A Framework for Medium-Fidelity Ducted Fan Design Optimisation

Application of a Throughflow Solver in a Genetic Algorithm for Fast Optimisation

Thomas Stephan Vermeulen

Delft University of Technology

TU Delft

This page is left blank intentionally.

A Framework for Medium-Fidelity Ducted Fan Design Optimisation

Application of a Throughflow Solver in a Genetic Algorithm for Fast Optimisation

by

Thomas Stephan Vermeulen

in partial fulfilment of the requirements for the degree of

Master of Science

in Aerospace Engineering

at the Delft University of Technology,

to be defended publicly on Thursday 4th September, 2025 at 10:00.

Project duration:	October 2024 – August 2025	
Thesis committee:	Dr. ir. W.P.J. Visser,	TU Delft, Main Supervisor
	Dr. ir. T. Sinnige,	TU Delft, Supervisor
	Dr. N.J. Wood,	GKN Aerospace Services Limited, Supervisor
	Dr. ir. G. la Rocca	TU Delft, Chairman
	Dr. A. Sciacchitano	TU Delft, Examiner

Cover:	Artistic rendering of the GKN Aerospace H2GEAR electric ducted fan aircraft concepts in flight [1]. Used with permission.
Copyright:	August 2025, T.S. Vermeulen Portions of the software incorporated herein is MTFLOW Software, © MIT 1997, used with permission. All Rights Reserved.
Word count:	29,800

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft

Preface

Time flies by when pursuing your passions. It feels like it was yesterday when I first walked through the doors of the faculty of Aerospace Engineering, starting my Bachelor of Science. Throughout the past six years, I have learned an incredible amount on a wide range of topics. From conceptual aircraft design, to multidisciplinary optimisation, to fuel cells, my time in Delft has been diverse, exciting, and challenging, yet rewarding.

Already during my Bachelor's, I discovered my passion for research, which I have pursued throughout both degrees. This culminated in this thesis project on ducted fans, a topic that, coincidentally, I had already begun working on in my BSc thesis, where we utilised distributed electric ducted fans in the conceptual design of a hydrogen fuel-cell-powered blended-wing-body aircraft. Unrelated, the idea for this particular Master's thesis research originated from a series of brainstorming sessions with Norman Wood during my internship at GKN Aerospace Services Limited in Bristol, UK, as part of the Hydrogen Systems team, which slowly matured into the exciting research performed in this thesis with Wilfried Visser and Tomas Sinnige. Having had the opportunity to not only research technology to help aviation become more sustainable, but also knowing that my research has a real-world application, has been incredibly rewarding.

This thesis would not have been possible without the constant support and encouragement of my supervisors, Wilfried Visser, Tomas Sinnige, and Norman Wood. The thesis meetings were a highlight of my week, as we discussed the interesting aspects of the research and generated new ideas and conclusions based on the results. Thank you for all the great suggestions, feedback, and for helping make this thesis so enjoyable. Additionally, I would like to thank Gianfranco la Rocca and Andrea Sciacchitano for taking part in my graduation committee and taking the time to assess my work. I hope you found it as interesting as I did.

I would also like to explicitly thank Tomas Sinnige and Maurice Hoogreef for their help in getting access to high-performance computing on one of the departmental servers. Without this access, the scope of the analyses performed in this thesis would not have been possible. A further thank you is due to Mark Drela at MIT for providing access to the MTFLOW software used in this thesis.

I want to thank all my friends and family for their support, motivation, and suggestions throughout my thesis. Last, but most definitely not least, my parents deserve my deepest thanks. For as long as I can remember, they have been my greatest supporters and have encouraged me to reach for the stars. I admit that I do not always immediately heed their advice, but it is appreciated nonetheless. Thank you, Dad, for the constant "Is your analysis working yet?" and for proofreading my work whenever I asked you to. A special thank you to my mother for arguably dealing with more stress from my thesis than I ever did.

I am proud to have contributed, even if only a small part, to helping aviation become more sustainable. I hope that readers of this thesis find it interesting, informative, and most of all useful and inspiring.

Thomas Stephan Vermeulen
Alphen aan den Rijn, the Netherlands
August 2025

Abstract

To help make aviation sustainable, novel aircraft propulsion concepts, such as electric ducted fans, are being researched. Ducted fans have been successfully applied in experimental aircraft, such as the Bell X-22A. However, modern research focuses on small-scale, low-speed applications, rather than on transport aircraft applications. This is often achieved using either lower-order, inviscid, incompressible methods, which limit their applicability to low Mach numbers, or computationally expensive CFD methods.

A gap, therefore, exists in medium-fidelity methods applicable to transport aircraft that combine high accuracy with low computational cost. Such methods would enable new, computationally efficient design optimisations and could be integrated into larger aircraft design workflows to help design more sustainable aircraft. Hence, this thesis was performed to identify and improve a medium-fidelity method and apply it to design optimisations of the complete ducted fan.

The existing Multi-passage ThroughFLOW streamline curvature throughflow solver is integrated into a newly developed unified ducted fan code. Validation of the newly integrated code against experimental wind tunnel data for the X-22A propulsor shows an excellent match in thrust and power coefficients. Comparison against the lower-fidelity Ducted Fan Design Code shows that the throughflow method is over $3.5\times$ more accurate in predicting the thrust coefficient. The power coefficient cannot be accurately modelled using the lower-fidelity method, further motivating the choice of using a throughflow method, which is more than $3\times$ as accurate in predicting the power coefficient for the worst-case advance ratio considered.

The unified code was implemented into an optimisation framework using the Unified Non-dominated Sorting Genetic Algorithm III. Optimisations were performed on the X-22A propulsor to demonstrate its applicability, where propulsor efficiency, frontal area, and total flight energy were used as objectives. Take-off, endurance, and combat cruise conditions were analysed individually, and an endurance flight profile was defined for a multi-point analysis. Results show the framework can effectively optimise the duct profile and blade camber distributions. However, the blade thickness distributions require further refinement. This is caused by the lack of blade row loss modelling.

The application of frontal area as a secondary objective to the single-point optimisation for the endurance cruise condition reveals that multi-objective optimisation offers greater flexibility in selecting a suitable design within the design and objective spaces. Both the single-objective and multi-objective optimisations converge to significantly improved designs compared to the reference design. A maximum efficiency improvement of $\Delta\eta_p = 7.09\%$ was observed for the endurance cruise condition, corresponding to a 10% reduction in frontal area.

The single-point optimised designs performed poorly at other conditions within the considered operational envelope. A multi-point energy optimisation was therefore defined for a simplified endurance mission profile of the X-22A. A reduction in total required energy of $\Delta E = 2\%$ was achieved, where negligible changes in the optimised design were observed, attributed to the algorithm's difficulty in satisfying both the take-off and endurance thrust level requirements.

The developed ducted fan code is significantly more accurate than the lower-fidelity methods currently in use, while requiring only $10 - 45$ s per analysed design. This efficiency, when integrated into the optimisation framework, makes it a suitable method for the conceptual design optimisation for future (electric) aircraft concepts. The robustness and computational efficiency of the framework also support broader applications, such as larger, multidisciplinary optimisation problems or the integration into a complete aircraft architecture optimisation framework.

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

Abbreviation	Definition
CFD	Computational Fluid Dynamics
CPU	Central Processing Unit
DFDC	Ducted Fan Design Code
GA	Genetic Algorithm
IGD	Inverted Generational Distance
MOO	Multi-Objective Optimisation
MTFLOW	Multi-passage ThroughFLOW
OOP	Object-Oriented Programming
PM	Polynomial Crossover
SBX	Simulated Binary Crossover
SLSQP	Sequential Least Squares Quadratic Programming
SOO	Single-Objective Optimisation
UDC	Unified Ducted fan Code
UNSGA-III	Unified Non-dominated Sorting Genetic Algorithm III

Symbols

Symbol	Definition	Unit
A	Area	[m^2]
a	Speed of sound	[$\mathrm{m\,s^{-1}}$]
B	Blade count	[-]
b	Bézier variable	[-]
C	Coefficient	[-]
CV	Constraint violation	[-]
c	Chord length	[m]
dz_{TE}	Normalised half thickness of the trailing edge	[-]
E	Energy	[J]
e	Partition count	[-]
f	Objective function	[-]
g	Inequality constraint	[-]
H	Mask	[-]
h	Enthalpy	[$\mathrm{J\,kg^{-1}}$]
J	Advance ratio	[-]
K	Population size	[-]
k	Rate constant	[$\mathrm{m^{-1}}$]
L	Length	[m]
ℓ	Lower bound of the design vector	[-]
M	Mach number	[-]
m	Meridional coordinate	[m]
$\dot{m}$	Mass flow	[$\mathrm{kg\,s^{-1}}$]
m'	Stretched meridional coordinate	[-]
n_{crit}	Critical amplification factor	[-]
P	Power	[W]

Symbol	Definition	Unit
p	Pressure	[Pa]
R	Blade radius	[m]
Re	Reynolds number	[-]
r	Local radius	[m]
r_{LE}	Leading edge radius	[-]
S_{rel}	Geometric blade slope	[-]
s	Distance along a meridional streamline	[m]
T	Thrust	[N]
T_θ	Circumferential blade thickness	[m]
t	Time	[s]
u	Upper bound of the design vector	[-]
v	Velocity	$[\mathrm{m\,s^{-1}}]$
W	Weight	[N]
x	Axial/chordwise coordinate	[m]
y	Thickness/depth-wise coordinate	[m]
z	Radial/height-wise coordinate	[m]
z_{TE}	Normalised trailing edge vertical displacement	[-]
α	Angle of attack	[rad]
α_{TE}	Trailing camberline angle	[rad]
β	Blade angle	[rad]
β_{TE}	Trailing wedge angle	[rad]
$\bar{\Gamma}$	Swirl	$[\mathrm{m^2\,s^{-1}}]$
γ_{LE}	Leading edge direction	[rad]
δ	Discrete uniform noise function	[-]
δ^*	Displacement thickness	[m]
ϵ	Continuous uniform noise function	[-]
ε	Scaled coefficient deviation	[-]
ζ	Camber slope angle	[rad]
η_p	Propulsor efficiency	[-]
θ	Blade-to-blade circumferential coordinate	[rad]
ι	Coordinate system transformation angle	[rad]
κ	Ratio of specific heats	[-]
Λ	Expansion ratio	[-]
μ	Dynamic viscosity	[Pa s]
ν	Fraction of the continuous non-zero reference values	[-]
ξ	Design vector	[-]
Π	Streamline face pressure	[Pa]
ρ	Density	$[\mathrm{kg\,m^{-3}}]$
σ	Maximum frontal area ratio	[-]
τ	Fraction of the design variable span for the continuous, zero-valued reference values	[-]
Υ	Skew at the leading edge of the blade section	[m]
ϕ	Sweep angle	[rad]
χ	Blade rotation angle	[rad]
ψ	Design variable span	[-]
Ω	Rotational rate	$[\mathrm{s^{-1}}]$
ω	Sliding window size for the termination conditions	[-]

Sub- and Superscripts

Script	Definition
avg	Average
C	Coefficient
c	Camber
e	Edge of the boundary layer
$exit$	Exit
inl	Domain inlet
$invisc$	Inviscid
LE	Leading edge
l	Lower
l	Lift
m	Meridional
max	Maximum
out	Domain outlet
P	Power
p	Pressure
$prop$	Propeller
r	Radial
ref	Reference
T	Thrust
TE	Trailing edge
TO	Take-off
t	Thickness
u	Upper
x	Axial
∞	Freestream condition
θ	Tangential or circumferential
0	Total property

1

Introduction

In recent years, with the increasing socio-political pressure on the aerospace industry to develop (more) environmentally friendly aircraft to help tackle global climate change, a growing body of research has formed into alternative propulsion concepts [2]–[6]. Examples of these concepts include alternative fuels such as (liquid) hydrogen, open fan propulsors, hybrid-electric propulsion architectures, or (electric) ducted fans. Ducted fans are seeing a resurgence in open literature, as the use of electric propulsion systems opens the way to new aircraft configurations. Due to the inherent flexibility in positioning the electric propulsors, new aircraft layouts, utilising concepts such as boundary layer ingestion or distributed propulsion, can be achieved much more easily compared to conventional aircraft.

Ducted fans, also referred to as ducted or shrouded propellers, have been researched and applied to aircraft since the 1930s by Stipa [7], who demonstrated the superior static performance of an intubed propeller and subsequently developed the Stipa monoplane featuring a ducted two-bladed propeller [8]. Although research into ducted fans has been conducted for over 50 years, there is a notable lack of modern studies focused on the conceptual design and architectural layout for large-scale applications of ducted fans, such as the commercial civil aviation industry [9]. Existing literature often focuses on analysing/optimising specific parts of the ducted fan using Computational Fluid Dynamics (CFD) methods, rather than the ducted fan as a whole. If a complete ducted fan is being optimised, this is often done using significant simplifications, such as the modelling of the fan as an actuator disk, while analysing the remainder of the ducted fan using a steady-state viscous method.

Alternatively, low-order axisymmetric panel methods such as the Ducted Fan Design Code (DFDC) [10] are commonly used for parametric design studies. However, these methods cannot be applied in compressible flow applications. A research gap is therefore identified in ducted fan analysis tools to enable medium-fidelity, fast, parametric studies of the complete ducted fan design. This thesis aims to help address the gap in medium-fidelity methods by developing a robust, computationally efficient ducted fan analysis tool and performing design optimisations using the developed tool.

To this end, in chapter 2, the research objective and definition have been defined. This is followed by a literature review of ducted fans presented in chapter 3, including a detailed description of Multi-passage ThroughFLOW (MTFLOW), which is a throughflow solver identified for use in this thesis project. Additionally, an overview of different optimisation algorithms is provided, along with a motivation for the genetic algorithm used. Chapter 4 then presents the steps taken in the modelling of the ducted fan and the adopted parameterisations. In chapter 5, the developed unified ducted fan code, which builds on MTFLOW, is discussed. The implementation of the developed optimisation framework is then presented in chapter 6. Verification and Validation are discussed in chapter 7. The setup of the design optimisations, applied to the Bell X-22A ducted propeller to demonstrate the methods developed in this thesis, is discussed in chapter 8. The results for the single-point design optimisations are discussed in chapter 9, while those for the multi-point design optimisations are discussed in chapter 10. Finally, the conclusions and recommendations of this thesis are presented in chapter 11.

2

Research Definition

This chapter describes the research definition of this thesis. Section 2.1 first describes the defined research objective, while section 2.2 outlines the associated research goals. The research questions and hypotheses are presented in section 2.3. Finally, section 2.4 describes the scope of the thesis.

2.1. Research Objective

This thesis aims to develop a robust and computationally efficient ducted fan analysis tool to address the identified gap between low-fidelity and CFD-based analysis methods. The tool will be applied using optimisation routines to optimise a ducted fan for a given set of objectives, operating conditions, and initial inputs. The aim is to develop a tool capable of analysing a wide range of design variables, such as chord lengths, blade counts, pitch angles, and operating conditions. The research objective is therefore:

To develop a robust, accurate, flexible and computationally efficient ducted fan design and analysis tool, and apply an optimisation method to investigate various ducted fan architectures to identify the optimal configuration, based on specific objectives, flight conditions, and design inputs.

2.2. Research Goals

Based on the defined research objective, four research goals have been identified. The research goals help to break down the research objective into a set of (chronological) activities to achieve the research objective. The defined research goals are:

1. To identify suitable ducted fan analysis methods which can be used in this thesis.
2. To develop a modelling tool for the ducted fan, including the impacts of both the fan and duct design variables on the performance of the ducted fan.
3. To create a (multi-objective) optimisation framework using the to be developed ducted fan modelling tool.
4. To analyse the optimised ducted fan designs obtained from different optimisation problems and compare the found designs.

2.3. Research Questions and Hypotheses

The research goals lead to the definition of the following main research questions:

1. What is the state of the art in ducted fan design and ducted fan modelling?
2. What is the effect of different objective functions in the optimisation procedure on the design and performance of the ducted fan?
3. What are the dominant design variables affecting the performance of the ducted fan for a given operating condition?

At the start of the thesis, the following research hypotheses were defined:

1. The duct length will not have a significant direct influence on the performance of the ducted fan, in line with results reported by Abrego and Bulaga [11].
2. The location of the fan within the ducted fan is a function of the operating conditions, with a more forward rotor location for (near-)static conditions, and an aft location for forward flight, in line with Gao and Xu [12].
3. The performance of the ducted fan will be more sensitive to the design of the duct's leading edge than that of the trailing edge.

2.4. Scope of the Thesis

Given the thesis's timeline and resources, limitations must be placed on its scope. The decision has been made to limit the scope of this thesis to single-stage, fixed pitch ducted fans. Additionally, the scope is limited to ducted fan applications for subsonic freestream Mach number flows. Although local supersonic flow within the flowfield, such as fan blade tip shocks, is considered within the scope of the thesis, this limits the scope of ducted fan research to designs typically seen in the open literature.

No noise considerations are made within this thesis, as this can warrant a separate thesis. It is assumed within this thesis that the overall construction of the ducted fan, including the connection of the duct to the centre body, remains comparable to that seen in current turbofan engines. Finally, no weight estimation considerations are performed in this thesis.

3

Literature Review

This chapter gives an overview of the literature on ducted fan applications and modelling approaches. Section 3.1 first provides a historic overview of ducted fan research, section 3.2 describes the fundamental principles of the ducted fan, section 3.3 reviews the state-of-the-art in ducted fan design, section 3.4 discusses different modelling techniques in open literature to analyse ducted fans, section 3.5 describes the methodology used in the Multi-Passage ThroughFLOW (MTFLOW) solver, and finally, section 3.6 gives an overview of different optimisation techniques and a motivation for the selected optimisation algorithm, the Unified Non-dominated Sorting Genetic Algorithm III (U-NSGA-III).

3.1. Historic Overview of Ducted Fan Research

In 1931, Stipa [7] was one of the first to experimentally investigate the effects of a duct around a propeller, demonstrating the performance benefits of a ducted propeller compared to an unshrouded propeller. A historic review of early studies predating the 1960s was presented by Sacks and Burnell [13], while recent work from Pereira [14] and Akturk and Camci [15] reviewed more modern studies and concepts. This includes the proposed double-ducted fan, which improves the crosswind performance of the ducted fan.

Numerous historical texts exist containing guidelines for designing ducted fans, as well as methods for estimating their performance. An important example is the generalised method of shrouded propeller performance estimation by the United Aircraft Corporation, Hamilton Standard [16]. This was one of the earliest compiled generalised performance methods for ducted fans in axial flow, enabling parametric performance studies. More recently, Wright and Poilenc [17] published a detailed, though more informal, set of design guidelines for ducted fans, aimed at aviation enthusiasts at the turn of the century.

Recent research on ducted fans predominantly focuses on small-scale vertical take-off and landing aircraft, unmanned aerial vehicles, or personal aerial vehicle applications [9], often overlooking larger transport aircraft applications. The core body of research on these large-scale ducted fans originates from the 1950s to the 1970s [9], underlining the need for new research and modelling techniques.

3.2. Fundamental Principles of the Ducted Fan

Compared to an open fan, the presence of the duct causes a reduction in slipstream contraction at static conditions, shown in figure 3.1. This reduction in slipstream contraction results in an increase in mass flow through the fan disk. In turn, this increased mass flow over the upstream surfaces of the duct produces a low-pressure field, resulting in a forward thrust being generated by the duct in addition to the thrust from the fan.

The thrust produced by the fan in the duct is typically lower than that of its unshrouded counterpart. However, the addition of the thrust produced by the duct results in a total thrust which can be significantly larger than that of the isolated fan [18]. A theoretical thrust increase of 26% is obtained at static conditions for $A_{exit} = A_{prop}$. Momentum theory shows that this performance improvement of the duct

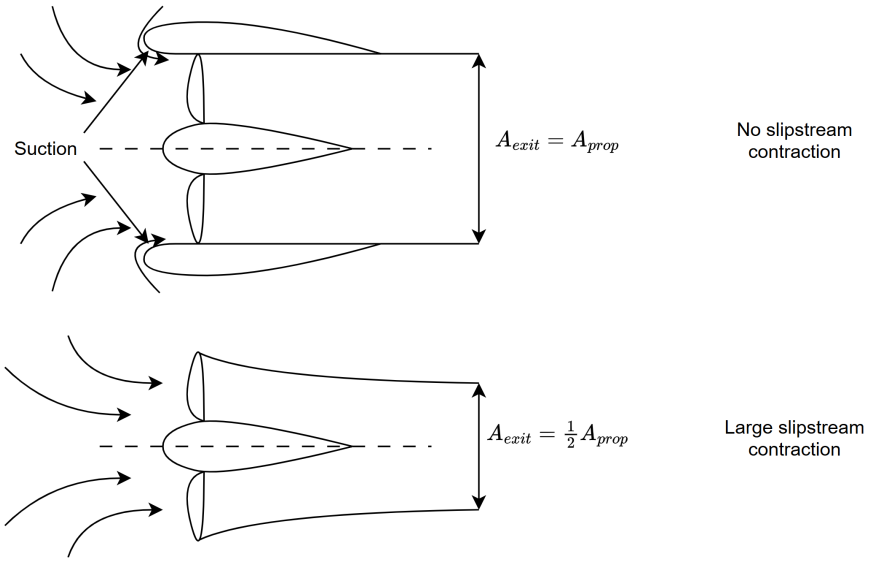


Figure 3.1: Comparison of the momentum flowfields of a shrouded and unshrouded propeller at static conditions. Reproduced from an original by Black, Wainauski and Rohrbach [18].

at static conditions can be achieved so long as the area expansion ratio satisfies $\Lambda > 0.5$, with the improvement being given by equations (3.1) and (3.2) [9].

$$T_{ducted\ fan} = T_{unshrouded} \cdot \sqrt[3]{2\Lambda} \qquad (3.1)$$

$$P_{ducted\ fan} = \frac{P_{unshrouded}}{\sqrt{2\Lambda}} \qquad (3.2)$$

The slipstream contraction for an open rotor at static conditions diminishes as the freestream velocity increases, since the initial momentum increases. As the effect of the duct is to reduce this contraction, this benefit also decreases. With increasing velocity, the thrust due to the presence of the duct decreases or becomes negative [18]. It is therefore key to consider all flight conditions in the design of a duct to minimise the performance penalty incurred.

3.3. State-of-the-Art in Ducted Fan Design

Zhang and Barakos [9] presented a review on ducted fans, targeted at compound rotorcraft, in which they outlined both historic and recent experiments, as well as theoretical studies on ducted fans for aeronautical applications. This work also contained a summary of key focus areas for ducted fan research, where noise emissions, crosswind effects, control systems, tip clearance, and coaxial ducted fans were identified as key aspects.

Both Parlett [19] and Graf, Fleming and Ng [20] investigated the effects of duct lip radius on static thrust, concluding that the static thrust increases with increasing lip radius. Graf, Fleming and Ng [20] further found that a small lip radius is more favourable in crosswind and forward flight conditions. Further research on the duct shape, including the performance impacts at different advance ratios, was conducted by Biava and Barakos [21]. They showed that the rear part of the duct acts as a diffuser, increasing the static pressure. This increases the overall thrust compared to an unshrouded propeller, contributing to the increased efficiency of the ducted fan system at static/low-speed conditions. Biava and Barakos [21] suggest that the loss of performance of the duct at higher speeds could be minimised by using variable geometry or multiobjective optimisation of the ducted fan system.

Although numerous articles exist on the design of the duct shape, limited attention has been paid to the location of the fan within the duct. This may be because the absolute location of the fan within the duct can vary significantly between two different duct designs, as stated by Sacks and Burnell [13]. Anderson, Lehmkuehler, Ho *et al.* [22] focused on investigating the location of a propeller within an annular wing. Pusher and tractor configurations were analysed, where an increase in static thrust was observed for the pusher configuration compared to the tractor configuration. This increase was attributed to a reduction in the drag of the duct.

Further work by Gao and Xu [12] suggests that the ducted fan performance is susceptible to the location of the rotor within the duct. Relative locations of 15%, 30%, and 65% of the duct length were analysed, where a rotor located at 15% of the duct length performed best, followed by a location at 65%. However, it is not possible to determine an overall trend of performance versus relative rotor location due to the low number of data points analysed.

The relative rotor location was further analysed by Bi, Kimmel and Haas [23]. A baseline configuration, with the fan located at $x = 0.2R$ from the duct inlet, was compared with $x = 0.05R$ and $x = 0.5R$, where R is the fan radius. Both of these configurations resulted in a 3% reduction in efficiency. Similarly to Gao and Xu [12], Bi, Kimmel and Haas [23] also mention that further studies are needed to optimise the fan location within the duct.

Typical ducted fan design studies tend to focus on optimising the blade design [24] or the duct [25], while keeping the remainder of the design constant, to improve performance at a fixed operating point. Such studies focus on the optimisation of a singular ducted fan architecture, rather than considering larger, fundamental changes to the ducted fan design, together with the integrated complete optimisation of the ducted fan system.

3.4. Modelling Techniques for Ducted Fans

To enable rapid analysis and optimisation of a complete ducted fan, a fast, accurate, and flexible design/analysis model is needed. While axial fan conceptual design is widely studied and documented in open literature by, for example, Dixon and Hall [26], Gambini and Vellini [27], or Denton [28], ducted fans experience a close coupling between the design of the blade row(s) and the duct design [18]. This necessitates the inclusion of duct design in the design process, along with the three-dimensional design of the blade row(s). The design process of a ducted fan typically consists of the meanline conceptual design, the design of the blade row(s), and detailed CFD simulation validation. The blade row design is often coupled with a (quasi) CFD simulation to improve fidelity [28].

This section discusses different conceptual modelling techniques for a ducted fan. Additionally, a motivation is given for the use of Multi-passage ThroughFLOW (MTFLOW) in this thesis. Section 3.4.1 describes incompressible flow models, while compressible flow modelling techniques are discussed in section 3.4.2.

3.4.1. Incompressible Flow Modelling Techniques

Common lower-order ducted fan design and analysis tools are based on a blade element momentum theory or actuator disk approach for the fan design, coupled with a vortex sheet panel grid to model the duct and centre body [29]. An example of the vortex sheet panel discretisation applied in such a method is shown in figure 3.2. These methods often include a coupled boundary layer of the duct and centre body to calculate the viscous forces, as they are otherwise inherently inviscid. A notable example of such a code is the Ducted Fan Design Code (DFDC) by Youngren, Drela and Sanders [10].

Although DFDC is widely used in academia and industry [31], it is no longer actively supported, with the latest version dating from late 2005 [10]. An improvement on DFDC, DuctAPE, has recently been published by Mehr and Ning [25]. DuctAPE implements the same methodology as DFDC but is specifically designed for efficient gradient-based design optimisation, and is one of the first implementations of a ducted fan design and analysis code for this purpose. However, the underlying equations and assumptions of both DFDC and DuctAPE limit their application to low Mach numbers, making them unsuitable for use in the context of this thesis's objectives.

3.4.2. Compressible Flow Modelling Techniques

To enable the widespread applicability of a ducted fan model, it is important to have the capability to analyse both high (subsonic) and low freestream Mach number flows. One potential candidate for this is the use of the compressible axisymmetric Euler equations to describe the inviscid part of the flow, coupled with a model for the boundary layer growth and viscous wakes on the axisymmetric meridional surfaces, in throughflow modelling [26].

Internal turbomachinery flows form one of the most complex aerodynamic problems due to the three-dimensional nature of the flowfield and the interactions between the different aerodynamic surfaces

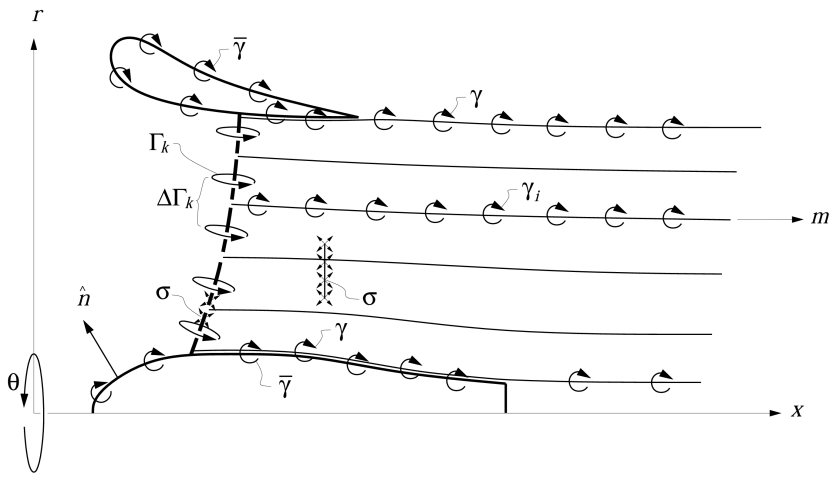


Figure 3.2: Vortex sheet discretisation used to represent the meridional velocity v_m in the Ducted Fan Design Code. Taken from Drela and Youngren [30].

[32], [33]. As such, reduced-order models are typically used at the beginning of the design process. These models are then refined throughout the design process. In doing so, a large portion of the final design is fixed during the initial one- and two-dimensional design phases [32].

Two-dimensional axisymmetric throughflow modelling plays a key role in these early design stages. Such models maintain the simplicity of one-dimensional models while achieving the accuracy of more complex 3D models. Throughflow models enable rapid simulation of the key turbomachinery flow aspects, including off-design analysis. This makes it a suitable approach for rapidly analysing or optimising a ducted fan [26], [32], [33], as desired within this thesis. Within throughflow modelling, the streamline curvature method is commonly used due to its simplicity, robustness, ability to handle supersonic flow, and advantageous nature for fan blade design [34]–[36].

By using a solver capable of solving the flow both through and around the ducted fan, such as Multi-passage ThroughFLOW (MTFLOW), the complete ducted fan can be analysed at a low computational cost with high fidelity [37]. MTFLOW uses an integral boundary layer method to resolve the axisymmetric boundary layers, together with a finite-volume discretised streamline-curvature formulation for the main inviscid flow [38]. This yields an order-of-magnitude reduction in computational cost compared to more detailed CFD flow solvers, with the finite volume discretisation correctly capturing any shock-waves. The flexibility of MTFLOW, together with its proven accuracy against validation cases in Renick [38] and Hanson [39], make it the tool of choice for this thesis.

3.5. Multi-Passage ThroughFLOW

As described in section 3.4, this thesis will use MTFLOW as a building block for the analysis of different ducted fan designs. This section presents an overview of MTFLOW. A further description of several aspects of MTFLOW is given in appendix A.

3.5.1. The Streamline Curvature Method

MTFLOW uses a streamline curvature method to resolve the flowfield shown in figure 3.3, based on a conservative finite volume implementation of the steady-state Euler equations developed by Drela, Giles and Thompkins Jr. [40]. This implementation addresses the numerical instability in supersonic flow and the incorrect positioning of shocks present in the original streamline curvature method presented by Novak [41]. The finite volume implementation, developed by Drela, Giles and Thompkins Jr. [40], is discussed in the following subsection. A description of the original streamline curvature method can be found in sources like Novak [41] or Renick [38]. The fundamental assumptions of the streamline curvature method, as given by Dixon and Hall [26], are:

- The flow is steady in both the absolute and relative frames of reference.
- The flow is axisymmetric.

- Outside of the blade rows, the effects of wakes from any upstream blade rows are mixed out to give a uniform circumferential condition.
- The blade rows are modelled as passage-averaged body forces and loss coefficients.

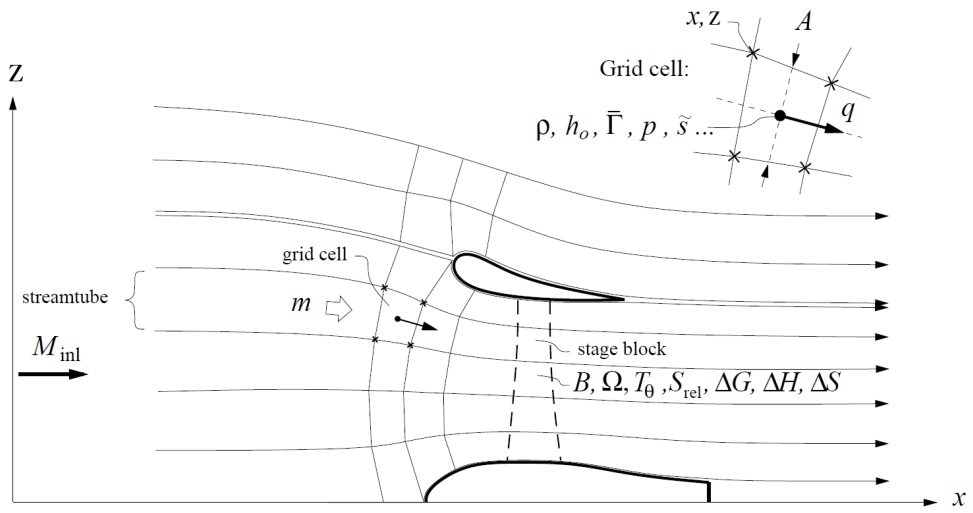


Figure 3.3: The axisymmetric meridional flowfield solved in MTFLOW, showing a streamline-based grid with a stage block in which the blade forcing fields are defined [37].

3.5.2. Finite Volume Implementation

Drela, Giles and Thompkins Jr. [40], [42] formulate the steady state equations using a streamtube conservation cell, shown in figure 3.4, where the densities ρ, meridional velocities v_m, end face pressures p, and streamline face pressures Π are the surface averages on each of the four cell surfaces.

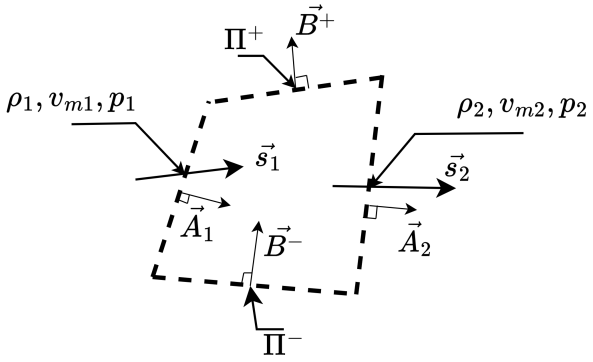


Figure 3.4: The conservation cell for the finite volume implementation of the streamline curvature method. Reproduced from an original by Drela, Giles and Thompkins Jr. [40].

Assuming there is no convection across the streamline faces, conservation of mass and momentum can be expressed as equations (3.3) and (3.4), where $\vec{s}$ is a unit vector in the direction of the local meridional velocity, $\vec{A}_{1,2}$ are the area vectors normal to the streamtube volume along the streamtube, and $\vec{B}^{\pm}$ are the area vectors normal to the streamtube volume and normal to the streamtube. Conservation of enthalpy is then formulated as equation (3.5), where κ is the ratio of specific heats, taken as $\kappa = 1.4$ for air within MTFLOW [37]. By further assuming that the pressure change is linear over the conservation cell, it follows that equation (3.6) must hold as well [40].

$$\rho_1 v_{m1} \vec{s_1} \vec{A}_1 = \rho_2 v_{m2} \vec{s_2} \vec{A}_2 \tag{3.3}$$

$$p_1\vec{A_1} + (\rho_1 v_{m_1}\vec{s_1}\cdot\vec{A_1})v_{m1}\vec{s_1} + \Pi^-\vec{B^-} = p_2\vec{A_2} + (\rho_2 v_{m_2}\vec{s_2}\cdot\vec{A_2})v_{m2}\vec{s_2} + \Pi^+\vec{B^+} \tag{3.4}$$

$$\frac{\kappa}{\kappa-1}\frac{p_1}{\rho_1} + \frac{1}{2}v_{m1}^2 = \frac{\kappa}{\kappa-1}\frac{p_2}{\rho_2} + \frac{1}{2}v_{m2}^2 \tag{3.5}$$

$$p_1 + p_2 = \Pi^+ + \Pi^- \tag{3.6}$$

To solve the meridional flowfield, an initial grid distribution is specified in MTFLOW using the methods developed by Thompson, Thames and Mastin [43]. In this process, the governing inviscid, incompressible, axisymmetric flow equations are first put into a streamfunction form and subsequently solved using an elliptic grid generation technique. The generated streamline grid then corresponds to the physical streamlines of an axisymmetric, inviscid, incompressible flow. A Newton method is used to solve the streamtube equations, resulting in the pressures Π^+ and Π^- on the streamtube walls. The corresponding pressure difference, together with the local curvature of the streamline, is then used to iteratively update the streamtube positions to their correct locations [38].

3.5.3. The Throughflow Formulation in Multi-Passage ThroughFLOW

MTFLOW further simplifies equations (3.3) to (3.6) by creating an explicit set of equations that is not differential. The meridional velocity is obtained from conservation of mass using the streamtube area, corrected for the blockage due to the presence of any blade rows as given by equation (3.7) [37], where $\dot{m}$ is the streamtube mass flow, A is the streamtube area, B is the number of blades and T_θ the circumferential blade thickness at the local radius z.

$$v_m = \frac{\dot{m}}{\rho A(2\pi z - BT_\theta)} \tag{3.7}$$

The streamwise momentum equation is given by equation (3.8), where the differentials are taken using a finite-volume conservative form along each streamtube to give correct shock capturing [37]. In this equation, v_θ is the tangential velocity, while $d(\Delta S)$ is the entropy change, Ω the rotational rate of the blade row(s) and $d\bar{\Gamma}$ the swirl change caused by the blade row(s).

Rather than solving a differential energy equation, MTFLOW solves for the total enthalpy h_0 using equation (3.9). This is a more efficient method, requiring fewer iterations. For the ducted fan concepts considered in this thesis, there is no external heat addition, such that $d(\Delta H) = 0$. Outside of the blade rows, it further holds that $d\bar{\Gamma} = 0$, so this equation reduces to a statement of constant enthalpy. For streamtubes with no swirl addition to the flow, equation (3.9) then reduces to $h_0 = h_{0_{inlet}}$. This allows the equation to be removed from the system, giving a significant speed-up in the solving process.

$$dp + \rho v_m dv_m + \rho v_\theta dv_\theta + pd(\Delta S) - \rho\Omega d\bar{\Gamma} = 0 \tag{3.8}$$

$$dh_0 - d(\Delta H) - \Omega d\bar{\Gamma} = 0 \tag{3.9}$$

The tangential momentum equation solved along a streamtube is defined by equation (3.10), which is equivalent to the preservation of swirl $\bar{\Gamma}$ if no forcing is applied. Any swirl accumulation along a streamtube is equal to the circulation shed off the propulsor blades [38]. The swirl change $d(\Delta G)$ across a stage can be either provided as an input or computed on a defined geometry in the analysis mode of the throughflow method [26].

$$d\bar{\Gamma} - d(\Delta G) = 0 = \bar{\Gamma}_i - \bar{\Gamma}_{i-1} - \Delta G_i + \Delta G_{i-1} \tag{3.10}$$

$$\Delta G_i = \begin{cases} \Delta G(\bar{x_i}, \bar{z_i}) & \text{in a stage} \\ 0 & \text{otherwise} \end{cases} \tag{3.11}$$

Due to the discrete nature of the computational domain, the discrete forcing field parameters are defined at each grid cell-centre, using equation (3.11). The tangential momentum equation can then be differenced to obtain the right-hand side of equation (3.10). This equation ensures that any field parameter term appears as a difference along the streamtube and never straddles the outflow edge of a stage, such that the changes across successive stage blocks are accumulated [37].

