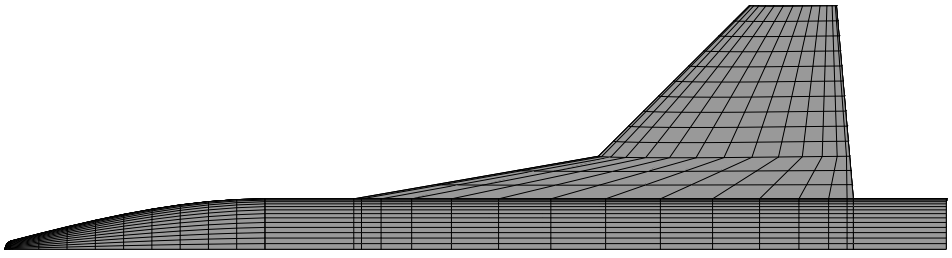
(a) Initial body pre-merge



(b) Initial wing pre-merge



(c) Exploded view of cut wing and resulting body x-discretisation



(d) Final merged wing-body with span-wise wing discretisation

Figure 6.3: Stages of wing-body merge procedure

chord lies fully within the body. A combination of these methods should ensure that a feasible configuration is output, regardless of the starting point.

Wing partition lies entirely within fuselage

It is assumed that at least some of the first partition of the wing lies out-with the fuselage. A major reason for this assumption is that if it were not the case, the first partition would have no effect on the overall performance of the configuration. As a result, the design variables creating this partition would also have little to no effect on performance. This is highly undesirable in the design optimisation process, as it clouds the sensitivity of these variables on the resulting cost functions. In a similar manner to the previous case, the semispan of the first partition is slowly increased until the entirety of its tip section is out-with the fuselage.

6.3 Structural Modelling

One of the major criteria to be determined in design a high-speed or re-entry vehicle is its ability to deal with the high structural loads expected throughout the flight profile. Particularly within the wings, where high bending moments can cause catastrophic failures, a robust structure must be employed. Of course, such a structure will be the major driver of weight within the wing, creating a highly sensitive trade-off between two of the most important factors in aircraft design. Structural feasibility should therefore be modelled as early as possible, providing an insight into the airworthiness of derived configurations. Furthermore, an initial estimation of wing weight allows further testing of a vehicles aerodynamic characteristics; ensuring enough lift is produced to maintain altitude at an efficient trim angle.

The employed structural model should reflect the conceptual nature of the design space, and moreover, compliment the level of fidelity used for the aerodynamic model. The addition of structural modelling results in an increasingly complex and computationally heavy cost function, due to the iterative method required to converge the combined aero-structural system at a given flight state. This requires the two analysis methods to be run sequentially, with the outputs of one being used to determine the inputs to the other. This tight coupling between the two disciplines means that an increased fidelity of one will not greatly increase overall accuracy, since its inputs are derived from a lower fidelity model. Therefore, in a similar manner to the aerodynamic module, an entry level structural model is defined, providing an initial estimate of wing elasticity, weight, and safety criteria.

A series of wing-box elements are modelled throughout the wing, and their properties reduced to 1D beam elements, connected end-to-end in the span-wise direction. This allows for a relatively simple coupling between structural and aerodynamic meshes, ultimately producing a deformed

wing based on the bending and twisting of the beam nodes. Each wing-box element is defined with forward and rear spars, along with upper and lower skins, both of which can have variable thickness. Stiffeners are also included in the model, both at the wing-box corners and within the skin panels. Modelling the wing-box structure in this way allow failure criteria to be determined in spar and skin segments independently, as will be detailed in the proceeding section.

6.3.1 Wing-box

As an initial estimation of static wing elasticity, a simple wing-box definition is reduced to an end-to-end span-wise beam model. In such structures, the upper and lower skins primarily carry the bending loads on the wing, with span-wise spars supporting shear loads. Defining the wing structure in this manner allows a straightforward box model to be created, in which cross sectional properties can be determined easily. Additional stiffness is provided through stringers, distributed evenly over both upper and lower skins. As in Bindolino [117], stringers at the front and rear spars are cross coupled in terms of area, while inner stringer areas are defined by a single variable. Wing-box segments are separated by beam nodes, as shown in figure 6.5. The average chord of each segment is thus used in the cross-section property computation, resulting in a beam model that is representative of a series of cuboid shaped wing-box segments. A portion of the structure is also assumed to lie inboard of the wing aerodynamic mesh, representing the fuselage joint. Therefore, the innermost node is assumed to be clamped, while the second is semi-flexible [118]. Since the physical attributes of the wing-box are computed prior to simplification, the volume of each component and thus overall weight of the wing-box can also be determined. From here, the total wing weight is estimated using an empirical method [119]. The loads produced by the wing structure are distributed uniformly along the elastic axis and applied to the element nodes by computing the work-equivalent nodal forces and moments [120].

Once the cross-section properties and weight estimations have been computed, the wing-box segments are reduced to a single span-wise beams, placed at computed wing-box shear centre. Each beam consists of two nodes, with translation and rotational degrees of freedom in x, y, and z directions, totalling 12 degrees of freedom per beam. Elemental stiffness matrices, $\mathbf{K}_e$, are therefore derived from equation (6.1).

$$\mathbf{K}_e = \begin{bmatrix}
\frac{EA}{L} & 0 & 0 & 0 & 0 & 0 & -\frac{EA}{L} & 0 & 0 & 0 & 0 & 0 \\
0 & 12\frac{EI_z}{L^3} & 0 & 0 & 0 & 6\frac{EI_z}{L^2} & 0 & -12\frac{EI_z}{L^3} & 0 & 0 & 0 & 6\frac{EI_z}{L^2} \\
0 & 0 & 12\frac{EI_y}{L^3} & 0 & -6\frac{EI_y}{L^2} & 0 & 0 & 0 & -12\frac{EI_y}{L^3} & 0 & -6\frac{EI_y}{L^2} & 0 \\
0 & 0 & 0 & \frac{GJ}{L} & 0 & 0 & 0 & 0 & 0 & -\frac{GJ}{L} & 0 & 0 \\
0 & 0 & -6\frac{EI_y}{L^2} & 0 & 4\frac{EI_y}{L} & 0 & 0 & 0 & 6\frac{EI_y}{L^2} & 0 & 2\frac{EI_y}{L} & 0 \\
0 & 6\frac{EI_z}{L^2} & 0 & 0 & 0 & 4\frac{EI_z}{L} & 0 & -6\frac{EI_z}{L^2} & 0 & 0 & 0 & 2\frac{EI_z}{L} \\
-\frac{EA}{L} & 0 & 0 & 0 & 0 & 0 & \frac{EA}{L} & 0 & 0 & 0 & 0 & 0 \\
0 & -12\frac{EI_z}{L^3} & 0 & 0 & 0 & -6\frac{EI_z}{L^2} & 0 & 12\frac{EI_z}{L^3} & 0 & 0 & 0 & -6\frac{EI_z}{L^2} \\
0 & 0 & -12\frac{EI_y}{L^3} & 0 & 6\frac{EI_y}{L^2} & 0 & 0 & 0 & 12\frac{EI_y}{L^3} & 0 & 6\frac{EI_y}{L^2} & 0 \\
0 & 0 & 0 & -\frac{GJ}{L} & 0 & 0 & 0 & 0 & 0 & \frac{GJ}{L} & 0 & 0 \\
0 & 0 & -6\frac{EI_y}{L^2} & 0 & 2\frac{EI_y}{L} & 0 & 0 & 0 & 6\frac{EI_y}{L^2} & 0 & 4\frac{EI_y}{L} & 0 \\
0 & 6\frac{EI_z}{L^2} & 0 & 0 & 0 & 2\frac{EI_z}{L} & 0 & -6\frac{EI_z}{L^2} & 0 & 0 & 0 & 4\frac{EI_z}{L}
\end{bmatrix} \tag{6.1}$$

Where E is the Young's modulus, A is the beam cross-sectional area, L is the beam length, G is the shear modulus, $I_{y,z}$ are the area moment of inertias, and J is the polar moment of inertia.

Equation (6.1) is in local coordinates, where the x-axis coincides with each beam's length. In order to solve the system, each elemental stiffness matrix must be transformed based on the angles between their local coordinate systems and the global one. These angles are used to form a transformation matrix, which then computes the global stiffness matrices as follows:

$$\mathbf{K}_e^g = \mathbf{T}^T \mathbf{K}_e^l \mathbf{T} \tag{6.2}$$

Where g and l denote global and local coordinate frames respectively.

With beams connected end-to-end, their global elemental stiffness matrices are collated on block diagonals, as shown in equation (6.3). The linear systems $\mathbf{Ku} = \mathbf{f}$ is then solved using Gaussian elimination, where $\mathbf{K}$ is the global stiffness matrix, $\mathbf{f}$ is the vector of forces and moments provided from the aerodynamic solver, along with weight estimations, and $\mathbf{u}$ is the resulting vector of structural displacements and rotations.

$$\mathbf{K} = \begin{bmatrix}
\mathbf{K}_{11} & \mathbf{K}_{12} & [0] & \dots & [0] \\
\mathbf{K}_{13} & \mathbf{K}_{14} + \mathbf{K}_{21} & \mathbf{K}_{22} & & \\
[0] & \mathbf{K}_{23} & \mathbf{K}_{24} + \mathbf{K}_{31} & & \\
\vdots & & & \ddots & \\
[0] & & & & \mathbf{K}_{n4}
\end{bmatrix} \tag{6.3}$$

Where the subscripts in $\mathbf{K}_{xy}$ define the beam number and $3x3$ sub-matrices of the global element stiffness matrices, respectively.

To ensure that the wing structure is capable of sustaining the high aerodynamic loads produced in

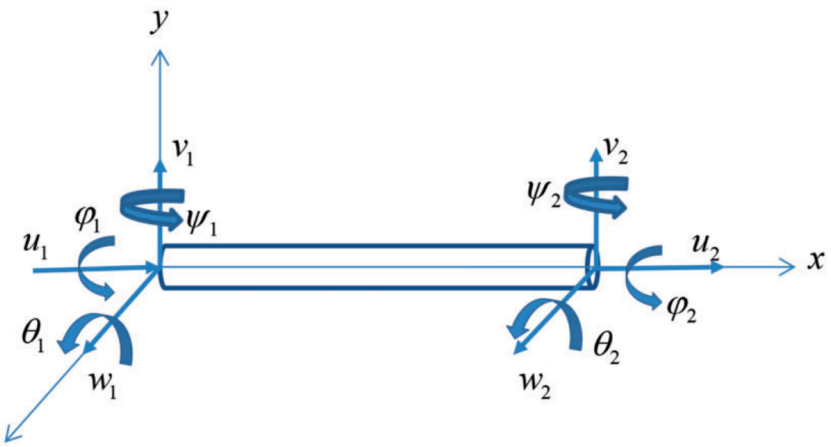


Figure 6.4: Displacements and rotations of two node beam element [119]

a given flight state, stress within the various components must be computed, and compared to the maximum allowable values for the structure materials. Stresses in each element are calculated using the equations provided by Elham and van Tooren [119]:

$$\sigma_{xx} = E\varepsilon_{xx} = E\left(\frac{du}{dx} + z_e\frac{d\psi}{dx} - y_e\frac{d\theta}{dx}\right) \tag{6.4}$$

$$\tau_{xy} = G\gamma_{xy} = G\left(\frac{dv}{dx} - \theta - z_e\frac{d\phi}{dx}\right) \tag{6.5}$$

$$\tau_{xz} = G\gamma_{xz} = G\left(\frac{dw}{dx} + \psi + y_e\frac{d\phi}{dx}\right) \tag{6.6}$$

Where y_e, z_e are the absolute distances of each wing-box element from the shear centre, with displacement and rotation directions defined in figure 6.4. Following this, failure criteria for each beam element is determined through:

$$F_y = \frac{\sigma \cdot SF}{\omega_y} + \left(\frac{\tau \cdot SF}{\tau_y}\right) - 1 \tag{6.7}$$

$$F_{b_e} = \frac{\sigma \cdot SF}{\sigma_b} - 1 \tag{6.8}$$

$$F_{b_s} = \frac{\tau \cdot SF}{\tau_b} - 1 \tag{6.9}$$

Here, failure due to material yield, F_y, Euler buckling, F_{b_e}, and shear buckling, F_{b_s}, are considered. The numerators in equations (6.7)-(6.9) denote the computed stress distributions in a given panel multiplied by a safety factor, SF, while denominators contain the maximum allowable material stresses.

To determine the maximum allowable stress that a given structure can undergo before it fails, the

equations from Elham et al. [121] are utilised. Assuming the entire structure is made of a single metal alloy, its mechanical properties can be used in determining these allowables [122]. Upper and lower panel buckling is calculated through the stiffened panel efficiency method, presented in Niu [123], in which an efficiency factor of 0.8 is adopted, assuming a realistic z-stringer joint is present in both panels. For skin panels under compression, the maximum buckling stress is computed using:

$$\sigma_{max} = \eta \sqrt{\frac{PE}{l_{rib}}} \tag{6.10}$$

$$P = \frac{\sigma A_u}{c} \tag{6.11}$$

Where η is the efficiency factor, P is the load intensity, and l_{rib} is the rib pitch. The upper skin panel limit is defined with the additional criteria of compressive yield with the minimum of the two values taken as the maximum stress allowable:

$$\sigma_{max_u} = min\left(F_{c_y}, \eta \sqrt{\frac{PE}{l_{rib}}}\right) \tag{6.12}$$

Finally, the maximum allowable spar panel shear stress is calculated using the equation provided in the US Military handbook [122]:

$$\tau_{max} = \frac{F_{t_y}(L) + F_{t_y}(LT) + F_{c_y}(L) + F_{c_y}(LT)}{4} \times \frac{2F_{s_u}}{F_{t_u}(L) + F_{t_u}(LT)} \tag{6.13}$$

Where t, c and s define tension, compression and shear; y and u denote yeild and ultimate stresses; while L and LT determine longitudinal and long transverse material directions. A constant safety factor of 1.5 is applied to the maximum stresses, and structural analysis terminated if any wing-box element is deemed to have failed.

Design Variables

Nine variables are required to create the rectangular wing-box structure as shown in figure 6.6. The position of the front and rear spars are necessary, in terms of non-dimensional chord, as well as constant values for spar and web thickness, $t_{s,w}$. The number of inner stringers has been included as a variable, N, to allow for a more versatile design space, which is the only discrete variable in the design space. As the stringers will be idealised as booms, three radii are required for the two sets of cross-coupled stringers at the spars, $A_{1,2}$, along with one variable that defines the radii of all those located inboard, A_3. Finally, the rib pitch is included as an optimisation variable, as this is a necessary parameter in the stiffened panel efficiency method [123]. From

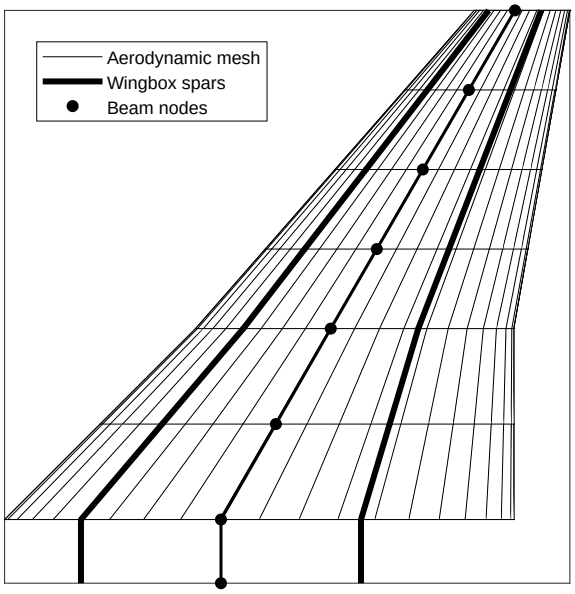


Figure 6.5: Aerodynamic and structural mesh visualisation

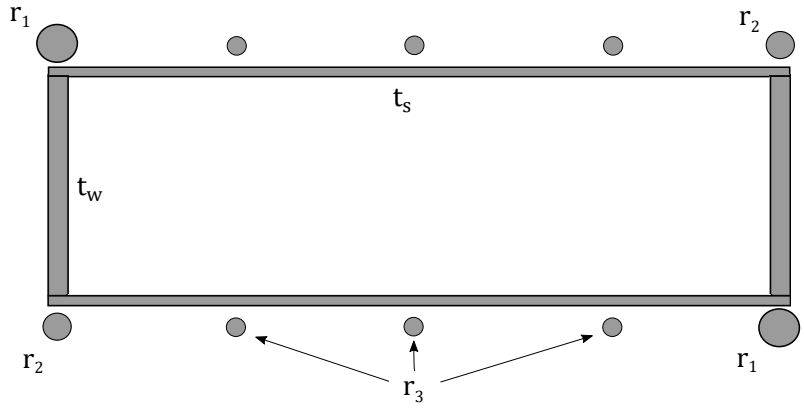


Figure 6.6: Wing-box skin, spar thickness and stringer area variables

these features, the wing-box cross-sectional properties can be calculated using the following equations [117]:

$$A = 2 \cdot (t_s \cdot c + t_w \cdot h + A_1 + A_2 + N \cdot A_3) \tag{6.14}$$

$$I_1 = 2 \cdot \left((t_w \cdot h + A_1 + A_2) \cdot \frac{c^2}{4} + \frac{c^3 \cdot t_s}{12} + A_3 \cdot c_3 \right) \tag{6.15}$$

$$I_2 = 2 \cdot \left(\frac{t_w \cdot h^3}{12} + (t_s \cdot c + A_1 + A_2 + N \cdot A_3) \cdot \frac{h^2}{4} \right) \tag{6.16}$$

$$J = 2 \cdot \left(\frac{c^2 \cdot h^2 \cdot t_w \cdot t_s}{c \cdot t_w + h \cdot t_s} \right) \tag{6.17}$$

Where c and h are the chord and height of the wing-box, and c_3 is the summed squared chord-

wise distance of every inner stringer from the neutral axis, and A_i is the resulting area from a given stringer radii. Note that since the box is rectangular, all stringers will be equidistant from the neutral axis in terms of height, hence why they have been grouped together in equation (6.16). Furthermore, note that ribs are not included in the wing-box properties calculation, however their weight contribution is estimated empirically [121].

6.3.2 Interface

Coupling the aerodynamic and structural solvers is achieved through the use of Gauss-Seidel fixed-point iterations. This method allows each solver to run in sequence, using the data gathered from the previous output of the other, until there are no further changes in their respective outputs. This process is completed as follows: the aerodynamic module is run initially, and once forces on individual panels of the undeformed mesh have been computed, they will be transferred to the structural solver, where equivalent loads on the beam nodes are calculated. Combining these forces and moments with those derived from the estimated wing weight then allows the resulting beam displacements and rotations to be found. The structural mesh motion is translated to the aerodynamic mesh, where the process repeats until convergence. With low-fidelity aerodynamic methods being employed, the aero-structural solution can be considered to have converged once the aerodynamic mesh motion $\Delta\mathbf{w}$, that is the change in mesh from the previous iteration, becomes negligible. In this work, we assume convergence when $||\Delta\mathbf{w}|| < 10^{-6}$.

