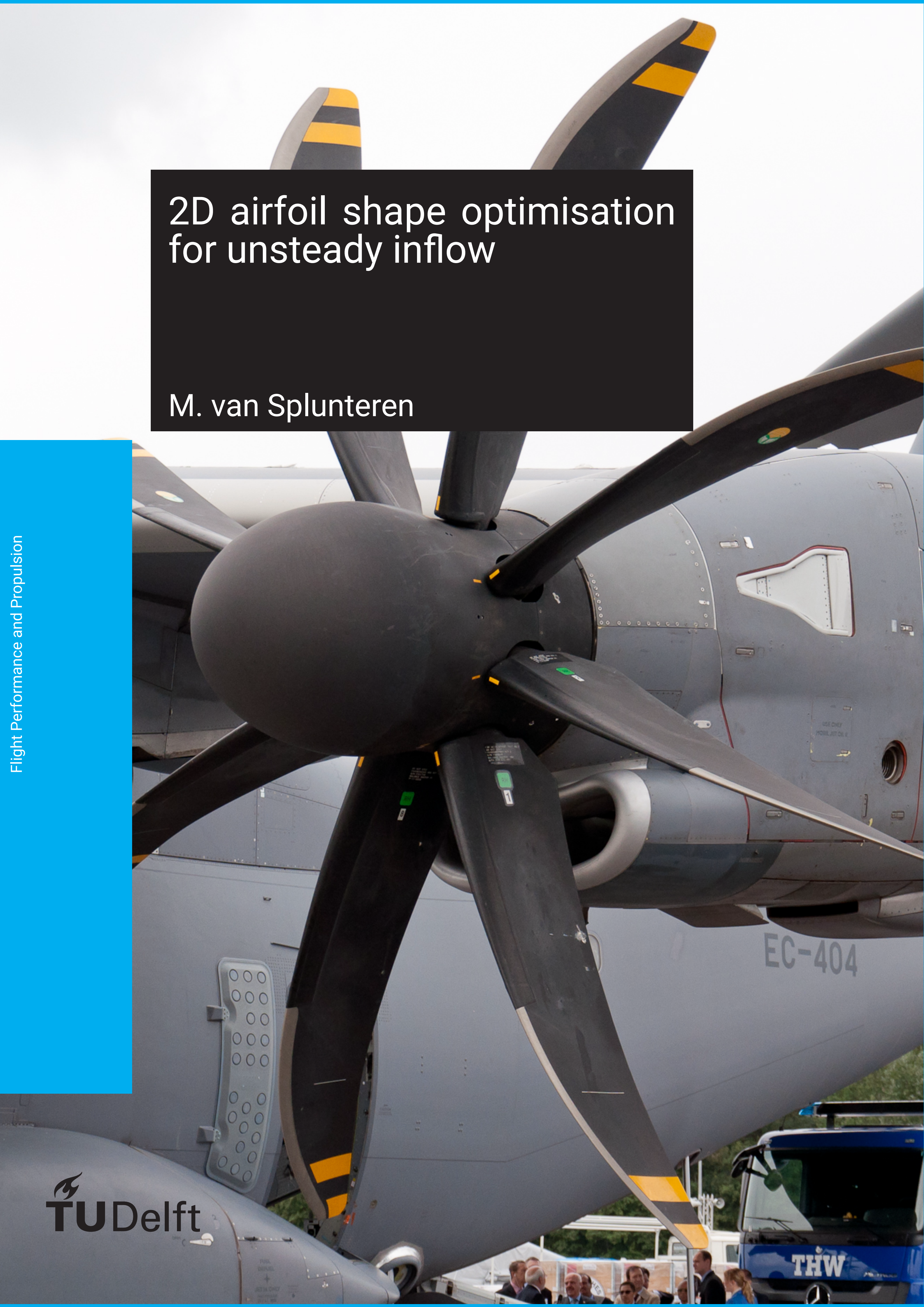
2D airfoil shape optimisation for unsteady inflow

M. van Splunteren

Flight Performance and Propulsion



TU Delft

Cover picture: Julian Herzog - `http://julianherzog.com/`

2D airfoil shape optimisation for unsteady inflow

by

M. van Splunteren

in partial fulfilment of the requirements for the degree of

Master of Science
in Aerospace Engineering

at the Delft Univesity of Technology,
to be defended publicly on Tuesday January 24, 2023 at 10:00 AM.

Student number:	4345614	
Thesis Supervisor:	Dr. ir. M. Pini	
	Dr. ir. T. Sinnige	
Thesis Committee:	Dr. ir. M. Pini	TU Delft
	Dr. ir. T. Sinnige	TU Delft
	Prof. dr. ir. G. la Rocca	TU Delft
	Dr. ir. A. H. van Zuijlen	TU Delft

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

Acknowledgements

Throughout this thesis I was in close contact with my supervisors, Matteo Pini and Tomas Sinnige. I would like to thank them for their support and their understanding of my position with regards to my schedule. Having a racing career next to studying aerospace engineering was not always easy but not less enjoyable. I would like to extend my sincere gratitude to them for their patience and their ease of reaschability. Furtmermore, I would like to thank my parents for always supporting me and showing a great interest in what I was doing. Finally, I would like to thank my girlfriend who was here for me during the entire project and always gave me the support and her continuous encouragement when it got tough.

Max van Splunteren
Delft, January 2023

Executive Summary

With emission standards being raised by The Advisory Council for Aviation Research and Innovation in Europe (ACARE) through a 75% reduction in CO_2 by the year 2050 [1], it becomes more important to make aircraft substantially more efficient. With the commercial aircraft sector emitting 70% more CO_2 than predicted in 2019 [28], it is of great importance to find alternative ways to lower emissions. This is why the Advanced Propulsion and Power Unit (APPU) project is investigating if the Auxiliary Power Unit (APU) can also be used as a propulsion system. This project aims to redesign the tail cone of the Airbus A320neo to fit a propeller.

In support of the APPU project, a propulsor must be designed to cope with Boundary Layer Ingestion (BLI). Existing research has focused on isolated BLI propellers or propeller optimisation problems for uniform inflow. Therefore the current research is pointed in the direction of an unsteady inflow propeller optimisation problem. Propellers have an inherently higher propulsive efficiency than turbofan engines; however, this comes at the cost of cruise speed and noise. BLI is a method to increase propulsive efficiency even further by ingesting the boundary layer created around the aircraft's fuselage. The engines ingest the slowed-down air and accelerate it to approximately free-stream airspeed, which uses less power to perform this acceleration. The accompanying effect is that the fuselage wake is minimised, and the aircraft's drag is reduced. This also introduces an azimuthal non-uniformity in the inflow field ahead of the propeller. The propeller is thus experiencing a time-varying inflow field which leads to a time-varying blade loading. Therefore, unsteady inflow conditions need to be considered when designing a propeller which is able to perform in these conditions. Unsteady inflow optimisation has proven to be very time-consuming, so the efficiency of the optimisation problem is of great importance. The efficiency of the proposed method for this research comes from the adjoint method being used in the optimisation process. The adjoint method allows the optimiser to compute all design variable gradients in a much shorter time frame. The ultimate goal of the optimisation is to maximise the performance of the APPU propeller while also minimising noise emission. The planform and airfoil design influence the performance and noise generation of the propeller. The fact that airfoil design plays a critical role in the performance and noise generation of the propeller. Therefore, in this research, the choice was made to go for an airfoil design-centric optimisation problem. The method for quantifying one of the possible sources of noise emission is through the lift response of the airfoil due to unsteady inflow effects. The assumption is made that reducing this lift response reduces the possibility of noise generation. This leads to the research's objective;

Development of an efficient design method tailored to a propeller airfoil while using the Harmonic Balance method combined with the Adjoint method to minimise the airfoil drag and limit the lift response to unsteady effects.

By using the adjoint method in the airfoil optimisation problem, the computational time necessary is greatly reduced compared to a finite difference methods. Determining the gradients of the design variables necessary for a gradient-based design optimisation problem with a finite differences method calculates the full flow equations for each design variable. Whereas the adjoint method computes the direct flow and adjoint equation which are comparable in computational time. This means that the computational time savings are quite substantial for a larger number of design variables. To further reduce the computational time, a Reduced Order Model (ROM) is used for the proposed design optimisation problem. With the propeller experiencing an inherently periodic inflow, the Harmonic Balance (HB) method is a suitable option. Due to the filtering that the Fourier-based HB method uses, it is able to obtain accurate solutions while only solving for a small number of harmonics. The reason for this is that a pseudo time marching derivative operator for the Euler equation is used. This operator is able to march to a steady state solution with only a small number of harmonics [22]. The HB solver transformed for the adjoint method is implemented in the open-source CFD simulation software SU2 by Rubino et al. [45], which will be used in this study.

The real inflow conditions need to be modelled to increase the fidelity of the optimisation problem. These inflow conditions of the propeller proposed for the APPU project are constructed using the data from the current research of the project. The tangential velocity experienced by the airfoil is computed using the cartesian velocity components. With the tangential velocity and the axial velocity, the change in the inflow angle of the propeller blade can be determined. In order for this to be implemented in the SU2 source code, a Fourier series expansion is computed using seven harmonics. This number is validated by a convergence study.

For the HB solver to function, a discrete number of frequencies are composed as input parameters. A time-accurate simulation of the baseline NACA 0010 airfoil is used. The Fast Fourier Transform of the real inflow condition shows that the periodic inflow is solely dependent on the base frequency of the propeller blade. The Power Spectral Density (PSD) function only shows integer multiples of the base frequency. Only five time instances or two frequencies were enough to represent the time-accurate simulation accurately. For additional fidelity in the simulation, one extra is used. In total, seven time instances will be used for the HB solver simulation.

The objective function was formulated such that the optimiser minimises the drag coefficient while keeping the lift coefficient constant at a given value. In the simulation, a secondary performance indicator has been implemented in the form of a lift coefficient Root Mean Squared Error (RMSE), or $C_{l_{RMSE}}$, value constraint. The error is measured compared to the average lift coefficient of the airfoil. The objective function and the corresponding constraints for the optimisation problem can be found in Equation 1. The parameterisation is done using the Hicks-Henne bump functions. These are the easiest and most robust in their operation. The approach to the optimisation problem is to use a Sequential Least Squares Programming (SLSQP) algorithm with derivatives. Lyu et al. [40] have shown that this is the most efficient optimisation algorithm for gradient-based aerodynamic shape optimisation.

$$
\begin{aligned}
\text{Minimise for } \overline{C_d} \quad & \overline{C_d}(\boldsymbol{U}_n, \boldsymbol{X}_n, \boldsymbol{\alpha}) \\
\text{Subject to} \quad & \overline{C_l} = \overline{C_{l_0}} \\
& \delta_{max} = \delta_{max_0} \\
& C_{l_{RMSE}} < \xi \cdot C_{l_{RMSE_0}}
\end{aligned}
\tag{1}
$$

The secondary performance indicator is expressed as the $C_{l_{RMSE}}$ value, indicating the amplitude of the lift coefficient over a full revolution of the propeller blade. As mentioned above, the assumption is made that the reduction of this metric would result in a lowered noise emission by the propeller. From literature, it is known that time-varying blade loading is a source of noise generation in the case of propellers [29, 56]. Blade loading can be related to the lift coefficient of the airfoil for a propeller. A more constant blade loading would yield lower noise emissions. Therefore, the $C_{l_{RMSE}}$ metric is used to quantify the amplitude of the lift coefficient change over a full period.

The adjoint method computes the gradients for the optimisation algorithm in this research. These gradients need to be validated by means of comparison to the gradients obtained using a finite differences model. The finite differences gradients are generally accepted as the true gradients. The computational time necessary for the finite differences method is highly dependent on the number of design variables. Therefore, the validation process uses only ten Hicks-Henne control points. Both the gradients of the drag objective and $C_{l_{RMSE}}$ value objective were obtained through both the finite differences method and the adjoint method. In both cases, the gradients computed with the different methods showed excellent agreement. The maximum deviation of the adjoint method compared to the finite differences method was 1.3%. This was only a singular control point, whereas the rest of the control points deviated between 0-0.2%.

Using the NACA 0010 airfoil, a baseline performance simulation has been performed to verify the HB solver frequencies and determine the convergence history. The time instances chosen show good agreement with the time-accurate simulation of the baseline airfoil. The direct and adjoint flow equations also show good convergence during the simulation. The contours of the Mach number show a strong normal shock at approximately $0.7\frac{x}{c}$ over the airfoil. A possible area of improvement could be for the optimiser to gain performance in the form of a reduced drag coefficient, as normal shocks over the airfoil tend to generate a substantial amount of drag.

Multiple optimisation runs have been proposed in this research to achieve an improved airfoil shape. As a verification of the optimiser, different inflow conditions have been run for the complex real inflow conditions and a more simple sinusoidal inflow condition. The real inflow conditions are subjected to a constraint limiting the lift coefficient's forced response. This constraint is set to a value lowered by a set percentage to create an incentive for the optimiser to push the $C_{l_{RMSE}}$ towards the lowered value. This means that the optimiser deals with a constraint violation from the start of the optimisation process. During the optimisation process for the simple sine inflow, constraining the forced lift response led to a lot of difficulties. Therefore, the choice was made to not use the force lift response constraint and focus on a baseline optimisation and only investigate the change in inflow conditions. The proposed optimisation runs are used to determine the effect of the inflow conditions and the $C_{l_{RMSE}}$ constraint on the airfoil design. During the post-processing of the optimisation runs, a discovery was made that the optimiser would approach the thickness constraint for all runs. Therefore, an extra baseline optimisation run was performed with an alternative thickness constraint for real inflow conditions, checking whether this thickness constraint influenced the outcome. Finally, an extra optimisation run is executed with the $C_{l_{RMSE}}$ value as an objective function. Unfortunately, this optimisation run was not able to converge to a solution. The optimiser was oscillating heavily and diverged after 4-8 evaluations, irrespective of the CFD solver settings that were changed. With the chosen solver, the optimisation problem is able to converge to a solution within 48 hours on either 48-core *AMD Opteron 6234 Processor* or ten-core *Intel Core i9-10850K CPU*.

The simulation with the sine wave inflow condition experienced more difficulties in converging with a $C_{l_{RMSE}}$ value constraint. Therefore, to verify that the optimiser still performs under different inflow conditions, an optimisation run was performed without $C_{l_{RMSE}}$ value constraint. The optimiser first had to change the airfoil shape to facilitate a similar average lift coefficient as the airfoil in real inflow conditions was producing. However, this was only a very small increase in lift coefficient. The optimiser reduces the $C_{l_{RMSE}}$ value during the optimisation by 8.5 % while not setting a constraint for this value. The drag coefficient showed a promising decrease of 75.1 %. Similar to what happened for the real inflow conditions, the normal shock disappeared, and a supersonic bubble of a lower overall Mach number appeared. The pressure distributions of the optimised airfoil showed a very gradual development over the chord length of the airfoil.

Both optimisation runs for real inflow conditions have shown a substantial decrease in drag coefficient compared to the baseline real inflow simulation in the order of 90 %. The secondary performance indicator has also proven to be successful for the real inflow optimisation runs, with a 10.5 % and 13.1 % decrease in $C_{l_{RMSE}}$ value. The influence of thickness has also been investigated after seeing the optimiser approaching the thickness constraint for each optimisation run. The thickness constraint was set to 5% of the chord. This led to an even better-performing airfoil compared to the baseline optimised airfoil, the $C_{l_{RMSE}}$ value reduced by 12.6%. The Mach number's contours also show a normal shock intensity decrease. Where the baseline airfoil had a strong normal shock over the airfoil, the optimised airfoil shows a bubble-like area of supersonic flow over the top of the airfoil. The overall Mach number of the flow over the airfoil has decreased in the optimisation process. The $C_{l_{RMSE}}$ value constrained optimisation also divided the supersonic bubble into two smaller bubbles. This further decreased the c_{l_α} gradient and made the airfoil lift coefficient less sensitive with respect to the angle of attack changes. Unfortunately, the optimisation using the $C_{l_{RMSE}}$ value as the objective function could not converge to a solution. Although the gradients showed good agreement in the gradient validation, the optimiser was not able to find a solution. The optimiser diverges between 4-8 evaluations after initialisation. Many different objective weights and constraints have been tried in order to get some sort of convergence.

So with the method presented, the research questions can be answered in the following way:

- It is possible to limit the forced lift response while also optimising for the $\frac{C_l}{C_d}$ performance of an airfoil;
- For different inflow conditions, the optimiser was able to converge to a similar airfoil shape;
- Choosing an Euler solver for the optimisation has increased the inaccuracies of the optimisation results by not including the viscous drag forces;
- The robustness of the SU2 Harmonic Balance solver can be increased. Small changes in the setup can make the simulation diverge;

- The forced response quantified as the lift coefficient RMSE value can be reduced up to 13 %, which is a decrease.

Aiming to improve the performance and noise emissions of the installed propeller proposed by the APPU project, this method will contribute to the APPU project as a step in the direction of sustainability and reduction of environmental pollution in the form of noise and CO_2 emissions.

Contents

Nomenclature

Abbreviations

2D	Two dimensional
3D	Three dimensional
ACARE	Advisory Council for Aviation Research and innovation in Europe
ALE	Arbitrairy Lagrangian-Eulerian
APPU	Advanced Power and Propulsion Unit
APU	Auxiliary Power Uni
BEMT	Blade Element Momentum Theory
BLI	Boundary Layer Ingestion
CFD	Computational Fluid Dynamics
FFD	Free Form Deformation
FVM	Finite Volume Method
HB	Harmonic Balance
ICAO	nternational Civil Aviation Organisation
POD	Proper Orthogonal Decomposition
PSD	Power Spectral Density
RANS	Reynolds Averaged Navier-Stokes
RBF	Radial Basis Functions
RMSE	Root Mean Squared Error
RMSE	Root Mean Squared Error
ROM	Reduced Order Method
SBRF	Surrogate-Based Recurrence Frame- work
TI	Time Instance
VLM	Vortex Lattice Method

Symbols

(x, y, z)	Cartesian coordinates, (m,m,m)
α	Angle of attack, deg
β	Pitch angle of the propeller blade, deg, Hicks Henne bump control parameter
$\boldsymbol{\alpha}$	Design variable vector in adjoint problem
$\bar{\boldsymbol{\tau}}$	Viscous stress tensor
$\boldsymbol{X_n}$	Physical grids vector, entry for each time instance
δ	Airfoil thickness
$\dot{E}_k$	Change in kinetic energy over time, J/s
$\dot{m}$	Mass flow, kg/s
η	Propulsive efficiency
γ	Ideal gas ratio
$\hat{\boldsymbol{u}}$	Fourier coefficient vector
κ	Thermal conductivity, W/kg.K
λ	Fixed point iteration
$\bar{\mathbf{I}}$	Identity matrix
$\mathbf{F_c}$	Convective flux vector
$\mathbf{F_v}$	Viscous flux vector
$\mathbf{f}$	Body forces vector
$\mathbf{v}$	Velocity vector, *(u,v,w)*
$\mathscr{D}_t$	Derivative operator
$\mathscr{G}$	Iteration operator for HB method
$\mathscr{J}$	Objective function for adjoint equation
$\mathscr{L}$	Lagrangian function

$\mathscr{N}$	Shifted Lagrangian
$\mathscr{R}$	Residual operator
μ	Dynamic viscosity, kg/m.s
Ω	Cell volume
ω	Frequency
ϕ	Blade inflow angle, deg
ψ	Propeller blade rotation angle, deg
ρ	Free stream density, kg/m^3
τ	Pseudo time for dual time stepping method
$\tilde{\boldsymbol{U}}$	Conservative variable vector
ξ	Scaling factor
A_n, B_n	Fourier series coefficients
B^i	Bezier coefficients for FFD method
C_d	Drag coefficient
C_l	Lift coefficient
C_Q	Torque coefficient
C_T	Thrust coefficient
$C_{L_{RMSE}}$	Lift coefficient Root Mean Squared Error value
$C_{l_{RMSE}}$	Lift coefficient RMSE
D	Drag force, N, Propeller diameter, m
$D_{k,n}$	Diagonal matrix entry for frequency k at time instance n
E	Energy, J
e_0	Total energy, J
$E_{k,n}$	Fourier transform of entity for frequency k at time instance n
H	Special operator matrix
h_0	Total enthalpy, J
L	Lift force, N
M	Mach number
M_n	Mesh deformation algorithm differentiable function
N	Number of frequencies used for the harmonic balance operator
n	Propeller speed, rev/s
N_b	Number of blades in the propeller disk
P_{in}	Power supplied by the engine, W
p_i	Static pressure at station i, kPa
P_{out}	Useful power delivered by propeller, W
p_{T_i}	Total pressure at station i, kPa
R	Universal gas constant, J/kg s
r	Propeller radius, m
T	Static temperature, K
T	Thrust, N, Number of time instances for RMSE determination
t	Time, s
U	Conservation variable vector
u_i	Velocity at station i, m/s
v_n	Normal velocity experienced by the airfoil section, m/s
v_t	Tangential velocity experienced by the airfoil section, m/s
v_{APP}	Apparent velocity experienced by the airfoil section, m/s
V_{ax}	Axial velocity experienced by propeller blade section, m/s
V_{eff}	Effective velocity experienced by propeller blade section, m/s
V_{in}	Velocity entering the thrust engine, m/s
V_{out}	Velocity leaving the thrust engine, m/s
v_{rot}	Rotational velocity experienced by the airfoil section due to the propeller blade rotation, m/s
V_{tgt}	Tangential velocity experienced by propeller blade section, m/s
X	Bezier box coordinates, (u,v,w), Input coordinate for Hicks Henne bump function
X_n	Fast Fourier transformation of frequency k
Y	Hicks Henne bump function
y_t	To be found RMSE parameter

List of Figures

List of Tables

1
Introduction

The Advisory Council for Aviation Research and Innovation in Europe (ACARE) have set a challenging goal for protecting the environment and Earth's energy supply. The aim is to have reduced CO_2 emissions per passenger per kilometre by 75% and the NO_x emissions by 90% in 2050 relative to 2000 [1]. According to Graver et al., the total CO_2 emissions from commercial aircraft operations in 2019 accumulated to 2.4% of the global CO_2 emissions from fossil fuels. This was a 32% increase over the five years before and 70% higher than assumed under the projections by the International Civil Aviation Organisation (ICAO) [28].

(a) EAG Hybrid Electric Regional Aircraft (HERA) 70

(b) Safran Open Rotor design study

Figure 1.1: Propeller concepts [3]

Since the dawn of the turbofan era, the overall efficiencies of the engines have been increasing steadily. Early turbojets achieved overall efficiencies of around 20%, with early turbofans increasing this value to 25%. Turbofan engines of today have overall efficiencies of around 35% [47]. The main drivers behind this efficiency increase are the higher turbine inlet temperatures and the increased bypass ratios. However, current aircraft propulsion trends are using propellers or propfans for their excellent propulsive efficiency. Alternatively, distributed propulsion is also of great interest to larger aircraft manufacturers. As seen in Figure 1.1, the future of the aircraft industry shows propellers to be the chosen path to pursue. Propfans have shown propulsive efficiencies higher than turbofans at speeds up to $M = 0.7 - 0.8$ for a given range, as shown in Figure 1.2. This higher efficiency can be attributed to the propfan being able to accelerate a larger mass of airflow by a smaller velocity increment. This can also be described as the bypass ratio increasing beyond what is possible for turbofans. Propfans do have a drawback; noise. Without the absorbing inner lining from the cowling, the noise created by the propfans can be perceived as uncomfortable for both the passengers and the airport environment.

Quantifying the efficiency of a propeller can be done using Equation 1.1 and is called the Froude efficiency. To achieve high propulsive efficiency, V_{out} should be close to V_{in}. Assuming the same mass of air is being accelerated, a small velocity increase leads to a small thrust. The efficiency can be high for a normal thrust level when a large mass of air is being accelerated by a small velocity increment. Ordinary and advanced turboprops (propfans) use a large diameter propeller, alternatively called a high Bypass Ratio engine,

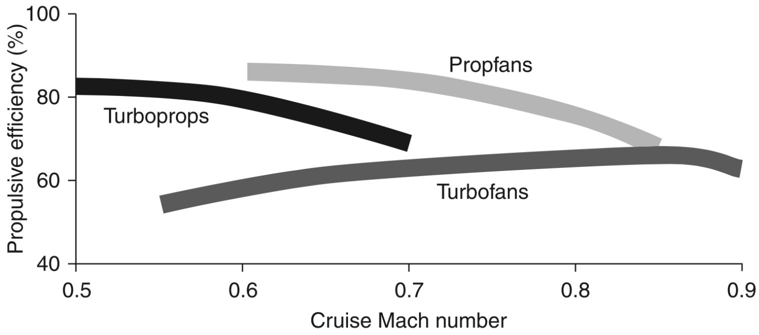


Figure 1.2: Propulsive efficiency for different engine configurations [47]

to accelerate a large mass of air by a small velocity increment and thus increase the propulsive efficiency. This is why the bypass ratio has been increasing steadily over the last couple of years in turbofan design.

$$\eta = \frac{P_{out}}{P_{in}} = \frac{TV_{in}}{\dot{E}_k} = \frac{2}{1 + \frac{V_{out}}{V_{in}}} \tag{1.1}$$

One of many ways to increase the efficiency of the aircraft is to integrate the advanced propulsion system in such a way that the aerodynamic efficiency is enhanced. Something like BLI is an example of this integration of the propulsion system in the aircraft to retrieve some of the energy which ordinarily would be lost. The Advanced Propulsion and Power Unit (APPU) project uses BLI and aims to redesign the Airbus A320neo to increase propulsive efficiency and therefore decrease CO_2 emissions [2]. This is done by means of transforming the Auxiliary Power Unit (APU) into a power and propulsion unit, the APPU. Furthermore, a propeller will be added to the aft part of the aircraft's tail cone to generate thrust. The APPU is a larger version of an APU due to the requirement of powering the electrical systems and the propeller. The aim is to be able to run the APPU on both hydrogen and kerosene, such that when using hydrogen, no CO_2 or NO_x is emitted. With this system, a 35% decrease in CO_2 emission over a flight of up to 2000 nm [2] is set to be achieved.

The propeller placement is an important factor in the efficiency gain of the APPU project. The boundary layer created by the fuselage of the airplane is ingested and accelerated by the propeller. This phenomenon is called Boundary Layer Ingestion (BLI) and has been applied to missiles, torpedoes, and ships [52]. The boundary layer grows as it progresses over the skin of any object and transitions from a laminar boundary layer to a turbulent boundary layer after some distance. The boundary layer also grows in thickness to a substantial length over the full length of the fuselage. Boundary layer flow is characterised by a slower velocity than the freestream velocity. When ingested by the propeller, less power is required to accelerate the air. Furthermore, the wake generated by a normal aircraft that dissipates into the free stream air is less pronounced for a BLI configuration aircraft. In ideal conditions, the propeller accelerates the boundary layer so the aircraft does not leave a wake in the freestream air. It cancels the wake and thus leaves less energy behind [39]. These two phenomena are the cause for a BLI configuration to increase the overall efficiency of the aircraft. In the case of a BLI configuration, the Froude efficiency, which is widely used as an efficiency measure, can grow to values greater than 100% [52].

Propellers that make use of the BLI principle deal with more distortion of the inflow due to the turbulent boundary layer ingested relative to non-BLI configurations. Steady-state models, as the name already suggests, can only model airflow, which does not change over time. For the case of a BLI propulsion system, this would mean that the fidelity of the model would drastically be reduced. Unsteady or even quasi-steady models can closely recreate the unsteady inflow conditions generated by the turbulent or unsteady boundary layer. This is why it is interesting to use such an unsteady or quasi-steady model for the airfoil shape optimisation problem. Analytical methods have been created to determine the in-plane loads of a propeller in isolated conditions at an angle of attack [16]. With this method, the unsteadiness of the propeller can be modelled with a steady inflow model. This method uses the rotational characteristic of the propeller to introduce a change in tangential inflow velocity due to the advancing blade experiencing a higher velocity and the retreating blade experiencing a lowered velocity. However, this method only suffices if the unsteady inflow characteristic is of a perfect sinusoidal shape and can be solved by the use of superposition. The superposition of two flows is a very strong method of computing such an inflow condition. However, this method relies on a number of assumptions of the linearised potential flow theory. Most complex unsteady flows are not able to be solved using superposition because the assumptions made for this theory are violated. This means that due to the complexity, the flow needs to be simulated as a whole. Therefore, numerical methods are preferred for the computation of unsteady, non-uniform inflow effects on propellers. Reynolds Averaged Navier-Stokes (RANS) simulations are an example of these numerical computations[53].

With the current technology of high-performance computing, it is possible to run these numerical simulations. However, unsteady simulations using a large number of design parameters still prove to be too complex to solve for a time-accurate flow. Reduced Order Models (ROM) can provide the solution to the problem of accurately modelling unsteady flows used for shape optimisation problems at a lowered computational cost. Due to the periodicity of the propeller motion, a suitable reduced order model is the Harmonic Balance (HB) method [27]. This ROM offers advantages over time accurate flows when a discrete number of known frequencies describe the flow characteristics. Combined with a propeller blade design problem, this is a very interesting strategy due to the main unsteady effects coming in a multiple of the base rotational frequency. For this reason, the frequencies used in the HB method are a direct relation between the rotational velocity of the propeller and the number of perturbations experienced per revolution [22].

In optimisation problems employing ROMs, the issue is to accurately determine the necessary design variable gradients. Adjoint methods have been studied for this purpose and have drawn attention due to their ability to resolve a large number of design variables simultaneously [25]. Finite differences methods compute an entire flow solution for each design parameter in order to compute the design vector gradients. The adjoint method determines the gradient vector by solving the direct flow equations first. These equations solve for the coefficients of the adjoint equations, which is subsequently solved [33]. This means that for any number of design variables, the adjoint method only has to solve two systems of equations which are comparable in computational cost. The adjoint optimisation method is very efficient in problems where the number of design variables exceeds the number of objective functions by a large amount [55]. So combining a ROM with the adjoint method would yield a very efficient optimiser. One effective and efficient adjoint HB solver has been created in the open-source code SU2 by Rubino et al.[45]. In this research, the HB solver created by Rubino et al. will be used to run the simulations with an additional performance indicator in the form of the lift coefficient response amplitude.

Many propeller or airfoil optimisation works are focused on reducing the average drag coefficient, as done by Rubino et al. [45] and Economon et al. [20]. Economon et al. compared the adjoint approach to unsteady design problems to finite differences methods but optimised for drag in both cases. In Economon's work, the average drag coefficient is decreased. The amplitude of the lift response increased by a factor of 2. In the work of Marinus[41], a 3D blade optimisation is performed with an increase in thrust as their objective. When unsteady propeller optimisation is considered, another important factor of propeller performance has to be taken into account, the amplitude of the lift response of the airfoil. From other literature, it can be deduced that time-varying blade loading and, therefore, the time-varying lift coefficient of the airfoil is of great influence on the noise emissions of the propeller [29, 56]. Therefore, a metric in the form of a Root Mean Squared Error value of the lift coefficient to quantify the amplitude of the time-varying lift coefficient can be used to very coarsely estimate the severity of noise emissions by the propeller.

The noise generation principles are described in the research by Kurtz [36], reviewing the aerodynamic noise from propellers. Various sources of noise which are of interest for this research are the thrust and torque noise blade slap noise. Thickness noise is also a form of periodic noise generation and is generally found to be small compared to the thrust and torque noise [36]. Torque and thrust noise is the mechanism that is caused by the pressure difference which acts over the propeller blade. The pressure difference is steady over the propeller for uniform inflow conditions. While for non-uniform conditions, the pressure is oscillating during the rotation. Blade slap noise is a high-amplitude periodic noise combined with modulated vortex noises. The vortex noises are caused by the impulsive fluctuation of the forces acting on the blades. The mechanisms at the root of these problems are blade vortex interaction, the stalling and reattachment of the flow that happens periodically over the blade. Blade vortex interaction is also responsible for the creation and destruction of shockwaves which occur over the blade due to periods of supersonic flow [36].