The final set of equations then allows for the effects of entropy, enthalpy, and rothalpy to enter the flowfield. This is done using the discrete stage blocks shown in figure 3.3. In these stage blocks, the circumferential blade thickness T_θ, swirl ΔG, and entropy ΔS are input as forcing terms in the flowfield equations [37].

3.5.4. Viscous Flow Modelling

MTFLOW uses a strongly coupled integral boundary layer method to resolve the boundary layers on the axisymmetric surfaces, if desired by the user [37]. The viscous and inviscid flows are strongly coupled through the displacement thickness δ^*. This enables an equivalent inviscid flow to be defined and resolved for the main flow away from the meridional surfaces, while the viscous losses over the meridional surfaces can also be resolved. The integral boundary layer equations used in MTFLOW are included in appendix A.2 as a reference.

3.5.5. Normalisation in Multi-Passage ThroughFLOW

The flow parameters in MTFLOW are defined using the reference Mach number M_{inl} and the reference Reynolds number Re_{inl}. These quantities are then used to determine the reference velocity v_{inl} and viscosity μ_{inl} through equations (3.12) and (3.13), where L_{ref} is the reference length used to normalise the geometric parameters, taken as the fan diameter [38].

$$v_{inl} = M_{inl} a_{inl} \quad (3.12) \qquad \mu_{inl} = \frac{\rho_{inl} v_{inl} L_{ref}}{Re_{inl}} \quad (3.13) \qquad Re_{inl} = \frac{\rho_{0_{inl}}}{\rho_{inl}} \frac{a_{0_{inl}}}{v_{inl}} \frac{\mu_{inl}}{\mu_{0_{inl}}} \quad (3.14)$$

MTFLOW defines the reference stagnation density and enthalpy as equations (3.15) and (3.16), with the reference stagnation speed of sound taken as $a_{0_{inl}} = 1$. The reference stagnation pressure is then $p_{0_{inl}} = 1/\kappa$, as implied by the state equation (3.17). The rotational rate of the blade row(s), Ω_{MTLOW}, is non-dimensionalised using the inlet reference velocity and the reference length as given by equation (3.18) [37]. All internal calculations use the stagnation-condition Reynolds number, equation (3.14), which uses Sutherland's viscosity law [37], [44]. All arithmetic and outputs in MTFLOW are expressed in dimensionless ratios, such as M, C_p, or v_m/v_{inl}.

$$\rho_{0_{inl}} = 1 \quad (3.15) \qquad h_{0_{inl}} = \frac{1}{\kappa - 1} \quad (3.16)$$

$$\kappa p = \rho a^2 = (\kappa - 1)\rho h \quad (3.17) \qquad \Omega_{MTFLOW} = \Omega \frac{v_{inl}}{L_{ref}} \quad (3.18)$$

3.5.6. The Multi-Passage ThroughFLOW Programs

MTFLOW consists of three distinct programs, following a similar structure to that found in MSES [45] or MISES [46], with the three programs being MTSET, MTFLO, and MTSOL [37]. The first program, MTSET, reads the axisymmetric meridional surface geometry input file and generates the initial inviscid, incompressible streamsurface grid. This grid is written to a binary state file. MTFLO then reads the blade forcing field distributions from a specified file and merges them into this state file. Finally, MTSOL is the main program in MTFLOW and is responsible for executing the Newton iterations to obtain the converged solution. A data flow diagram of the MTFLOW software is included for reference in appendix A.4.

3.5.7. Multi-Passage ThroughFLOW Outputs

MTFLOW can output three sets of data. The most commonly used output is the integrated force data, which includes the force (T_C) and power coefficient (P_C) viscous and inviscid contributions, and the propulsor efficiency, $\eta_p = T_C/P_C$. Additionally, MTFLOW can output the boundary layer data for each of the meridional surfaces obtained from the integral boundary layer equations, as well as the flowfield parameters at every grid cell in the domain [37]. The definitions for the thrust and power coefficients T_C, P_C in MTFLOW are given for reference in appendix A.3.

3.6. Optimisation Algorithms

Within this thesis, a ducted fan optimisation framework will be developed and applied. This requires a review of optimisation procedures. The focus of this section is on multi-objective optimisation due to the complex (multidisciplinary), (multi-)objective nature of ducted fan optimisation, where the number of objectives must be easily changeable within the framework. Two optimisation methods are considered in this thesis: gradient-based methods and Genetic Algorithms (GAs).

A gradient-based optimisation routine uses the gradient information obtained from the design vector to

optimise a design [47]. Considering the MTFLOW software, this means at least $2n$ MTFLOW evaluations for a 2nd-order differencing scheme, or $3n$ for a 3rd-order scheme are required, where n is the number of design variables [48]. This negatively impacts computational efficiency. Additionally, since both continuous and discrete variables are used in MTFLOW, gradients of, for example, the blade count are not continuous, requiring additional handling. As such, gradient-based methods are not considered further, and the focus is put on GAs.

A GA can easily accommodate a wide range of variability in a design in the initial parent population, allowing the algorithm to incorporate a larger variability into the evaluation process [49]. These methods work by evolving a population of design vectors through selection, reproduction, and mutation to improve the objective values, in a process inspired by Darwinian evolution [47]. However, the ability of a GA to find a local or global minimum is severely affected by the population size, since too small a population may lead to a poor resolution [49]. Since all individuals can be evaluated simultaneously [47], [49], a genetic algorithm can be computationally efficient when using parallelisation. This reduces the computational time needed to perform the optimisation and is often a requirement to solve an optimisation problem with larger population sizes efficiently.

It is generally accepted that genetic algorithms have ”a useful role to play in the initial stages of a multidisciplinary design optimisation, where the size of the problem is relatively small but the variability large” [47, p. 88]. This illustrates the suitability of genetic algorithms for optimisation, considering the objective of this thesis, where a large spread in individuals may reasonably be expected. The output from the GA can be further improved using a gradient-based method to confirm the optimality of the found solution [47] as the next step in the design process. Such a step is therefore not implemented in this thesis.

As the number of objective functions and constraints need not be fixed within the optimisation framework, a singular optimisation procedure that can handle different numbers of objective functions and constraints without requiring (extensive) rework is desired. A suitable algorithm to fulfil this purpose has been identified in the Unified Non-dominated Sorting Genetic Algorithm III (U-NSGA-III) [50].

U-NSGA-III is an improvement of the NSGA-III algorithm, developed to unify traditional single- and many-objective optimisation algorithms while maintaining a competitive numerical efficiency. It forms part of the decomposition-based algorithms, which decompose the problem into several sub-problems through the use of reference directions, with each reference direction targeting different parts of the Pareto front [50], [51].

The benefit of U-NSGA-III over other algorithms is the one-time implementation of the solution representation, operators, objective, and constraint formulations across different objective dimensions [50]. Experiments have shown that this algorithm can match or even exceed specific peer optimisation algorithms for single, dual, multiple, or many objective problems [50]. This makes it a prime candidate for use within this thesis project. A Python implementation of U-NSGA-III is given in the Pymoo framework by Blank and Deb [52].

4

Modelling of the Ducted Fan

This chapter describes the methodology used in this thesis to construct the aerodynamic profile shapes of the centrebody, duct and fan blade section, together with the associated parameterisation of the blade sections and the modelling of the blade row forcing fields. Section 4.1 first introduces the Bézier-Parsec 3434 parameterisation method. Next, sections 4.2 and 4.3 describe the adopted parameterisations. Finally, section 4.4 describes the blade forcing field modelling used and the steps required to compute the blade row inputs for the MTFLOW solver, based on the blade geometries.

4.1. The Bézier-Parsec 3434 Parameterisation Method

To model the aerodynamic profiles in the ducted fan, the Bézier-Parsec parameterisation method developed by Rogalsky [53] is used. This method combines the strengths of the well-known PARSEC method with Bézier curves to give a flexible aerodynamic shape parameterisation method. The profile shape is defined using four Bézier curves: the leading and trailing edge thickness and camber curves, shown in figure 4.1.

This is beneficial, as it gives the ducted fan model direct control over the camber distribution, which is used internally in MTFLOW to compute the swirl generated by the blade row(s), which is discussed in section 4.4. In turn, this enables more direct feedback to the optimisation framework, improving performance. The Bézier-Parsec method has also been applied in genetic algorithms before by Rogalsky [53] and Kharal and Saleem [54], further supporting its use in this thesis.

$$\begin{aligned} x(a) &= x_0(1-a)^3 + 3x_1a(1-a)^2 + 3x_2a^2(1-a) + x_3a^3 \\ y(a) &= y_0(1-a)^3 + 3y_1a(1-a)^2 + 3y_2a^2(1-a) + y_3a^3 \end{aligned} \tag{4.1}$$

$$\begin{aligned} x(a) &= x_0(1-a)^4 + 4x_1a(1-a)^3 + 6x_2a^2(1-a)^2 + 4x_3a^3(1-a) + x_4a^4 \\ y(a) &= y_0(1-a)^4 + 4y_1a(1-a)^3 + 6y_2a^2(1-a)^2 + 4y_3a^3(1-a) + y_4a^4 \end{aligned} \tag{4.2}$$

To ensure sufficient control over the trailing edge shape, 4th-degree trailing edge camber and thickness curves are used, whereas 3rd-degree curves are used for the leading edge. This is referred to as the BP3434 method. A third-order Bézier curve is given by equation (4.1), while a fourth-order Bézier curve is defined by equation (4.2) [55].

A total of ten aerodynamic parameters and five Bézier parameters is needed to apply the BP3434 method. The ten aerodynamic parameters used are the leading edge radius r_{LE}, the trailing camber line angle α_{TE}, the trailing wedge angle β_{TE}, the trailing edge vertical displacement z_{TE}, the leading edge direction γ_{LE}, the location of the camber crest (x_c, y_c), the location of the thickness crest (x_t, y_t), and the half thickness of the trailing edge dz_{TE} [53]. The five Bézier parameters are b_0, b_2, b_8, b_{15} and b_{17}, and are defined as the ratios between the endpoints of the curves they define [53]. For example, variable b_{17} describes the inflexion point of the trailing edge camber and is defined as

$b_{17} = (x - x_c)/(1 - x_c)$, where x is the actual x-value of the fourth control point on the curve. The control point definitions for the Bézier curves are presented in tables 4.1 and 4.2.

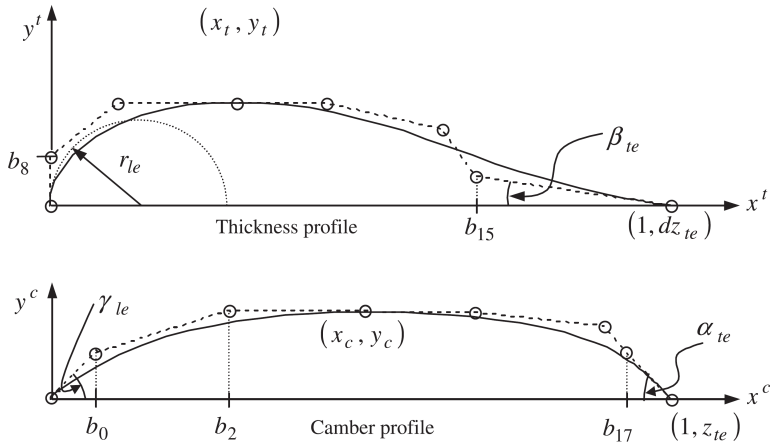


Figure 4.1: Definition of the thickness and camber curves for the Bézier-Parsec 3434 parameterisation method. Taken from Derksen and Rogalsky [55].

Table 4.1: Definition of the control points for the thickness distributions. Taken from Derksen and Rogalsky [55].

Leading edge		Trailing edge	
$x_0 = 0$	$y_0 = 0$	$x_0 = x_t$	$y_0 = y_t$
$x_1 = 0$	$y_1 = b_8$	$x_1 = \dfrac{7x_t + \frac{9b_8^2}{2r_{LE}}}{4}$	$y_1 = y_t$
$x_2 = \dfrac{-3b_8^2}{2r_{LE}}$	$y_2 = y_t$	$x_2 = 3x_t + \dfrac{15b_8^2}{4r_{LE}}$	$y_2 = \dfrac{y_t + b_8}{2}$
$x_3 = x_t$	$y_3 = y_t$	$x_3 = b_{15}$	$y_3 = dz_{TE} + (1 - b_{15})\tan(\beta_{TE})$
		$x_4 = 1$	$y_4 = dz_{TE}$

Table 4.2: Definition of the control points for the camber distributions. Taken from Derksen and Rogalsky [55].

Leading edge		Trailing edge	
$x_0 = 0$	$y_0 = 0$	$x_0 = x_c$	$y_0 = y_c$
$x_1 = b_1$	$y_1 = b_0 \tan(\gamma_{LE})$	$x_1 = \dfrac{3x_c - y_c \cot(\gamma_{LE})}{2}$	$y_1 = y_c$
$x_2 = b_2$	$y_2 = y_c$	$x_2 = \dfrac{-8y_c \cot(\gamma_{LE}) + 13x_c}{6}$	$y_2 = \dfrac{5y_c}{6}$
$x_3 = x_c$	$y_3 = y_c$	$x_3 = b_{17}$	$y_3 = z_{TE} + (1 - b_{17})\tan(\alpha_{TE})$
		$x_4 = 1$	$y_4 = z_{TE}$

The thickness distribution is defined perpendicular to the camber distribution [44, p. 326], such that the camber slope angle ζ is required, given by equation (4.3). The upper and lower surface coordinates are then defined using equation (4.4), where the subscripts t and c refer to the thickness and camber distributions, respectively.

$$\zeta = \arctan(\frac{dy_c}{dx}) \qquad (4.3)$$

$$\begin{aligned} x_u &= x_c - y_t \sin(\zeta) & y_u &= y_c + y_t \cos(\zeta) \\ x_l &= x_c + y_t \sin(\zeta) & y_l &= y_c - y_t \cos(\zeta) \end{aligned} \qquad (4.4)$$

To enforce a feasible profile shape for the BP3434 parameterisation, it is required that all Bézier curves are one-to-one. Additionally, equation (4.5) must be satisfied for the profile to have a positive thickness

and the correct crest curvature [53].

$$0 < b_8 < \min\left(y_t, \sqrt{-2r_{LE}x_t/3}\right) \tag{4.5}$$

The BP3434 method does not allow for negative camber distributions. This could be addressed by adding a Boolean variable to the BP3434 design variables, which controls the mirroring of the upper and lower surfaces of the profile about the x-axis. However, this has not been implemented in this thesis, as this is considered out of scope for this thesis.

4.2. Blade Row Parameterisation

The blade rows are defined using two sub-parameterisations: the distribution of the blade shape along the span, and the blading in three-dimensional space. The distribution of the blade shape along the blade span is defined by specifying $n \geq 2$ BP3434 parameterised sections along the blade span. The blading is then defined using the geometric blade angle β_{geom_i}, leading edge sweep ϕ_i, chord length c_i, and leading edge skew Υ, defined at the same radial stations z_i corresponding to the defined n blade profiles along the blade radius R. Besides these radial distributions, the root leading edge x-coordinate x_{root}, the set angle at the blade tip, β_{tip}, and the blade count B must be defined. However, since MTFLOW solves the axisymmetric meridional flowfield, the solver is indifferent to the variations in the skew Υ along the blade span. As such, the leading edge skew is held fixed such that $\Upsilon = 0$ everywhere.

The root sweep angle is fixed at $\phi_{root} = 0°$, since the leading edge x-coordinate is defined by equation (4.6). By defining the geometric blade angle at the tip to be fixed at $\beta_{geom_{tip}} = 0°$, a further design variable is removed, and more direct control over the blade angle distribution defined by equation (4.12) is obtained. The adopted blade parameterisation is shown in figure 4.2.

$$x_{LE_i} = z_i \tan(\phi_i) + x_{root} \tag{4.6}$$

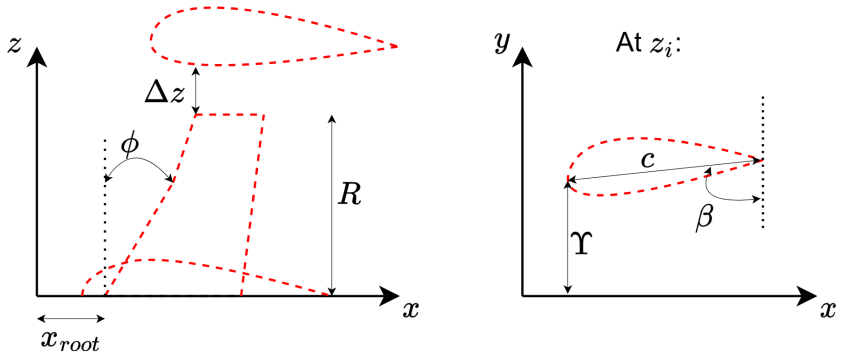


Figure 4.2: The adopted three-dimensional blade row parameterisation.

4.3. Axisymmetric Meridional Geometry Parameterisation

The centrebody and duct geometries are represented using the BP3434 method described in section 4.1, where the chord length and leading edge x-coordinates are introduced as additional variables. The leading edge of the centrebody is held fixed at $x = 0$ m, while the duct leading edge x-coordinate is a design variable.

The duct z_{LE} is determined using the blade row parameterisation shown in figure 4.2 and the duct geometry, using a minimum tip gap Δz. The leading edge x-coordinate of the blade tip is defined as $x_{LE_{tip}} = x_{root} + r \cdot \tan(\phi_{tip})$. With a known duct geometry and leading edge x-coordinate, the leading edge z-coordinate of the duct can then be expressed as equation (4.7), where $z_{duct_{lower\ surface}}$ is the local lower surface z-coordinate at the leading or trailing edge of the blade tip.

$$z_{LE_{duct}} = R + \Delta z + \max(|z_{duct_{lower\ surface}}(x_{LE_{tip}})|, |z_{duct_{lower\ surface}}(x_{TE_{tip}})|) \tag{4.7}$$

4.4. Blade Row Forcing Field Modelling

To model a blade row in MTFLOW, the swirl change ΔG, entropy change ΔS, and circumferential blade thickness T_θ across a stage block are required, as described in section 3.5.3. To model the entropy distribution, a cascade solver is needed, since $\Delta S = f(x, z)$ on the stage block. This is left out of scope in this thesis due to resource limitations. Future work should look into including the entropy distributions in each stage block. This leaves the swirl distribution ΔG and circumferential blade thickness T_θ as distributions of interest.

To define the swirl distribution ΔG, an external cascade solver such as MISES [46] or Multall [28] can be used. This is the most accurate approach; however, due to the same reasoning as above, it is considered outside the scope of this thesis. An alternative option is to compute the swirl generation based on the provided geometry of the blade row using a simplified model. Since the blade rows have a defined geometry using the BP3434 parameterisation, this approach is used in this thesis.

4.4.1. Swirl Generation

The tangential momentum equation was defined as equation (3.10). In the analysis mode of the through-flow method, the swirl evolution $d\bar{\Gamma}/ds$ along a streamtube can be computed directly from a given blade geometry using the geometric blade slope distribution, S_{rel}, of the blade camberline. This is done using a blade row flow-tangency condition by solving for a rate equation for the swirl, given by equation (4.8), where s is the distance along the meridional streamline.

$$\frac{d\bar{\Gamma}}{ds} = k(\bar{\Gamma_0} - \bar{\Gamma}) \tag{4.8}$$

$$\bar{\Gamma}_0 = (v_m S_{rel} + \Omega z)z \tag{4.9}$$

In this equation, $\bar{\Gamma}_0$ is the high-solidity limiting swirl, derived from the local relative-frame flow tangency condition to the blade slope S_{rel} as equation (4.9). The rate constant k is obtained by requiring the correct blade lift in the low solidity limit. This rate constant is given by equation (4.10), where modest incidence and swirl angles are assumed, together with the incompressible low-solidity 2D lift curve slope $C_{l_\alpha} = 2\pi \text{ rad}^{-1}$, such that the right-hand simplification can be used [37].

$$k = \frac{B}{4\pi z} C_{l_\alpha} \frac{v_m \sqrt{1 + S_{rel}^2}}{\sqrt{v_m^2 + (\Omega z - \bar{\Gamma}/z)^2}} \approx \frac{B}{2z} \tag{4.10}$$

Equations (4.8) to (4.10) are available in MTFLOW and used in this thesis, where the unified ducted fan code developed in chapter 5 computes the blade slope and circumferential blade thickness distributions based on the defined parameterisations in sections 4.2 and 4.3. The methods developed in this thesis to compute the blade slope and circumferential blade thickness are described in sections 4.4.3 and 4.4.4.

4.4.2. Transformation to the Cylindrical Coordinate System

To compute the geometric blade slope S_{rel} and the circumferential blade thickness T_θ, a cylindrical coordinate system is used. This cylindrical coordinate system is defined so that the cylindrical axis coincides with the x-axis. By applying a standard coordinate transformation from the Cartesian coordinate system (x, y, z), with $\iota = y/z$, equation (4.11) is obtained [56].

$$(x, y, z)_{cyl} = \begin{bmatrix} x \\ z\sin(\iota) \\ z\cos(\iota) \end{bmatrix} \tag{4.11}$$

4.4.3. Calculation of the Geometric Blade Slope

To compute the geometric blade slope S_{rel} of the camberline, several steps are taken. From a BP3434 parameterisation, the thickness and camber distribution for the blade section are obtained. These can be used to construct the upper and lower blade surfaces through equations (4.3) and (4.4).

Each blade section is then placed at a blade angle β_i, defined using the geometric blade angle β_{geom_i} of the local section, the tip blade angle $\beta_{geom_{tip}}$, and the set angle at the tip β_{tip}, using equation (4.12). Since $\beta_{geom_{tip}} = 0°$, this simplifies to the right-hand side of the equation shown. The blade section is rotated counterclockwise over $\chi_i = 90° - \beta_i$ to comply with the MTFLOW sign conventions.

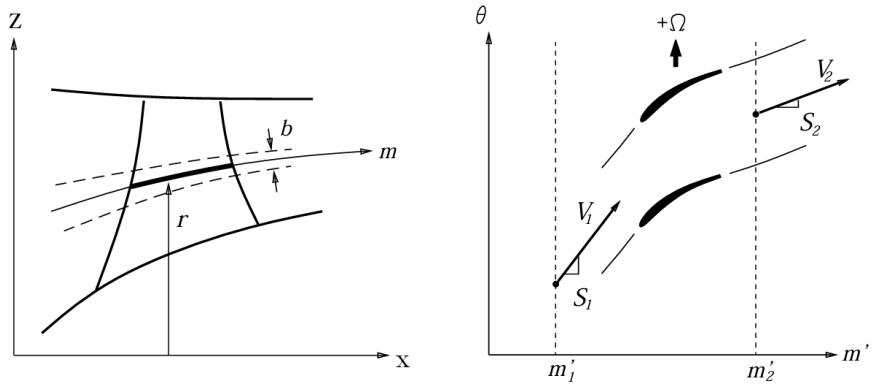


Figure 4.3: The blade-to-blade streamsurface coordinate system [46].

$$\beta_i = \beta_{geom_i} + \beta_{tip} - \beta_{geom_{tip}} = \beta_{geom_i} + \beta_{tip} \quad (4.12) \qquad \theta_{camber} = \arctan(\frac{y_{camber}}{z_{camber}}) \quad (4.13)$$

By then applying the defined cylindrical coordinate transformation, the geometric blade slope S_{rel} is defined in the blade-to-blade stream surface coordinate system of figure 4.3 as equation (4.17). Since the $m' - \theta_{camber}$ plane is angle-preserving, the blade slope is directly related to the blade angle. The circumferential angle θ_{camber} is then defined as equation (4.13). To compute the blade slope, the m' coordinate is defined using equation (4.14) [57].

Here, r is the local streamsurface radius defined by equation (4.16), and dm is the physical arc length increment projected onto the meridional $r - z$ plane, given by equation (4.15). For a general blade section, this integral has no analytical solution [46], [57], so numerical integration is required. In this thesis, trapezoidal integration is used [46].

$$m' = \int \frac{dm}{dr} \quad (4.14) \qquad dm = \sqrt{dr^2 + dx^2} \quad (4.15) \qquad r = \sqrt{y_{camber}^2 + z_{camber}^2} \quad (4.16)$$

$$S_{rel} = \frac{d\theta_{camber}}{dm'} = \tan(\beta) \quad (4.17)$$

4.4.4. Calculation of the Circumferential Blade Thickness

The circumferential blade thickness is the arc length occupied by the blade at every coordinate (x, z) along the duct passage [38], and is used to account for the local passage blockage due to the blade row(s). Using the profile blade rotation angle χ and coordinate transformation defined earlier, the circumferential blade thickness can be expressed as equation (4.18). In this equation, the subtended angles for the upper and lower blade surfaces, θ_i, are given by equation (4.19). To avoid complete blockage in any streamtube, it is required that $T_\theta B < 2\pi z$ within the entire blade row.

$$T_\theta = z|\theta_{upper} - \theta_{lower}| \quad (4.18) \qquad \theta_{subtended} = \arctan(\frac{y}{z}) \quad (4.19)$$

5

Development of the Unified Ducted Fan Code

This chapter describes the developed Unified Ducted fan Code (UDC), which builds upon the through-flow solver MTFLOW. This code integrates MTFLOW into a Python-based tool, expanding functionality by incorporating a (semi-) automated grid generation routine and adaptive exception handling. Section 5.1 first describes the motivation for the development of the UDC. Next, section 5.2 describes the developed grid generation interface, followed by a description of the numerical blade fields in section 5.3. Finally, section 5.4 describes the solving procedure developed and implemented in the UDC. The developed UDC code is contained in reference [58], while its implementation is further described in appendix C.1.

5.1. Motivation for Automation

Although MTFLOW is a flexible solver, it is heavily dependent on user inputs, such as the number of streamlines and streamwise grid points, the radial distribution of streamlines, the number of Newton iterations, the operating conditions $M_{inl}, Re_{inl}, n_{crit}$, and the input file containing the blade row forcing field data. Additionally, execution of the MTSOL Newton iterations requires manually handling any crashes and non-convergence of the solution.

These factors make the direct use of MTFLOW in any large-scale, automated analyses inefficient and potentially unstable. The UDC developed in this thesis addresses this by integrating the MTFLOW software into a Python-based (semi-)automated interface. This UDC must then perform the following functions to address the shortcomings of the isolated MTFLOW software:

- Generate the required input files based on the geometry parameterisation described in chapter 4.
- Automatically generate and check the computational grid used in the MTFLOW software.
- Detect MTSOL crashes and recover the flow solution from the last saved state.
- Detect non-converging solutions and handle them appropriately.
- Detect issues with the viscous flow convergence and handle them appropriately.

To create the UDC, the following modelling assumptions are made:

- All operating conditions analysed are steady state with axisymmetric flow, and no blade stage stall or surge.
- No choking occurs at the analysed operating conditions.
- Analyses which are not converging after 50 iterations are assumed to have oscillating residuals.
- The inlet Mach number M_{inl} in MTSOL is assumed equal to the freestream Mach number M_{∞}. This is true if there are no significant entropy losses and heat release specified at the inlet [37],

which is the case for the ducted fan modelling used in this thesis, where the domain inlet is positioned one reference length ahead of the ducted fan itself.

- The duct and centrebody geometries are assumed to be adiabatic.
- Modest incidence and swirl angles are assumed, such that the first-order blade loading model implemented in MTFLOW can be used.

5.2. Development of a Grid Generation Interface

To automate grid construction and create the grid validity checks, the MSET program is integrated into an automated interface in the UDC. MTSET offers a high degree of flexibility in creating the initial computational streamline grid. However, the default grid parameters can result in too coarse a grid being used. Additionally, MTSET does not verify the validity of a generated grid, nor does it have the capability to repair an invalid grid automatically. These shortcomings are addressed in the developed interface in this thesis. Section 5.2.1 first discusses the default grid parameters used to construct the computational grids in the UDC, followed by the grid construction procedure in section 5.2.2, including verification of the validity of the obtained grid. The grid repair implementation is discussed in section 5.2.3.

5.2.1. Definition of the Standard Computational Grid Parameters

The developed automated grid generation interface in this thesis uses a set of standard grid parameters, defined by table 5.1, which increases the resolution of the computational grid by:

1. Increasing the number of streamwise grid points to 200 from the default value of 141 to allow for a higher resolution of the blade forcing field at minimal computational cost penalties. This, as the cost scales linearly with the number of grid points [37].
2. Increasing the number of streamlines to the maximum value of 45 from the default value of 35 to yield an improved radial discretisation and accuracy of the optional viscous analysis in resolving the boundary layer(s).

Table 5.1: The grid parameters used to construct the computational grid in the unified ducted fan code.

Grid parameter	Set value	Repaired value
Number of streamwise grid points	200	200
X-spacing parameter	0.8	0.9
Exponent for airfoil side points	0.8	0.7
Number of streamlines	45	45

A further increase of the streamwise grid points to 400 was tested during validation. However, no meaningful improvement in accuracy was found. The streamlines are bunched towards the stagnation streamlines of the centrebody and duct, to yield a better resolution of the flowfield around the axisymmetric meridional surfaces and to avoid that spurious gains are modelled. An equal split between the internal and external segments of the computational domain is used for the streamlines.

The x-spacing parameter in MTSET controls the repelling force between the quasi-normal gridlines, with a larger value corresponding to a squarer grid. The exponent for airfoil side points controls the distribution of grid nodes between the axisymmetric meridional surfaces and the up- and downstream portions of the computational domain, where a smaller value corresponds to a larger fraction of the grid nodes being positioned on the axisymmetric meridional surfaces. These parameters are set to their default values to create a balanced grid, but are changed when repairing an invalid grid using the developed automated interface.

An additional automation implemented in this thesis is the construction of the domain size. This is done relative to the reference length L_{ref}. The inlet plane is positioned one reference length ahead of the most forward axisymmetric meridional surface leading edge, while the outlet plane is located one reference length aft of the most aft axisymmetric meridional surface trailing edge, as given by equation (5.1). These values are chosen based on tests in this thesis using the example cases in MTFLOW, to ensure the domain is large enough to capture the upstream and downstream influences

of the ducted fan.

$$x_{inl} = \frac{\min(x_{LE_{centrebody}}, x_{LE_{duct}}) - L_{ref}}{L_{ref}} \qquad x_{out} = \frac{\max(x_{TE_{centrebody}}, x_{TE_{duct}}) + L_{ref}}{L_{ref}} \tag{5.1}$$

The height of the computational domain is defined using the duct leading edge radial coordinate, as shown in equation (5.2). The factor of 2.5 is chosen to ensure that the streamwise disturbances on the outmost streamline are minimised, approximating the freestream condition.

$$z_{top} = z_{LE_{duct}} \frac{2.5}{L_{ref}} \tag{5.2}$$

5.2.2. Grid Construction and Verification

Using the default surface node distributions on the axisymmetric surfaces, approximately 40% of the streamwise grid points are located on the meridional surfaces, while the remainder are divided between the upstream and downstream sections of the domain. The grid parameters described in section 5.2.1 are then used to define the grid using MTSET's internal grid generation procedure [37].

This procedure uses an elliptic successive line over-relaxation grid-smoother to eliminate kinks and overlaps, ensuring that the grid streamlines correspond to the streamlines of an incompressible, inviscid flow solution. When executing MTSET, this smoother is executed by default only once. The developed automated interface in the UDC continues executing this step until the maximum change between consecutive grids satisfies $\Delta \leq 1 \times 10^{-6}$, or until 30 smoothing passes have been executed. This helps improve the convergence speed of the analyses, as the quality of the starting grid is improved. The upper iteration limit is enforced to prevent infinite loops in the event of an invalid grid and is chosen based on testing in this thesis.

The MTFLOW software does not include grid validity checks. Grid validity checking is therefore developed and implemented in the UDC by solving the inviscid axisymmetric flowfield without forcing fields for a low freestream Mach number ($M_\infty = 0.1$) flow. The resulting flowfield problem typically yields a converged solution within seconds, making it the fastest automated method to check grid validity.

5.2.3. Grid Repair

Invalid grids typically appear as a self-intersecting grid in the wake of the centrebody or the duct. An example of such a grid is shown on the left in figure 5.1, with the repaired grid shown on the right-hand side.

The required changes to repair such a grid are specific to each particular grid. Testing in this thesis shows that an invalid grid can be repaired by adjusting the x-spacing and exponent for airfoil side points parameters. This is implemented in the developed automated interface by increasing the x-spacing parameter to 0.9 and reducing the exponent for the airfoil side points to 0.7. This results in a smoother distribution of grid nodes along the leading and trailing edges. However, it cannot be guaranteed that this repaired grid is feasible. If the repaired grid remains infeasible, the user must manually adjust the grid parameters until an acceptable distribution is achieved.

5.3. Generation of the Blade Row Forcing Fields

Within MTFLOW, each blade row is defined on a discrete grid in a stage block. This grid has a maximum size of 16-by-16 (x, z) coordinate pairs, and must be logically rectangular. However, each stage can be given a different dimension. The MTFLO subprogram uses bicubic splines to interpolate and evaluate the stage blocks at all the internal grid nodes. This interpolation is sensitive to spline overshooting when small x-spacings are used in the input coordinates [37], [38]. Although airfoil profiles are typically represented using a cosine spacing with an increased number of data points at the leading and trailing edges, the stage blocks therefore use a constant spacing in x with ten data points in the developed UDC to avoid these overshoots. This spacing is based on applications of MTFLOW in open literature [37]–[39] and testing within this thesis.

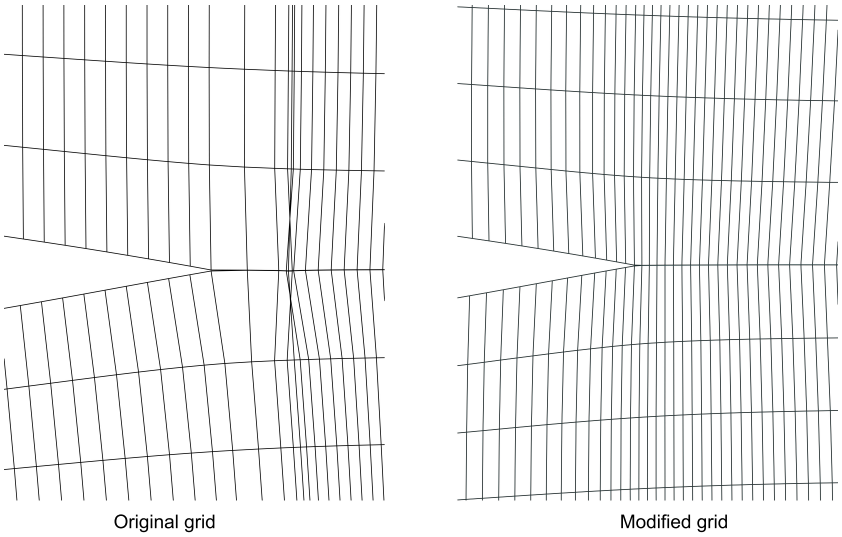


Figure 5.1: An infeasible grid and its repaired counterpart. The grid is repaired by decreasing the exponent for airfoil side points and increasing the x-spacing parameter.

The radial stations in the stage blocks in the UDC are defined to be equal to the radial sections used in the parameterisation. This gives direct control over the interpolation degree used in MTFLOW. When defining the blade row using $n \leq 4$ radial sections, MTFLOW uses a linear interpolation, whereas $n > 4$ results in a cubic spline interpolation with natural boundary conditions along the blade span being used [37].

5.4. Solving the Flowfield

The developed UDC solves the axisymmetric meridional flowfield in two distinct steps. First, an inviscid solve is performed. The initial streamline grid obtained from MTSET in the developed automated grid generation interface corresponds to an incompressible, inviscid flowfield. By first computing an inviscid solution, an updated streamline grid is obtained, which is a better starting point for the viscous iterations. Additionally, this enables the extraction of the inviscid flowfield properties. This may be desirable for rapid inviscid design optimisations, as the viscous iterations account for the majority of evaluation time in the UDC. The Newton solver in MTSOL has a built-in convergence target of 1×10^{-6} for all residuals. This is supplemented by an iteration limit of 50 iterations in the UDC implementation. After 50 iterations, the UDC assumes the solution will not converge any further, and non-convergence handling is applied. This iteration limit is based on testing performed in this thesis.

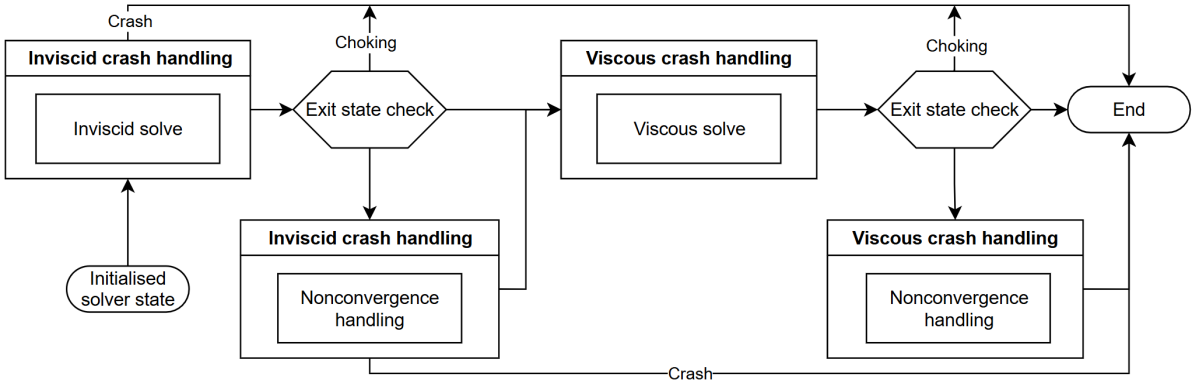


Figure 5.2: Flowchart visualising the progression from the initialised solver state through the inviscid and viscous iterations to the final solution, including the developed crash and non-convergence handling. The solving steps are contained within crash handlers to ensure appropriate crash detection and correction.

The developed non-convergence handling is discussed in section 5.4.1. The procedures used if MTFLOW crashes are discussed in section 5.4.2. This crash handling is developed in the UDC to increase robustness and enable automated analyses, as solution recovery is often possible for crashed analyses. A flow diagram showing the progression from the initialised solver state through the inviscid and viscous iterations to the final solution is shown in figure 5.2.

5.4.1. Development of Non-Convergence Handling

The developed non-convergence handling in the UDC enables reliable solutions to be obtained, even if the residuals do not converge far enough, as can be the case for low Mach numbers ($M_\infty < 0.1$) or significant flow separation. This is caused by the Newton matrix for the iterations being less well-conditioned [46]. At higher Mach numbers, testing in this thesis shows that non-convergence typically occurs in excessively narrow cells at the leading edge of the centrebody or duct.

These residuals typically oscillate in a repeating pattern, allowing an assumption to be made that this oscillation occurs around the true solution. The UDC implements temporal averaging over the last $n = 10$ iterations to obtain the results for a non-converging analysis. Since MTFLOW uses a binary state file for which the structure is unknown, this averaging approach is applied to the forces output file, described in section 3.5.7, as these are the quantities of interest in this thesis. Negligible changes are typically observed in the flowfield variables across these non-converging iterations, allowing for this approach to be used.

5.4.2. Development of Crash Handling

The MTFLOW software does not contain any crash handling. To enable automated large-scale analyses, crash handling is therefore developed in the UDC, where two classifications of crashes are made: inviscid and viscous. Inviscid crashes occur when the flowfield, due to unphysical forcing fields or axisymmetric geometries, experiences a curling of the streamlines, resulting in numerical singularities. Such an analysis cannot be recovered or repaired. As such, if an inviscid crash is detected, the developed UDC exits and returns zero for all outputs.