As arbitrary wing shapes and structures will be created and analysed in a large design space optimisation, with many infeasible designs expected in the early stages, an efficient interface is required between the two disciplines. Therefore, the process is accelerated with a version of Aitken's [124] relaxation parameter, ω, which is applied to the structural displacements and rotations, shown below:

$$\omega_{i+1} = -\omega_i \frac{(\mathbf{u}_{i-1})^T(\mathbf{u}_i - \mathbf{u}_{i-1})}{|\mathbf{u}_i - \mathbf{u}_{i-1}|^2} \tag{6.18}$$

$$\omega_0 = 0.5 \tag{6.19}$$

This method has been used extensively in fluid-structure interaction applications [125–127]. Calculation of the parameter is carried out through the use of structural displacements from previous and current iterations to accelerate the systems convergence. Once found, it relaxes the structural displacements of the current iteration, and thus the motion of the aerodynamic mesh. Since a previous cycle is required in order to calculated the parameter, a value of $\omega = 0.5$ is used for the initial cycle.

6.3.3 Interpolation Methods

By their nature, aerodynamic and structural definitions for a given wing configuration often result in meshes with no overlapping nodes, along with node patches that may lie entirely out with that of the partner mesh. Of course, the aim of an aerodynamic mesh is to capture the shape of a given body as precisely as possible, allowing the calculation of accurate surface properties. This results in a highly sensitive mesh with rigid requirements, which in terms of accuracy, is limited only by the choice of fluid solver. On the other hand, structural definitions for aerospace applications vary in literature from full finite element representations of typical skin, spar, rib configurations, to idealised beam or stick models. In either instance, a method is required to transfer information between meshes. As the location of aerodynamic nodes, where external forces are computed, differ from that of structural nodes, where displacements and rotations are calculated, loads and mesh motion from the respective mesh must be translated to the other.

Various methods have been utilised to achieve this. The importance however lies in a given schemes ability to transfer this information in a consistent and conservative manner, as first laid out in Brown [128]. Consistency requires that the sum of forces acting on the fluid model equals that of the nodal forces applied to the structure through the principle of virtual work. In a similar manner, conservative demands that the work done on the aerodynamic mesh from displacements computed on the structural model, be equal to the work performed on the structure.

For cases where structural and fluid meshes share the same or similar boundaries, but inner nodal locations and/or mesh densities vary, bi-linear interpolation can provide a simple yet accurate transfer of information. In situations where the finite element grid is rather irregular, an area coordinate approach can be adopted. Here the FEM mesh is mapped to an unstructured triangular grid. CFD points encapsulated within the sub-triangles made from an element of the unstructured mesh are then identified, with the force being calculated from the area of these sub-triangles. Full details of this method can be found in Guruswamy [129]. It should be noted that in both instances, conservation is not guaranteed, and fine grids for both disciplines are required to generate accurate results.

Linear and higher-order isoparametric finite elements such as plates or shells are often used in wing modelling, employing shape functions to transfer the local or computational space displacements to the physical domain, where they can be interpolated to the fluid grid. However, finding the location of the fluid grid points that lie within a given element in local coordinates is a more difficult challenge that cannot be performed analytically, and requires the numerical approach of inverse isoparametric mapping (IIM) [130]. Upon completion, the forces acting on the fluid nodes can be proportionally distributed to the structural element nodes. In the case of fluid nodes being out with the structural boundary, linear extrapolation must be performed. Ex-

amples of successful IIM implementation in fluid-structure interaction problems can be found in the works of Byun and Samareh [131, 132].

When fluid and structural meshes have considerably different boundaries, or perhaps the structural model lies completely within the fluid mesh, such as beam or wing-box models, the rigid link method introduced by Brown [128] has proved popular. Initially introduced for CFD modelling, various high-fidelity aero-structural optimisation have utilised this simple method [118, 133]. Coupling the two domains is achieved by finding the nearest structural node to a given node on the fluid surface, and using the vector connecting these two points to transfer information. In terms of structural displacements and rotations, this is done as follows:

$$\mathbf{w}_A = \mathbf{w}_S + \theta_S \times \mathbf{r} \tag{6.20}$$

Where $\mathbf{w}_S$ and θ_S are the displacements and rotations of the donating structural node, $\mathbf{r}$ is the vector defined from the structural node to the fluid node, and $\mathbf{w}_A$ is the resulting displacement of the aerodynamic node. Similarly, the rigid link vector is used in the translation of forces and moments from the fluid grid to the structural load vector through virtual work. It is important that the rigidity of this vector is emphasized, meaning that when an iterative scheme is applied, $\mathbf{r}$ should only be calculated in the pre-processing phase with the undeformed meshes. This method is commonly used in high-fidelity analyses [118, 133]. However recently, it has also been used for low-fidelity coupling [119, 127]. Both of the referenced works use a simple idealised beam model for the wing structure, and exploit the rigid link method by aligning the beam nodes with the fluid mesh in the span-wise direction. This means that finding the closest structural node to a given fluid point can be bypassed, and allows node specific expressions to be derived for loading and mesh motion transfer [127].

In terms of mesh motion, radial basis function (RBF) interpolation [134] has grown popular in recent years, due to its versatility and ease of implementation. In this method, two arbitrary point cloud meshes, structured or unstructured, can be coupled without any code alterations required. Essentially, RBF interpolation calculates how a given node or surface point will be influenced by that of every node within a structural or aerodynamic mesh. This means that mesh motion can be easily computed from one mesh to another by simple matrix manipulation. A brief overview is provided here, however in depth explanations can be found in references [135, 136]. The problem is formulated as follows:

$$s(\mathbf{x}) = \sum_{j=1}^{N} a_j \phi(||\mathbf{x} - x_j||) + p(\mathbf{x}) \tag{6.21}$$

In this instance, $s(\mathbf{x})$ is the function evaluated at $\mathbf{x}$, j depicts the centre or node within the mesh, with N being the total number of nodes. The function ϕ represents a radial function

with respect to Euclidean distance. Many options are available for this, which can be found in Rendall [136]. By setting $p(\mathbf{x})$ as a linear polynomial, the interpolation does not change, regardless of translation or rotation of the mesh, meaning coupling matrices need only be defined once in the pre-processing phase. Coefficients a_j are defined by the interpolation conditions:

$$s(x_j) = y_j,\ 1 \leq j \leq N \tag{6.22}$$

Where y is the known positions of the centres or nodes. With the inclusion of the polynomial in equation (6.21), a second requirement states that:

$$\sum_{j=1}^{N} a_j q(x_j) = 0 \tag{6.23}$$

For all polynomials q equal to or lesser degree than that of p. Importantly, this ensures that the total information transfer is equal within or between meshes, making it a conservative approach.

In the present work, the rigid link method has been employed, to be used in both virtual work transfer of forces, and mesh motion. The overriding reason for this is the choice of structural model, in this case a single discretised beam. As mentioned previously, beam nodes are aligned with the aerodynamic mesh in a span-wise direction, allowing simple coupling of the two disciplines. RBF interpolation has also been implemented, for use in future work involving physical wing-box structural definitions.

6.4 Implementation to Optimiser

6.4.1 Design Loop

The overall architecture of the described framework is shown in figure 6.7. Functions in grey refer to material covered in chapter 3, with the routines that make up the cost function being the main focus here.

Two loops are performed within the cost function; the outer attempting to find the trim angle of attack, and the inner solving the aero-structural system for the current angle. In terms of trim analysis, the lift force for a given flight state is compared to the total configuration weight, and the angle of attack updated accordingly. At a given inclination angle, aerodynamic forces acting on the undeformed mesh are computed, which are transferred to the structural solver to calculate displacements and rotations. These deformations are used to displace the aerodynamic mesh, and this process is repeated until either convergence or failure criteria is met. If converged, overall aerodynamic and structural properties are gathered, and a check is performed to determine if the aircraft is at trim conditions. Until this stability criteria has been met, the angle of attack

is iterated, the aerodynamic and structural meshes are initialised to their undeformed state, and the aero-structural loop is repeated. Upon convergence of the entire aero-structural trim system, violation characteristics are computed, and any penalties added to the final cost function values.

To reduce the number of outer loop iterations, boundaries are set on the allowable trim angle of attack range, and the final α_{trim} value is only computed to three decimal places. Furthermore, if the wing structure is deemed to have failed at any point within this process, the cost function is continued with a rigid airframe. Previously, the cost function would have been terminated entirely, and the configuration marked as infeasible. However, this "death penalty" approach caused a large amount of noise in the cost function, with relatively similar designs having potentially optimal and infeasible designations. By allowing analysis to continue with a structurally incapable design, comparisons can be drawn amongst the remaining performance metrics, and penalties applied purely based on the structural infeasibility.

After all particle cost function values have been calculated in a given iteration, they are compared to previous particle bests and Pareto front designs, updating both populations as required. As discussed previously, there are no widely accepted methods that define convergence of a multi-objective population based optimiser. Therefore, the PSO algorithm is run for a set number of iterations, defined by the user.

6.4.2 Constraint Management

While the design space in this work is mostly continuous, the output cost functions for a given set of designs are not. This is due to the fact that analysed configurations may have attributes that are physically undesirable, but that may be seen as desirable to the optimiser, and cannot be constrained or calculated prior to analysis. In such cases, a quadratic loss violation function is used, which compares vehicle characteristics to predefined minimum and/or maximum values, and applies a penalty to the cost function values depending on how far out with the boundaries a violating attribute lies. Adding to the objective function by a constraint derived value is more commonly known as a penalty method, discussed in chapter 3. This transforms a constrained optimisation into an unconstrained one, by combining cost and constraint into a single function:

$$f'(\mathbf{x},R) = f(\mathbf{x}) + \phi\left(g(\mathbf{x}),h(\mathbf{x}),R\right) \tag{6.24}$$

Where f' is the new function to be minimised, and ϕ is the violation function calculated by:

$$\phi\left(g(\mathbf{x}),h(\mathbf{x}),R\right) = R\left\{\sum_{k=1}^{p}\left[max(0,g_k(\mathbf{x}))\right]^2 + \sum_{l=1}^{q}\left[h_l(\mathbf{x})\right]^2\right\} \tag{6.25}$$

With terms consistent with that of equation (3.10) [48]. The value of the penalty parameter is

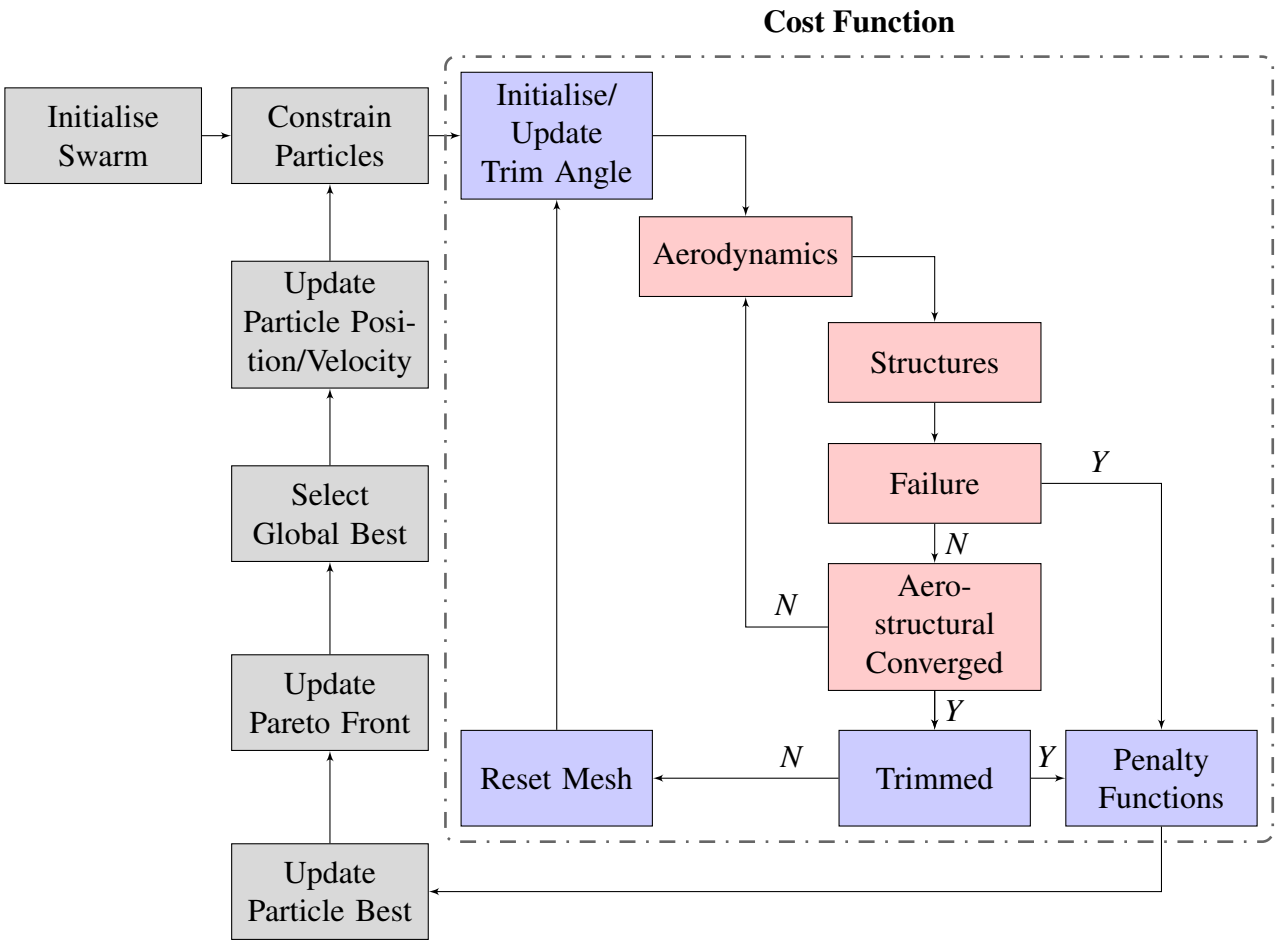


Figure 6.7: Multi-objective aero-structural trim PSO iteration flow chart

chosen so that it is considerably larger than values arising from the cost functions, ensuring that non-violating particles will always have a lower f' value than violating ones.

This method allows the optimiser to analyse infringing configurations without seeing them as optimal, while also not disregarding them entirely. Furthermore, smoothing the cost functions in this manner allows the optimiser to tend towards non-violating values, improving overall convergence.

6.5 Results

To demonstrate the capability of the overall framework, a multi-objective, multi-flight state optimisation was conducted, and compared to that of the X-34 configuration. Three objectives were sought to be minimised across Mach numbers of 3 and 6, with the remaining flow properties defined by that of experiments carried out in Brauckmann [44]. The overall optimisation problem is defined in table 6.1, with design variable boundaries in table 6.2, and results shown in the form of a Pareto front in figure 6.8. Particles are coloured in order to demonstrate their relative

objective function values versus the rest of the front. Here, red indicates a low drag coefficient, green tends towards minimal trim angle, and blue shows reduced wing mass. The optimisation was run for 600 iterations with 1200 particles across 8 cores, and completed in approximately 72 hours.

As mentioned previously, the X-34 configuration has been used as a baseline to provide constraints on designs, and present a comparison for optimal results. In particular, since the X-34 was designed to be launched from a carrier aircraft, strict geometric requirements have been placed on candidate designs, along with those demanding a maintenance or improvement in terms of performance. Note that no attempt has been made to model the X-34 structure, and thus its performance is based on a rigid airframe. Furthermore, only the wing and body sections have been modelled in this instance. The original wing mass is an estimation based on the overall structural weight of the vehicle, and the empty weight has been utilised in the trim computation, implying an unpowered descent phase.

Three optimal configurations have been extracted from the 58 found on the Pareto front, and are shown with pressure coefficient distributions at Mach 3 in figure 6.9, along with the baseline shape. Furthermore, performance characteristics are compared in table 6.3, again shown in terms of maximum values across the two flight states. Note that this generally corresponds to Mach 3 in terms of force and moment coefficients, and Mach 6 in terms of trim angle. Similarities can be drawn from the aerofoil shapes presented in all cases, particularly in configurations I and II, along with similar planform shapes in I and III. However, stark contrasts are observed from their objective functions. Unsurprisingly, configuration II provides a minimal mass option, due to its reduced wing area. Furthermore, despite their planform similarities, configuration I demonstrates a low drag coefficient design, with configuration III performing best in terms of trim angle, showing the highly sensitive nature of high-speed aerofoil design.

Much like the baseline shape, thinner aerofoils are generally present at the root of designs, due to their large inboard chord. A drastic increase in lift coefficient compared to the baseline can also be observed from the three designs, despite a reduction in trim angle. However, the large sweep of the X-34 strake, and thus reduction in drag that it provides, is not seen across the resulting configurations. Furthermore, in an attempt to maintain structural integrity at such high speeds, increased mass values are present across the Pareto front versus that of the X-34 estimation. A combination of these properties point to potential inaccuracies in weight prediction, since the X-34 has a considerably larger wing area than almost all of the resulting designs.