Using the description of the two noise-generating mechanisms and the knowledge from literature that time-varying blade loading is of influence to the noise generation. The conclusion can be drawn that the amplitude of the lift response is of importance to the noise generation of the propeller. Limiting this by means of the $C_{L_{RMSE}}$ is thus a viable option when designing for the optimal airfoil and propeller blade shape.

Research objective

Based on the presented findings, an argument can be made that there is a demand for a propeller design method which is able to perform in unsteady conditions while improving the performance regarding the $\frac{C_L}{C_D}$ ratio and also the forced lift response. The propeller consists of two equally important parts, the blade planform and the airfoil. In propeller operation, the apparent velocity is the velocity which the propeller blade experiences. This is a combination of the rotational velocity and the freestream velocity of the aircraft. This means that the propeller blade is going to perform in the transonic range. The wave drag will become an important factor from this point since shockwaves will start to form over the airfoil. This effect, combined with the limited research into unsteady airfoil optimisation, led to the choice for this research to be pointed in the direction of airfoil optimisation rather than planform optimisation. Reducing the adverse pressure gradients over the airfoil and the shockwave intensity is the method's main goal, as well as reducing the amplitude of the forced lift response of the airfoil due to the time-varying inflow conditions. With this in mind and the drive to speed up the optimisation process, the choice is made to use an Euler solver. At a later stage, more complex solvers can be used for this design method to include viscous effects by, for instance, using the full set of Navier-Stokes equations. The design method created should be able to produce an airfoil which is able to perform in off-design conditions, the unsteady inflow of the APPU project. For this, the adjoint method will be used due to its efficiency in optimisation problems combined with the HB method created by Rubino et al. [45]. Therefore, the use of the open-source simulation and design software SU2 is implied. The research objective can be formulated for the current study as

To create and run an efficient aerodynamic design method for a 2D propeller airfoil while using the Harmonic Balance method combined with the Adjoint method to minimise the drag of the airfoil and limit the lift response to unsteady inflow.

The research objective stated above can be divided into the following sub-objectives.

- Create a method to determine the 2D lift response of the airfoil in unsteady flow conditions.
- Perform a 2D Harmonic Balance Adjoint design optimisation problem with the drag coefficient as the objective function for an unsteady sine wave inflow condition.
- Determine the real inflow conditions at the propeller disk using data from the APPU project and implement this in the design problem.
- Perform the 2D HB Adjoint design optimisation problem with the drag coefficient as the objective for the real unsteady inflow conditions.
- Perform the 2D HB Adjoint design optimisation problem above while step wise lowering the lift coefficient amplitude each optimisation run

Report Outline

Some background into propeller performance and analysis is given in chapter 2, where the basics of propeller mechanics and how the performance of propellers can be quantified will be explained. Next, chapter 3 deals with the theory behind reduced order models, the harmonic balance solver and the adjoint equations specifically developed for the harmonic balance solver. Followed by the reconstruction of the real inflow of the APPU propeller in chapter 4. The general solver setup, including the harmonic balance frequency selection, grid generation and airfoil parameterisation, is presented in chapter 5. Next, the optimisation problem setup is discussed in chapter 6, where the secondary performance indicator, along with the gradient validation and the baseline performance of the NACA 0010 is described. The results follow in chapter 7 and finally, the conclusion and recommendations in chapter 8.

2

Why a propeller?

In this chapter, the basics of the APPU project are explained as well as the concept of propellers and the importance of the propeller in this project. Aerodynamic performance principles and methods of the propeller are presented in this chapter for the purpose of general background.

In section 2.1, the background behind the APPU project and the goal set to achieve the flightpath 2050 aim are explained. Followed by the basics of propeller theory in section 2.2, and the propeller performance method is elaborated upon in section 2.3. The methods of propeller performance analysis are presented in section 2.4 with the RANS solver, and its governing equations are explained in section 2.5.

2.1. Advanced Propulsion and Power Unit (APPU) Background

Ever since the dawn of powered flight, innovation has led the aircraft propulsion sector towards the improvement of turbojet and turbofan engines. These engines were seen as the future of aircraft propulsion and still are leading in terms of usability at high cruise speed and high altitude. First introduced in 1930 by Frank Whittle, the turbojet has propelled many commercial and military planes [23]. From the 1950s, the first commercial airliners adopted turbojet engine configurations into everyday use, such as the Comet and the Boeing 707 [47].

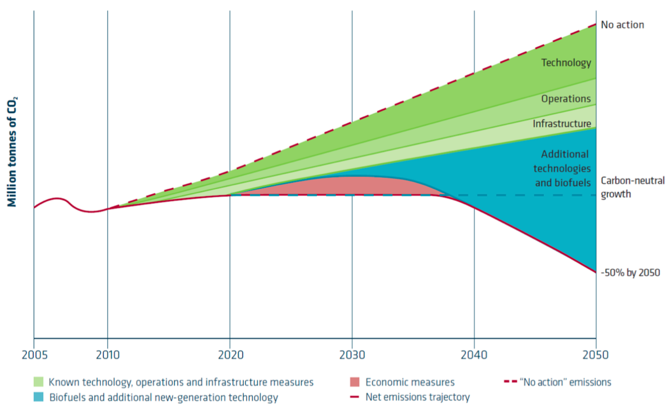


Figure 2.1: IATA's CO_2 emissions reduction roadmap [6]

The technological leaps in aircraft performance were driven by speed, range and comfort in the early twentieth century. This changed with the new millennium when the main driver became fuel efficiency due to the depletion of natural resources and global warming. These two consequences of burning too much fossil fuels led to the creation of the EU Flightpath 2050 by ACARE. They set the goal to half the net aviation carbon emissions by 2050 compared to 2005 [1]. As can be seen in Figure 2.1, without action, the CO_2 emissions will rise to astronomical levels. Figure 2.1 also shows that technology alone is not enough to achieve the -50% CO_2 reduction goal. Additional operational, infrastructure and biofuel improvements are necessary.

The Advanced Propulsion and Power Unit (APPU) project is working towards the Flightpath 2050 goal by using a propeller positioned on the aft part of the tail cone powered by an auxiliary power unit running on hydrogen. With this configuration, the CO_2 emissions for a flight of up to 2000nm can be reduced by 30% [2]. This is already a substantial step towards the EU Flightpath 2050 goal. The CO_2 emission reduction is attributed to the combination of using hydrogen and the use of BLI.

Positioning the propeller on the aft part of the aircraft, as seen in Figure 2.2[1], creates the opportunity to use an effect called Boundary Layer Ingestion (BLI). This effect uses the wake created by the fuselage of the aircraft and accelerates this such that in ideal conditions, no wake deficit is left [39]. The energy which would normally dissipate into the air is now accelerated to free stream velocity and can be considered a gain in efficiency. Due to the positioning, the propeller is introduced to unsteady inflow, and the design needs to adjust accordingly. Van Arnhem has created a method to design a propeller blade for non-uniform inflow conditions however has not taken the unsteadiness of the flow into account [5]. This research takes a look at the real APPU aircraft's unsteady inflow and creates an efficient 2D airfoil optimisation solution to solve for the unsteadiness.

Figure 2.2: Rendering of the preliminary design of the APPU project[1]

2.2. Basic Propeller Theory

As readily known, the fundamentals of propulsion rely on Newton's second law of motion. The force on an object is equal to the change in magnitude and direction of the momentum of an object. In the case of a propeller, the mass flow experiences a velocity difference over the propeller disk is a form of momentum change which will translate to a force acting on the propeller. Combining Newton's second law with the continuity equation leads to the thrust equation, as shown below.

$$T = \dot{m}(V_{out} - V_{in}) \tag{2.1}$$

Propellers produce the velocity increase when looking at Bernoulli's equation by the introduction of a pressure difference created over the propeller disk. The actuator disk theory is a simplified method of determining the propeller performance in accordance with momentum theory, where a pressure difference is assumed over an infinitely thin propeller disk. The flow actuator disk theory is considered incompressible, inviscid and steady, while the disk loading of the propeller is assumed to be constant as well. For this reason, the effects of rotation are neglected [17]. With this method, only the pressure, total pressure and velocity distribution can be determined over the propeller disk.

The actuator disk theory is ideal for calculating the initial performance characteristics of the propeller and understanding the flow field before and aft of a propeller disk. However, with actuator disk theory, no

[1]Accessed 2-3-2022 - `https://www.tudelft.nl/lr/appu`

blade geometry is taken into account, so this method is not suited for any type of blade design. Van Arnhem [5] used actuator disk theory as an initial understanding of how the flow field behaves through a propeller before starting the design optimisation problem. Lv et al. [39] and Blumenthal [10] have used this method to investigate the inflow and effect of BLI on the performance of the aircraft.

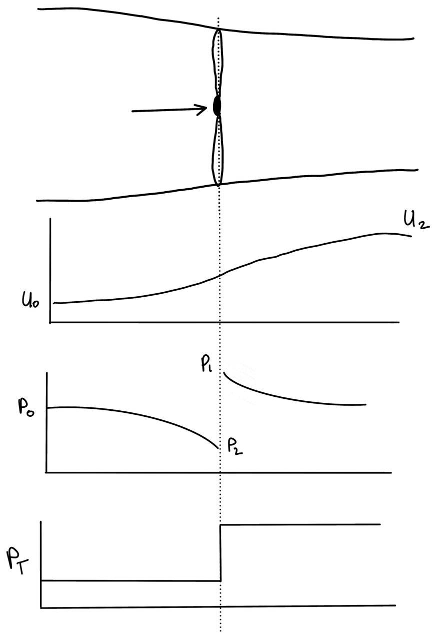


Figure 2.3: Schematic view of the actuator disk theory with the slipstream contraction, velocity, pressure and total pressure variation across the propeller disk. Adapted from Kerrebrock [35]

The basic principle of a propeller is that it is a collection of rotating wings that generate lift and drag. A propeller blade can be divided into multiple airfoil sections, with each having a different pitch angle, chord and thickness. The forces acting on the airfoil are presented schematically in Figure 2.4. The torque and thrust of the propeller are a consequence of the lift and drag force acting on the airfoil. Due to the rotational velocity as well as the axial velocity, the airfoil section experiences a given inflow angle ϕ, varying over time in unsteady situations. The axial and rotational velocity components combined is called the effective or apparent velocity. This is what the airfoil section sees and operates with during the propeller revolution. In a 2D case such as Blade Element Momentum (BEM) theory, the unsteady effects can be simulated by varying the angle of attack α of the airfoil while keeping the effective velocity constant.

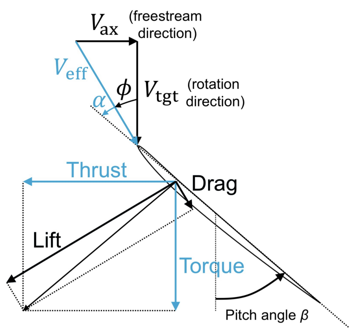


Figure 2.4: Thrust generation principle of propellers. Taken from AE4130/AE4135 [48]

The torque and thrust per blade element can be calculated using Equation 2.2 and Equation 2.3 with ϕ being the inflow angle and r the radial distance from the centre of rotation to the specific blade element.

$$dT = dL\cos\phi - dD\sin\phi \tag{2.2}$$

$$dQ = (dL\sin\phi + dD\cos\phi)r \tag{2.3}$$

These equations only represent the 2D sectional loads of a propeller blade. Computing the 3D loads requires integration over the span of the blade. The total thrust is given by summing the sectional loads and multiplying by the number of blades N_b as can be seen in Equation 2.4. For the total torque, the process is the same and is given in Equation 2.5.

$$T = N_b \int_{hub}^{tip} dT dr \tag{2.4}$$

$$Q = N_b \int_{hub}^{tip} dQ dr \tag{2.5}$$

Additional 3D effects need to be considered to determine the total thrust generated by a propeller. Similarly to a finite wing, the blade loading of a propeller blade tends towards zero at the tips. The Coriolis force acting in a chordwise direction leads to a favourable pressure gradient which is able to delay the separation over the airfoil. Where the centrifugal force can cause a spanwise pumping effect which in turn causes boundary layer thinning [12]. The spanwise flow structures due to rotational effects are often called the Himmelskamp Effect. For these 3D effects, there are correction factors which ensure that the summation of the 2D forces represents the 3D forces. Due to the propeller being a finite wing, a tip vortex is generated when in operation. These vortices will create losses in the process. The Prandtl tip loss factor [31] can be implemented to account for these losses in the process of calculating the 3D forces.

2.3. Propeller Performance

Propeller engines such as turboprops have a higher bypass ratio than turbojets producing a higher Froude propulsive efficiency. As shown in Figure 1.2, propellers lose their efficiency superiority compared to turbofans at higher cruise speeds. In this situation, the propeller tip speeds are approaching the transonic range and create shock waves locally over the airfoil. These shock waves increase drag and noise levels and therefore lower the overall propulsive efficiency.

Propeller performance is often described using the non-dimensional coefficient given in Equation 2.10 and Equation 2.11. This is to compare different size propellers in different configurations and to determine the loading of the propeller.

$$C_T = \frac{T}{\rho n^2 D^4} \tag{2.6}$$

$$C_q = \frac{Q}{\rho n^2 D^5} \tag{2.7}$$

Another non-dimensional number which is of interest is the advance ratio J. It represents the distance a propeller travels while making one full revolution, and can be expressed using the freestream velocity V_∞, propeller diameter D and propeller rotational speed n.

$$J = \frac{V_\infty}{nD} \tag{2.8}$$

The above-mentioned performance indicators are applicable for full propeller blade or disk systems. Since this work will investigate a 2D airfoil, some alternative performance indicators have to be used. When looking at Equation 2.10 and Equation 2.11, there are non-dimensional numbers which are at the root of these performance indicators which can be used. As discussed in section 2.2, the thrust and torque of a propeller can be determined using the 2D airfoil lift and drag forces which rely on the lift and drag coefficient. When substituting Equation 2.2 into Equation 2.4 leads to:

$$T = N_b \int_{hub}^{tip} \left(dL\cos\phi - dD\sin\phi\right) dr \tag{2.9}$$

Correction factors still need to be applied to the total thrust to account for the various 3D effect. However, for the purpose of deriving a 2D performance indicator, this is not necessary. The sectional lift and drag are the lift and drag force generated by the 2D airfoil section and can be expressed as:

$$dL = C_L \frac{1}{2} \rho V_{eff}^2 \quad (2.10) \qquad dD = C_D \frac{1}{2} \rho V_{eff}^2 \quad (2.11)$$

This leads to the conclusion that the thrust coefficient is proportional to the lift and drag coefficient. With the drag coefficient having an inverse proportionality to the thrust coefficient.

$$C_T \propto C_L, C_D \quad (2.12)$$

Similarly, the torque coefficient can be derived like this, leading to the same proportionalities. This means that using the 2D lift and drag coefficient as performance indicators is a possibility to represent the propeller blade performance as a whole.

The noise generated by a propeller compared to the shrouded turbofan engines can be experienced as a nuisance to the passengers and the environment around the airport [49]. The casing around a turbofan engine absorbs some of the noise created by the fan. However, the casing can also be lined with an acoustically absorbing liner to further decrease the noise emissions [7]. Dedicated noise calculations are not a part of this research. However, as discussed, the assumption is made that a decrease in lift response amplitude resulting from unsteady effects results in a lowered noise emission from the propeller. So if the amplitude of the lift response is kept to a minimum, the argument can be made that the noise induced by the propeller blade is kept to a minimum. So as a secondary performance indicator for this research, the Root Mean Squared Error value of the lift coefficient $C_{L_{RMSE}}$ can be used. This value calculated the error compared to the time average lift coefficient of the airfoil.

2.4. Propeller Analysis

In order to determine the propeller performance, the airflow over the airfoil needs to be resolved. Using analytical methods such as the actuator disk method, Vortex Lattice Method (VLM) or Blade Element Momentum Theory (BEMT) are the most efficient, but the solvers are created using a number of assumptions. Computational Fluid Dynamics (CFD) models are capable of capturing the flow physics around any aerodynamic shape much more accurately than analytical models. The drawback to CFD models is that computationally it is much more expensive due to the large number of calculations made compared to analytical models. CFD methods use equations derived from the conservation laws as governing equations to resolve the fluid flow.

The analytical methods, such as VLM and BEMT, are not optimised to resolve unsteady flight conditions. These methods rely on a number of assumptions and are generally only applicable in steady conditions. Some examples of unsteady flight conditions are gusty winds, flutter phenomenon and fast manoeuvring flight at a high angle of attack. The unsteady aerodynamic loads which come from these conditions can have ample effect on the aircraft's aeroelastic and dynamic stability characteristics. The main source of unsteady effects or nonlinearities are shock wave motions and separated flows. These nonlinearities have a substantial impact on the performance and stability of the aircraft and can cause limit cycle oscillations [58]. In the creation of VLM and BEMT, a steady flow assumption was made and is therefore not applicable to this study.

With the current leap in technology of computing power, the limit to what was possible to simulate using a CFD model has been steadily expanding. Even combining numerical CFD models with design optimisation problems has proven to be successful. Cousins et al. [15] have used CFD to analyse and design a distortion-tolerant BLI fan. The fan and inlet are optimised for aerodynamic and vibrational modes since the distortion makes for extra vibration during operation. The experiment conducted by Cousins to validate the research confirmed the successful design using CFD.

Numerical models are also interesting when using complex inflow characteristics such as BLI or another unsteady flow phenomenon. Due to the highly coupled nature of a BLI propulsion system, the aerodynamical effects of BLI are not easy to analyse using analytical methods. Therefore CFD is used in this case to resolve the flow to a more detailed level and understand what happens before the flow enters the propulsor. Yildirim [57] used CFD to optimise the highly coupled STARC-ABL propulsion system, where the main objective was to change the shape of the casing to improve efficiency.

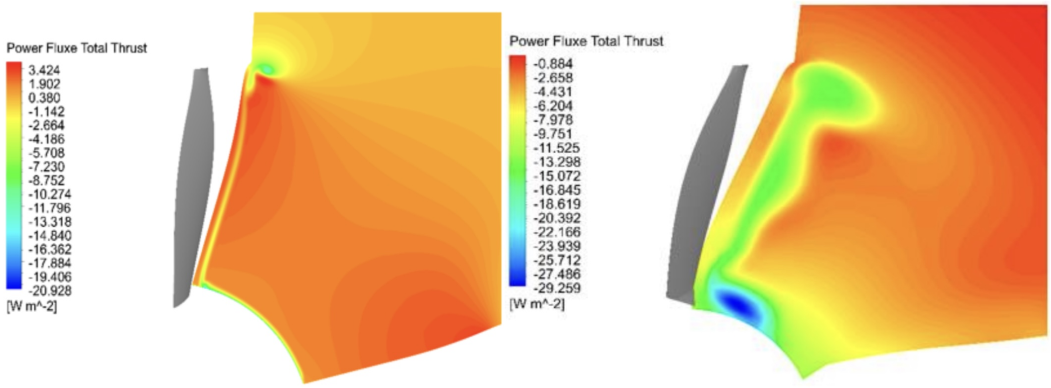


Figure 2.5: Example of CFD computations used for predicting propeller performance taken from [44]

In this research, a design optimisation problem is resolved using SU2 [21], an open-source and readily available CFD software. Depending on what kind of turbulence modelling is used can be moderate in its computational expensiveness. Due to its open-source nature, it is possible to alter the source code in order to add or change features of the software. In pursuit of creating a new parameter for propeller performance, this is the best choice in CFD software. Furthermore, the function that is chosen to be used in this research has been developed in SU2 by Rubino et al. [45].

2.5. Navier Stokes Based Aerodynamic Solver

In this research, a compressible inviscid Navier-Stokes-based aerodynamic solver will be used. The simplified compressible Navier-Stokes equations, or Euler equations, will be solved within SU2. With the Euler equations, some fidelity is lost in the form of the viscous flows around the airfoil and the turbulence that it creates. It limits the optimisation problem in drag coefficient computation due to the absence of the viscous drag component. The airfoil shape will be slightly different than what is expected for a full viscous simulation. The difference will mainly be in the aft part of the airfoil. The absence of viscous forces and separation due to the inviscid simulation may lead to sudden shape changes. The main objective for the optimiser will be to create a less sudden increase in velocity over the airfoil. A shallow ramp up the suction side of the airfoil would enable a less sudden increase in velocity. However, this leads to the airfoil returning to the chord line in a sudden way and would normally lead to separation.

On the other hand, the Euler solver is very efficient in finding surface sensitivities by using differential geometry formulas. With the surface sensitivities, the optimiser can see exactly what shape changes have the most effect with regard to the objective function. Economon, Baeza et al., and Bueno et al. used an Euler solver in their research to find the optimal shape for an airfoil. [8, 11, 19]. The Euler equations are the Navier-Stokes equations, derived from the conservation laws, mass, momentum and energy in the absence of thermal conductivity and viscosity. In this section, the conservation and Navier-Stokes equations are presented.

2.5.1. Conservation laws

The Navier-Stokes equations make use of conservation laws to describe a compressible viscous flow. Eliminating the thermal conductivity and the viscous effects leads to the creation of the Euler equations. For the conservation of mass, momentum and energy, a finite control volume is assumed in space. The fluid flowing through the control volume can be considered to be either static or moving. The equations are valid for any point of a three-dimensional inviscid flow and are presented in their differential form.

Mass

Where t represents time and $\mathbf{v}$ is the velocity vector.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \tag{2.13}$$

Momentum

With **f** representing the body force vector. However, these are often not considered in an absolute frame of reference for a propeller application.

$$\frac{\partial}{\partial t}(\rho\mathbf{v}) + \nabla\cdot(\rho\mathbf{v}\mathbf{v}) + \nabla p = \rho\mathbf{f} \tag{2.14}$$

Energy

Where h_o is total enthalpy, e_o is total energy expressed as $e_o = e + \frac{1}{2}\rho u^2$.

$$\frac{\partial}{\partial t}(\rho e_o) + \nabla\cdot(\rho h_o\mathbf{v}) = \rho(\mathbf{f}\cdot\mathbf{v}) \tag{2.15}$$

2.5.2. Governing Equations

As stated above, the Euler solver in SU2 solves the simplified Navier-Stokes equations, which are called the Euler equations. Hence the solver is named the Euler solver. These equations are presented in their differential form below, with S being a generic source term.

$$\mathscr{R}(U) = \frac{\partial U}{\partial t} + \nabla\cdot\mathbf{F}^c(U) - S = 0 \tag{2.16}$$

In the current solver, the conservation variables are the working variables which are presented in Equation 2.17.

$$U = \left\{\rho, \rho\mathbf{v}, \rho E\right\}^{\mathsf{T}} \tag{2.17}$$

The convective flux given as $\mathbf{F}^c$ in Equation 2.16 is presented below.

$$\mathbf{F}^c = \left\{\begin{array}{c} \rho\mathbf{v} \\ \rho\mathbf{v}\times\mathbf{v} + p\bar{\mathbf{I}} \\ \rho E\mathbf{v} + p\mathbf{v} \end{array}\right\} \tag{2.18}$$

With ρ,p and T being the fluid density, static pressure and temperature, respectively. The flow velocity in Cartesian coordinates is presented as $\mathbf{v} = \{u, v, w\}^{\mathsf{T}} \in \mathbb{R}^3$ and E the total energy per unit mass. Assuming the flow consists of a perfect gas with a specific heat ratio of γ and a universal gas constant R. The closure of the system can be defined as the pressure:

$$p = (\gamma - 1)\rho\left[E - 0.5(\mathbf{v}\cdot\mathbf{v})\right]. \tag{2.19}$$

The temperature is computed following the ideal gas relation.

$$T = \frac{p}{(\rho R)} \tag{2.20}$$

The Euler solver uses a finite volume method (FVM) to discretise the equations in space. The FVM is built upon a dual grid with vertex-based schemes using a standard edge-based data structure, evaluating the convective and viscous fluxes at the edge midpoint.

2.5.3. Linear Preconditioners

Flow discretisation often leads to large linear systems of equations that have prohibitive memory requirements. In some cases, it is accepted that iterative methods are used. However, they are not as robust as direct solvers. This is why the alternative option is to improve the robustness of the iterative solvers by using an accurate preconditioner. The process of using a preconditioner makes the iterative solver practically possible [13]. The preconditioners have had different approaches to solving the problem over the years.

Chorin [14] added a time derivative of the pressure term to the continuity equation artificially with a multiplication factor α. The resultant scheme is a symmetrical hyperbolic system for inviscid terms. This means that the system was well-posed a numerical method for hyperbolic systems could be used to advance the system in time. α was then chosen for the system to come to a steady state in a swift manner.

In another work, Turkel [54] expanded the method of Chorin by adding a pressure time derivative to the momentum equation and introducing another parameter β, leading to a system being optimised for both α and β. The process of preconditioning can be accelerated by using processes like Biconjugate Gradient Stabilised (BiCGSTAB) or Generalised Minimal RESidual (GMSRES). However, the choice of the preconditioner is of much greater importance than that of the accelerator [13].

Within the SU2 framework, there are four possibilities for linear preconditioners [21], each having its benefits. The $JACOBI$, Jacobi Block, preconditioner has the lowest computational overhead and also has the lowest effectiveness. For a solver efficiency objective, this would be a good option if the ultimate accuracy is not of great importance. The $LUSGS$, Lower-Upper Symmetric Gauss-Seidel, preconditioner is a step up from the $JACOBI$ preconditioners concerning the computational overhead and effectiveness. However, the memory footprint of this preconditioner is the lowest of the four possibilities. So for a simulation which has a high memory requirement, this $LUSGS$ would be best. The ILU, Incomplete Lower Upper factorisation with a connectivity-based sparse pattern, preconditioner has the highest cost but also the highest effectiveness. This preconditioner is used when the accuracy of the simulation is of great importance. Finally, the $LINELET$, Line-implicit Jacobi, preconditioner has the distinctive treat of solving tridiagonal systems along grid lines normal to walls and using a Jacobi solver everywhere else. For simulations relying on high accuracy along the wall of an object, this preconditioner would suit best.

3

The adjoint method for harmonic balance

As mentioned in section 2.4, the combination of CFD and design optimisation for unsteady inflow is a very computationally expensive problem. Reduced Order Models (ROM) and the adjoint method could substantially decrease the computational time of such a problem. In this chapter, a brief introduction is given to the use of ROMs in section 3.1 followed by the presentation of the ROM, which will be used for this research in section 3.2. Finally, the adjoint method will be discussed in section 3.3.

3.1. Reduced Order Models (ROM)

With the advances in computing power, the limit of solving for degrees of freedom has steadily been rising. Solving complex numerical problems still require a large amount of computational time. In the aim to reduce this time, Reduced Order Models can be used. The creation and usage of ROMs can be explained twofold; providing a method to solve for the dynamics of a system at a lower computational cost, and secondly, providing a means to readily interpret system dynamics [38].

ROMs have the ability to reduce the complexity of data computation and thus increase the efficiency of the numerical problem. This allows for a much larger number of degrees of freedom in a shape optimisation problem, which up until now has been limited to a steady-state approach due to time-accurate simulations needing much more computational time to resolve for the number of degrees of freedom. The possibility of simulating unsteady conditions for shape optimisation problems can greatly improve the accuracy of the design predictions. This leads to a more realistic representation of the airfoil performance and thus leading to an increase in operational efficiency. The use of ROMs can be offset by the potential loss in accuracy and robustness of the model, which in turn leads to a large number of different reduced-order models.

Some examples of ROMs are Proper Orthogonal Decomposition (POD), Surrogate-Based Recurrence Framework (SBRF), Volterra theory, Radial Basis Functions (RBF) and Harmonic Balancing (HB). Li et al. have created an alternative method, a reduced order model for unsteady aerodynamics based on machine learning [37]. In their work, they have used a long short-term memory network in order to account for various airfoil geometries.

Within the explanation of reduced order model specifics, the nonlinearity term is often used. However, both the structural and fluid dynamics of the problems used in computational physics are of the linear kind. Meaning that the solution of interest can be created out of a linear combination of a set of basis functions [38]. POD and HB models, for instance, apply a linear technique to solve for unsteadiness. Furthermore, to solve for nonlinearities, these techniques account for the nonlinear coupling of terms acting within the linear space defined by the basis functions. The nonlinearity term refers to the shocks that occur over the airfoil, which are resolved using these methods. Since the method of Rubino et al. [45] has proven to be successful in turbomachinery applications for this research, the same HB solver will be used to increase the computational efficiency of the optimisation problem.

3.2. Harmonic balance method

In turbomachinery design problems, the Harmonic Balance method has been readily implemented in design optimisation problems [45], as well as for the prediction of unsteady aerodynamic loads of helicopters [22]. Turbomachinery components and helicopter blades experience a highly periodic flow in their operation, also when encountering unsteady flow phenomena. The harmonic balance method is a good option because this method can deliver significant savings for time-periodic, unsteady, nonlinear problems [38]. The propeller disk is an example of such a problem. Therefore, the HB method is a good option for this research. Anand et al. [4] have used the harmonic balance method to run design optimisations for the aeroelastic behaviour of a transonic compressor blade and a supersonic impulse turbine blade. The harmonic balance method used in their work is based on the multi-frequency harmonic balance solver presented by Rubino et al. [45]. In the work of Anand et al., they found that this method was very efficient in the computation of fully-turbulent quasi-periodic unsteady flows. Keeping in mind the aim to create an efficient optimisation method, the efficiency findings by Anand et al. confirm the choice of an HB solver for the problem of this research.

In the current research, the objective is to create an efficient design optimisation method which is able to improve the propeller blade performance of the APPU propeller. Keeping in mind that the inflow of an installed propeller has a highly periodic nature. Therefore, the logical decision is to also include the harmonic balance method in this optimisation problem. The method that is used for the current work has been developed by Rubino et al. [45]. In their work, the objective was to optimise airfoil shapes for different applications using a harmonic balance solver. These different applications were all periodic in nature, and it was found that the choice of a harmonic balance solver greatly improved the efficiency of the entire design optimisation problem. The following subsections will discuss the equations necessary to implement the Harmonic Balance solver into SU2. This is divided into three pieces, time discretisation, harmonic balance operator and the harmonic balance in the time domain. All the equations given in this paper are taken from the work of Rubino et al. [45].

3.2.1. Time Discretisation

In order to solve the flow equations for a discrete number of frequencies, time discretisation has to be applied. So for a cell volume, Ω, the Navier-Stokes equation can be transformed in its semi-discrete form for $t > 0$ as given in Equation 3.1. With the conservation variables in vector form for a three-dimensional problem where ρ is density, v_x is the velocity in each specific direction, and E the total energy: $\boldsymbol{U} = (\rho, \rho v_1, \rho v_2, \rho v_3, \rho E)$.

$$\Omega \frac{\partial \boldsymbol{U}}{\partial t} + \mathscr{R}(\boldsymbol{U}) = 0 \tag{3.1}$$

The assumption is made that the generic cell volume Ω and its boundary $\partial\Omega$ vary their position in time without deforming with velocity $\boldsymbol{u}_\Omega$. The residual operator for spatial integration of $\boldsymbol{F}^c$ and $\boldsymbol{F}^v$, the convective and viscous fluxes, respectively, is expressed as $\mathscr{R}$. Using the work of Donea et al. [18], the residual operator can be expressed as a function of the convective and viscous flux using an Arbitrary Lagrangian–Eulerian (ALE) formulation.

$$\begin{aligned} \mathscr{R}(\boldsymbol{U}) = f(\boldsymbol{F}^c, \boldsymbol{F}^v) \quad for \quad \Omega, t > 0 \\ \boldsymbol{v} = \boldsymbol{u}_\Omega \quad for \quad \partial\Omega, t > 0 \end{aligned} \tag{3.2}$$

With the convective and viscous fluxes defined as:

$$\boldsymbol{F}^c = \left\{ \begin{array}{c} \rho(\boldsymbol{v} - \boldsymbol{u}_\Omega) \\ \rho \boldsymbol{v} \times (\boldsymbol{v} - \boldsymbol{u}_\Omega) + p\bar{\boldsymbol{I}} \\ \rho E(\boldsymbol{v} - \boldsymbol{u}_\Omega) + p\boldsymbol{v} \end{array} \right\} \qquad \boldsymbol{F}^v = \left\{ \begin{array}{c} \cdot \\ \mu\bar{\boldsymbol{\tau}} \\ \mu\bar{\boldsymbol{\tau}} \cdot \boldsymbol{v} + \kappa \nabla T \end{array} \right\} \tag{3.3}$$

In the fluxes the pressure is defined as p, static temperature as T, thermal conductivity as κ, dynamic viscosity as μ and finally the viscous stress tensor as $\bar{\boldsymbol{\tau}}$ which can be expressed as:

$$\bar{\boldsymbol{\tau}} = \nabla \boldsymbol{v} + \nabla \boldsymbol{v}^T - \frac{2}{3}\bar{\boldsymbol{I}}(\nabla \cdot \boldsymbol{v}) \tag{3.4}$$

The turbulence modelling is defined according to the Boussinesq hypothesis by defining both the laminar and turbulent parts. The dynamic viscosity for this problem looks like: $\mu = \mu_l + \mu_t$ and the thermal conductivity like: $\kappa = \kappa_l + \kappa_t$.