Viscous crashes originate in either the centrebody viscous wake collapsing or the curling of the trailing duct streamline. A centrebody viscous wake collapse typically occurs when excessive swirl is present near the centreline of the ducted fan, caused by an incorrect forcing field. This causes abrupt curvature of the bounding streamtube, yielding local Mach number spikes. Duct wake curling occurs when there is an unphysical pressure difference across the centreline of the duct wake, as a result of an intermediary Newton step, which causes the centreline to curl into itself. Both of these crashes can typically be countered by sequentially solving the axisymmetric meridional boundary layers in three steps:

1. Solve the flowfield, including the centrebody boundary layer, using the inviscid flowfield as a starting guess for the solution.
2. Solve the flowfield, including the centrebody and outer duct boundary layers, using the flowfield including the centrebody boundary layer as a starting guess for the solution.
3. Solve the flowfield, including all boundary layers, using the flowfield including the centrebody and outer duct boundary layers as a starting guess for the solution.

This method is particularly effective at countering duct wake curling, as the sequential solving strategy avoids the occurrence of the unphysical pressure differences. Within the UDC, these three steps are implemented as non-exclusive sequential steps. Should any of these three steps fail, the previously stored converged solution state is used to start the next step. This means that if the centrebody boundary layer individually also causes a crash, it is subsequently excluded from the solution, and the final solution may, at most, contain only the converged inner and outer duct boundary layers. An expanded method, where the failed axisymmetric boundary layers are retried after the initial convergence loop, was tested but showed no meaningful improvements in robustness while resulting in a significantly increased computational cost. Although the method described in this subsection is more robust compared to converging all boundary layers simultaneously, it is considerably slower due to its sequential nature. It is therefore not used as the default solving approach in the UDC.

6

The Ducted Fan Optimisation Framework

This chapter describes the developed ducted fan optimisation framework in this thesis, which uses the UDC developed in chapter 5. Section 6.1 first describes the different steps which define U-NSGA-III, the used optimisation algorithm. Sections 6.2 and 6.3 then describe the reference directions and initial population used in the initialisation of the algorithm in the framework, while 6.4 describes the developed repair operator. The objective functions and constraints implemented in the framework are discussed in sections 6.5 and 6.6. Finally, the termination conditions are discussed in section 6.7. The developed optimisation framework is available in [58], and is further described in appendix C.2.

6.1. The Unified Non-dominated Sorting Genetic Algorithm III

As described in section 3.6, U-NSGA-III is used in the developed optimisation framework. This section gives a description of the optimisation procedure using U-NSGA-III. A generalised M-objective genetic algorithm ducted fan design optimisation problem with population size K, subject to J inequality constraints and having a design vector $\vec{\xi}$ of n elements, can be defined as equation (6.1) [59]. The design vector is bounded by the bound vectors $\vec{u}, \vec{\ell}$. Note that no equality constraints $h_k(\vec{\xi})$ are defined. This is a common approach in genetic algorithms, where the equality constraints are transformed to inequality constraints as $g_j(\vec{\xi}) = |h_k(\vec{\xi})| - \epsilon \leq 0$, where ϵ is a user-defined tolerance [60].

$$\begin{array}{lll} \min & f_m(\vec{\xi}) & m = 1, \ldots, M \\ \text{s.t.} & g_j(\vec{\xi}) \leq 0 & j = 1, \ldots, J \\ & \ell_i \leq \xi_i \leq u_i & i = 1, \ldots, n \end{array} \tag{6.1}$$

The solution process using U-NSGA-III is visualised in figure 6.1. The algorithm is initialised using a set of n reference directions on a unit simplex and an initial population of design vectors. The objective functions and constraint values are then computed for the initial population. At generation i, a subset of the population is chosen as parents for the next generation $i + 1$. This subset is selected through a niching-based selection operator, which considers two distinct options:

1. Two solutions corresponding to two different reference directions: one solution is chosen at random, to preserve multiple niches in the child generation $i + 1$.
2. Two solutions corresponding to the same reference directions: the solution closer to the reference direction is chosen.

If one of the two parent solutions being considered is infeasible, the traditional selection mechanism of NSGA-III is used, where the least-represented reference directions are maintained. This selection procedure is repeated $K/2$ times on generation i to form $K/2$ parent individuals. By repeating this procedure one more time on a shuffled population, a total of K parent individuals can be defined.

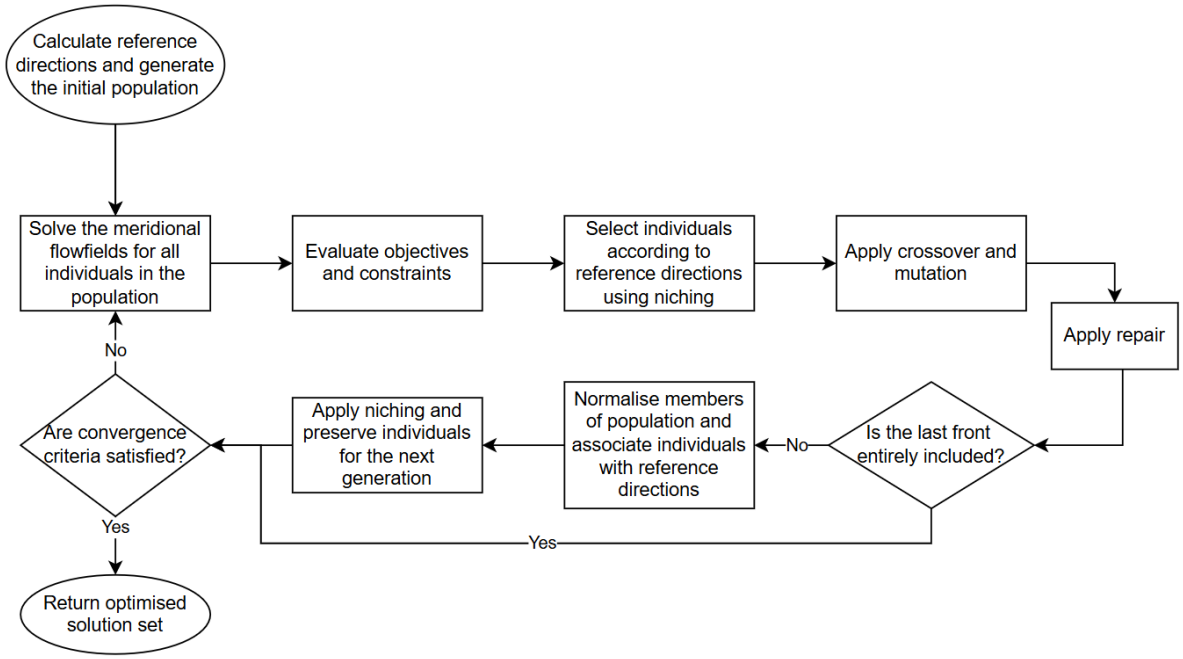


Figure 6.1: Flowchart of the optimisation framework for a ducted fan design optimisation problem using U-NSGA-III.

These K parent individuals are then used to create the offspring population for generation $i+1$ using the crossover and mutation operators of the genetic algorithm. The mutation and crossover operators differ for continuous and discrete design variables. In this thesis, the Pymoo framework [52] is used, which employs Polynomial Mutation (PM) and Simulated Binary Crossover (SBX) operators for continuous design variables, defined by Deb, Sindhya and Okabe [61].

In these operators, the design variables are represented in binary form, and an exponential probability distribution is used to simulate crossover or mutation. Application of PM and SBX to the discrete design variables is achieved by applying a rounding to the nearest integer after application of the PM/SBX operator(s). Should the variable be exactly halfway between two integers, a rounding is applied to the nearest even integer. These discrete variable operators are available as a custom operator in the Pymoo framework [52].

An additional step in the standard U-NSGA-III process is introduced in the developed optimisation framework of this thesis in the form of a repair operator. Due to the guided random search nature of a genetic algorithm, it is not possible to guarantee a priori that a design vector is physical. As such, each population of design vectors may contain a large number of unphysical design vectors. Examples include invalid BP3434 parameterisations, a fan positioned ahead of the leading edge of the duct, or a fan which experiences complete blockage at some x, z coordinate. Evaluating such designs, if possible, is meaningless for the evolution of the genetic algorithm and hinders convergence due to their unphysical objective values. As such, a repair operator is developed to correct these unphysical design vectors.

The repaired population is then subjected to non-dominated sorting to preserve the non-dominated fronts in generation $i+1$. If the selected set of the repaired population is smaller than the required population size, a further niching strategy is used to select additional individuals in generation i. Once the complete next population is obtained, the solution process restarts until the termination criteria are met.

6.2. Generating the Reference Directions

The reference directions help to maintain diversity and a well-distributed Pareto front [62]. Most applications of reference directions in genetic algorithms use the structured Das and Dennis [62] method. Since this method is highly structured, it cannot produce an arbitrary number of points. Such flexibility

is desirable in the framework, as it enables different population sizes to be defined as inputs, rather than relying on the algorithm to determine population sizes based on the number of objectives using the default approach of U-NSGA-III [50]. A flexible number of reference directions gives more freedom in the population size, and avoids over- or under-sampling of the objective space.

Additionally, the total number of points on the unit simplex increases rapidly with the number of objectives in the Das and Dennis [62] method, with most points lying on the boundary of the simplex rather than its interior. A uniformly and evenly distributed distribution of reference points is desired to avoid the creation of sparse or crowded areas in the objective space [50]. To address this, a recent procedure by Blank, Deb, Dhebar *et al.* [63] proposes the use of a generalised Riesz potential energy concept [64] to obtain a well-spaced distribution. This Riesz s-energy method is used in the optimisation framework to compute the reference directions for use within the U-NSGA-III algorithm, and is further described in appendix B.

The total number of points on the unit simplex is defined using the binomial coefficient defined by equation (6.2) [62], where e controls the total number of reference directions. The total number of reference directions is taken to be the closest integer to the population size K, such that one population member can be expected to be found for each reference direction. This corresponds to choosing $e = \arg\max_e e \mid n \leq K$.

$$n = \begin{pmatrix} M + e - 1 \\ e \end{pmatrix} \tag{6.2}$$

6.3. Biased Initialisation of the Initial Population

To generate the initial population for the optimisation algorithm in the developed framework, biased initialisation is used. A known design vector $\vec{\xi_{ref}} = [\xi_{ref_1}, \ldots, \xi_{ref_n}] \in \mathbb{R}^n$ is required for this procedure to work, where random perturbations are then used to construct a biased population centred around this reference design. The design variables are bounded by the bound-vectors $\vec{\ell} = [\ell_1, \ldots, \ell_n]$ and $\vec{u} = [u_1, \ldots, u_n]$, with corresponding span $\psi_i = u_i - \ell_i$.

Since the design vector contains both continuous and discrete variables, Boolean masks H for the real and integer variables are introduced as:

$$H_j^R = \begin{cases} 1, & \text{if variable } j \text{ is real,} \\ 0, & \text{otherwise,} \end{cases} \qquad H_j^I = \begin{cases} 1, & \text{if variable } j \text{ is integer,} \\ 0, & \text{otherwise.} \end{cases}$$

An additional mask is introduced for the continuous reference design variables equal to zero, since any perturbations otherwise applied using the below methods would have no effect, as:

$$H_j^0 = H_j^R \, \mathbf{1}_{\{\xi_j = 0\}}, \quad H_j^{\neq 0} = H_j^R \, \mathbf{1}_{\{\xi_j \neq 0\}}$$

This step is not required for the integer design variables, since only the blade count is integer and is logically non-zero. By further letting ν be the continuous spread as a fraction of the reference value and τ the fraction of the design variable span ψ, a perturbed design vector $\widetilde{\xi^i}$ can be defined as equation (6.3).

$$\widetilde{\xi_j^i} = \xi_{ref_j} + H_j^{\neq 0} \epsilon_j \xi_{ref_j} \nu + H_j^0 \epsilon_j \psi_j \tau \nu + H_j^I \delta \quad \text{for } j \in 0, \ldots, n \tag{6.3}$$

In this equation, ϵ is a continuous uniform noise distribution and δ is a discrete uniform noise distribution, defined by equation (6.4), where a and b represent the range of additions to the blade count. To ensure both higher and lower blade count individuals compared to the reference exist in the initial population, it is required that $a < 0$ and $b > 0$, provided that the blade count in $\vec{\xi_{ref}}$ is not equal to one of the design variable bounds. To obtain the complete initial population, the perturbed design vectors from equation (6.3) must be clipped within the bound vectors $\vec{\ell}, \vec{u}$ using equation (6.5). The initial population can then be constructed by letting individual 1 be equal to the reference design, and individuals $2, \ldots, K$ equal to clipped perturbed individuals.

$$\varepsilon \sim \mathcal{U}([-1, 1]^n), \qquad \delta \sim \mathcal{U}_{\mathbb{Z}}(a, b) \tag{6.4}$$

$$\xi_j^i = \min(\max(\widetilde{\xi_j^i}, \ell_j), u_j) \tag{6.5}$$

6.4. Development of the Repair Operator

Within the developed optimisation framework, a repair operator is developed and applied to ensure that the evaluated design vectors are physical. The framework implements five repair operations, which are discussed in order of execution below. After each repair operation, the changed design variables are clipped back into the bounded design space using equation (6.5).

6.4.1. Bézier-Parsec Parameterisation Repair

The design variables for the BP3434 parameterisations of the different geometries are not guaranteed to yield thickness and camber distributions which are one-to-one and satisfy equation (4.5). The latter is resolved by replacing b_8 in the design vector with a mapping variable, $\breve{b_8} \in [0, 1]$, such that b_8 is defined by equation (6.6).

$$b_8 = \breve{b_8}\sqrt{-2r_{LE}x_t/3} \tag{6.6}$$

To enforce a one-to-one curve for the thickness and camber distributions, it is a sufficient condition to enforce that the control points for each of the curves are sorted. This is not a required condition, since non-sorted control points may still yield a one-to-one curve, as reported by Rogalsky [53]. However, there is no direct link between the control points in table 4.1 and 4.2 and the one-to-one condition of the curve if the control points are allowed to be non-ordered. Forcing the control points to be one-to-one offers a direct link between the curve and the control points, enabling a repair operation to be performed.

A restriction is placed on the design variables b_{15} and b_{17}, which control the inflexion points on the trailing edge thickness and camber curves. Only requiring that $x_{2_t} < b_{15} < 1$ and $x_{2_c} < b_{17} < 1$ can result in sharp kinks in the corresponding distributions, yielding infeasible profile shapes. It is therefore required that $x_{2_t} + \epsilon < b_{15} < 1$ and $x_{2_c} + \epsilon < b_{17} < 1$, where the offset is taken as $\epsilon = 0.05$ based on testing in this thesis.

The repair then enforces the control points to be ordered such that $x_0 < x_1 < x_2 < x_3 < x_4$ and $y_0 < y_1 < y_2 < y_3 < y_4$ for both the leading-edge and trailing-edge curves by adjusting the design variables $\breve{b_8}, b_{15}, \beta_{TE}$ for the thickness distributions, and $b_0, b_2, b_{17}, y_c, \gamma_{LE}, \alpha_{TE}$ for the camber distributions. The control points are then iteratively adjusted until all Bézier curves are one-to-one.

6.4.2. Blade Sweep, Chord Distribution, and Duct Location Repair

The developed optimisation framework further defines a repair operator for the sweep angles, ϕ_i, in equation (6.7) to enforce that the leading edge x-coordinates, x_{LE_i}, are monotonically increasing along the blade span, correcting unrealistic leading edge x-coordinate distributions. The repaired leading edge x-coordinates $\tilde{x}_{\mathsf{LE},i}$ are then defined through equation (6.8).

$$\phi_i = \tan^{-1}\left(\frac{\tilde{x}_{\mathsf{LE},i}}{z_i}\right), \quad i \geq 2 \tag{6.7}$$

$$\tilde{x}_{\mathsf{LE},i} = \max_{j \leq i} x_{\mathsf{LE},j} \tag{6.8}$$

A similar repair is defined for the chord length distribution c_i along the blade span, which forces the outboard section of the fan blades to have a monotonically decreasing chord length, as given by equation (6.9). This repair is applied to avoid structural and loading issues for the blades. An additional repair operator is used for the axial location of the leading edge of the duct, such that $x_{LE_{duct}} \leq x_{root}$. This ensures that the duct encloses the fan, which is a logical requirement for a ducted fan.

$$c_i = \min_{j \leq i_{\mathsf{mid}}} c_j, \quad \forall i \geq i_{\mathsf{mid}}. \tag{6.9}$$

6.4.3. Blockage Repair

The final repair operation used in the repair operator is a blockage repair. As described in section 4.4.4, the blade circumferential thickness must satisfy $T_\theta < 2\pi z/B$ at every location z along the blade span. The local blockage can therefore be adjusted by either modifying the blade shape and blade angle distribution to adjust the circumferential blade thickness T_θ or by reducing the number of blades B. From a computational perspective, reducing the blade count is preferred, as it avoids the required

additional profile repair loops when iteratively changing the blade shape. As such, the blade count is iteratively reduced until complete blockage no longer occurs anywhere along the blade span.

6.5. Objective Functions

To perform optimisations of a ducted fan design, three objective functions are identified in this thesis and implemented in the developed optimisation framework. These objectives are presented in table 6.1. As the genetic algorithm is designed to minimise objective functions, the maximisation of the propulsor efficiency in f_1 is achieved by minimising the negative value, in line with the procedure used by Seada and Deb [50].

Table 6.1: Defined objective functions implemented in the genetic algorithm.

Objective identifier	Definition
f_1	$-\eta_p$
f_2	$A_{front}/A_{front_{ref}}$
f_3	$\sum_{i=0}^{n} E_{phase_i} / \sum_{i=0}^{n} E_{ref_{phase_i}}$

The frontal area objective, f_2, is considered in the optimisation framework to enable a trade-off between performance and size of the ducted fan in a multi-objective optimisation. The frontal area is straightforwardly computed from the design variables and normalised by the reference value obtained from the reference design vector $\vec{\xi_{ref}}$. This ensures a similar magnitude to the other objectives so that no unintended weighting is introduced in a multi-objective optimisation.

To compute the normalised total required energy over a flight (phase) for a multi-point optimisation, f_3, the UDC is used at a representative mean condition of the flight phase, such as the mid-cruise condition. The atmospheric properties from the 1993 ICAO international standard atmosphere [65] are then used to compute the power from the power coefficient P_C output by the UDC, from which the energy is obtained as $E_{phase_i} = P_{phase_i} \cdot t_{phase_i}$. This objective is equivalent to the propulsor efficiency for a single-point analysis at constant thrust, but enables a multi-point, single-objective analysis across multiple operating conditions.

6.6. Constraints

Several constraints are defined for the developed optimisation framework, which are discussed in this section. Within the developed UDC, MTFLOW imposes no limitations on the nature of the analysis performed, allowing propulsor efficiencies $\eta_p < 0$ or $\eta_p > 1$ to be obtained. An efficiency greater than one can occur when unphysical inputs are given in the blade forcing fields. To analyse meaningful designs, a constraint is enforced on the upper bound of the propulsor efficiency, ensuring it remains lower than or equal to the theoretical propulsive efficiency in the absence of any losses. This results in the constraint defined in equation (6.10) [66], where the theoretical propulsive efficiency is obtained using the dimensionalised thrust T and mass flow $\dot{m}$ based on the UDC outputs and ambient conditions. A physical design will always ensure that this constraint is ≤ 0, as the propulsor efficiency can never exceed the propulsive efficiency due to the additional losses considered.

A negative efficiency occurs if the ducted fan has a negative thrust. Such conditions are automatically handled by using a thrust constraint that enforces that the thrust lies within ϵ of the reference design. This ensures that the optimised design can still perform the intended mission. This constraint is given by equation (6.11). Finally, an optional constraint is identified to limit the frontal area, given by equation (6.12), where $\sigma \geq 1$ is a user-defined factor to restrict the resulting frontal area of the ducted fan. This can be used to represent the installation requirements of the ducted fan onto the aircraft.

$$\eta_p - \frac{2}{2 + \frac{T}{\dot{m} v_\infty}} \leq 0 \quad (6.10) \qquad |\frac{T}{T_{ref}} - 1| - \epsilon \leq 0 \quad (6.11) \qquad \frac{A_{front}}{A_{front_{ref}}} - \sigma \leq 0 \quad (6.12)$$

The constraints are implemented in the optimisation framework using the feasibility-first approach de-

scribed by Deb [67]. In this approach, a penalty term is used to compare infeasible solutions based solely on their constraint violation values, as defined by equation (6.13). In this equation, f_{max} is the maximum function value of all the feasible individuals in the current population.

$$F(\vec{\xi}) = \begin{cases} f(\vec{\xi}) & \text{if } g_j(\vec{\xi}) \leq 0, \forall j \in J \\ f_{max} + \sum_{j=1}^{J} g_j(\vec{\xi}) & \text{otherwise} \end{cases} \tag{6.13}$$

6.7. Termination Conditions

Within the developed optimisation framework, equal termination conditions are used for both single-objective and multi-objective optimisations, as shown in table 6.2. An objective space tolerance of 1×10^{-5} is used to match the accuracy of the MTFLOW outputs in the developed UDC. To limit computational cost, a maximum number of generations is defined as $n_{gen_{max}} = 100$.

Due to the lack of gradient information in the genetic algorithm, the convergence history of the algorithm must be taken into account when defining the termination conditions. This is done by considering the objective space termination condition over a sliding window of $\omega = 15$ generations. This means that the earliest termination opportunity of the algorithm occurs after fifteen generations. This sliding window size is based on testing in this thesis. No termination condition for the design space has been defined, as the nature of the algorithm considered effectively guarantees that the objective space is converged before the design space is. Testing was conducted on all optimisation problems considered in this thesis, which confirmed this.

Table 6.2: Termination conditions for the single- and multi-objective optimisation problems.

Termination condition	Value
Sliding window size ω	15
Maximum change between subsequent objective values	1×10^{-5}
Maximum number of generations $n_{gen_{max}}$	100

For single-objective optimisation, the objective value of the optimal individual is used in the termination condition. For a multi-objective optimisation, the running metric proposed by Blank and Deb [68] is used to determine if the objective space has converged. Unlike other methods for assessing convergence in a multi-objective problem, which typically rely on knowledge of the true Pareto front, this metric compares the non-dominated solution set of generation i with that of generation $i+1$ to obtain a running Inverted Generational Distance (IGD) metric. The IGD is the inverse measure of the distance between the solution set Z and a defined reference solution set A, defined by equation (6.14) [52], [69]. This reference set is taken here as the previous generation. In this equation, $\hat{d}_i$ represents the Euclidean distance, such that $p = 2$, from z_i to the nearest reference point in A.

$$IGD(A) = \frac{1}{|Z|} \left(\sum_{i=1}^{|Z|} \hat{d}_i^p \right)^{1/p} \tag{6.14}$$

An individual ξ^* in a population $\mathcal{F}$ for an M-dimensional optimisation problem is non-dominated if there does not exist another $\xi \in \mathcal{F}$ such that $f_i(\xi) \leq f_i(\xi^*)$ for all $i \in \{1, \ldots, m\}$ and $f_j(\xi) < f_j(\xi^*)$ for at least one $j \in \{1, \ldots, m\}$ [70]. By then estimating the convergence of the extreme points of the consecutive generations and comparing the running IGD metric with the specified threshold, a robust and systematic termination criterion is obtained. This method has the advantage of enabling analysis of the convergence behaviour of the problem.

7

Verification and Validation

This chapter describes the verification and validation steps taken in this thesis. Section 7.1 first describes the verification steps undertaken for the integrations of the MTSET, MTSOL, and MTFLO programs within the UDC. Next, section 7.2 describes the validation undertaken of the developed UDC against wind tunnel test data of the Bell X-22A propulsor.

7.1. Verification of the Multi-Passage ThroughFLOW Code Integrations

Verification in this thesis was conducted by verifying the individual Python interfaces for the MTSET, MTSOL, and MTFLO programs within the developed UDC. These interfaces control both the normal flow of their respective programs and handle local errors and exceptions. They are therefore tested to ensure they function as expected using the example geometry files provided with MTFLOW. Each sub-interface was executed, and the outputs were compared against those obtained from manually executing the corresponding program to ensure they were identical. The example files were adjusted to ensure they contained the following specific errors, triggering each error-handling routine:

- ADJ1: Adjust the duct lower surface coordinates so that they are no longer one-to-one. This enables verification of the developed MTSET error handling, as it causes an infinite loop in the elliptic grid smoothing. Additionally, by creating a sufficiently extreme duct lower surface, an inviscid MTSOL crash can be caused, allowing for verification of the inviscid crash handling implementation.
- ADJ2: Increase the trailing edge thickness of the centrebody to check the developed MTSOL viscous crash handling. Since MTFLOW builds on MSES, it cannot handle trailing edge thicknesses larger than a few percent of the chord [37], [45]. Larger trailing edge thicknesses will cause the streamlines to curl at the trailing edge of the centrebody, resulting in local singularities in the flowfield, crashing the solver
- ADJ3: Reverse the rotation direction of the blade row to flip the sign of the swirl generation along the streamlines. This yields a non-converging flowfield, enabling verification of the non-convergence handling.

7.2. Validation of the Unified Ducted Fan Code

Although MTFLOW has been previously validated for marine propulsion applications [38], [39], the UDC requires further validation as the blade row modelling has not been previously validated. As such, this section presents the validation exercise undertaken in this thesis to validate the UDC, which integrates MTFLOW.

Validation is performed only for low freestream Mach number flows, where wind tunnel test data from the Bell X-22A ducted propeller, analysed by Mort and Gamse [71], is used due to the availability of detailed

geometry and test data. Future work can expand the performed validation using high-fidelity CFD. The X-22A ducted propeller and the modelling steps taken in the UDC are described in section 7.2.1, followed by the analysed operating conditions in section 7.2.2. The modelling approach used in DFDC is described in section 7.2.3. The obtained validation results are described in section 7.2.4.

7.2.1. The Bell X-22A Ducted Propeller

The X-22A was a flight research aircraft featuring a dual tandem ducted propeller configuration [72], as shown in figure 7.1. Development of the aircraft started in 1962, with test flights continuing through 1984. Each ducted propeller featured a three-bladed variable-pitch propeller and an elevon for further control. The duct was connected to the centrebody using two diagonal struts, a vertical strut, and a horizontal power strut, to which the elevon was connected.

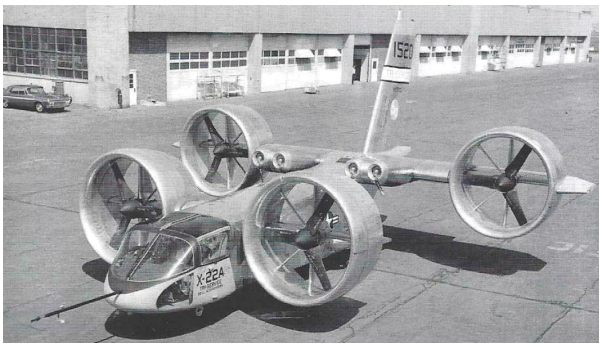


Figure 7.1: The Bell X-22A in 1965, showing the propeller and strut arrangement. Taken from Miller [72].

Mort and Gamse [71] document the geometry of the centrebody, duct, struts, and the radial distributions of the chord length c, thickness to chord ratio y_t/c, and blade angle β_{geom}, but give no specification of the blade profile shape distribution. While Achton [73] reports the use of a modified NASA 0010-64 blade profile, no further specifics for the radial profile distribution are given. Jiang, Liu and Zheng [74] use a NACA $24XX$ profile distribution along the blade span, where the thickness distribution is given by Mort and Gamse [71]. This same blade profile distribution is used in this thesis to maintain consistency with previous publications. The blade is modelled using nine radial sections at $z_i/R = 0.2, 0.3, \ldots, 1.0$. The radial distributions given by Mort and Gamse [71] are then used to construct the blade with a zero-sweep midchord line using the defined radial sections.

Due to the limitations of MTFLOW in handling trailing edge thickness, the reported centrebody was modified using an extended smoothing spline with $c_{centrebody} = 1.68\ \mathrm{m}$, compared to $c_{centrebody} = 1.02\ \mathrm{m}$ for the original centrebody, to yield a feasible trailing edge thickness. This smoothed extrapolated profile is shown in figure 7.2. The diagonal struts model utilises a NACA 0040 profile, whereas the horizontal strut representation employs a NACA 0024 profile to achieve the correct thicknesses [71].

The duct profile of the X-22A has negative camber and can therefore not be modelled using the BP3434 parameterisation method. During validation, the exact geometry of the X-22A duct and the smoothed spline centrebody are used, bypassing the relevant steps in the UDC that use the BP3434 parameterisation. This yields the most accurate total propulsor geometry possible for validation.

The vertical strut was modelled as equal to the horizontal strut to simplify the modelling effort. In reality, the vertical strut consists of a vane and hinge, to which the elevon is attached. However, validation is performed against the elevon-out data. Removing the exit vane results in a high geometric similarity between the horizontal and vertical struts, allowing for this assumption to be made. The complete input geometry data set for the X-22A, as used by and generated in the UDC, is included in appendix D.

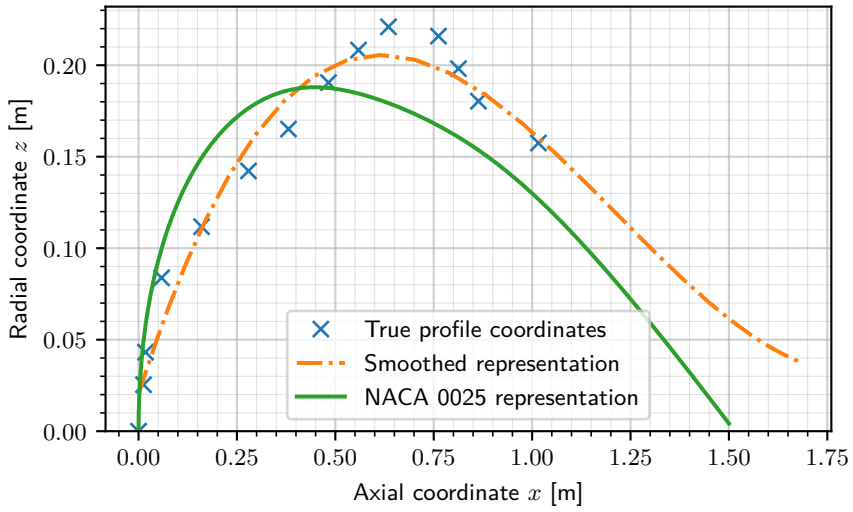


Figure 7.2: Comparison of the true X-22A centrebody geometry with the smoothed spline representation and a NACA 0025 airfoil used for the optimisations in this thesis.

7.2.2. Analysed Operating Conditions

Mort and Gamse [71] report elevon-out thrust and power coefficients at tip set angles $\beta_{tip} = 14.5°$ and $\beta_{tip} = 24.5°$ obtained using rotational rates $\Omega \in [20, 44]\ \mathrm{s}^{-1}$. Since MTFLOW and thus the UDC cannot be run with a zero-Mach number condition [37], data was compared for $J \in [0.3, 0.6]$. Since Mort and Gamse [71] do not report the operating conditions used, a constant freestream velocity of $v_\infty = 26\ \mathrm{m\,s}^{-1}$ was used to ensure the rotational rate of the analysed conditions remained within the tested range. The standard computational grid described in section 5.2.1 was used, where analyses were performed at standard sea-level conditions with a critical amplification factor of $n_{crit} = 9$.

7.2.3. Ducted Fan Design Code Modelling

Since DFDC is commonly used in open literature, it is added to the validation exercise to allow for a comparison between the accuracy of the UDC and DFDC. The case files used in DFDC are included in appendix D.2 for reference, where only the propeller is modelled for simplicity. The aerodynamic properties of the blade row are defined at three radial sections: $z_i/R = 0, 0.4, 1$, using NACA 2435, 2421, and 2408 profiles, respectively. These profiles are based on the radial thickness distribution given by Mort and Gamse [71]. The required rotor aerodynamic parameters are obtained from XFOIL analyses at $M = 0.0764, Re = 2 \times 10^6, n_{crit} = 9$, with the stall blade section parameters in DFDC left at their default values. A constant lift curve slope of $C_{l_\alpha} = 2\pi\ \mathrm{rad}^{-1}$ is used to remain consistent with the assumptions made in section 4.4.

7.2.4. Comparison of the Unified Ducted Fan Code with Wind Tunnel Data and Ducted Fan Design Code Predictions

The obtained UDC, wind tunnel, and DFDC trends for the thrust coefficient, C_T, and power coefficient, C_P, are shown in figure 7.3, where the UDC was unable to obtain solutions for $J < 0.4$ at $\beta_{tip} = 24.5°$. The experimental data have a read-off uncertainty of $\Delta C_T = \Delta C_P = \Delta J = \pm 0.01$, as indicated by the error bars in the figures, as they were read from digitised graphs.

As shown in figure 7.3a, both the UDC and DFDC achieve good fits between the thrust coefficient and the experimental data for $\beta_{tip} = 14.5°$. By defining a deviation as equation (7.1), it is seen that the UDC has an average thrust coefficient which is smaller than the read-off uncertainty for both tip angles, as shown in table 7.1. This is the result of the increased accuracy of the throughflow method and the blade loading model used. The UDC is at least $3.5\times$ as accurate as DFDC in both the average and maximum thrust coefficient deviations across the considered tip angles and advance ratios.

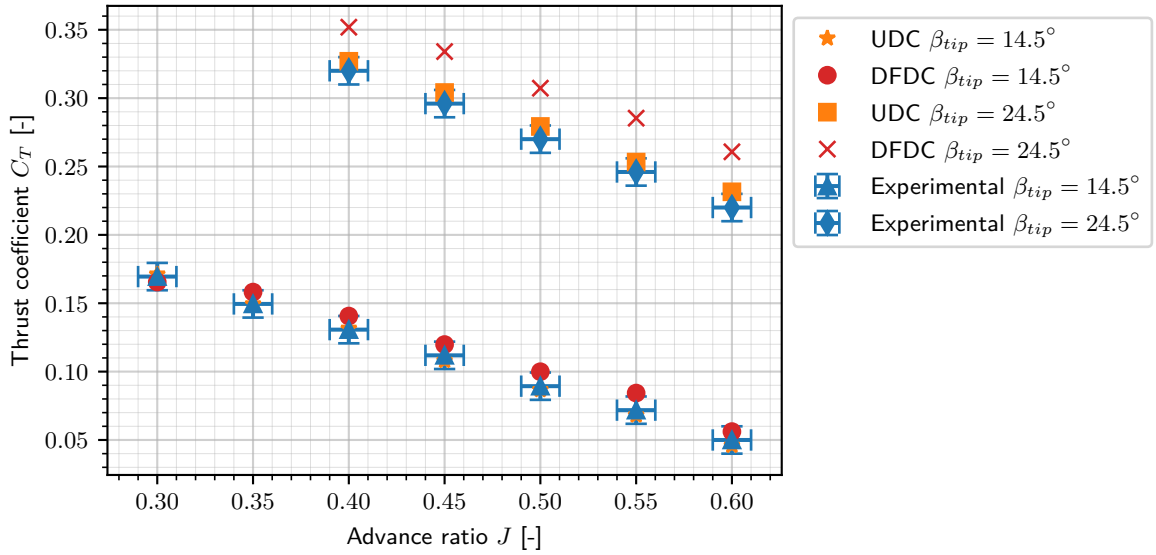
$$\varepsilon = \frac{|C_{exp} - C_{UDC,DFDC}|}{|\Delta C|} \tag{7.1}$$

Table 7.1: Accuracy of the UDC and DFDC compared to the experimental data.

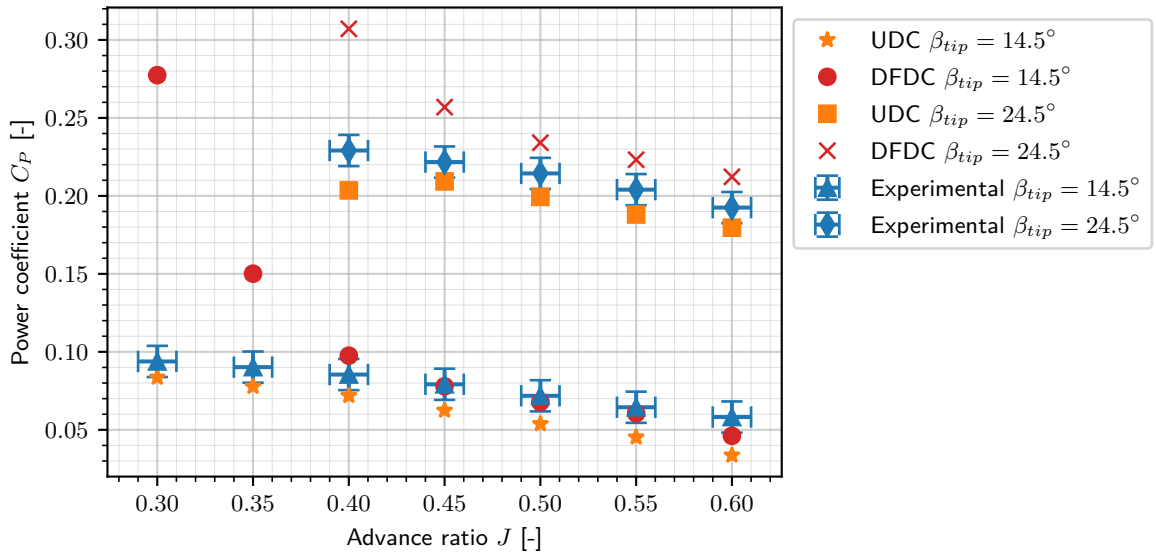
Parameter	$\beta_{tip} = 14.5°$		$\beta_{tip} = 24.5°$	
	UDC	DFDC	UDC	DFDC
$\varepsilon_{C_T,avg}$	0.24	0.85	0.86	3.75
$\varepsilon_{C_T,max}$	0.40	1.25	1.15	4.08
$\varepsilon_{C_P,avg}$	1.64	3.95	1.64	3.44
$\varepsilon_{C_P,max}$	2.54	18.36	2.56	7.80

The accuracy of the UDC is further illustrated when analysing the power coefficients in figure 7.3b and table 7.1. DFDC is unable to represent the trends in power coefficient accurately and diverges at lower advance ratios. This is reflected in the high average and maximum deviations. On the other hand, the UDC can represent the correct trend and yields a close match with the experimental data, with average and maximum deviations within $3\Delta C_P$. At $J = 0.4$ and $\beta_{tip} = 24.5°$, an increased deviation is observed, caused by a crash of the inner duct boundary layer. The general underprediction of the power coefficient by the UDC is attributed to the changed centrebody geometry, which does not suffer from the substantial wake formation seen for the actual geometry in the wind tunnel. Additionally, the lack of blade row loss modelling and the use of the first-order blade load modelling method further contribute to this underprediction.

The significantly higher accuracy and consistency of the UDC across both tip set angles seen in table 7.1, is a strong argument in favour of using the UDC over panel-based methods like DFDC, even for lower Mach number conditions like analysed here. The accuracy of the UDC aligns with validation results obtained by Renick [38] and Hanson [39], thereby validating the UDC for use in this thesis. The lack of available validation data for higher Mach number flows is recommended to be addressed in future work using high-fidelity CFD.