Various nuances are present in the remaining Pareto front designs, particularly in terms of planform and aerofoil shape. However, many of the geometric characteristics discussed can also be

Table 6.1: Aero-structural trim optimisation problem definition

		Comments
Minimise	$C_{D_{trim}}$	Optimise trim angle and drag coefficient at trim based on maximum respective values from both flight states, and overall wing mass
	α_{trim}	
	m_{wing}	
Subject to		
Variable boundaries	$\mathbf{x}_{lb} \leq \mathbf{x} \leq \mathbf{x}_{ub}$	Defined in table 6.2
Performance constraints	$L \approx W$	
	$C_m \approx 0$	Trim & stability conditions
	$C_{m_\alpha} < 0$	
	$F_y \leq 0$	Yield and buckling failure criteria for all skin panels spar webs
	$F_{b_{e,s}} \leq 0$	
Baseline constraints	$C_{L_{trim}} \geq C_{L_{base}}$	Improve or maintain baseline trim lift coefficient
	$0.5 \leq A_{wing}/A_{base} \leq 1.5$	Maintain similar wing planform area and wing root, tip, and trailing edge thickness
	$0.5 \leq t_{wing}/t_{base} \leq 1.5$	
	$b = b_{base}$	Enforced by design variable definition
	$t_{le} \geq t_{base}$	Ensure minimum leading edge for heating requirements
General constraints	$\frac{dt}{db} \leq 0$	Wing thickness cannot increase outboard
	$30^\circ \leq \Lambda_{le} \leq 80^\circ$	
	$0.1 \leq c_{t_{max}} \leq 0.7$	Non-dimensional chord location of maximum aerofoil thickness
	$\left(\frac{dt}{dc}\right)_{t < t_{max}} \geq 0$	Aerofoil thickness cannot decrease up to maximum thickness
	$\left(\frac{dt}{dc}\right)_{t > t_{max}} \leq 0$	Aerofoil thickness cannot increase after maximum thickness
	$\left(\frac{d^2u}{dx^2}\right)_{u < u_{max}} \leq 0$	Upper surface curvature cannot increase up to apex

Table 6.2: Aero-structural trim optimisation design variable boundaries

Component	Variable	Minimum	Maximum	Comments
Wing Planform	c_1, m	6	12	Root chord
	$\lambda_{1,2}$	0.4	1	
	Λ_{te}, deg.	-20	20	Variables from figure 6.2
	b_1, m	0.1	0.9	Dimensionalised by baseline span ($b_2 = 1 - b_1$)
	ϕ, deg.	0	20	Dihedral
	x_0	0	0.6	Non-dimensional wing offset
Aerofoil	$r_{le_{u,l}}$, m	0.001	0.1	Variables from figure 4.2, one set for each root, joint, and tip aerofoils
	x_u	0.2	0.7	
	z_u	0.02	0.17	
	z_{xx_u}	-1.2	0	
	x_l	0.2	0.7	
	z_l	-0.08	-0.02	
	z_{xx_l}	0	1.2	
	z_{te}	-0.02	0.02	
	Δz_{te}	0	0.02	
	α_{te}, deg.	-25	2	
	β_{te}, deg.	3	40	
Wing-box	$c_{1,2}$	[0.15, 0.65]	[0.25, 0.85]	Non-dimensional chord location of front and rear spars
	$t_{s,w}$, m	0.001	0.05	Variables from figure 6.6
	r_{1-3}, m	0.005	0.05	
	N	0	5	Number of inboard stringers
	l_{rib}, m	0.5	1.5	Span-wise distance between ribs

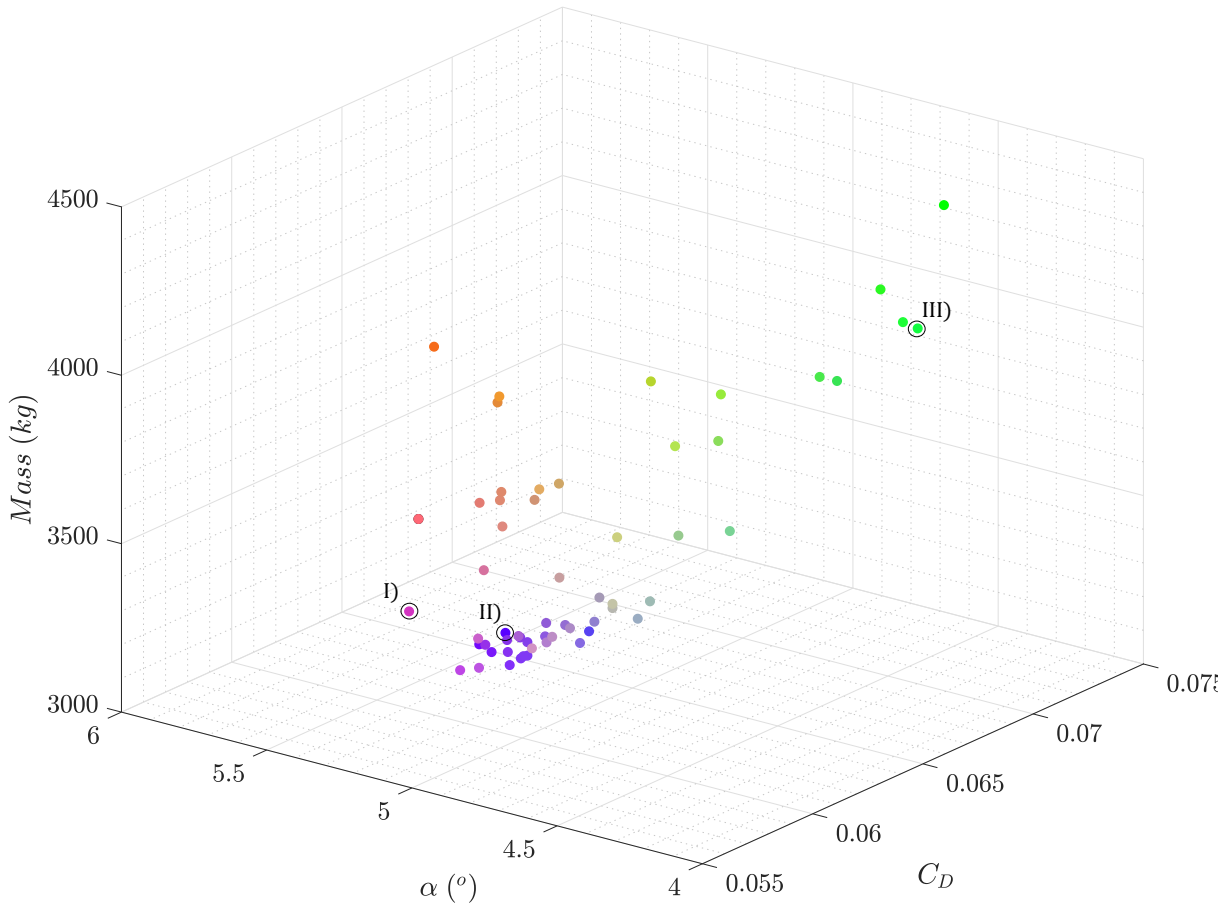
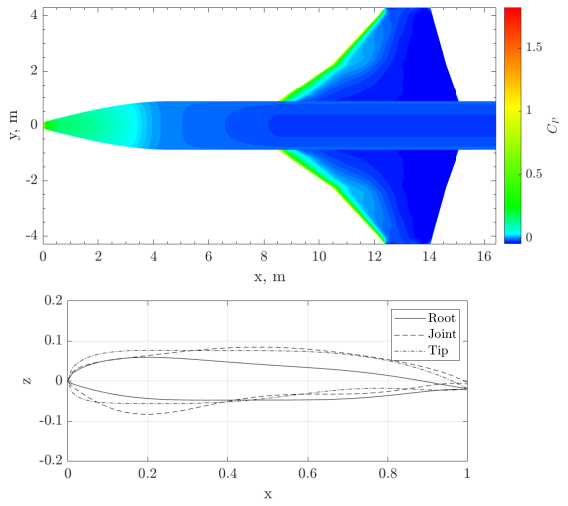
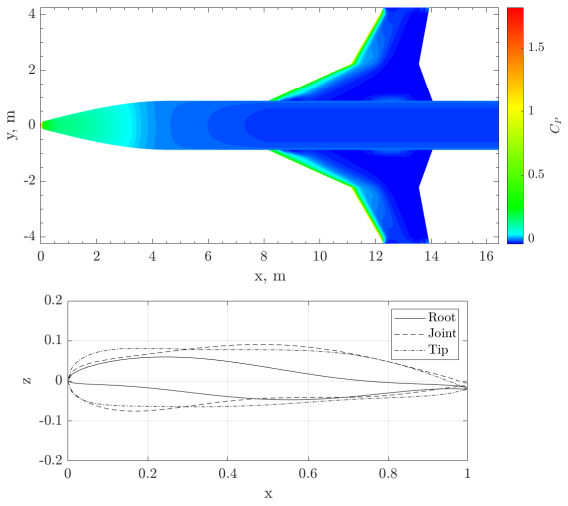


Figure 6.8: Aero-structural trim optimisation Pareto front with selected configurations highlighted

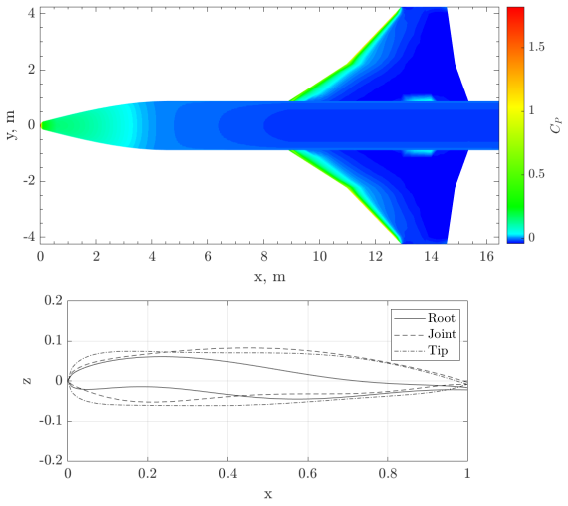
seen in multiple configurations, showing convergence tendencies for the given problem. Structurally, only two configurations contain any inner stringers, demonstrating a preference for outboard stiffening. The remaining structural variables, normalised in figure 6.10 with respect to their minimum and maximum boundaries from table 6.2, are similar across all Pareto front designs. As these variables only have an effect on one cost function, wing mass, this convergence should not be surprising. In an attempt to minimise mass, the structural variables tend towards the boundaries that create such a wing-box, while ensuring components are not subject to failure under the intense aerodynamic loads. In particular, the forward spar location and skin thickness remain almost constant throughout, contributing minimal mass to the output designs. On the other hand, upper front and lower rear coupled stringers, along with the location of the rear spar, demonstrate the highest variation within the structures. The scope of these variables therefore have a large influence on failure characteristics of the wing-box, as they do not consistently result in a minimal mass component. Of course, the low-fidelity structural model may be a factor in producing these results, with component failure playing a large role in the optimisation process. Further flexibility is therefore desired in the wing-box design, to ensure fewer failing



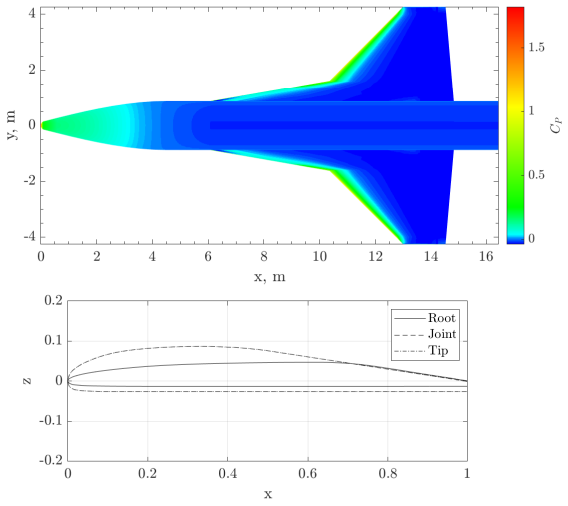
(a) Configuration I



(b) Configuration II



(c) Configuration III



(d) Baseline

Figure 6.9: Selected Pareto front configurations (a - c), and baseline (d), showing pressure coefficient distribution at Mach 3 trim conditions, and non-dimensional aerofoil sections at root, partition joint, and tip

Table 6.3: Optimal design performance characteristics

	Baseline	Configuration I	Configuration II	Configuration III
$C_{D_{trim}}$	0.0402	0.0551	0.0625	0.0702
$\alpha_{trim}(^\circ)$	5.244	4.983	5.231	4.326
$m_{wing}(kg)$	3000	3800	3091	4430
$C_{L_{trim}}$	0.0565	0.069	0.0779	0.0724
$C_{m_{trim}}$	-0.0173	-0.0114	-0.0164	-0.0007

configurations, and a broader search space of feasible structures.

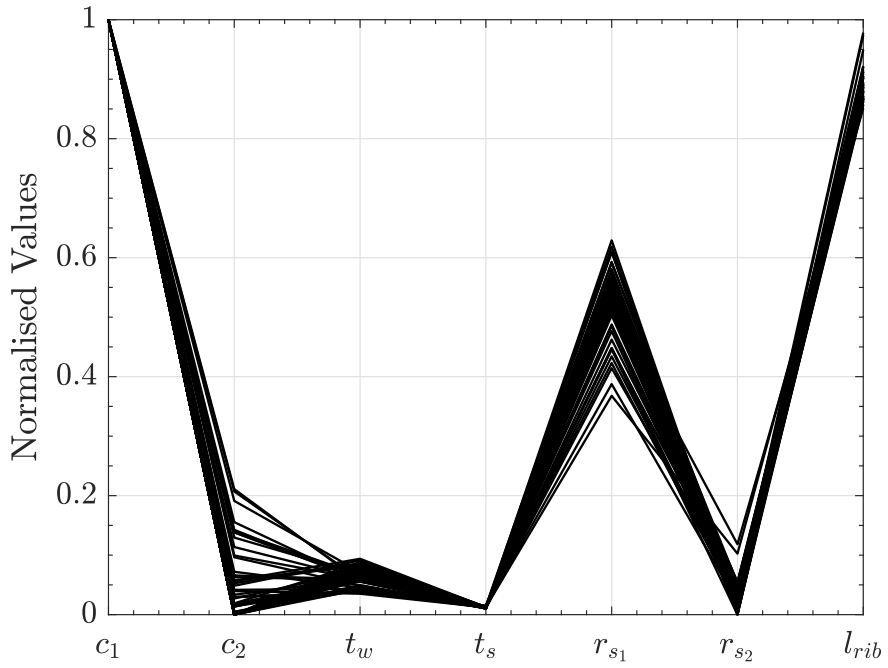


Figure 6.10: Normalised structural design variables

6.6 Conclusions

Presented is a large search space global design optimisation framework for super-hypersonic vehicles. Low-fidelity surface inclination methods have been employed for aerodynamic analysis, with a simple wing-box model being reduced to a single discretised beam to approximate structural elasticity. Gauss-Seidel fixed-point iterations have been utilised to couple the aero-structural system, with Aitken relaxation used to accelerate convergence. Particle swarm has been chosen as the optimisation algorithm, due to its superior handling of continuous, large design space problems over gradient-based or other gradient-free methods. The combination of these tools has produced a versatile design optimisation system capable of handling a broad

design space, and large number of function evaluations, without significant computational cost.

A detailed description of the design variables has been provided, with explanation of a penalty approach for constraint management and undesirable characteristics or infeasible designs. A multi-objective, multi-flight state aero-structural trim optimisation has been performed, with comparison to the wing-body X-34 RLV configuration, and select Pareto front designs shown. A diverse set of configurations have been produced, proving the optimisers capability in handling large design spaces and highly non-linear objective functions. This is due to its flexibility in allowing undesirable or infeasible characteristics to be analysed and appropriately quantified. Baseline improvements can generally be found in terms of trim angle and lift coefficient; however mass estimation has been highlighted as a crucial limiting factor in producing lower drag and overall wing weight. Further research requires deeper flexibility within the structural model, along with improved mass estimation from a given wing-box structure, in order to produce a wider array of feasible designs.

With continued research and development of the presented framework, particularly in terms of additional analysis modules, it is hoped that such techniques can play a larger role in the design phases of future reusable launch vehicles. Further work will seek to optimise full RLV configurations, with the inclusion of thermal constraints for maximum dynamic pressure flight states. The key disciplines yet to be included within the framework: trajectory and propulsion integration/performance, will be studied and implemented at a similar fidelity to the current analysis modules. Furthermore, higher fidelity aero-structural models will be utilised in both reduced design space optimisations, and large design space multi-fidelity environments. Incorporating these features while ensuring modularity of the framework is maintained will produce a powerful set of tools, capable of handling a wide range of problem spaces at various stages of the design process.

Chapter 7

Global Super-Hypersonic Vehicle Optimisation

The primary goal of this research is to produce novel designs for super-hypersonic vehicles. This requires a fully parametric vehicle definition, and sufficient analysis to provide a meaningful quantification of optimality. An improved fuselage definition is therefore employed, allowing a broader scope of designs to be generated, compared to that of the previously described constant cross-section fuselage design space. Furthermore, this definition allows lifting-body optimisations to be carried out, in addition to wing-body design spaces. In terms of analysis, a mass estimation technique is implemented, allowing trim characteristics, overall vehicle mass, and payload fractions to factor into objective and constraint functions. Both cruise and max-Q flight conditions are considered, with the former used to define optimality in conjunction with uncertainty analysis, while the latter ensures maximum surface temperatures are not exceeded. Static stability is also considered in both longitudinal and lateral dimensions, further quantifying the feasibility of produced vehicles.

The highly constrained design space requires careful consideration in terms of constraint handling: allowing infeasible designs to tend towards feasible regions of the design space; ensuring constraints are not unintentionally weighted more importantly than others; while maintaining a high quality output of feasible Pareto front results. To this end, a continuation of the previously described constraint handling methods is employed here, allowing infeasible designs to progress through the optimisation, finding local regions of feasibility, before slowly prioritising non-violating designs only.

7.1 Vehicle Generation

7.1.1 Arbitrary Fuselage Definition

To maximise control over the fuselage shape, both the cross-sectional shape and longitudinal distribution are defined independently through class shape transformations. Introduced by Kulfan [104], class functions allow a wide variety of geometries to be generated, with shape functions providing control over critical geometric properties, such as leading edge radius. They are defined as follows:

$$C_{N_0}^{N_1}(x) = x^{N_0}(1-x)^{N_1} \tag{7.1}$$

$$S(x) = \sum_{i=0}^{n} A_i S_i(x) \tag{7.2}$$

$$S_i(x) = \binom{n}{i}(1-x)^{n-i}x^i \tag{7.3}$$

Where the powers N_0 and N_1 define the shape at the beginning and end of the curve respectively, and A_i are user defined polynomial coefficients. As in the work presented by Di Giorgio et al. [27], a single class shape transformation is used for streamwise distribution, whereas upper and lower functions are defined separately for the cross-sectional shape.

$$C_{C_{U,L}}(\eta) = w\left[\eta^{N_C}(1-\eta)^{N_C}\right],\ S_{C_{U,L}} = \sum_{i=0}^{n} A_{C_i}\cdot S_i(\eta) \tag{7.4}$$

$$C_D(\varepsilon) = \frac{h}{2}\left[\varepsilon^{N_{D_0}}(1-\varepsilon)^{N_{D_1}}\right],\ S_D = \sum_{i=0}^{n} A_{D_i}\cdot S_i(\varepsilon) \tag{7.5}$$

Where η, ε are the non-dimensionalised length and radial coordinates, and w, h are desired width and height of the vehicle. Also note that since the fuselage is symmetric in the x-axis, only one value of N_C is used in the cross-sectional class function.

Combining these functions in both cross-sectional and streamwise dimensions results in a point cloud representation of the fuselage. With x and y coordinates derived from their respective non-dimensional counterparts, the remaining dimension to be determined, z, is found from the following:

$$z_{U,L}(\varepsilon,\eta) = C_C(\eta)S_{C_{U,L}}(\eta)C_D(\varepsilon)S_D(\varepsilon) \tag{7.6}$$

Thus, the entire fuselage is defined by four edge controlling parameters ($N_{C_{U,L}}, N_{D_{0,1}}$), two scaling factors (w, h), and three vectors of Bernstein polynomial coefficients ($\mathbf{A}_{C_{U,L}}, \mathbf{A}_D$). The num-

ber of coefficients in longitudinal and radial polynomials can vary depending on the users desired control over the respective dimensions.