The discretisation of the Navier-Stokes equation as given in Equation 3.1 using an implicit Euler scheme leads to the following equation:

$$\mathscr{D}_t(\boldsymbol{U}^{q+1}) + \mathscr{R}\boldsymbol{U}^{q+1} = 0 \tag{3.5}$$

In the implicit Euler scheme, the physical time step index is denoted as q and the time derivative operator as $\mathscr{D}_t$. This equation can also be used in a dual time stepping approach for relaxation purposes of the solution at each time step. Using a pseudo time τ, the discrete Navier-Stokes equation would look like this:

$$\Omega\frac{\Delta \boldsymbol{U}^{q+1}}{\Delta\tau} + \Omega\mathscr{D}_t(\boldsymbol{U}^{q+1}) + \mathscr{R}\boldsymbol{U}^{q+1} = 0 \tag{3.6}$$

3.2.2. Harmonic Balance Operator

Discrete Fourier transformation is used to get the Fourier coefficients for each time instance. The coefficients are expressed using the following equation.

$$\hat{\boldsymbol{u}}_k = \frac{1}{N}\sum_{n=0}^{N-1}\boldsymbol{U}_n e^{-i\omega_k t_n} \tag{3.7}$$

For a harmonic balance problem, the number of time instances that are evaluated is the positive and negative of each frequency plus the zero frequency. This means that the number of time instances $N = 2K+1$ where K represents the number of frequencies and $\omega_k = 2\pi f_k$. The conservative variable vector $\tilde{\boldsymbol{U}}$ is evaluated at each time instance in time instance vector $\boldsymbol{t}$. The corresponding Fourier coefficients then can be expressed in its own vector $\hat{\boldsymbol{u}}$. These vectors are all presented in Equation 3.8.

$$\begin{gathered}\tilde{\boldsymbol{U}} = [\boldsymbol{U}_0, \boldsymbol{U}_1, ..., \boldsymbol{U}_{N-1}] \\ \boldsymbol{t} = [t_0, t_1, ..., t_{N-1}] \\ \hat{\boldsymbol{u}} = [\hat{\boldsymbol{u}}_0, \hat{\boldsymbol{u}}_1, ..., \hat{\boldsymbol{u}}_{K-1}]\end{gathered} \tag{3.8}$$

To prevent numerical instabilities, an odd number of frequencies, therefore the $0Hz$ frequency is taken into account, will be used for this study [26]. The number of frequencies K can be expressed as $\omega = [0, \omega_1, ..., \omega_K, \omega_{-K}, \omega_{-1}]$. The discrete Fourier transform (DFT) matrix is defined as:

$$E_{k,n} = \frac{1}{N}e^{-i\omega_k t_n} \qquad n,k \in [0,N] \tag{3.9}$$

Where the inverse of this matrix (IDFT) is equal to:

$$E^{-1}_{k,n} = e^{i\omega_k t_n} \qquad n,k \in [0,N] \tag{3.10}$$

The Fourier coefficient can be calculated using the following equation:

$$\hat{\boldsymbol{u}} = \boldsymbol{E}\tilde{\boldsymbol{U}} \tag{3.11}$$

And conversely, the conservative variables can be calculated using the following equation:

$$\tilde{\boldsymbol{U}} = \boldsymbol{E}^{-1}\hat{\boldsymbol{u}} \tag{3.12}$$

In case the frequencies are not integer multiples of f_1 and the ordinary or inverse discrete Fourier transform matrix are constructed, the analytical expression for the corresponding inverse matrix is impossible to obtain. The IDFT is possible to be computed using numerical matrix inversion methods.
Using a spectral operator for each of the evaluated time instances of the conservative variables $\tilde{\boldsymbol{U}}$, the time operator of the discrete Navier-Stokes equation, given in Equation 3.5, can be approximated using spectral interpolation. This will result in the following equation:

$$\mathscr{D}_t(\boldsymbol{U}) \approx \mathscr{D}_t(\tilde{\boldsymbol{U}}) \tag{3.13}$$

With $\hat{\boldsymbol{u}}$ being independent of time, the time operator can be rewritten using Equation 3.11 and Equation 3.12 into:

$$\mathscr{D}_t(\tilde{\boldsymbol{U}}) = \mathscr{D}_t(\boldsymbol{E}^{-1}\hat{\boldsymbol{u}}) = \frac{\partial \boldsymbol{E}^{-1}}{\partial t}\hat{\boldsymbol{u}} = \frac{\partial \boldsymbol{E}^{-1}}{\partial t}\boldsymbol{E}\tilde{\boldsymbol{U}} \tag{3.14}$$

Equation 3.10 is used for the partial derivative of the IDFT with respect to time and can be written,

$$\frac{\partial \boldsymbol{E}^{-1}}{\partial t} = \boldsymbol{E}^{-1}\boldsymbol{D} \tag{3.15}$$

leading to a diagonal matrix with the following entries.

$$D_{k,n} = i\omega_k \delta_{k,n} \tag{3.16}$$

Alternatively, the diagonal matrix can be given as:

$$\boldsymbol{D} = diag(0, i\omega_1, ..., i\omega_K, i\omega_{-K}, ..., i\omega_{-1}) \tag{3.17}$$

The spectral operator matrix H can be computed by combining Equation 3.14 and Equation 3.15 is presented below.

$$\boldsymbol{H} = \boldsymbol{E}^{-1}\boldsymbol{D}\boldsymbol{E} \tag{3.18}$$

In the spectral operator matrix $\boldsymbol{E}^{-1}$ is given analytically by Equation 3.10, with $\boldsymbol{E}$ being computed by means of inverting $\boldsymbol{E}^{-1}$ using Gaussian elimination. From this, the approximated time operator can be expressed using the spectral operator matrix and the conservative variables as presented below.

$$\mathscr{D}_t(\tilde{\boldsymbol{U}}) = \boldsymbol{H}\tilde{\boldsymbol{U}} \tag{3.19}$$

3.2.3. Harmonic balance in time-domain

In order to make an accurate prediction for unsteady aerodynamics, the harmonic balance method has to be converted back to the time domain. This is explained in the following subsection. Given that $\tilde{\boldsymbol{U}}$ is the vector containing the conservative variables evaluated at N time instances, the dual time stepping discrete Navier-Stokes equation given in Equation 3.6 can be written for each time instance as:

$$\Omega\frac{\Delta \boldsymbol{U}_n^{q+1}}{\Delta\tau} + \Omega\mathscr{D}_t(\boldsymbol{U}_n^{q+1}) + \mathscr{R}\boldsymbol{U}_n^{q+1} = 0 \tag{3.20}$$

Linearising the residual leads to the following:

$$\mathscr{R}(\boldsymbol{U}_n^{q+1}) = \mathscr{R}(\boldsymbol{U}_n^{q}) + \frac{\partial\mathscr{R}(\boldsymbol{U}_n^{q})}{\partial\boldsymbol{U}_n^{q}}\Delta\boldsymbol{U}_n = \mathscr{R}(\boldsymbol{U}_n^{q}) + \boldsymbol{J}\Delta\boldsymbol{U}_n \tag{3.21}$$

Equation 3.19 is possible to be rewritten due to $\mathscr{D}_t$ being a linear operator.

$$\mathscr{D}_t(\boldsymbol{U}_n) = \mathscr{D}_t(\boldsymbol{U}_n^{q+1} - \boldsymbol{U}_n^{q}) = \mathscr{D}_t(\boldsymbol{U}_n^{q+1}) - \mathscr{D}_t(\boldsymbol{U}_n^{q}) \tag{3.22}$$

From Equation 3.20, the second term can be manipulated into the following:

$$\mathscr{D}_t(\boldsymbol{U}_n^{q+1}) = \sum_{k=0}^{N-1} H_{n,k}\Delta\boldsymbol{U}_k + \sum_{k=0}^{N-1} H_{n,k}\boldsymbol{U}_k^{q} \tag{3.23}$$

Now all terms of Equation 3.20 have a linearised expression and therefore are transformed into:

$$\left(\frac{\Omega\boldsymbol{I}}{\Delta\tau} + \boldsymbol{J} + \Omega H_{n,n}\right)\Delta\boldsymbol{U}_n + \mathscr{R}(\boldsymbol{U}_n^{q}) = -\sum_{k=0}^{N-1}(1-\delta_{n,k})H_{n,k}\Delta\boldsymbol{U}_k - \sum_{k=0}^{N-1} H_{n,k}\boldsymbol{U}_k^{q} \tag{3.24}$$

Using the semi-implicit approach to solve for Equation 3.24:

$$\left(\frac{\Omega\boldsymbol{I}}{\Delta\tau} + \boldsymbol{J}\right)\Delta\boldsymbol{U}_n = \tilde{\mathscr{R}}_n(\hat{\boldsymbol{U}}^{q}) \tag{3.25}$$

With the residual expressed as:

$$\tilde{\mathscr{R}}_n(\hat{\boldsymbol{U}}^{q}) = -\mathscr{R}(\boldsymbol{U}_n^{q}) - \sum_{k=0}^{N-1} H_{n,k}\Delta\boldsymbol{U}_k - \sum_{k=0}^{N-1} H_{n,k}\boldsymbol{U}_k^{q} \tag{3.26}$$

For each time instance, Equation 3.25 is solved in a segregated manner. This means that an unsteady inflow problem can be solved by $2K+1$ nonlinear systems of equations where the flow is characterised by K frequencies. In the method of Rubino et al. [45], and thus this study, the harmonic balance method approach presented above is implemented in the SU2 code [21].

For the harmonic balance solver to represent any generic quantities over a larger time vector t^* of length N^* from the original N time instances Equation 3.27 can be used. In this equation $\boldsymbol{E}^{*-1}$ represents the interpolated IDFT matrix now having size $N^* \times N$. This matrix is created using $E^{*-1}_{n,k} = e^{i\omega_k t^*_n}$

$$\Gamma^* = \boldsymbol{E}^{*-1}(\boldsymbol{E}\Gamma) \tag{3.27}$$

3.3. Adjoint method

Design optimisation problems generally rely on stochastic methods or gradient-based methods. An example of the stochastic method is genetic algorithms and simulated annealing [55]. Stochastic methods also compute a gradient and are able to find the global minimum of an objective function, while they are incredibly time-consuming. Gradient-based optimisations, on the other hand, are a better option for routine design optimisation problems when the efficiency of computation is important.

In the gradient-based methods, one method to compute the gradients is a finite differences method. The finite differences method resolves the full-flow solution for each design parameter once. So for a problem with a large number of degrees of freedom, this would be computationally very expensive. The adjoint method, on the other hand, only resolves the direct flow equations once and the adjoint flow equations once for any N number of design variables. Both the direct flow and adjoint flow equations of the adjoint method have similar computational complexity. Therefore, the time necessary for computation is similar, leading to a substantial benefit for the adjoint method over the finite differences method.

3.3.1. Discrete vs continuous adjoint

Continuous adjoint methods have been developed using Lagrange multipliers, also called adjoint variables, combining the variation of a cost function with field equations with respect to design and flow field variables. The collection of terms associated with the variation of flow field variables leads to the adjoint equation and its corresponding boundary conditions. The gradient is produced by the variation in terms associated with design variables. The continuous adjoint approach also has a discrete nature to it since the field, and adjoint equations with the corresponding boundary condition need to be discretised in order to obtain numerical solutions. With a more refined mesh, the continuous adjoint approach yields a more exact gradient. The advantage of the continuous adjoint approach is that the linearised PDEs use the same numerical iteration techniques to get to their solution.

Discrete adjoint methods rely on the application of control theory directly to a set of discrete field equations. The derivation of the discrete adjoint equations is done using the collection of terms and multiplying them by the discrete flow variable variation. When the discrete equation is solved in an exact manner, the result of the Lagrange multiplier is an exact gradient for an inexact cost function. This leads to the derivatives of the discrete adjoint equation being consistent with finite differences gradients with respect to an objective $\boldsymbol{\alpha}$ irrespective of mesh size [42]. One of the main disadvantages of a discrete adjoint approach is higher computational time and memory overhead due to computing and storing the exact Jacobian for each time instance and iteration.

In the past, mainly the continuous adjoint approach has been implemented as an initial step towards adjoint design optimisation problems, as did Jameson et al. [34] and SU2 [21]. However, with much more computing power readily available nowadays, the discrete adjoint approach is much more appealing because mesh issues are less of a problem, leading to the possibility of dynamic meshes. Therefore Rubino et al. [45] used the discrete adjoint method in their work, creating an unsteady design optimisation method using the harmonic balance method. In order to use the adjoint method in combination with the harmonic balance method, slight alterations have to be made to the adjoint equations. In the following section, the adjoint equations derived for the harmonic balance solver are presented, coming from the method created by Rubino et al. [45].

3.3.2. Adjoint governing equations

Using Equation 3.25 the fixed point iteration reformulation can be made for each $\boldsymbol{U}_n$ as follows:

$$\boldsymbol{U}_n^{q+1} = \mathscr{G}_n(\boldsymbol{U}_n) \tag{3.28}$$

In this equation, $\mathscr{G}_n$ for each time instance n is considered the iteration operator for the pseudo time stepping method. If $\left\|\frac{\partial \mathscr{G}_n}{\partial \boldsymbol{U}_n}\right\| < 1$, so $\mathscr{G}_n$ is contractive, a unique fixed point solution $\boldsymbol{U}_n^*$ can be found, according to the Banach fixed point theorem [9], such that:

$$\widehat{\mathscr{R}}_n(\boldsymbol{U}^*) = 0 \quad \Longleftrightarrow \quad \boldsymbol{U}_n^* \mathscr{G}_n(\boldsymbol{U}^*) \tag{3.29}$$

In order to use the adjoint method for a design problem, an objective function has to be stated in the form of $\mathscr{J}$ with a vector design variable stated as $\boldsymbol{\alpha}$. The adjoint objective can be expressed as:

$$\begin{aligned} &\text{Minimise for } \boldsymbol{\alpha} \quad \mathscr{J}(\boldsymbol{U}(\boldsymbol{\alpha}), \boldsymbol{X}(\boldsymbol{\alpha})) \\ &\text{Subject to} \quad \boldsymbol{U}_n(\boldsymbol{\alpha}) = \mathscr{G}_n(\boldsymbol{U}(\boldsymbol{\alpha}), \boldsymbol{X}_n(\boldsymbol{\alpha})), \quad n = 1,2,...N \\ &\qquad\qquad\quad \boldsymbol{X}_n(\boldsymbol{\alpha}) = M_n(\boldsymbol{\alpha}) \end{aligned} \tag{3.30}$$

Where $M_n(\alpha)$ represents the differentiable function connected to the mesh deformation algorithm. $\boldsymbol{X}_n$ represents the physical grids constructed for each individual time instance. The spectral average is taken of the objective function using Equation 3.27 for each time instance. The spectral average is able to represent any quantity of interest over a larger time vector than the original N time instances used in the HB solver.

Leading to:

$$\mathscr{J} = f(\mathscr{J}(\boldsymbol{U}_1, \boldsymbol{X}_1), \mathscr{J}(\boldsymbol{U}_2, \boldsymbol{X}_2), ..., \mathscr{J}(\boldsymbol{U}_N, \boldsymbol{X}_N)) \tag{3.31}$$

The design optimisation Lagrangian with adjoint variables μ and λ can be expressed as:

$$\mathscr{L} = \mathscr{J} + \sum_{n=1}^{N} \left\{ (\mathscr{G}_n(\boldsymbol{U}(\boldsymbol{\alpha}), \boldsymbol{X}_n(\boldsymbol{\alpha})) - \boldsymbol{U}_n(\boldsymbol{\alpha}))^\top \boldsymbol{\lambda}_n + (M_n(\boldsymbol{\alpha}) - \boldsymbol{X}_n(\boldsymbol{\alpha}))^\top \boldsymbol{\mu_n} \right\} \tag{3.32}$$

With constraint equations equal to:

$$\begin{aligned} &\boldsymbol{U}_n(\boldsymbol{\alpha}) - \mathscr{G}_n(\boldsymbol{U}(\boldsymbol{\alpha}), \boldsymbol{X}_n(\boldsymbol{\alpha})) = 0, \quad n = 0,1,...,N-1 \\ &\boldsymbol{X}_n(\boldsymbol{\alpha}) - M_n(\boldsymbol{\alpha}) = 0 \end{aligned} \tag{3.33}$$

The differential of the Lagrangian equation can be expressed in order to circumvent the explicit notation of the independent variables:

$$\begin{aligned} d\mathscr{L} = &\sum_{n=0}^{N-1} \left(\frac{\partial \mathscr{J}^\top}{\partial \boldsymbol{U}_n} + \sum_{k=0}^{N-1} \frac{\partial \mathscr{G}_k^\top}{\partial \boldsymbol{U}_n} \boldsymbol{\lambda}_k - \boldsymbol{\lambda}_n \right) d\boldsymbol{U}_n + \\ &\sum_{n=0}^{N-1} \left(\frac{\partial \mathscr{J}^\top}{\partial \boldsymbol{X}_n} + \frac{\partial \mathscr{G}_n^\top}{\partial \boldsymbol{X}_n} \boldsymbol{\lambda}_n - \boldsymbol{\mu}_n \right) d\boldsymbol{X}_n + \\ &\sum_{n=0}^{N-1} \frac{\partial M_n^\top}{\partial \boldsymbol{\alpha}} \boldsymbol{\mu}_n d\boldsymbol{\alpha} \end{aligned} \tag{3.34}$$

This leads to the adjoint equations being expressed as:

$$\frac{\partial \mathscr{J}^\top}{\partial \boldsymbol{U}_n} + \sum_{k=0}^{N-1} \frac{\partial \mathscr{G}_k^\top}{\partial \boldsymbol{U}_n} \boldsymbol{\lambda}_k = \boldsymbol{\lambda}_n \tag{3.35}$$

and

$$\frac{\partial \mathscr{J}^\top}{\partial \boldsymbol{X}_n} + \frac{\partial \mathscr{G}_n^\top}{\partial \boldsymbol{X}_n} \boldsymbol{\lambda}_n = \boldsymbol{\mu}_n \tag{3.36}$$

The order of solving the adjoint equations is as follows; first, the solution of Equation 3.35 is required to solve for Equation 3.36. As seen for the flow solver Equation 3.35 can be solved as a fixed point iteration in $\boldsymbol{\lambda}_n$ with $\boldsymbol{U}_n^*$ as the numerical solution of the flow equation Equation 3.29

$$\lambda_n^{q+1} = \frac{\partial \mathscr{N}}{\partial \boldsymbol{U}_n}(\boldsymbol{U}_n^*, \boldsymbol{\lambda}^q, \boldsymbol{X}_n) \tag{3.37}$$

Leading to a shifted Lagrangian $\mathscr{N}$ expressed as:

$$\mathscr{N} = \mathscr{J} + \sum_{n=1}^{N} \mathscr{G}_n^\top (\boldsymbol{U}, \boldsymbol{X}_n) \boldsymbol{\lambda}_n \tag{3.38}$$

And because $\mathscr{G}_n$ is contractive, it can be seen from Equation 3.39 that $\frac{\partial \mathscr{N}}{\partial \boldsymbol{U}_n}$ is also contractive.

$$\left\| \frac{\partial}{\partial \lambda_n} \left(\frac{\partial \mathscr{N}}{\partial \hat{U}_n} \right) \right\| = \left\| \frac{\partial \mathscr{G}_n^\top}{\partial \hat{U}_n} \right\| = \left\| \frac{\partial \mathscr{G}_n}{\partial \hat{U}_n} \right\| < 1 \tag{3.39}$$

This means that according to the fixed point theorem used for Equation 3.37, the convergence rate of Equation 3.37 is the same as the primal flow solver. The $\frac{\partial \mathscr{N}}{\partial \boldsymbol{U}_n}(\boldsymbol{U}_n^*, \boldsymbol{\lambda}^q, \boldsymbol{X}_n)$ term of this equation has been obtained by applying Algorithmic Differentiation to the source code which computes $\mathscr{G}_n$. The AD tool originates from a CoDipack [46] and uses a Jacobi taping method combined with the Expression Templates feature of C++. This all leads to only a small increase in computational time which is good for the efficiency purposes of this model. The gradients of the objective function $\mathscr{J}$ with respect to each design variable $\boldsymbol{\alpha}$ are then determined using the converged flow and adjoint solution. The gradients are computed using the following relation

$$\frac{\mathrm{d}\mathscr{L}^\top}{\mathrm{d}\boldsymbol{\alpha}} = \frac{\mathrm{d}\mathscr{J}^\top}{\mathrm{d}\boldsymbol{\alpha}} = \frac{\partial M_n^\top(\boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}} \mu_n \tag{3.40}$$

4

Real Propeller Inflow Conditions

The optimisation method tailored for unsteady periodic inflow conditions set as the end product of this research needs some form of unsteady inflow conditions. Using the APPU project as a test case, the real inflow conditions at the propeller disk have to be determined. The first part of this process is the data handling and preparation, which is presented in section 4.1. Secondly, the inflow is recreated using a Fourier series expansion which is presented in section 4.2.

4.1. Inflow data handling

Due to the positioning of the propeller on the APPU project, the unsteady inflow has to be expected at the propeller disk. This unsteadiness exists due to the boundary layer being formed around the fuselage. The propeller is positioned behind the vertical tail stabiliser, which also produces a wake at the propeller disk. The boundary layer can cause the airflow to become unsteady, and therefore appropriate inflow conditions need to be set for the simulations. The inflow conditions in 3D can be considered unsteady and non-uniform due to the boundary layer formation. As seen in Figure 4.1, the full propeller blade experiences the non-uniformity created by the boundary layer. Only when the 2D airfoil is taken the non-uniformity disappears, and the inflow becomes uniform again assuming that there is no circumferential variation of inflow. However, in the case of the APPU project there is a circumferential variation of inflow conditions. Therefore, the inflow of the optimisation method is to be modelled as an unsteady uniform inflow.

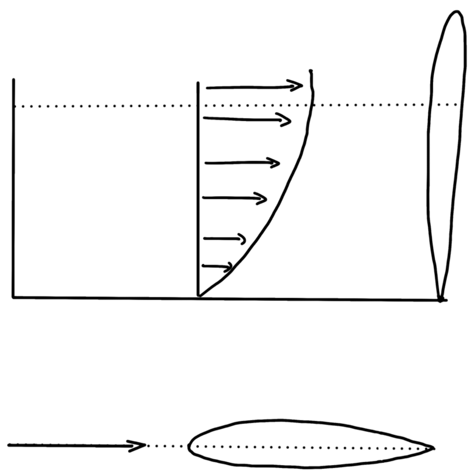


Figure 4.1: Assuming a radial uniform approach for 2D airfoil

Component velocity data acquired by means of a RANS simulation of the full fuselage has been supplied for the region close to the propeller disk. This data comes from another team within the APPU project working group. They have positioned an actuator disk at the location of the propeller, as can be seen in Figure 4.2.

The received data represents the (x,y,z) component velocity of a fan-on case at 10 cm in front of the propeller disk. This means that the component velocity experiences the effects of both the unsteadiness due to the fuselage boundary layer and the acceleration due to the fan on the case.

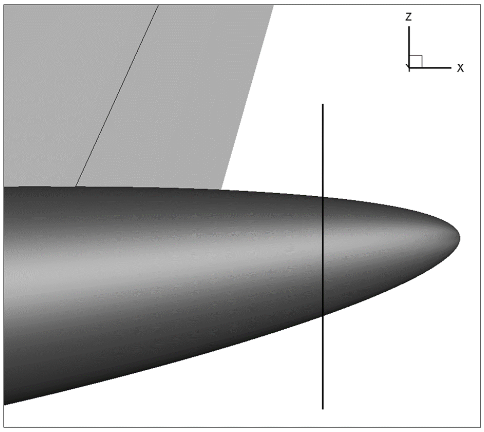


Figure 4.2: Render of the RANS model used to determine the inflow conditions of the APPU project

The actuator disk theory has been discussed in section 2.2 and has already been used in multiple conceptual studies [5, 10, 39]. In the case of the APPU project, they have used the actuator disk theory in order to predict how the flow around the fuselage near the propeller disk behaves.

In this work, the inflow characteristics are used in order to create an airfoil which is able to cope with such inflow unsteadiness. The velocity change that the airfoil experiences is dependent on the axial and tangential velocity. Since the axial velocity is not changed in the current research, the change in velocity that the airfoil experiences is solely dependent on the tangential velocity. Spanwise flow patterns, or radial velocity changes, are not taken into account for this design problem. In this work, no 3D effects are taken into account. Although the 3D effects do influence the inflow conditions of the 2D approximation, the effects are assumed to be negligible for the purpose of this research. Marinus et al. [41] have found that for a full propeller blade, the 3D effects are mainly important to the acoustic prediction of the airfoil.

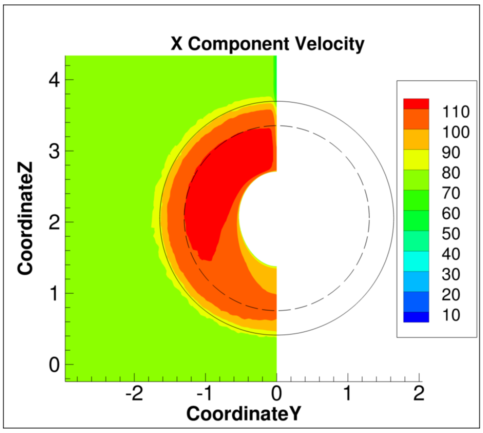
The cruise Mach number of the proposed aircraft A320neo is $M = 0.78$. With the knowledge that the propeller tip Mach number is best to be kept below the speed of sound, and the radius of the propeller is set to $1.643m$. It can be determined that the propeller rotational speed should not exceed 1000 rpm. At altitude, the cruise speed is equal to $V_\infty = 230m/s$. Assuming that there is no change in velocity due to the disturbance created by the boundary layer and the apparent velocity V_{app} being the combined rotational velocity and axial velocity. The speed of sound at cruise altitude of $h = 35000ft$ is equal to $a_{cruise} = 295m/s$:

$$V_{app} = \sqrt{V_\infty^2 + V_{rot}^2} = \sqrt{V_\infty^2 + (\omega r)^2} = \sqrt{230^2 + \left(\frac{2\pi \cdot 1000}{60} \cdot 1.643\right)} = 287.2 \tag{4.1}$$

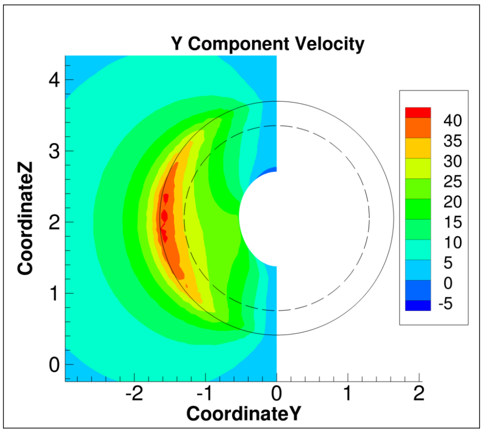
Variable	Value
Radius r **[m]**	1.643
Centre Coordinate [m,m,m]	(36, 0, 2.055)
Rotational Speed ω **[rpm]**	1000
Free stream velocity V_∞ **[m/s]**	79.8
Free stream static temperature T_s **[K]**	300
Free stream Mach number M **[-]**	0.23

Table 4.1: Inflow simulation parameters

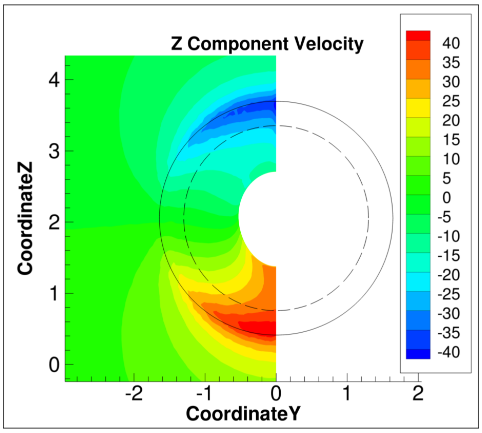
This leads to $M_{tip} = 0.97$, just below the speed of sound. One can argue that this is still too high for the propeller to function efficiently. This creates a strong normal shock over the airfoil and substantially increases



(a) X component velocity



(b) Y component velocity



(c) Z component velocity

Figure 4.3: Velocity profile 10 cm ahead of the propeller disk

the drag. Even at a point that lies at 70% of the span of the propeller blade, a normal shock will be present over the airfoil due to the local inflow Mach number being $M = 0.88$. This leads to the option to lower the rotational speed of the propeller or lower the cruise Mach number. For this reason, the choice has been made in this research to lower the cruise speed by 200 m/s. In this way, the inflow Mach number used for the Euler solver in this research will be equal to $M = 0.796$. The 70% span location is chosen in a semi-arbitrary way combined with engineering common sense. The location will not be influenced by the possible tip vortices created by the propeller or alternative effects present by the spinning hub. The centre coordinate of the propeller can be found in Table 4.1 as well as the rest of the inflow simulation parameters. The rotational speed is not used specifically for the inflow simulation, only for the determination of the frequency of the inflow. In Figure 4.3, the dashed line represents the to-be-examined point on the propeller disk.

The component velocity data grid is expressed in the Cartesian coordinate system, while the propeller inflow conditions need to be in the polar coordinate system. So a transformation is applied using Equation 4.2 and Equation 4.3. The centre of rotation is located at 2.055 meters from the origin with respect to the z-axis. Therefore, this has been subtracted from the z coordinates, as can be seen in Equation 4.2.

For the propeller blade, rotation angle ψ counterclockwise has been assumed positive. Therefore, the y coordinates have been changed sign in order for ψ to remain positive over the full rotation. Furthermore, when $z < 2.055$, an additional π had to be added to the angle in order for it to have the right sign.

$$r = \sqrt{y^2 + (z - 2.055)^2} \tag{4.2}$$

$$\begin{aligned} \psi &= \tan\frac{-y}{z-2.055} & z \geq 2.055 \\ \psi &= \tan\frac{-y}{z-2.055} + \pi & z < 2.055 \end{aligned} \tag{4.3}$$

Again this work is not part of this research in itself, the data was supplied by the APPU team as an input for this research. As can be seen in Figure 4.4, the grid nodes are not equally divided over the entire propeller disk. The mesh is substantially finer in the parts close to the vertical tailplane and close to the hub of the propeller. This means a higher resolution of data is expected close to the vertical tailplane and the hub. Away from the hub, the data is relatively sparsely distributed over the propeller disk. This would lead to a larger inaccuracy in terms of inflow conditions. Two options can be considered to solve this issue: extrapolating the data between the missing points on the proposed distance from the hub or taking a wider range of points around the proposed distance from the hub. Solely using the points at a distance of 1.15 m from the centre of rotation would yield no data points. Taking the points lying on ± 1% of the proposed radius would yield a signal which would be unusable to accurately represent the unsteady nature of the inflow. Table 4.2 displays the number of elements in their respective range and segment of the revolution in degrees. So in order to extrapolate the data, a range of ± 1% already has to be used. Furthermore, the resolution of the data is approximately 10 degrees per element in the 10-180 degrees section of the disk. Interpolating that data would give a very linear behaviour over these elements. Therefore, it is opted to go for widening the range of the proposed radius. Taking the values of the surrounding elements in this case to augment the data fidelity rather than extrapolating. The chosen range to consider is ± 3% around the 1.15 m aim, meaning that all data points that lie between $1.1155 < r < 1.1845$ are considered to be included in the inflow signal.