(a) Comparison of the thrust coefficient C_T between wind tunnel data, the developed UDC, and DFDC at $M_\infty = 0.0764$, $Re_\infty = 3.798 \times 10^6$, $n_{crit} = 9$.



(b) Comparison of the power coefficient C_P between wind tunnel data, the developed UDC, and DFDC at $M_\infty = 0.0764$, $Re_\infty = 3.798 \times 10^6$, $n_{crit} = 9$.

Figure 7.3: Comparison of the thrust and power coefficients obtained from wind tunnel data, the UDC, and DFDC.

8

Setup of the X-22A Propulsor Design Optimisations

To demonstrate the capabilities and robustness of the developed UDC implemented in the created optimisation framework, the Bell X-22A ducted propulsor is used in this thesis for its detailed geometrical data, and abundance of developmental and operational documentation available for the aircraft and propulsor [71], [73]–[76]. This chapter describes the setup of the design optimisations performed in this thesis as part of this demonstration. Section 8.1 discusses the modelling of the axisymmetric meridional geometry used to define the reference design vector. This is followed by a description of the considered operating conditions and a simplified flight profile in section 8.2. Section 8.3 presents the different formal optimisation problems considered, while the bounds on the design variables are presented in section 8.4.

8.1. Modelling of the X-22A Axisymmetric Meridional Geometry

The X-22A meridional geometry used in chapter 7 cannot be represented using the BP3434 method. To address this, the centrebody is modelled using a NACA 0025 profile with $c_{centrebody} = 1.5\ \mathrm{m}$. A comparison of the true centrebody geometry, the smoothed spline used during validation, and the NACA 0025 profile used in the optimisations was shown in figure 7.2. The different trailing edge shape improves the numerical stability of the optimisations, as significant centrebody flow separation is avoided. Since the centrebody is kept constant throughout this thesis, the impact of this change on the performance of the optimised designs is limited and does not impact the outcomes of the optimisations, which focus on demonstrating the applicability of the developed tools in this thesis.

To model the X-22A duct in the initial design vector, $\vec{\xi_{ref}}$, a Sequential Least-Squares Quadratic Programming (SLSQP) minimisation of the ℓ_2 error of the BP3434 profile representation to the true duct shape is performed, as the duct has negative camber. The resulting BP3434 duct geometry is shown in figure 8.1. All further mentions of the reference design in this thesis, including thrust and power levels and associated propulsor efficiencies, correspond to those obtained using the reference axisymmetric meridional geometry defined in this section, together with the defined rotor and struts from chapter 7.

8.2. Analysed Operating Conditions and Flight Profile

Table 8.1 presents the three considered operating conditions of the X-22A propulsor in this thesis, corresponding to a conventional take-off, an endurance cruise, and a combat cruise condition. The combat cruise is equivalent to the cruise for maximum range but is named in line with the naming convention used in the X-22A programme [76].

To derive the thrust levels of the endurance and combat cruise conditions, a fixed lift-to-drag ratio of $L/D = 5$ is assumed for both the endurance and combat cruise conditions, with a maximum take-off weight of $W_{TO} = 71.196\ \mathrm{kN}$ and a total fuel weight of $W_f = 10.857\ \mathrm{kN}$, as taken from the flight manual

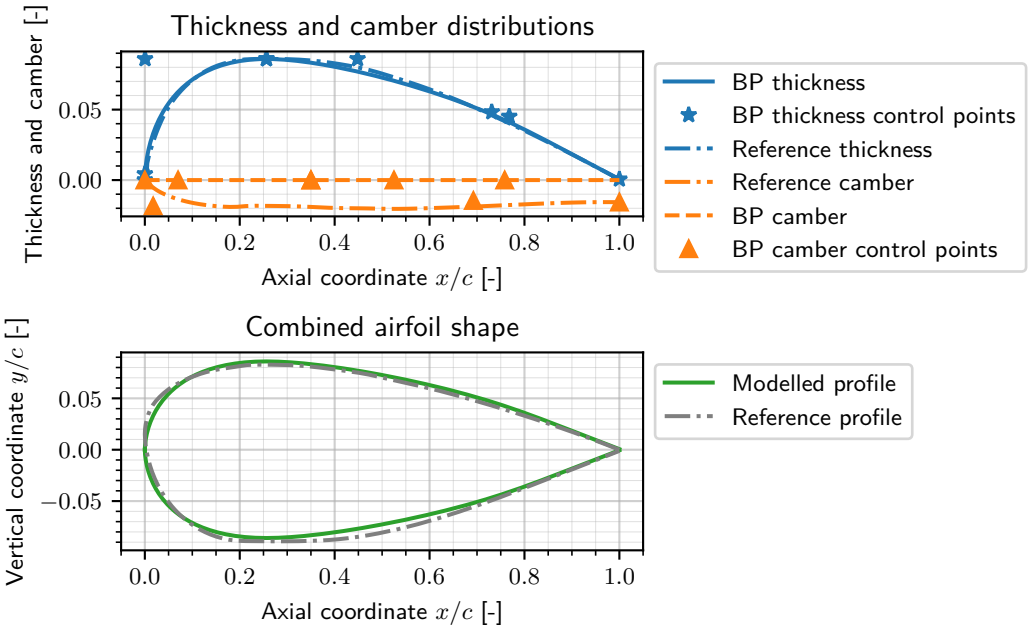


Figure 8.1: Comparison of the X-22A duct profile modelled using the Bézier-Parsec 3434 method and the true X-22A duct profile.

[75]. Both the endurance and combat cruise conditions are evaluated at their mid-phase points, where half the total fuel weight is assumed to remain.

The take-off thrust is derived from a conventional take-off of the X-22A. A vertical take-off, as typically performed by the X-22A, requires defining an analysis at a (near-)zero Mach number with a hover thrust of $T_{hover} = 16.689$ kN per propulsor at the mentioned maximum take-off weight. Such an operating condition deteriorates the numerical stability of MTFLOW in the UDC, as discussed in section 5.4.1. As such, the mission profile is adjusted to use a conventional take-off condition at $M_\infty = 0.125$ with a reduced thrust level, taken as half the hover thrust. All optimisations performed in this thesis used a reference blade tip angle of $\beta_{tip} = 14.5°$, as this thesis is limited to fixed-pitch ducted fans. Additionally, a critical amplification factor $n_{crit} = 9$ is used in all UDC analyses.

Table 8.1: Analysed operating conditions of the Bell X-22A.

Operating Condition	Parameter	Value	Unit
Take-off	h	0	m
	M_∞	0.125	–
	T_{ref}	8.317	kN
	Ω_{ref}	42	s^{-1}
	$t_{take-off}$	1800	s
Endurance	h	3048	m
	M_∞	0.2	–
	T_{ref}	3.272	kN
	Ω_{ref}	44	s^{-1}
	$t_{endurance}$	6120	s
Combat	h	3048	m
	M_∞	0.3	–
	T_{ref}	3.319	kN
	Ω_{ref}	58.5	s^{-1}

The use of a constant lift-to-drag ratio at the endurance and combat cruise conditions, together with

the same weight, logically yields a constant thrust level. The variations in thrust levels observed in table 8.1 is the result of the thrust matching performed, where the rotational rate was matched to within $\Delta\Omega = 0.5\ \mathrm{s}^{-1}$.

To execute a multi-point design optimisation, a flight profile needs to be defined, which is taken as an endurance profile in this thesis. As a multi-point analysis involves multiple UDC evaluations, it is desired to minimise the number of points to minimise the computational cost. The choice is therefore made to model the flight profile using a take-off and a mid-cruise point.

The take-off condition from table 8.1 is used for $t_{take-off} = 20\ \mathrm{min}$ to compensate for the reduced power required compared to the vertical take-off reported by Achton [73] in their endurance mission, and to account for the start-up, taxi-out, and descent/landing segments of the flight. According to the flight manual, a take-off fuel weight of $W_f = 10.857\ \mathrm{kN}$, with a total take-off weight $W_{TO} = 71.196\ \mathrm{kN}$ corresponds to an endurance of $t_{endurance} = 1.7\ \mathrm{h}$ at $M_\infty = 0.2$ and $h = 3048\ \mathrm{m}$ [75]. For the reference geometry described in this chapter and the operating conditions listed in table 8.1, the total energy required for the endurance mission is found to be equal to $E_{endurance_{ref}} = 2.249\ \mathrm{GJ}$.

8.3. Definition of the Optimisation Problems

Several design optimisations are performed in this thesis for the X-22A propulsor to demonstrate the capabilities of the developed UDC and optimisation framework in the conceptual design of ducted fans. These are conducted in order of increasing complexity, where both Single-Objective Optimisation (SOO) and Multi-Objective Optimisation (MOO) problems are considered. The first considered problem is a single-objective, single-point optimisation of the propulsor efficiency, f_1 from table 6.1, given by equation (8.1).

$$\begin{aligned} \min \quad & f_1(\vec{\xi}) \\ \text{s.t.} \quad & g_1(\vec{\xi}) = \eta_p - \frac{2}{2 + \frac{T}{\dot{m} v_\infty}} \leq 0 \\ & g_2(\vec{\xi}) = \left|\frac{T}{T_{ref}} - 1\right| - \epsilon \leq 0 \\ & \ell_i \leq \xi_i \leq u_i \qquad i = 1, \ldots, n \end{aligned} \tag{8.1}$$

$$\begin{aligned} \min \quad & f_1(\vec{\xi}) \\ \text{s.t.} \quad & g_1(\vec{\xi}) = \eta_p - \frac{2}{2 + \frac{T}{\dot{m} v_\infty}} \leq 0 \\ & g_2(\vec{\xi}) = \left|\frac{T}{T_{ref}} - 1\right| - \epsilon \leq 0 \\ & g_3(\vec{\xi}) = \frac{A_{front}}{A_{front_{ref}}} - \sigma \leq 0 \\ & \ell_i \leq \xi_i \leq u_i \qquad i = 1, \ldots, n \end{aligned} \tag{8.2}$$

An additional single-objective, single-point analysis is defined, given by equation (8.2), which extends equation (8.1) by including the frontal area constraint of equation (6.12). The multi-objective optimisation problem defines the frontal area, f_2, as an additional objective, and is given by equation (8.3). Finally, the single-objective, multi-point minimisation of the total flight energy is given by equation (8.4), where g_1 and g_2 must be applied to both operating conditions considered in the flight profile.

$$\begin{aligned} \min \quad & f_1(\vec{\xi}), f_2(\vec{\xi}) \\ \text{s.t.} \quad & g_1(\vec{\xi}) = \eta_p - \frac{2}{2 + \frac{T}{\dot{m} v_\infty}} \leq 0 \\ & g_2(\vec{\xi}) = \left|\frac{T}{T_{ref}} - 1\right| - \epsilon \leq 0 \\ & \ell_i \leq \xi_i \leq u_i \qquad i = 1, \ldots, n \end{aligned} \tag{8.3}$$

$$\begin{aligned} \min \quad & f_3(\vec{\xi}) \\ \text{s.t.} \quad & g_1(\vec{\xi})_j = \eta_{p_j} - \frac{2}{2 + \frac{T_j}{\dot{m}_j v_{\infty_j}}} \leq 0 \quad j = 1, 2 \\ & g_2(\vec{\xi})_j = \left|\frac{T_j}{T_{ref_j}} - 1\right| - \epsilon \leq 0 \quad j = 1, 2 \\ & \ell_i \leq \xi_i \leq u_i \qquad i = 1, \ldots, n \end{aligned} \tag{8.4}$$

For all optimisation problems considered, $\epsilon = 0.01$ such that the thrust is maintained to within 1% of the reference value, and $\sigma = 1.05$ so that the frontal area cannot exceed 105% of that of the reference design. The duct geometry and rotor are optimised, while the struts and centrebody have a fixed design to balance the size of the optimisation problems, while still demonstrating the capabilities of the developed UDC and optimisation framework. The rotor is parameterised using four equally spaced radial sections. The strut spans are fixed equal to $z_{LE_{duct}}$ to ensure the duct always clips them.

A constant minimum tip gap of $\Delta z = 1.016\ \mathrm{cm}$, equal to that of the real X-22A [71], is used to control the z-coordinate of the duct using equation (4.7). Although the tip gap has a substantial influence on the

performance of the ducted fan, this influence is not included in the current demonstration due to the lack of loss modelling. All optimisations use a population size of $K = 100$, where the initial population size is doubled to ensure a broad sampling of the design space. The continuous population initialisation parameters are taken as $\nu = 0.5$, $\tau = 0.25$, while the discrete parameters are defined as $a = 0$, $b = 12$. Where pseudo-random numbers are used in the operators of the genetic algorithm, a seed of 42 is used, in reference to Adams [77].

8.4. Bounds of the Design Variables

The design variables can be divided into two groups: profile shape and geometric design variables. For the profile shapes, the design variable bounds shown in table 8.2 are used. These bounds are based on the application of the BP3434 method by Rogalsky [53]. For the duct geometry, the bounds are adjusted such that $y_t \in [0.05, 0.20]$. This avoids excessively thick or thin duct geometries, particularly in the initial population. The remaining design variable bounds are defined in table 8.3.

Table 8.2: Bounds for the dimensionless Bézier-Parsec 3434 design variables.

Parameter	ℓ	u	Parameter	ℓ	u
b_0	0.050	0.10	x_c	0.200	0.500
b_2	0.125	0.30	y_c	0	0.150
b_8	0.050	0.70	z_{TE}	0	0.050
b_{15}	0.700	0.95	dz_{TE}	0	0.005
b_{17}	0.700	0.95	r_{LE}	−0.200	−0.001
x_t	0.150	0.40	β_{TE}	0.001	0.400
y_t	0.020	0.30	α_{TE}	0.001	0.200
			γ_{LE}	0.001	0.200

Table 8.3: Bounds for the geometric design variables.

Parameter	Unit	ℓ	u	Parameter	Unit	ℓ	u
c_{duct}	m	1.00	1.50	Ω	s^{-1}	20.00	80.00
$x_{LE_{duct}}$	m	−0.50	0.50	R	m	0.50	1.50
x_{root}	m	0	0.40	c_i	m	0.10	0.75
β_{tip}	rad	0.10	$5\pi/18$	ϕ_i	rad	0	$\pi/3$
B	-	3	15	β_i	rad	0	$\pi/3$

The rotational rate of the fan, Ω, is duplicated in the case of a multi-point analysis, such that the rotational rate is unique for each operating point. In this thesis, the analyses are limited to fixed-pitch ducted fans, as described in section 2.4. Variable-pitch ducted rotor stages can be analysed by also duplicating the reference blade angle, β_{tip}, for each operating point, in a multi-point analysis.

9

Single-Point Design Optimisations of the X-22A Propulsor

This chapter presents the results of the single-point design optimisations of the X-22A propulsor performed in this thesis. Both single-objective optimisations, presented in section 9.1, and a multi-objective optimisation, presented in section 9.2, were performed, and the optimised designs were compared. Additionally, a review of the practical application of single-point optimisation on performance at off-design operating conditions is presented in section 9.3.

9.1. Single-Objective Single-Point Optimisation

Design optimisations were conducted with increasing complexity, starting with single-objective, single-point optimisations, defined by equation (8.1). An overview of key performance metrics for each performed optimisation is shown in table 9.1, from which it is seen that the endurance optimisation terminated with a zero average constraint violation, indicating an entirely feasible population. This is an encouraging result, as the algorithm can then fully focus on improving the objective function during the so-called diversity phase of the algorithm [68]. However, this is not a requirement for the meaningfulness of the algorithm's output.

For the combat analysis, the final population consisted of 45 infeasible individuals, while the take-off analysis had 6 infeasible individuals. Additional function evaluations, using an increased sliding convergence window size, population size K, or different seed/initialisation parameters ν, τ, and δ, may yield further improvements in the propulsor efficiency. However, this falls outside of the scope of this thesis due to computational resource limitations.

Table 9.1: Overview of the single-objective, single-point efficiency optimisation results.

Parameter	Unit	Take-off	Endurance cruise	Combat cruise
n_{gen}	-	18	79	30
$CV_{avg,final}$	-	5.00×10^{-5}	0	2.52×10^{-3}
$\eta_{p_{ref}}$	$\%$	71.05	79.64	75.43
$\eta_{p_{opt}}$	$\%$	75.35	84.83	85.45
$g_{1_{opt}}$	-	-1.70×10^{-2}	-1.23×10^{-3}	-8.56×10^{-2}
$g_{2_{opt}}$	-	-4.24×10^{-3}	-3.16×10^{-3}	-3.76×10^{-3}

Optimisations were run on three different computers: a desktop with 16 CPU threads, a laptop with 22 CPU threads, and a server with 96 CPU threads. On the server, 61 available threads were used. The optimisations demonstrated evaluation rates between $10 - 45$ s per design, illustrating the fast nature of the developed UDC and the implemented optimisation framework.

9.1.1. Convergence Behaviour of the Single-Objective Analyses

To demonstrate the robustness of the developed optimisation framework, the convergence of the average constraint violation, CV_{avg}, population diversity, and the objective values are inspected. The population diversity is quantified using the normalised mean Euclidean distance between the design vectors of a generation, where the variables are normalised using the corresponding minimum and maximum values of each generation. The resulting trends are shown in figure 9.1 for the endurance cruise condition. This condition was chosen due to the slowed observed convergence, as indicated by the number of generations required to reach termination, n_{gen}, in table 9.1. The trends for the other single-point, single-objective optimisations performed in this thesis are included for reference in appendix E.

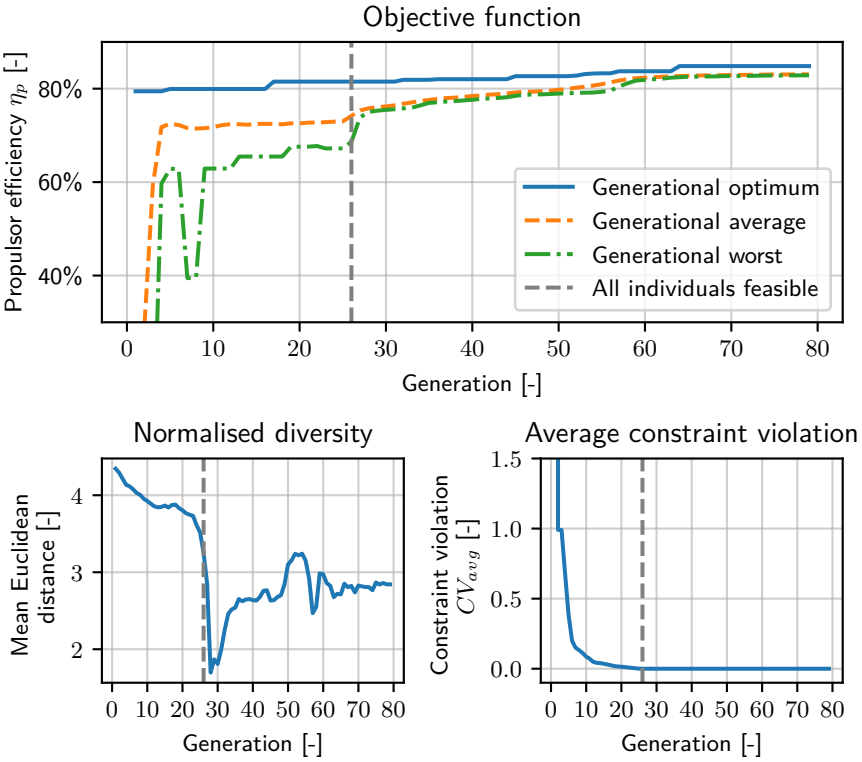


Figure 9.1: Convergence of the objective function value, normalised population diversity, and average constraint violation for the single-point, single-objective efficiency optimisation at the endurance cruise condition of the X-22A propulsor.

The biased initialisation method used in section 6.3 introduces significant diversity, but consists mainly of infeasible designs, with only the reference design being fully feasible. This results in the large constraint violation seen for the initial population. To improve the computational efficiency of the algorithm, it is preferred to have more or only feasible design vectors in the initial population. In future work, it is therefore recommended to implement a physics-based (analytical or simplified) design method to generate a set of feasible designs in the initialisation of the optimisation framework.

The presence of (slightly) infeasible design vectors in the initial population is not a weakness of the framework, however. These design vectors significantly contribute to maintaining diversity in the population, allowing the selection operator to trade the exploration of new regions in the design space against minor constraint violations. This is observed in the convergence of the objective function value, with the first efficiency improvements occurring early in the optimisation process when there are still significant constraint violations in the population.

The observed convergence rate of the average constraint violations is driven by the thrust constraint, $g_2(\xi)$, with the full population becoming feasible after 26 generations. The convergence of the population's constraint violations may be improved by including an internal Ω-matching loop in the optimisation framework to automatically determine the rotational rate required to satisfy or minimise the violation of

the thrust constraint based on a lower-order method. This was not included in the present work due to resource limitations in both time and computational cost.

Once all individuals become feasible in generation 26, the generational average and worst propulsor efficiencies are seen to converge to similar values, as the optimisation focuses on a specific region in the design space. The diversity increases in subsequent generations and oscillates, corresponding to the diversity phase of the algorithm [68]. The high normalised final population diversity is seen to originate in the blade count B, which varies for the final population between $6 \leq B \leq 12$.

Due to the lack of gradient information in the genetic algorithm, no statement can be made on the optimality of the found solution [47]. As such, it is advised in future work to use the found optimal solution or final population as a starting point for a gradient-based optimisation method to determine the optimality and further quantify the design space surrounding the found solution(s).

9.1.2. Comparison of the Optimised Meridional Designs

A meridional view of each of the optimised designs is shown in figure 9.2. As shown in this figure, the duct chord length has an inverse dependency on the Mach number. At higher Mach numbers, the duct drag increases, as mentioned in section 3.2. By reducing the duct length, the duct wetted area is reduced, minimising this increase.

Additionally, an increase in the diffuser expansion ratio $\Lambda = A_{exit}/A_{fan}$ is observed with increasing Mach number. The combat cruise design, in particular, is at risk of flow separation on the inside of the duct at off-design conditions due to the largest expansion ratio of $\Lambda = 1.23$. By increasing the expansion ratio, pressure recovery is increased, thereby improving the propulsor efficiency. It is likely that for higher freestream Mach numbers, this expansion ratio is reduced, in line with results reported by Black, Wainauski and Rohrbach [18], who found significant performance penalties for larger expansion ratios at $M_\infty \geq 0.4$. However, this falls outside of the operational envelope of the X-22A, which has a maximum speed of $M_{max} = 0.44$ at $h = 3048$ m [76], and was therefore not further investigated in this thesis.

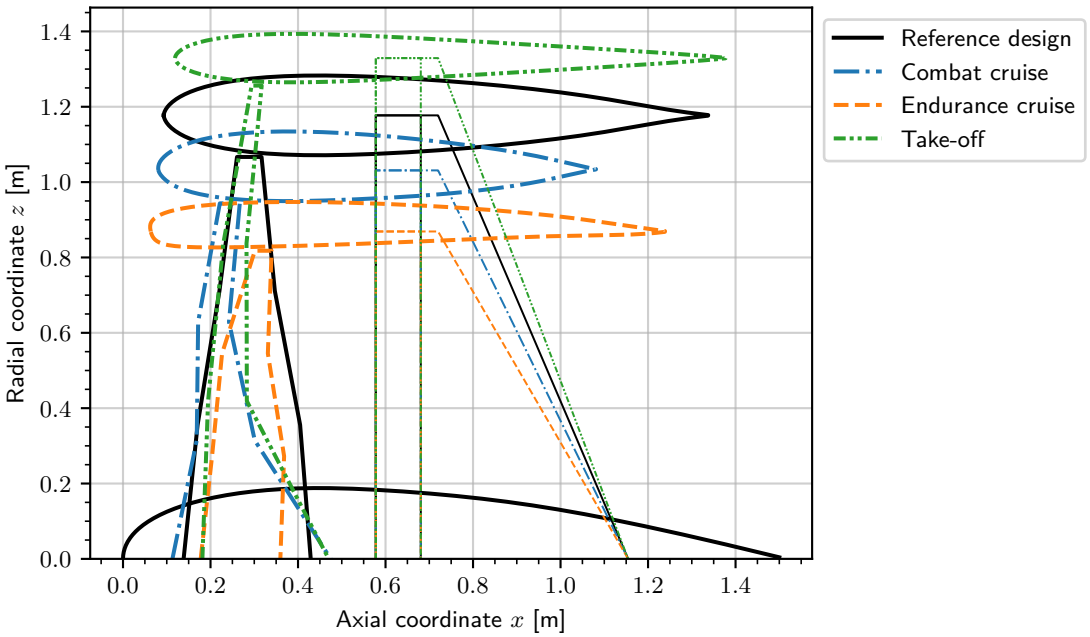


Figure 9.2: Meridional view of the optimised and reference designs for the single-point propulsor efficiency optimisations. Note the constant centrebody geometry and the constant scaled geometry of the struts.

Due to the different fan meridional geometries, it is not possible to draw a clear conclusion on the diameter; further analysis is required to accurately quantify the effects of both thrust setting and Mach

number. For the take-off design, the propulsive efficiency is improved by increasing the fan diameter and thus the mass flow, such that a smaller velocity increment in the slipstream yields the same thrust [66]. The lack of a clear trend with increasing Mach number is attributed to the premature convergence of the combat cruise condition, indicated by the high number of infeasible individuals in the final population discussed at the beginning of this section.

Multiple evaluations of the optimisation framework are needed when premature convergence occurs, using the approach described at the start of this section. As shown in the convergence overview for the combat cruise optimisation in appendix E, the combat optimisation got stuck in a local optimum. Performing additional optimisation using different initialisation parameters can help to overcome local optima and allows for the selection of a design from the resulting optimised solution set.

A final trend is identified in the camber distribution of the duct profile, characterised by a forward shift of the locus of camber with increasing Mach number. For the magnitude of camber, it is observed that y_c decreases with increasing Mach number or thrust setting. The combat design has the lowest non-zero camber of $y_{c_{combat}} = 0.011$, while the endurance condition camber is $y_{c_{endurance}} = 0.018$. The take-off duct is symmetric. However, similarly to the discussion of fan diameter, further analyses are required to confirm these trends.

The duct thickness of the combat cruise optimised design is significantly larger than that of the take-off and endurance optimised designs, which have a similar thickness distribution, as seen in figure 9.3. This is a result of the premature convergence of the optimisation to a local optimum, combined with a limitation of the parameterisation method used for the duct. An increased expansion ratio Λ is only possible when the trailing edge vertical displacement z_{TE} increases, together with the duct thickness y_t. This is an inherent limitation of the BP3434 method. A tailored parameterisation method for the duct, offering more direct control over the expansion ratio, Λ, will likely yield the expected thinner duct geometry.

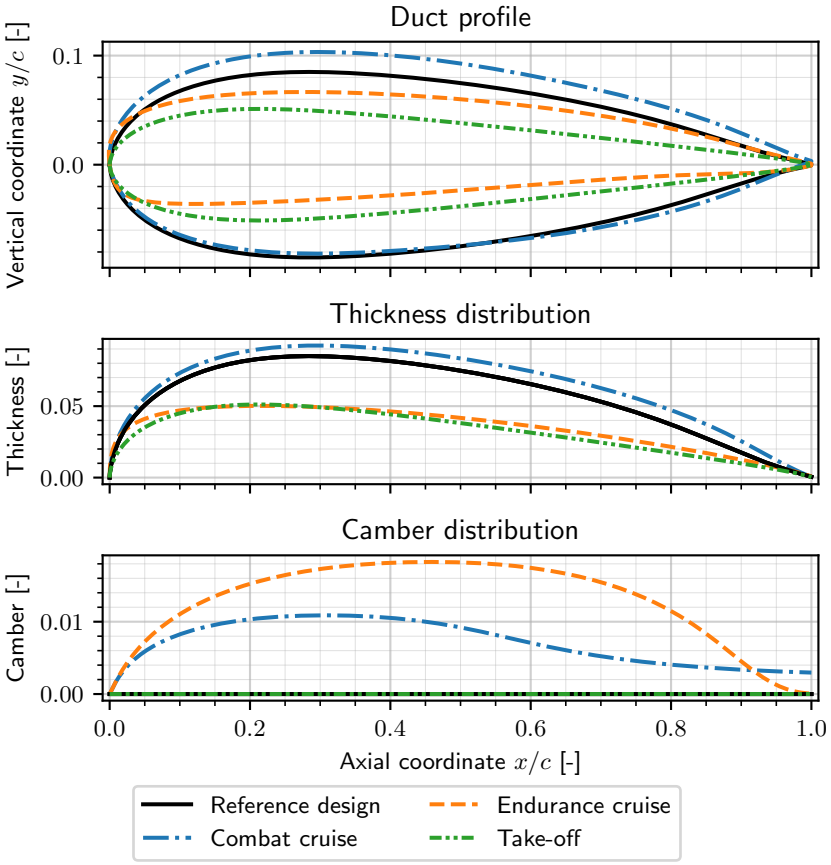


Figure 9.3: Comparison of the single-point efficiency-optimised duct profiles against the reference design.

9.1.3. Comparison of the Optimised Fan Designs

Key fan blade data is shown in table 9.2 for all optimised designs. The root leading edge x coordinate is also expressed in a normalised definition as a fraction of the duct length, in line with Gao and Xu [12], as $x_{\breve{root}} = (x_{root} - x_{LE_{duct}})/c_{duct}$. An interesting result is obtained for this normalised location, where the take-off and combat optimised designs have a similar normalised location of the fan root. The results obtained for the root coordinate $x_{\breve{root}}$ do not show the trend hypothesised in section 2.3, thus contradicting the results reported by Gao and Xu [12]. This illustrates the complex design space of the ducted fan.

The effects of the high-thrust, low-speed conditions during takeoff are evident, with a lower reference blade tip angle and increased fan diameter. The increased blade count observed for all optimised designs is a direct result of the first-order blade loading model, where the rate constant k has a linear dependency on the blade count as defined by equation (4.10). A higher blade count results in an increased slope $d\bar{\Gamma}/ds$ at the same radial location z, enabling a shorter blade chord to be used, as is shown in figure 9.4 for all designs in the outboard blade span.

Table 9.2: Key blading data for each of the single-point efficiency-optimised designs.

Parameter	Unit	Reference design	Take-off	Endurance cruise	Combat cruise
x_{root}	m	0.150	0.187	0.187	0.139
$x_{\breve{root}}$	-	0.0458	0.0540	0.105	0.0581
β_{tip}	°	14.5	7.9	17.5	19.0
B	-	3	14	10	6
Ω	s^{-1}	N/A	47.1	53.0	61.5
Ω_{ref}	s^{-1}	N/A	42.0	44.0	58.5
R	m	1.067	1.256	0.818	0.944

The radial distributions of the chord length, sweep angle, and blade angle are shown in figure 9.4. From this figure, it can be seen that all three designs have a similar sweep angle distribution, with a reduction at the 67% span station, resulting in the (nearly) straight leading edges seen in figure 9.2. The large reduction in leading edge sweep angle observed for the combat cruise optimised design is the result of the repair operator, described by equation (6.7).

The algorithm uses the root section as a control point for the interpolation to construct the blade profile outside of the centrebody. As such, large differences are seen in the chord length and blade angles at the root for the three operating conditions. However, outside of the centrebody, the chord lengths have a much smaller variance, suggesting there is a plateau in the design space for the chord length. Interestingly, the outboard blade angle distribution remains, by comparison, much more constant compared to the inboard blade distribution across the optimised designs. Nevertheless, a reduction is still seen compared to the reference. These changes indicate the algorithm is focused on adjusting the radial blade loading distribution on the inboard span section.

Figure 9.5 shows the blade profiles at each of the four defined radial sections. It can be seen that the inboard blade profiles experience little change in the maximum camber, while the camber distributions have changed marginally to improve the blade slope S_{rel}. For the outboard blade sections, the maximum camber changed significantly as the algorithm adjusted the blade loading. For increasing Mach numbers, the point of maximum camber shifts aft at 67% and 100% of the blade span. There is no clear trend visible in the magnitude of the camber due to the non-linear dynamics of the camber distribution, blade angle distribution, and blade count in the blade row swirl generation.

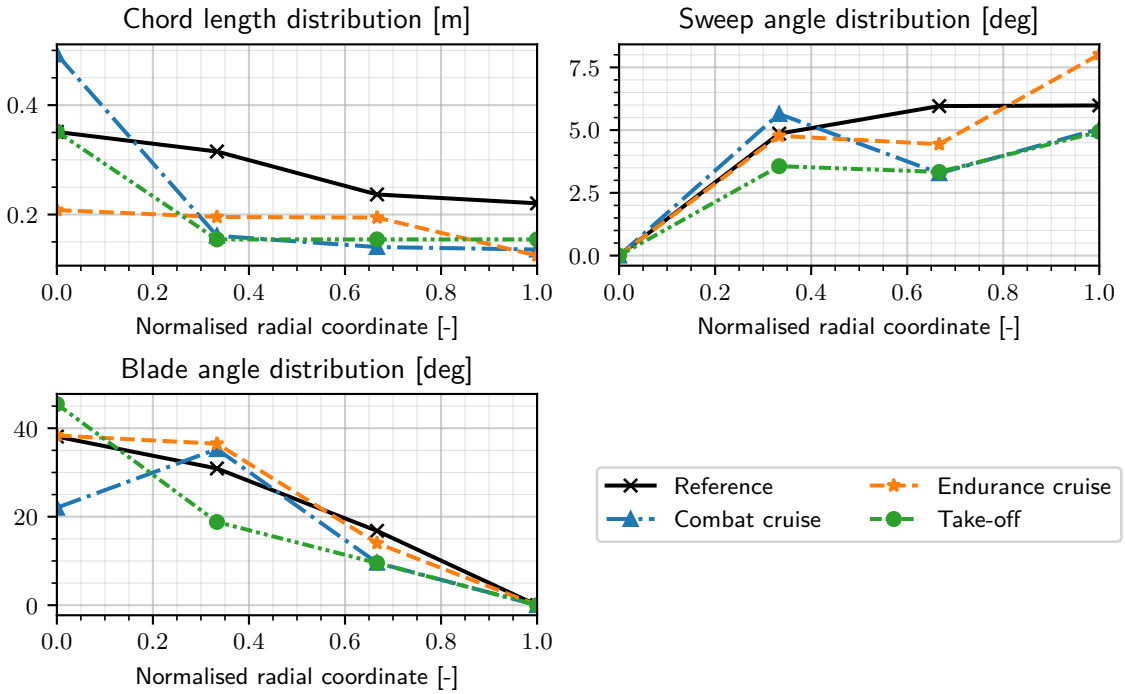


Figure 9.4: Radial distributions for each of the single-point efficiency-optimised designs.

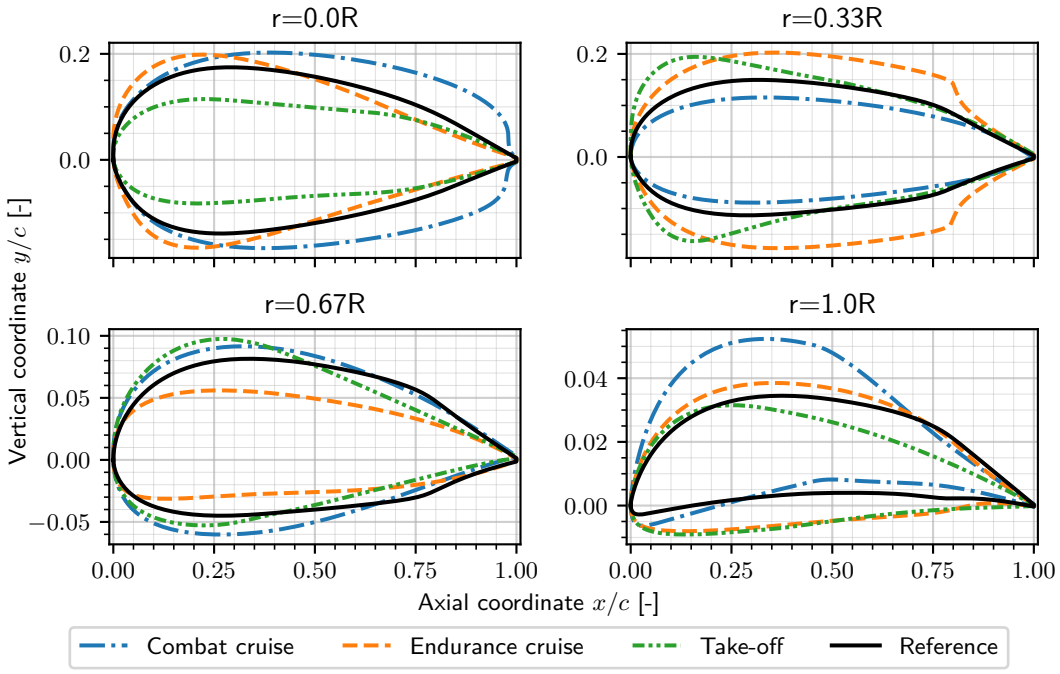


Figure 9.5: Blade profiles at the radial sections for each of the single-point efficiency-optimised conditions.

The blade loading model described in section 4.4 does not capture the flowfield through the blade rows, but only the forcing resulting from the circumferential thickness and swirl distributions. As such, the effects of the blade profile thickness distributions are not sufficiently captured by the UDC, resulting in unrealistic trailing edge blade shapes. This is shown at the root and 33% span stations for the combat and endurance cruise optimised designs. While these blade designs are valid BP3434 parameterisations, they would encounter severe flow separation at their trailing edges, making them undesirable.

To avoid such blade designs, it is necessary to integrate a higher-order blade loading model or a dedicated blade cascade solver in the UDC to obtain more physically accurate blade shapes. In the UDC, the thickness distribution of the blade has only a small influence on the performance of the ducted fan, as the losses over the blade row are neglected. The effect of the blade thickness is captured in the circumferential blade thickness, which affects the blockage. In equation (3.7), this blockage affects the meridional velocity, which in turn increases the high-solidity limiting swirl given by equation (4.9). For blade profiles such that $T_{\theta_i} B \ll 2\pi z_i$, this effect is much smaller than the direct effect of the geometric blade slope S_{rel} of the camber distribution. Although the thickness distribution is thus not reliably defined by the algorithm, the camber distribution is. Further studies can focus on determining the optimal thickness distributions to complement the found optimised camber distributions.

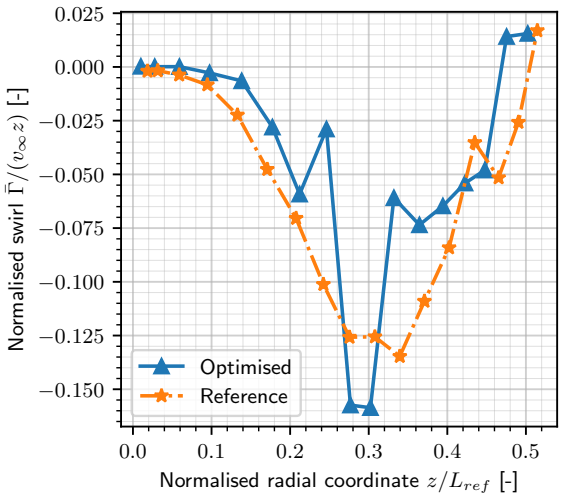
9.1.4. Comparison of the Outlet Swirl and Duct Pressure Coefficients

To further analyse the changes made by the algorithm to the designs and their impact on the resulting flowfield, a comparison is made between the reference and optimised design normalised swirl distributions $\bar{\Gamma}/(v_\infty z)$ at the outlet plane of the computational domain, as well as the pressure distribution C_p over the duct geometry. The resulting swirl comparisons are shown in figure 9.6, while the pressure distributions are shown in figure 9.7.