In this work, four coefficients define Bernstein polynomials. However, with streamwise symmetry, only two have to be defined for cross-sectional upper and lower distributions. The fuselage is discretised using cosine spacing in both stream and span-wise dimensions, to focus on accurate depictions of leading and trailing edges, as well as upper and lower surface joints, where maximum curvature is often present.

7.1.2 Aerofoil & Wing

A large design space is also desired in the generation of lifting surfaces. The shape optimisation work carried out in chapter 4 showed the direct Bézier curve aerofoil definition to have highest level of control in creating such geometries. Furthermore, it validated the use of design space boundaries and allowed geometric constraints to ensure only feasible designs are seen as optimal. Therefore, the same design space and constraints have been implemented in this optimisation, enabling optimal aerofoil designs to be searched for a global level, producing diverse shapes that are tailored to maximise performance of a given vehicle.

Wings are produced using the same methodology as chapter 6, with two partitions once again employed to allow variation of planform variables across the span. Two aerofoils are used to define cross-sectional properties, with the outer partition maintaining constant cross-section shape. The inclusion of a third aerofoil at the wing-tip provides little potential performance improvement in contrast with the large increase in number of variables. Such insensitivities are to be avoided generally in design optimisation, and were therefore omitted from this work in favour of a more concise design space. Allowing the wing to vary in aerofoil shape from root remains a necessity however, with inboard strakes generally employing thinner non-dimensionalised sections to account for their increased chord.

7.1.3 Improved Wing-body Merging

One of the major contributors to non-linearities in the cost function space is the merging of wing-body shapes. Defining these geometries separately before attempting to combine them in a single non-intersecting surface will undoubtedly add noise to the overall optimisation, since the intersection points are dependent on aerofoil, wing, and fuselage definitions, and thus cannot be quantified in advance of vehicle generation. Nonetheless, defining the initial wing partition from the vehicle centreline becomes less intuitive when the fuselage is allowed to vary in cross-sectional shape. This is due to the wide bodies produced by such a design space, increasing the proportion of wing-span that will be discarded during the merging process (see figure 6.3). This

creates a clouded design space in terms of inboard aerofoil definition and partition span, since the prevalence of these design choices on vehicle performance will vary heavily between wide and narrow fuselages. Furthermore, variable boundaries in terms of planform shape become harder to define when the extremes of fuselage cross-section definitions have to be considered, with difficulties in enabling large dihedral wings to be consistently merged with wide fuselages when defining wings from the vehicle centreline, for example.

An improved wing-body merging technique has therefore been devised, to alleviate some of the non-linearities stemming from this combination process, as well as allowing variables to be defined consistently, regardless of fuselage shape. Instead of defining wings from the vehicle midline, a scattered interpolant is generated from the fuselage surface, with wing x-offset used to determine the position on the fuselage at which the leading edge intersection point will occur. The wing is then displaced to this position, and projected inboard into the centreline, allowing the remaining intersection points to be computed. This enables wing position to be physically defined on the fuselage surface, rather than at the midline. Furthermore, it allows lifting surfaces to be placed anywhere on the vehicle with relative ease, giving a consistent definition to partition span (now wet span), and inboard aerofoil section, which is now defined at the point of merge, rather than vehicle centreline.

7.1.4 Mass Estimation

In order to determine system level characteristics such as potential payload fraction and trim conditions of the vehicle, a weight estimation technique must be employed. As with all analysis modules in a large design space optimisation, versatility and rapid results are a necessity. The suite of statistical correlations found in the HASA [137] has therefore been implemented in the current framework.

Specifically designed to handle a wide variety of hypersonic vehicle configurations, this iterative scheme computes component weight contributions from the gross weight, and geometric inputs such as fuselage length and reference area, until convergence. Normally, fuel fraction is also an input, however here the data in Harloff et al. [137] is used to estimate fuel fraction as a function of gross weight throughout iterations.

Payload fraction can be implemented as a design variable, however it is more intuitive to compute maximum payload from usable fuselage volume. With the fuel fraction known, the required propellent volume can be calculated. This value is taken away from the overall usable volume, allowing payload to be determined. A constant efficiency factor of $\eta = 0.7$ is applied to the volume enclosed by the fuselage, to account for structural materials and fuel tanks, and a constant payload density of 50kgm^{-3} assumed. This method removes redundant fuselage areas that

would be present in the design variable approach, while having the secondary effect of ensuring the converged system can house the required propellent, which is not guaranteed by the HASA method.

7.2 Optimisation Problem

Once again multi-objective, multi-flight state optimisations were conducted. In this instance, the inclusion of a versatile fuselage definition allows both lifting-body and wing-body design spaces to be analysed. Furthermore, with a fully parametric definition of vehicle components; a mass estimation routine specific to high-speed vehicles; and robust constraints covering thermal thresholds, stability criteria, and system level design; these optimisations can be carried out in a global context, allowing all vehicle components to be defined by the optimiser.

Aerothermodynamic, trim and vehicle sizing considerations are used in defining objective and constraint functions. Three cost functions are employed: overall vehicle mass, aerodynamic efficiency at trim angle of attack in the cruise flight state, and payload fraction, allowing a broad spectrum of designs to compete for optimality on the Pareto front. Ultimately, a diverse set of results is desired, with common characteristics across all designs, or those performing best for a particular cost function, highlighted for further analysis.

Configurations are analysed at an altitude of 30km and Mach numbers of 5 and 8, to test vehicles in typical cruise and max-Q conditions. Rather than defining a set angle of attack, the vehicles are trimmed at each flight state, in order to define performance objectives and constraints. To avoid over-optimisation of aerodynamic efficiency at the design points, a robust design method is employed. Instead of deriving performance metrics from the given flight states alone, configurations are tested at slightly off design points, and aerodynamic performance determined by the summation:

$$\text{min: } \mu(f) + n\sigma(f)^2,\ f = -\frac{C_L}{C_D} \tag{7.7}$$

Where n is the sigma level used to weigh the influence of variance on the cost function, set to $n = 6$ in the optimisation problem described here.

The overall optimisation problem is shown in table 7.1, with design variable boundaries employed in table 7.2. Note the number of non-dimensionalised variables, which are put in place to allow reasonable design choices to be made based on key parameters such as fuselage length.

Without this, variable boundaries would have to be captured for vehicles both large and small, which would result in infeasible designs such as minimum length fuselage combined with maximum wing chord or span.

Aerodynamic properties are output with respect to the computed trim angle of attack for a given vehicle, which for feasible configurations directly enforces $L = W$. However, vehicles unable to trim as a result of large weight estimation will still output aerodynamic characteristics at the highest allowable trim angle, set to 35°. Therefore, an additional constraint is required to ensure the vehicle has trimmed at the given flight state. This additional check is preferred to discarding vehicles that fail to trim, allowing such designs to tend towards feasibility. With trim analysis complete, small perturbations are made to angle of attack and sideslip, allowing vehicle longitudinal and lateral static stability to be determined. The output derivatives are used as further constraints, ensuring optimal designs are stable as well as aerodynamically efficient. Finally, the surface temperature output at Mach 8 trim conditions must remain below a set maximum, which has been defined to represent material allowables.

Geometric constraints are also invoked to ensure feasibility among designs, which cannot be guaranteed by the parametric design space alone. Internal fuselage volume must be large enough to house the computed payload and fuel volumes calculated using the HASA program. A combination of variables and constraints ensures that wings decrease in both chord and thickness outboard, while also asserting reasonable leading and trailing edge sweep angles. Aerofoil sections require a variety of constraints to guide the optimisation toward desirable designs, due to the wide scope of Bézier control point variables. These ensure reasonable thickness distribution and surface curvature, and have been validated against a large set of comparable aerofoils found in the UIUC database [107], as discussed in chapter 4.

7.2.1 Constraint Handling

Due to the highly constrained nature of this problem, a continuation of the constraint handling methods discussed in chapter 3 was incorporated in this work. Recalling the ε-constrained method introduced by Takahama and Sakai [82], the constrained optimisation replaces the usual cost function comparison between solutions with an ε level comparison, which for $\varepsilon \geq 0$ takes the form:

$$(f_1, \phi_1) \leq_\varepsilon (f_2, \phi_2) \Leftrightarrow \begin{cases} f_1 \leq f_2, & \text{if } \phi_1, \phi_2 \leq \varepsilon \\ f_1 \leq f_2, & \text{if } \phi_1 = \phi_2 \\ \phi_1 < \phi_2, & \text{otherwise} \end{cases} \tag{7.8}$$

Table 7.1: Global super-hypersonic vehicle optimisation problem definition

		Comments
Minimise	Mass, kg	
Maximise	$(C_L/C_D)_{trim}$	
	Payload fraction	
Subject to		
Variable boundaries	$\mathbf{x}_{lb} \leq \mathbf{x} \leq \mathbf{x}_{ub}$	Defined in table 7.2
Performance based	$L \approx W$	Trim conditions
	$C_m \approx 0$	
	$C_{m_\alpha} < 0$	Stability criteria
	$C_{L_\beta} < 0$	
	$T_{max} \leq 2050\text{K}$	Assumed thermal threshold
Geometric based	$V_{pay} + V_{fuel} \leq \eta V_{fuse}$	Sufficient internal fuselage volume
	$\frac{dt}{db} \leq 0$	Wing thickness cannot increase outboard
	$30^\circ \leq \Lambda_{le} \leq 80^\circ$	
	$0.1 \leq c_{t_{max}} \leq 0.7$	Non-dimensional chord location of maximum aerofoil thickness
	$\left(\frac{dt}{dc}\right)_{t < t_{max}} \geq 0$	Aerofoil thickness cannot decrease up to maximum thickness
	$\left(\frac{dt}{dc}\right)_{t > t_{max}} \leq 0$	Aerofoil thickness cannot increase after maximum thickness
	$\left(\frac{d^2u}{dx^2}\right)_{u < u_{max}} \leq 0$	Upper surface curvature cannot increase up to apex

The value of ε therefore dictates whether f_1 is an improvement over f_2 based on the standard cost function comparison, or by a comparison of constraint function values. By dynamically reducing the value of ε throughout the optimisation, violating particles are able to tend towards feasibility, rather than being prematurely discarded, thus improving the optimiser's ability to search for feasible regions of the design space.

The initial value of ε is defined by the violation functions of the initial population, as shown in equation (7.10), and is dynamically reduced until some final iteration t_f, after which the constraint tolerance remains at a constant value ε_f. The reduction method introduced by Takahama and Sakai rapidly decays constraint tolerance during early iterations, before slowly tending towards a positive final value. However, if only non-violating particles are to be considered optimal, ie. $\varepsilon = 0$ at $t > t_f$, which is the most logical way to handle constraints, the decay function cannot be employed in its current format.

Table 7.2: Global super-hypersonic vehicle design variable boundaries

	Variable	Minimum	Maximum	Comments
Fuselage	l, m	10	40	
	w	0.1	0.8	Dimensionalised by fuselage length
	$h_{U,L}$	0.1	0.8	
	$N_{C_{U,L}}$	[0.1 0.001]	[1 1]	Span-wise upper and lower edge control
	$N_{D_{0,1}}$	[0.25 0.001]	[0.95 0.8]	Longitudinal edge control
	$\mathbf{A}_{C_U}$	0.05	1	Span-wise control points
	$\mathbf{A}_{C_L}$	-1	-0.2	
	$\mathbf{A}_D$	0.3	1	Longitudinal area distribution
Wing	c	[0.25 0.2 0.1]	[0.9 0.8 0.7]	Dimensionalised by fuselage length
	Λ, deg.	-45	45	Partition trailing edge sweep
	b	0.1	0.5	Dimensionalised by fuselage length
	ϕ, deg.	0	45	Partition dihedral
	x_0	0.1	0.75	Non-dimensional leading edge offset from nose
Aerofoil	$\mathbf{x}_{u,l}$	$1-\cos\frac{i\pi}{2n}-\frac{1}{n}$	$1-\cos\frac{i\pi}{2n}+\frac{1}{n}$	Half-cosine spacing from 0 to 1, where n is number of free control points ($n=4$)
	$z_{u_{2-5}}$	0	0.2	Non-dimensional, control points fixed at extremities
	$z_{l_{2-5}}$	-0.2	0	

To allow a final constraint tolerance of zero, and thus eliminating violating particles from the Pareto front entirely, a second reduction method has been used here. After some intermediate time, t_m, the remaining positive constraint tolerance is linearly reduced to zero. This approach provides further flexibility in the selection of intermediate constraint tolerance, ε_m, allowing the user to experiment with rate of decay. The constraint tolerance value throughout a given optimisation is summarised as follows:

$$\varepsilon_t = \begin{cases} \varepsilon_0 e^{(-Bt)} & \text{if } t \leq t_m \\ (t - t_m)\frac{-\varepsilon_m}{t_f - t_m} & \text{if } t_m < t < t_f \\ 0 & \text{if } t > t_f \end{cases} \tag{7.9}$$

$$\varepsilon_0 = \frac{1}{2}\left(\frac{1}{n}\sum_{j=1}^{n}\phi + min(\phi)\right) \tag{7.10}$$

$$B = \frac{log(\varepsilon_0/\varepsilon_m)}{t_m} \tag{7.11}$$

$$\tag{7.12}$$

In this work, constraint tolerance is non-zero for half of the total number of optimisation iterations. The first two thirds of this time uses the Takahama and Sakai approach, reducing initial tolerance to $\varepsilon_m = 10^{-3}$. The final third then linearly reduces the constraint tolerance to zero. This allows an initial decay that is not overly steep, thus preventing a tightly constrained problem being dominated by minimisation of the violation function, rather than cost functions. Furthermore, the linear reduction provides ample time for fine tuning; allowing particles with small violations to find their way towards feasible design points.

As discussed in chapter 4, steps must be taken to ensure the optimisation is not driven by a handful of constraints. Without proper scaling, initial constraint tolerance may be much greater than the value chosen to tend towards ($\varepsilon_0 >> \varepsilon_m$), requiring rapid reduction during early iterations. In this situation, particles who violate the improperly scaled constraints will be quickly discarded, in favour of those who may show poor performance elsewhere. Typical or expected values of each constraint are therefore specific in the initialisation phase, and used to normalise constraint functions throughout the optimisation. This ensures that particles are not chosen due to one constraint typically being a higher magnitude than another. Furthermore, this normalisation of constraints guarantees that particles with a constraint value close to the intermediate value of ε_m will be close to feasible, regardless of whether or not a localised feasible region exists in the design space. Such particles should therefore be encouraged to find these pockets of feasibility, which is why a linear reduction of $\varepsilon_m \rightarrow \varepsilon_f = 0$ has been employed.

7.3 Results

7.3.1 Lifting-body

The Pareto front for lifting-body design space is shown in figure 7.1. Particles are coloured in order to demonstrate their relative objective function values with respect to the Pareto popula-

tion. Here, red indicates a high aerodynamic efficiency, green tends towards minimal mass, and blue denotes high payload fraction. Two-dimensional views of the Pareto front are also shown in figure 7.1, with colour intensity showing particle performance in the third dimension. The Pareto front was limited at 300 particles, to allow a large number of optimal designs, while maintaining variation through crowding distance. The optimisation was run for 1000 iterations with 120 particles across 6 cores, and completed in approximately 12 hours.

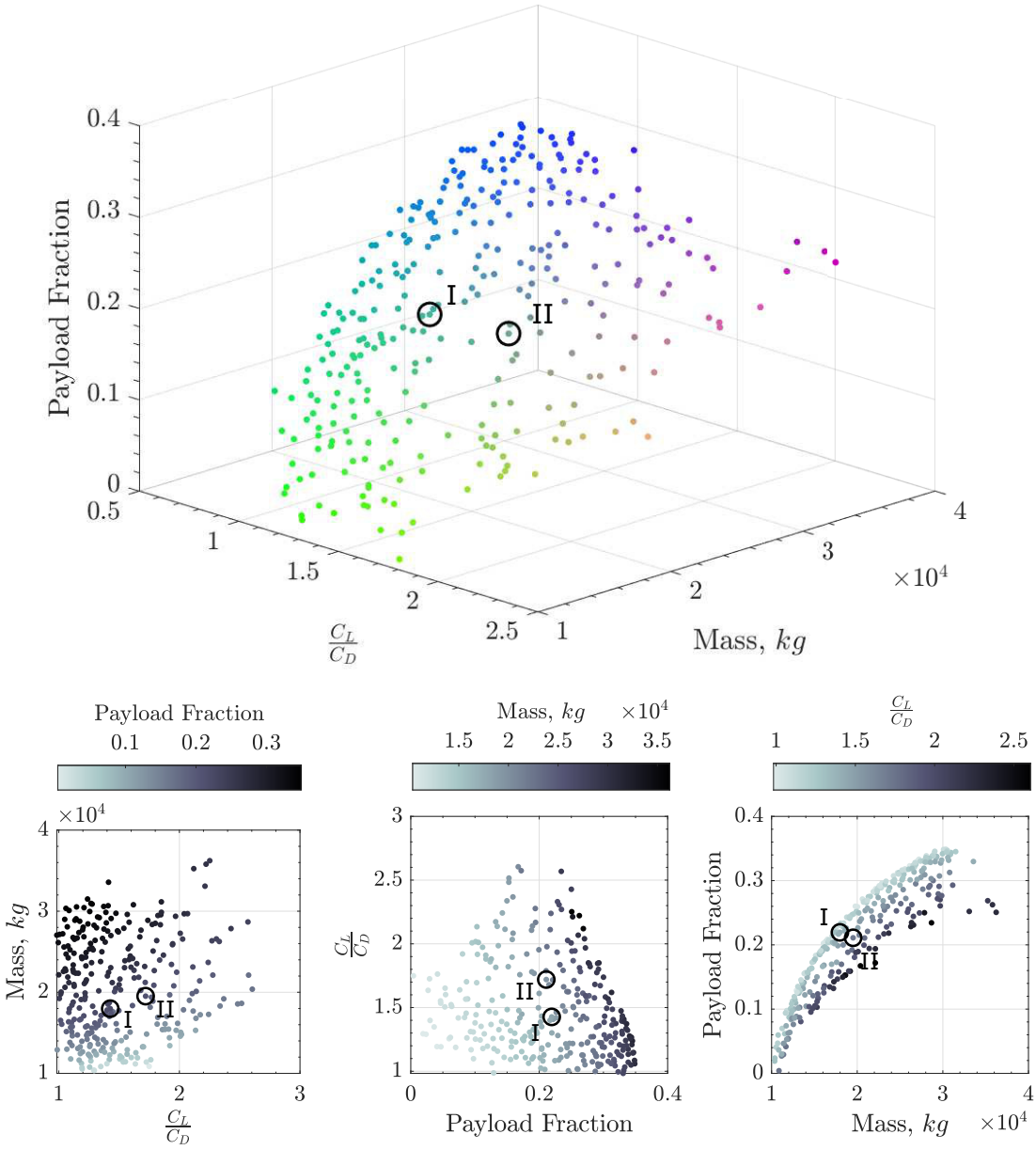
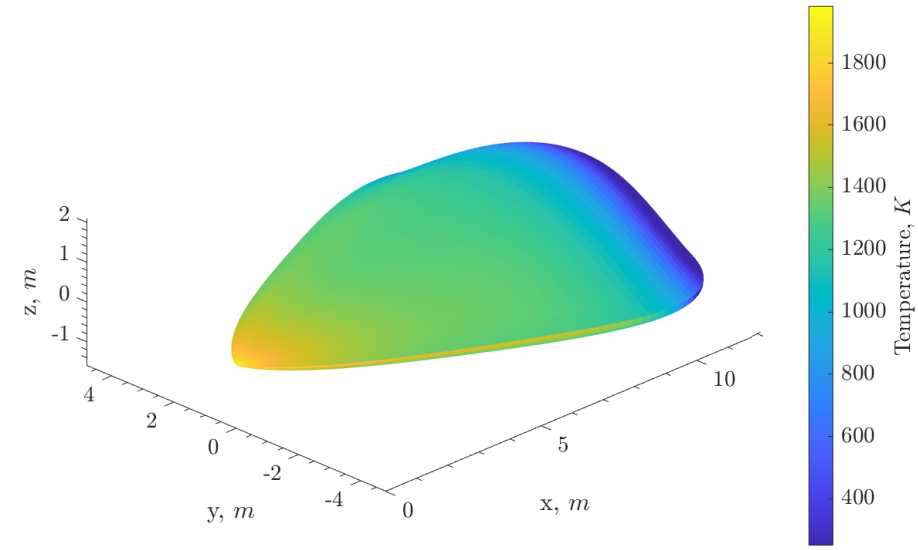