	Number of elements 0-180 degrees	**Number of elements 0 - 10 degrees**	**Number of elements 10 - 180 degrees**
1% range	117	79	18
3% range	349	237	112
5% range	616	402	214

Table 4.2: Data point distribution for various spreads around 1.15 m radius

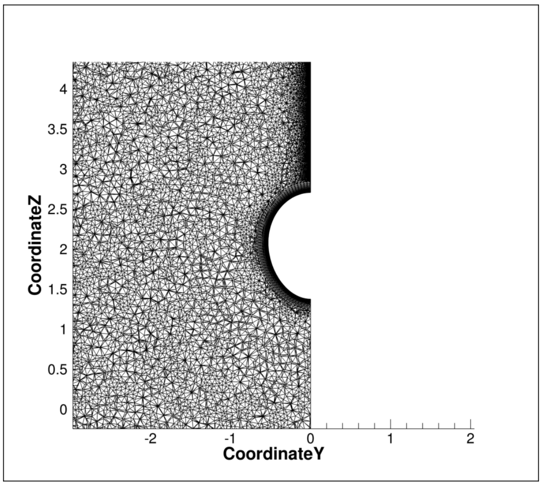


Figure 4.4: Mesh 10cm upstream of propeller disk for APPU project simulation

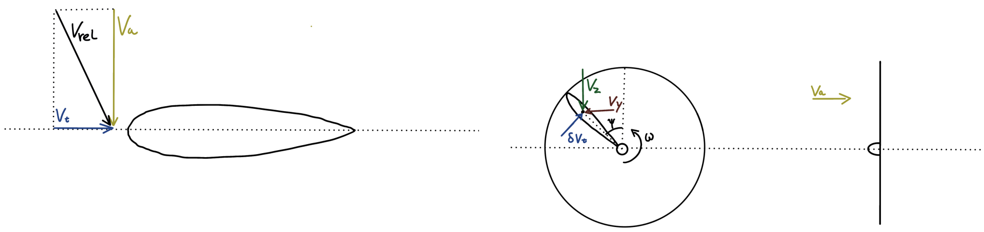
For simplicity, for the remainder of this paper, the tangential velocity change experienced by the propeller airfoil section is referred to as the velocity normal to the airfoil or the change in tangential velocity, δv_t. The change in tangential velocity changes over one full revolution due to the change in v_y and v_z as can be seen in Figure 4.3. Therefore the change in tangential velocity can be defined as

$$\delta v_t = v_y \cos\psi + v_z \sin\psi \tag{4.4}$$

The relative velocity, v_{rel}, which the airfoil of the propeller sees, consists of a number of components. First, the rotational velocity of the propeller blade, so $v_{rot} = \omega r$. Secondly, the change in normal velocity is due to the unsteady inflow conditions experienced by the propeller disk, or δv_t. These two components together can be combined into the tangential velocity of the propeller blade or v_t. Finally, the axial velocity that the propeller disk experiences due to the cruise speed of the aircraft minus any boundary layer effects is expressed as v_a. An overview of the used velocity orientations is given in Figure 4.5.

$$v_t = \delta v_t + \omega r \tag{4.5}$$

$$v_{rel} = \sqrt{v_t^2 + v_a^2} \tag{4.6}$$



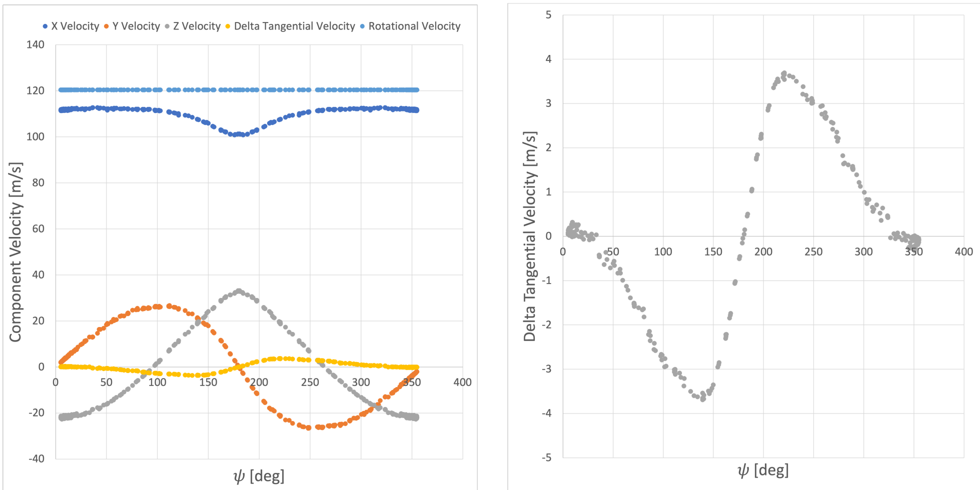
(a) Airfoil section view velocities

(b) Propeller normal view velocities

Figure 4.5: Schematic view of the velocity orientations used for this research

Due to the symmetry of the aircraft, only half of the propeller disk has been simulated. Therefore to get the full rotation for the airfoil inflow, the inflow component signals have been mirrored. The resulting component velocities can be seen in Figure 4.6. As seen in Table 4.1, the freestream Mach number is $M = 0.23$. Significantly lower than the cruise Mach number of the airbus or the proposed simulation Mach number. This will influence the results of the proposed method. The aim of the research is to create a method which is able to optimise the shape of the airfoil efficiently in an off-design flight condition. Therefore, the inflow conditions do not have to match the real conditions exactly as long as the same tendencies are there. With the current data, this is the case, such that the key behaviours of the inflow conditions are captured in the inflow. Increasing the blade loading for a given advance ratio would yield a greater velocity difference created over the propeller disk. This would also lead to a slight increase in the velocities directly in front of the propeller disk. Consequently, the amplitude of the change in tangential velocity as given in Figure 4.6b would be higher.

In the CFD simulation, the change in the angle of attack needs to be known as input to the problem. Therefore some extra steps are needed. Both the axial and tangential velocities need to be taken into account for the angle of attack. There is a change in not only the tangential velocity but also the axial velocity, as can be seen in Figure 4.3a. The angle of attack experienced by the airfoil, assuming that the pitch angle β stays constant, changes due to the change in inflow angle. As can be seen in Figure 2.4 the pitch angle β is an addition of the angle of attack α and the inflow angle ϕ and is expressed below:



(a) All component velocities (b) Change in tangential velocity

Figure 4.6: Component velocity for a Fan-on situation of the APPU project configuration over a full rotation

$$\beta = \phi + \alpha \tag{4.7}$$

In order to know the change in the angle of attack, the change in inflow angle has to be determined. The inflow angle is determined by the axial and tangential velocity which the propeller blade experiences. Therefore using the following equation in vector form, the change in inflow angle is determined over the entire revolution. Where $\bar{\phi}$ is defined as the average of $\tan^{-1}\left(\frac{\boldsymbol{v_x}}{\boldsymbol{v_t}}\right)$. Assuming that the blade pitch remains constant during the simulation, the change in inflow conditions is the opposite sign of the change in angle of attack, or $d\alpha = -d\phi$. The change in angle of attack over a full revolution in degrees is pictured in Figure 4.7. However, during the simulation with SU2, to achieve the unsteady inflow, the airfoil pitch will be varied over time. Therefore in order to get the angle of attack during the simulation as shown in Figure 4.7, the change in inflow angle will be used as the pitch angle of the airfoil.

$$d\boldsymbol{\alpha} = \tan^{-1}\left(\frac{\boldsymbol{v_x}}{\boldsymbol{v_t}}\right) - \bar{\alpha} \tag{4.8}$$

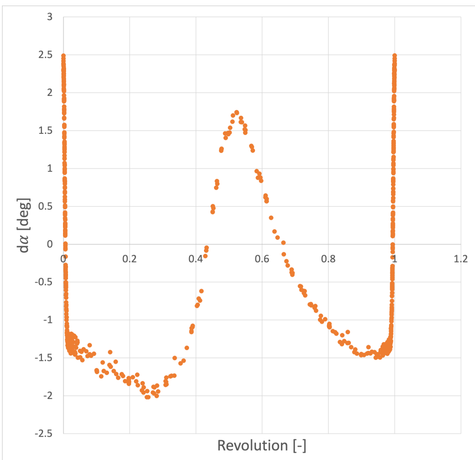


Figure 4.7: The change in angle of attack over a normalised full revolution

4.2. Fourier series expansion

With a Fourier series expansion, the inflow characteristic behaviour can be modelled as the angle of attack for the simulation. The reason for using a Fourier series is that with a distinct number of Fourier terms the inflow can be represented in a very detailed manner. Furthermore, the Harmonic balance method also uses a discrete Fourier transformation to obtain the Fourier coefficients for each individual time instance. The change in inflow angle $d\phi$ will be simulated by SU2 as changing the pitch angle of the airfoil. The Fourier series expansion leads to an approximation of the change of angle of attack by means of sin and cos function. This makes it a periodic function and thus very suitable for a propeller case. The accuracy of the Fourier series expansion is defined by the number of terms included in the series. The Fourier series expansion is defined as:

$$f(x) = \frac{A_0}{2} + \sum_{k=1}^{\infty}\left(A_k \cos\frac{2\pi kx}{L} + B_k \sin\frac{2\pi kx}{L}\right) \tag{4.9}$$

Where A_k and B_k represent the Fourier coefficients, L the period over which the series is expanded, and the integer k represents the number of terms used in the expansion and thus the k^{th} harmonic of the expanded Fourier series. The Fourier coefficients are dependent on the function $f(x)$, which represents the change in the angle of attack in this case. The Fourier coefficients can be expressed as

$$\begin{aligned} A_0 &= \frac{1}{\pi}\int_{\pi}^{-\pi} f(x)dx \\ A_k &= \frac{1}{\pi}\int_{\pi}^{-\pi} f(x)\cos(kx)dx \\ B_k &= \frac{1}{\pi}\int_{\pi}^{-\pi} f(x)\sin(kx)dx \end{aligned} \tag{4.10}$$

Before the number of terms necessary for the Fourier series expansion can be determined, the inflow conditions are simplified. As seen in Figure 4.8, the wake of the tailplane creates a large disturbance in the inflow.

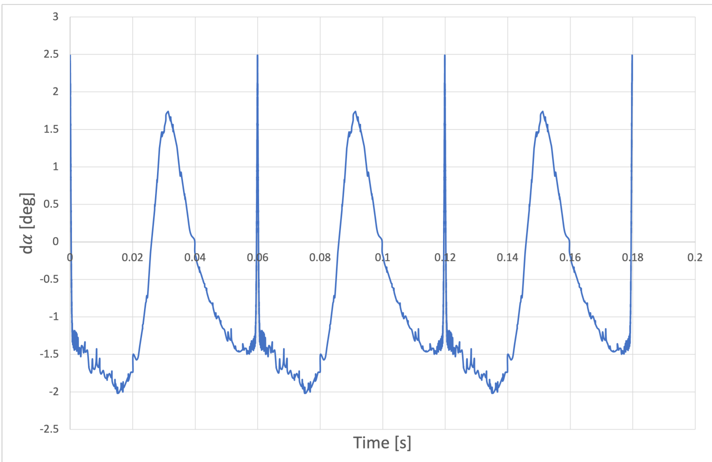


Figure 4.8: 3 periods of change in angle of attack including the vertical tailplane disturbance

According to the data supplied, combined with the assumed rotational speed, the disturbance has a period of fewer than two milliseconds. The harmonic balance solver will not be able to resolve this disturbance due to the short duration with a limited amount of frequencies. Introducing more frequencies means more computing time for the optimiser which makes the computing of the airfoil shape too slow. This leads to the choice of discarding the wake of the tailplane in the real inflow modelling. Some accuracy is therefore lost in the process not only in terms of lift and drag coefficient but also in terms of noise emissions. The periodic wake impingement leads to an impulsive loading of the propeller blade and leads to increased noise emissions [50]. Getting rid of the wake is something that has to be considered thoroughly. The optimisation problem of this research is focused on reducing the drag and forced lift response in unsteady inflow conditions. Therefore, a simplification with respect to the real case can be applied by omitting the vertical tail plane wake. The main goal is to investigate whether it is possible to reduce such quantities with inflow conditions close to the real case. Having a very accurate depiction of this real case is therefore not necessary and therefore the wake can be omitted.

Therefore the choice has been made to exclude the vertical tailplane disturbance from the inflow conditions. The method for removing the wake is by omitting the first and last 5 degrees of azimuthal rotation of the propeller blade from the initial data set provided. The period of the inflow condition is shortened by 10 degrees for this reason. The benefit of removing 5 degrees of azimuthal rotation is that the inflow conditions at the start and end of the new period agree very well, and no interpolation is necessary. Furthermore, to get the true change in the angle of attack of the airfoil, the signal is modified such that the first data point of the inflow conditions lies at the 0-degree mark. Due to the noisy signal around the start of the signal, the first data point seems to lie above the 0-degree mark, but this is not the case.

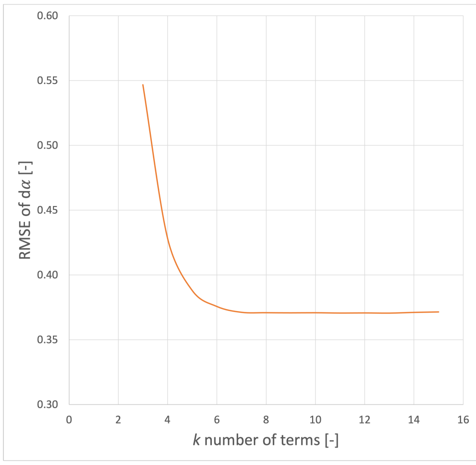


Figure 4.9: Convergence of the RMSE value of the change of angle of attack for varying k number of terms

The amount of detail obtained from high k values is not necessary because this level of definition will not be expressed in the lift and drag coefficient in the simulation. Furthermore, a convergence study using the Root Mean Squared Error (RMSE) has been performed to find the ideal k number of terms. The number of terms have been varied from 3 up to 15 and has shown a stabilising effect around seven terms, as confirmed by Figure 4.9. Therefore the logical choice is to use seven terms in the Fourier series expansion to recreate the $d\alpha$ signal.

The change in angle of attack with its corresponding Fourier series expansion is presented in Figure 4.10. The Fourier coefficients corresponding to this expansion are given in Figure 4.3. The relative importance of the coefficients is displayed in Figure 4.11 and shows that for both A_n and B_n, the first term is the most important. With each term, the importance decreases in importance with the B_n terms showing a steeper decrease in importance. The Fourier coefficients have been generated by a dedicated script which uses the data exported from the inflow conditions simulation. Using the scipy library available in Python Equation 4.10 can be solved for. Within the source code of SU2, the movement of the airfoil can be defined by means of a moving gird. The change in angle of attack for the airfoil can be defined by using Equation 4.9 and programmed into SU2 as a moving airfoil grid.

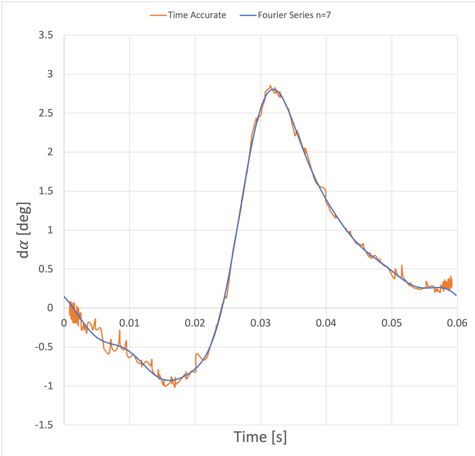


Figure 4.10: Fourier series expansion of the inflow angle for $k = 7$

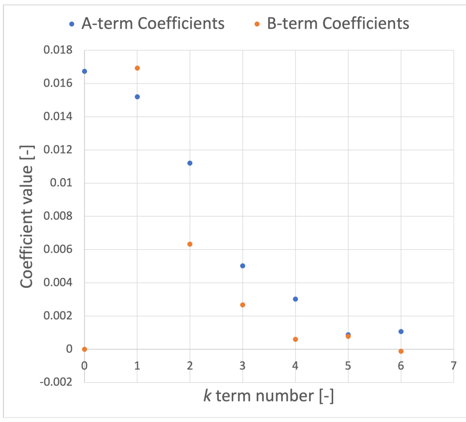


Figure 4.11: Fourier series coefficients per k term

	$\mathbf{A_k}$	$\mathbf{B_k}$
k = 0	0.016729	0
k = 1	0.015204	0.016928
k = 2	0.011203	0.006322
k = 3	0.005015	0.002667
k = 4	0.003024	0.000599
k = 5	0.000868	0.000775
k = 6	0.001067	-0.000124

Table 4.3: Fourier coefficients for the Fourier series expansion with $k = 7$

5

SU2 simulation setup

In this chapter, the SU2 solver setup is discussed. There are many possibilities for setting up a CFD simulation, so only the most important choices will be discussed. In this research, as has been made clear in section 2.5, the Euler solver will be used for the adjoint-based design optimisation using harmonic balancing.

First the changes in the source codoe of SU2 is disscussed in section 5.1. Next, in order to specify the frequencies that were used for the harmonic balance equation, a Fast Fourier transform has been performed on the signal of the change in angle of attack, presented in section 5.2. The general solver parameters will be elaborated upon in section 5.3. Furthermore, the airfoil parameterisation method is discussed in section 5.4. Finally, the grid generation will be presented in section 5.5.

5.1. SU2 source code changes

The starting branch for this research came from the research of Rubino et al. [45]. The branch is located on the GitHub of Antonio Rubino under the name *feature_tHB_AD*[1].

The $C_{l_{RMSE}}$ value is computed within the same C++ function where the time average of the lift and drag coefficient are computed. The already complex code has been kept simple by using one of the already pre-programmed but not used parameter containers. The efficiency parameter is replaced for the $C_{l_{RMSE}}$ value in the SU2 code while still being called efficiency in the code. Adding a separate RMSE parameter to the code would be substantially more difficult and time-consuming than just using one that is not used anyway. In order to get the solver to include this as a possible objective function, the $C_{l_{RMSE}}$ value has to be retrieved in more files. The two adjoint solver files, *solver_adjoint_mean.cpp* and *solver_adjoint_tne2.cpp*, define how the objective functions are interpreted. In these files, the efficiency objective case was changed to represent the $C_{l_{RMSE}}$ value.

Running an optimisation problem using the discrete adjoint method required some more bug fixing. As the result of the simulation for each individual time instance, a separate file is created with a different numerical suffix. So in the case of seven time instances, seven files are created. Normally during the optimisation runs, a function within the Python environment of SU2 handles the renaming of the files. However, this function did not do its job, and the number of time instances needed to be hardcoded into this function. Therefore, in the *tools.py* file for the *expand_zones* function, the number of zones or time instances have been added.

5.2. Harmonic balance frequencies

As input to the harmonic balance solver, a number of frequencies need to be selected for the solver to resolve the unsteady flow. The number of frequencies or time instances solved will be kept relatively low due to the diminishing returns in simulation efficiency. More time instances mean a larger computing time. A Fourier

[1] `https://github.com/arubino/SU2/tree/feature_tHB_AD`

transform is used to get the input signal from the time domain into the frequency domain. When the signal is of periodic nature, the signal will be transformed using a discrete Fourier transform instead of a continuous one. This also leads to the possibility of performing a Fast Fourier Transform (FFT), which exploits the symmetries of the discrete Fourier transform.

The change of inflow angle over a normalised revolution can be seen in Figure 4.7. In order to use an FFT first, the signal has been repeated 100 times in order to get a good resolution for the tonal content. Furthermore, the signal has been put in the right time frame by setting one revolution equal to $0.06s$. This time period corresponds to the rotational velocity of the propeller blade of 1000 RPM. The FFT is expressed as

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-i2\pi kn/N} \tag{5.1}$$

Then by using this symmetry, the following can be calculated:

$$X_{k+N} = \sum_{n=0}^{N-1} x_n \cdot e^{-i2\pi(k+N)n/N} = \sum_{n=0}^{N-1} x_n \cdot e^{-i2\pi n} \cdot e^{-i2\pi kn/N} \tag{5.2}$$

Noting that $e^{-i2\pi n} = 1$, so therefore:

$$X_{k+N} = \sum_{n=0}^{N-1} x_n \cdot e^{-i2\pi kn/N} = X_k \tag{5.3}$$

So this means that:

$$X_{k+1\cdot N} = X_k, \quad \text{for any integer } i \tag{5.4}$$

The FFT computes a Power Spectral Density (PSD) function over all the possible frequencies that can represent the given signal. The frequencies which have a non-zero value in the PSD are the frequencies which are present in the original signal. These non-zero PSD frequencies are also used to determine how many frequencies need to be used in order to accurately represent the unsteady lift and drag forces by the HB solver. The FFT of the signal given in Figure 4.7 is presented in Figure 5.1, and as expected, all frequencies are an integer multiple of the $16.6667Hz$ rotational frequency.

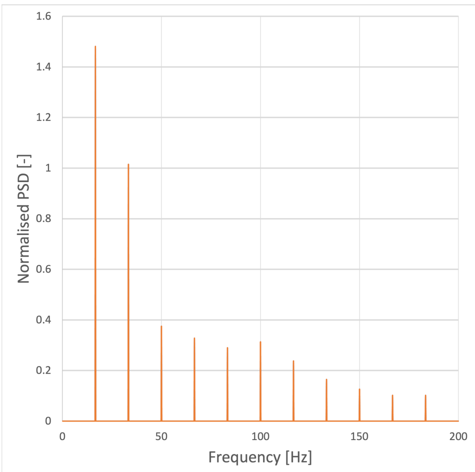


Figure 5.1: Fourier transformed inflow angle signal including the tailplane disturbance

The wake of the vertical tail stabiliser makes for extra instability in the flow and can be seen as more frequencies being present with a higher PSD value in Figure 5.1. As mentioned in section 4.2, the wake of the tailplane is not taken into account for the inflow modelling. Therefore, this disturbance should also be excluded from the determination of the input frequencies for the HB solver. In Figure 5.2, the Fourier transform of the change in inflow conditions without the wake of the tailplane has been presented, and the data from the first and last 5 degrees of the revolution have been omitted.

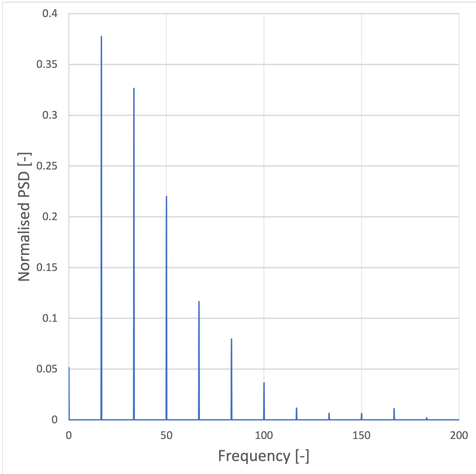
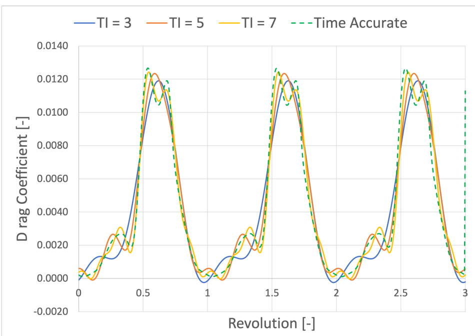


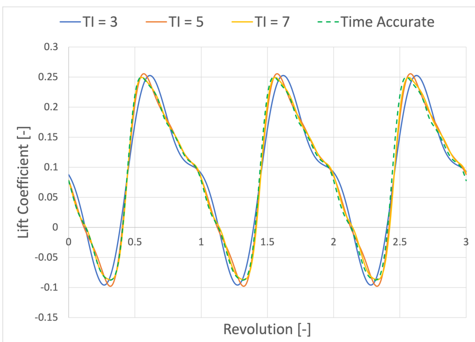
Figure 5.2: Fourier transformed inflow angle signal without tailplane wake

The frequencies over 100 Hz in Figure 5.2 do not play a role in the inflow signal. Therefore, these frequencies do not have to be considered input frequencies for the HB solver. These inputs represent two time instances for the HB solver, the positive and negative frequency. In order to reduce the risk of numerical instability, the $0Hz$ frequency is added to create an odd number of time instances [26]. So the total number of time instances $N = 2K + 1$ where K is the number of frequencies. Rubino et al. [45] have found the number of time instances required for a design optimisation problem with a more complex inflow would need nine time instances, so four frequencies to solve the HB problem. The number of time instances are the number of conservative variables which are solved by the harmonic balance operator. The residuals begin solved in a segregated manner as discussed in chapter 3, and lead to a nonlinear system of N equations needing to be solved. Transforming back to the time domain interpolation is used by means of Equation 3.27.

To the determination of how many time instances are necessary for the optimisation problem in this work, the time-accurate solution of the flow was obtained. This time-accurate solution was obtained using an unsteady Euler simulation with the same inflow characteristics and free stream conditions. The time-accurate solution has been obtained using a 2nd order dual time stepping method with the time step small enough that 160 time steps per period were performed in order to get a well-resolved solution in time.



(a) Drag coefficient



(b) Lift coefficient

Figure 5.3: Drag and lift coefficient per revolution of the propeller blade, obtained from the Fourier series expansion using different numbers of time instances.

Due to the periodicity of the propeller problem and the observation that the resolved frequencies from the FFT are integer multiples of the base frequency, a Fourier series expansion can be used to show the convergence of the harmonic balance solver to the unsteady simulation. In Figure 5.3, the solutions to these multiple Fourier series expansions are presented. The obvious observation is that more time instances mean a better convergence with respect to the time-accurate unsteady simulation.

	RMSE Lift coefficient	**RMSE Drag Coefficient**
TI = 3	0.021665289	0.001266272
TI = 5	0.01089096	0.001017016
TI = 7	0.009622848	0.000759854

Table 5.1: RMSE values for the lift and drag coefficient agreement compared to the time-accurate unsteady solution

Figure 5.3 shows that three time instances are able to define the lift coefficient to a fairly accurate degree. However, improvement can always be found, as seen in Table 5.1, representing the RMSE measures of the approximation compared to the time-accurate signal. Keeping in mind that the number of time instances $N = 2K + 1$ with K being the number of frequencies, one frequency would be sufficient to represent the lift coefficient. However, the drag coefficient does need some extra frequencies to accurately approximate the drag signal. Figure 5.2 shows that there are two dominant frequencies in the inflow signal. The third frequency is less dominant but still a player in the Fourier transformation. This suggests that at least 3 frequencies are required to accurately represent the lift and drag coefficient. The suspicion is confirmed when looking at Table 5.1, where 7 time instances are needed to represent the time history of the drag coefficient. More time instances would yield a more accurate drag coefficient signal but also a higher computational time. This is something to be avoided. The lift coefficient is the most important coefficient for this optimisation and is currently very well represented with only 7 time instances.

The results for the average quantities calculated with a various number of time instances compared to an Euler time accurate simulation has been run. As can be seen in Table 5.2, 5 time instances are not enough to accurately calculate the average drag coefficient. Although the 5 time instance simulation would require substantially less computing time, a 15% deviation from the time-accurate simulation is too much. The accuracy of using 7 time instances is quite good, only 2.5% deviation from the time average simulation, as seen in Table 5.2. Using 9 time instances compared to 7 time instances would introduce a 20% increase in computing time necessary. For the limited accuracy increase of only 0.8% point in average drag coefficient seems a bit of a disproportionate investment. Therefore the choice has been made to go with 7 time instances instead of 9.

	Lift coefficient	**Drag coefficient**	**CD Difference [%]**	**Time [min] on 20 cores**
Time Accurate	0.0861	0.00433	-	111.88
HB 5 TI	0.0648	0.00359	-14.81	32.80
HB 7 TI	0.0866	0.00422	-2.57	52.68
HB 9 TI	0.0864	0.00440	1.71	64.38

Table 5.2: Average quantity comparison between HB solver results and an Euler time accurate simulation

The harmonic balance solver only solves the lift and drag coefficient as output for the number of time instances given as input. This means that for both the calculation of the average lift and drag coefficient and the amplitude of the lift response, the same amount of data points are used as time instances are used in the simulation. The choice is made to run the harmonic balance solver with three input frequencies, so seven time instances originated from the requested slight increase in fidelity for the RMSE value of the lift coefficient. These frequencies are the base frequency, or the rotational frequency of the propeller blade, and the base frequency times 2 and 3. Leading to a total of 7 frequencies in rad/s presented in Equation 5.5

$$[0, 104.72, 209.44, 314.16, -104.72, -209.44, -314.16] \tag{5.5}$$

5.3. Euler-based CFD setup

In Table 5.3, a number of general solver parameters are presented that are used for the Harmonic Balance simulation. The unsteady Euler simulation parameters are given in Table 5.3. The free stream Mach number has been chosen relatively close to the A320neo cruise Mach number. This would mean that the propeller tip Mach number would approach the speed of sound. Therefore, it has been chosen to lower the cruise Mach number of the aircraft such that the apparent Mach number is close to the A320neo cruise Mach number. The cruise velocity of the aircraft was lowered to 200 m/s, which equates to $M_\infty = 0.68$ at cruise altitude. Combining this with the rotational speed of the propeller set at 1000 rpm leads to an apparent Mach number of $M_{app} = 0.796$ at the proposed 70 % span. The CFL number is set at 1.5. Higher CFL numbers can increase the efficiency of the simulation, coming at the cost of accuracy and risk of divergence. The spectral period is set at the time necessary for the propeller blade to complete one full revolution. As for the grid movement parameters, the ω value has been set to the base rotational velocity of the propeller blade. The complete expansion is done within the C++ code as mentioned in section 4.1, where the base frequency is the only frequency used for the Fourier series expansion. The amplitude has been set to equal one since the expanded Fourier series already takes care of the scaling part for the correct change in the angle of attack. In other words, this amplitude represents the number of degrees the grid will move normally. For this study, this is handled by the expansion.

For the unsteady Euler equation, the same free stream conditions have been used for the harmonic balance simulation. As mentioned in the previous section, the time step is set to have 160 steps per period to have a good resolution in time. These 160 time steps are repeated ten times to get rid of all the transient behaviour the simulation can have. The number of internal iterations for the unsteady Euler simulation is set to 500.

HB solver

Variable	Value
Freestream Mach number [-]	0.796
Freestream Temperature [K]	293
Freestream Pressure [Pa]	101325
α_0[deg]	0
Spectral Period [s]	0.06
Grid Movement ω [rad/s]	104.72
Grid Movement Origin [x/c]	0.25
Grid Movement Pitching Amplitude [-]	1
CFL Parameter [-]	1.5
Adjoint Objective [-]	Drag

Unsteady EULER

Variable	Value
Freestream Mach number [-]	0.796
Freestream Temperature [K]	293
Freestream Pressure [Pa]	101325
α_0[deg]	0
Time step [s]	0.000375
Total simulation time [s]	0.6
Internal iterations [-]	500
Grid Movement ω [rad/s]	104.72
Grid Movement Origin [x/c]	0.25
Grid Movement Pitching Amplitude [-]	1
CFL Parameter [-]	1.5

Table 5.3: HB solver & unsteady RANS simulation parameters

SU2 provides multiple linear preconditioners to be used for the simulation. As mentioned in subsection 2.5.3 the options are $JACOBI$, $LUSGS$, ILU and $LINLET$. For discrete adjoint methods, the only compatible preconditioners within SU2 are the JACOBI and ILU methods [21]. In preliminary runs of the simulation, it has been proven that the ILU preconditioner is more efficient in finding an optimal solution than the JACOBI preconditioner and will therefore be used for the remainder of this study.

5.4. Airfoil parameterisation method

Parameterising is the transformation of a geometric shape into a mathematical model. This step is indispensable in the design optimisation problem because it controls the fidelity and range of control of the problem. The choice of parameterisation method is highly problem-dependent and therefore has led to a large number of different parameterisation techniques. Within the standard version of SU2, only two main techniques are available for shape parameterisation, Hicks-Henne [32] bump functions or Free-Form Deformation (FFD) [19]. The remaining other parameterisation methods could be implemented in the source code, but the options currently in the software are enough to correctly parameterise the airfoil.