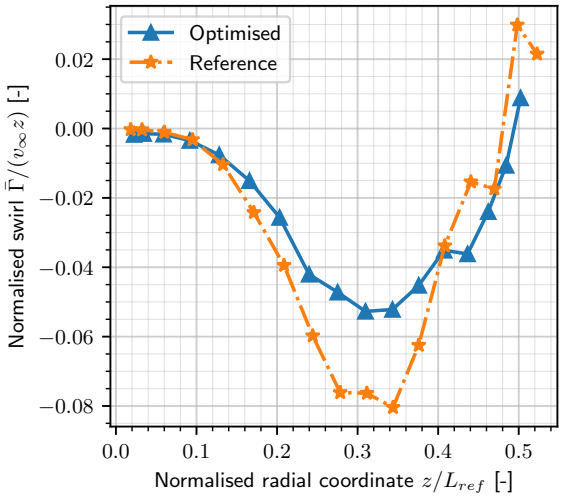
The swirl distributions in figure 9.6 show the expected loading distribution of the axial fans, with a peak in exit swirl near the mid-span, corresponding to the most heavily loaded radial section of the fan blades. To positively contribute to the thrust produced by the ducted fan, the fan should generate swirl such that $\bar{\Gamma} < 0$ as indicated by equations (3.8) and (3.10). The expected tip losses are therefore also clearly visible for both the reference and optimised designs, where the normalised swirl is positive at the outlet for the streamlines passing near or through the tip.

The effects of the optimised fan can be seen in the magnitude of the exit swirl distributions. Residual swirl at the outlet of the computational domain corresponds to wasted energy as this momentum is not aligned with the flow, resulting in a loss of efficiency. An optimised design would therefore have a reduced total integrated exit swirl. This is seen across all operating conditions in figure 9.6.

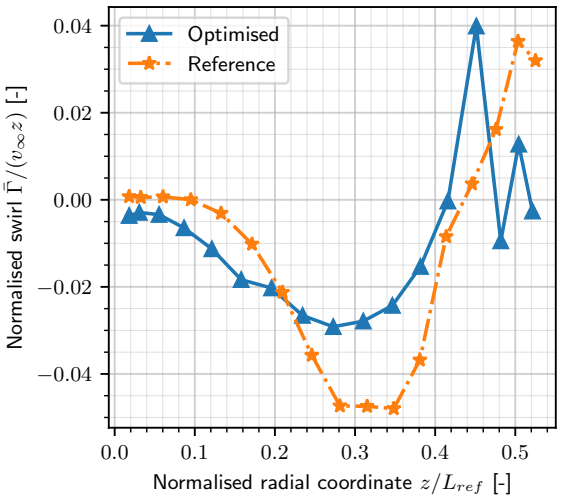
The link between total exit swirl and propulsor efficiency is also seen, with the smallest total outlet swirl occurring for the combat optimised design, which has the highest efficiency as seen in table 9.1, compared to the take-off condition, which has the lowest efficiency and correspondingly highest total exit swirl. The combat cruise optimised design shows an oscillatory behaviour of the outlet swirl for the outboard span. This is attributed to the finite streamline resolution, where the changed ducted fan design affects the streamtube size and distribution through the ducted fan. This, in turn, affects the resulting domain exit swirl distribution. Regardless of this oscillatory behaviour, all three optimised designs have a noticeable reduction in exit swirl, including a reduction in tip losses. Further reductions in the exit swirl may be achieved by including the design of the struts in the optimisation, as these also act as swirl recovery vanes in the X-22A propulsor.



(a) Comparison of the outlet swirl distribution between the optimised and reference designs at the take-off condition.



(b) Comparison of the outlet swirl distribution between the optimised and reference designs at the endurance cruise condition.



(c) Comparison of the outlet swirl distribution between the optimised and reference designs at the combat cruise condition.

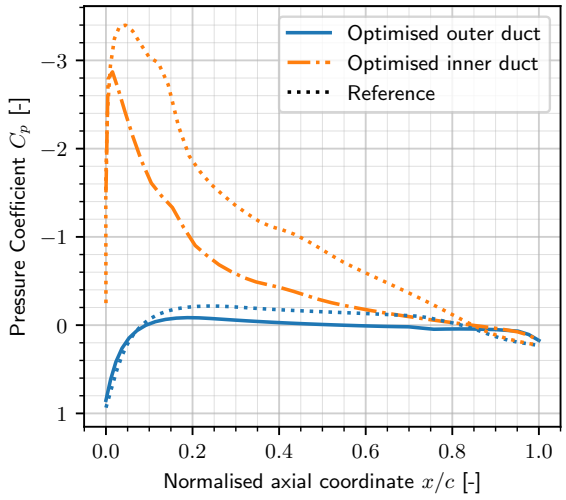
Figure 9.6: Comparison of the exit swirl distributions between the reference and single-point, single-objective efficiency-optimised designs. Note that only the exit swirl for the streamlines passing through the fan is shown.

When analysing the pressure coefficient distributions over the duct surfaces in figure 9.7, it can be seen that the outer duct distribution experiences only minor changes for all optimised designs. This indicates a focus of the algorithm on adjusting the pressure distribution on the inner duct surface, which is correctly identified as being the most critical to improve performance. For all three optimised designs, the pressure difference across the upper and lower surfaces of the duct has reduced, as indicated by the reduced areas enclosed by the upper and lower pressure distributions. The effects of the optimised duct thickness and camber distributions are seen across all optimised designs, with a reduction of the inner duct suction and a focusing of the suction towards the lip of the duct. This is particularly true for the take-off and endurance-optimised designs, where a refined suction peak is seen.

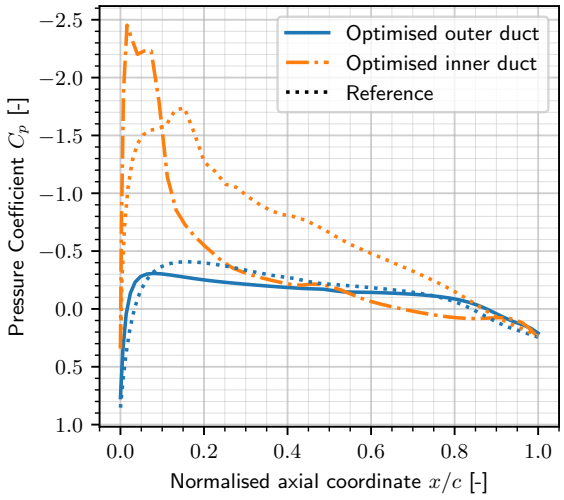
This suction peak is smaller for the take-off optimised design compared to the reference. Correspondingly, this duct has a substantially smaller thrust contribution originating from the pressure distribution, being $\Delta T_{duct,invisc_{TO}} = -47.12\%$ compared to the reference design. In contrast, the endurance and combat cruise conditions have increased inviscid thrust contributions of $\Delta T_{duct,invisc_{endurance}} = 54.85\%$ and $\Delta T_{duct,invisc_{combat}} = 33.14\%$, as their resulting pressure distribution results in a force which is more aligned with the axial flow direction.

For the combat-optimised design, the suction peak is more smeared out due to the significantly different

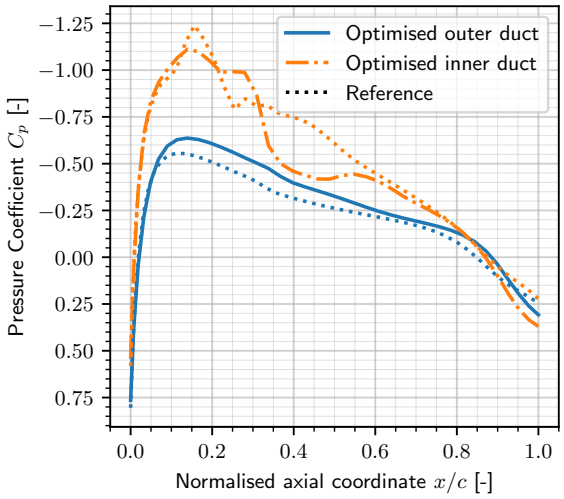
inner duct surface curvature resulting from the differing thickness distribution. A sharp kink is observed in $dC_p/d(x/c)$ at $x/c = 0.30$, which corresponds to the location of maximum thickness x_t of the duct. Despite this sharp kink and subsequently high adverse pressure gradient, no flow separation was observed. For all three designs, a reduction in the axial friction force acting on the duct is observed, ranging from 4.59% for the take-off design to 26.80% for the combat cruise optimised design, contributing to an improved efficiency.



(a) Comparison of the pressure distributions between the optimised and reference designs at the take-off condition.



(b) Comparison of the pressure distributions between the optimised and reference designs at the endurance cruise condition.



(c) Comparison of the pressure distributions between the optimised and reference designs at the combat cruise condition.

Figure 9.7: Comparison of the pressure distributions between the reference and single-point, single-objective optimised designs.

9.1.5. Effects of the Optimised Designs on the Contributions to the Ducted Fan Thrust

The optimised ducted fan designs presented in this section have correspondingly different loading distributions between the centrebody, duct, and the fan. This was hinted at in the previous subsection, where the duct pressure distributions and domain outlet swirl distributions were compared, and is further discussed in this subsection. For each of the single-point optimised designs, the breakdown in thrust contributions is shown in table 9.3. It can be seen that the thrust contribution of the centrebody changed for the optimised designs. This is caused by changes in the pressure distributions and associated wake formations at the trailing edge of the centrebody, resulting from the optimised fan and duct designs.

An opposite trend is seen between the take-off condition and the endurance/combat cruise conditions, where the contribution of the fan is reduced for the latter optimised designs, but increased for the take-off optimised design. For the take-off optimised design, the duct contribution reduced by 7.29%, opposite to what is expected for a (near) static condition, where the duct positively contributes to the net ducted fan thrust as discussed in section 3.2. An increased contribution would thus be expected for the optimised design. The lack of occurrence of this behaviour is attributed to premature convergence of the take-off optimisation, indicated by the low generation count n_{gen} in table 9.1. Yet, despite this premature convergence, a significant performance improvement was noted for the take-off condition, with a propulsor efficiency improvement of $\Delta\eta_p = 4.5\%$, illustrating the capabilities of the developed framework.

The reduction in duct thrust contribution also increases the relative duct losses, as indicated by the duct efficiency, $\eta_{duct} = T_{duct,net}/T_{duct,invisc}$, as the viscous force for the take-off condition reduced by 4.59%, while the thrust contribution reduced significantly more. This viscous force is dominated by profile drag, which remains nearly constant for the take-off condition, as the wetted area increases due to the increased diameter and the near-constant duct chord length compared to the reference design. For the take-off optimised design, a duct efficiency reduction of $\Delta\eta_{duct} = -2.49\%$ was observed. The endurance and combat cruise optimised designs did not show this behaviour, with duct efficiency improvements of $\Delta\eta_{duct_{endurance}} = 8.96\%$ and $\Delta\eta_{duct_{combat}} = 29.47\%$, respectively. These improvements in duct efficiency are attributed to the increase in duct pressure forces, while the viscous forces remain nearly constant.

Table 9.3: Comparison of the contributions of the duct, centrebody, and fan to the total thrust for the optimised designs expressed as fractions of the total thrust.

Element	Unit	Take-off		Endurance cruise		Combat cruise	
		Reference	Optimised	Reference	Optimised	Reference	Optimised
T_{duct}	%	20.01	12.72	12.74	16.68	9.86	18.08
$T_{centrebody}$	%	1.40	1.07	1.83	4.45	2.21	3.24
T_{fan}	%	78.59	86.20	85.43	78.87	87.93	78.68

9.2. Multi-Objective Single-Point Optimisation

Although the U-NSGA-III genetic algorithm has been demonstrated to perform well for single-objective problems in both the previous section and the original paper by Seada and Deb [50], the reference direction niching in the algorithm only works when at least two objectives are used. Using a multi-objective optimisation of a ducted fan design is more representative of a real-world application, as aspects such as weight or physical size can have a significant impact on the overall design of an aircraft. Additionally, this also serves as a demonstration of the capabilities of the developed algorithm in a multi-objective optimisation, opening a pathway to future, more complex (multidisciplinary) analyses. An example would be the inclusion of an electric motor sizing code to determine the motor weight given a required peak power as found from the UDC analysis, or the introduction of the weight of the complete ducted fan propulsor as an objective.

This weight estimation must be sensitive to changes in all design variables. It could therefore be based on a semi-empirical method to capture changes in the rotor design, struts, centrebody, and duct, such as the method developed by Lolis [78]. Parts of this method have been applied in previous work by van den Dungen [79] in the weight estimation of a ducted fan used as propulsive empennage, demonstrating its suitability for ducted fan weight estimation.

A Multi-Objective Optimisation (MOO), defined by equation (8.3), was executed to optimise both the propulsor efficiency and frontal area of the ducted fan at the endurance cruise condition. To compare the effects of formally introducing a second objective, a single-objective optimisation was also performed with a maximum frontal area constraint defined by $g_3(\xi)$, equation (8.2), where $\sigma = 1.05$. The found non-dominated solution set is shown in table 9.4. From this table, it can be seen that the single-objective (area-constrained) optimisations and the multi-objective optimisation converge to different points in the objective space. This is a result of the reference direction niching selection operator used in the MOO.

While the area-constrained SOO did not achieve better efficiency than the regular SOO from section 9.1, the MOO identified one non-dominated solution with improved efficiency, albeit at the cost of a higher frontal area compared to the regular SOO. This indicates a positive correlation between frontal area and efficiency. Further improvements in the propulsor efficiency may be achieved when using different initialisation parameters for the SOO and MOO.

Table 9.4: The non-dominated solution set for the single-point endurance cruise condition multi-objective optimisation, compared against the single-objective optimised designs.

Parameter	Unit	SOO	SOO g_3	MOO Solution		
				1	2	3
$\eta_{p_{opt}}$	%	84.83	82.31	79.24	80.43	86.73
$\frac{A_{front_{opt}}}{A_{front_{ref}}}$	%	54.51	51.69	52.88	58.18	90.67
$g_{1_{opt}}$	-	-1.23×10^{-3}	-2.48×10^{-2}	-5.89×10^{-2}	-4.82×10^{-2}	-3.41×10^{-2}
$g_{2_{opt}}$	-	-3.16×10^{-3}	-3.34×10^{-3}	-5.41×10^{-3}	-8.58×10^{-3}	-5.50×10^{-4}

9.2.1. Convergence of the Multi-Objective Analysis

The evaluated feasible design space is shown in figure 9.8. This figure shows the different regions of the objective space occupied by the single-objective and multi-objective optimisations. This is caused by the MOO algorithm's niching, which encourages the algorithm to explore different regions of the design and associated objective space compared to the SOO analysis. The significantly different regions of the objective space being occupied by the SOO and MOO optimisations highlight a need to run the optimisation(s) for varying seeds and initialisation parameters τ, ν, δ to obtain a total "complete" objective space overview.

This is especially true, as single-objective solutions at higher frontal areas and associated efficiencies would be expected, given that the higher corresponding mass flow enables a lower velocity rise in the slipstream to maintain a constant thrust, which improves the propulsive efficiency, leading to an improved propulsor efficiency as well. The lack of such solutions is attributed to the optimisation framework converging to local optima at lower frontal areas as a result of the complex design space. It is hypothesised that further analyses will result in a more uniform objective space distribution. However, such studies have not been performed in this thesis due to time and resource limitations. Nevertheless, the shown objective space illustrates the suitability of the developed optimisation framework in identifying improved designs at the conceptual design stage.

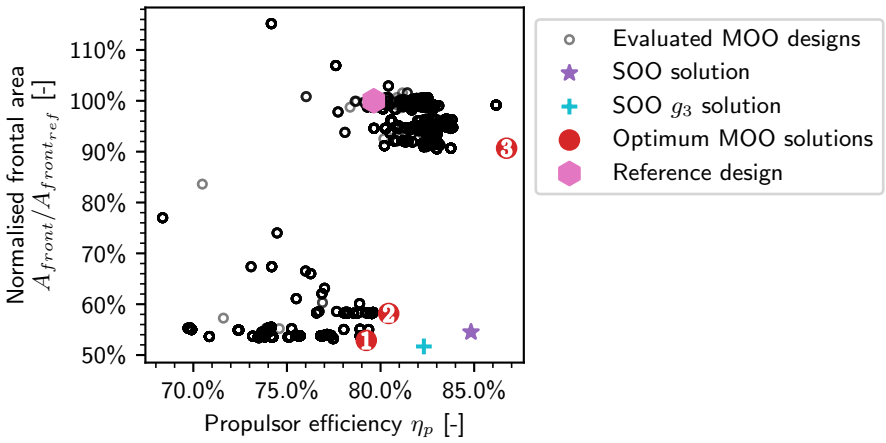


Figure 9.8: Feasible evaluated objective space for the multi-objective single-point optimisation at the endurance cruise condition of the X-22A propulsor. Note the design space occupied by the single-objective optimisation outputs.

The convergence overview of the multi-objective optimisation is shown in figure 9.9. As shown in this

figure, the analysis terminated after 100 generations, during which the last 11 generations maintained a constant objective space. In the running metric plot, the black markers shown correspond to generations where the change in either the ideal or worst objective vector was larger than the objective space termination criteria of 1×10^{-5}. The running metric indicates that the most significant improvements occur during the first 40 generations, corresponding to the convergence phase of the algorithm.

The effects of the diversity phase of the algorithm on the population diversity are seen in figure 9.9 as well, with the final population diversity being nearly equal to that of the initial population. This further illustrates the robustness of the algorithm, as it searches a large design space to find the optimal solution(s). This behaviour makes the algorithm particularly suitable for conceptual design optimisations, where larger, fundamental changes to the ducted fan may be considered, such as blade counts, fan diameter, or fan location within the duct.

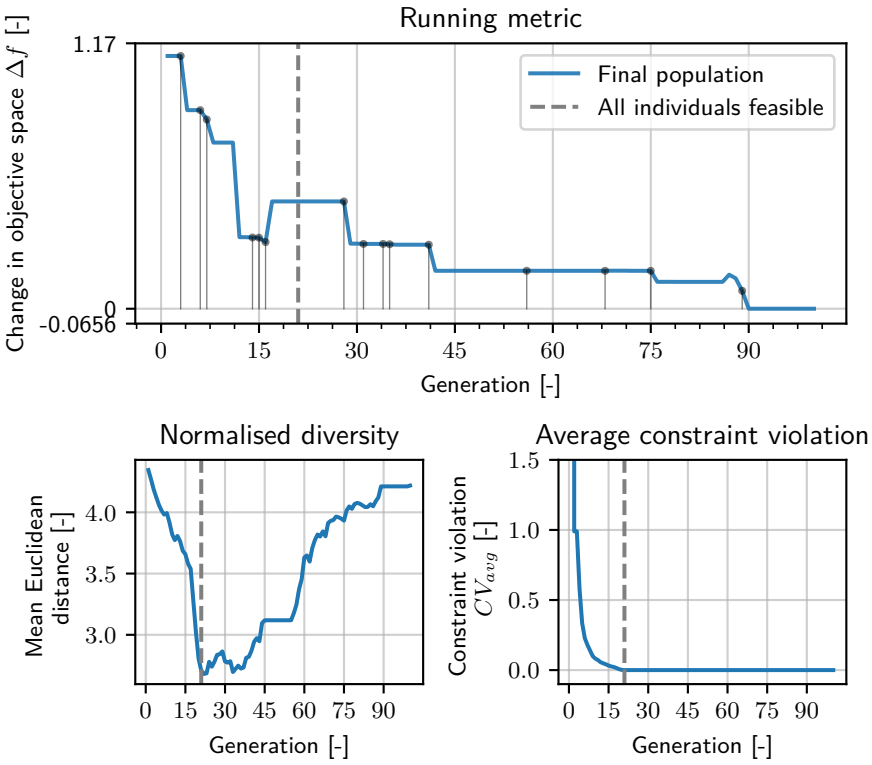


Figure 9.9: Convergence of the running metric, normalised population diversity, and average constraint violation for the single-point multi-objective optimisation of the endurance cruise condition of the X-22A propulsor.

9.2.2. Analysis of the Non-Dominated Solution Set

A meridional view of the non-dominated solutions is shown in figure 9.10. As seen in this figure, significantly reduced duct chord lengths are obtained for the area-constrained SOO and the MOO solution set compared to the design presented in section 9.1. This further illustrates the difference in design space and associated objective space analysed by the MOO, SOO, and area-constrained SOO. Interestingly, the most efficient design, individual 2, with the largest diameter, has a meridional blade shape similar to that of the reference design. This supports the argument made by Mort and Gamse [71], who mention that the X-22A propulsor is already well-designed, with minor improvements being achievable in the surrounding design space.

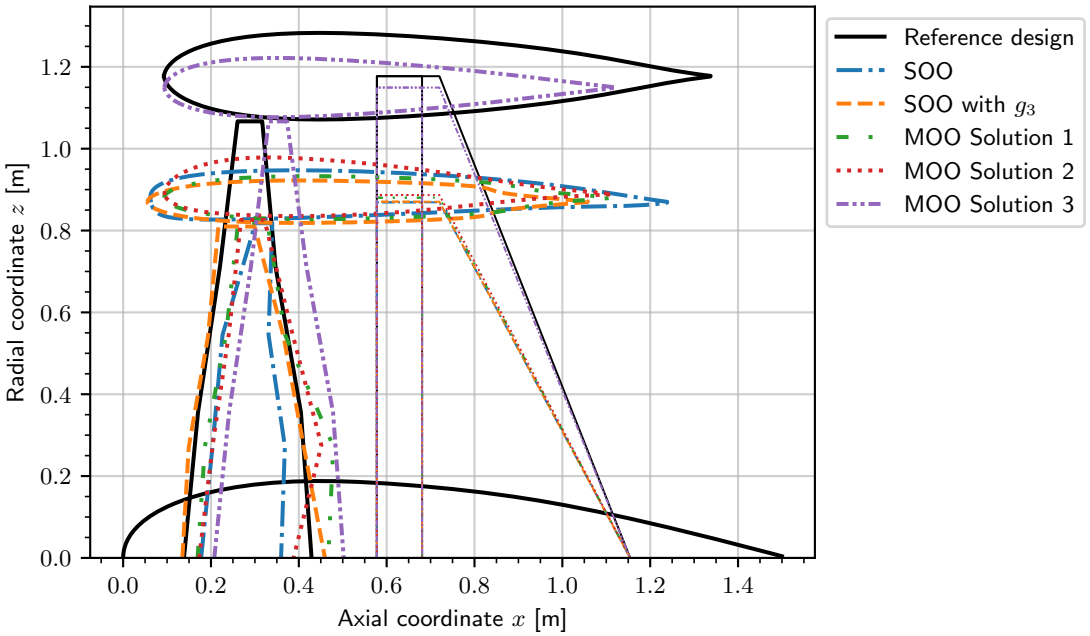


Figure 9.10: Meridional view of the optimised single-objective and multi-objective designs for the endurance cruise condition. Note the constant centrebody and constant scaled geometry of the struts.

This is further shown when comparing the blading parameters in table 9.5 and figure 9.11, with solution 3 having a similar blade count, tip blade angle, blade angle distribution, blade diameter, and sweep angle distribution compared to the reference design. The key difference is an aft shift of the fan row, a reduction in tip chord length, and a decrease in rotational rate.

Table 9.5: Comparison of the blading data obtained for the single-objective optimised designs and the most efficient multi-objective optimised non-dominated solution for the endurance cruise condition.

Parameter	Unit	Reference design	SOO	SOO g_3	MOO Solution 3
x_{root}	m	0.150	0.187	0.144	0.220
$\breve{x_{root}}$	-	0.0458	0.105	0.0884	0.123
β_{tip}	°	14.5	17.5	21.1	14.5
B	-	3	10	5	3
Ω	s^{-1}	44.0	53.0	50.9	43.2
R	m	1.067	0.818	0.810	1.067

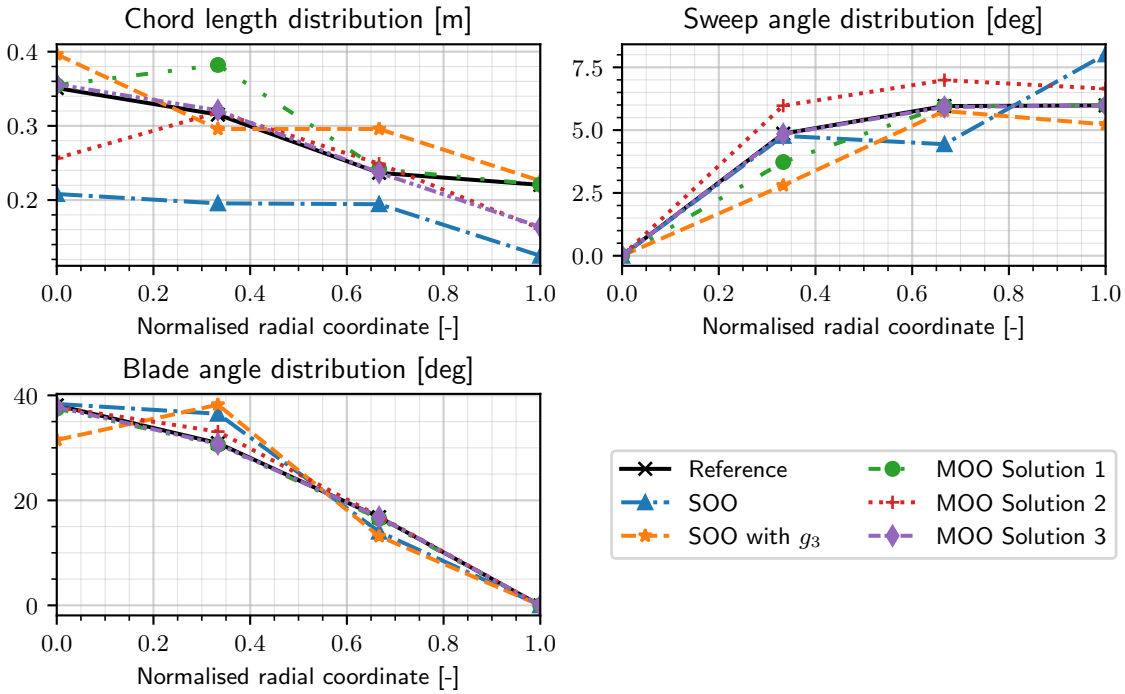


Figure 9.11: Radial distributions for each of the single-objective and multi-objective designs for the endurance cruise operating condition.

9.3. Cross-Operating Point Performance

Section 9.1 showed that significant efficiency improvements were obtained for the single-objective optimised designs. It is of interest to quantify the performance of each design at all considered operating conditions, as this shows how useful a design truly is. As such, an efficiency matrix was constructed, shown in table 9.6. The rotational rate for each design was adjusted to satisfy the thrust constraint, $g_2(\xi)$, equation (6.11), at each operating condition.

Both the endurance and combat cruise optimised designs were unable to reach the required take-off thrust level while satisfying the efficiency feasibility constraint, $g_1(\xi)$, equation (6.10). The endurance optimised design achieved an efficiency of $\eta_p = 64.07\%$ with a feasibility violation of $CV = 3.13 \times 10^{-2}$, while the combat optimised design achieved $\eta_p = 71.42\%$ with $CV = 3.78 \times 10^{-2}$. The inability of these designs to achieve the specified take-off thrust level invalidates them for use in the X-22A, as they cannot fulfil the original mission profile of the aircraft.

Table 9.6: Cross-evaluation matrix of the propulsor efficiencies obtained from the single-point, single-objective optimised designs, across all three considered flight conditions.

Operating Condition	Unit	Take-off optimised design	Endurance cruise optimised design	Combat cruise optimised design
Take-off	%	75.35	N/A	N/A
Endurance cruise	%	73.25	84.83	83.03
Combat cruise	%	59.98	83.36	85.45

When evaluating the take-off optimised design on the endurance mission profile defined in section 8.2, a normalised required energy of 105.44% of the reference design's energy is found, illustrating the poor performance. This loss in performance at other conditions, and the inability of the endurance and combat optimised designs to reach the take-off thrust level, motivates the use of a multi-point optimisation, where the relative impacts of each flight condition/flight phase can be captured to ensure a design is obtained that performs better for the complete flight profile.

10

Multi-Point Design Optimisation of the X-22A Propulsor

As shown in chapter 9, the optimised designs found for each operating condition perform poorly at the other conditions considered in the operational envelope. A design optimisation considering a (simplified) flight profile is therefore needed. This chapter presents the results obtained for the multi-point design optimisation of the X-22A propulsor for the endurance mission profile defined in section 8.2, demonstrating the capabilities of the developed optimisation framework.

Section 10.1 first discusses the key performance metrics obtained, followed by a comparison of the meridional design and fan design in sections 10.2 and 10.3. Section 10.4 then compares the outlet swirl and duct pressure distributions for the two considered operating points in the flight profile. Finally, section 10.5 discusses improvements identified for the multi-point analysis to enhance its robustness.

10.1. Comparison of the Optimised Design Performance Metrics

A single-objective, multi-point optimisation was performed to minimise the total energy required for the defined flight profile, defined by equation (8.4). This optimisation considered the same design variables defined in chapter 8, with the rotational rate being unique for each operating point. Key performance metrics for this single-objective optimisation are shown in table 10.1. The multi-point optimised design has a total energy reduction of $\Delta E = 2\%$. This reduction originates in minor improvements in both the take-off and endurance cruise propulsor efficiencies, with the take-off efficiency experiencing an improvement of $\Delta\eta_{p_{TO}} = 0.29\%$, while the endurance efficiency improved by $\Delta\eta_{p_{endurance}} = 1.27\%$. This demonstrates the logical focus of the algorithm on reducing the endurance energy required compared to the take-off energy, as the endurance cruise is the most energy-intensive phase of the flight profile.

The same computational speed observed in section 9.1 is also seen in the multi-point optimisation, with an average evaluation time per design of $22\ \mathrm{s}$, further illustrating the fast nature of the developed framework. The observed computational speed corresponds to the lower end of the range reported in section 9.1 when accounting for the two analysed operating points. This is a result of the analysis being run on the server, which, due to the higher CPU thread count, is significantly faster in evaluating the different populations.

Table 10.1: Comparison of the multi-point optimised design performance metrics to the reference design.

Parameter	Unit	Reference design	Multi-point optimised
n_{gen}	-	N/A	58
$\frac{E_{flight}}{E_{flight_{ref}}}$	%	100	98.06
$\frac{A_{front}}{A_{front_{ref}}}$	%	100	99.98
$\eta_{p_{TO}}$	%	71.05	71.34
$\eta_{p_{endurance}}$	%	79.64	80.91
$g_{1_{TO}}$	-	-1.10×10^{-2}	-9.51×10^{-3}
$g_{2_{TO}}$	-	-1.00×10^{-2}	-1.19×10^{-3}
$g_{1_{end}}$	-	-1.09×10^{-1}	-9.40×10^{-2}
$g_{2_{end}}$	-	-1.00×10^{-2}	-4.40×10^{-3}

10.2. Comparison of the Optimised Meridional Designs

A comparison of the meridional views for the multi-point optimised design with the single-point and reference designs is shown in figure 10.1. From this figure, it is clear that the multi-point optimisation hardly changed the meridional design of the ducted fan. A slight forward shift is observed for the duct, equal to $\Delta x_{LE_{duct}} = 3.7 \times 10^{-3}$ m, while the duct chord length reduced by $\Delta c_{duct} = 2.7 \times 10^{-3}$ m. The negligible changes to the meridional design of the multi-point optimised design suggest that these do not significantly contribute to the changes in performance observed for the ducted fan in table 10.1.

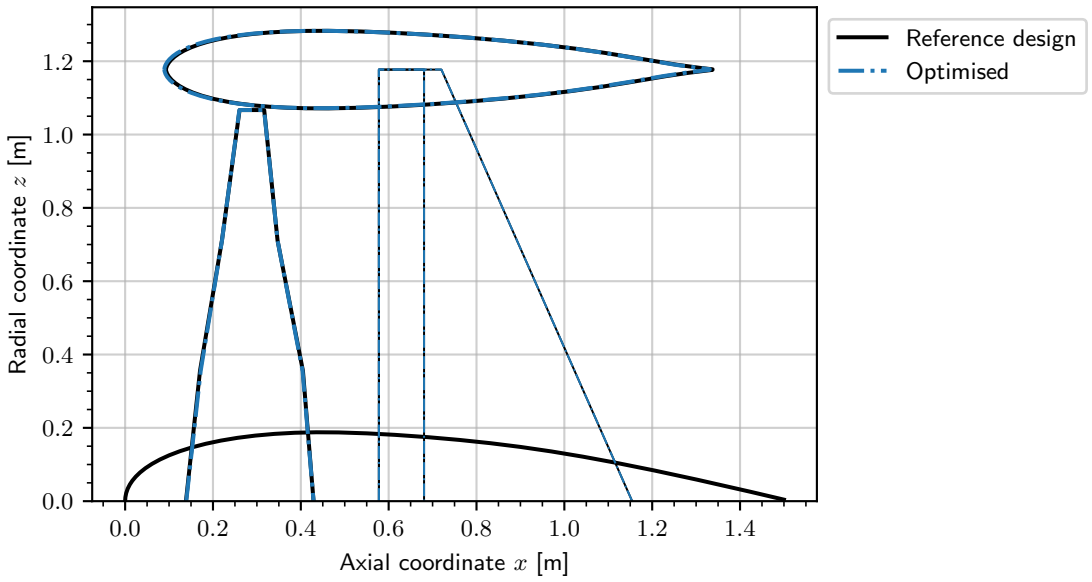


Figure 10.1: Meridional view of the multi-point optimised design compared against the reference design.

10.3. Comparison of the Optimised Fan Design

The minor changes observed for the meridional design are also present in the optimised fan design, with a maximum observed change in key fan blading data defined in tables 9.2 and 9.5 of $\Delta \leq 1 \times 10^{-3}$ compared to the reference design. Analysis of the radial distributions of chord length, sweep angle, and blade angle in figure 10.2, shows that no significant changes are made to the blade angle, and sweep angle distributions, while a reduction in tip chord length is seen of $\Delta c_{tip} = 1.2 \times 10^{-3}$ m. The only significant changes in the optimised design occurred in the blade profiles, shown in figure 10.3.

This figure shows that the camber distributions were adjusted to yield a more favourable blade slope by reducing the maximum camber at the 66% span station to reduce the mid-span losses observed in figure 9.6b. Additionally, changes to the thickness distribution were made, resulting in an unphysical blade shape at the 33% span station. The lack of design changes made by the optimisation algorithm further illustrates the algorithm's inability to move away from the reference design. Several possible solutions to address this are discussed in section 10.5.

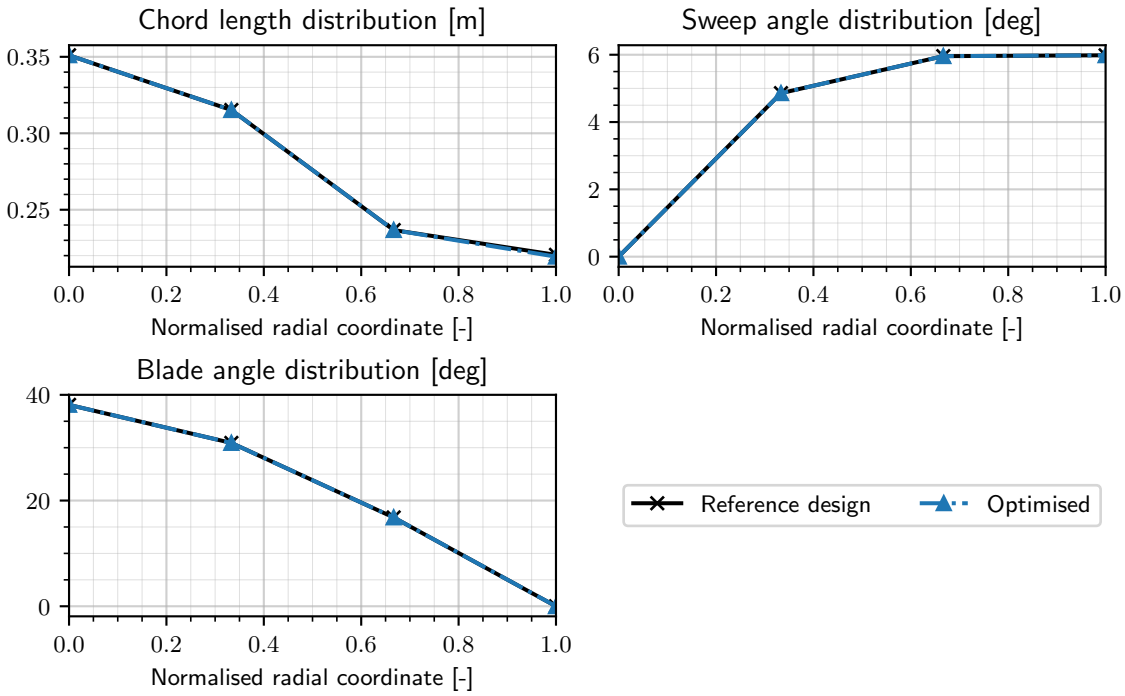


Figure 10.2: Comparison of the radial distributions for the multi-point optimised design against the reference design.