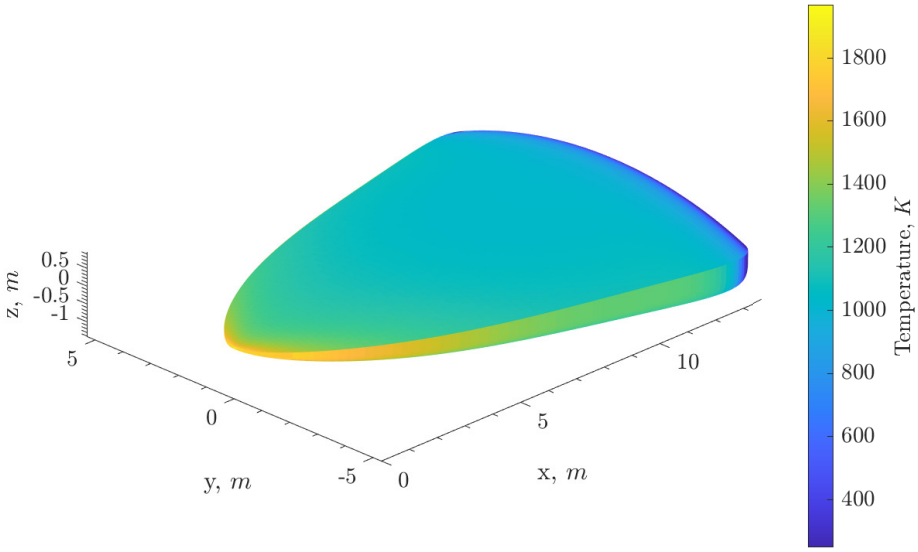
Figure 7.1: Pareto front visualisation of lifting-body optimisation

As expected, a highly competitive front exists between mass minimisation and payload fraction maximisation, with configurations behind this front demonstrating improved aerodynamic effi-

ciency. Various nuanced designs are present within the Pareto front, with similar cost function values not necessarily correlating to similar designs. This is to be expected within a large design space, multi-objective optimisation, with a number of loosely correlated regions of the design space mapping to similar areas of the cost function space. An example of this is illustrated in figure 7.2, with two trade-off designs exhibiting varied characteristics despite largely similar cost function values.



(a) Configuration I



(b) Configuration II

Figure 7.2: Trade-off designs with surface temperature distribution at Mach 8 from tri-objective lifting-body optimisation

Configuration I lies close to the optimal mass-payload front, utilising bell shaped curves on the upper surface to increase payload volume while maintaining aerodynamic efficiency. Note that

prior to the inclusion of lateral stability as a optimisation constraint, vehicles would exhibit this shape on the lower surface also. With lateral stability demanded, vehicles are forced to maintain a convex, and relatively flat lower surface. Configuration II presents an almost wedge like shape, compensating for volume lost at the centre with increased height toward the extremities. This allows a reasonable payload fraction to be maintained, only showing a slight reduction compared to configuration I. Mass is also increased due to an increase in length compared to configuration I, with aerodynamic efficiency being the main improvement due to minimal frontal area.

Similarities between vehicles are also present throughout the Pareto front, which is visualised in figure 7.3. Here, normalised design variables are collected from configurations up to the 10th percentiles in each cost function, along with trade-off designs, and shown in terms of mean and quartile boundaries. This illustrates the general trends associated with each cost function, along with design variable sensitivities, allowing key performance characteristics to be determined. Unsurprisingly, minimal mass vehicles consistently minimise overall length, with some flexibility around height. Width is also smallest within the minimal mass subgroup, but remains reasonably high to maintain lift and static stability, as well as sufficient internal volume. In line with configuration I, fuselage height is consistently minimised for optimal aerodynamic efficiency, with maximum payload designs being much more liberal to accommodate the required volume.

The remaining parameters are considerably less stringent, allowing a number of different shapes to be considered optimal within each subgroup. Lower longitudinal edge parameters (N_{C_1}) must remain relatively high to avoid bell shaped underbodies that violate static stability constraints, while leading edges (N_{D_0}) balance reduced drag and maximum heating. One clear trend across all subgroups is the steady increase in longitudinal area distribution ($A_{D_{0-3}}$), which is also shown in both configurations in figure 7.2.

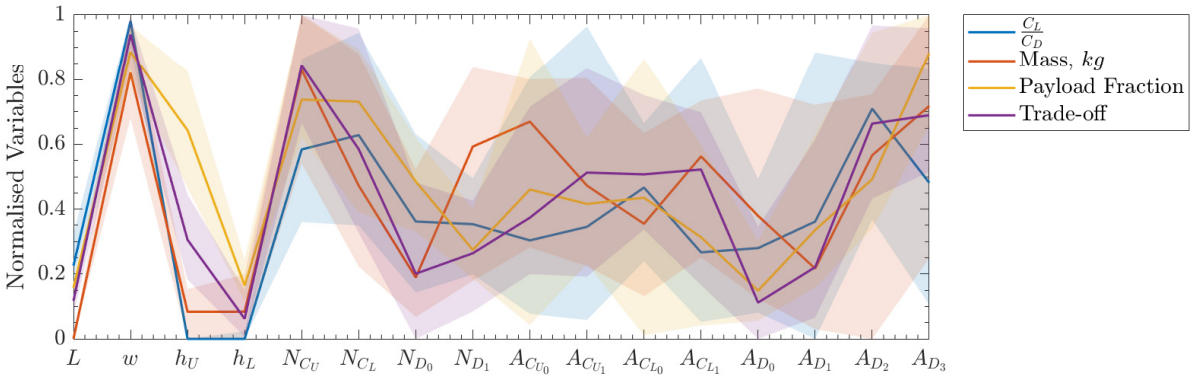


Figure 7.3: Parallel plot displaying mean and quartiles of normalised lifting-body variables representing 10th percentile performers in each cost function, and trade-off designs

7.3.2 Wing-body

Results for the wing-body design space are presented in a similar manner to the preceding case, with three and two dimensional views of the Pareto front shown in figure 7.4. Once again a highly competitive front exists between payload fraction and vehicle mass, with the former slightly lower than the lifting-body case overall, due to the addition of wing weight. Aerodynamic efficiency also struggles to match those found in the lifting-body design space, which on the surface appears counter intuitive. However, this is explained by the need for increased fuselage height, in order to merge with generated wings, and therefore increased frontal area with respect to incoming flow.

Two trade-off designs are once again extracted from the Pareto front, shown in figure 7.5, demonstrating varied characteristics despite similar cost function values. Configuration III presents the bell shape upper surface commonly found in the lifting-body Pareto front, with highly swept wing sections creating a blended wing-body vehicle. Configuration IV on the other hand exhibits a more typical wing-body design; merging a fully convex fuselage with larger, moderately swept wing sections. These thinner bodies are viable within this design space, with wings compensating for reduced fuselage lift. Configuration IV also exhibits a considerably smaller nose than any of the previously observed designs. This is reflected in its maximum surface temperature, which lies on the constraint boundary.

An overall view of optimal subgroups is displayed in figure 7.6, with the inclusion of wing design variables. Note that the body design space remains the same as the lifting-body optimisation, and thus direct comparisons between resulting designs can be drawn. A reduction in fuselage width is evident, with wings producing enough lift to allow an increase body fineness ratio. Trade-off designs retain a relatively high width however, maximising body lift and internal volume to retain reasonable payload fractions. Optimal payload designs contain longer fuselages with high fore-body distribution compared to other subgroups, preferring bulkier shapes over maximum width. Large base regions are present amongst the majority of designs, maximising usable volume while maintaining aerodynamic efficiency.

In terms of wing design, lateral static stability remains a key driver, with low dihedral present across the majority of initial wing partitions, minimising the impact of perturbed sideslip. Root chords remain high across subgroups, with maximum mass employing larger tip chords in combination with reduced outboard partition span. Optimal aerodynamic efficiency unsurprisingly shows higher usage of large span partitions combined with high rearward sweep, while all designs typically employ a reasonably high wing leading edge offset to ensure static longitudinal stability.

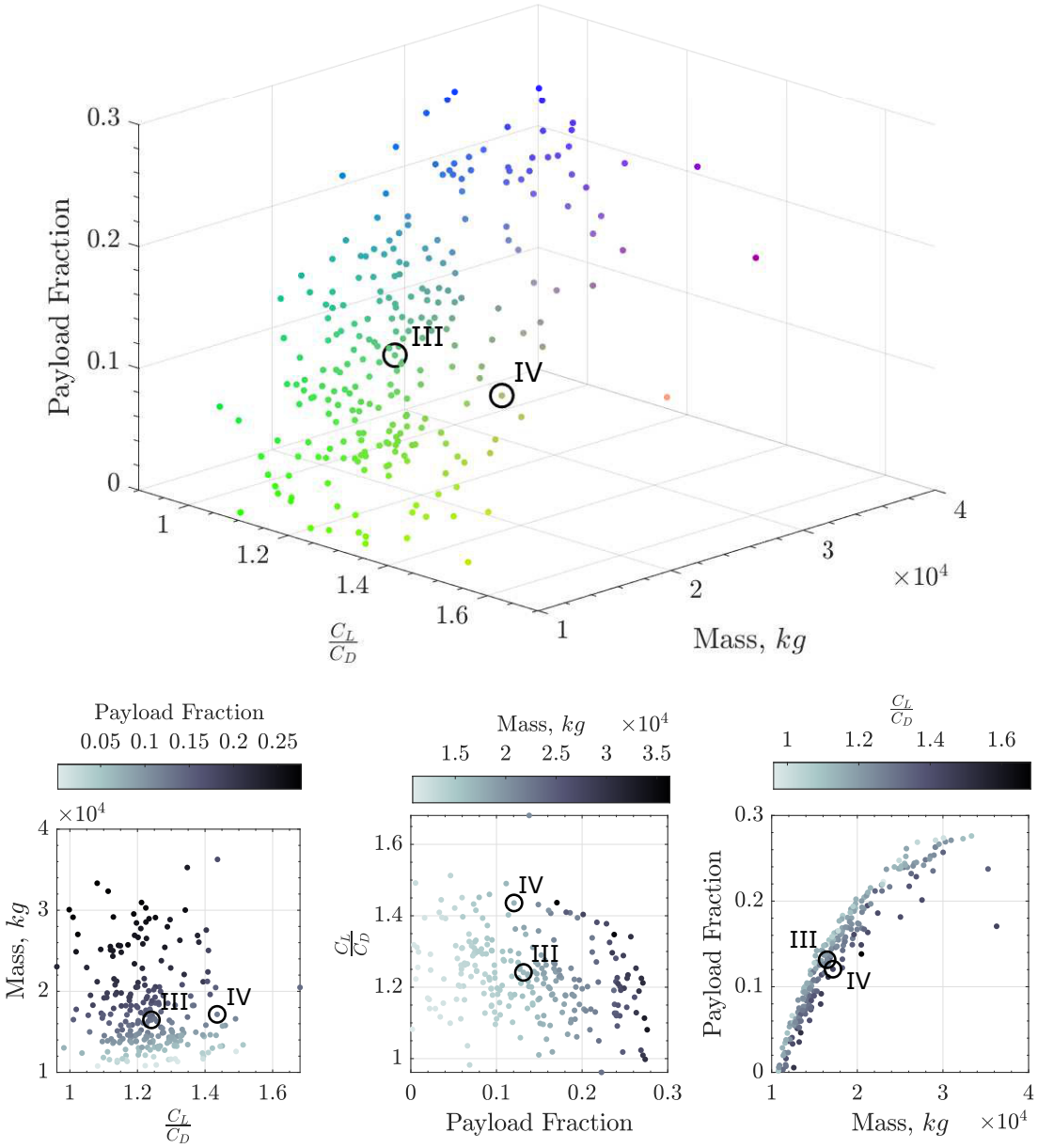
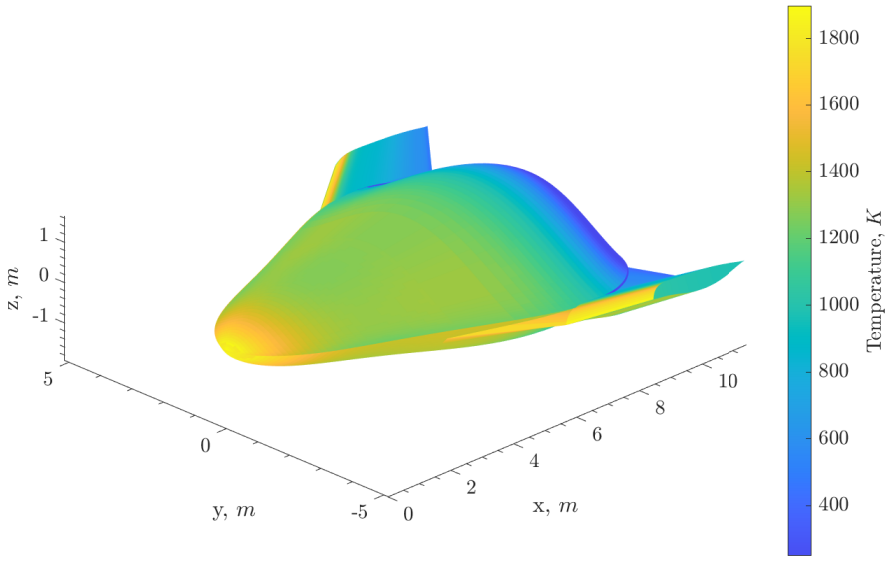


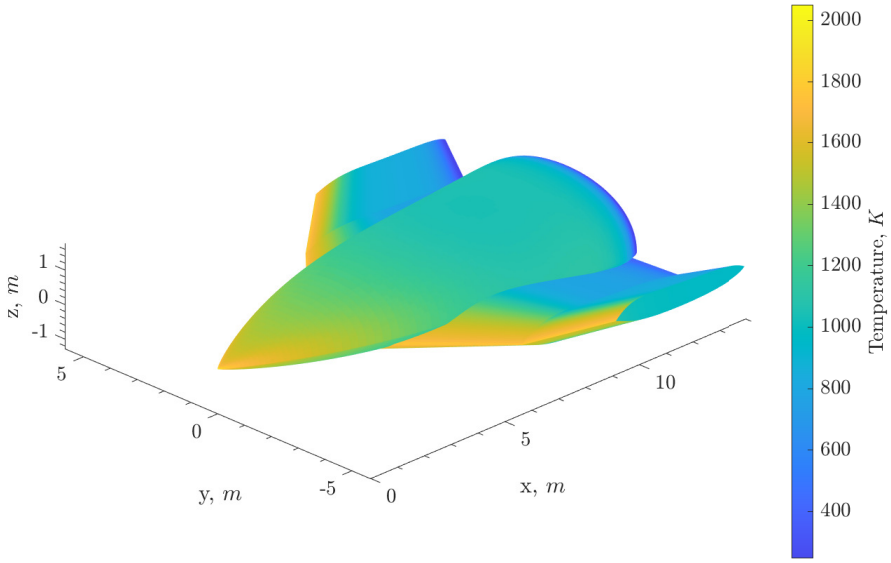
Figure 7.4: Pareto front visualisation of wing-body optimisation

7.4 Conclusion

This chapter brings together the analysis and optimisation methods discussed previously to conduct global multi-objective, multi-disciplinary design optimisations of full super-hypersonic vehicles. The previously introduced wing and aerofoil generation techniques are combined with a versatile parametric fuselage generator, allowing the optimiser to produce novel and diverse results. The low-fidelity aerodynamics module is combined with a RLV/HST mass estimation technique, enabling trim characteristics and payload fraction to further quantify overall vehicle performance and optimality.



(a) Configuration III



(b) Configuration IV

Figure 7.5: Trade-off designs with surface temperature distribution at Mach 8 from tri-objective wing-body optimisation

To allow for a large design space to be implemented, without producing undesirable or infeasible designs, a robust constraint handling technique has been utilised, allowing geometric and performance based constraints to be enforced directly, rather than placing overly tight boundaries on the design space. Since the combination of large design space and stringent constraints can lead to a highly non-linear objective space, an improved variable constraint tolerance method has also been implemented. This allows violating configurations to be chosen in the various selection processes during the early phases of the optimisation, avoiding the over dependence on constraint function minimisation that often leads to pre-mature convergence of the optimiser.