The Hicks-Henne shape function method [32] is based on the analytical approach of shape parameterisation. It is also the most commonly used shape parameterisation method for 2D problems. It uses a linear combination of basis functions and a set of perturbation functions which are linearly added to the geometry. Hicks-Hennne shape functions or bumps are defined as:

$$f_i(x) = \sin\left(\pi x^{\frac{\log 0.5}{\log t_1}}\right)^{t_2} \tag{5.6}$$

The perturbation functions $f_i(x)$ can be added to the base shape function to represent either the upper or lower side of the airfoil shown in Equation 5.7. The t_1 and t_2 values can be changed in order to create a better fit in the design problem. With $\bar{\alpha}_i$ being the design variables in the parameterisation problem.

$$y(x) = y_{base} + \sum_{i=1}^{M} \bar{\alpha}_i f_i(x) \tag{5.7}$$

The fact that the Hicks Henne functions change the shape of the airfoil by changing the $\bar{\alpha}_i$ value and thus the shape of each perturbation function. Make the use of this analytical method less initiative for hands-on airfoil design, whereas for optimisation problems, this is not an issue.

The free-form deformation method, on the other hand, is very robust and is suitable for complex geometries. The method uses surface morphing techniques which originate from deforming solid models. The geometry is encapsulated in a lattice box which in turn is parameterised by a Bezier box. The points on the Bezier box can be defined by the following equation:

$$X(u,v,w) = \sum_{i,j,k=0}^{l,m,n} P_{i,j,k} B_j^l(u) B_j^m(v) B_j^n(w) \tag{5.8}$$

With B^i representing the Bezier coefficient of the order i and (u,v,w) are values between 0 and 1. The Cartesian coordinates of the points on the geometry are transformed onto the parametric space of the Bezier box. The box is deformed according to the design variables, and thus the enclosed geometry will be deformed. The deformed Cartesian coordinates can then be calculated using Equation 5.8.

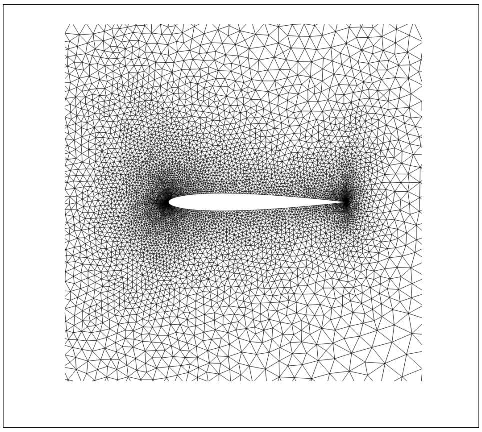
The ease of use of the Hicks-Henne method has led to the choice to implement this in the simulation for this study. Furthermore, the extended robustness of the FFD method is not necessary for the simple airfoil geometry. The deformation of the airfoil needs to be done with high fidelity. Therefore, a large number of Hicks Henne bump functions are needed for geometry parameterisation. In this work, similar to what has been done in the work of Rubino et al. [45] 50 Hicks Henne bump functions will be used to define the shape of the airfoil in the optimisation problem [32].

5.5. Grid generation

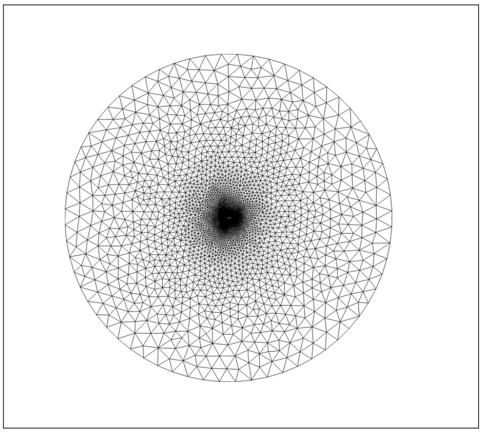
For this simulation, the NACA0010 airfoil is taken as the initial airfoil shape. This is an excellent base airfoil for the optimisation problem due to its symmetric shape and medium thickness. One could argue that this airfoil is fractionally thicker than expected for a propeller blade airfoil. This is easily solvable in further research. However, this research is focused on creating a method which is universally applicable to propeller airfoil shapes. This means that the airfoil choice at this point is relatively arbitrary. For this reason, the thickness of the airfoil will be kept the same throughout the entire optimisation.

For mesh generation, the readily available programme GMSH has been used due to the simplicity of operation and the possibility of exporting to .su2 extension formats. The airfoil shape has been created using the NACA 4-series equations and written to the GMSH input file extension .geo format. From there, GMSH has a very intuitive GUI which leads through the process of creating the mesh boundaries and SU2 markers. An internal volume was added to the 2D airfoil at the end of the .geo file before the mesh was exported into SU2 format. The internal volume acts as a placeholder such that SU2 is able to write the volume files after each simulation.

The domain of the grid is defined as a circular domain which spans 20 chord lengths in diameter as shown in Figure 5.4b. The choice for a circular domain is related to this research using an Euler solver, which only computes the pressure difference in the domain. No viscous effects are simulated, so there is no need for a C domain to investigate the wake created by the airfoil. Furthermore, a circular O gird generally has a lower cell count than a C gird. This means that for the same performance, a higher cell count can be chosen in the case of an O grid.



(a) NACA 0010 airfoil mesh



(b) NACA 0010 mesh domain

Figure 5.4: The 2D NACA 0010 airfoil and its domain

The mesh, as seen in Figure 5.4, is an unstructured 2D mesh composed entirely out of triangular elements, and due to the simulation being inviscid, no boundary layer refinement is necessary. The mesh has been created using the MeshAdapt algorithm [24], which uses local mesh modifications in order to refine the mesh. Within the mesh cells, the long edges are split while the short edges are collapsed. Furthermore, if an improved geometrical configuration is obtained, the edges are swapped. Within GMSH, the mesh scaling parameter is used in order to generate multiple options in mesh refinement. This is to find a well-balanced mesh in terms of the number of elements and accuracy of representing the flow. Because, as commonly known, more elements in a mesh mean a more detailed flow resolution. Therefore a mesh refinement study is done with meshes varying from approximately 9000 elements to approximately 30000 elements. The meshes and their respective amount of elements are presented in Table 5.4

Mesh #	Number of elements	Boundary Points	Airfoil Points
1	8696	49	198
2	11590	52	198
3	15376	64	198
4	19554	80	198
5	25004	98	198
6	32042	110	198

Table 5.4: Meshes used for the grid refinement study

The simulation that has been used for this refinement study is similar to what will be used in the optimisation problem. The inflow conditions are also implemented in this simulation. Furthermore, the simulation would be stopped at a maximum of 15000 iterations to see how the resolve time compares to the convergence of the residuals. The simulation uses the same frequencies as inputs for the HB solver, as well as the same free stream conditions. The convergence of the meshes is close to the operating conditions of the optimisation method. All simulations were run on the HPC cluster at the TU Delft using 32 cores in parallel. The meshes are shown in the number of elements on the X axis in both Figure 5.6 and Figure 5.5.

All meshes show similar behaviour in terms of lift and drag coefficient. Mesh 1 is not represented in the figures since this simulation diverged after approximately 600 iterations. Leading to the logical conclusion that 9000 elements are not enough for this optimisation problem.

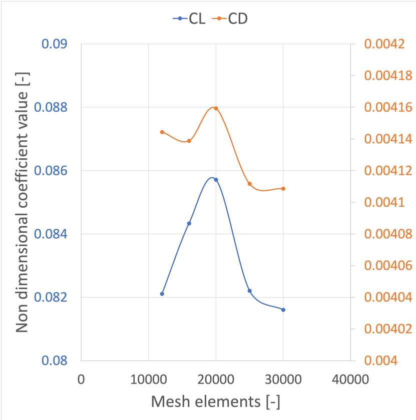
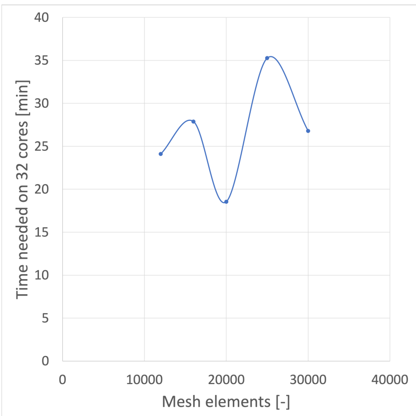
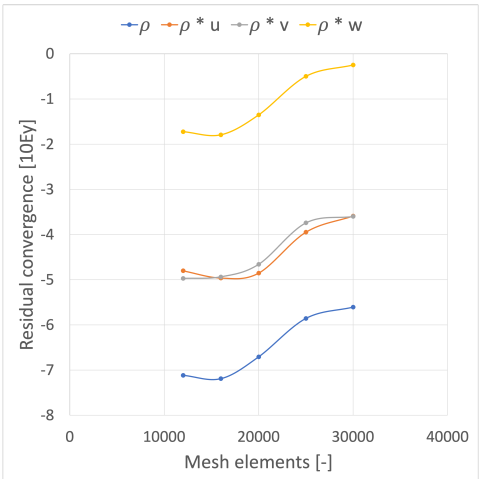


Figure 5.5: Lift and drag coefficient behaviour for different meshes

The convergence, shown in Figure 5.6, of the rest of the meshes showed the behaviour which would be expected. Mesh 5 and mesh 6 show a lack of convergence in Figure 5.6 when compared to the other meshes. Mesh-2 shows good promise in terms of convergence and resolving time. However, being closer to the diverged 9000 element mesh the choice is to discard this mesh as more complex shapes are expected in the optimisation process. So this leaves mesh-3 and 4 as possible candidates for use in the optimisation. Mesh 4 shows interesting results regarding the time necessary to compute the results. Out of all the meshes, it needs the least amount of time to converge to a solution. Therefore mesh 4 will be used for the rest of the research. It has approximately *19500 elements* with 198 points along the airfoil shape and 80 points along the boundary of the domain.



(a) Convergence time over number of elements



(b) Residual convergence over number of elements

Figure 5.6: Mesh refinement study showing the residual convergence and resolve time

6

Optimisation Problem Setup

In this chapter, the optimisation problem setup is described. The objective function and its constraints are given in section 6.1, with the optimisation strategy presented in section 6.2. The performance of the airfoil is determined by the average lift-over-drag ratio obtained from the unsteady simulation. However, another performance measure is introduced in this study in the form of a Root Mean Squared value of the lift response and will be presented in section 6.3. The gradient validation of the adjoint method compared to the finite differences method will be elaborated upon in section 6.4. Next, the method is verified by remaking the JCP paper written by Rubino et al. in section 6.5. Finally, a baseline performance simulation will be presented in order to have a comparison for the optimised shape. This will be presented in section 6.6.

6.1. Optimisation objective function and constraints

The optimisation problem is defined by the objective function and its constraints. First, the objective function will be discussed, followed by the constraints and their reasoning.

As discussed in section 2.3, the performance of a 2D airfoil can be expressed as the lift-over-drag ratio of the airfoil. A better design results in a higher lift-over-drag ratio. In the case of an unsteady simulation, the behaviour over the entire unsteady period has to be optimised. Only taking one time instance into account for the optimisation problem leads to an ill-optimised airfoil shape. Increasing the lift-over-drag ratio can be achieved by decreasing the drag coefficient of the airfoil while keeping the lift coefficient equal to the starting value. The objective function can then be defined as the average drag coefficient over all time instances.

The optimisation is subject to a number of constraints which are set to reach the requested performance increase and is also used as a secondary performance objective. An equality constraint is introduced for the average lift coefficient of the airfoil due to the lift-over-drag ratio needing to increase at a given lift coefficient. The average value for the lift coefficient equality constraint is used because the lift value of one time instance does not hold sufficient information over a periodic flow. The initial lift coefficient given as $\overline{C_{l_0}}$ is determined by a direct harmonic balance simulation using the initial NACA 0010 airfoil.

The Root Mean Squared Error (RMSE) of the lift response, or $C_{l_{RMSE}}$, is introduced as an inequality constraint. The reason for the use of the RMSE of the lift will be elaborated upon in section 6.3. This inequality constraint can be used as a secondary objective. It can be manually lowered by means of changing the value ξ until the simulation diverges, as seen in Equation 6.1. This is what has been done to achieve the performance increase with respect to the RMSE of the lift value.

Finally, an inequality constraint for the maximum thickness of the airfoil has been set to be greater than the value at the start of the optimisation. Smaller propellers have a thickness at 90% chord ranging from 6% thickness to 13% thickness[51]. Although this propeller is significantly larger than the one in the work of Slavik et al., the 10% thickness would still be the right choice. Thinner airfoils will yield less drag however the aerodynamic forces would not allow a propeller blade to take such a shape. The full objective function with the corresponding constraints is presented in Equation 6.1. Where δ is defined as the thickness of the airfoil

$$
\begin{aligned}
\text{Minimise for } \overline{C_d} \quad & \overline{C_d}(\boldsymbol{U}_n, \boldsymbol{X}_n, \boldsymbol{\alpha}) \\
\text{Subject to} \quad & \overline{C_l} = \overline{C_{l_0}} \\
& \delta_{max} > \delta_{max_0} \\
& C_{l_{RMSE}} < \xi \cdot C_{l_{RMSE_0}}
\end{aligned}
\tag{6.1}
$$

As an alternative optimisation run, the $C_{l_{RMSE}}$ value has been used as the objective function. Therefore, the drag coefficient changes to an inequality constraint smaller than the initial value. The other constraints are kept the same to be able to compare the different optimisation runs. The objective function and set of constraints can be seen in Equation 6.2

$$
\begin{aligned}
\text{Minimise for } C_{l_{RMSE}} \quad & C_{l_{RMSE}}(\boldsymbol{U}_n, \boldsymbol{X}_n, \boldsymbol{\alpha}) \\
\text{Subject to} \quad & \overline{C_l} = \overline{C_{l_0}} \\
& \delta_{max} > \delta_{max_0} \\
& \overline{C_d} < \overline{C_{d_0}}
\end{aligned}
\tag{6.2}
$$

6.2. Optimisation strategy

Within SU2, the optimisation process is guided by a python script compiled from the source code. This script takes all the necessary steps for the optimisation process to minimise the objective function. The default optimisation algorithm used in SU2 is the Sequential Least Squares Programming (SLSQP) algorithm originating from the scipy pack.

Lyu et al. [40] have benchmarked a number of optimisation algorithms for an aerodynamic shape problem by means of a RANS simulation in SU2. They found that the overall performance of the SLSQP algorithm works best in the aerodynamic shape optimisation problems. After an initial instability where the objective function fluctuates to extreme values in the first couple of iterations, the SLSQP algorithm quickly finds the minimum for the objective function. This is the reason why the SLSQP algorithm is chosen for this work. Quick convergence and the robustness of operation make for a very suitable transonic airfoil optimisation algorithm.

As mentioned above, the shape optimisation process is managed by the *shape_optimisation.py* file compiled from the source code. The schematic view of the optimisation strategy can be seen in Figure 6.1. Here, the different steps the python script takes are presented. The airfoil parameterisation, which is performed using the 50 Hicks Henne bump functions, is done within the geometry deformation block. For every updated design vector $\vec{x}$, the geometry deformation block is activated to change the airfoil shape.

6.3. Secondary performance indicator

During any design optimisation problem, the objective function dictates the direction of the optimisation strategy. In this research, the unsteady optimisation problem uses the average drag coefficient as the objective function. This leads to the possibility of the lift response amplitude increasing as did in the case of Economons work [20], where the amplitude increased by a factor of 2. The lift fluctuations can be related to a change in blade loading. Typically the change in blade loading over a revolution results in extra noise created [30, 43].

For this reason, a secondary performance indicator has been added to the source code of SU2 in the form of the RMSE value of the lift response. This indicator is able to give an indication of the amplitude of the oscillating lift coefficient. The error measure used for this indicator is therefore set to the time average of the lift coefficient. When encountering unsteady inflow conditions, the change in lift coefficient over time leads

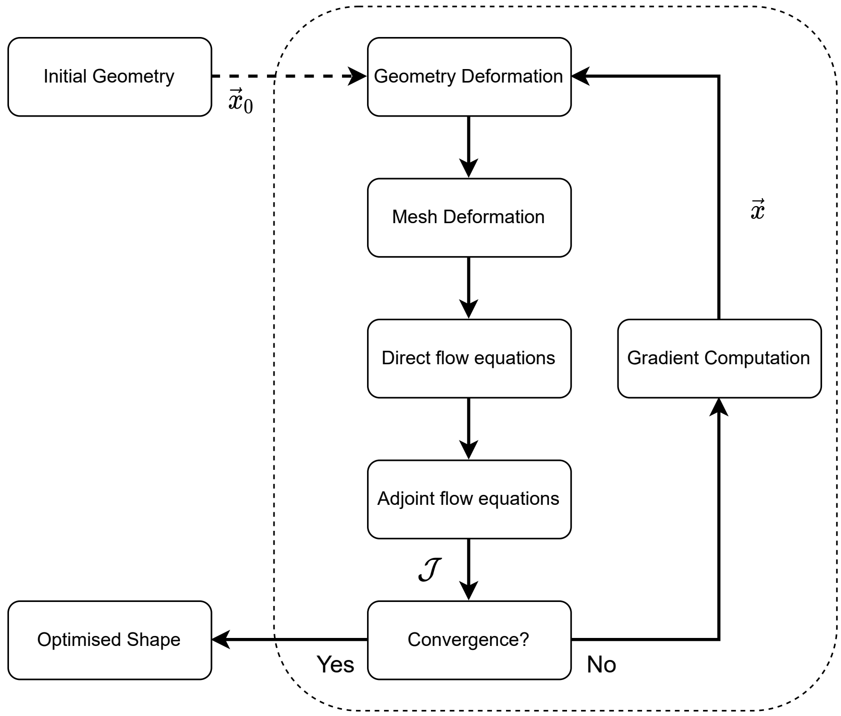


Figure 6.1: Optimisation strategy embedded within SU2

to a time-varying blade loading. The increased noise in the installed condition is dominated by the propeller experiencing a time-varying loading. This increase in noise is mainly a nuisance to the passengers inside the aircraft and the environment around the aircraft.

From literature, it is known that the time-varying blade loading of a propeller influences the noise production of a propeller blade [29, 56]. Limiting this can reduce the loading noise produced by the propeller and aid in more comfort for the passengers and less of a nuisance to the environment. As mentioned in section 2.3, the thrust coefficient of the propeller can be related to the lift coefficient of the airfoil. Reducing the variation of the lift coefficient over the rotational period of the propeller would conversely lower the variation of the blade loading of the propeller and noise emissions. A measure to quantify the variation of the lift coefficient has to be employed for the CFD solver to converge to a proper shape. Therefore the Root Mean Squared Error (RMSE) value of the lift coefficient will be used as this metric. The reason for using the RMSE value rather than the Root Mean Squared value is that, irrespective of where the average lift coefficient is over the period, the RMSE value will always give a measure of the amplitude of the periodical response. The equation for the RMSE value can be found below:

$$RMSE = \sqrt{\frac{\sum_{t=1}^{T}(\hat{y}_t - y_t)^2}{T}} \tag{6.3}$$

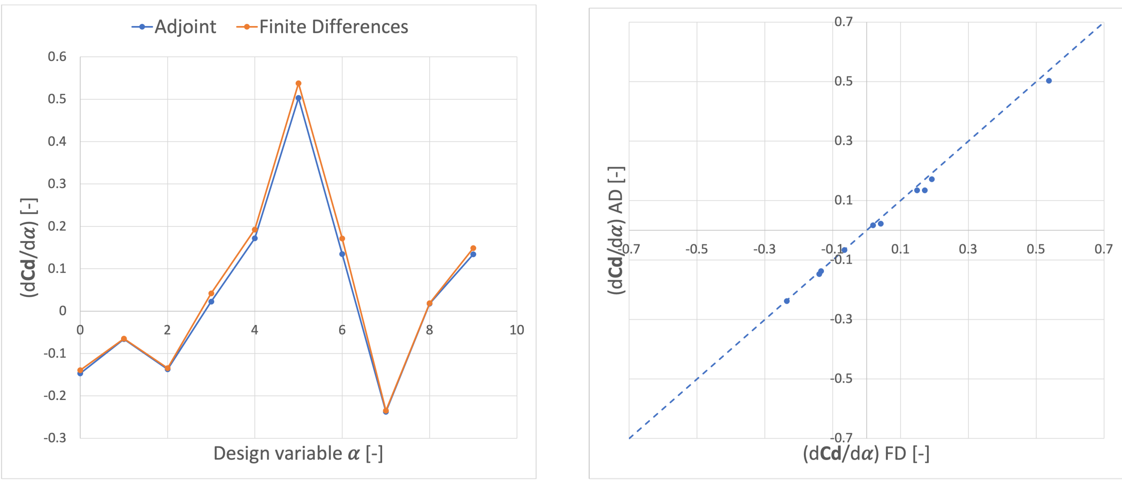
As mentioned, $\hat{y}$ is equal to the time average lift coefficient of all the time instances, y_t is defined as the lift coefficient for each individual time instance, and T is the total number of time instances used for the simulation. The expression which is used for the RMSE value of the lift response is given below.

6.4. Gradient validation

The optimisation scheme which is used for this research is a gradient-based optimisation method. This means that in order to get to the optimal solution, the gradient of the objective function has to be determined. The adjoint method is an alternative method to determine the gradient of each individual design variable of the objective function. In order to use the adjoint gradients in a design optimisation problem, the validity of the gradients has to be checked in the form of a gradient validation. The gradients determined by a second-order finite differences method are considered to be the exact gradients and are therefore used as the baseline. The adjoint method gradients are compared to these exact gradients. Ideally, the gradients of both methods match each other. However, due to numerical errors, the gradients of both methods can have some sort of deviation. The step size taken for the second-order finite differences method is 0.001. This is the same step size which the adjoint method used.

As mentioned in section 3.3, the main advantage of the adjoint method is the reduced computational time when calculating the objective function gradients. The finite differences method resolves the direct flow equation for each design variable once, whereas the adjoint method only resolves the direct and adjoint equation once for all design variables. So the total computational time of the finite differences is highly reliant on the number of design variables, where this is not the case for the adjoint method. For this reason, only ten control points are taken for this gradient validation process, five on the suction side and five on the pressure side, going from the leading edge to the trailing edge. It is expected that the intermediate points on the airfoil will show similar results to the control points. The control points are evenly spaced on the upper and lower side of the airfoil as shown in Figure 6.4.

Figure 6.2 shows the results of the gradient validation process performed using the NACA 0010 airfoil for the real inflow conditions with respect to the drag coefficient. Where Figure 6.2a shows the gradients computed using both the adjoint and Finite differences method for each individual design variable. Figure 6.2b shows the gradients of the adjoint design variables with respect to the finite differences method. In the ideal situation, all the points lie on the dashed y=x line. Both methods used a step size of 0.001 during the process of determining the gradient.



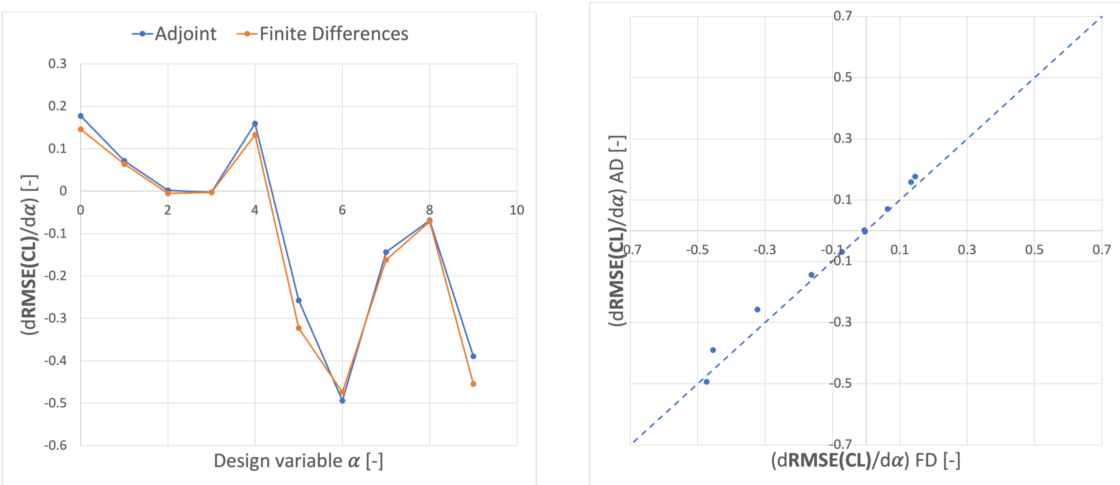
(a) FD and Adjoint gradients with respect to design variables

(b) FD drag gradients with respect to the AD gradients

Figure 6.2: Drag objective comparison between the gradients obtained using a Finite Differences method and the Adjoint method

The gradient validation process for the alternative optimisation run, where the $C_{l_{RMSE}}$ value has been used as the objective function, is shown in Figure 6.3. Similar to what has been said above, Figure 6.3a shows the gradients computed using both the adjoint and Finite differences method for each individual design variable. Figure 6.3b shows the gradients of the adjoint design variables with respect to the finite differences method.

As seen in Figure 6.2 and Figure 6.3, the gradients of both the finite differences method and the adjoint method agree and show a good match. The first five gradients are located on the pressure side, and the last five are on the suction side, as seen in Figure 6.4. The gradients for both the drag coefficient and the $C_{l_{RMSE}}$ objective show a slight discrepancy between the two methods, but this is expected due to numerical errors. The gradients are close enough to assume the gradient validation to be complete. The gradients of the adjoint method can be used in the further optimisation runs for both the drag coefficient and $C_{l_{RMSE}}$ objective functions.



(a) FD and Adjoint gradients with respect to design variables

(b) FD drag gradients with respect to the AD gradients

Figure 6.3: $C_{l_{RMSE}}$ objective comparison between the gradients obtained using a Finite Differences method and the Adjoint method

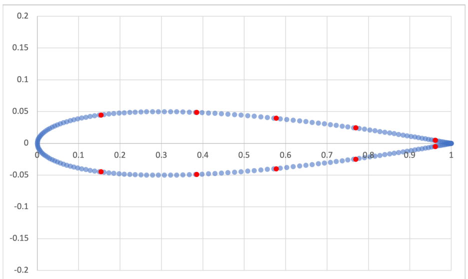
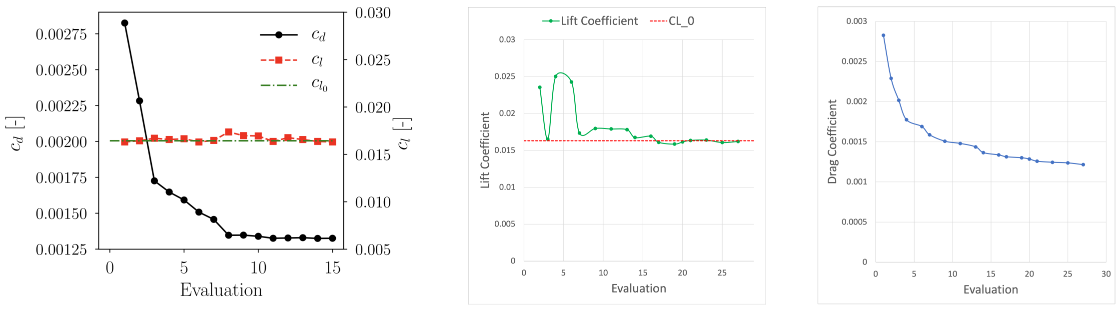


Figure 6.4: NACA 0010 gradient validation points

6.5. Method verification

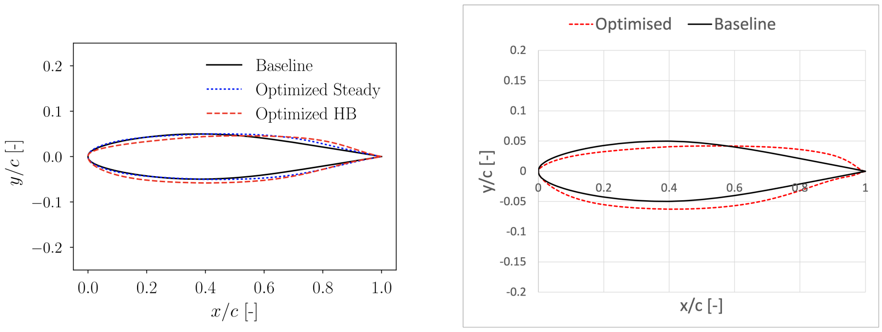
The method was verified by rerunning the optimisation presented by Rubino et al. [45]. Initially, the verification run did not converge while using the same SU2 settings as Rubino et al. used. After some inspection and research on the SU2 website, it was mentioned that for the adjoint method, SU2 only supports either the ILU or JACOBI linear preconditioner. Rubino et al. used the LU SGS linear preconditioner in their work. After changing the linear preconditioner to the ILU, it solved the issue and led to a successful verification run. The lift coefficient converged to the same value as in the Rubino et al. paper. The drag coefficient showed the same reduction of 50% as the paper did. The convergence for both the lift and drag are shown in Figure 6.5. The drag coefficient in the verification run shows similar behaviour as for the original paper. The only difference is the number of evaluations necessary to reach the convergence. This could be due to some updates that have been performed on SU2 since the original research.

The airfoil shapes, seen in Figure 6.6, also look to be deformed in a similar way to the Rubino et al. research. Both HB-optimised airfoils show more body on the pressure side of the airfoil across the full chord length. Also, the increased thickness at around 0.9 x/c can be found in both optimisation runs. Although the amount of evaluations necessary is not the same, the endpoint of the optimisation schemes is the same. These are the main points of this verification run and prove that the method currently in the SU2 source code is the right method. This leads to the conclusion that the results which are returned can be assumed to be valid.



(a) Rubino optimisation [45] (b) Lift coefficient verification (c) Drag coefficient verification

Figure 6.5: Convergence of the lift and drag coefficient verification of the Rubino et al. [45] paper.



(a) Rubino optimisation airfoil geometry [45] (b) Airfoil geometry verification run

Figure 6.6: Airfoil geometry verification of the Rubino et al. [45] paper.

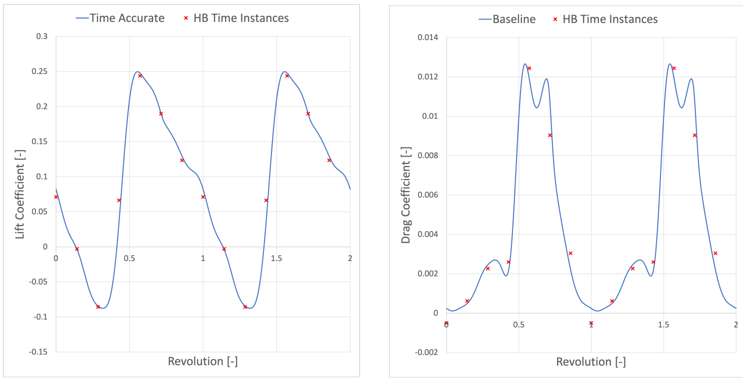
6.6. Baseline performance

The baseline NACA0010 airfoil has been simulated using the initial solver conditions given in chapter 5. Using the inflow conditions modelled by the previously mentioned inflow simulation presented in chapter 4. In this section, the results of this baseline run will be presented, combined with the possible bottlenecks this airfoil has for the initial conditions. Furthermore, the 2nd order dual time stepping simulation that has been performed for the frequency selection is presented, as is the steady state performance of the airfoil.

The lift coefficient coming from the time-accurate simulation shows good agreement with the proposed change in angle of attack, as seen in Figure 4.10. The drag coefficient looks a bit complex, but this is not something that is strange in this situation. The initial lift coefficient is higher than 0 due to the inflow conditions being modelled as they are now. In section 4.2, the choice was made to have the first datapoint start at 0 degrees angle of attack point to get the change in angle of attack. This led to the modelled angle of attack starting at a slightly higher than 0 value. This should not influence the simulation substantially due to the general tendency of the change in the angle of attack is still there.