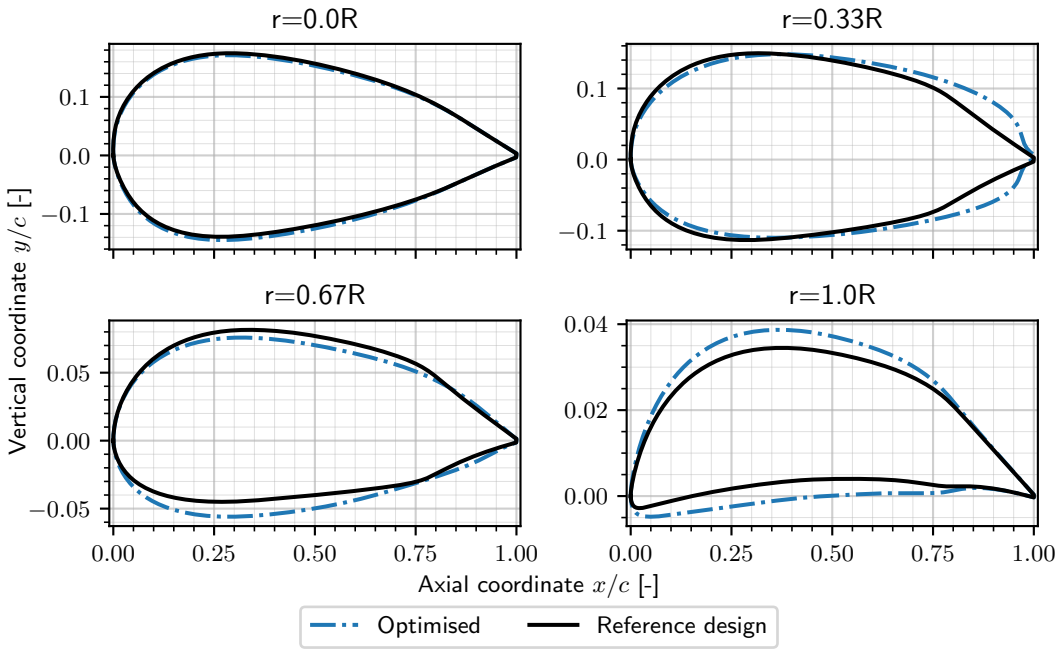
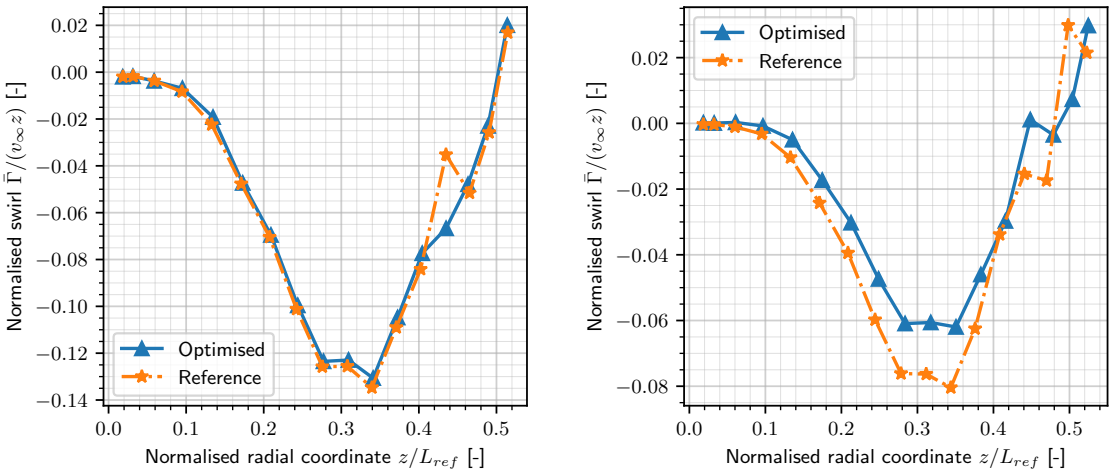


Figure 10.3: Comparison of the multi-point optimised blade profiles against the reference design.

10.4. Analysis of the Outlet Swirl and Duct Pressure Distributions

Similarly to the analysis for the single-point optimised designs, a comparison is made between the multi-point optimised and reference design normalised outlet swirl distributions $\bar{\Gamma}/(v_\infty z)$ and pressure distributions C_p over the duct geometry in the multi-point analysis. The outlet swirl distributions are shown in figure 10.4, while the pressure distributions over the duct are shown in figure 10.5.

The observed focus of the algorithm on the endurance flight phase is also seen when inspecting the outlet swirl distributions in figure 10.4, with a noticeable reduction in exit swirl for the endurance condition. In contrast, the take-off swirl distribution remained similar to the reference design, with a smoothed swirl distribution for the outboard sections of the blade span in figure 10.4a. This is caused by a different streamline distribution for the optimised design compared to the reference design, which affects the intersections of the blade stage grid with the streamline grid, resulting in the observed difference. The endurance cruise condition, on the other hand, has a noticeable reduction in exit swirl, including a decrease in tip losses due to the optimised blade profiles. This contributes to the observed improvement in efficiency at the endurance cruise condition.

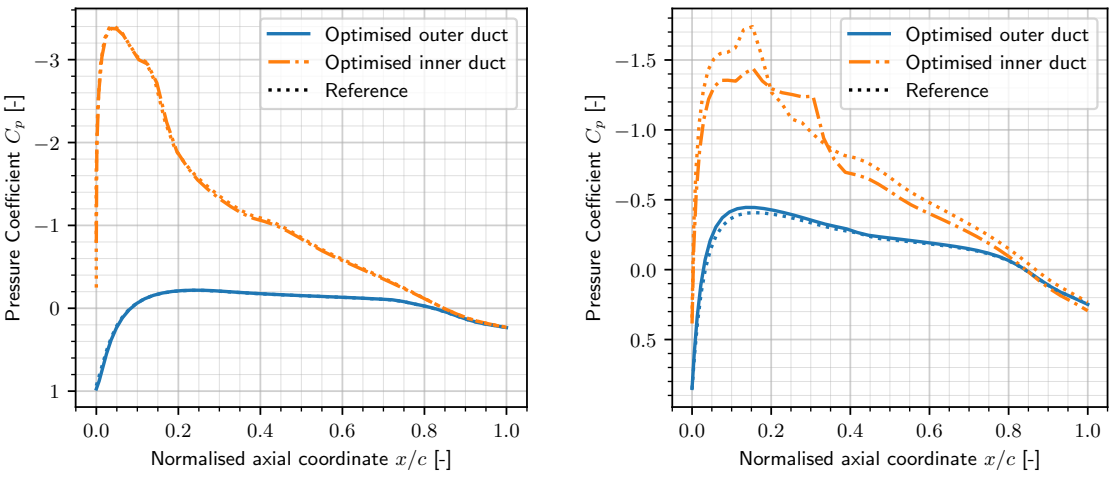


(a) Comparison of the outlet swirl distribution between the multi-point optimised and reference designs at the take-off condition.

(b) Comparison of the outlet swirl distribution between the multi-point optimised and reference designs at the endurance condition.

Figure 10.4: Comparison of the exit swirl distributions between the reference and multi-point, single-objective optimised designs. Note that only the exit swirl for the streamlines passing through the fan is shown.

When comparing the pressure distributions over the duct between the multi-point optimised and reference design in figure 10.5, a similar trend is observed. The effects of the changed blade profiles and marginal forward shift of the duct are seen for the endurance cruise condition, with a reduction in magnitude and widening of the suction peak. Due to the minor design changes made to the optimised design, the significant alterations to the pressure distributions observed in figure 9.7 are not seen here. The $dC_p/d(x/c)$ kink and high subsequent adverse pressure gradient seen near $x/c = 0.30$ for the optimised inner duct at the endurance condition corresponds to the point of maximum thickness x_t of the duct, and was not observed to result in flow separation in the converged flowfield.



(a) Comparison of the pressure distributions between the multi-point optimised and reference designs at the take-off condition.

(b) Comparison of the pressure distributions between the multi-point optimised and reference designs at the endurance condition.

Figure 10.5: Comparison of the pressure distributions between the reference and multi-point optimised designs.

10.5. Improvements to the Multi-Point Optimisation Process

As demonstrated in this chapter, although able to find an improved design, the developed optimisation framework struggles to move away from the reference design and finds an optimised design for which the improvements cannot be considered significant enough for a conceptual design stage. This is especially true when comparing the optimised multi-point design with the single-point designs for the take-off and endurance cruise conditions, which showed large changes in fan diameter, duct design, blade count, and fan blading. Inspection of the algorithm's progression during evaluation reveals that this is caused by the difficulty in satisfying the thrust constraints at both operating conditions.

The initial population generated using the methods defined in section 6.3 loses diversity quickly in the multi-point optimisation, as very few of the individuals can (nearly) satisfy both thrust constraints while improving the objective function value. As the X-22A propulsor used variable pitch blades, the fixed pitch assumption used in this thesis is identified as a shortcoming. Using a fixed pitch angle across both take-off and endurance cruise conditions is suboptimal, with significant efficiency improvements achievable at both conditions when employing variable pitch.

To address these issues for a fixed-pitch analysis, a different constraint handling technique may be used. While the optimisation framework currently uses the feasibility-first constraint handling method [67], ϵ-constraint handling, which is noted to perform well for difficult-to-satisfy constraints [52], may yield better results. This method was not implemented in this thesis due to resource limitations.

The ϵ-constraint handling approach allows the feasibility criterion to be changed over time using equation (10.1), where t is the percentage of the analysis completed based on the current generation count i and the maximum number of generations, and ε is the fraction of the maximum generation count beyond which the feasibility-first approach should be used again [52].

$$\epsilon = \max(0, 1 - \frac{t}{\varepsilon}) \cdot CV_{i_{max}} \tag{10.1}$$

Inspection of the analysis during evaluation showed a large number of individuals for whom the rotational rate was either substantially too high or too low. As such, the same internal Ω-matching loop proposed in section 9.1 is suggested to improve the diversity of the feasible population set early in the optimisation process, helping the algorithm identify improvements in the design. Finally, it is recommended that the use of different initialisation parameters ν, τ, δ, different population sizes K, and different genetic algorithm hyperparameters be investigated. These hyperparameters control, amongst others, the cross-over and mutation of the algorithm, and were left at their default values throughout this thesis.

11

Conclusions and Recommendations

In this thesis, a Unified Ducted fan Code (UDC) has been developed and validated against experimental wind tunnel data for the Bell X-22A propulsor. This UDC has subsequently been integrated into an optimisation framework using the Unified Non-dominated Sorting Genetic Algorithm III (U-NSGA-III), which was applied to the X-22A as a demonstration of the capabilities of this framework. The conclusions based on the findings of this thesis are presented in section 11.1, while the recommendations for future work are presented in section 11.2.

11.1. Conclusions

This section presents the conclusion based on the research performed in this thesis.

11.1.1. Validation of the Developed Unified Ducted Fan Code

Validation of the UDC demonstrates improved accuracy compared to experimental wind tunnel data of the Bell X-22A ducted propulsor, surpassing the accuracy of the Ducted Fan Design Code (DFDC). Validation was performed at two reference tip set angles, $\beta_{tip} = 14.5°$ and $\beta_{tip} = 24.5°$. The thrust coefficients have an average deviation from the experimental data smaller than the read-off uncertainties of the experimental data, $\Delta C_T = \Delta C_P = 0.01$, and are over $3.5\times$ as accurate as those obtained from DFDC.

The power coefficients have an average underprediction of $1.64\times$ the read-off uncertainty, with a maximum underprediction of $2.56\times$. This is attributed to the lack of loss modelling over the blade row(s) and the modifications made to the centrebody and strut geometries to simplify the modelling effort. The comparison with the DFDC further illustrates the usefulness of the UDC, as DFDC is unable to accurately model the power coefficient, with an average deviation of more than $3\times$ and a maximum deviation of over $18\times$ the uncertainty.

11.1.2. Single-Point Single-Objective Optimisation Studies

Single-point optimisations, using the developed optimisation framework, were performed for the X-22A ducted fan at take-off, endurance, and combat cruise conditions. The optimisations resulted in propulsor efficiency improvements of $\Delta\eta_p = 4.3\% - 10\%$, while maintaining thrust within 1% of the target settings. The following trends were identified as a function of the freestream Mach number and thrust setting:

- The duct chord length decreases with Mach number.
- The duct camber reduces with Mach number and thrust setting, with the locus of maximum camber shifting towards the leading edge.
- The diffuser expansion area ratio increases with increasing Mach number.

The optimised duct designs all had a changed pressure distribution, affecting the thrust contribution of the duct. A maximum increase was seen for the combat cruise condition, where the duct thrust

contribution increased by 82.2%. The residual swirl at the domain exit was found to reduce for all optimised designs, with a reduction seen in swirl magnitude at the midspan sections and the blade tip. Both of these changes contribute to the observed improvements in efficiency. At the take-off condition, the observed increase in performance originates from a reduction in the duct viscous forces, accompanied by an increased efficiency of the blade row. For the endurance and combat-optimised designs, the efficiency improvement originates from both the increase in duct thrust contribution and the improvement in blade row efficiency.

11.1.3. Single-Point Multi-Objective Optimisation Study

To investigate the effects of a second objective on the optimisation framework, the propulsor efficiency was maximised while the frontal area was minimised at the endurance cruise condition. This optimisation study showed that:

- The added complexity increased computational cost, with the multi-objective optimisation terminating at 100 generations, compared to 79 for the single-objective endurance cruise condition optimisation.
- Reference direction niching enabled the optimisation to find three unique non-dominated solutions, which varied significantly.
- The complexity of the optimisation problem is evident from the clustering of solutions in the objective space.
- Multi-objective optimisation can achieve a better propulsor efficiency and provides more insights into the design space compared to single-objective optimisation, even when a maximum frontal area constraint is used.

The multi-objective optimisation found a better efficiency by $\Delta\eta_p = 2\%$, while the frontal area is 36% higher compared to the single-objective optimisation. Still, this is 9% lower than the reference value. The inclusion of more diverse regions in the design space is a strong argument for the use of multi-objective genetic optimisation. This is expected to be particularly true when multidisciplinary optimisation is performed, where other design disciplines, such as weight estimation of the ducted fan, are also considered.

11.1.4. Multi-Point Single-Objective Optimisation Study

A multi-point, single-objective optimisation was performed to minimise the flight energy over a simplified endurance flight profile of the X-22A, resulting in a $\Delta E = 2\%$ reduction in total energy compared to the reference design. The optimised design shows no significant changes to the meridional geometry of the ducted fan, or to the radial distributions of chord length, sweep, and blade angles; however, it has different camber and thickness distributions for the blade row. The changes made to the design logically favour improvements in propulsor efficiency during the endurance cruise phase, as this is the most energy-intensive segment of the considered flight profile.

The optimised design improvement of 2% is not deemed significant, especially since the single-point optimisations reveal more substantial and fundamental changes to the ducted fan design, which partially overlap, like the increased blade count or reduced duct thickness. The lack of changes made to the design by the optimiser is attributed to its difficulty in satisfying the posed thrust constraints. This is partially caused by the fixed-pitch modelling and the absence of a physics-based repair operator for the fan rotational rates.

11.1.5. Answers to the Research Objective and the Posed Research Questions

As answers to the research questions posed at the start of this thesis, it is seen that:

1. Introducing a second objective significantly increases computational cost, but enables reference direction niching in U-NSGA-III, allowing for a better overall solution to be found and providing better insights into the design space of the ducted fan. This paves the way for future, more complex multi-disciplinary design optimisations.
2. The dominant design variables are the blade count, blade tip angle, chord length, and camber distribution of the blades. Additionally, a significant impact of the duct profile design and duct

chord length on overall performance is observed.

3. Although full CFD methods offer increased accuracy and can model viscous losses throughout the whole domain, the applied throughflow method can accurately analyse the performance of a wide range of ducted fan designs, and has a significantly higher accuracy compared to currently used inviscid, incompressible methods.

This thesis hypothesised that the duct length does not have a significant direct influence on the performance of the ducted fan. This hypothesis is rejected, as clear reductions in duct length are observed for the single-point optimised designs for the take-off, endurance cruise, and combat cruise optimised designs. No clear conclusions can be drawn on the location of the fan within the ducted fan as a function of operating conditions. Finally, the increased sensitivity of the ducted fan performance to the design of the duct leading edge compared to the trailing edge was seen, confirming the hypothesis.

The UDC and optimisation framework are significantly more accurate for conceptual design, analysis, and optimisation compared to the currently used panel method, the Ducted Fan Design Code. Although no automated physics-based design generation method has been implemented, the biased initialisation method makes the developed toolset suitable as a second step in the conceptual design process. If an initial ducted fan design is defined, the developed toolset can be reliably used to optimise the design for the given application. When future work has improved the robustness of the multi-point optimisation and implemented the recommendations of section 11.2, the developed design and analysis framework of the ducted fan can be applied to a wide range of design problems and could even be linked with the main aircraft design to yield an integrated, complete aircraft architecture, optimisation framework.

11.2. Recommendations

Based on the research performed in this thesis, key recommendations for future work are:

11.2.1 The inclusion of a blade row loss modelling method. The developed framework cannot optimise the blade thickness distributions due to a lack of loss modelling in the meridional flowfield. An external cascade solver, such as MISES or Multall, has been proposed, with MISES being a good candidate due to its capability to link with the MTFLOW software used in this thesis.

11.2.2 The implementation of a physics-based simplified design method to rapidly generate an initial population of (near-) feasible design vectors to improve the overall feasibility of the initial population. This helps to improve the performance of the developed framework and is the next step in the creation of a complete (automated) ducted fan design, analysis, and optimisation framework, for which the developed tools in this thesis form a substantial building block.

11.2.3 The development of a ducted fan weight estimation routine to enable the use of weight as an objective to enable multi-disciplinary design optimisation studies.

11.2.4 The expansion of validation of the developed framework to higher Mach numbers. Future work can validate the UDC against CFD simulations to confirm its accuracy and reliability further. It is expected that an expansion to or replacement of the used first-order blade loading model is required, as the current method assumes a two-dimensional blade section lift slope equal to the incompressible value of $2\pi \text{ rad}^{-1}$.

11.2.5 The application of a physics-based repair operator for the rotational rates of the ducted fan designs to improve robustness of the optimisation framework.

11.2.6 The use of a different constraint handling technique, ϵ-constraint handling, in the multi-point optimisation, and variable pitch blade rows, to enhance robustness. This constraint handling technique allows constraint violations to persist early in the optimisation process, helping high-performing individuals steer the algorithm toward a better objective value.

11.2.7 The use of variable-pitch rotors in the design of ducted fans to improve the propulsor efficiency at all considered operating conditions.

References

[1] GKN Aerospace Services Ltd. 'GKN Aerospace's cryogenic hyperconducting technology unlocks hydrogen electric propulsion for large aircraft'. Accessed: 08-07-2025. (Jul. 2022), [Online]. Available: https://www.gknaerospace.com/news-insights/news/gkn-aerospace-s-cryogenic-hyperconducting-technology-unlocks-hydrogen-electric-propulsion-for-large-aircraft/.

[2] T. Walker III, M. Tatsutani and J. Lewis, 'Decarbonizing Aviation: Enabling Technologies for a Net-Zero Future', Clean Air Task Force, Boston, MA, USA, Apr. 2024.

[3] J. Kos, B. Peerlings, M. Lim, W. Lammen, J. P. Duque and E. van der Sman, 'Novel propulsion and alternative fuels for aviation towards 2050', Royal NLR - Netherlands Aerospace Centre, Amsterdam, Netherlands, NLR-CR-2022-094, Sep. 2022.

[4] H. Kim, A. Perry and P. Ansell, 'A Review of Distributed Electric Propulsion Concepts for Air Vehicle Technology', in *Proc. AIAA/IEEE Electr. Aircr. Technol. Symp.*, Cincinnati, OH, USA, Jul. 2018. DOI: 10.2514/6.2018-4998.

[5] A. Esqué, F. Franz, D. Riefer, R. Riedel and G. Mulder. 'Decarbonizing aviation: Executing on net-zero goals'. Accessed: 27-11-2024. (Jun. 2023), [Online]. Available: https://www.mckinsey.com/industries/aerospace-and-defense/our-insights/decarbonizing-aviation-executing-on-net-zero-goals#/.

[6] C. Bowman, 'Visions of the Future: Hybrid Electric Aircraft Propulsion', NASA, Salt Lake City, UT, USA, Presentation, Jul. 2016. [Online]. Available: https://ntrs.nasa.gov/api/citations/20170002633/downloads/20170002633.pdf.

[7] L. Stipa, 'Experiments with Intubed Propellers', NACA, Washington, D.C., USA, Tech. Rep. NACATM655, Aug. 1931.

[8] L. Stipa, 'Stipa Monoplane with Venturi Fuselage', NACA, Washington, D.C., USA, Tech. Rep. NACA-TM-753, Sep. 1934.

[9] T. Zhang and G. Barakos, 'Review on ducted fans for compound rotorcraft', *Aeronaut. J.*, vol. 124, no. 1277, pp. 941–974, Jul. 2020. DOI: 10.1017/aer.2019.164.

[10] H. Youngren, M. Drela and S. Sanders. 'DFDC'. Accessed: 12-11-2024. (Dec. 2005), [Online]. Available: https://web.mit.edu/drela/Public/web/dfdc/.

[11] A. Abrego and R. Bulaga, 'Performance Study of a Ducted Fan System', in *Proc. AHS Aerodyn., Acoust., Test Eval. Tech. Spec. Meeting*, San Francisco, CA, USA, Jan. 2002.

[12] Y. Gao and Y. Xu, 'The Overall Design of Variable Diameter Ducted Fan in the Aircraft', *Aerospace*, vol. 9, no. 7, p. 387, 2022. DOI: 10.3390/aerospace9070387.

[13] A. Sacks and J. Burnell, 'Ducted propellers—a critical review of the state of the art', *Prog. Aerosp. Sci.*, vol. 3, pp. 85–135, 1962. DOI: 10.1016/0376-0421(62)90017-9.

[14] J. Pereira, 'Hover and Wind-Tunnel Testing of Shrouded Rotors for Improved Micro Air Vehicle Design', Ph.D. dissertation, Univ. of Maryland, College Park, MD, USA, 2008.

[15] A. Akturk and C. Camci, 'Double-Ducted Fan as an Effective Lip Separation Control Concept for Vertical-Takeoff-and-Landing Vehicles', *J. Aircraft*, vol. 59, no. 1, pp. 233–252, 2022. DOI: 10.2514/1.C036386.

[16] United Aircraft Corporation, Hamilton Standard, *Generalized Method of Shrouded Propeller Performance Estimation*. Windsor Locks, CT, USA: Hamilton Standard, 1975.

[17] G. Wright and F. de Poilenc, *Ducted Fan Design Volume 1*, 2nd ed. West Covina, CA, USA: Mass Flow, 2001.

[18] D. Black, H. Wainauski and C. Rohrbach, ‘Shrouded propellers - A comprehensive performance study’, in *Proc. 5th Annu. Meeting and Tech. Display*. Philadelphia, PA, USA, Nov. 1968. DOI: 10.2514/6.1968-994.

[19] L. Parlett, ‘Aerodynamic characteristics of a small-scale shrouded propeller at angles of attack from 0 to 90 degrees’, NACA, Langley Field, VA, USA, Tech. Rep. NACA-TN-3547, Nov. 1955.

[20] W. Graf, J. Fleming and W. Ng, ‘Improving Ducted Fan UAV Aerodynamics in Forward Flight’, in *Proc. 46th Aerosp. Sci. Meeting and Exhib.* Reno, NV, USA, Jan. 2008. DOI: 10.2514/6.2008-430.

[21] M. Biava and G. Barakos, ‘Optimisation of Ducted Propellers for Hybrid Air Vehicles Using High-Fidelity CFD’, *Aeronaut. J.*, vol. 120, no. 1232, pp. 1632–1657, Apr. 2016. DOI: 10.1017/aer.2016.78.

[22] M. Anderson, K. Lehmkuehler, D. Ho, K. Wong and P. Hendrick, ‘Propeller Location Optimisation for Annular Wing Design’, in *Proc. Int. Micro Air Vehicle Conf. and Flight Competition*, Toulouse, France, Sep. 2013.

[23] N. Bi, K. Kimmel and D. Haas, ‘Performance Investigation of Ducted Aerodynamic Propulsors’, in *Proc. 1st Int. Symp. Marine Propulsors*, Trondheim, Norway, Jun. 2009.

[24] J. Guo and Z. Zhou, ‘An Efficient Blade Design Method of a Ducted Fan Coupled with the CFD Modification’, *Aerosp.*, vol. 9, no. 5, p. 241, Apr. 2022. DOI: 10.3390/aerospace9050241.

[25] J. Mehr and A. Ning, ‘DuctAPE: A Steady-State, Axisymmetric Ducted Fan Analysis Code Designed for Gradient-Based Optimization’, presented at the AIAA Aviation Forum and Ascend, Las Vegas, NV, USA, Jul. 2024. DOI: 10.2514/6.2024-4297.

[26] S. Dixon and C. Hall, *Fluid Mechanics and Thermodynamics of Turbomachinery*, 7th ed. London, U.K.: Butterworth, 2014.

[27] M. Gambini and M. Vellini, *Turbomachinery Fundamentals, Selection and Preliminary Design* (Springer Tracts Mech. Eng.), 1st ed. New York, NY, USA: Springer, 2021.

[28] J. Denton, ‘Multall — An Open Source, Computational Fluid Dynamics Based, Turbomachinery Design System’, *J. Turbomach.*, vol. 139, no. 12, p. 121 001, Sep. 2017. DOI: 10.1115/1.4037819.

[29] T. Zhang and G. Barakos, ‘High-Fidelity CFD Validation and Assessment of Ducted Propellers for Aircraft Propulsion’, *J. Am. Helicopter Soc.*, vol. 66, no. 1, pp. 1–28, Jan. 2021. DOI: 10.4050/jahs.66.012008.

[30] M. Drela and H. Youngren, ‘Axisymmetric Analysis and Design of Ducted Rotors’, MIT Computational Design laboratory, Cambridge, MA, USA, Dec. 2005.

[31] P. Sargent and W. Anemaat, ‘Benchmarking a Robust Panel Code for Ducted Fan VTOL Aircraft Design’, in *Proc. Appl. Aerodyn. Conf.*, Atlanta, GA, USA, Jul. 2018. DOI: 10.2514/6.2018-4212.

[32] S. Acarer and U. Özkol, ‘An extension of the streamline curvature through-flow design method for bypass fans of turbofan engines’, *Proc. Inst. Mech. Eng. G: J. Aerosp. Eng.*, vol. 231, no. 2, pp. 240–253, 2017. DOI: 10.1177/0954410016636159.

[33] N. Cumpsty, *Compressor Aerodynamics*, 5th ed. Malabar, FL, USA: Krieger, 2004.

[34] M. Casey and C. Robinson, ‘A New Streamline Curvature Throughflow Method for Radial Turbomachinery’, *J. Turbomach.*, vol. 132, no. 3, p. 031 021, Apr. 2010. DOI: 10.1115/1.3151601.

[35] J. Denton, ‘Throughflow Calculations for Transonic Axial Flow Turbines’, *J. Eng. Power*, vol. 100, no. 2, pp. 212–218, Apr. 1978. DOI: 10.1115/1.3446336.

[36] P. Tiwari, A. Stein and Y. Lin, ‘Dual-Solution and Choked Flow Treatment in a Streamline Curvature Throughflow Solver’, *J. Turbomach.*, vol. 135, no. 4, p. 041 004, Jun. 2013. DOI: 10.1115/1.4007444.

[37] *A User’s Guide to MTFLOW 2.01 - Multi-passage ThroughFLOW Design/Analysis Program*, MIT Computational Design laboratory, Cambridge, MA, USA, Jun. 2010.

[38] D. Renick, ‘An Analysis Procedure for Advanced Propulsor Design’, M.S. thesis, MIT, Cambridge, MA, USA, May 1999.

[39] C. Hanson, 'Integrated Lifting-Surface and Euler/Boundary-Layer Theory Analysis Method for Marine Propulsors', M.S. thesis, MIT, Cambridge, MA, USA, Feb. 2001.

[40] M. Drela, M. Giles and W. Thompkins Jr., 'Conservative streamtube solution of steady-state euler equations', in *Proc. 17th Fluid Dynamics, Plasma Dynamics, and Lasers Conf.*, Snowmass, CO, USA, Jun. 1984. DOI: 10.2514/6.1984-1643.

[41] R. Novak, 'Streamline Curvature Computing Procedures for Fluid-Flow Problems', *J. Eng. Power*, vol. 89, no. 4, pp. 478–490, Oct. 1967. DOI: 10.1115/1.3616716.

[42] M. Drela, M. Giles and W. Thompkins Jr., 'Newton solution of direct and inverse transonic Euler equations', in *Proc. 7th Comput. Phys. Conf.*, Cincinnati, OH, USA, Jul. 1985. DOI: 10.2514/6.1985-1530.

[43] J. Thompson, F. Thames and C. Mastin, 'Automatic numerical generation of body-fitted curvilinear coordinate system for field containing any number of arbitrary two-dimensional bodies', *J. Comput. Phys.*, vol. 15, no. 3, pp. 299–319, Jul. 1974. DOI: 10.1016/0021-9991(74)90114-4.

[44] J. Anderson, *Fundamentals of Aerodynamics*, 6th ed. New York, NY, USA: Mcgraw-Hill, Apr. 2016.

[45] *A User's Guide to MSES 3.05*, MIT Department of Aeronautics and Astronautics, Cambridge, MA, USA, Jul. 2009.

[46] *A User's Guide to MISES 2.63*, MIT Computational Design laboratory, Cambridge, MA, USA, Feb. 2008.

[47] J. Sobieszczanski-Sobieski, A. Morris and M. van Tooren, *Multidisciplinary Design Optimization supported by Knowledge Based Engineering*, 1st ed. Chichester, U.K.: Wiley, 2015.

[48] P. Olver, *Introduction to Partial Differential Equations* (Undergrad. Texts in Math.), 1st ed. Cham, Switzerland: Springer, Nov. 2013.

[49] S. Katoch, S. Chauhan and V. Kumar, 'A review on genetic algorithm: Past, present, and future', *Multimed. Tools Appl.*, vol. 80, no. 5, pp. 8091–8126, Feb. 2021. DOI: 10.1007/s11042-020-10139-6.

[50] H. Seada and K. Deb, 'A Unified Evolutionary Optimization Procedure for Single, Multiple, and Many Objectives', *IEEE Trans. Evol. Comput.*, vol. 20, no. 3, pp. 358–369, Jul. 2016. DOI: 10.1109/TEVC.2015.2459718.

[51] M. Emmerich and A. Deutz, 'A tutorial on multiobjective optimization: Fundamentals and evolutionary methods', *Nat. Comput.*, vol. 17, no. 3, pp. 585–609, May 2018. DOI: 10.1007/s11047-018-9685-y.

[52] J. Blank and K. Deb, 'Pymoo: Multi-Objective Optimization in Python', *IEEE Access*, vol. 8, pp. 89 497–89 509, Apr. 2020. DOI: 10.1109/ACCESS.2020.2990567.

[53] T. Rogalsky, 'Acceleration of differential evolution for aerodynamic design', Ph.D. dissertation, University of Manitoba, Winnipeg, MB, Canada, Mar. 2004.

[54] A. Kharal and A. Saleem, 'Neural networks based airfoil generation for a given Cp using Bezier–PARSEC parameterization', *Aerosp. Sci. Technol.*, vol. 23, no. 1, pp. 330–344, 2012. DOI: 10.1016/j.ast.2011.08.010.

[55] R. Derksen and T. Rogalsky, 'Bezier-PARSEC: An optimized aerofoil parameterization for design', *Adv. Eng. Softw.*, vol. 41, no. 7, pp. 923–930, 2010. DOI: 10.1016/j.advengsoft.2010.05.002.

[56] M. Gadalla, M. Tezzele, A. Mola and G. Rozza, 'BladeX: Python Blade Morphing', *J. Open Source Softw.*, vol. 4, no. 34, p. 1203, 2019. DOI: 10.21105/joss.01203.

[57] H. Youngren and M. Drela, 'Viscous/inviscid method for preliminary design of transonic cascades', in *Proc. 27th Joint Propuls. Conf.*, Sacramento, CA, USA, Jun. 1991. DOI: 10.2514/6.1991-2364.

[58] T. Vermeulen, *Unified Ducted fan Code and Optimisation Algorithm*, Archived GitHub repository, Accessed: 18-08-2025, Aug. 2025. [Online]. Available: https://github.com/TSVermeulen/AircraftDuctedFanOptimisation.

[59] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, 1st ed. New York, NY, USA: Wiley, 2001.

[60] A. Chehouri, R. Younes, J. Perron and A. Ilinca, 'A Constraint-Handling Technique for Genetic Algorithms using a Violation Factor', *J. Comput. Sci.*, vol. 12, no. 7, pp. 350–362, Jul. 2016. DOI: 10.3844/jcssp.2016.350.362.

[61] K. Deb, K. Sindhya and T. Okabe, 'Self-adaptive simulated binary crossover for real-parameter optimization', in *Proc. 9th Annu. Conf. Genet. Evol. Comput.*, London, U.K., 2007, pp. 1187–1194. DOI: 10.1145/1276958.1277190.

[62] I. Das and J. Dennis, 'Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems', *SIAM J. Optim.*, vol. 8, no. 3, pp. 631–657, 1998. DOI: 10.1137/S1052623496307510.

[63] J. Blank, K. Deb, Y. Dhebar, S. Bandaru and H. Seada, 'Generating Well-Spaced Points on a Unit Simplex for Evolutionary Many-Objective Optimization', *IEEE Trans. Evol. Comput.*, vol. 25, no. 1, pp. 48–60, 2021. DOI: 10.1109/TEVC.2020.2992387.

[64] D. Hardin and E. Saff, 'Minimal Riesz energy point configurations for rectifiable d-dimensional manifolds', *Adv. Math.*, vol. 193, no. 1, pp. 174–204, 2005. DOI: 10.1016/j.aim.2004.05.006.

[65] *Manual of the ICAO Standard Atmosphere - extended to 80 kilometres / 262,500 feet*, 3rd ed., Doc 7488, ICAO, Montréal, Quebec, Canada, 1993.

[66] J. Ruijgrok, *Elements of airplane performance*, 2nd ed. Delft, Netherlands: VSSD, Aug. 2009.

[67] K. Deb, 'An efficient constraint handling method for genetic algorithms', *Comput. Methods Appl. Mech. Eng.*, vol. 186, no. 2, pp. 311–338, 2000. DOI: 10.1016/S0045-7825(99)00389-8.

[68] J. Blank and K. Deb, 'A Running Performance Metric and Termination Criterion for Evaluating Evolutionary Multi- and Many-objective Optimization Algorithms', in *Proc. IEEE Congr. Evol. Comput.*, Glasgow, U.K., 2020, pp. 1–8. DOI: 10.1109/CEC48606.2020.9185546.

[69] H. Ishibuchi, H. Masuda, Y. Tanigaki and Y. Nojima, 'Modified Distance Calculation in Generational Distance and Inverted Generational Distance', in *Evol. Multi-Criterion Optim.*, Cham, Switzerland: Springer, Mar. 2015, pp. 110–125. DOI: 10.1007/978-3-319-15892-1_8.

[70] C. C. Coello and M. R. Sierra, 'A Study of the Parallelization of a Coevolutionary Multi-objective Evolutionary Algorithm', in *Proc. MICAI 2004, Adv. Artif. Intell.*, Berlin, Heidelberg: Springer, 2004, pp. 688–697. DOI: 10.1007/978-3-540-24694-7_71.

[71] K. Mort and B. Gamse, 'A wind-tunnel investigation of a 7-foot-diameter ducted propeller', NASA, Moffett Field, CA, USA, Tech. Rep. NASA-TN-D-4142, Aug. 1967.

[72] J. Miller, *The X-Planes X-1 to X-45*, 3rd ed. Hinckley, U.K.: Midland Publ., Dec. 2001.

[73] J. Achton, 'Model X-22A demonstration planning and progress report', Bell Aerosystems, Buffalo, NY, USA, Tech. Rep. 2127-931001T, Oct. 1967.

[74] T. Jiang, Y. Liu and Y. Zheng, 'Ducted Fan Design for VTOL Aircraft Flight Mission Based on Bayesian Optimization', *J. Aircr.*, vol. 62, no. 3, pp. 660–671, May 2025. DOI: 10.2514/1.C037541.

[75] *Utility Flight Manual X-22A V/STOL Research Aircraft*, Publication No. 2127950012, Bell Aerosystems Company, Buffalo, NY, USA, Mar. 1969.

[76] H. Marquardt, 'X-22A Progress Report No, 73 Final (April-December 1969) (Including Flight Test Summary)', Buffalo, NY, USA, Tech. Rep. 2127933073, Apr. 1970.

[77] D. Adams, *The Ultimate Hitchhiker's Guide to the Galaxy The Complete Trilogy in Five Parts*, 42nd anniversary edition. London, U.K.: Pan Books, Oct. 2020.

[78] P. Lolis, 'Development of a Preliminary Weight Estimation Method for Advanced Turbofan Engines', Ph.D. dissertation, Cranfield University, Cranfield, U.K., Jul. 2014.

[79] N. van den Dungen, 'Synthesis of an Aircraft Featuring a Ducted-Fan Propulsive Empennage', M.S. thesis, Delft University of Technology, Delft, Netherlands, Apr. 2017.

A

Multi-Passage ThroughFLOW

This appendix provides additional information on the Multi-passage ThroughFlow software used in this thesis. Appendix A.1 first describes the boundary conditions and constraints in MTFLOW. This is followed by a description of the viscous flow modelling in appendix A.2. Appendix A.3 then describes the methods used to compute the thrust power coefficients T_C, P_C. Next, appendix A.4 presents the data flow diagram of MTFLOW. Finally, appendix A.5 presents the compiler flags used to compile the MTFLOW software for use in this thesis. The equations described in this appendix are copied from *A User's Guide to MTFLOW 2.01 - Multi-passage ThroughFLOW Design/Analysis Program* [37] and provided as reference material in this thesis.

A.1. Boundary Conditions and Constraints

Three types of outer boundary conditions exist in MTFLOW: a solid wall, an unbounded flow (subsonic source and doublet or a supersonic wave), and a constant pressure, implemented through a jet boundary. In the context of this thesis, the unbounded flow boundary condition is the most relevant. For a subsonic flow, this boundary condition sets the pressure on the outermost boundary to that of a point source and doublet, positioned on the symmetry axis [37].

The MTSOL solver computes the strengths of the source and doublet to minimise the deviation between the numerically computed streamline pattern and the analytically derived pattern. For a supersonic flow, the local Prandtl-Meyer relation is enforced to ensure outgoing shockwaves are absorbed without significant reflection. The source and doublet are positioned at the x coordinate corresponding to the volumetric centroid of the axisymmetric meridional geometries during the initial grid generation in MTSET.

MTFLOW defines a total of seven pairs of global variables and constraints. To enforce a well-posed problem for an unbounded flow boundary condition, it is required to explicitly define the mass flow using the inlet Mach number M_{inl}. For a ducted fan, the presence of internal airfoil elements requires using leading-edge and trailing-edge Kutta conditions. If the trailing-edge Kutta condition is not imposed, the flow will curl, causing the duct's trailing streamline to shift to one side of the element. If the leading-edge Kutta condition is not imposed, the leading-edge stagnation point will end up at the wrong location [37].

If the flow through the ducted fan is choked, the mass flow can no longer be computed using the inlet Mach number, M_{inl}, in equations (3.7) and (3.12). Instead, it must be defined using the normalised mass-average pressure ratio at the outlet plane, $p_{out}/p_{0_{inl}}$. MTSOL automatically switches the variable and constraint selection should this occur. This pressure ratio should therefore be input if there is a chance of choking occurring for the analysed operating conditions. In this thesis, it is assumed that no choking occurs, such that this step is not required.

A.2. Viscous Flow Modelling

This section presents the viscous equations being solved for the axisymmetric meridional surfaces in MTFLOW. The viscous equations are strongly coupled to the inviscid main flow through the displacement thickness, δ^* [37]. The displacement of the axisymmetric streamlines using this displacement thickness to yield the effective body is shown in figure A.1. This effective body is then used to solve the inviscid Euler equations using the methods described in section 3.5.

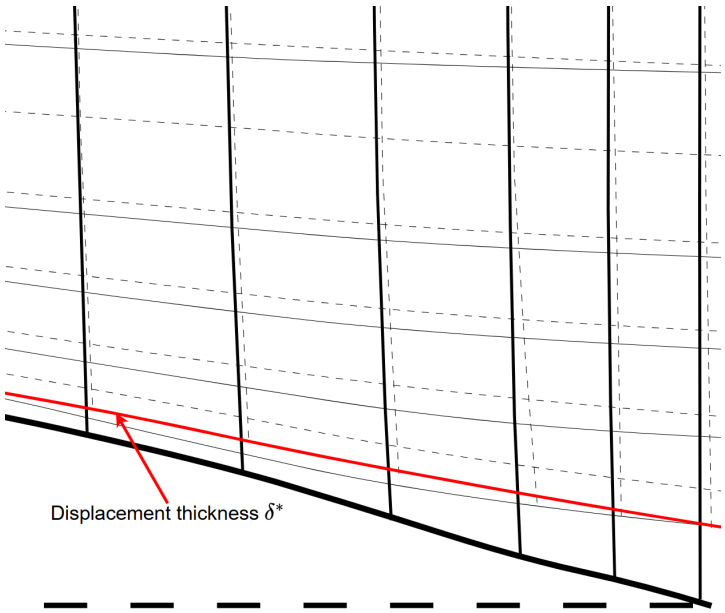


Figure A.1: Inviscid and viscous streamline positions obtained using the displacement thickness δ^*. The elliptically-generated inviscid streamlines are shown as solid lines, while the displaced viscous streamlines are shown as dashed lines. Reproduced based on an original by Renick [38].