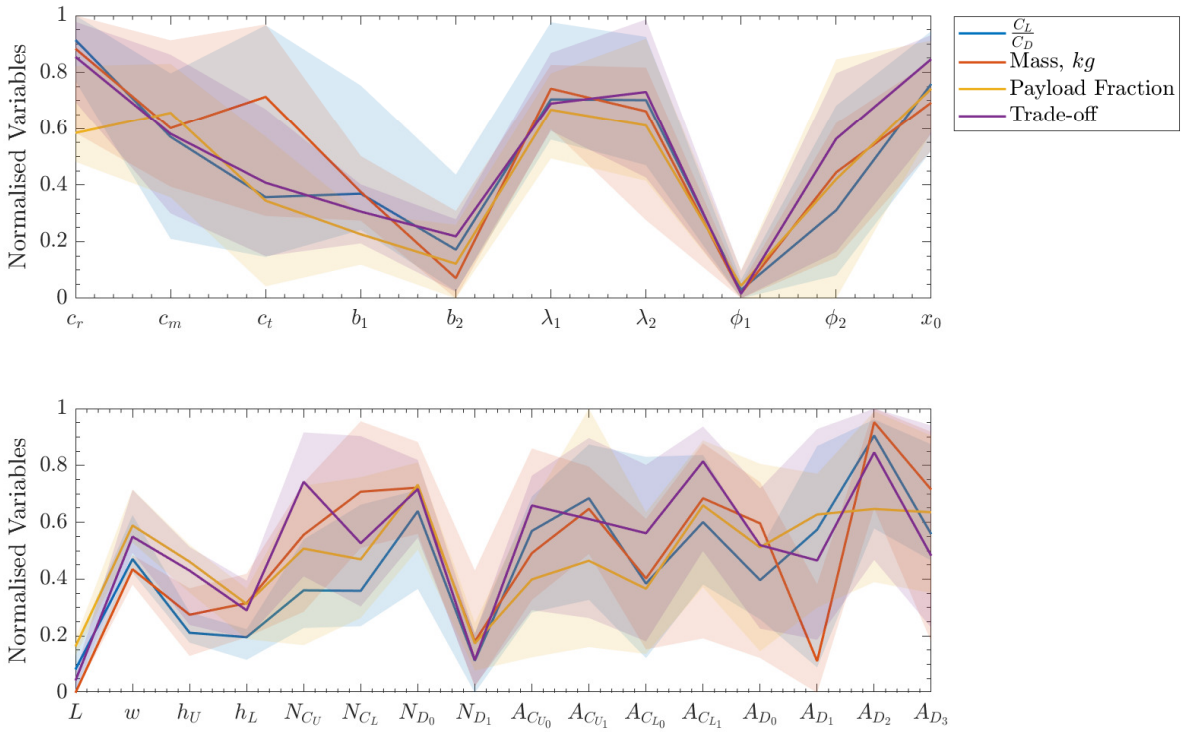


Figure 7.6: Parallel plot displaying mean and quartiles of normalised fuselage (upper) and wing (lower) variables representing 10th percentile performers in each cost function, and trade-off designs

Results are shown for a multi-objective multi-disciplinary design optimisation, where arbitrary lifting-body and wing-body design spaces are analysed in terms of mass, aerothermodynamics and trim conditions for typical cruise and max-Q flight states. Uncertainty is included in terms of aerodynamic efficiency to ensure that designs are not being over optimised for a specific design point. Realistic constraints are enforced in order to increase the viability of output designs, with further analysis disciplines such as trajectory and propulsion models sought to build upon this in future work.

Overall, the wing-body design space exhibits poorer convergence characteristics than the lifting-body case, which is observed from comparison of the resulting Pareto fronts. This is somewhat expected due to the considerably increased number of optimisation variables required for wing design; in terms of both partition planform shape, and aerofoil definitions necessary for inboard and outboard sections. Nonetheless, the non-linearity induced by merging wing-body shapes also plays a major role in convergence. Future work should therefore attempt to define fuselage and wing/tail surfaces in a more continuous fashion, to enable the optimiser to search the design space more efficiently. Such an approach would have the added effect of producing more realistic joint sections between the two surfaces, providing a better understanding of aerodynamic efficiency, and allowing interpolation methods to be more readily applied, if such sections are

continuous in nature.

Furthermore, combining these design spaces to allow both lifting-body and wing-body designs in a single optimisation should be considered. This has been experimented with in the current research, however the binary nature of this problem is not suited to continuous optimisation methods, requiring either a continuous design space as discussed above, or a hierarchical optimisation technique.

Chapter 8

Conclusions & Future Research

The overall goal of this research was to develop robust tools for large design space exploratory optimisation of super-hypersonic vehicles. The author's first hand experience of iterative design cycles has shown the inefficiencies and over reliance on experience, and/or a pre-existing design. The world is becoming increasingly data driven, and modern day computational power allows for huge parametric studies and optimisation work to be carried out. While many aerospace organisations have embraced these methods within the design process, they are often limited in scope, and do not allow for the assessment of novel configurations. In doing so, the relative performance of designs, whether it be at the beginning or end of the design process, cannot be quantified in a global context.

The current framework has therefore been developed in an attempt to provide such context; allowing global design spaces to be thoroughly analysed, and initial designs output based on entry level performance modelling. This ability to rapidly quantify high performing regions of arbitrary design spaces has far reaching applicability, with high-speed transport and reusable launch vehicles being the focus of this research, due to their historical stagnation and high variability among current concepts. The high cost and risks associated with these design processes should therefore seek to maximise the available computational power to explore novel designs, and identify design characteristics with promising performance.

The ability to generate and analyse these global design spaces requires a high degree of flexibility in the chosen methods. Design variables for the various geometric components must be capable of generating a wide scope of possible designs, in a consistent manner that does not introduce unnecessary noise within the resulting cost function space. Analysis methods must be capable of quantifying performance regardless of the input geometry, handling various nuanced and non-standard shapes.

Such methods have been incorporated here, along with optimisation procedures that promote a

diverse search of designs, while ensuring the combination of large design space and low-fidelity analysis methods are not exploited to produce mathematically optimal, but real-world infeasible designs. This approach of quantifying optimality subject to feasibility constraints is absolutely necessary to minimise inaccuracies produced by low-fidelity analysis methods, and over optimisation of the disciplines analysed, at the expense of others.

It must be emphasised that this research is only a starting point, with a number of improvements in existing architecture, along with implementation of further analysis and optimisation methods required to reach meaningful results in an industrial context. This work has focussed on the initial ground-up development of the necessary framework, prioritising the key requirements of vehicle generation, aerodynamic analysis, and optimisation tools. These implementations have been benchmarked and validated to ensure their fitness for purpose within the context of global design optimisation, before carrying out such problems for a number of design spaces. Branches of this research have extended into further analysis modules, along with multi-fidelity optimisation, scratching the surface of areas under constant development within the wider engineering community.

8.1 Analysis & Optimisation Modules

8.1.1 Aerodynamics

The scope of this work required rapid analysis to be carried out, with thousands of designs requiring performance characterisation in a given optimisation. Moreover, analysis methods had to be capable of producing reliable results for arbitrary designs, considering the envisaged large design spaces and heuristic optimisation techniques employed. Low-fidelity, panel based methods were therefore selected for aerodynamic analysis, allowing performance to be derived in a rapid fashion from freestream flow and surface properties alone.

A number of inviscid methods were implemented for the various combinations of inclination angle and geometric components anticipated in this work, allowing the relevant physics to be appropriately estimated. Their implementation allowed situation dependent default techniques to be set, with relevant checks in place to enable autonomous method switching on a panel by panel basis, if defaults were inappropriate. The inclusion of viscous methods allowed local approximations of skin friction and surface temperature to be made, further quantifying vehicle performance and feasibility.

Various validation cases were considered, comparing the implemented methods to experimental and CFD data found in literature. These captured body shapes, lifting surfaces and overall

vehicles; verifying correct implementation of the described methods, and validating their use within a conceptual design context.

8.1.2 Optimisation

In tandem with analysis methods, the envisaged large design space exploration in this research required the implementation of versatile, rapid optimisation methods. Particle swarm was therefore selected as the primary optimisation technique in this work, due to its population based nature, and improvement over similar methods when employed within continuous design space problems. Both single and multi-objective variants were deployed, with a focus on diversity throughout.

Constraint handling also factored heavily into the development of these techniques, with both penalty and adaptive tolerance methods utilised. The combination of exploratory design spaces and highly constrained optimisation problems required effective determination and handling of violating designs, allowing infeasible regions of the design space to be investigated for pockets of feasibility. Appropriate handling of such designs ensures optimisations are repeatable, and not dominated by the first non-violating solution.

The developed optimisers were tested against industry standard population based methods, for both constrained and unconstrained single and multi-objective problems. Performance gains were evident for the single-objective case, with the developed PSO either finding the global optima more reliably than other methods, or showing similar performance with a drastic reduction in cost function calls. Multi-objective PSO results also demonstrated equal to or better performance than industry standard methods, although both performed well for the given problems. Local optimisation methods were also employed, providing a further comparison within a relevant aerodynamic optimisation, which is recounted in the proceeding section.

8.2 Low-Fidelity Global Aerofoil Optimisation

Stemming from initial wing optimisation work, in which produced aerofoils performed well for the given cost functions, but were clearly infeasible in a real-world context, this research focussed much more on aerofoil optimisation than initially anticipated. The naive assumption that defining optimality alone would result in well behaved geometries led to additional focus being placed on constraint handling techniques, which are vastly under reported in similar research. Ensuring feasible designs are output requires either appropriate constraints, or limiting design variable boundaries. While the latter maybe be acceptable in local optimisations, the potentially vast reduction of feasible design space means that it cannot be considered global, regardless of

the optimisation method employed. Furthermore, placing such limits on constructive aerofoil methods vastly under utilises them, with deformative approaches from an initial design likely more suitable.

To benchmark the various generative aerofoil methods implemented, and validate their respective design space boundaries, shape optimisations were carried out, in which target aerofoils were to be replicated as closely as possible by each method. A large database of pre-existing designs was used for this, providing a robust validation test. Furthermore, these designs allowed feasibility constraints to be derived and validated, enabling their use throughout the remainder of this research.

Following this, aerodynamic aerofoil optimisations were carried out for both single-objective PSO algorithm, and a multi-point local optimisation technique. These optimisations were repeated multiple times for varying starting points, with the PSO algorithm producing improved aerofoils on each occasion compared to the best local results, while also showing a high level of consistency across runs. This provided further verification of the implemented optimiser, showing its fitness for purpose in a relevant optimisation problem.

While this consistency in aerofoil performance, combined with little variation in physical aerofoil shape, increased confidence that the globally optimal region of the design space had been found, it was acknowledged that this may not be the case for increasing fidelity. This hypothesis proved to be the case, with the disparity between fidelities particularly stark in the baseline case; where perceived improvements with respect to the baseline design in fact showed a performance degradation when analysed using high-fidelity methods.

8.3 Extension to Multi-fidelity

A logical continuation in the form of multi-fidelity optimisation was therefore conducted, allowing high-fidelity analysis to be integral in defining optimality throughout. To achieve this, a number of considerations had to be taken into account; namely automating the usage of high-fidelity solvers in a reliable and efficient manner within the overall optimisation process. Open-source grid generation and CFD software were employed through interfaces to the developed framework, with their usage validated by a parametric study in comparison to CFD data found in literature. To maximise the probability of convergence in automated CFD analysis, the previously derived geometric constraints were used as requirements, ensuring only well behaved aerofoils could be considered for such analysis.

Maintaining a global design space with regions of infeasibility therefore led to the implemen-

tation of a probabilistic surrogate model, allowing selective sampling to predict the difference in performance metrics between low and high-fidelities within regions of interest. A two level design of experiments was also implemented pre-optimisation, first testing a large sample population for geometric constraint violations, before selecting a diverse subset of feasible designs to generate the initial surrogate model.

Promising aerofoils were selected for high-fidelity analysis during the optimisation, in order to quantify uncertainties within the surrogate, confirming or rejecting the candidate aerofoil as the best observed so far. The predictive model was then updated to reflect the new observed data points, increasing prediction accuracy in explored regions. Initial usage of these techniques provided unsatisfactory results; often showing performance degradation in output aerofoils compared to CFD results of low-fidelity optima. This premature convergence was tackled by tending model predictions toward lower confidence boundaries during periods of optimisation stall, enabling more regions of the design space to be analysed; finding pockets of improved performance where limited sampling produces speculative predictions.

Multi-fidelity results proved a consistent improvement over low-fidelity analysis alone, demonstrating the usefulness of selective high-fidelity sampling in a predominately low-fidelity optimisation, when appropriate surrogate model techniques are employed. Furthermore, the methods invoked to maintain a global design space, while only requiring a relatively small number of high-fidelity observations to produce consistent results, show promise for further use within large design space optimisation work.

8.4 Multi-disciplinary RLV Wing Design Optimisation

With aerofoil generative methods, design variable boundaries, and feasibility constraints validated and successfully utilised within an aerodynamic optimisation, extending this research to three dimensional configurations was the natural next step. An initial wing-body generator was developed; defining wings on a planform partition basis, and fuselages from nose, constant cross section aft-bodies, and interpolating between them to determine fore-body shapes. This allowed for an approximate model of a previous RLV concept to be created, which was utilised for aerodynamic validation as well as a baseline configuration to benchmark designs in an optimisation context.

Structural considerations were taken into account for this baseline RLV wing optimisation, in which an approximate wing-box model was reduced to a 1D span-wise beam. Variables defining the wing-box model were incorporated within the design space, allowing minimisation of mass to be considered as an objective function within the optimisation, subject to structural integrity.

A mapping technique was required to transfer aerodynamic loads to the 1D structural model, and resulting displacements and rotations back to the aerodynamic mesh, along with an iterative technique to converge the overall aerostructural system. Wing-box failure criteria were defined in terms of material allowables for each skin and spar panel, with failing configurations continuing analysis in a rigid manner, allowing infeasible designs to tend towards feasibility, as opposed to employing a "death penalty" approach.

A multi-objective, multi-disciplinary design optimisation was carried out, with aerodynamic performance, trim angle, and wing mass defining optimality. Wing planform, aerofoil sections, and wing-box variables formed the design space, with constraints ensuring geometric and structural feasibility, along with performance characteristics relating to the baseline design. A number of nuanced designs in terms of wing and aerofoil shape were output, with structural designs showing a high degree of convergence.

8.5 Global Super-hypersonic Vehicle Design Optimisation

The final piece of optimisation work carried out in this research focussed on a full parametric representation of lifting-body and wing-body vehicles. To achieve this, an improved fuselage generation method had to be employed, allowing considerably more control over designs compared to the constant cross-section method previously defined. Merging of wing-body vehicles was also updated, to provide a more consistent definition of design variables within the optimisation.

System level characteristics were incorporated within vehicle analysis, utilising a component sizing and mass estimation framework developed specifically for HST and RLV configurations. This allowed trim characteristics to be determined for the wide spectrum of possible designs, along with estimations of payload and fuel fractions. Simple feasibility checks could also be put in place, ensuring internal fuselage volume was sufficient to house these estimated values. Further constraints were put in place regarding maximum allowable temperature, as well as static longitudinal and lateral stability derivatives, providing a vast improvement in the feasibility of output designs.

A number of methods were employed to avoid over-optimisation at design points, and appropriately handle violating geometries found during the optimisation. In particular, direct performance cost functions were replaced by statistical variants; analysing vehicles at slightly off design points, to incorporate uncertainty characteristics and reduce dependency on rigid flight states. Constraint handling techniques discussed in chapter 3 were updated to provide a smoother reduction of constraint tolerance throughout the optimisation, allowing infeasible characteristics

within configurations to be further explored and tended towards feasibility, rather than prematurely discarded.

Optimisations were carried out for lifting-body and wing-body design spaces, with objective functions encapsulating aerodynamic efficiency, vehicle mass and payload fraction. A diverse population of optimal designs was produced, despite the highly constrained cost function space. This provided further demonstration of the developed framework's capability; producing highly competitive Pareto fronts from large design space optimisation problems, that consider a number of intertwining disciplines to determine optimality and feasibility.

8.6 Continuation of Implemented Methods

The majority of future work unsurprisingly revolves around improving existing methods, and incorporating further analysis disciplines within the overall optimisation framework. The research carried out here has been intentionally broad, attempting to build a framework from the ground up which encompasses geometry definition, optimisation techniques, multi-disciplinary and multi-fidelity analysis methods. In covering this ground, there are many unexplored nuances and combinations of methods that could be analysed further. The proceeding discussion is by no means exhaustive, but it captures some aspects that would be powerful additives with relatively little effort required, along with some more vague ideas that require in-depth thought and planning. Furthermore, this discussion is mostly limited to areas which the author finds most interesting, both personally and with respect to furthering the state-of-the-art. Essentially, the developed framework could be pulled in any number of directions, with the following acting as a roadmap of where the current author would journey.

8.6.1 Geometry

One of the main goals of this research has been to open up the design space as much as possible, and quantify infeasible designs instead of setting arbitrary variable boundaries that enforce predetermined behaviour. With the exception of tailplane surfaces, the design space employed in chapter 7 is about as large as a continuous design space could be; parametrically defining the entire wing-body. Of course, many more variables could be added to increase control over each component, but the general scope would remain the same. However, a design space with discontinuities, and potentially hierarchical in nature, would allow multiple types of configurations to co-exist in the same optimisation. This was experimented with in the current framework; attempting to allow both wing-body and lifting-body designs, by removing wings below a set span. However, due to the highly constrained design space, this consistently resulted in the first type of non-violating configuration to dominate the Pareto front.

Instead of mimicking binary decisions using a continuous variable, it is much more logical to use a discontinuous values to make decisions at a system level. This would allow an interchangeable parts approach, with a number of competing designs in terms of wing-body, lifting-body, and tailplane configurations. Decisions variables could define further system level characteristics, such as propulsive systems or the integration of a body flap.

A number of geometry elements could therefore be added to the design space, which require little effort in modelling. The most time consuming part of including additional surfaces within large design optimisations is ensuring their consistent integration; accurately modelling boundary regions, and avoiding interference with other independent surfaces. Tailplane configurations for example, which have been experimented with in the current research, would require some error proofing to avoid intersections with wing surfaces. With this in place, their design variables remain consistent with those for wing generation, and could be merged with the fuselage using the existing toolset. In terms of optimisation implementation, a hierarchical design variable could define configuration (conventional, vertical only, twin, v-tail), with additional variables specifying radial angle as required.

Continuity

One of the more frustrating aspects of this work was the merging of wing and fuselage surfaces. With large design spaces unable to guarantee consistently intersecting parts for all combinations of wing, fuselage, and aerofoil parameters. Discarding designs that could not be immediately merged would introduce a large amount of noise within the optimisation, and therefore an iterative technique was employed to fix such designs, allowing performance analysis to be carried out. This method is by no means perfect, and will also introduce some noise to the overall optimisation, since design variables have to be altered in the process. Further work is required to streamline this amalgamation of parts, potentially leading to the implementation of single wing-body definition, removing the combination process altogether. This would allow design variables to be directly mapped onto output designs, while further aligning the framework with wider research, due to the increased interest in blended wing-body vehicles.

Continuity over component merging provides little benefit without continuous definitions of the individual components. In terms of wings, this could be achieved by smoothing known discontinuous regions, ie. partition joints, wing-tips and trailing edges. A generally improved definition would be the removal of individual partitions; instead defining the wing from span-wise polynomials, with aerofoil sections and chords defined at discrete positions. A distribution function could also be employed in terms of span-wise chord, providing further continuity across the wing.

Further control can be gained over the fuselage using a polynomial definition as well, which has been experimented with in the present work. Hermite fuselage definitions, which have been employed in previous works such as Dirkx and Mooij [24], produce piecewise continuous surfaces from a series of longitudinal control point sets. This allows for a broad set of shapes to be generated, with increasing number of control points in either longitudinal or lateral dimensions providing increased control as required. The implementation of this method requires robust constraints to enable both lifting-body and slender designs to be created in a single design space, but it importantly provides continuity over the full surface, in both dimensions, through the use control point derivatives. Hermite polynomials have been used for fore-body definitions in chapter 6, using nose and aft-body positions as control points to provide continuity. Full fuselage definitions were also tested prior to employed the class shape function approach in chapter 7. The intuitive design space definition of Hermite control points in this application is a cylindrical coordinate system with z axis aligned with body centreline, with control points sets to be placed at various z locations, varying in radial distance r and increasing azimuth angle θ. This allows simple constraints to be established, and scope to reduce the design space complexity by fixing consecutive longitudinal control points at a constant radial angle.