The lift and drag coefficient of the NACA 0010 airfoil at each time instance obtained using the direct flow equations based on the harmonic balance method are presented in Table 6.1. The values are also shown in Figure 6.7 and show good agreement with the time-accurate simulation. Interestingly the lift coefficient has better agreement with the time-accurate solution than the drag coefficient. The drag coefficient shows a slight outlier in the first time instance. Here the HB drag coefficient dips below the expected value of the time-accurate simulation. As mentioned in section 4.2, the most important feature is the lift coefficient in this optimisation. Therefore the slight discrepancy in drag coefficient is regarded as within the allowable tolerances of inaccuracy.

The convergence rate of the direct flow and adjoint equations are shown in Figure 6.8 as a function of the number of iterations. The density residual shows that the adjoint solver convergence is inherited from the direct flow solver. The time of the direct flow solver to compute a single iteration is 0.06 seconds on 48 cores



(a) Lift coefficient (b) Drag coefficient

Figure 6.7: Time accurate lift and drag coefficients

Time instance	Lift coefficient Adjoint	Drag coefficient Adjoint
0	0.0711	-0.0005
1	-0.0028	0.0006
2	-0.0857	0.0022
3	0.0662	0.0025
4	0.2438	0.0124
5	0.1899	0.0090
6	0.1233	0.003

Table 6.1: Lift and drag coefficient at each individual time instance for the NACA 0010 airfoil simulated using the real inflow.

AMD Opteron 6234 Processor running at 2.4 GHz. The ratio of computational time between the adjoint solver and the direct flow solver is approximately 1.3.

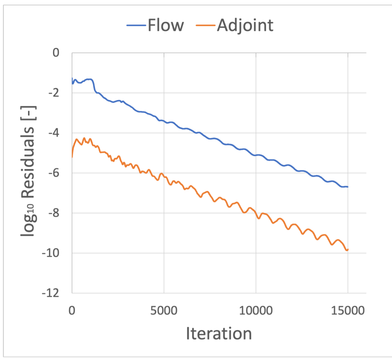
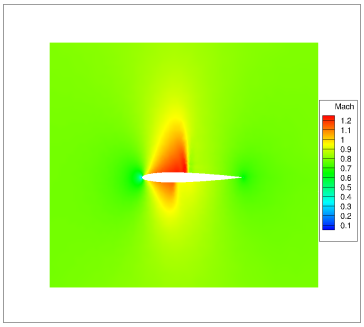
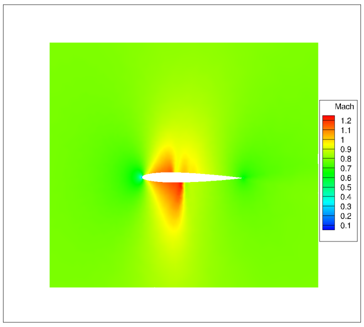


Figure 6.8: Solver convergence history for the direct flow and adjoint density residual over an entire simulation.

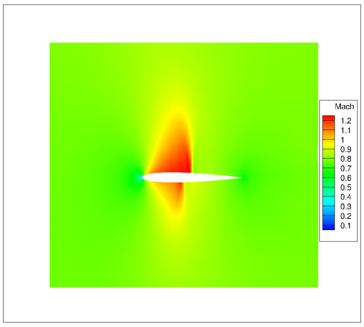
The baseline run has been performed using the same seven time instances as will be used during the design optimisation problem. Figure 6.9 shows the contour plots of the Mach number and Figure 6.10 the pressure distributions for three time instances divided evenly over the full period. What is evident is that the airfoil experiences a strong shock over the suction side throughout the entire period. This can be seen by the strong decrease in C_P when looking at Figure 6.10. This causes the drag coefficient to increase relative to an airfoil in the absence of the shock. This is also where the optimiser can prove to be effective by lowering the intensity of the shock as well as the drag coefficient. The RMSE value of the lift coefficient response can be lowered by decreasing the c_{l_α} value which makes the airfoil less sensitive to the change in angle of attack. The lift coefficient RMSE value of the NACA 0010 airfoil amounts to 0.1036 for the harmonic balance method and 0.1108 for the time-accurate results. The value itself, on its own, does not really say something. Rather the reduction of the value is something that is important in this study. Comparing the reduction of both the HB results and the time-accurate results is of interest to check whether the HB solver accurately represents the results.



(a) T=0

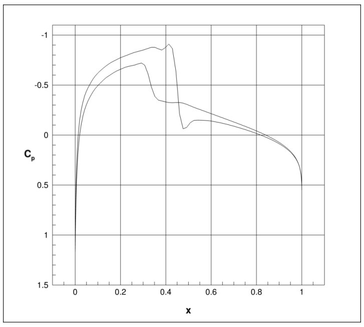


(b) T=3/7

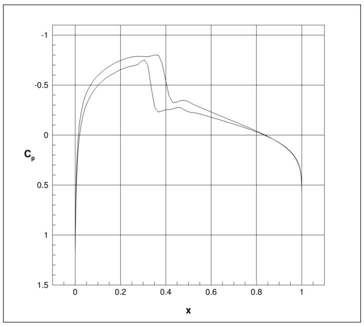


(c) T=6/7

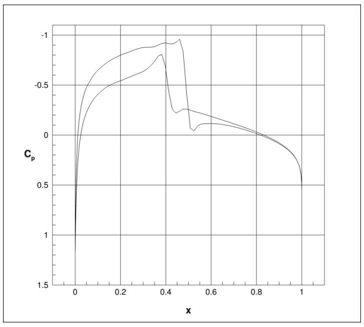
Figure 6.9: NACA 0010 Mach number contours for three different time instances computed with the adjoint harmonic balance method



(a) T=0



(b) T=3/7



(c) T=6/7

Figure 6.10: NACA 0010 pressure coefficient distributions for three different time instances computed with the adjoint harmonic balance method

7

Results

In this chapter, the results presented were obtained using the adjoint-based harmonic balance optimiser developed in the former chapters. The different optimisation runs are presented along with the explanation for the importance of each run in section 7.1. An optimisation is run using simple sine inflow conditions and is presented in section 7.2. Where in section 7.3 and section 7.4, the real inflow baseline and $C_{l_{RMSE}}$ value constrained optimisation runs are presented respectively. Next, an optimisation run has been performed to verify one of the constraints in section 7.5. Finally, an optimisation run is performed where the $C_{l_{RMSE}}$ is used as the objective function, and an inequality constraint is set for the drag coefficient in section 7.6.

7.1. Optimisation runs

For the optimisation problem presented in this work, multiple different inflow conditions and constraints have been used to test the optimiser for robustness and to gain an understanding of the importance of the inflow condition. As mentioned in section 6.2, one of the constraints is used as a secondary objective-like parameter to reduce the lift response amplitude of the airfoil for the real inflow conditions of a propeller in unsteady inflow. The effectiveness of this secondary objective function has yet to be determined. Therefore, this will be compared to a non $C_{l_{RMSE}}$ value constrained problem. Furthermore, an optimisation run is performed with the $C_{l_{RMSE}}$ value as the objective function of the optimisation. No real off-design conditions were assessed during the optimisation processes.

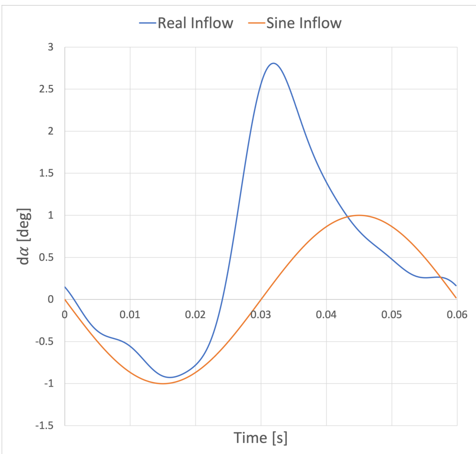


Figure 7.1: Comparison between the real inflow and sine inflow conditions as used for this research

The baseline optimisation problem is run for both the real unsteady and simple sine unsteady inflow conditions to determine the influence of the inflow characteristics. Initially, the optimisation was run using a simple sine inflow with a $C_{l_{RMSE}}$ value constraint, but this optimisation kept diverging. Therefore, the choice was made to revert back to a non $C_{l_{RMSE}}$ value constraint optimisation run to determine the influence of a different inflow. The simple sine inflow is characterised by the airfoil angle of attack varying over time like a sine function. Running the optimiser with a different inflow condition is for the purpose of checking whether the optimiser is not tailoring the airfoil shape for only one set of inflow conditions. If the optimiser did so, the optimiser settings would need to be changed, and a more robust optimiser would need to be created.

Figure 7.1 shows a comparison between the real inflow and the sine inflow conditions. The goal of this investigation is to check whether the optimiser while encountering a different periodical unsteady inflow condition, a similar airfoil shape can be found. In order to keep a similar airfoil performance in terms of lift coefficient, the lift coefficient is set to be equal to the original baseline NACA 0010 lift coefficient. The drag coefficient is taken as an objective for both these optimisation runs.

Secondly, the influence of the $C_{l_{RMSE}}$ value constraint is demonstrated by means of optimisation runs with and without $C_{l_{RMSE}}$ value constraints. The effectiveness of this method is determined by the reduction in $C_{l_{RMSE}}$ compared to the baseline optimisation for real inflow. An overview of what simulations have been run is given in Table 7.1.

During the research, the airfoil shapes kept approaching the constraint value of the maximum thickness. Therefore, an extra optimisation run is performed such that the influence of this constraint can be investigated. The maximum thickness constraint is lowered to a value of 5 % of the chord instead of the 10 % that was set initially. Only a drag coefficient objective without $C_{l_{RMSE}}$ value constraint for real inflow conditions is performed in this case.

Finally, an optimisation run is performed with the $C_{l_{RMSE}}$ value as the objective function of the optimisation. For the drag coefficient, an inequality constraint is initially set to achieve a similar drag reduction as the other optimisations. The rest of the constraints are also kept the same as for the other optimisation runs.

Inflow conditions	$C_{l_{RMSE}}$ **constrained**	**Non** $C_{l_{RMSE}}$ **constrained**	$C_{l_{RMSE}}$ **objective**
Real unsteady inflow conditions	✓	✓, ✓ (0.5 Thickness)	✓
Sinusoidal unsteady inflow conditions		✓	

Table 7.1: Overview of the various optimisation runs performed in this research

For the $C_{l_{RMSE}}$ constraint optimisation runs, different levels of reduction in $C_{l_{RMSE}}$ have been experimented with, ranging from 10% reduction to 40% reduction in 10% point steps. The reduction of the $C_{l_{RMSE}}$ values are all compared to the NACA 0010 baseline airfoil. The preliminary results of these optimisation runs showed that the 10 % and 20 % constraint value optimisation got stuck at a similar point as where the baseline optimisation ended up. The 30 % reduction constraint gave the optimiser more of an incentive to reduce this value. This incentive has translated into a mere 1 % extra reduction in $C_{l_{RMSE}}$ value. During the optimisation, there was a substantial constraint violation throughout all the evaluations. The optimiser was stopped by hand due to the optimiser not finding a better shape.

During the optimisation, the $C_{l_{RMSE}}$ value did not reach the requested 30% reduction. It did converge to an overall 13 % reduction compared to the NACA 0010 baseline airfoil. With the objective of attaining an as large as possible reduction of $C_{l_{RMSE}}$ while also optimising for the time average drag coefficient, this is considered a successful optimisation. The choice is therefore made to implement an inequality constraint of $C_{l_{RMSE}} < 0.7 * C_{l_{RMSE_0}}$ for the rest of the $C_{l_{RMSE}}$ values constraint optimisation runs. Due to the $C_{l_{RMSE}}$ value being taken as the objective function for the final optimisation run, the drag coefficient needed to be set as an inequality constraint. The inequality constraint is set as $\overline{C_d}$ is less than the $\overline{C_d}$ of the NACA 0010 airfoil drag coefficient.

In section 6.6, the baseline performance of the NACA 0010 airfoil has been presented, and this is used as the initial condition. The Euler solver used in this study only computes the pressure drag created by the pressure variation over the airfoil. Viscous drag is an important part of the total drag of an airfoil. However, to gain a better understanding of the airfoil surface sensitivities and the possible solutions to these sensitivities, the Euler solver is used due to its efficient characteristics. In Figure 6.2, the surface sensitivities with respect to the drag coefficient of the NACA0010 airfoil can be found. However, the main gain for the optimiser will be to reduce the shock intensity of the normal shock over the suction side of the airfoil. All contour plots of the Mach number in this work have been given the same maximum and minimum colour values as well as the same colour levels for each time instance to fully capture the improvement over the airfoil.

The overall duration of each real inflow optimisation run using 48 cores *AMD Opteron 6234 Processor* running at 2.4 GHz amounted to approximately 24-48 hours, varying between optimisation runs. The simple sine inflow simulations have been run on a different computer. This simulation has been run on a ten-core *Intel Core i9-10850K CPU* running at 3.6 GHz. This simulation duration also amounted to approximately 24 hours before coming to a converged solution. For comparison, running a finite difference method-based optimisation for the same amount of design variables, the computational duration would increase by a factor of 25. Therefore, the argument can be made that the efficiency with respect to the optimiser's performance objective of this study has been met.

7.2. Baseline optimisation for simple sine inflow

A baseline optimisation has been run for a simple sinusoidal inflow condition. The reason for this is to eventually be able to verify that the optimiser is able to converge to similar airfoil shapes for different inflow conditions. This section will treat the simple sinusoidal inflow conditions as an initial optimisation test.

With these inflow conditions, the optimiser shows a thinner forward part on the airfoil compared to the NACA 0010 airfoil, as seen in Figure 7.2. The aft part of the airfoil has also changed substantially compared to the NACA 0010 airfoil. The main aim of the optimiser is to lower the average drag coefficient while keeping the lift coefficient equal to the NACA 0010 airfoil. The large shockwave that was observed going over the suction and pressure side of the NACA 0010 airfoil is to be removed since this is the main cause of drag in this problem. The airfoil shape has less of a steep increase in thickness, and the point of maximum thickness has been shifted more aft. This has a large influence on the position and intensity of the shockwave.

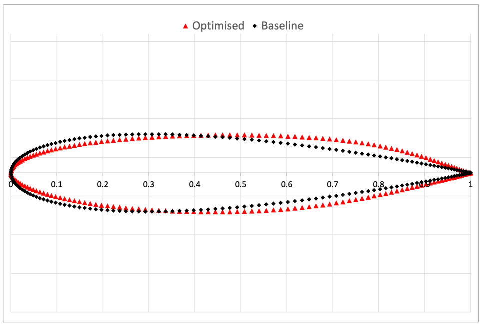


Figure 7.2: NACA 0010 versus the optimisation for a simple sinus wave inflow

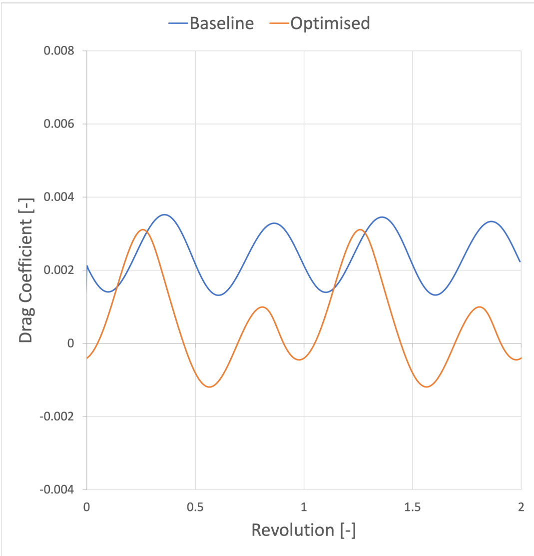
Although the requested optimisation gradient accuracy for the SU2 optimisation of $1E-6$ is not observed in the optimisation results. The gradient accuracy is defined as the gradient of the objective function at that evaluation. However, the optimiser achieved convergence according to the solver with a gradient value of $8.89E-4$ and terminated with a successful optimisation as a result. This leads to the suspicion that the optimiser has reached the gradient accuracy somewhere in the next gradient evaluations, but this is not documented by the solver. The difference between the last subsequent C_D value was $3.93E-6$. A drag coefficient reduction of approximately 78 % has been achieved in this optimisation run. The lift coefficient constraint has been violated by a mere 0.13 %, close enough to the constraint value to assume convergence. The $C_{l_{RMSE}}$ value has been reduced by 8.2 % without a constraint for the $C_{l_{RMSE}}$ value set. A time-accurate simulation has been run with the optimised airfoil shape to verify the HB results. Both the lift coefficient constraint and the $C_{l_{RMSE}}$ value show very good agreement between the HB results and the time-accurate results. The drag coefficient shows a little deviation from the HB results. The optimised airfoil shape has provided a more complex drag coefficient time history, as seen in Figure 7.3. Calculating the average lift and drag coefficient with only 7 data points in the SU2 solver can lead to small inaccuracies in the overall drag coefficient and lift coefficient. In the case of the drag coefficient, it is assumed to be the reason for this.

Parameter	HB results	TA results
C_D difference between evaluations [-]	3.93E-6	-
Drag coefficient reduction [%]	78.0	72.2
Lift coefficient constraint violation [%]	0.13	0.06
Cl_{RMSE} reduction [%]	8.2	8.1
Maximum thickness [%]	10.01	-

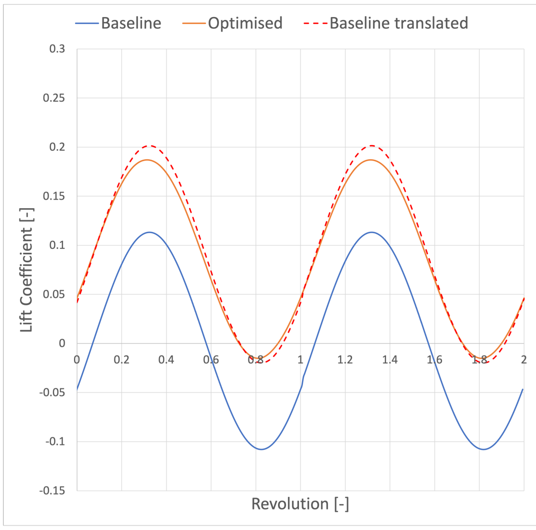
Table 7.2: Optimisation performance for baseline optimisation scenario for a sine wave inflow condition

The maximum thickness constraint fell short by approximately 0.5 % compared to the constraint value, showing a very decent optimiser performance. All relevant optimisation results are presented in Table 7.2 and the shape optimisation history for all parameters given in Figure 7.4.

Figure 7.3 shows the time-accurate simulations of the baseline and optimised airfoil shapes in a simple sine inflow condition. The lift coefficient shows the behaviour to be expected. The average lift coefficient is slightly higher due to the constraint set being the same as the average lift coefficient for the real inflow conditions. To be able to see the difference in Cl_{RMSE} value, the baseline lift coefficient time accurate simulation has been translated to start at the same value as the optimised airfoil, as seen in Figure 7.3 by the red dashed line. The drag coefficient, on the other hand, shows very interesting behaviour. The sine-shaped inflow condition has turned into a more complex drag coefficient shape. This is probably due to the disappearance of the normal shock over the suction and pressure side of the airfoil, as seen in section 6.6. The supersonic bubble changes, as seen in Figure 7.6 size and shape every time instance.



(a) Drag coefficient

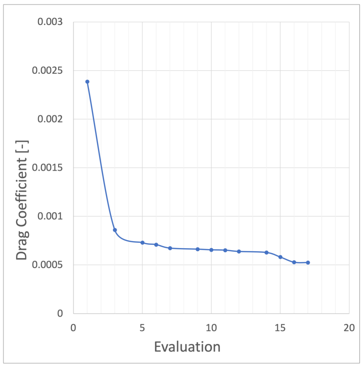


(b) Lift coefficient

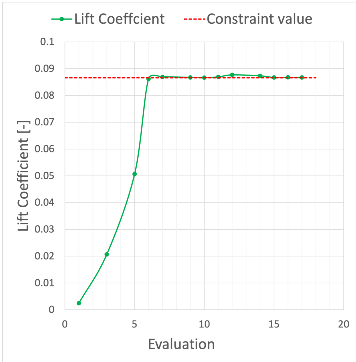
Figure 7.3: Lift and drag coefficient from the time accurate simulation for the baseline and optimised airfoil in simple sine inflow conditions

Figure 7.4 shows the performance of the optimiser for each evaluation during the optimisation process. The lift coefficient has been rising for the first 4 evaluations because of the constraint value being set to the same value as for the real inflow conditions. This did not lead to any disturbance in the optimiser and reduced the drag coefficient. This could be because the optimiser has proven to be robust or because the increase in lift coefficient is very little. The optimisation took approximately 17 evaluations to converge to a solution.

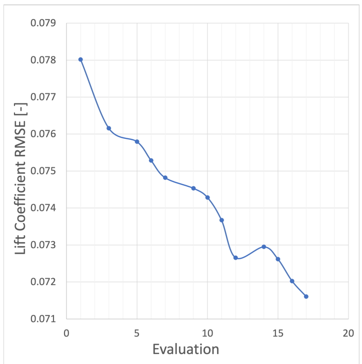
The pressure distributions show similar behaviour as for the real inflow optimisation. The forward part of the airfoil has a relatively flat pressure profile. The shock has completely disappeared compared to the NACA



(a) Drag coefficient



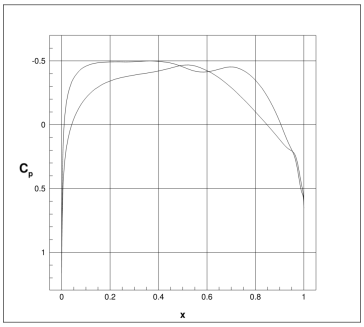
(b) Lift coefficient



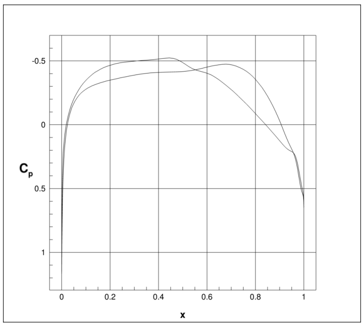
(c) $C_{l_{RMSE}}$ value

Figure 7.4: Optimiser performance of optimisation for sine wave inflow conditions given by the convergence figures of the drag and lift coefficient. The secondary performance objective, the $C_{l_{RMSE}}$ value, is also presented in this figure.

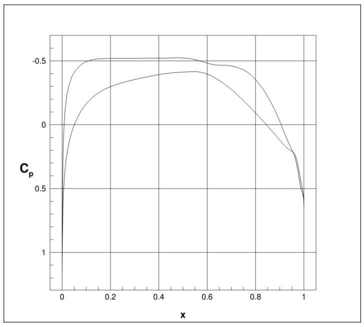
0010 airfoil. The pressure coefficient has stabilised over the suction and pressure side. Due to the less severe increase in the angle of attack, the airfoil is able to keep the pressure distribution at this level. Meaning that over the chord length of the airfoil, the pressure coefficient seems to stay constant until the halfway point of the chord.



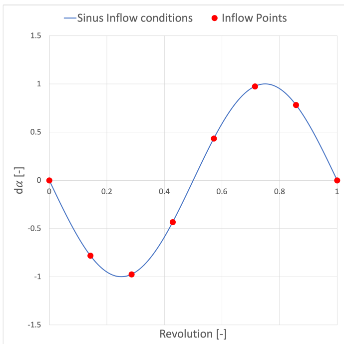
(a) T=0



(b) T=3/7



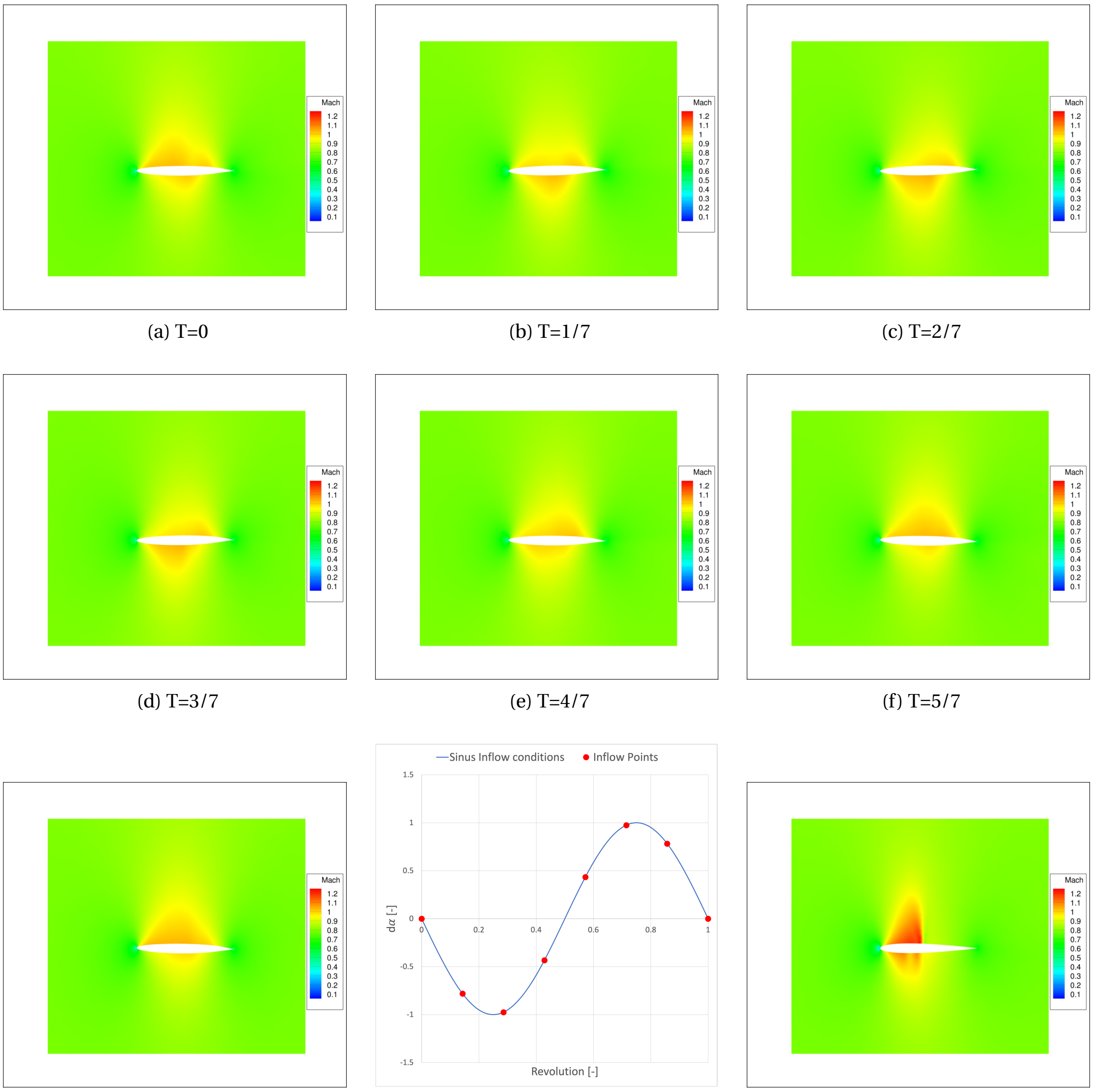
(c) T=6/7



(d) Time point overview

Figure 7.5: The pressure distributions of various time instances of the optimised airfoil shape for a drag coefficient objective function in simple sinusoid inflow

Finally, the contour plots of the Mach number for the airfoil of this optimisation are presented in Figure 7.6. With the baseline NACA 0010 airfoil contour of the Mach number being presented in Figure 7.6*i*. The first thing that stands out is the similar behaviour as all optimisation runs. The strong normal shock over the suction side has completely disappeared for the optimised shape. Furthermore, the shock on the pressure side of the airfoil also has disappeared. The overall Mach number has decreased over the airfoil attributing to the drag coefficient reduction.



(a) T=0 (b) T=1/7 (c) T=2/7

(d) T=3/7 (e) T=4/7 (f) T=5/7

(g) T=6/7 (h) Time point overview (i) NACA 0010 comparison

Figure 7.6: Mach contours of the $C_{l_{RMSE}}$ constrained NACA 0010 optimisation for a sine wave inflow condition. All time instances computed with the adjoint harmonic balance method are presented. Figure (i) shows the baseline contour of the Mach number of the NACA 0010 airfoil.

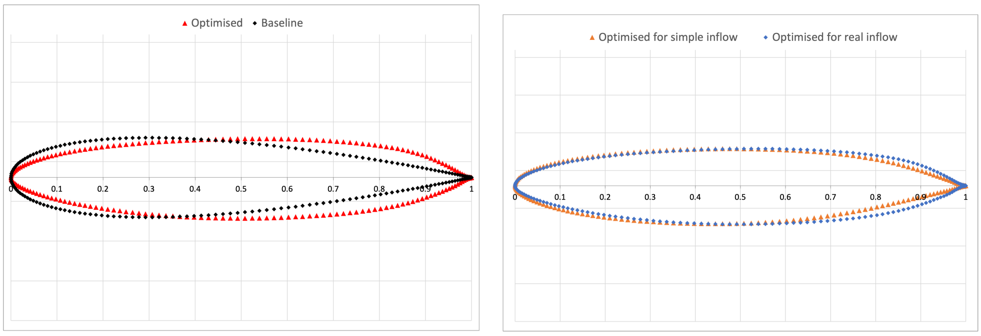
7.3. Baseline optimisation for real inflow

This section treats the optimisation run performed using the real inflow conditions as mentioned in chapter 4. The airfoil shape, general results and pressure distributions are presented along with the contour plots of the Mach numbers for all time instances. The two different inflow conditions are compared with regard to the airfoil shape to which the optimiser has converged, checking the optimiser's robustness to different inflow conditions in the process.

What is immediately evident when looking at the airfoil shape in Figure 7.7 is that the optimiser converged to an airfoil shape that substantially changed the forward and aft part of the airfoil. The forward part has become more slender while the aft part has gotten more thickness to it. Moving the point of maximum thickness is beneficial for decreasing the shock intensity over the airfoil. A more aft point of maximum thickness of the airfoil leads to a less sudden increase in velocity over the airfoil. Sudden increases in velocity over the airfoil can cause severe shockwaves.

This is something which is also seen in section 7.2 where the inflow conditions are very different. The trailing edge also shows a similar shape to the simple inflow simulations. However, the thickness at the aft part of the airfoil has decreased more compared to the simple sine inflow case. The simple inflow case has less defined forward and aft changes compared to the real inflow conditions. This could be due to the extreme increase in the angle of attack for the real inflow situation.

Due to the simulation assuming an inviscid flow, the possibility of separation is not taken into account when computing the lift and drag coefficient. The trailing edge of the airfoil shows an increased thickness compared to the baseline airfoil. The suction side shows a steep return to the centerline of the airfoil at around $0.95x/c$. This can also cause the airfoil to experience separation at the trailing edge and cause a substantial increase in drag.

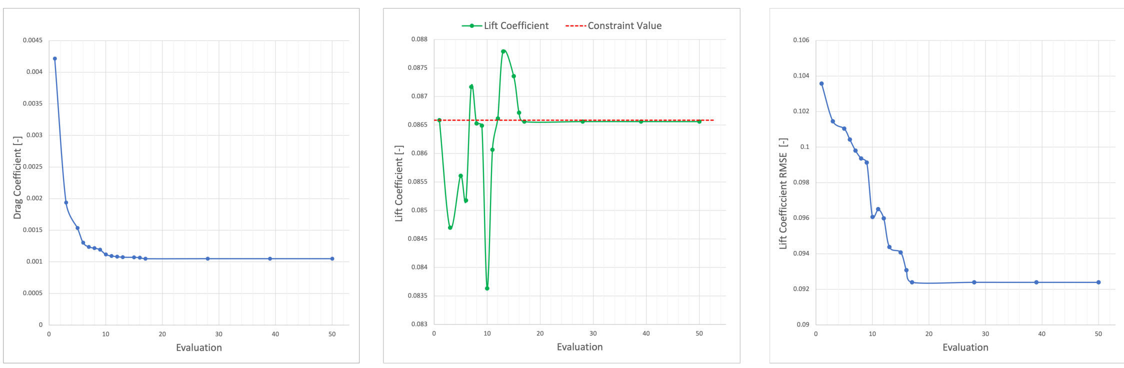


(a) Comparison between the airfoil shape for the NACA 0010 and baseline optimised case

(b) Baseline optimisation for real inflow versus optimisation for sine wave inflow

Figure 7.7: Airfoil comparison of the baseline NACA 0010 optimisation problem for a sine wave inflow condition and the baseline optimisation problem for real inflow conditions.