Since MTFLOW solves an axisymmetric flowfield, the tangential boundary layer velocity component is fully decoupled. It is therefore ignored inside the boundary layer, such that only the meridional component of the velocity is used [37]. MTFLOW defines the edge velocity, edge total enthalpy, edge density, and edge viscosity along the edge of the boundary layer, which are directly related to the inviscid flowfield properties in the streamtube closest to the boundary layer.

Along the boundary layers and associated axisymmetric wakes, MTFLOW computes the momentum thickness θ, displacement thickness δ^*, and shear stress coefficient C_τ. These variables, in turn, define the secondary viscous flow variables, being the edge Mach number M_e in equation (A.1), momentum-thickness Reynolds number Re_θ in equation (A.2), shape parameter H in equation (A.3), kinetic-energy shape parameter H^* in equation (A.4), density flux shape parameter H^{**} in equation (A.5), skin friction coefficient C_f in equation (A.6), and dissipation coefficient $C_\mathcal{D}$ in equation (A.7) [37]. In this dissipation coefficient, $\mathcal{D}$ is the dissipation integral, which is the total dissipation of energy as a result of the viscosity in the boundary layer per unit of time per unit of area, expressed as equation (A.8) [45], where z_e is the distance to the edge of the boundary layer beyond which the integrant is zero.

$$M_e = \sqrt{\frac{v_e^2}{(\kappa-1)(h_{0_e} - \frac{1}{2}v_e^2)}} \qquad \text{(A.1)} \qquad\qquad Re_\theta = \frac{\rho_e v_e \theta}{\mu_e} \qquad \text{(A.2)}$$

$$H = \frac{\delta^*}{\theta} \qquad \text{(A.3)} \qquad H^* = \frac{\theta^*}{\theta} \qquad \text{(A.4)} \qquad H^{**} = \frac{\delta^{**}}{\theta} \qquad \text{(A.5)}$$

$$C_f = \frac{\tau_w}{\frac{1}{2}\rho_e v_e^2} \qquad \text{(A.6)} \qquad C_\mathcal{D} = \frac{\mathcal{D}}{\rho_e v_e^3} \qquad \text{(A.7)} \qquad \mathcal{D} = \int_0^{z_e} \tau_w \frac{\partial v}{\partial z} dz \qquad \text{(A.8)}$$

The primary viscous variables are defined by the integral momentum equation, given by equation (A.9), and the kinetic energy equation, given by equation (A.10). Additionally, a shear-stress coefficient evolution equation is used, defined by equation (A.11) [37]. Near the meridional surfaces, the local

effective circumference b includes a radial offset due to the local displacement thickness, such that $b = 2\pi z - BT_\theta + \pi\delta^*\hat{n}\hat{z}$. In the symmetry-axis-body wake, this circumference is $b = \pi\delta^*$.

$$\frac{d\theta}{\theta} = \frac{C_f}{2}\frac{ds}{\theta} - (H+2)\frac{du_e}{u_e} - \frac{d\rho_e}{\rho_e} - \frac{db}{b} \tag{A.9}$$

$$\frac{dH^*}{H^*} = \left(\frac{2C_{\mathcal{D}}}{H^*} - \frac{C_f}{2}\right)\frac{ds}{\theta} + \left(H - 1 - \frac{2H^{**}}{H^*}\right)\frac{du_e}{u_e} \tag{A.10}$$

$$\frac{dC_\tau}{C_\tau} = K_c\left(C_{\tau_{EQ}}^{1/2} - C_\tau^{1/2}\right)\frac{ds}{\delta} + 2\left(\frac{\delta}{u_e}\frac{du_e}{ds}\right)_{EQ}\frac{ds}{\delta} - 2\frac{du_e}{u_e} \tag{A.11}$$

In the integral momentum equation, all differentials $d()/()$ are equivalent to $d(\rho_e v_e^2\theta b)/(\rho_e v_e^2\theta b)$. This ensures that the total momentum defect area remains a conserved variable. Similarly, the kinetic energy equation and shear-stress coefficient evolution equation conserve the kinetic energy defect area, $\rho_e v_e^3\delta^*\hat{n}\cdot\hat{z}$. Equations (A.9) to (A.11) are discretised and solved together with the inviscid flowfield equations if a viscous solve is performed in MTFLOW.

A.3. Definition of the Thrust and Power Coefficients

This section describes the equations used to compute the thrust and power coefficients T_C and P_C. The thrust coefficient is calculated through integration of the momentum change along the x-axis over all streamtubes as equation (A.12) [37].

$$T_C = \frac{2}{\rho_{inl}v_{inl}^2 L_{ref}^2}\left(\sum_j (v_{+\infty_j} - v_{-\infty_j})\dot{m}_j - \sum_{BL}(\rho_e v_e^2 2\pi z\theta)_{+\infty}\right) = T_{C_{inviscid}} + T_{C_{viscous}} \tag{A.12}$$

In this equation, $v_{\pm\infty}$ is the downstream/upstream streamtube velocity, if the flow were isentropically taken to the reference pressure p_{inl}, as defined by equation (A.13). In this equation, the stagnation quantities are taken from the grid inlet/outlet to obtain $v_{\pm\infty}$. This yields the logical result that an inviscid thrust or drag force can only result from a total pressure and or total enthalpy change from the inlet to the outlet of the computational domain. The density ρ_e is the density at the edge of the boundary layer, while v_e is the edge velocity and θ the momentum thickness.

$$v_{\pm\infty} = \left(2h_0\left[1 - \left(\frac{p_{inl}}{p_0}\right)^{\frac{\kappa-1}{\kappa}}\right]\right)^{\frac{1}{2}} \tag{A.13}$$

$$P_C = \frac{2}{\rho_{inl}v_{inl}^3 L_{ref}^2}\left(\sum_j (h_{0_{+\infty j}} - h_{0_{-\infty j}})\dot{m}_j\right) \tag{A.14}$$

The summation of the body wakes in equation (A.12) is computed using the Squire-Young relation to the grid-outlet viscous wake point. This relation extrapolates the wake momentum deficit to far downstream, where p_{inl} dominates, such that equation (A.15) holds. The boundary layer shape factor H_{out} is the shape parameter at the outlet plane, while $H_{+\infty}$ is the shape parameter in the far downstream wake, taken as $H_{+\infty} = 1 + (\kappa-1)M_\infty^2$ [37]. This implicitly assumes an adiabatic wall at near-unity Prandtl numbers.

$$(\rho_e v_e^2 2\pi z\theta)_{+\infty} = (\rho_e v_e^2 2\pi z\theta)_{out}\cdot\left(\frac{p_{inl}}{p_{e_{out}}}\right)^{(H_{out}+H_{+\infty})/(2\kappa M_{inl}^2)} \tag{A.15}$$

The power coefficient is found by integrating the total enthalpy change over all streamtubes, as given by equation (A.14). This uses the same adiabatic wall assumption, such that the enthalpy defect thickness δ_H of the viscous boundary layer is zero everywhere and therefore does not contribute. The axial force coefficients for the meridional surfaces, i.e. the centrebody and duct, are found through surface force integrations as equation (A.16), where the integral is taken counter-clockwise around each element. This is equivalent to summing the pressure and viscous contributions, as shown on the right-hand side.

$$T_{C_{element}} = \frac{2}{\rho_{inl} v_{inl}^2 L_{ref}^2} \oint 2\pi z(pdz - \tau dx) = C_{T_{p,element}} + C_{T_{f,element}} \tag{A.16}$$

A.4. Data Flow Diagram

This section presents the data flow diagram for the Multi-passage ThroughFLOW (MTFLOW) software, as given by [37], in figure A.2. This data flow diagram also shows the optional integration of MTFLOW with the MISES cascade solver to accurately resolve the flowfield and losses through the blade row(s). To execute MTFLOW without the link with the MISES cascade solver, the walls.xxx and tflow.xxx input files are required, while the state file tdat.xxx is generated by MTFLOW. The file extension, xxx, for the input files should correspond to the analysis name used within MTFLOW.

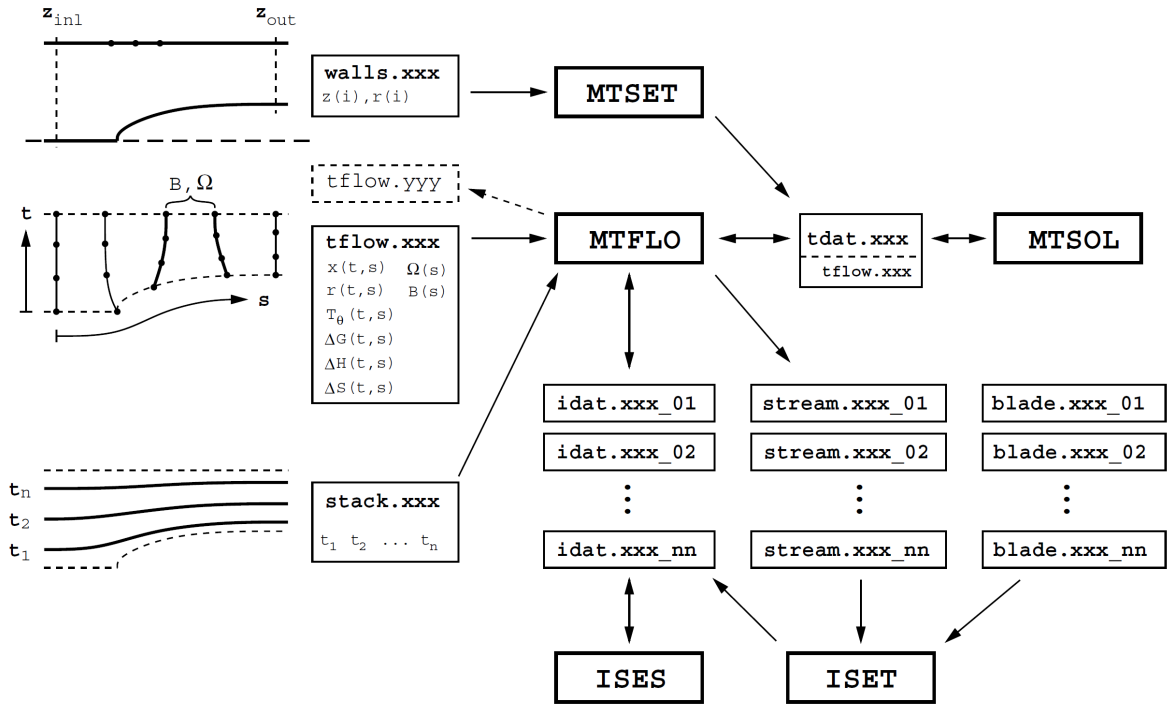


Figure A.2: Data flow diagram of the Multi-passage ThroughFLOW software, including the optional linking with the MISES cascade solver programs, ISES and ISET [37].

A.5. Compilation of the Multi-Passage ThroughFLOW Software

As mentioned in this thesis, no source code changes were made to the original MTFLOW software. To compile the MTFLOW software, the gfortran compiler was used, with the following makefile parameters enabled compared to the default file provided with the MTFLOW source code:

- **-fdefault-real-8**: Enable double precision.
- **-fbounds-check -finit-real=inf -ffpe-trap=invalid,zero -fdollar-ok**: Enables various checking routines.
- **linchk.o**: Enables the use of linearisation checking routines.

B

The Riesz s-Energy Reference Direction Generation Method

This appendix describes the Riesz s-energy reference direction creation method developed by Blank, Deb, Dhebar *et al.* [63]. For an s-dimensional space, the generalised Riesz potential energy between two points $\mathbf{z}_i$ and $\mathbf{z}_j$ is defined as equation (B.1). The so-called potential energy in this method is based on the phenomenon in nature where multi-body systems eventually settle into a state of minimum overall potential energy, and is proportional to the inverse distance between the two reference points.

The dimension s is defined to be dependent on the number of objectives of the optimisation problem as $s = M^2$ based on trial-and-error studies. For n points, the overall s-energy is then defined by equation (B.2). The method then finds the $\mathbf{z}$-matrix of size $n \times M$ which minimises the s-energy U_T while ensuring that every $\mathbf{z}_i$ vector lies on the unit simplex, i.e. satisfying equation (B.3). In Pymoo, a gradient-based optimisation method is then used to find this $\mathbf{z}$-matrix, where the gradients are projected back onto the unit simplex [52], [63].

$$U(\mathbf{z}_i, \mathbf{z}_j) = \frac{1}{||\mathbf{z}_i - \mathbf{z_j}||^s} \tag{B.1}$$

$$U_T(\mathbf{z}) = \frac{1}{2} \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} \frac{1}{\|\mathbf{z}_i - \mathbf{z}_j\|^s}, \quad \mathbf{z} \in \mathbb{R}^{n \times M} \tag{B.2}$$

$$\sum_{m=1}^{M} z_i^m = 1 \tag{B.3}$$

A comparison between the structured Das and Dennis method and the Riesz s-energy method, for the reference directions of an arbitrary three-objective optimisation problem, is shown in figure B.1 for a total of 105 reference directions. Since the Das and Dennis method requires integer partitions, it is less flexible compared to the energy method and can either generate 91 reference directions using 12 partitions, or 105 reference directions using 13 partitions.

The limitations of the Das and Dennis method are further shown when comparing the found number of reference directions as a function of the number of objectives in figure B.2. The Riesz s-energy method can generate a constant number of uniformly distributed reference directions for $m \geq 2$, while the Das and Dennis method relies on the partition number, and is seen to have an exploding number of reference directions as the number of objectives increases, where it is known that only very few of these directions lie in the simplex interior [63].

Within this thesis, at most two objectives are used. For such low objective counts, the difference in reference directions between the two methods remains small. However, the Riesz s-energy method is implemented as preparation for future, higher objective count optimisations. This contributes to the general, widely applicable nature of the optimisation framework developed in this thesis.

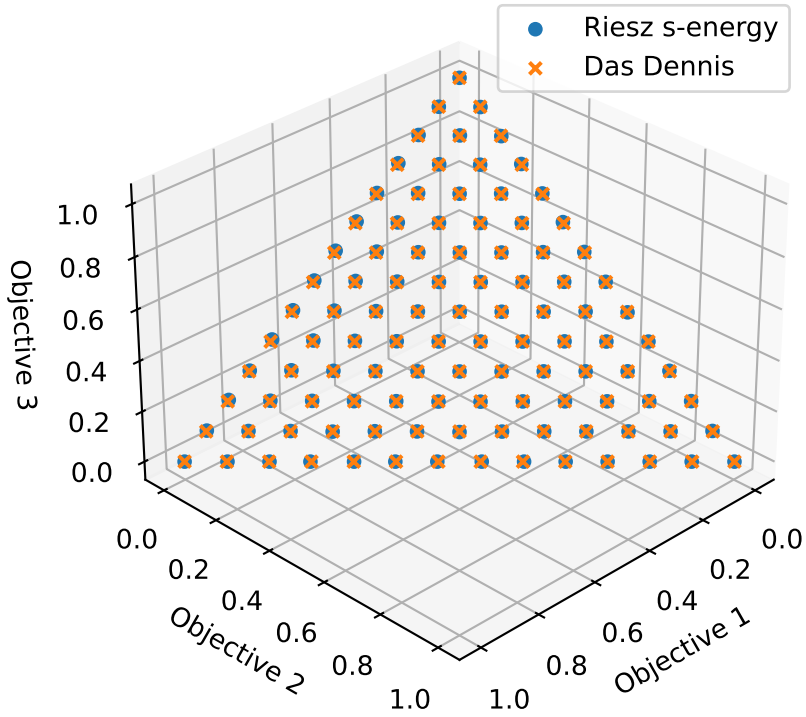


Figure B.1: Comparison of 105 reference directions for three objectives, generated using the Riesz s-energy method and the structured Das and Dennis method.

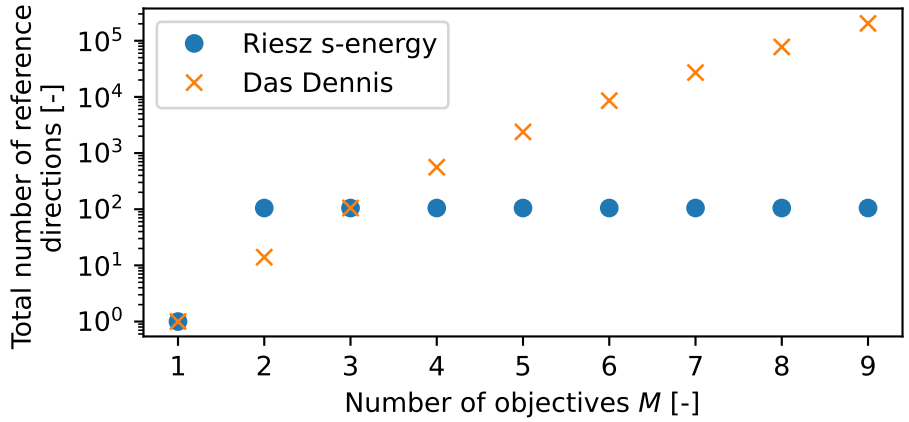


Figure B.2: Comparison of the number of reference directions as a function of the number of objectives of the optimisation problem. The Das and Dennis method is used with a constant 13 partitions, while the Riesz s-energy uses a fixed (maximum) of 105 reference directions. Note that for a single objective, both methods yield a single reference direction.

C

Implementation of the Developed Ducted Fan Optimisation Framework

This appendix discusses the code implementation of the ducted fan optimisation framework and the Unified Ducted fan Code (UDC) developed in this thesis. All developed code and tools in this thesis are written using Object-Oriented Programming (OOP) in Python and are publicly available in reference [58]. Appendix C.1 first describes the implementation of the UDC, followed by a description of the implemented ducted fan optimisation framework in appendix C.2.

C.1. Implementation of the Developed Unified Ducted Fan Code

The developed UDC can be subdivided into four steps: input processing, grid generation, solving, and output processing, as shown in figure C.1. The input handling step is responsible for converting the parameterised geometry, defined using Python dictionaries, into the MTSET and MTFLO input files, walls.xxx and tflow.xxx. These files are also shown in figure A.2. The automated grid generation, developed in section 5.2, consists of a loop which generates the computational grid and evaluates it in the absence of any blade rows to determine grid validity. Should the grid be invalid, the repaired grid parameters, defined in table 5.1, are used, and the grid is regenerated and re-evaluated. In the optimisation studies performed in this thesis, the design is considered infeasible/unphysical if the repaired grid is also invalid, to enable automated analyses. However, for manual investigations of different ducted fan designs, the user may manually adjust the grid parameters further to obtain a valid computational grid.

Following the input of the blade row(s), the solving approach developed in section 5.4 is executed. This solving approach was also illustrated in figure 5.2, which includes the developed non-convergence and crash handling approaches, which are not repeated in figure C.1 for the sake of brevity. Once a solution is obtained from the UDC, output processing is used to obtain the desired output data. This output can be any combination of the integrated force and power coefficients, the boundary layer data, and the flowfield data. A visualisation module is included in the UDC, which handles all flowfield visualisations and improves upon the visualisation capabilities of MTFLOW to enable publication-ready visualisations of all flowfield and boundary layer variables. This module was developed as the built-in visualisation routines of MTFLOW were found to be unsuitable for detailed analyses.

An overview of the different modules in the developed UDC is shown in figure C.2. All developed modules include basic geometry inputs, allowing for straightforward testing. The file handling module is responsible for the input processing step in figure C.1. The UDC module, in effect, serves as a coupling module that integrates the developed submodules to form the complete UDC tool developed in this thesis. The output handling module consists of both processing and visualisation routines. The output processing subroutines can parse all possible output data from the UDC. In this thesis, only the integrated force coefficient output files are used. Future work can use this routine to interrogate the boundary layer or flowfield data. The visualisation subroutines can create contour plots of all primary and secondary flowfield parameters, such as the local Mach number M, normalised density ρ/ρ_∞, normalised

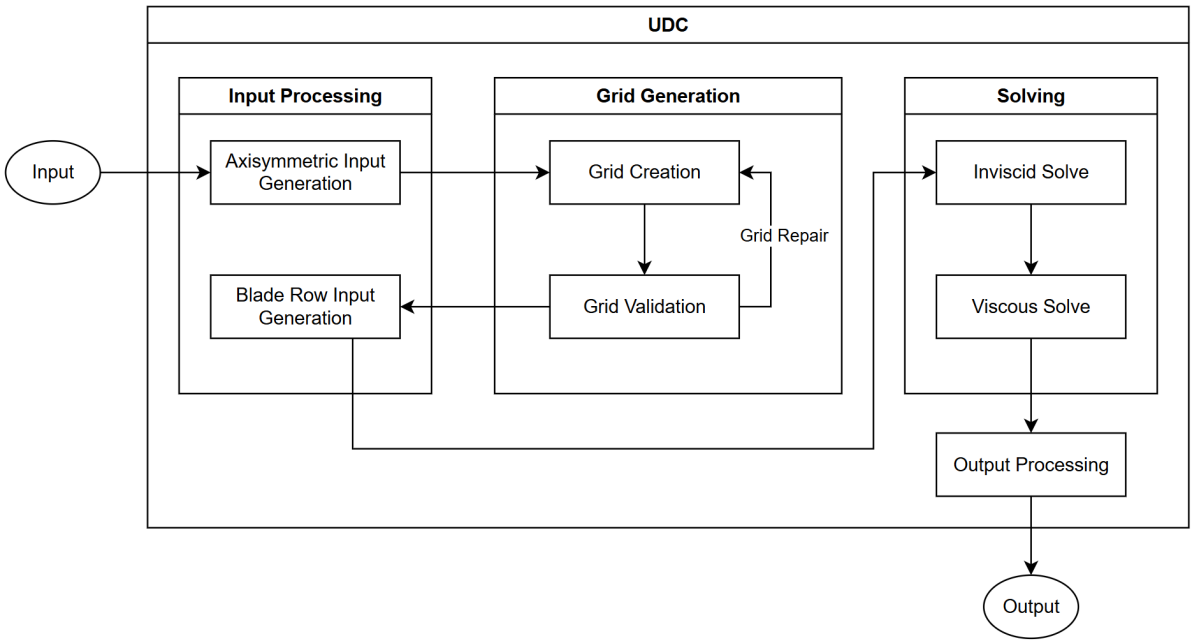


Figure C.1: Flow diagram of the developed Unified Ducted fan Code, showing the progression from an initial parameterised input geometry to the output.

swirl velocity v_θ/v_∞, or pressure coefficient C_p. Additionally, streamline plots can be constructed, as well as the evolutions of all boundary layer parameters along x. To generate these visualisations, the correct output type must be defined when calling the UDC to ensure the required output files exist for the visualisation subroutine to use.

C.2. Implementation of the Ducted Fan Optimisation Framework

This section discusses the code implementation of the ducted fan optimisation framework developed in this thesis. A pipeline diagram of the different Python modules developed as part of the optimisation framework is shown in figure C.3. Four submodule categories were identified in the development of the optimisation framework:

1. Core utilities and configuration.
2. Population/individual interface.
3. Genetic algorithm implementation.
4. Outputs & post-processing.

The focus of the development of both the UDC and the optimisation framework in this thesis has been the creation of a generalised toolset which can be used in as wide a range of applications as possible. As such, the optimisation framework is designed to require minimal user inputs to define objectives, constraints, operating conditions, and the reference design. This is handled by the core utilities and is further described in appendix C.2.2. The population and individual interface submodule ensures that a single source of truth exists when (de-)constructing design vectors. This is particularly important for the repair operator, which adjusts the design vectors of the population. This module further handles the creation of the initial population as described in section 6.3. The implementation of this submodule is described in appendix C.2.3. The implementation of the genetic algorithm, including a default set of objectives and constraints which can be selected in the configuration, is described in appendix C.2.4.

The results from a genetic optimisation, performed using the developed framework, are stored in a serialised .dill file. The post-processing submodule can read this output file to generate convergence overviews or visualise the design of the optimised design(s). This module was used to create all comparative graphs in chapters 9 and 10. Executing the optimisation framework is done through any one of the main files, which are described in appendix C.2.1.

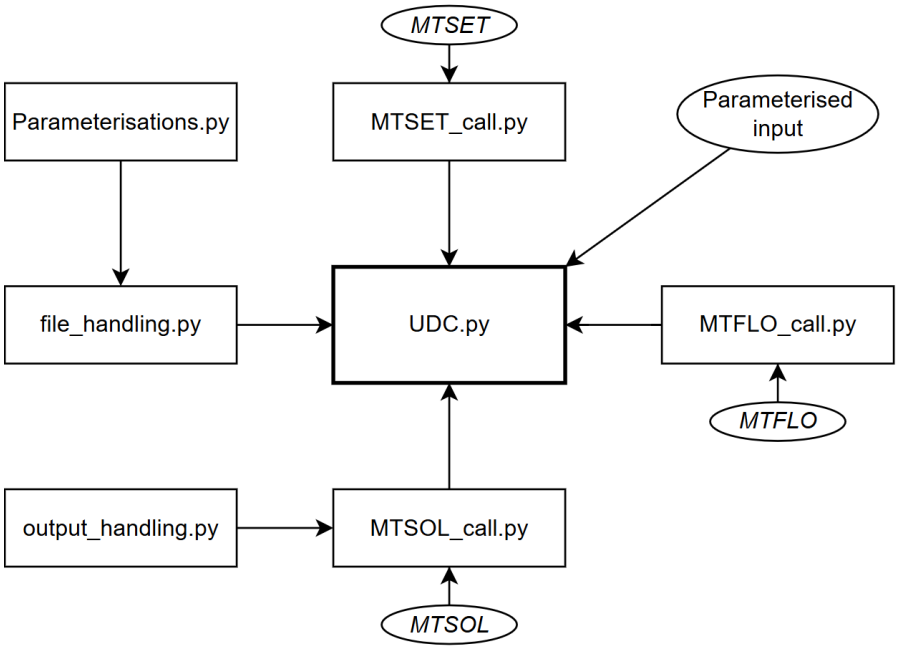


Figure C.2: File diagram of the developed Unified Ducted fan Code, showing the different modules present in the developed code.

C.2.1. Interfacing the Developed Optimisation Framework

Interfacing with the developed optimisation framework is achieved through any one of the main, main-parallelised, or main-parallelised-NSGAIII modules. These three modules implement U-NSGA-III in a serial evaluation scheme, a basic $\mu + \lambda$ genetic algorithm in a parallelised scheme, and U-NSGA-III in a parallelised scheme. For practical applications, the U-NSGA-III parallelised evaluation is required to speed up the evaluation of larger populations or multi-point optimisation problems. Parallelised evaluation is achieved using a Starmap interface, which is a straightforward supported method for parallelisation in Pymoo [52]. This Starmap limits the number of parallel evaluations to 61 due to implementation limitations. Future work can extend the parallelisation implementation to utilise direct subprocesses or threads to remove this limitation and further improve computational efficiency.

By default, the developed optimisation framework spawns as many parallel workers as possible using equation (C.1), to account for the CPU thread usage of each evaluation and any reserved computational power which must remain available. The parallel worker count is bounded by the population size, such that $n_{parallel} \leq K$. To maximise usage of available CPU threads, it is recommended to set $n_{CPU\ threads\ per\ analysis} = 1$.

$$n_{parallel} = \frac{n_{total\ CPU\ threads} - n_{reserved\ threads}}{n_{CPU\ threads\ per\ analysis}} \tag{C.1}$$

Executing the main interface is the start of the optimisation process, which is controlled by the configuration in appendix C.2.2. During an optimisation, the main interface prints the following parameters for each generation to the console for user inspection: $n_{gen}, n_{evals}, CV_{min}, CV_{avg}, \varepsilon$, and an indicator, where ε and the indicator are defined by Blank and Deb [68]. For a single-objective optimisation, ε corresponds directly to the improvements in the objective function. Termination then occurs when either the number of function evaluations exceeds the defined limit or when ε is equal to a constant value for the complete length of the sliding convergence window.

By default, the optimisation framework stores a copy of the algorithm for each generation, together with the constraint data and objective functions, for the final found optimum solution(s) and the final population. These results are stored in a serialised .dill file, which can be reloaded into a Python object for post-processing. The storage of this convergence history can be memory-intensive and is therefore

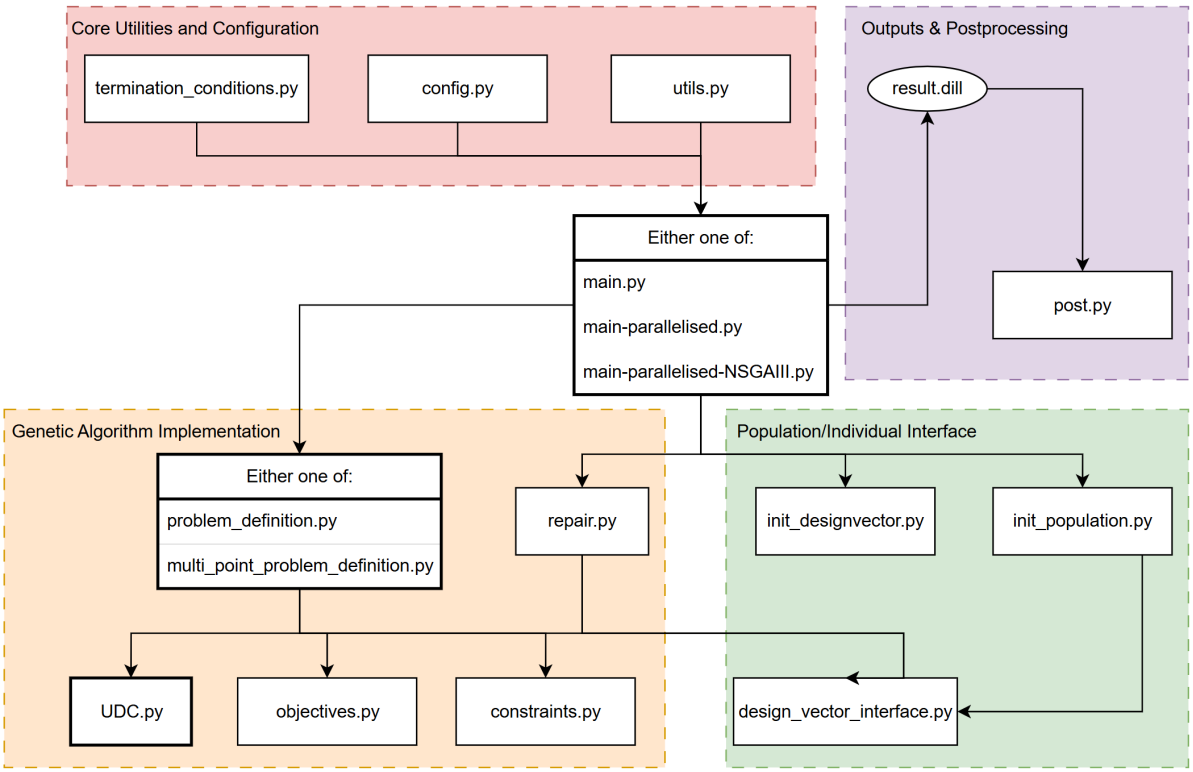


Figure C.3: The modular ducted fan optimisation framework pipeline diagram.

recommended to be turned off for larger optimisation studies using the *save_history* flag in the main interface, such as multi-point optimisations, unless required to inspect the convergence behaviour in detail. Alternatively, a custom callback method can be implemented to save the convergence metrics of interest. This was not done in this thesis. When the *save_history* flag is disabled, a basic convergence analysis can still be performed based on the ε indicator and the convergence of the minimum and average constraint violations.

C.2.2. Core Utilities and Configuration

The core utilities and configuration submodule handles the setup and configuration of the optimisation framework. This includes defining the termination conditions and setting up the reference directions, as well as selecting appropriate objective functions, constraints, and operating conditions in the configuration. The configuration file also contains the parameterised geometry of the reference design, which can readily be adjusted to optimise/investigate different designs. If any of the existing objective functions or constraints are desired to be used, they can be selected using identifiers in the configuration file. New constraints and objectives can be added to their respective files in figure C.3. The configuration further contains the initialisation parameters τ, ν δ, the global seed, the population size K, the initial population size, taken in this thesis as $2K$, and the maximum number of generations to complete the definition of the optimisation problem.

C.2.3. Population and Individual Interface

The population and individual interface submodule generates the initial population using the methods developed in section 6.3, based on the defined design vector in *init_designvector*. The design vector is constructed based on the defined parameterisations in chapter 4, and the setup of the optimisation problem in the configuration. This enables optimisations of the isolated duct, fan, or specific blade row(s) to be easily conducted as well. The design vector interface breaks down the complete design vector into Python dictionaries for use by other (sub-)modules, and reconstructs the design vector from dictionaries output by the repair operator.

C.2.4. Genetic Algorithm Implementation

Implementation of the generalised optimisation problems is done within the genetic algorithm module. This module defines two main types of optimisation problems, using two problem definition submodules: single-point and multi-point. When only defining a single operating condition in the configuration of appendix C.2.2, these two submodules are functionally equivalent. However, the single-point problem submodule is more numerically efficient, as it does not require the file editing procedures performed in the multi-point problem definition to account for different operating conditions.

Both the single-point and multi-point problem definition submodules have the option to save all MT-FLOW state files generated by the UDC during the optimisation in a dedicated archiving folder. This is useful during debugging and testing, but it is recommended to be disabled during operational use, as the total size of this archiving folder can surpass $10\ \mathrm{GB}$ for a typical optimisation, as each state file has a size of $\approx 1.1\ \mathrm{MB}$. If the *save_history* flag is enabled, the state files can be reconstructed using the UDC and the corresponding design vectors by re-evaluating the designs, which is a more memory-efficient method.

The genetic algorithm module is effectively a coupling module, much like the UDC module described in the previous section, which links the objectives, constraints, and UDC modules together. This ensures that the correct data is passed between them, allowing the Pymoo optimisation algorithm evaluator to return the correct objectives and constraints. Additionally, the repair operator defined in section 6.4 is implemented in this module.

D

Bell X-22A Validation Input Dataset

This appendix presents the input data used during validation of the developed Unified Ducted fan Code (UDC) in chapter 7. Appendix D.1 first presents the inputs for the UDC used during validation. This is followed by the Ducted Fan Design Code (DFDC) inputs in appendix D.2. The input files presented in this chapter are also contained in reference [58], with the UDC validation files automatically being generated by the X22A_validator.py module. The DFDC input files are included within the Validation subdirectory. The radial distributions of the propeller chord length, blade angle, and thickness-to-chord ratio used in validating the UDC are shown in figure D.1.

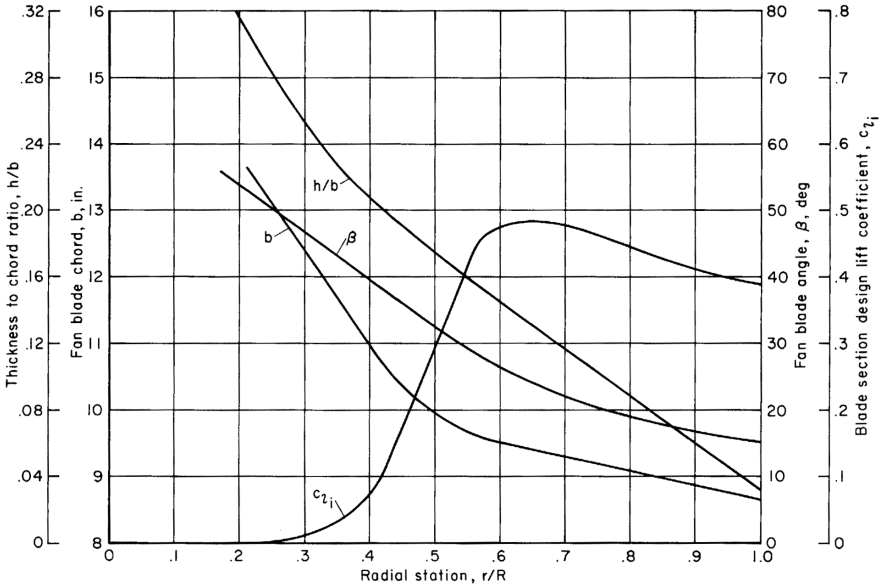


Figure D.1: Propeller blade characteristics along the blade span for the X-22A. Taken from Mort and Gamse [71].

Table D.1: Rotational rates for the X-22A propeller used in validating the developed UDC. The non-dimensionalised rotational rate is shown for $v_\infty = 26\ \mathrm{m\,s^{-1}}$ and $L_{ref} = 2.1336\ \mathrm{m}$.

Advance Ratio J [-]	Rotational Rate Ω [$\mathrm{s^{-1}}$]	Ω_{MTFLOW} [-]
0.3	40.62	20.94
0.35	34.82	17.95
0.4	30.46	15.71
0.45	27.08	13.96
0.5	24.37	12.57
0.55	22.16	11.42
0.6	20.31	10.47

D.1. X-22A Inputs for the Unified Ducted Fan Code

This section contains the walls.xxx and tflow.xxx input geometry for the X-22A as used during validation of the UDC. Listing D.1 first shows the axisymmetric walls.xxx input file, containing the X-22A centrebody and duct geometry. Listing D.2 then shows the forcing field input file tflow.xxx for a reference blade angle of $\beta_{tip} = 14.5°$, while listing D.3 shows the forcing field input file for $\beta_{tip} = 24.5°$. It should be noted that these input files are generated automatically by the X22A_validator based on the input data as part of the automated interface of the UDC. An overview of the rotational rates used for validation is given in table D.1.