Finally, prioritising continuity among geometry definitions and their intersecting boundaries would also allow improved characterisation of surface flow through proper usage of integrated streamlines, as discussed in section 2.5. Works such as Kinney [36] present a good application of this, also demonstrating the intense work required to guarantee appropriate mesh generation for arbitrarily generated designs.

8.6.2 Analysis Methods

As discussed in chapter 5, employing low-fidelity analysis methods alone may lead to misleading results. This could be dealt with in terms of multi-fidelity optimisation in the 2D aerofoil case, due to their consistency in terms of chord, and geometric constraints that increased the likelihood of convergent CFD solutions. However, the scope of design space employed in the three dimensional problems of chapter 6 and 7, is far beyond the current automated meshing state-of-the-art. To limit the effects of relying solely on low-fidelity methods, uncertainty considerations and multiple flight states were employed. Nonetheless, extending to three dimensions from the work in chapter 5 should be experimented with. Relatively little research has been carried out employing multi-fidelity aerodynamic methods within population based optimisers, due to the aforementioned difficulties in automating high-fidelity analysis of large design spaces.

The methods used in this work serve as a stepping stone towards such problems, allowing only designs that pass well defined geometric tests to be analysed using high-fidelity methods. Combining these constraints with those relevant to three dimensional bodies should allow for a sim-

ilar implementation, although rigorous testing will be required to ensure accurate analysis can be obtained for feasible regions of the design space.

This work should be slowly grown out of the design variables used in chapter 5. For example, beginning with a constant planform shape that only varies in aerofoil cross-sectional shape, which should again be constant across the wing span. Once reliable automated meshing and analysis can be carried out for the constant planform case, experimentations can be made with varying aerofoil section from root to tip. As a final step, planform variables such as wing span, sweep, and chord can be implemented. Note that this discussion is limited to a single wing partition, as a significant amount of work will have to be conducted even for a small number of additional design variables. It is also important to remember that this work is concerned with global design optimisation, and therefore a gradient based approach may be preferred if only a limited design space is to be searched, or automated analysis proves overly challenging for a larger search space.

As discussed in chapter 6, an improved structural model is required to better capture possible wing-box formations, and their resulting weight distribution. Improvements to the current model would include additional spar panels within the box, and variable outer spar locations across the wing-span. This would be a good starting point, eventually moving away from the 1D beam model, using physical wing-box locations as structural nodes. Radial basis functions have already been implemented within this framework for interfacing such models with aerodynamic meshes, and FEM analysis could also be incorporated within a multi-fidelity context.

Employing such methods with arbitrary fuselage shapes, particularly in a blended wing-body context, also requires further analysis. This work restricted structural modelling to use within a conventional wing-body configuration only, with a considerable amount of work required to ensure accurate structures can be deployed in an arbitrary context. Furthermore, improved wing mass estimations are required with increasing structural fidelity, and should be validated against RLV and HST configurations if data is available. In the context of global vehicle optimisation, the inclusion wing-boxes within the design space will supersede the current HASA empirical method of wing weight.

8.6.3 Optimisation

Once again, a number of options are available for the development of optimisation methods within the current framework. As already mentioned, implementation of hierarchical design spaces would allow a variety of subspaces to make up the final design. These design spaces are much more suited to the genotype/binary nature of genetic algorithms, which are well document in research, and could be implemented within this work with relative ease.

Hybridised optimisers are another interesting tool, which have been previously used to combine GA and PSO algorithms [138], as well as gradient based methods following on from heuristic results. The latter has been experimented with in the current work, but ultimately showed little performance gains. Instead, gradient methods should be used to further develop Pareto front particles throughout optimisations, to speed up convergence and allow promising areas of the design space to be thoroughly explored. This hybridisation could also utilise multi-fidelity methods, enabling gradient techniques to explore more accurate cost function spaces, following a given particles confirmation of performance from high-fidelity analysis.

Generally, while the PSO optimisers developed and implemented in this work have performed well for the given problems, there is much room for improvement within this work and indeed the state-of-the-art. There are some misconceptions in research that highly constrained, large design space problems can be solved with similar population numbers as those used for benchmark test functions. This is not the case, and there is little substitute for increased sampling of the design space. However, PSO algorithms were developed for relatively small population numbers, making a single global best at each iteration an intuitive approach. The usefulness of this method drops with increasing population however, overly relying on the best observed particle. In single-objective methods, this is dealt with by weighting local bests much higher than the global in particle velocity computation. Nonetheless, the idea of a global best in this sense remains intuitive, since one particle will have a lower cost function than all others. Extending to multi-objective problems loses much of this intuition, with diverse Pareto fronts being reduced to a single particle. To increase diversity within this research, two global best particles were employed, focussing on both ideal trade-off and extremity Pareto front particles.

Going forward, more of the Pareto front should be considered within each iteration, employing a number of interconnected subswarms, as opposed to the star topology typically used in multi-objective problems. Moreover, some highly constrained, large design space benchmark problems should be a produced in an open-source context; allowing multiple institutions the opportunity to contribute towards the state-of-the-art, and enabling such methods to be benchmarked against traditional variants for a number of relevant problems. Efficient selection of best particles throughout iterations should allow for an alleviation in cost function calls to achieve the same convergence, thus reducing computational load, and allowing further analysis to be utilised in defining optimality and feasibility.

8.7 Additional Analysis Methods

To further develop the current framework in terms of low-fidelity analysis, further disciplines must be accounted for. Each additional form of analysis produces more realistic designs, since it incrementally captures more of the real-world problem. Even at the most intense levels of analysis in this work; multiple flight states each requiring a converged aerostructural solution, the time taken to complete analysis is generally less than one second per configuration. Thus, vast scope remains in terms of further analysis to determine optimality and feasibility.

8.7.1 Control

A starting point for further analysis modules would be the addition of control surfaces, whether it be ailerons on wing-body vehicles or a body flap on lifting-body configurations. The former has already been experimented with in the current work, using variables to define position and size in terms of normalised wing span. The inclusion of control surfaces allows a more typical trim routine to be carried out, and provides additional performance constraints such as control surface effectiveness. Moreover, the ability to deflect control surfaces allows trajectory analysis to be carried out, which would be a major addition to the current framework.

8.7.2 Trajectory & Propulsion

Obtaining performance data over a trajectory rather than static flight states would open up a number of options in terms of determining optimality, as well as providing mission profile dependent constraints. In the early phases of this work, a simple re-entry dynamics program was created, with the vision of analysing unpowered descent phases of configurations generated throughout optimisations. Unsurprisingly, this proved to be a bridge too far for the current research, but incorporation of trajectory analysis is absolutely required to further the developed framework.

Nonetheless, employing a rigid trajectory would have a major impact on resulting designs, which would invalidate much of the research up to this point, by unnecessarily and arbitrarily constraining the design space. Instead, trajectories would have to be optimised as well as vehicle components; determining designs initially, before performing a trajectory optimisation subproblem. Although this sounds relatively intense from a computational standpoint, a pre-determined set of flight states could be analysed and used as data points to be interpolated between within the trajectory subproblem. This pre-determined set would of course have to be bounded by minimum and maximum trajectory altitude, Mach number, angle of attack, and control surface deflection. However, the computational requirements of this 4D dataset could be easily quantified, remaining relatively consistent between designs, and the number of internal points could be manipulated depending on computational resources, to enable rapid trajectory optimisation

of arbitrary vehicle configurations.

This essentially creates a surrogate model of vehicle aerodynamic performance, with intermediate flight states determined through interpolation of the 4D dataset. Furthermore, in the interest of utilising computational resources efficiently, performance criteria could be used to determine whether each design should undergo trajectory optimisation, with poor performers compared to the previously best observed skipping such analysis. The ability to quantify this would be objective function dependent, however, assuming a multi-objective cost function space, those which are independent of trajectory, such as weight or overall aerodynamic efficiency, could be used to make such determinations.

Of course, some trajectories would require a knowledge of propulsion system employed. This could be as simple as assuming an additional weight and constant thrust output, to fully integrating engine components within designs, incorporating additional performance models, and analysing their aerodynamic properties, along with structural effects. Regardless of implementation, the chosen propulsive method would undoubtedly have a large impact on the design space. Limiting the variability of possible designs to allow for propulsion integration is of course justified, and alleviates some of the computational load in terms of optimiser convergence, allowing additional focus to be placed on performance characterisation.

Bibliography

[1] G. E. Dorrington, "Performance of air-breathing single-stage-to-orbit vehicles," *Acta Astronautica*, vol. 22, no. 1, pp. 17–41, 1990.

[2] NASA, "Access to Space Study," tech. rep., Office of Space Systems Development, NASA Headquarters, Washington, 1994.

[3] M. Bentley, *Spaceplanes*. New York: Springer, 2009.

[4] M. J. Bayer, "Comparative Assessment of Rocket-Propelled Single-Stage-to-Orbit Concepts," *Journal of Spacecraft and Rockets*, vol. 40, no. 2, pp. 273–278, 2003.

[5] R. I. Baumgartner, "Venturestar™ single stage to orbit reusable launch vehicle program overview," in *AIP Conference Proceedings 387*, pp. 1033–1040, 1997.

[6] D. C. Freeman, T. A. Talay, and R. E. Austin, "Reusable launch vehicle technology program," *Acta Astronautica*, vol. 41, no. 11, pp. 777–790, 1997.

[7] C. G. Miller, "Development of X-33/X-34 Aerothermodynamic Data Bases: Lessons Learned and Future Enhancements," tech. rep., NASA Langley Research Center, Hampton, VA, 1999.

[8] T. Stindt, J. Merrifield, M. Fossati, L. Ricciardi, C. A. Maddock, M. West, K. Kontis, B. Farkin, and S. Mcintyre, "Aerodynamic Database Development for a Future Reusable Space Launch Vehicle, the Orbital 500R," in *The International Conference on Flight vehicles, Aerothermodynamics and Re-entry Missions and Engineering*, (Monopoli, Italy), 2019.

[9] V. S. Reddy, "The SpaceX Effect," *New Space*, vol. 6, no. 2, pp. 125–134, 2018.

[10] C. Niederstrasser, "Small Launch Vehicles A 2018 State of the Industry Survey," tech. rep., Northrop Grumman Corporation, 2018.

[11] W. G. Huber, "Post-Saturn Launch Vehicle Study (Part II) Condensed Summary Report," tech. rep., NASA Future Projects Office, 1964.

[12] D. Preller and M. K. Smart, "Reusable Launch of Small Satellites Using Scramjets," *Journal of Spacecraft and Rockets*, vol. 54, no. 6, pp. 1317–1329, 2017.

[13] M. Landon, D. Hall, J. Udy, and E. Perry, "Automatic supersonic/hypersonic aerodynamic shape optimization," in *12th Applied Aerodynamics Conference*, Fluid Dynamics and Co-located Conferences, American Institute of Aeronautics and Astronautics, jun 1994.

[14] A. E. Gentry, D. N. Smyth, and W. R. Oliver, "The Mark IV Super-Hypersonic Arbitrary Body Program, Volume II Program Formulation," tech. rep., Douglas Aircraft Company, Long Beach, California, 1973.

[15] T. W. Sederberg and S. R. Parry, "Free-Form Deformation of Solid Geometric Models," *ACM SIGGRAPH Computer Graphics*, vol. 20, no. 4, pp. 151–160, 1986.

[16] K. Chiba, S. Obayashi, K. Nakahashi, A. P. Giotis, and K. C. Giannakoglou, "Design Optimization of the Wing Shape for the RLV Booster Stage Using Evolutionary Algorithms and Navier-Stokes Computations on Unstructured Grids," in *International Congress on Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems* (G. Bugeda, J. Désidéri, J. Periaux, M. Schoenauer, and G. Winter, eds.), (Barcelona), EUROGEN, 2003.

[17] T. Tatsukawa, T. Nonomura, A. Oyama, and K. Fujii, "Aerodynamic Design Exploration for Reusable Launch Vehicle Using Genetic Algorithm with Navier Stokes Solver," *Transactions of the Japan Society for Aeronautical and Space Sciences, Aerospace Technology Japan*, vol. 10, no. 28, pp. 55–63, 2012.

[18] Q. Zhang, K. Ye, Z.-Y. Ye, and W.-W. Zhang, "Aerodynamic Optimization for Hypersonic Wing Design Based on Local Piston Theory," *Journal of Aircraft*, vol. 53, no. 4, pp. 1065 – 1072, 2016.

[19] Y. Shen, W. Huang, T. Zhang, and L. Yan, "Parametric modeling and aerodynamic optimization of EXPERT configuration at hypersonic speeds," *Aerospace Science and Technology*, vol. 84, no. 1, pp. 641–649, 2019.

[20] F. Massobrio, R. Viotto, M. Serpico, A. Sansone, M. Caporicci, and J.-M. Muylaert, "EXPERT: An atmospheric re-entry test-bed," *Acta Astronautica*, vol. 60, no. 12, pp. 974–985, 2007.

[21] S. Sheffer and G. Dulikravich, "Constrained optimization of three-dimensional hypersonic vehicle configurations," in *31st Aerospace Sciences Meeting*, Aerospace Sciences Meetings, American Institute of Aeronautics and Astronautics, jan 1992.

[22] K. Bowcutt, "Hypersonic aircraft optimization including aerodynamic, propulsion, and trim effects," in *4th Symposium on Multidisciplinary Analysis and Optimization*, American Institute of Aeronautics and Astronautics, sep 1992.

[23] K. G. Bowcutt, "Multidisciplinary Optimization of Airbreathing Hypersonic Vehicles," *Journal of Propulsion and Power*, vol. 17, no. 6, pp. 1184–1190, 2001.

[24] D. Dirkx and E. Mooij, "Optimization of entry-vehicle shapes during conceptual design," *Acta Astronautica*, vol. 94, no. 1, pp. 198–214, 2014.

[25] R. Wuilbercq, F. Pescetelli, A. Mogavero, E. Minisci, and R. E. Brown, "Robust multidisciplinary design and optimisation of a reusable launch vehicle," in *19th AIAA International Space Planes and Hypersonic Systems and Technologies Conference*, 2014.

[26] F. Deng, Z.-H. Jiao, J. Chen, D. Zhang, and S. Tang, "Overall Performance Analysis–Oriented Aerodynamic Configuration Optimization Design for Hypersonic Vehicles," *Journal of Spacecraft and Rockets*, vol. 54, no. 5, pp. 1015–1026, 2017.

[27] S. Di Giorgio, D. Quagliarella, G. Pezzella, and S. Pirozzoli, "An aerothermodynamic design optimization framework for hypersonic vehicles," *Aerospace Science and Technology*, vol. 84, no. 1, pp. 339–347, 2019.

[28] L. Lees, "Hypersonic Aerodynamics," in *Fifth Internation Aeronautical Conference*, (Los Angeles), pp. 241–276, Inst. of Aeronautical Sciences, New York, 1955.

[29] J. Anderson, *Hypersonic and High-Temperature Gas Dynamics*. American Institute of Aeronautics and Astronautics, 2nd ed., 2006.

[30] M. L. Rasmussen, "On Hypersonic Flow Past an Unyawed Cone," *AIAA Journal*, vol. 5, no. 8, pp. 1495–1497, 1967.

[31] E. Eckert, "Engineering relations for heat transfer and friction in high-velocity laminar and turbulent boundary-layer flow over surfaces with constant pressure and temperature," *Transactions of the ASME*, vol. 78, no. 6, pp. 1273–1283, 1956.

[32] W. E. Meador and M. K. Smart, "Reference Enthalpy Method Developed from Solutions of the Boundary-Layer Equations," *AIAA Journal*, vol. 43, no. 1, 2005.

[33] V. Dicristina, "Three-dimensional laminar boundary-layer transition on a sharp 8 degree cone at Mach 10," *AIAA Journal*, vol. 8, no. 5, pp. 852–856, 1970.

[34] E. R. van Driest, "The Problem of Aerodynamic Heating," *Aeronautical Engineering Review*, pp. 26–41, 1956.

[35] T. Jazra and M. Smart, "Development of an Aerodynamics Code for the Optimisation of Hypersonic Vehicles," in *47th AIAA Aerospace Sciences Meeting*, (Orlando, Florida), 2009.

[36] D. J. Kinney, "Aero-Thermodynamics for Conceptual Design," in *42nd AIAA Aerospace Sciences Meeting and Exhibit*, (Reno, Nevada), 2004.

[37] M. A. Lobbia, "Optimization of Waverider-Derived Crew Reentry Vehicles Using a Rapid Aerodynamics Analysis Approach," in *53rd AIAA Aerospace Sciences Meeting*, (Kissimmee, Florida), 2015.

[38] E. J. Landrum and C. D. Babb, "Wind-Tunnel Force and Flow Visualization Data at Mach Numbers From 1.6 to 4.63 for a series of Bodies of Revolution at Angles of Attack From -4 ° to 60 °," tech. rep., NASA Langley Research Center, Hampton, Virginia, 1979.

[39] D. M. Giles and D. D. Marshall, "Aerodynamic performance enhancement of a NACA 66-206 airfoil using supersonic channel airfoil design," in *46th AIAA Aerospace Sciences Meeting and Exhibit*, (Reno, Nevada), 2008.

[40] K. Higgins, "Comparison of Engineering Correlations for Predicting Heat Transfer in Zero-pressure-gradient Compressible Boundary Layers with CFD and Experimental Data," tech. rep., DSTO Defence Science and Technology Organisation, Victoria, Australia, 2008.

[41] J. J. McNamara, P. P. Friedmann, K. G. Powell, and B. J. Thuruthimattam, "Aeroelastic and Aerothermoelastic Behavior in Hypersonic Flow," *AIAA Journal*, vol. 46, no. 10, pp. 2591–2610, 2008.

[42] J. McNamara, *Aeroelastic and aerothermoelastic behavior of two and three dimensional lifting surfaces in hypersonic flow*. Phd thesis, University of Michigan, 2005.

[43] Z. Wan, Y. Wang, Y. Liu, and C. Yang, "A high-efficiency aerothermoelastic analysis method," *Science China: Physics, Mechanics and Astronomy*, vol. 57, no. 6, pp. 1111–1118, 2014.

[44] G. J. Brauckmann, "X-34 Vehicle Aerodynamic Characteristics," *Journal of Spacecraft and Rockets*, vol. 36, no. 2, pp. 229–239, 1999.

[45] H. D. Fuhrmann, J. Hildebrand, T. Lalicata, and O. S. Corporation, "Aerothermodynamic Overview, X-34," *Journal of Spacecraft and Rockets*, vol. 36, no. 2, pp. 153–159, 1999.

[46] R. Sullivan and B. Winters, "X-34 program overview," in *34th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, 1998.