Figure 7.8 shows the shape optimisation history for all parameters. The drag objective quickly decreases and finds a stable value at approximately 17 evaluations into the problem. This is also the point where the lift coefficient has stabilised to the 0.03% constraint violation value, and the $C_{l_{RMSE}}$ value stabilises. However, the optimiser is not able to find more improvement to the problem for the next 45 evaluations. Therefore, the optimisation process has been stopped by hand. Even though the lift constraint has a very small violation, the optimisation is assumed to be converged.



(a) Drag coefficient

(b) Lift coefficient

(c) RMSE lift value

Figure 7.8: Optimiser performance for the baseline optimisation problem given by the convergence figures of the drag and lift coefficient. The secondary performance objective, the $C_{l_{RMSE}}$ value, is also presented in this figure.

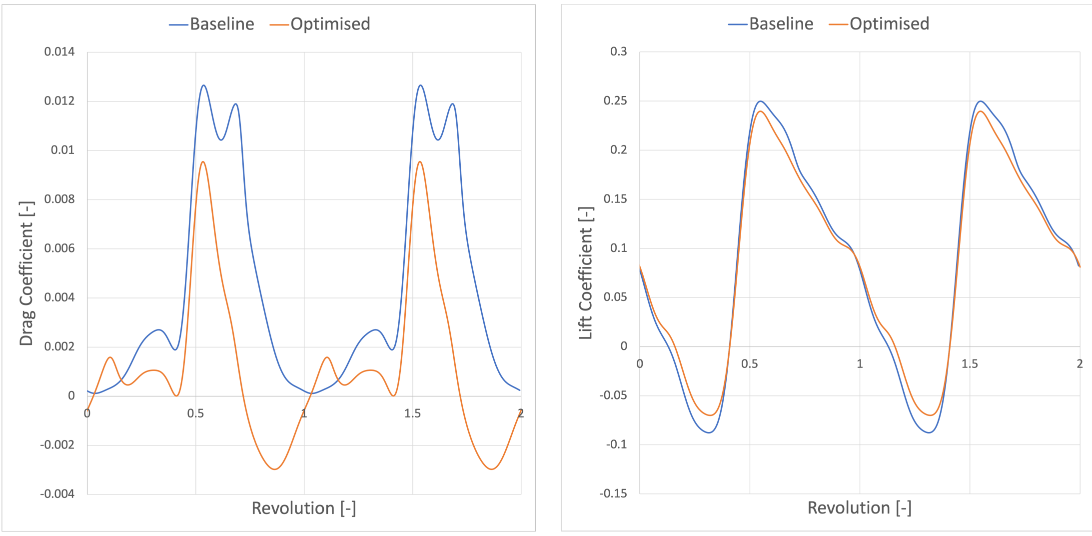
For this optimisation, the requested gradient accuracy of $1E-6$ has not been achieved, similar to the optimisation for simple inflow conditions. The lowest gradient the optimiser was able to reach was $1.744E-3$. However, the difference between the two last C_D values was $2.3E-10$. This leads to the assumption that the optimiser was not able to find a better solution and therefore the optimiser was stopped by hand. Looking at the lift, drag, and $C_{l_{RMSE}}$ value over the course of the optimisation, given in Figure 7.8, it seems that the optimiser has reached convergence. The lift constraint violation was limited to 0.03%, with an overall drag coefficient reduction of 75.1%. The maximum thickness has been overshot by a negligible amount. Interestingly the $C_{l_{RMSE}}$ value also reduced for the baseline optimisation without any specific constraints added to the configuration file. In the baseline optimisation, the optimiser has achieved a reduction of 10.8% in the $C_{l_{RMSE}}$ value. All relevant optimisation results are presented in Figure 7.9. Again a time-accurate simulation has been run with the optimised airfoil shape to verify the HB results. Similar to the simple inflow optimisation, the lift coefficient constraint and the $C_{l_{RMSE}}$ value show good agreement for both the HB and time-accurate results. The lift coefficient shows a small difference between the HB and time-accurate results. However, this is small enough to consider this an expected numerical inaccuracy. The drag coefficient shows a slightly larger deviation than the lift coefficient but not as large as for the simple inflow conditions. Indicating that this drag coefficient reduction can be assumed to be correct.

Parameter	HB results	TA results
Optimisation Accuracy [-]	2.3E-10	-
Drag coefficient reduction [%]	75.1	72.6
Lift coefficient constraint violation [%]	0.03	1.3
$C_{l_{RMSE}}$ reduction [%]	10.8	10.9
Maximum thickness [%]	10.03	-

Table 7.3: Optimisation performance for baseline optimisation scenario

Figure 7.9 shows the time-accurate solutions of both the baseline NACA 0010 airfoil and the baseline drag coefficient optimised airfoil. Interestingly the time history of the drag coefficient has changed shape substantially. The lift coefficient has remained the original shape while surprisingly having a slightly lower positive and negative peak. In turn, this leads to the reduction of the $C_{l_{RMSE}}$ value. The peak increase of the drag coefficient is still there, but it is less pronounced and reduces quicker than for the NACA 0010 airfoil. The drag coefficient has reduced more to negative values, possibly due to the sudden change in orientation of the airfoil. A leading edge downward motion can cause the drag coefficient to turn negative. The airfoil encounters leading-edge suction due to the effective upwash. This causes the pressure drag to become negative and propel the airfoil.

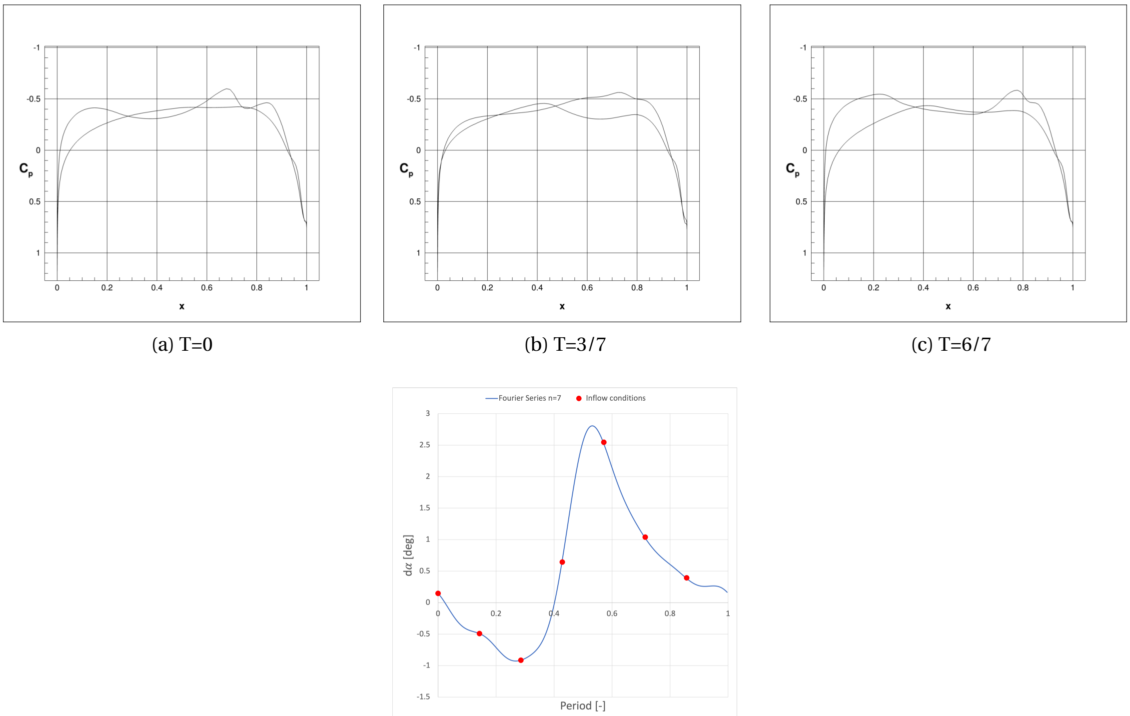
In order to see what is going on over the airfoil, the pressure distributions have been plotted in Figure 7.10. Comparing these to the NACA 0010 pressure distributions, given in Figure 6.10, the shock has disappeared. These plots also show that for Figure 7.10*a* and Figure 7.10*c*, there are two bubbles of air which are accelerated more than the rest. These bubbles are located on the forward and aft part of the airfoil. This can also be seen when looking at Figure 7.17*a* and Figure 7.17*g*. The airflow over the airfoil is relatively stable since the airfoil does not have a large pitch angle or a large change in pitch angle. When looking at Figure 7.10*b*, a bubble of possible supersonic flow can be seen on the aft part of the suction side. The airflow is supersonic locally when compared to Figure 7.17*d*. The drop in pressure coefficient on the suction side is so far aft that the pressure is not able to recover. This indicates that the dip could point to a weak shock happening over the airfoil. This point in time is the point where the airfoil experiences the largest change in the pitch direction. The large change in pitch angle could be the reason for the localised increase in velocity.



(a) Drag coefficient

(b) Lift coefficient

Figure 7.9: Lift and drag coefficient from the baseline time accurate simulation compared to the baseline optimised airfoil HB simulation



(a) T=0

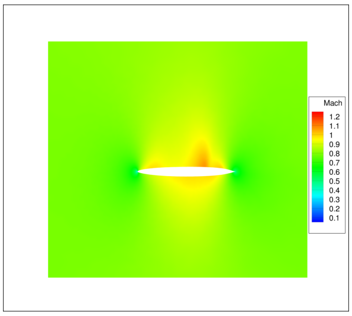
(b) T=3/7

(c) T=6/7

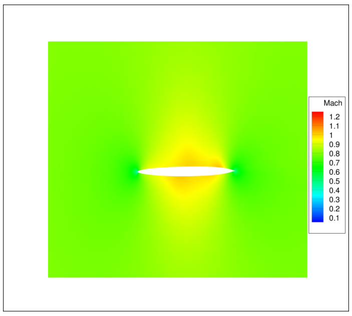
(d) Time point overview

Figure 7.10: The pressure distributions of various time instances of the optimised airfoil shape for a drag coefficient objective function in real inflow conditions

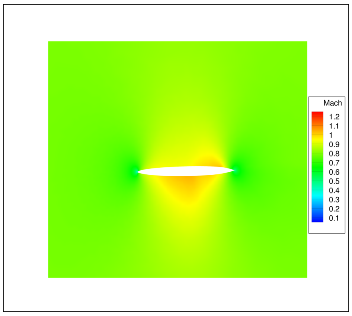
Figure 7.17 shows the collection of the contours of the Mach number. Each time instance represents a red dot in Figure 7.17*h*. When looking at the contour plots of the baseline optimisation run, an interesting change in the flow over the airfoil can be observed. For reference, the initial time instance of the baseline NACA0010 airfoil is shown in Figure 7.17*i*. The supersonic flow over the optimised airfoil has reduced substantially in intensity, and the normal shock has disappeared completely. Some supersonic flow is still present over the airfoil. However, this is substantially less compared to Figure 7.17i. This is the reason why the drag coefficient has decreased over the entire time period.



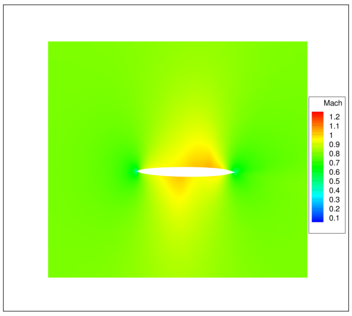
(a) T=0



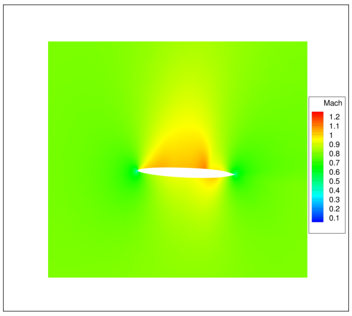
(b) T=1/7



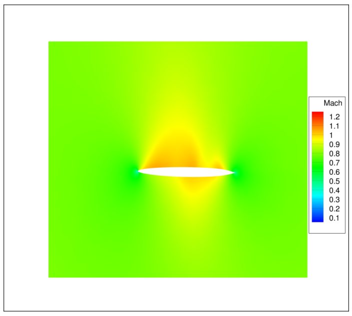
(c) T=2/7



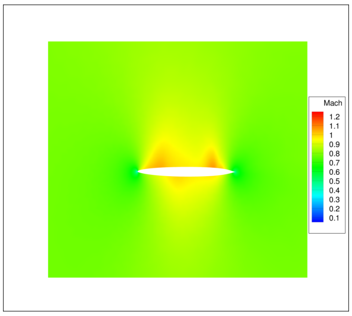
(d) T=3/7



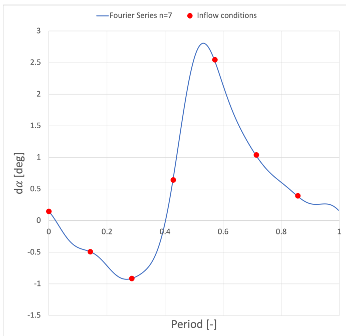
(e) T=4/7



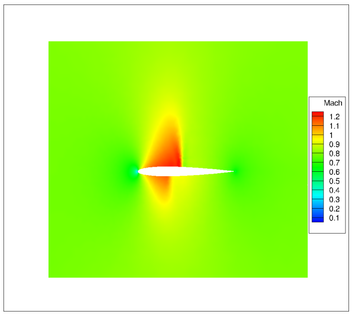
(f) T=5/7



(g) T=6/7



(h) Time point overview



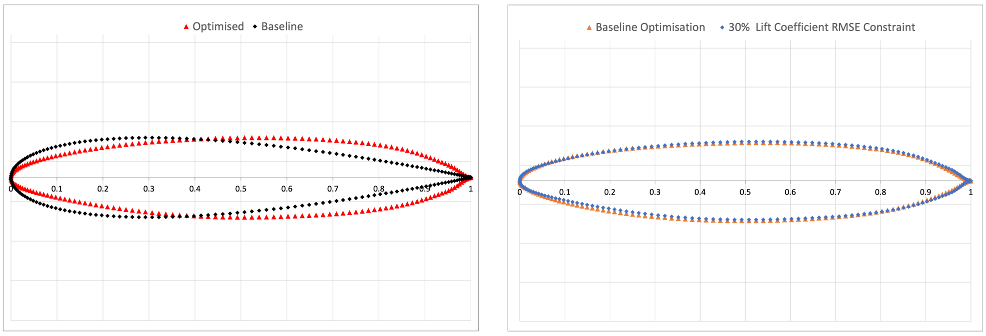
(i) NACA 0010 comparison

Figure 7.11: Mach contours of the baseline NACA 0010 optimisation for the real inflow condition. All time instances computed with the adjoint harmonic balance method are presented.

7.4. $C_{l_{RMSE}}$ constrained optimisation for real inflow

This section treats the optimisation problem that has introduced a constraint value on the $C_{l_{RMSE}}$ value for the real inflow conditions. The objective function for this optimisation is still the drag coefficient of the airfoil. By introducing a constraint violation from the beginning of the optimisation, the optimiser is forced to find a good solution for both the drag coefficient and the $C_{l_{RMSE}}$ value.

Similarly to what was seen for the baseline optimisation problem in section 7.3, the $C_{l_{RMSE}}$ constrained problem changed the airfoil leading edge to a smaller radius leading edge compared to the NACA 0010 airfoil. The aft part of the optimised airfoil shows a very similar shape when looking at Figure 7.12b. There is a very slight difference in airfoil shape, from 0.95 to 1 $\frac{x}{c}$ a slight difference can be observed in Figure 7.12b. The forward part of the airfoil has decreased slightly in thickness which means that the camber has increased slightly compared to the baseline optimisation airfoil. However, these differences are minimal, so the effects of the shape change are expected to be minimal. The thicker aft part of the airfoil could lead to more separation when the viscous effects are taken into account.

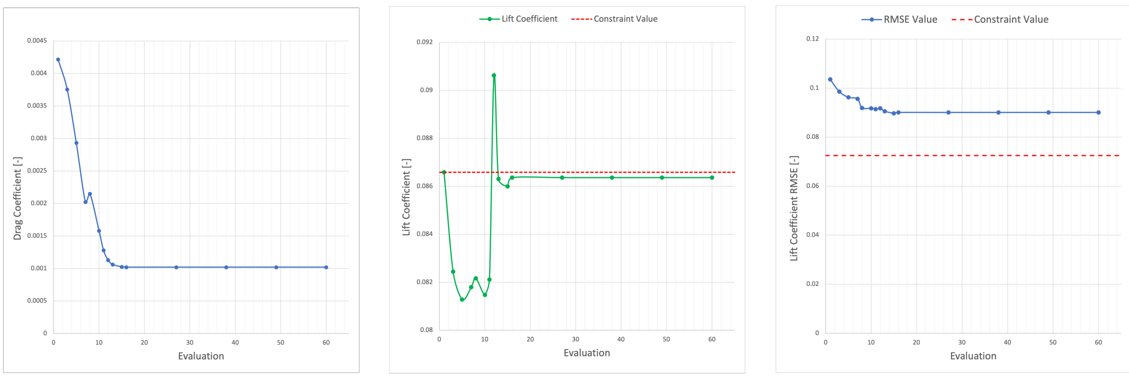


(a) NACA 0010 versus the $C_{l_{RMSE}}$ value constrained optimisation for real inflow conditions

(b) Baseline optimised airfoil versus $C_{l_{RMSE}}$ value constrained optimised airfoil for real inflow

Figure 7.12: Airfoil comparison for a constrained $C_{l_{RMSE}}$ value optimisation problem and the baseline optimisation problem for real inflow conditions

The shape optimisation history of the most relevant parameters is given in Figure 7.13. The optimiser was able to find a fairly stable solution at approximately 15 evaluations. For the rest of the time, the optimiser was not able to find any better solutions. Therefore, the optimisation was stopped by hand due to the optimiser not finding any better solutions for almost 50 evaluations. Although the $C_{l_{RMSE}}$ value constraint violation has not been achieved by almost 15 % point, the results of the optimisation still can be considered to be converged.



(a) Drag coefficient

(b) Lift coefficient

(c) $C_{l_{RMSE}}$ value

Figure 7.13: Optimiser performance for a constrained $C_{l_{RMSE}}$ value given by the convergence figures of the drag and lift coefficient. The secondary performance objective, the $C_{l_{RMSE}}$ value, is also presented in this figure.

The set gradient accuracy of $1E-6$ has not been achieved during the optimisation. The lowest gradient the optimiser was able to reach was $1.326E-3$. However, the difference between the two last C_D values was $3.49E-9$. Similar to the baseline optimisation for real inflow conditions this led to the assumption that the optimiser was not able to find a better solution and that convergence was reached. The 30% $C_{l_{RMSE}}$ value reduction as a constraint was too stringent but did reach a slightly larger $C_{l_{RMSE}}$ value reduction than the other optimisations. Again the lift coefficient violation of 0.25 % and the $C_{l_{RMSE}}$ value constraint violation would prevent the simulation from converging. The lift constraint violation was limited to 0.25%, with an overall drag coefficient reduction of 75.8%. Again the maximum thickness of the airfoil showed a negligible difference from the requested constraint value. This leads to the assumption that the optimiser would like to create an airfoil with a lower maximum thickness. The $C_{l_{RMSE}}$ value did not reach the aimed 30 % reduction.

It only achieved a 13.1 % reduction where the baseline optimisation already achieved a 10 % reduction in $C_{l_{RMSE}}$ value. This does mean that the optimiser has worked with a constant constrained violation and thus changed the airfoil shape accordingly. However, the difference in $C_{l_{RMSE}}$ value reduction is not as much as was envisioned. The fact that the constraint violation has been there could be a reason why the objective accuracy is one order of magnitude larger compared to the baseline optimisation in the same amount of run-time. All relevant optimisation results are presented in Table 7.4. A time-accurate simulation has been run with the optimised airfoil shape to verify the HB results. The $C_{l_{RMSE}}$ value shows a similar result to the HB results. The lift coefficient deviates by approximately 1.5 % point, which is still reasonable considering the average lift coefficient is compared. The drag coefficient deviation is slightly larger than the lift coefficient but can be considered within the engineering expected numerical inaccuracies.

Parameter	HB results	TA results
Optimisation Accuracy [-]	3.490E-9	
Drag coefficient reduction [%]	75.8	72.0
Lift coefficient constraint violation [%]	0.25	1.7
$C_{l_{RMSE}}$ reduction [%]	13.1	13.0
Maximum thickness [%]	10.01	-

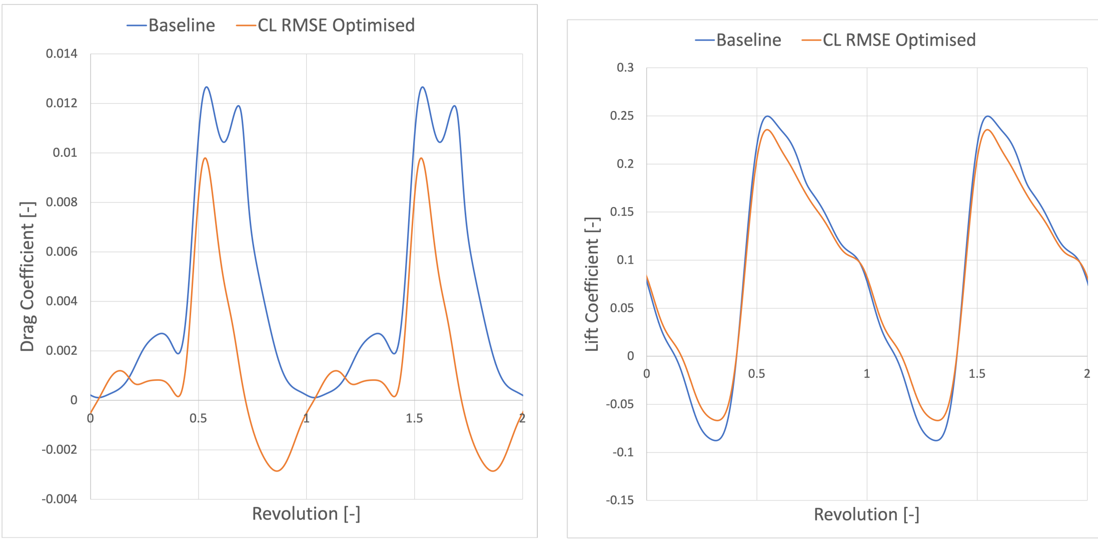
Table 7.4: Optimisation performance for $C_{l_{RMSE}}$ value constraint optimisation scenario

The time-accurate simulation given in Figure 7.14 shows the decrease in drag coefficient and $C_{l_{RMSE}}$ value, indicated in Table 7.4. The drag coefficient has been lowered by lowering the absolute peak of the drag coefficient at around 0.6 revolutions. Only from the 0.05 until 0.15 revolution point has the drag increased, while for the rest of the revolution, the drag coefficient is below the baseline NACA 0010 drag coefficient time history. The lift coefficient also shows a slight difference compared to the NACA 0010 time history. The most positive and most negative peaks show a lower value which equates to that lowered $C_{l_{RMSE}}$ value. The difference between the baseline and $C_{l_{RMSE}}$ value constrained optimisation is not large, as seen in both Table 7.4 and Figure 7.14*b*. The reason for this could be that the optimiser does not like to use the $C_{l_{RMSE}}$ value metric in the objective function. Although the gradients that were presented in the gradient validation in section 6.4 showed very good agreement between the two methods, it could be that this metric is not as effective as hoped. With the additional 2% point reduction in $C_{l_{RMSE}}$ value, the optimisation could not be categorised as a major success. However, the method did show that it is possible to lower the lift coefficient amplitude while minimising the drag coefficient.

The baseline optimised airfoil time accurate simulation has now been added to capture the full picture of $C_{l_{RMSE}}$ value constrained optimisation process. Figure 7.15 shows the three time-accurate simulations for both the drag coefficient and the lift coefficient. The drag coefficient, in Figure 7.15*a*, shows a slight difference in the first part of the revolution. As expected, the lift coefficient also shows the slightly decreased $C_{l_{RMSE}}$ value in the form of the two peaks, positive and negative, to be slightly less than the baseline optimisation.

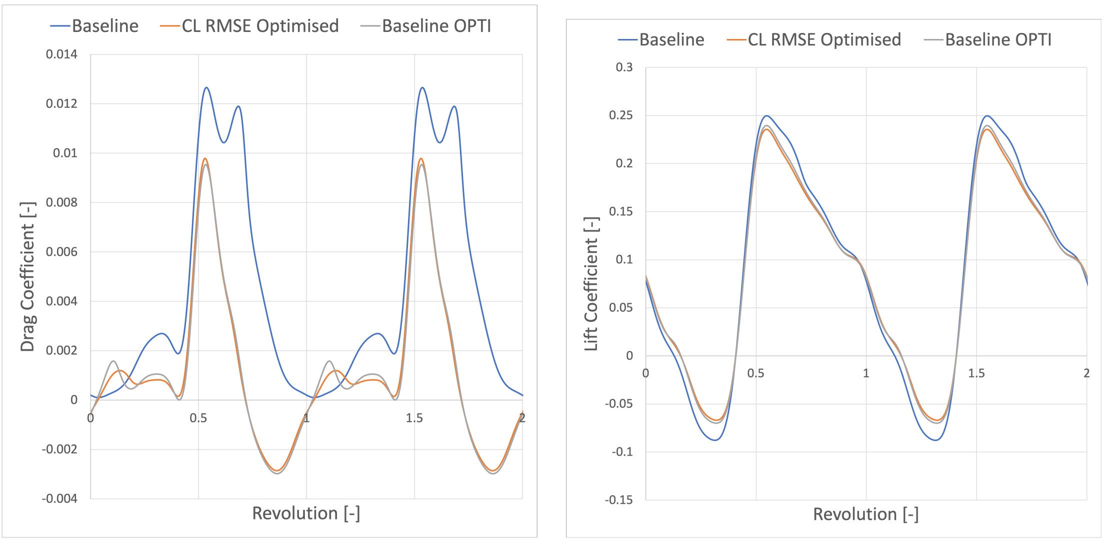
The pressure distributions of the $C_{l_{RMSE}}$ value constraint optimised airfoil are shown in Figure 7.16. Again the most evident change compared to the NACA 0010 airfoil pressure distributions, given in Figure 6.10, is that the shock has disappeared. The pressure coefficient slowly increases in negative number towards the aft part of the airfoil. On the first and last timestamp, Figure 7.16*a* and Figure 7.16*c* respectively, the suction side of the airfoil shows a slight dip in this rising behaviour. After this dip, the pressure coefficient continues to rise towards the aft part of the airfoil before the pressure coefficient is reduced again. The dip can also be seen in Figure 7.17 as the disruption of the two supersonic bubbles. At the T = 3/7 point given in Figure 7.16*b*, the suction side also shows the dip in pressure coefficient but further towards the aft part of the airfoil. The pressure coefficient is not able to return to the original value before the end of the airfoil. This could indicate a weak shock over the airfoil. However, the intensity of the shock is much lower than for the original NACA 0010 airfoil.

A comparison between the baseline optimisation and $C_{l_{RMSE}}$ value constraint optimisation is given in Figure 7.16*d* and *e*. As expected, the difference between the baseline and the $C_{l_{RMSE}}$ value constraint optimisation shape is minimal. The biggest difference can be seen in the forward part of the airfoil. The slope of the



(a) Drag coefficient (b) Lift coefficient

Figure 7.14: Comparison between the time accurate solution of the NACA 0010 airfoil and $C_{l_{RMSE}}$ value constrained optimised airfoil for the drag coefficient objective function for real inflow.

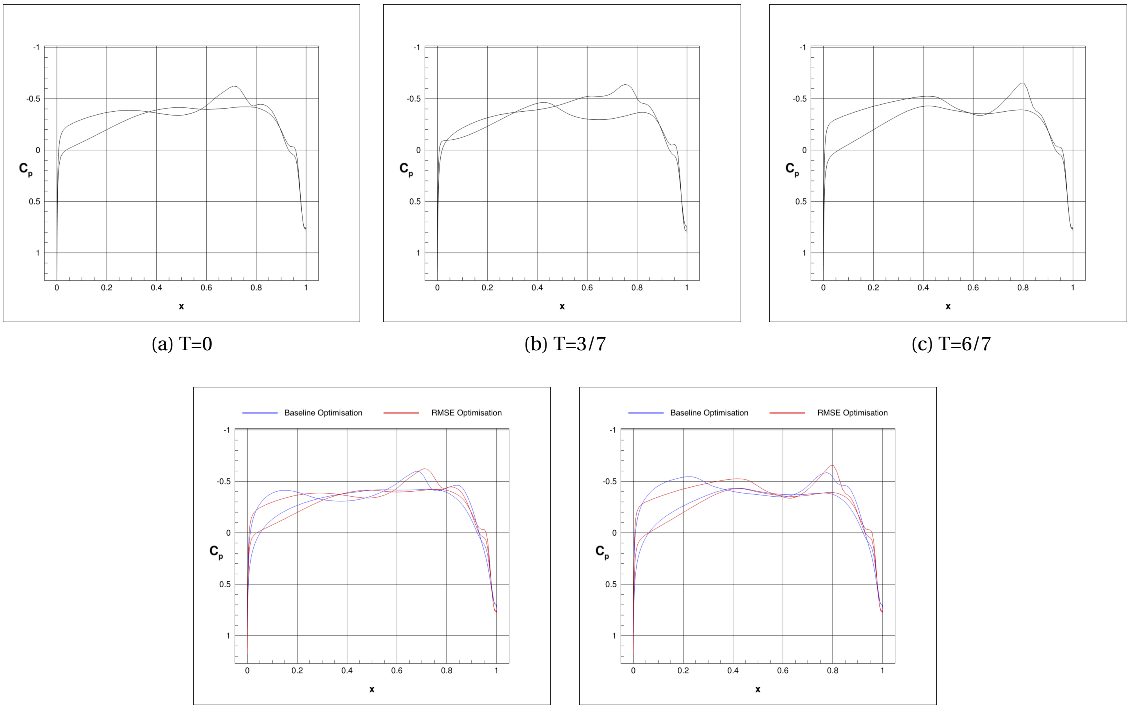


(a) Drag coefficient (b) Lift coefficient

Figure 7.15: Comparison between the time accurate solution of the NACA 0010 airfoil, baseline optimised airfoil and $C_{l_{RMSE}}$ value constrained optimised airfoil for the drag coefficient objective function in real inflow conditions

increase in pressure coefficient has gotten more gradual for the $C_{l_{RMSE}}$ value constraint shape. This would indicate that the forward part of the airfoil has a less severe reaction in terms of pressure to the change in inflow conditions. However, this is not observed as much in the $C_{l_{RMSE}}$ value.