Listing D.1: Axisymmetric walls.xxx input file

```
X22A_validation
-1.044    1.583    0.0    1.417
0.74202    0.01792
0.73972    0.01813
0.73284    0.01878
0.72146    0.01998
0.70571    0.02187
0.68579    0.02461
0.66192    0.02836
0.63438    0.03320
0.60350    0.03917
0.56963    0.04619
0.53318    0.05404
0.49458    0.06242
0.45427    0.07088
0.41272    0.07893
0.37044    0.08601
0.32790    0.09158
0.28561    0.09516
0.24407    0.09637
0.20376    0.09497
0.16515    0.09093
0.12870    0.08439
0.09484    0.07571
0.06395    0.06541
0.03642    0.05418
0.01254    0.04279
-0.00738    0.03205
-0.02313    0.02273
-0.03451    0.01554
-0.04139    0.01099
-0.04369    0.00943
-0.04139    -0.01099
-0.03451    -0.01554
-0.02313    -0.02273
-0.00738    -0.03205
0.01254    -0.04279
0.03642    -0.05418
0.06395    -0.06541
0.09484    -0.07571
0.12870    -0.08439
```

```
0.16515    -0.09093
0.20376    -0.09497
0.24407    -0.09637
0.28561    -0.09516
0.32790    -0.09158
0.37044    -0.08601
0.41272    -0.07893
0.45427    -0.07088
0.49458    -0.06242
0.53318    -0.05404
0.56963    -0.04619
0.60350    -0.03917
0.63438    -0.03320
0.66192    -0.02836
0.68579    -0.02461
0.70571    -0.02187
0.72146    -0.01998
0.73284    -0.01878
0.73972    -0.01813
0.74202    -0.01792
999.0    999.0
0.58333    0.55621
0.55417    0.56143
0.52500    0.56638
0.46667    0.57552
0.40833    0.58379
0.35000    0.59106
0.29167    0.59719
0.28214    0.59821
0.23333    0.60181
0.21131    0.60307
0.17500    0.60432
0.14583    0.60451
0.12202    0.60411
0.11667    0.60369
0.08750    0.60161
0.05833    0.59768
0.04375    0.59468
0.02917    0.59058
0.01458    0.58448
0.00730    0.57970
0.00000    0.56696
0.00421    0.55066
0.00842    0.54411
0.01262    0.53923
0.01683    0.53523
0.02104    0.53181
0.02525    0.52882
0.02945    0.52617
0.03366    0.52378
0.03787    0.52162
0.04208    0.51966
0.04628    0.51787
0.05049    0.51623
0.05470    0.51474
0.05891    0.51337
0.06312    0.51213
0.06732    0.51099
0.07153    0.50996
0.07574    0.50902
0.07995    0.50819
0.08415    0.50744
0.08836    0.50678
0.09257    0.50620
0.09678    0.50570
0.10099    0.50528
0.10519    0.50494
0.10940    0.50468
0.11361    0.50449
0.11782    0.50438
0.12202    0.50435
```

```
0.21131    0.50457
0.22414    0.50487
0.23697    0.50535
0.24979    0.50599
0.26262    0.50679
0.27545    0.50774
0.28828    0.50883
0.30111    0.51005
0.31394    0.51140
0.32677    0.51286
0.33959    0.51444
0.35242    0.51612
0.36525    0.51790
0.37808    0.51977
0.39091    0.52172
0.40374    0.52374
0.41656    0.52583
0.42939    0.52798
0.44222    0.53017
0.45505    0.53241
0.46788    0.53469
0.48071    0.53700
0.49353    0.53932
0.50636    0.54166
0.51919    0.54401
0.53202    0.54635
0.54485    0.54868
0.55768    0.55100
0.57050    0.55329
0.58333    0.55555
```

Listing D.2: Forcing field tflow.xxx input file for $\beta_{tip} = 14.5°$. Note that Ω_{MTFLOW} is shown for $J = 0.6$.

```
NAME
X22A_validation
END
STAGE
NBLADE
3
END
OMEGA
-10.47198
END
DATYPE
x    r    T    Sr
*0.46869    0.46869    0.46869    1
END
SECTION
0.14957    0.0    0.0    0.0
0.18106    0.0    0.0    0.0
0.21231    0.0    0.0    0.0
0.24243    0.0    0.0    0.0
0.27334    0.0    0.0    0.0
0.30519    0.0    0.0    0.0
0.33508    0.0    0.0    0.0
0.36574    0.0    0.0    0.0
0.39670    0.0    0.0    0.0
0.42841    0.0    0.0    0.0
END
SECTION
0.14957    0.21336    0.00000    0.93538
0.18106    0.21336    0.07161    0.86353
0.21231    0.21336    0.08503    0.80653
0.24243    0.21336    0.08550    0.77801
0.27334    0.21336    0.08043    0.75619
0.30519    0.21336    0.07166    0.74204
0.33508    0.21336    0.06085    0.72337
0.36574    0.21336    0.04595    0.69810
0.39670    0.21336    0.02478    0.67960
0.42841    0.21336    0.00187    0.66790
END
```

```
SECTION
0.17600    0.32004    0.00000    1.14753
0.20139    0.32004    0.04705    1.09275
0.22525    0.32004    0.05674    1.02506
0.24999    0.32004    0.05854    0.98917
0.27669    0.32004    0.05642    0.95663
0.30191    0.32004    0.05216    0.94106
0.32702    0.32004    0.04635    0.92054
0.35138    0.32004    0.03588    0.89543
0.37679    0.32004    0.01747    0.87179
0.40197    0.32004    0.00132    0.85239
END
SECTION
0.20202    0.42672    0.00000    1.49547
0.22095    0.42672    0.02829    1.41957
0.24092    0.42672    0.03488    1.32157
0.26052    0.42672    0.03598    1.27471
0.27965    0.42672    0.03492    1.23542
0.29917    0.42672    0.03236    1.22014
0.31743    0.42672    0.02923    1.19277
0.33772    0.42672    0.02225    1.15300
0.35619    0.42672    0.01105    1.12057
0.37595    0.42672    0.00083    1.09197
END
SECTION
0.22332    0.53340    0.00000    2.14319
0.23773    0.53340    0.01921    1.86636
0.25181    0.53340    0.02350    1.73543
0.26653    0.53340    0.02440    1.66918
0.28193    0.53340    0.02362    1.62224
0.29596    0.53340    0.02201    1.59934
0.31111    0.53340    0.01977    1.54932
0.32535    0.53340    0.01533    1.48421
0.33997    0.53340    0.00740    1.44269
0.35465    0.53340    0.00058    1.42072
END
SECTION
0.23724    0.64008    0.00000    2.88562
0.24861    0.64008    0.01170    2.42204
0.25997    0.64008    0.01438    2.22826
0.27190    0.64008    0.01490    2.13260
0.28336    0.64008    0.01447    2.07008
0.29437    0.64008    0.01350    2.03774
0.30635    0.64008    0.01218    1.96233
0.31781    0.64008    0.00928    1.87023
0.32955    0.64008    0.00438    1.81394
0.34074    0.64008    0.00036    1.78541
END
SECTION
0.24600    0.74676    0.00000    3.36084
0.25549    0.74676    0.00827    3.00235
0.26515    0.74676    0.01021    2.72485
0.27405    0.74676    0.01058    2.60326
0.28438    0.74676    0.01040    2.48141
0.29385    0.74676    0.00971    2.43316
0.30341    0.74676    0.00879    2.35305
0.31258    0.74676    0.00676    2.27028
0.32216    0.74676    0.00331    2.19720
0.33198    0.74676    0.00027    2.13722
END
SECTION
0.25312    0.85344    0.00000    4.78109
0.26099    0.85344    0.00459    3.62294
0.26927    0.85344    0.00569    3.25099
0.27693    0.85344    0.00588    3.09285
0.28495    0.85344    0.00573    2.97868
0.29306    0.85344    0.00536    2.90553
0.30099    0.85344    0.00491    2.76285
0.30875    0.85344    0.00372    2.62119
0.31671    0.85344    0.00183    2.53875
0.32485    0.85344    0.00015    2.49436
```

```
END
SECTION
0.25834    0.96012    0.00000    5.96711
0.26506    0.96012    0.00294    4.23870
0.27162    0.96012    0.00361    3.78543
0.27909    0.96012    0.00376    3.55564
0.28549    0.96012    0.00368    3.42252
0.29240    0.96012    0.00344    3.32649
0.29923    0.96012    0.00316    3.13945
0.30602    0.96012    0.00234    2.96524
0.31294    0.96012    0.00112    2.86739
0.31963    0.96012    0.00010    2.81685
END
SECTION
0.26138    1.06680    0.00000    6.92793
0.26754    1.06680    0.00133    4.65826
0.27366    1.06680    0.00163    4.13088
0.27972    1.06680    0.00169    3.89241
0.28593    1.06680    0.00166    3.72786
0.29214    1.06680    0.00156    3.61193
0.29834    1.06680    0.00146    3.38896
0.30413    1.06680    0.00113    3.20360
0.31046    1.06680    0.00055    3.09167
0.31659    1.06680    0.00004    3.03409
END
END
STAGE
NBLADE
4
END
OMEGA
0
END
DATYPE
x    r    T    Sr
*0.468691    0.468691    0.468691    1
END
SECTION
0.57785    0.05750    0.00000    0.0
0.64142    0.05750    0.11843    0.0
0.70470    0.05750    0.14182    0.0
0.77423    0.05750    0.14445    0.0
0.83725    0.05750    0.13702    0.0
0.89814    0.05750    0.12508    0.0
0.96386    0.05750    0.10746    0.0
1.02811    0.05750    0.08283    0.0
1.09155    0.05750    0.04322    0.0
1.15443    0.05750    0.00305    0.0
END
SECTION
0.57785    1.15000    0.00000    0.0
0.59353    1.15000    0.02922    0.0
0.60914    1.15000    0.03499    0.0
0.62630    1.15000    0.03564    0.0
0.64184    1.15000    0.03380    0.0
0.65686    1.15000    0.03086    0.0
0.67308    1.15000    0.02651    0.0
0.68893    1.15000    0.02043    0.0
0.70458    1.15000    0.01066    0.0
0.72009    1.15000    0.00075    0.0
END
END
STAGE
NBLADE
2
END
OMEGA
0
END
DATYPE
x    r    T    Sr
```

```
*0.468691    0.468691    0.468691    1
+0    0    0    0
END
SECTION
0.57772    0.05750    0.00000    0.0
0.58922    0.05750    0.03252    0.0
0.60042    0.05750    0.03843    0.0
0.61259    0.05750    0.03869    0.0
0.62341    0.05750    0.03642    0.0
0.63523    0.05750    0.03250    0.0
0.64633    0.05750    0.02757    0.0
0.65760    0.05750    0.02079    0.0
0.66901    0.05750    0.01107    0.0
0.68059    0.05750    0.00082    0.0
END
SECTION
0.57772    1.15000    0.00000    0.0
0.58922    1.15000    0.03252    0.0
0.60042    1.15000    0.03843    0.0
0.61259    1.15000    0.03869    0.0
0.62341    1.15000   0.03642    0.0
0.63523    1.15000    0.03250    0.0
0.64633    1.15000    0.02757    0.0
0.65760    1.15000    0.02079    0.0
0.66901    1.15000    0.01107    0.0
0.68059    1.15000    0.00082    0.0
END
END
END
```

Listing D.3: Forcing field tflow.xxx input file for $\beta_{tip} = 24.5°$. Note that Ω_{MTFLOW} is shown for $J = 0.6$.

```
NAME
X22A_validation
END
STAGE
NBLADE
3
END
OMEGA
-10.47198
END
DATYPE
x    r    T    Sr
*0.46869    0.46869    0.46869    1
END
SECTION
0.14957    0.0    0.0    0.0
0.18386    0.0    0.0    0.0
0.21780    0.0    0.0    0.0
0.25478    0.0    0.0    0.0
0.28704    0.0    0.0    0.0
0.32272    0.0    0.0    0.0
0.35809    0.0    0.0    0.0
0.39207    0.0    0.0    0.0
0.42626    0.0    0.0    0.0
0.46119    0.0    0.0    0.0
END
SECTION
0.14957    0.21336    0.00000    0.65159
0.18386    0.21336    0.07982    0.60003
0.21780    0.21336    0.09511    0.55454
0.25478    0.21336    0.09575    0.52910
0.28704    0.21336    0.09028    0.51276
0.32272    0.21336    0.08049    0.50089
0.35809    0.21336    0.06757    0.48516
0.39207    0.21336    0.05082    0.46459
0.42626    0.21336    0.02733    0.44942
0.46119    0.21336    0.00206    0.43978
END
SECTION
```

```
0.17503    0.32004    0.00000    0.80776
0.20368    0.32004    0.05420    0.77276
0.23313    0.32004    0.06598    0.71881
0.26181    0.32004    0.06771    0.69212
0.29106    0.32004    0.06514    0.66893
0.31994    0.32004    0.06023    0.65651
0.34850    0.32004    0.05359    0.64156
0.37824    0.32004    0.04091    0.62105
0.40716    0.32004    0.01985    0.60280
0.43573    0.32004    0.00150    0.58773
END
SECTION
0.20075    0.42672    0.00000    1.04388
0.22449    0.42672    0.03483    0.99435
0.24666    0.42672    0.04202    0.93281
0.26961    0.42672    0.04347    0.90002
0.29434    0.42672    0.04200    0.87085
0.31764    0.42672    0.03893    0.85971
0.34081    0.42672    0.03483    0.83982
0.36328    0.42672    0.02733    0.81353
0.38675    0.42672    0.01303    0.78845
0.41001    0.42672    0.00098    0.76780
END
SECTION
0.22195    0.53340    0.00000    1.42743
0.24033    0.53340    0.02464    1.27865
0.25959    0.53340    0.03023    1.19379
0.27844    0.53340    0.03110    1.15338
0.29668    0.53340    0.03003    1.12577
0.31455    0.53340    0.02795    1.11104
0.33372    0.53340    0.02500    1.08097
0.35145    0.53340    0.01966    1.03891
0.37070    0.53340    0.00915    1.00955
0.38880    0.53340    0.00071    0.99511
END
SECTION
0.23549    0.64008    0.00000    1.79565
0.25126    0.64008    0.01611    1.58291
0.26638    0.64008    0.01956    1.47968
0.28210    0.64008    0.02023    1.42678
0.29738    0.64008    0.01960    1.39096
0.31339    0.64008    0.01818    1.37080
0.32852    0.64008    0.01641    1.33358
0.34448    0.64008    0.01258    1.27669
0.35985    0.64008    0.00604    1.24218
0.37527    0.64008    0.00047    1.22381
END
SECTION
0.24389    0.74676    0.00000    1.99956
0.25742    0.74676    0.01195    1.87144
0.27058    0.74676    0.01465    1.73807
0.28436    0.74676    0.01523    1.66995
0.29863    0.74676    0.01485    1.60956
0.31260    0.74676    0.01381    1.58295
0.32576    0.74676    0.01249    1.54489
0.33936    0.74676    0.00969    1.49855
0.35350    0.74676    0.00452    1.45656
0.36686    0.74676    0.00036    1.42419
END
SECTION
0.25100    0.85344    0.00000    2.49847
0.26329    0.85344    0.00712    2.13143
0.27533    0.85344    0.00868    1.97370
0.28791    0.85344    0.00896    1.89488
0.29907    0.85344    0.00872    1.84502
0.31150    0.85344    0.00811    1.81242
0.32333    0.85344    0.00736    1.75383
0.33523    0.85344    0.00575    1.67963
0.34747    0.85344    0.00276    1.63376
0.35976    0.85344    0.00022    1.60994
END
```

```
SECTION
0.25639    0.96012    0.00000    2.82180
0.26721    0.96012    0.00475    2.37490
0.27798    0.96012    0.00584    2.18726
0.28928    0.96012    0.00605    2.09444
0.30012    0.96012    0.00588    2.03285
0.31055    0.96012    0.00549    1.99800
0.32191    0.96012    0.00496    1.92432
0.33275    0.96012    0.00378    1.83978
0.34317    0.96012    0.00187    1.79152
0.35436    0.96012    0.00015    1.76424
END
SECTION
0.25966    1.06680    0.00000    3.03910
0.26960    1.06680    0.00220    2.53417
0.27960    1.06680    0.00272    2.32537
0.29013    1.06680    0.00282    2.22278
0.30025    1.06680    0.00275    2.15492
0.31066    1.06680    0.00257    2.11337
0.32063    1.06680    0.00237    2.03575
0.33079    1.06680    0.00186    1.94363
0.34120    1.06680    0.00089    1.88896
0.35110    1.06680    0.00007    1.86174
END
END
STAGE
NBLADE
4
END
OMEGA
0
END
DATYPE
x    r    T    Sr
*0.468691    0.468691    0.468691    1
+0    0    0    0
END
SECTION
0.57785    0.05750    0.00000    0.0
0.64142    0.05750    0.11843    0.0
0.70470    0.05750    0.14182    0.0
0.77423    0.05750    0.14445    0.0
0.83725    0.05750    0.13702    0.0
0.89814    0.05750    0.12508    0.0
0.96386    0.05750    0.10746    0.0
1.02811    0.05750    0.08283    0.0
1.09155    0.05750    0.04322    0.0
1.15443    0.05750    0.00305    0.0
END
SECTION
0.57785    1.15000    0.00000    0.0
0.59353    1.15000    0.02922    0.0
0.60914    1.15000    0.03499    0.0
0.62630    1.15000    0.03564    0.0
0.64184    1.15000    0.03380    0.0
0.65686    1.15000    0.03086    0.0
0.67308    1.15000    0.02651    0.0
0.68893    1.15000    0.02043    0.0
0.70458    1.15000    0.01066    0.0
0.72009    1.15000    0.00075    0.0
END
END
STAGE
NBLADE
2
END
OMEGA
0
END
DATYPE
x    r    T    Sr
```

```
*0.468691    0.468691    0.468691    1
+0    0    0    0
END
SECTION
0.57772    0.05750    0.00000    0.0
0.58922    0.05750    0.03252    0.0
0.60042    0.05750    0.03843    0.0
0.61259    0.05750    0.03869    0.0
0.62341    0.05750    0.03642    0.0
0.63523    0.05750    0.03250    0.0
0.64633    0.05750    0.02757    0.0
0.65760    0.05750    0.02079    0.0
0.66901    0.05750    0.01107    0.0
0.68059    0.05750    0.00082    0.0
END
SECTION
0.57772    1.15000    0.00000    0.0
0.58922    1.15000    0.03252    0.0
0.60042    1.15000    0.03843    0.0
0.61259    1.15000    0.03869    0.0
0.62341    1.15000    0.03642    0.0
0.63523    1.15000    0.03250    0.0
0.64633    1.15000    0.02757    0.0
0.65760    1.15000    0.02079    0.0
0.66901    1.15000    0.01107    0.0
0.68059    1.15000    0.00082    0.0
END
END
END
```

D.2. Ducted Fan Design Code X-22A Case Definitions

This section presents the case files for the Ducted Fan Design Code analyses at the two reference blade angles considered for the X-22A. Listing D.4 first presents the input file for a reference blade angle of $\beta_{tip} = 14.5°$, while listing D.5 presents the input case file for $\beta_{tip} = 24.5°$.

Listing D.4: Input case file for $\beta_{tip} = 14.5°$ used to obtain the Ducted Fan Design Code validation data

```
DFDC Version 0.70
X22A_DFDC_validation

OPER
! Vinf     Vref      RPM
  26      26      3000
! Rho      Mu        Vso      altkm
  0.0      0.0       0.0      0.0
! XDwake   Nwake
  0.8      20
! Lwkrlx
  F
ENDOPER


AERO
!  #sections
     3
!  Xisection
   0.0000
!  A0deg         dCLdA         CLmax          CLmin
   0        6.2800        2       -2
!  dCLdAstall    dCLstall       Cmconst          Mcrit
  0.50000       0.20000        0.0000       0.70000
!  CDmin         CLCDmin         dCDdCL^2
  0.10000E-01  0.0       0.50000E-02
!  REref         REexp
  2.000E+06  0.35000
!  Xisection
   0.4
!  A0deg         dCLdA         CLmax          CLmin
   -2.1       6.2800        1.7       -1.1
```

```
!  dCLdAstall   dCLstall      Cmconst          Mcrit
  0.50000      0.20000       0.0000       0.70000
!  CDmin        CLCDmin         dCDdCL^2
  0.05800E-01  0.38      0.50000E-02
!  REref        REexp
  2.000E+06  0.35000
!  Xisection
   1
!  A0deg        dCLdA         CLmax          CLmin
   -2.0       6.2800        1.7       -1.1
!  dCLdAstall   dCLstall      Cmconst          Mcrit
  0.50000      0.20000       0.0000       0.70000
!  CDmin        CLCDmin         dCDdCL^2
  0.0550E-01  0.25      0.50000E-02
!  REref        REexp
  2.000E+06  0.35000
ENDAERO


ROTOR
!       Xdisk         Nblds       NRsta
  0.3556                3           18
!  #stations
    17
!           r         Chord         Beta
0.21336 0.3510  52.6
0.26670 0.3302  49.0
0.32004 0.3152  45.8
0.37338 0.2956  42.2
0.42672 0.2794  38.5
0.48006 0.2644  34.9
0.53340 0.2528  31.3
0.58674 0.2448  28.1
0.64008 0.2413  25.4
0.69342 0.2390  23.1
0.74676 0.2367  20.8
0.80010 0.2332  19.0
0.85344 0.2309  18.1
0.90678 0.2286  16.7
0.96012 0.2251  15.8
1.01346 0.2217  14.9
1.06680 0.2205  14.5
ENDROTOR


GEOM
DuctCenterbody
1.58318    0.03823
1.57826    0.03867
1.56358    0.04007
1.53930    0.04263
1.50571    0.04666
1.46319    0.05251
1.41226    0.06049
1.35351    0.07084
1.28762    0.08358
1.21536    0.09854
1.13760    0.11530
1.05523    0.13316
0.96922    0.15122
0.88058    0.16839
0.79036    0.18350
0.69960    0.19539
0.60937    0.20303
0.52073    0.20560
0.43473    0.20263
0.35236    0.19400
0.27459    0.18005
0.20234    0.16152
0.13645    0.13956
0.07769    0.11560
```

```
 0.02676    0.09129
-0.01574     0.06837
-0.04934     0.04850
-0.07362     0.03314
-0.08830     0.02344
-0.09321     0.02012
-0.08830     -0.02344
-0.07362     -0.03314
-0.04934     -0.04850
-0.01574     -0.06837
 0.02676    -0.09129
 0.07769    -0.11560
 0.13645    -0.13956
 0.20234    -0.16152
 0.27459    -0.18005
 0.35236    -0.19400
 0.43473    -0.20263
 0.52073    -0.20560
 0.60937    -0.20303
 0.69960    -0.19539
 0.79036    -0.18350
 0.88058    -0.16839
 0.96922    -0.15122
 1.05523    -0.13316
 1.13760    -0.11530
 1.21536    -0.09854
 1.28762    -0.08358
 1.35351    -0.07084
 1.41226    -0.06049
 1.46319    -0.05251
 1.50571    -0.04666
 1.53930    -0.04263
 1.56358    -0.04007
 1.57826    -0.03867
 1.58318    -0.03823
 999.   999.
 1.24460    1.18673
 1.18237    1.19786
 1.12014    1.20843
 0.99568    1.22793
 0.87122    1.24556
 0.74676    1.26108
 0.62230    1.27416
 0.60198    1.27635
 0.49784    1.28402
 0.45085    1.28670
 0.37338    1.28938
 0.31115    1.28978
 0.26035    1.28892
 0.24892    1.28803
 0.18669    1.28358
 0.12446    1.27520
 0.09334    1.26880
 0.06223    1.26006
 0.03111    1.24703
 0.01557    1.23685
 0.00000    1.20968
 0.00897    1.17489
 0.01795    1.16091
 0.02693    1.15049
 0.03591    1.14196
 0.04488    1.13467
 0.05386    1.12829
 0.06284    1.12262
 0.07182    1.11753
 0.08079    1.11292
 0.08977    1.10873
 0.09875    1.10492
 0.10773    1.10143
 0.11670    1.09824
 0.12568    1.09533
```

```
0.13466    1.09267
0.14364    1.09024
0.15261    1.08804
0.16159    1.08605
0.17057    1.08426
0.17955    1.08266
0.18852    1.08125
0.19750    1.08002
0.20648    1.07896
0.21546    1.07807
0.22443    1.07734
0.23341    1.07678
0.24239    1.07638
0.25137    1.07615
0.26035    1.07607
0.45085    1.07654
0.47822    1.07719
0.50559    1.07821
0.53296    1.07958
0.56033    1.08128
0.58770    1.08330
0.61507    1.08563
0.64244    1.08823
0.66981    1.09111
0.69718    1.09424
0.72455    1.09761
0.75192    1.10120
0.77929    1.10499
0.80666    1.10898
0.83403    1.11313
0.86141    1.11745
0.88878    1.12190
0.91615    1.12648
0.94352    1.13117
0.97089    1.13595
0.99826    1.14081
1.02563    1.14573
1.05300    1.15069
1.08037    1.15568
1.10774    1.16069
1.13511    1.16568
1.16248    1.17066
1.18985    1.17561
1.21722    1.18050
1.24460    1.18532
ENDGEOM
```

Listing D.5: Input case file for $\beta_{tip} = 24.5°$ used to obtain the Ducted Fan Design Code validation data

```
DFDC Version 0.70
X22A_DFDC_validation

OPER
! Vinf    Vref     RPM
  26     26     3000
! Rho     Mu       Vso     altkm
  0.0     0.0      0.0     0.0
! XDwake  Nwake
  0.8     20
! Lwkrlx
  F
ENDOPER


AERO
!  #sections
     3
!  Xisection
   0.0000
!  A0deg        dCLdA         CLmax         CLmin
   0       6.2800       2      -2
```

```
!  dCLdAstall   dCLstall      Cmconst          Mcrit
  0.50000      0.20000       0.0000       0.70000
!  CDmin        CLCDmin         dCDdCL^2
  0.10000E-01  0.0       0.50000E-02
!  REref        REexp
  2.000E+06  0.35000
!  Xisection
   0.4
!  A0deg        dCLdA         CLmax          CLmin
   -2.1       6.2800        1.7       -1.1
!  dCLdAstall   dCLstall      Cmconst          Mcrit
  0.50000      0.20000       0.0000       0.70000
!  CDmin        CLCDmin         dCDdCL^2
  0.05800E-01  0.38       0.50000E-02
!  REref        REexp
  2.000E+06  0.35000
!  Xisection
   1
!  A0deg        dCLdA         CLmax          CLmin
   -2.0        6.2800        1.7       -1.1
!  dCLdAstall   dCLstall      Cmconst          Mcrit
  0.50000      0.20000       0.0000       0.70000
!  CDmin        CLCDmin         dCDdCL^2
  0.0550E-01  0.25       0.50000E-02
!  REref        REexp
  2.000E+06  0.35000
ENDAERO


ROTOR
!       Xdisk         Nblds        NRsta
  0.3556                3            18
!  #stations
    17
!           r         Chord          Beta
0.21336 0.3510  62.6
0.26670 0.3302  59.0
0.32004 0.3152  55.8
0.37338 0.2956  52.2
0.42672 0.2794  48.5
0.48006 0.2644  44.9
0.53340 0.2528  41.3
0.58674 0.2448  38.1
0.64008 0.2413  35.4
0.69342 0.2390  33.1
0.74676 0.2367  30.8
0.80010 0.2332  29.0
0.85344 0.2309  28.1
0.90678 0.2286  26.7
0.96012 0.2251  25.8
1.01346 0.2217  24.9
1.06680 0.2205  24.5
ENDROTOR


GEOM
DuctCenterbody
1.58318    0.03823
1.57826    0.03867
1.56358    0.04007
1.53930    0.04263
1.50571    0.04666
1.46319    0.05251
1.41226    0.06049
1.35351    0.07084
1.28762    0.08358
1.21536    0.09854
1.13760    0.11530
1.05523    0.13316
0.96922    0.15122
0.88058    0.16839
```

```
 0.79036    0.18350
 0.69960    0.19539
 0.60937    0.20303
 0.52073    0.20560
 0.43473    0.20263
 0.35236    0.19400
 0.27459    0.18005
 0.20234    0.16152
 0.13645    0.13956
 0.07769    0.11560
 0.02676    0.09129
-0.01574     0.06837
-0.04934     0.04850
-0.07362     0.03314
-0.08830     0.02344
-0.09321     0.02012
-0.08830     -0.02344
-0.07362     -0.03314
-0.04934     -0.04850
-0.01574     -0.06837
 0.02676    -0.09129
 0.07769    -0.11560
 0.13645    -0.13956
 0.20234    -0.16152
 0.27459    -0.18005
 0.35236    -0.19400
 0.43473    -0.20263
 0.52073    -0.20560
 0.60937    -0.20303
 0.69960    -0.19539
 0.79036    -0.18350
 0.88058    -0.16839
 0.96922    -0.15122
 1.05523    -0.13316
 1.13760    -0.11530
 1.21536    -0.09854
 1.28762    -0.08358
 1.35351    -0.07084
 1.41226    -0.06049
 1.46319    -0.05251
 1.50571    -0.04666
 1.53930    -0.04263
 1.56358    -0.04007
 1.57826    -0.03867
 1.58318    -0.03823
999.   999.
 1.24460    1.18673
 1.18237    1.19786
 1.12014    1.20843
 0.99568    1.22793
 0.87122    1.24556
 0.74676    1.26108
 0.62230    1.27416
 0.60198    1.27635
 0.49784    1.28402
 0.45085    1.28670
 0.37338    1.28938
 0.31115    1.28978
 0.26035    1.28892
 0.24892    1.28803
 0.18669    1.28358
 0.12446    1.27520
 0.09334    1.26880
 0.06223    1.26006
 0.03111    1.24703
 0.01557    1.23685
 0.00000    1.20968
 0.00897    1.17489
 0.01795    1.16091
 0.02693    1.15049
 0.03591    1.14196
```

```
0.04488    1.13467
0.05386    1.12829
0.06284    1.12262
0.07182    1.11753
0.08079    1.11292
0.08977    1.10873
0.09875    1.10492
0.10773    1.10143
0.11670    1.09824
0.12568    1.09533
0.13466    1.09267
0.14364    1.09024
0.15261    1.08804
0.16159    1.08605
0.17057    1.08426
0.17955    1.08266
0.18852    1.08125
0.19750    1.08002
0.20648    1.07896
0.21546    1.07807
0.22443    1.07734
0.23341    1.07678
0.24239    1.07638
0.25137    1.07615
0.26035    1.07607
0.45085    1.07654
0.47822    1.07719
0.50559    1.07821
0.53296    1.07958
0.56033    1.08128
0.58770    1.08330
0.61507    1.08563
0.64244    1.08823
0.66981    1.09111
0.69718    1.09424
0.72455    1.09761
0.75192    1.10120
0.77929    1.10499
0.80666    1.10898
0.83403    1.11313
0.86141    1.11745
0.88878    1.12190
0.91615    1.12648
0.94352    1.13117
0.97089    1.13595
0.99826    1.14081
1.02563    1.14573
1.05300    1.15069
1.08037    1.15568
1.10774    1.16069
1.13511    1.16568
1.16248    1.17066
1.18985    1.17561
1.21722    1.18050
1.24460    1.18532
ENDGEOM
```

E

Optimisation Algorithm Convergence Behaviour

This appendix contains the convergence overviews of the single-point, single-objective optimisations performed in chapter 9. Appendix E.1 first presents the convergence overview for the take-off condition optimisation, followed by the overview for the combat cruise condition optimisation in appendix E.2. Finally, appendix E.3 presents the convergence overview for the optimisation at the endurance cruise condition using a frontal area constraint.

E.1. Efficiency Optimisation at the Take-Off Condition

The convergence overview for the optimisation at the take-off condition is shown in figure E.1. This figure shows that the objective function value remained constant after generation 3, resulting in termination before the complete population became feasible. The final population had an average constraint violation of $CV_{avg} = 5.00 \times 10^{-5}$, and a maximum constraint violation of $CV_{max} = 1.40 \times 10^{-3}$.

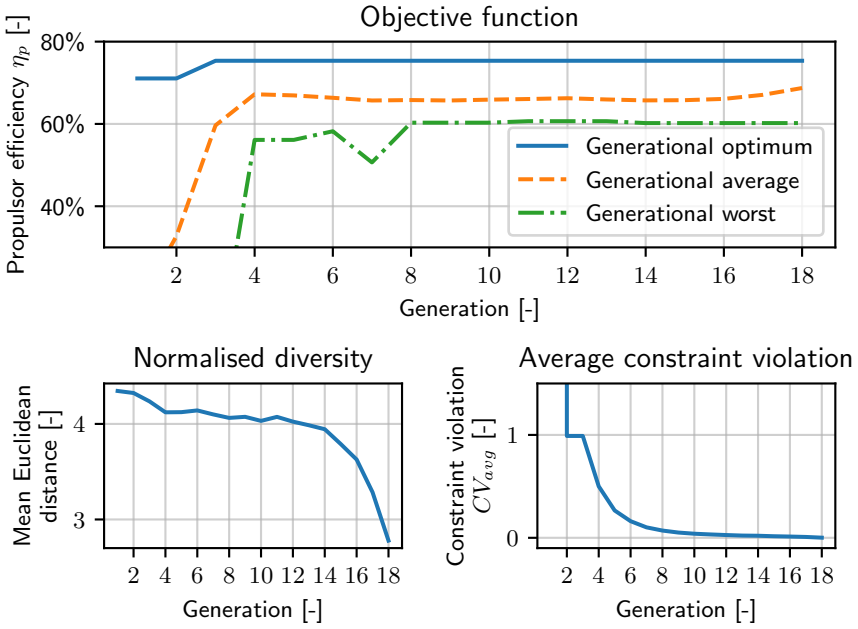


Figure E.1: Convergence of the propulsor efficiency, normalised population diversity, and average constraint violation for the single-point, single-objective optimisation at the take-off condition of the X-22A propulsor.

E.2. Efficiency Optimisation at the Combat Condition

The convergence overview for the single-point, single-objective propulsor efficiency optimisation at the combat cruise condition is shown in figure E.2. This figure shows the premature convergence discussed in section 9.1. The population diversity and objective functions both stagnate early in the optimisation process, causing the objective space termination condition to be satisfied after 30 generations when the average constraint violation is still non-zero. The final population had an average constraint violation of $CV_{avg} = 2.52 \times 10^{-3}$, and a maximum constraint violation of $CV_{max} = 1.11 \times 10^{-2}$.

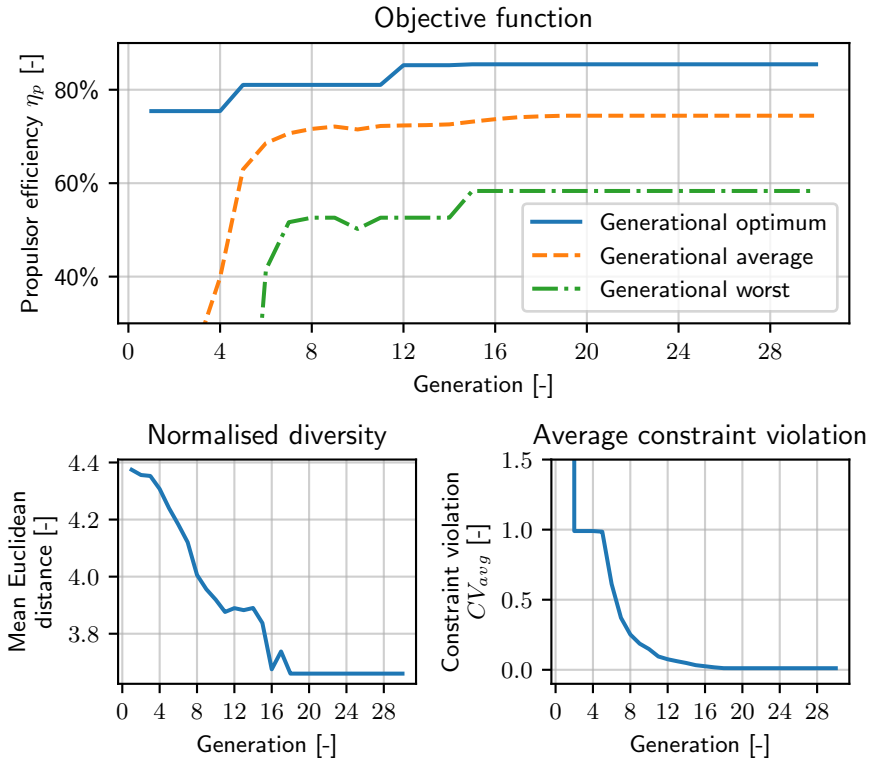


Figure E.2: Convergence of the propulsor efficiency, normalised population diversity, and average constraint violation for the single-point, single-objective optimisation at the combat cruise condition of the X-22A propulsor.

Further analyses, with different seeds for the pseudo-random number generation used within the genetic algorithm, or modified initialisation parameters τ, ν, δ, can help to avoid this and yield a better result. However, this was not investigated in this thesis due to time and resource limitations. It should be noted, however, that this premature convergence does not invalidate the obtained design.

Different analyses for the take-off optimisation in appendix E.1 were performed at the start of the execution phase of this thesis to determine a suitable set of initialisation parameters. A combat-cruise condition propulsor efficiency optimisation conducted using $\nu = 0.25$ also suffered from premature convergence, while yielding a near-identical design compared to the one presented in this thesis, which was obtained using $\nu = 0.5$. This illustrates the need to run multiple optimisations for the same operating condition using different initialisation parameters, as discussed in chapter 9.

E.3. Frontal Area-Constrained Efficiency Optimisation at the Endurance Condition

In section 9.2, the optimised designs obtained from the multi-objective optimisation are compared against the result obtained from the single-objective optimised design in section 9.1, and a frontal-area constrained single-objective optimisation, given by equation (8.2). The convergence overview of this area-constrained optimisation is given by figure E.3. Similar premature convergence as seen for the combat cruise condition optimisation is seen in this figure as well. The final population had

an average constraint violation $CV_{avg} = 3.89 \times 10^{-3}$, while the maximum constraint violation was $CV_{max} = 1.47 \times 10^{-2}$. Using a larger sliding window for the objective space termination condition is expected to resolve this.

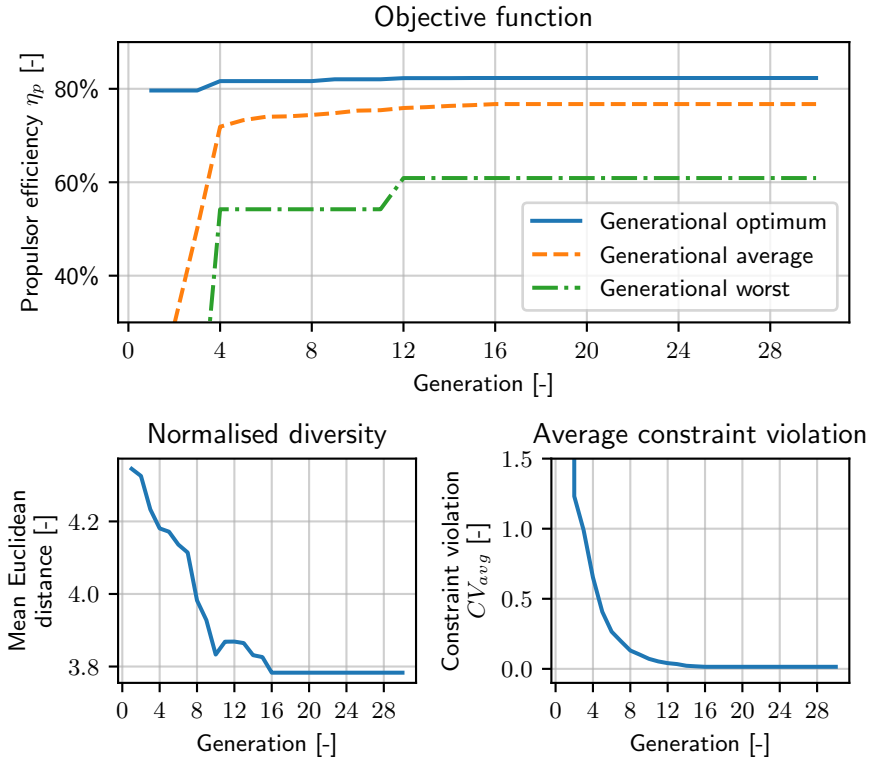


Figure E.3: Convergence of the propulsor efficiency, normalised population diversity, and average constraint violation for the single-point, single-objective efficiency optimisation, with a frontal area constraint, at the endurance cruise condition of the X-22A propulsor.