[47] A. Tewari, *Atmospheric and Space Flight Dynamics: Modeling and Simulation with MATLAB® and Simulink®*. Birkhäuser Basel, 2007.

[48] J. S. Arora, *Introduction to Optimum Design*. Elsevier, third ed., 2012.

[49] E. Polak and G. Ribiere, "Note sur la convergence de méthodes de directions conjuguées," *Revue française d'informatique et de recherche opérationnelle, sé- rie rouge*, vol. 3, no. 1, pp. 35–43, 1969.

[50] P. E. Gill, W. Murray, and M. H. Wright, "Practical Optimization," tech. rep., Systems Optimization Laboratory, Stanford University, 1981.

[51] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer, second ed., 2006.

[52] D. A. Van Veldhuizen and G. B. Lamont, "Evolutionary Computation and Convergence to a Pareto Front," in *Genetic Programming 1998 Conference*, pp. 221–228, 1998.

[53] M. Kumar, M. Husian, N. Upreti, and D. Gupta, "Genetic Algorithm: Review and Application," *International Journal of Information Technology and Knowledge Management*, vol. 2, no. 2, pp. 451–454, 2010.

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

[55] J. H. Holland, *Adaptation in Natural and Artificial Systems*. MIT Press, 1975.

[56] R. Hassan, B. Cohanim, O. de Weck, and G. Venter, "A comparison of particle swarm optimization and the genetic algorithm," in *1st AIAA multidisciplinary design optimization specialist conference*, (Austin, TX), American Institute of Aeronautics and Astronautics, 2005.

[57] C. A. Coello Coello, G. T. Pulido, and M. S. Lechuga, "Handling Multiple Objectives With Particle Swarm Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 256–279, 2004.

[58] P. C. Fourie and A. A. Groenwold, "The particle swarm optimization algorithm in size and shape optimization," *Structural and Multidisciplinary Optimization*, vol. 23, no. 4, pp. 259–267, 2002.

[59] J. J. Durillo, J. García-Nieto, A. J. Nebro, C. A. Coello Coello, F. Luna, and E. Alba, "Multi-Objective Particle Swarm Optimizers: An Experimental Comparison," in *International Conference on Evolutionary Multi-Criterion Optimization*, pp. 495–509, 2009.

[60] M. R. Sierra and C. A. Coello Coello, "Improving PSO-based Multi-Objective Optimization using Crowding, Mutation and e-Dominance," in *International Conference on Evolutionary Multi-Criterion Optimization*, pp. 505–519, 2005.

[61] S. Zapotecas Martínez and C. A. Coello Coello, "A multi-objective particle swarm optimizer based on decomposition," in *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pp. 69–76, 2011.

[62] R. Eberhart and J. Kennedy, "A New Optimizer Using Particle Swarm Theory," in *Sixth International Symposium on Micro Machine and Human Science*, pp. 39–43, 1995.

[63] Y. Shi and R. Eberhart, "A Modified Particle Swarm Optimizer," in *IEEE International Conference on Evolutionary Computation*, pp. 69–73, 1998.

[64] M. Richards and D. Ventura, "Choosing a starting configuration for particle swarm optimization," *IEEE International Conference on Neural Networks*, vol. 3, pp. 2309–2312, 2004.

[65] L. Ju, Q. Du, and M. Gunzburger, "Probabilistic methods for centroidal Voronoi tessellations and their parallel implementations," *Parallel Computing*, vol. 28, no. 1, pp. 1477–1500, 2002.

[66] M. D. McKay, R. J. Beckman, and W. J. Conover, "Comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, vol. 21, no. 2, pp. 239–245, 1979.

[67] J. Kennedy and R. Mendes, "Population structure and particle swarm performance," in *Proceedings of the 2002 Congress on Evolutionary Computation, CEC 2002*, pp. 1671–1676, 2002.

[68] A. E. Muñoz Zavala, A. Hernández Aguirre, and E. R. Villa Diharce, "The singly-linked ring topology for the particle swarm optimization algorithm," in *Proceedings of the 11th Annual Genetic and Evolutionary Computation Conference, GECCO-2009*, pp. 65–71, 2009.

[69] Q. Liu, W. Wei, H. Yuan, Z.-H. Zhan, and Y. Li, "Topology selection for particle swarm optimization," *Information Sciences*, vol. 363, no. 1, pp. 154–173, 2016.

[70] J. J. Liang and P. N. Suganthan, "Dynamic multi-swarm particle swarm optimizer," in *IEEE Swarm Intelligence Symposium, SIS 2005*, pp. 127–132, IEEE, 2005.

[71] A. S. Mohais, R. Mendes, C. Ward, and C. Posthoff, "Neighborhood Re-structuring in Particle Swarm Optimization," in *Zhang S., Jarvis R. (eds) AI 2005: Advances in Artificial*

Intelligence. AI 2005. Lecture Notes in Computer Science, pp. 776–785, Springer, Berlin, Heidelberg, 2005.

[72] A. S. Mohais Christopher Ward Christian Posthoff and S. Augustine, "Randomized Directed Neighborhoods with Edge in Particle Swarm Optimization," in *IEEE Congress on Evolutionary Computation*, pp. 548–555, 2004.

[73] X. Zhao, X. S. Gao, and Z. C. Hu, "Evolutionary programming based on non-uniform mutation," *Applied Mathematics and Computation*, vol. 192, no. 1, pp. 1–11, 2007.

[74] T. Bäck, *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, New York, 1996.

[75] A. Ben-Tal, "Characterization of Pareto and Lexicographic Optimal Solutions," *Fandel G., Gal T. (eds) Multiple Criteria Decision Making Theory and Application. Lecture Notes in Economics and Mathematical Systems*, vol. 177, 1980.

[76] C. R. Raquel and P. C. Naval, "An Effective Use of Crowding Distance in Multiobjective Particle Swarm Optimization," in *GECCO '05: Proceedings of the 7th annual conference on Genetic and evolutionary computation*, pp. 257–264, 2005.

[77] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[78] X. Li, "A Non-dominated Sorting Particle Swarm Optimizer for Multiobjective Optimization," in *Cantú-Paz E. et al. (eds) Genetic and Evolutionary Computation — GECCO 2003. Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, 2003.

[79] K. Amadori, C. Jouannet, and P. Krus, "Use of Panel Code Modeling in a Framework for Aircraft Concept Optimization," in *11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, (Portsmouth, Virginia), pp. 1–13, 2006.

[80] H. P. Buckley, B. Y. Zhou, and D. W. Zingg, "Airfoil Optimization Using Practical Aerodynamic Design Requirements," *Journal of Aircraft*, vol. 47, no. 5, pp. 1707–1719, 2010.

[81] T. P. Runarsson and X. Yao, "Stochastic ranking for constrained evolutionary optimization," *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 3, pp. 284–294, 2000.

[82] T. Takahama and S. Sakai, "Constrained Optimization by ε Constrained Particle Swarm Optimizer with ε-level Control," in *Soft Computing as Transdisciplinary Science and Technology. Advances in Soft Computing, vol 29* (A. Abraham, Y. Dote, T. Furuhashi, M. Köppen, A. Ohuchi, and Y. Ohsawa, eds.), Berlin, Heidelberg: Springer, 2005.

[83] The Mathworks, "Optimization Toolbox™ User's Guide," 2020.

[84] H. H. Rosenbrock, "An Automatic Method for finding the Greatest or Least Value of a Function," *The Computer Journal*, vol. 3, no. 3, pp. 175–184, 1960.

[85] L. A. Rastrigin, "Systems of external control," tech. rep., Mir, Moscow, 1974.

[86] E. M. L. Beale, "Nonlinear programming using a general mathematical programming system," *Design and implementation of optimization software*, pp. 259–279, 1978.

[87] S. Mishra, "Some new test functions for global optimization and performance of repulsive particle swarm method," *MPRA Paper No. 2718*, 2006.

[88] A. Townsend, "Constrained optimization in Chebfun," 2014.

[89] P.-A. Simionescu and D. G. Beale, "New Concepts in Graphic Visualization of Objective Functions," in *Proceedings of DETC'02 ASME 2002 Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, (Montreal, Canada), 2002.

[90] C. M. Fonseca and P. J. Fleming, "An Overview of Evolutionary Algorithms in Multiobjective Optimization," *Evolutionary Computation*, vol. 3, no. 1, pp. 1–16, 1995.

[91] T. Binh, "MOBES: A multiobjective evolution strategy for constrained optimization problems," tech. rep., 1997.

[92] K. Deb and D. Kalyanmoy, *Multi-Objective Optimization Using Evolutionary Algorithms*. New York: John Wiley & Sons, Inc., 2001.

[93] A. Osyczka and S. Kundu, "A new method to solve generalized multicriteria optimization problems using the simple genetic algorithm," *Structural optimization*, vol. 10, no. 2, pp. 94–99, 1995.

[94] R. M. Hicks, E. M. Murman, and G. N. Vanderplaats, "An Assessment of Airfoil Design by Numerical Optimization," tech. rep., NASA, 1974.

[95] M. Drela, "Pros and Cons of Airfoil Optimization," in *Frontiers of Computational Fluid Dynamics*, pp. 1–19, World Scientific, 1998.

[96] S. L. Padula and W. Li, "Options For Robust Airfoil Optimization Under Uncertainty," in *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, (Atlanta, Georgia), 2002.

[97] Y. Liang, X. Q. Cheng, Z. N. Li, and J. W. Xiang, "Multi-objective robust airfoil optimization based on non-uniform rational B-spline(NURBS) representation," *Science China Technological Sciences*, vol. 53, no. 10, pp. 2708–2717, 2010.

[98] K. Shimoyama, A. Oyama, and K. Fujii, "Development of Multi-Objective Six Sigma Approach for Robust Design Optimization," *Journal of Aerospace Computing, Information, and Communication*, vol. 5, no. 8, pp. 215–233, 2008.

[99] A. Arias-Montaño, C. A. Coello, and E. Mezura-Montes, "Multiobjective evolutionary algorithms in aeronautical and aerospace engineering," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 5, pp. 662–694, 2012.

[100] S. Z. Martínez, A. Arias-Montaño, and C. A. Coello Coello, "Constrained multi-objective aerodynamic shape optimization via swarm intelligence," *GECCO 2014 - Proceedings of the 2014 Genetic and Evolutionary Computation Conference*, pp. 81–88, 2014.

[101] Y. Bu, W. Song, Z. Han, and J. Xu, "Aerodynamic optimization design of airfoil based on CST parameterization method," *Xibei Gongye Daxue Xuebao/Journal of Northwestern Polytechnical University*, vol. 31, no. 5, pp. 829–836, 2013.

[102] R. Duvigneau, "Adaptive Parameterization using Free-Form Deformation for Aerodynamic Shape Optimization," tech. rep., 2006.

[103] D. Masters, T. Rendall, C. B. Allen, and D. J. Poole, "A Geometric Comparison of Aerofoil Shape Parameterisation Methods Aerodynamic Optimization View project SABRE Shape Adaptive Blades for Rotorcraft Efficiency View project," in *54th AIAA Aerospace Sciences Meeting*, (San Diego, California), 2016.

[104] B. M. Kulfan and J. E. Bussoletti, ""Fundamental" Parametric Geometry Representations for Aircraft Component Shapes," in *11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, (Portsmouth, Virginia), 2006.

[105] R. W. Derksen and &. T. Rogalsky, "Optimum aerofoil parameterization for aerodynamic design," *Computer Aided Optimum Design in Engineering*, vol. XI, no. 1, pp. 197–206, 2009.

[106] H. Sobieczky, "Parametric Airfoils and Wings," *Fujii K., Dulikravich G.S. (eds) Recent Development of Aerodynamic Design Methodologies. Notes on Numerical Fluid Mechanics (NNFM)*, vol. 65, no. 1, pp. 71–87, 1999.

[107] UIUC Applied Aerodynamics Group, "UIUC Airfoil Database, https://m-selig.ae.illinois.edu/ads/coord_database.html," 2021.

[108] B. Naujoks, L. Willmes, W. Haase, T. Bäck, and M. Schütz, "Multi-point Airfoil Optimization Using Evolution Strategies," in *European Congress on Computational Methods in Applied Sciences and Engineering ECCOMAS*, (Barcelona), 2000.

[109] X. He, J. Li, C. A. Mader, A. Yildirim, and J. R. R. A. Martins, "Robust aerodynamic shape optimization-from a circle to an airfoil," *Aerospace Science and Technology*, vol. 87, no. 1, pp. 48–61, 2019.

[110] F. Palacios, M. R. Colonno, A. C. Aranake, A. Campos, S. R. Copeland, T. D. Economon, A. K. Lonkar, T. W. Lukaczyk, T. W. Taylor, and J. J. Alonso, "Stanford University Unstructured (SU2): An open-source integrated computational environment for multi-physics simulation and design," *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition 2013*, 2013.

[111] C. Geuzaine and J.-F. Remacle, "Gmsh: a three-dimensional finite element mesh generator with built-in pre-and post-processing facilities," *International Journal for Numerical Methods in Engineering*, vol. 79, no. 11, pp. 1309–1331, 2009.

[112] J.-F. Remacle, J. Lambrechts, B. Seny, E. Marchandise, A. Johnen, and C. Geuzaine, "Blossom-Quad: a non-uniform quadrilateral mesh generator using a minimum cost perfect matching algorithm," *International Journal for Numerical Methods in Engineering*, vol. 89, no. 9, pp. 1102–1119, 2010.

[113] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Massachusetts Institute of Technology: MIT Press, 2006.

[114] The Mathworks, "Statistics and Machine Learning Toolbox™ User's Guide," 2021.

[115] M. S. Eldred and D. M. Dunlavy, "Formulations for Surrogate-Based Optimization with Data Fit, Multifidelity, and Reduced-Order Models," in *11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, (Portsmouth, Virginia), pp. 2450–2469, 2006.

[116] D. Rajnarayan, A. Haas, and I. Kroo, "A Multifidelity Gradient-Free Optimization Method and Application to Aerodynamic Design," in *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, (Victoria, British Columbia Canada), 2008.

[117] G. Bindolino, G. Ghiringhelli, S. Ricci, and M. Terraneo, "Multilevel Structural Optimization for Preliminary Wing-Box Weight Estimation," *Journal of Aircraft*, vol. 47, no. 2, pp. 475–489, 2010.

[118] T. R. Brooks, G. K. W. Kenway, and J. R. R. A. Martins, "Benchmark Aerostructural Models for the Study of Transonic Aircraft Wings," *AIAA Journal*, vol. 56, no. 7, pp. 2840–2855, 2018.

[119] A. Elham and M. J. L. van Tooren, "Tool for preliminary structural sizing, weight estimation, and aeroelastic optimization of lifting surfaces," *Journal of Aerospace Engineering*, vol. 230, no. 2, pp. 280–295, 2016.

[120] S. Chauhan and J. R. R. A. Martins, "Low-fidelity aerostructural optimization of aircraft wings with a simplified wingbox model using OpenAeroStruct," in *EngOpt 2018 Proceedings of the 6th International Conference on Engineering Optimization*, (Lisboa), pp. 418–431, Springer, 2018.

[121] A. Elham, G. La Rocca, and M. J. L. van Tooren, "Development and implementation of an advanced, design-sensitive method for wing weight estimation," *Aerospace Science and Technology*, vol. 29, no. 1, pp. 100 – 113, 2013.

[122] "Military Handbook, Metallic Materials and Elements for Aerospace Vehicle Structures, MIL-HDBK-5G," tech. rep., 1994.

[123] M. C. Y. Niu, *Airframe Stress Analysis and Sizing*. Hong Kong: Commit Press Ltd, 2nd ed., 1999.

[124] B. M. Irons and R. C. Tuck, "A Version of the Aitken Accelerator for Computer Iteration," *International Journal of Numerical Methods in Engineering*, vol. 1, no. 3, pp. 275–277, 1969.

[125] U. Küttler and W. A. Wall, "Fixed-point fluid-structure interaction solvers with dynamic relaxation," *Computational Mechanics*, vol. 43, no. 1, pp. 61–72, 2008.

[126] G. J. Kennedy and J. R. R. A. Martins, "Parallel Solution Methods for Aerostructural Analysis and Design Optimization," in *13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference*, (Fort Worth), 2010.

[127] J. P. Jasa, J. T. Hwang, and J. R. R. A. Martins, "Open-source coupled aerostructural optimization using Python," *Structural and Multidisciplinary Optimization*, vol. 57, no. 1, pp. 1815–1827, 2018.

[128] S. Brown, "Displacement extrapolations for CFD+CSM aeroelastic analysis," in *38th Structures, Structural Dynamics, and Materials Conference*, (Reno, Nevada), AIAA, 1997.

[129] G. P. Guruswamy, "A review of numerical fluids/structures interface methods for computations using high-fidelity equations," *Computers and Structures*, vol. 80, pp. 31–41, 2002.

[130] V. Murti and S. Valliappan, "Numerical inverse isoparametric mapping in remeshing and nodal quantity contouring," *Computers & Structures*, vol. 22, no. 6, pp. 1011–1021, 1986.

[131] C. Byun and G. P. Guruswamy, "Wing-Body Aeroelasticity on Parallel Computers," *Journal of Aircraft*, vol. 33, no. 2, pp. 421–428, 1996.

[132] J. A. Samareh, "Discrete Data Transfer Technique for Fluid-Structure Interaction," in *18th AIAA Computational Fluid Dynamics Conference*, (Miami, Florida), 2007.

[133] J. R. R. A. Martins, J. J. Alonso, and J. J. Reuther, "A Coupled-Adjoint Sensitivity Analysis Method for High-Fidelity Aero-Structural Design," *Optimization and Engineering*, vol. 6, no. 1, pp. 33–62, 2005.

[134] D. S. Broomhead and D. Lowe, "Radial basis functions, multi-variable functional interpolation and adaptive networks," tech. rep., Royal Signals and Radar Establishment Malvern (United Kingdom), 1988.

[135] A. Beckert and H. Wendland, "Multivariate interpolation for fluid-structure-interaction problems using radial basis functions," *Aerospace Science and Technology*, vol. 5, no. 2, pp. 125–134, 2001.

[136] T. C. S. Rendall and C. B. Allen, "Unified fluid-structure interpolation and mesh motion using radial basis functions," *International Journal for Numerical Methods in Engineering*, vol. 74, no. 1, pp. 1519–1559, 2008.

[137] G. J. Harloff and B. M. Berkowitz, "HASA-Hypersonic Aerospace Sizing Analysis for the Preliminary Design of Aerospace Vehicles," tech. rep., NASA, 1988.

[138] T. Takahama, S. Sakai, and N. Iwane, "Constrained Optimization by the ε Constrained Hybrid Algorithm of Particle Swarm Optimization and Genetic Algorithm," in *Zhang S., Jarvis R. (eds) AI 2005: Advances in Artificial Intelligence. AI 2005. Lecture Notes in Computer Science, vol 3809*, pp. 389–400, Springer, Berlin, Heidelberg, 2005.