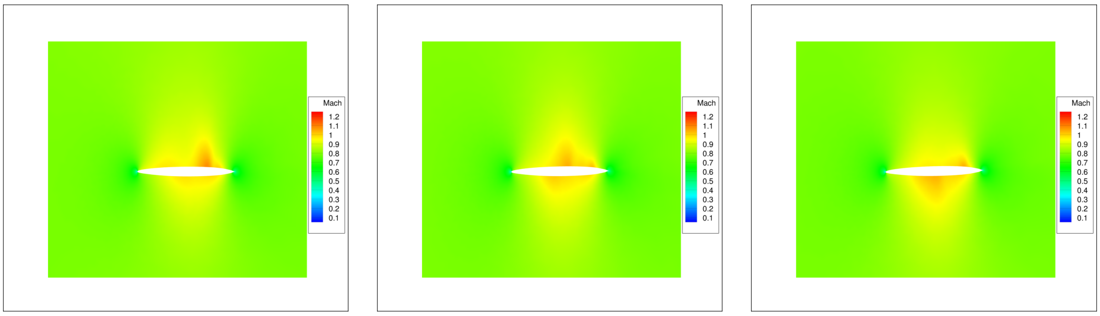
The contours of the Mach number of the $C_{l_{RMSE}}$ constrained optimisation are presented in Figure 7.17. In this figure, the contours of the Mach numbers at each time instance represent a red dot in Figure 7.17*h*. There are some similarities to the Mach contours for the $C_{l_{RMSE}}$ value optimisation and the baseline optimisation, as shown in section 7.3. The strong normal shock over the suction side of the airfoil has disappeared and made a place for multiple small bubbles of supersonic flow. There is only a very slight difference between the $C_{l_{RMSE}}$ value constrained and baseline optimisation. Upon very close inspection, the supersonic bubble on the forward part of the airfoil shows a slightly less steep angle upwards. This is confirmed when looking



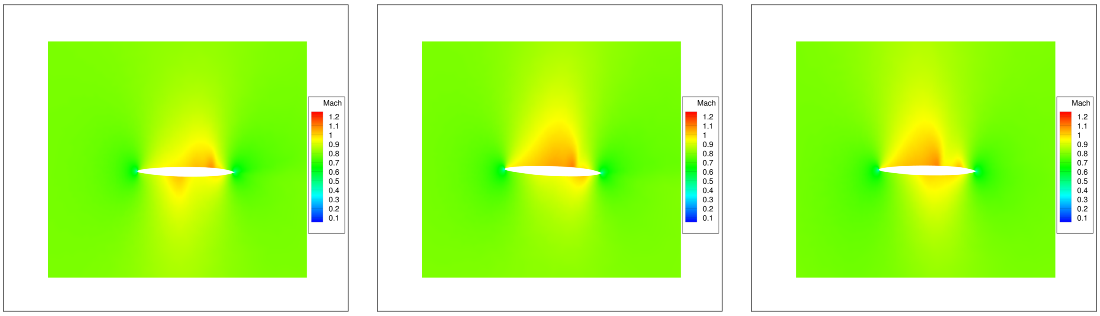
(a) T=0
(b) T=3/7
(c) T=6/7
(d) Comparison T=0
(e) Comparison T=6/7

Figure 7.16: The pressure distributions of various time instances of the optimised airfoil shape for a drag coefficient objective function in real inflow conditions including RMSE constraint

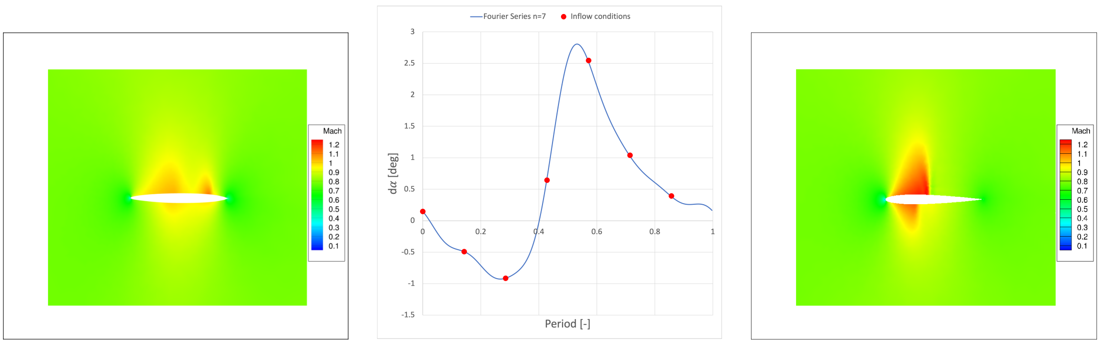
back at Figure 7.16, where the biggest difference between the two optimisations lies on the forward part of the airfoil. The rest of the plots show a very similar picture as the baseline optimisation. This is expected due to the very little change in the airfoil shape.



(a) T=0 (b) T=1/7 (c) T=2/7



(d) T=3/7 (e) T=4/7 (f) T=5/7



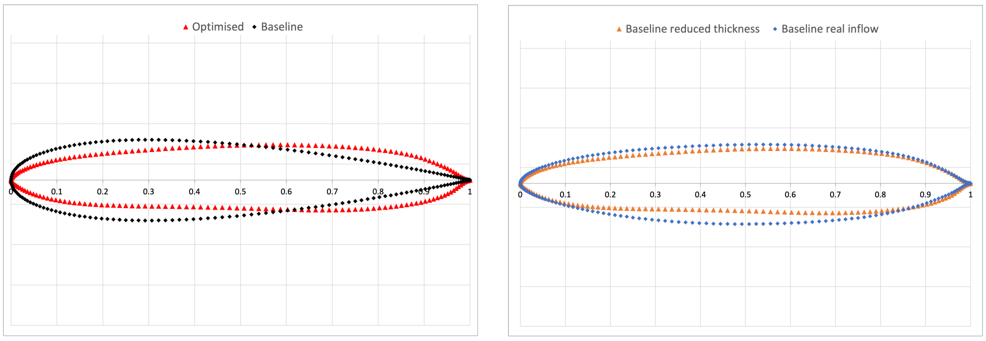
(g) T=6/7 (h) Time point overview (i) NACA 0010 comparison

Figure 7.17: Mach contours of the NACA 0010 optimisation with a 30% reduced $C_{l_{RMSE}}$ value constraint for the real inflow condition. All time instances computed with the adjoint harmonic balance method are presented.

7.5. Alternative thickness constraint

During the optimisation runs the results showed that the thickness of the airfoil would constantly approach the thickness constraint. Based on these results another optimisation run was performed using a different constraint value for the thickness of the airfoil. This optimisation run is performed to check whether the optimiser would also approach the thickness constraint if it were set to only 5 % of the chord length.

As expected, the thickness of the airfoil has decreased substantially due to this constraint change. As seen in Figure 7.18*a*, the airfoil has once again converged to a smaller radius leading edge compared to the NACA 0010 airfoil, as well as the increased thickness in the aft part of the airfoil. Compared to the airfoil shape as a result of the baseline optimisation, the airfoil has a reduced thickness from 0.1 to 0.8 $\frac{x}{c}$. The trailing edge has remained the same shape.

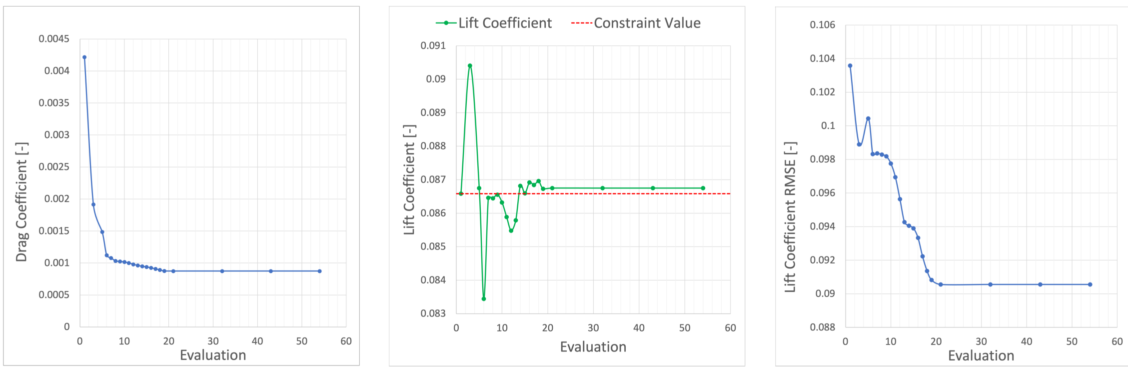


(a) NACA 0010 versus the $C_{l_{RMSE}}$ value constrained optimisation for real inflow conditions

(b) Baseline optimised airfoil versus $C_{l_{RMSE}}$ value constrained optimised airfoil for real inflow

Figure 7.18: Airfoil comparison for alternative thickness constraint optimisation problem and the baseline optimisation problem for real inflow conditions

The history of the optimiser's most important parameters is given in Figure 7.19. The optimiser showed the same behaviour as the baseline optimisation, with a quick convergence after approximately 18 evaluations. The lift coefficient initially showed some instability. However, this is related to the optimiser finding the optimal thickness of the airfoil now that the thickness constraint is less stringent. Same as for all optimisation runs, the lift coefficient has been overshot by a small margin, and this prevented the optimisation from recognising the convergence. The optimiser has been stopped by hand due to the inability to improve the shape in over 30 evaluations. Therefore, the optimisation can be assumed to be converged.



(a) Drag coefficient

(b) Lift coefficient

(c) $C_{l_{RMSE}}$ value

Figure 7.19: Optimiser performance for a constrained $C_{l_{RMSE}}$ value given by the convergence figures of the drag and lift coefficient. The secondary performance objective, the $C_{l_{RMSE}}$ value, is also presented in this figure.

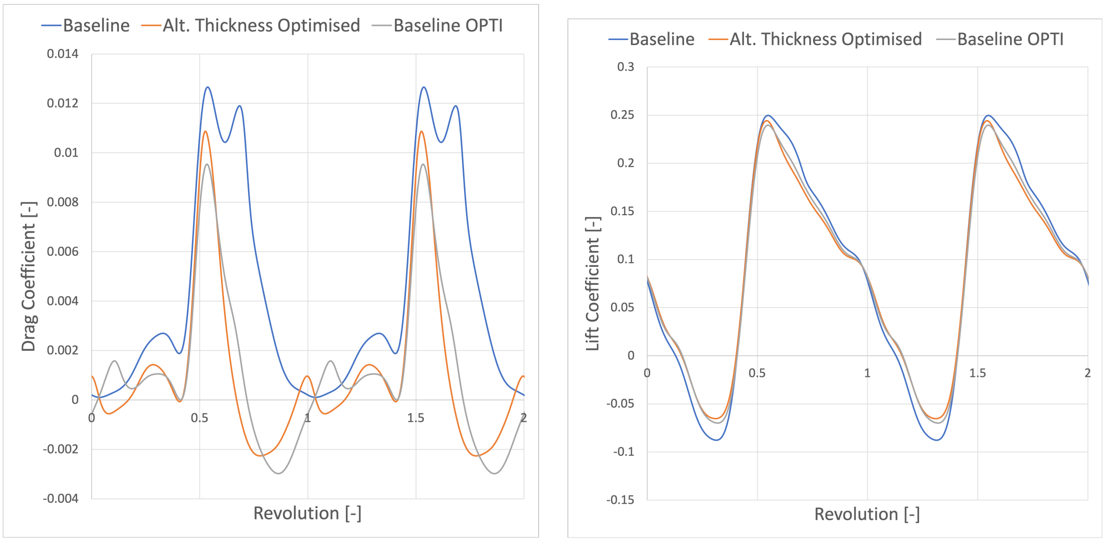
All optimisation parameters are given in Table 7.5. As mentioned above, the optimisation has been stopped by hand due to the optimiser failing to find a better airfoil shape. The set gradient accuracy of $1E-6$ has not been achieved during the optimisation. The lowest gradient the optimiser was able to reach was $9.512E-4$. However, the difference between the two last C_D values was $4.07E-10$. Similar to the baseline optimisation for real inflow conditions this led to the assumption that the optimiser was not able to find a better solution and that convergence was reached convergence in the process. The drag coefficient has reduced by 79.3%, which is a larger reduction than that of the baseline optimisation problem. The increased reduction can be attributed to the lowered thickness of the airfoil. The lowered thickness of the airfoil also reduces the risk of supersonic flow due to the increase in thickness being more gradual than the baseline optimisation airfoil. The lift coefficient has been violated by only 0.19%. Remarkably, the $C_{l_{RMSE}}$ value also decreased more compared to the baseline optimisation problem. A 12.5% reduction of forced lift response has been observed for this alternative thickness optimisation problem. The thickness has not approached the constraint value of 5% of the chord, and it stayed at around 8% thickness. Same as for the other optimisation runs, a time-accurate simulation has been run with the optimised airfoil shape to verify the HB results. Both the lift coefficient and the $C_{l_{RMSE}}$ value show very good agreement. Again the drag coefficient results show a slight difference between the HB and time-accurate results. Approximately a 5% point difference can be observed for the drag coefficient reduction. This can be caused by the drag coefficient changing substantially. Having 7 data points to determine the average drag coefficient with a complex drag coefficient time history, as seen in Figure 7.20, can lead to inaccuracies.

Parameter	HB results	TA results
Optimisation Accuracy [-]	4.07E-10	-
Drag coefficient reduction [%]	79.3	74.0
Lift coefficient constraint violation [%]	0.19	0.8
$C_{l_{RMSE}}$ reduction [%]	12.5	12.5
Maximum thickness [%]	8	-

Table 7.5: Optimisation performance for alternative thickness optimisation scenario

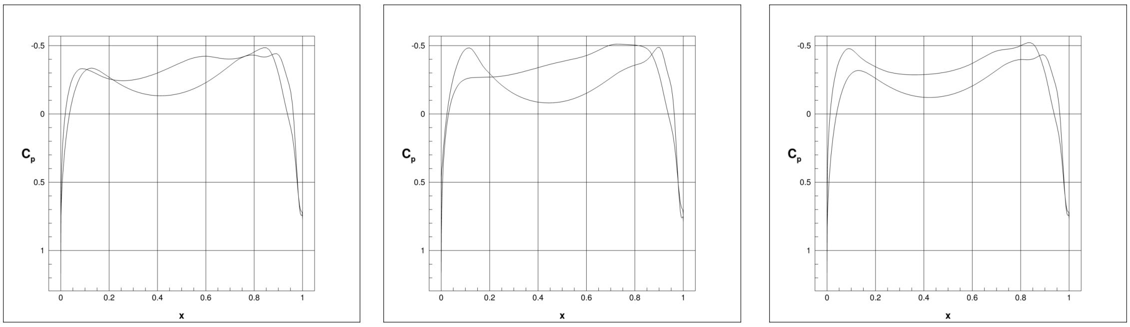
The picture presented in Table 7.5 is not as clear in the time-accurate simulation solution. The extra $C_{l_{RMSE}}$ value constraint is not as clearly visible on the time-accurate simulation as expected. The lift coefficient, shown in Figure 7.20*b*, indicates that the lift coefficient negative peak is closer to the average. However, the positive peak is further away from the average. It seems that the lift coefficient time history has been shifted a bit upwards. Even though the $C_{l_{RMSE}}$ value metric says the value is lower compared to the baseline optimisation, the time-accurate solution would suggest that this is not the case. A possible explanation is that the $C_{l_{RMSE}}$ value is based on only 7 data points. Therefore, a wrong picture can be given by only that value. For this reason, the time-accurate simulations of the optimised airfoils are given as a supplement. The drag coefficient also changed substantially over the course of one revolution. The peak of the drag coefficient is higher compared to the baseline optimisation airfoil. Again this is not the picture that was expected when looking at the drag reduction given in Table 7.5.

Now looking at the pressure distributions of the alternative thickness optimisation airfoil, given in Figure 7.21. The most notable thing is that the pressure side of the airfoil has two points of higher negative pressure, the leading edge and the trailing edge. This can also be seen in Figure 7.22, where the pressure side of the airfoil has two of these small supersonic bubbles for time instances T = 1/7 through T = 3/7. The comparison between the pressure coefficient plots also illustrates this behaviour. The baseline optimisation airfoil shows a much more adverse pressure coefficient on the pressure side. The pressure distributions would indicate a different outcome than what is seen in Table 7.5.

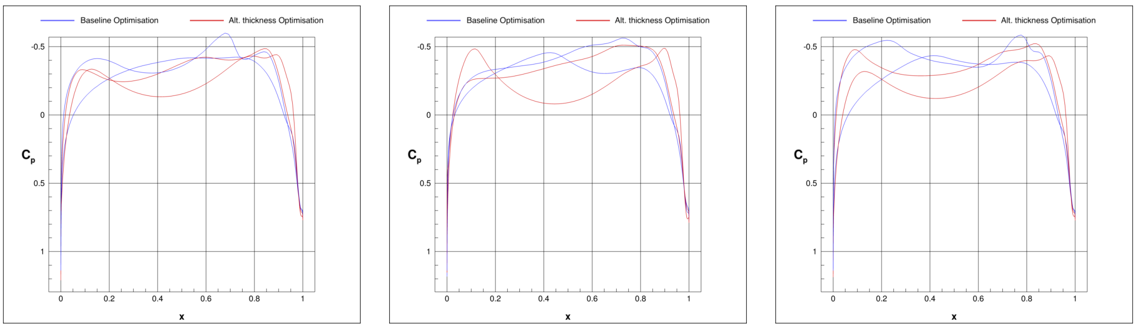


(a) Drag coefficient (b) Lift coefficient

Figure 7.20: Comparison between the time accurate solution of the NACA 0010 airfoil, baseline optimised airfoil and alternative thickness constraint optimised airfoil for the drag coefficient objective function in real inflow conditions



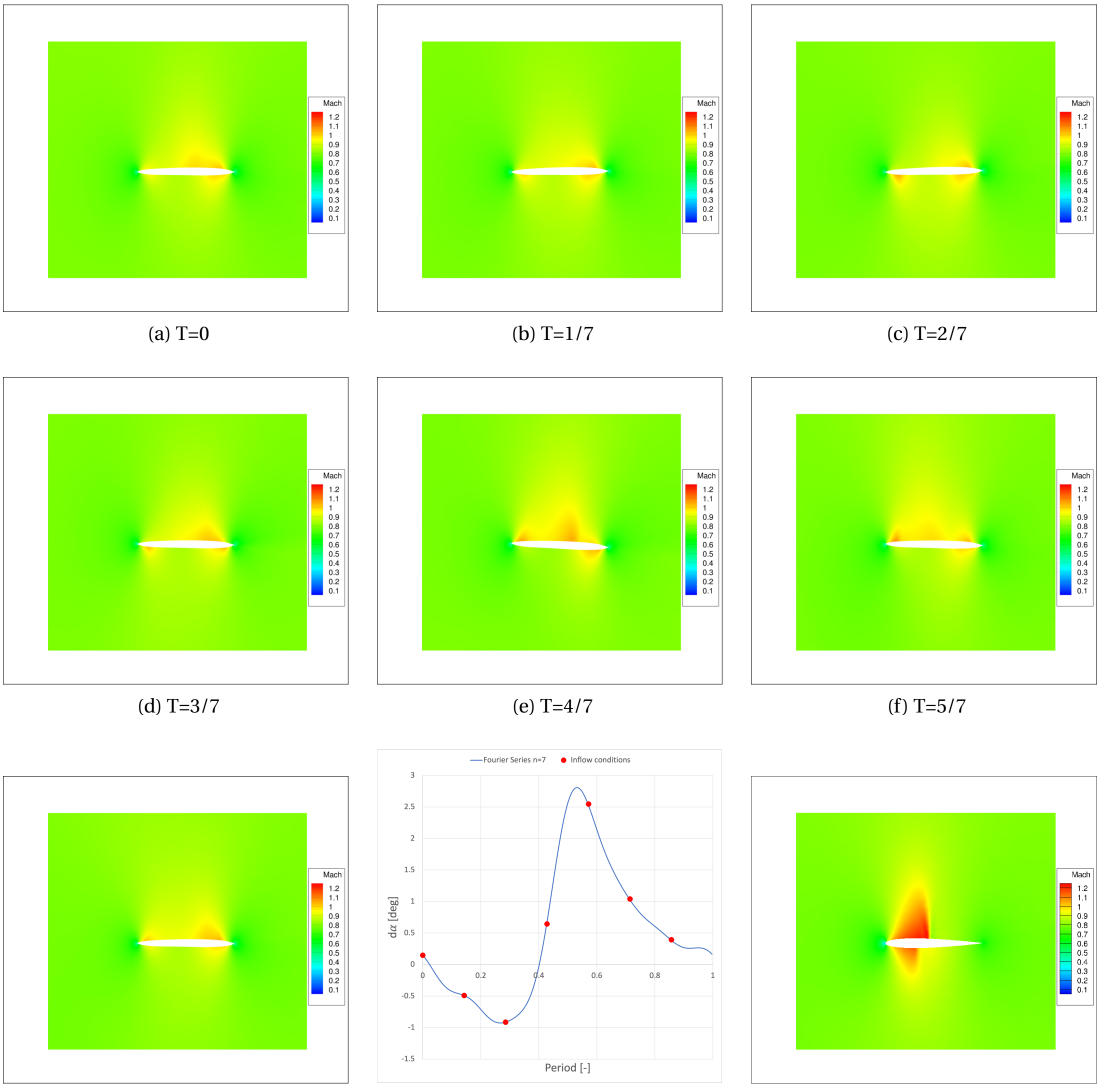
(a) T=0 (b) T=3/7 (c) T=6/7



(d) Comparison T=0 (e) Comparison T=3/7 (f) Comparison T=6/7

Figure 7.21: The pressure distributions of various time instances of the optimised airfoil shape for a drag coefficient objective function in real inflow conditions for an alternative thickness constraint

The contours of the Mach numbers are shown in Figure 7.22. Clearly, the normal shock has disappeared compared to the NACA 0010 airfoil, as is the case for all optimised airfoils. As mentioned above, for time instance, T = 1/7 through T = 3/7, the airfoil shows two small supersonic bubbles on the pressure side at either end of the airfoil. It seems that these supersonic bubbles have decreased in size but increased in intensity when compared to the baseline optimisation airfoil.



(a) T=0 (b) T=1/7 (c) T=2/7

(d) T=3/7 (e) T=4/7 (f) T=5/7

(g) T=6/7 (h) Time point overview (i) NACA 0010 comparison

Figure 7.22: Mach contours of the NACA 0010 optimisation with an alternative thickness constraint for the real inflow condition. All time instances computed with the adjoint harmonic balance method are presented.

7.6. $C_{l_{RMSE}}$ objective function optimisation for real inflow

For this optimisation run, some tweaking of the parameters was necessary. The optimiser has proven to be very sensitive to the $C_{l_{RMSE}}$ value as the objective function. All parameters that have been changed are related to the values set of the various constraints and the objective function. First, due to an unknown bug, the objective weight of the objective function had to be set to a negative value. The SU2 solver would recognise the $C_{l_{RMSE}}$ value as the correct, but negative values were shown in the log files. It seems like some conversion was done with the $C_{l_{RMSE}}$ value metric as objective. The source of the problem was not the calculation of the $C_{l_{RMSE}}$ value but rather the steps that were taken by the SU2 source code thereafter. This is possible due to the objective function being calculated in a different function in the SU2 source code than where the constraints are calculated. The source of this problem could not be found. Therefore, a workaround was used, making the objective weight of the $C_{l_{RMSE}}$ value negative. This solved the problem of the optimiser wanting to push the airfoil to non-physical shapes. Initially, the optimiser would invert the airfoil shape from as early as iteration 4 while using the same constraints as for the other optimisation runs. Figure 7.23 is an example of one of these non-physical airfoil shapes.

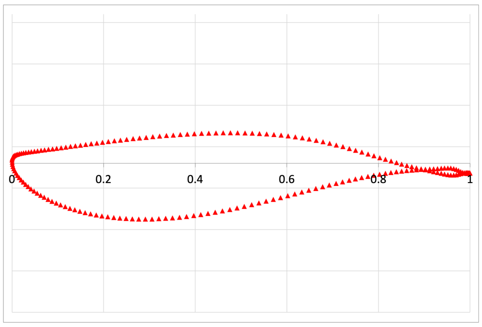
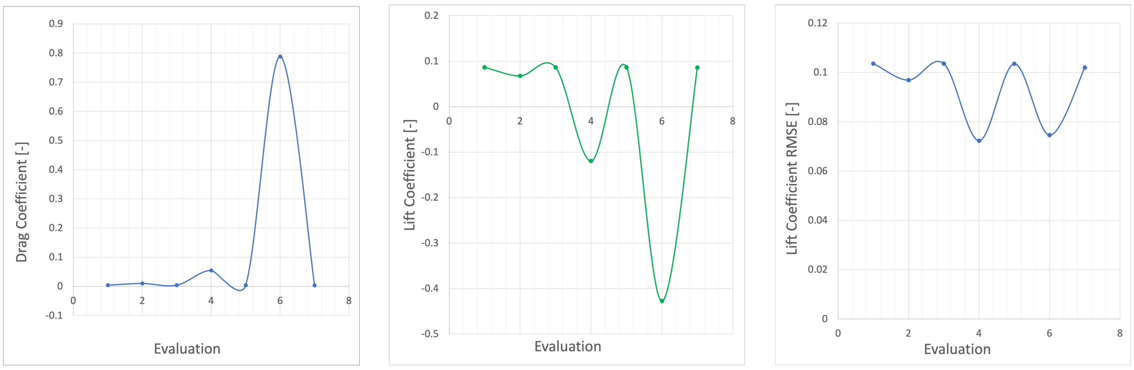


Figure 7.23: Airfoil shape after 4 iterations of the $C_{l_{RMSE}}$ objective

The next step was to get the optimisation to run smoothly. Referring back to the gradient validation in section 6.4, the gradients for this objective function showed a good match. This would indicate that the use of the $C_{l_{RMSE}}$ value as the objective function would be possible. In order to try to get any sort of convergence, many different scaling factors for the constraints and objective function have been tried. Furthermore, the lift coefficient has been changed from an equality constraint to an inequality constraint. All optimisation runs would diverge between evaluations 4 and 8. The shape of the airfoil of the converged evaluation before the divergence would not be very different from the NACA 0010 airfoil, as would be expected at the start of the optimisation problem. As seen in Figure 7.24, the optimiser does not know where to go with the optimisation and diverges after 7 evaluations. It is very unfortunate that the optimisation with the $C_{l_{RMSE}}$ value as the objective function failed. This could have been a very interesting finding. The limited reduction of the baseline optimisation and $C_{l_{RMSE}}$ value constrained optimisation gave some insight into the problem. However, not enough to fully say something about the subject. Further research into directly minimising the $C_{l_{RMSE}}$ value by using it as the objective function has not been pursued.



(a) Drag coefficient (b) Lift coefficient (c) $C_{l_{RMSE}}$ value

Figure 7.24: Optimiser performance for an optimisation problem using the $C_{l_{RMSE}}$ value as objective function

8

Conclusions and Recommendations

8.1. Conclusions

The main objective of this research was to find and create an aerodynamic optimisation method for a 2D propeller airfoil in unsteady inflow conditions. During the optimisation, the aim was to reduce the drag for a given lift coefficient and the forced response of the unsteady inflow conditions. Using a real test case in the form of the APPU project, the inflow conditions were recreated and implemented in SU2. For this research, an HB solver introduced by Rubino et al. [45] was used to increase the efficiency of the optimisation method compared to an ordinary time-accurate finite differences method.

The harmonic balance method used for these optimisation runs has proven to be the right choice regarding computation speed. The computational time difference was significant and clearly visible in the gradient validation, which is considered a small gradient computation problem of 10 control points. The adjoint method could compute the gradients faster by approximately a factor of 5. This, combined with the time-efficient harmonic balance method, which has been extensively tested by Rubino et al., leads to fast convergence.

The lack of viscous flows created inaccuracies in the optimisation method for the lift coefficient. The lift coefficient is slightly overestimated due to the Euler solver not simulating the boundary layer and the separation of the airflow. The lift fluctuations also change when viscosity is considered in the simulation. However, the intention of this research is to see whether the method is possible to use such a metric in the optimisation problem. Reducing the $C_{l_{RMSE}}$ value by 10-15% during the various optimisation runs can be considered a mediocre success for this research. On the other hand, in other research, the $C_{l_{RMSE}}$ value increased by a factor of two after the optimisation. Therefore, to be able to reduce this value and still optimise for drag is already a form of gain. However, some more explicit evidence of the airfoil being able to reduce this value would be nice to have. Unfortunately, the optimiser could not complete a run for the $C_{l_{RMSE}}$ value objective even though the gradients of the Finite differences and adjoint method showed a good match. The airfoil converged to a similar leading edge shape for both the real inflow and the simple sinus wave inflow. The trailing edge of the optimised airfoils differs in terms of the thickness of the aft part of the airfoil. The more gradual slope running from the leading edge towards the trailing edge of the airfoil leads to less acceleration of the air. In turn, this leads to less pronounced normal shocks over the suction side of the airfoil. The change in normal shock intensity is seen for all optimisation runs. It seems that the airfoil has become symmetrical on both the x and y-axis.

The simple sine inflow conditions optimisation has shown that the airfoil would take a similar shape as for the real inflow conditions. The most notable difference between the two inflow conditions was the aft part of the airfoil. The optimiser would increase the thickness of the airfoil substantially for the real inflow conditions. This indicates that a thicker aft part of the airfoil is able to cope with more extreme inflow condition changes. Keeping the lift coefficient constant while decreasing the drag coefficient and $C_{l_{RMSE}}$ value at the same time.

The reduction of the $C_{l_{RMSE}}$ value could be attributed to the change in the shape of the forward part of the airfoil. This conclusion comes from the comparison of the baseline optimisation and the $C_{l_{RMSE}}$ constrained optimisation. Therefore, the assumption is that the optimisation with the $C_{l_{RMSE}}$ value as objective would have influenced the airfoil's leading edge and forward part. The trailing edge of all optimised airfoils was almost identical, even the airfoil created with a less stringent maximum thickness requirement.

The thickness of the airfoil showed to be a good method for improving the performance of the airfoil. The drag coefficient reduced 4% point more than the baseline optimisation while the $C_{l_{RMSE}}$ value reduced 2% point more than the baseline optimisation. This could mean that thinner propeller blades would be the way to go. However, since no stress calculations are performed in this research, no information can be presented about whether this would be possible for this case.

The reliability and robustness objective which was set out to be attained with this optimisation method has not been achieved. The SU2 CFD solver needed a lot of time and effort to get it to run a single optimisation run. Various functions needed to be changed; still, some optimisations would prove to be very difficult to run. The scaling factors of all constraints were highly sensitive. Small changes would lead to the optimiser deforming the airfoil into non-physically possible shapes, and consequently, the flow solver would diverge. Sometimes, the objective weights have been set to negative to solve a further issue. So, either more time must be invested in getting more robust results out of the SU2 software, or a different software package needs to be used. On the other hand, reliability has been achieved using different inflow conditions, the complex real inflow and the simple sinus unsteady inflow. For both these inflow conditions, the optimiser was able to find an airfoil shape which converged to a which was good enough. Although the convergence exit was not reached for all optimisations, the solvers could find an optimal solution.

To conclude the research:

- It is possible to limit the forced lift response while also optimising for the $\frac{C_l}{C_d}$ performance of a 2D airfoil;
- The choice for an Euler solver to increase the computational efficiency led to large inaccuracies due to the lack of viscous flow;
- The adjoint method has helped in lowering the overall computational time of the optimisation. Using a finite differences method would have substantially increased the computational time;
- The forced response quantified as the lift coefficient RMSE value can be reduced up to 13 %, which is a decrease;
- The optimiser has shown, by using different inflow conditions, that the reduction in normal shock intensity over the airfoil can be linked to the reduction in lift response;
- Reliability and robustness of the optimiser can be better. Small changes in the setup of the optimiser can already lead to divergence. This is not desired when off-design conditions are to be simulated.
- The optimiser can work with various inflow conditions and converges to a solution under these conditions.

8.2. Recommendations

In this study, several assumptions have been made to make the study possible in the given time frame. Due to the assumptions made, further study could be conducted to complement this research. The recommendations for further study are given below:

- In some cases of mesh deformation, the adjoint flow solver residuals would stagnate and experience difficulties converging to the requested accuracy. Using a structured mesh, higher accuracy could be achieved in the computation of gradients.
- URANS simulations can be introduced to refine the results and have a drag coefficient resembling the real conditions. The HB adjoint solver has the capacity to run a more complex CFD simulation in a reasonable computational time frame.

- Checking for alternative options in terms of software to run a Harmonic Balance solver for an optimisation problem. The current software package has proven to be error-prone and lacks reliability.
- To further analyse the full propeller blade, multiple airfoil sections must be optimised for and combined into a full propeller blade. The method proposed for this would be to divide the propeller into several sections. These optimised airfoil sections can be used to determine the rest of the airfoil shape by means of interpolation. The full planform can be optimised using a simplified model such as BEM.
- This full blade can then be used to run a full 3D simulation to check for additional 3D effects.
- Noise calculations can be added to the research and generate a detailed view of the noise emissions sensitivity to the airfoil shape. An estimate of the noise emissions of a certain airfoil can be made using a linear approach to the thickness and loading noise. The linear form of the Ffowcs Williams-Hawkings equation could be an option in this case.

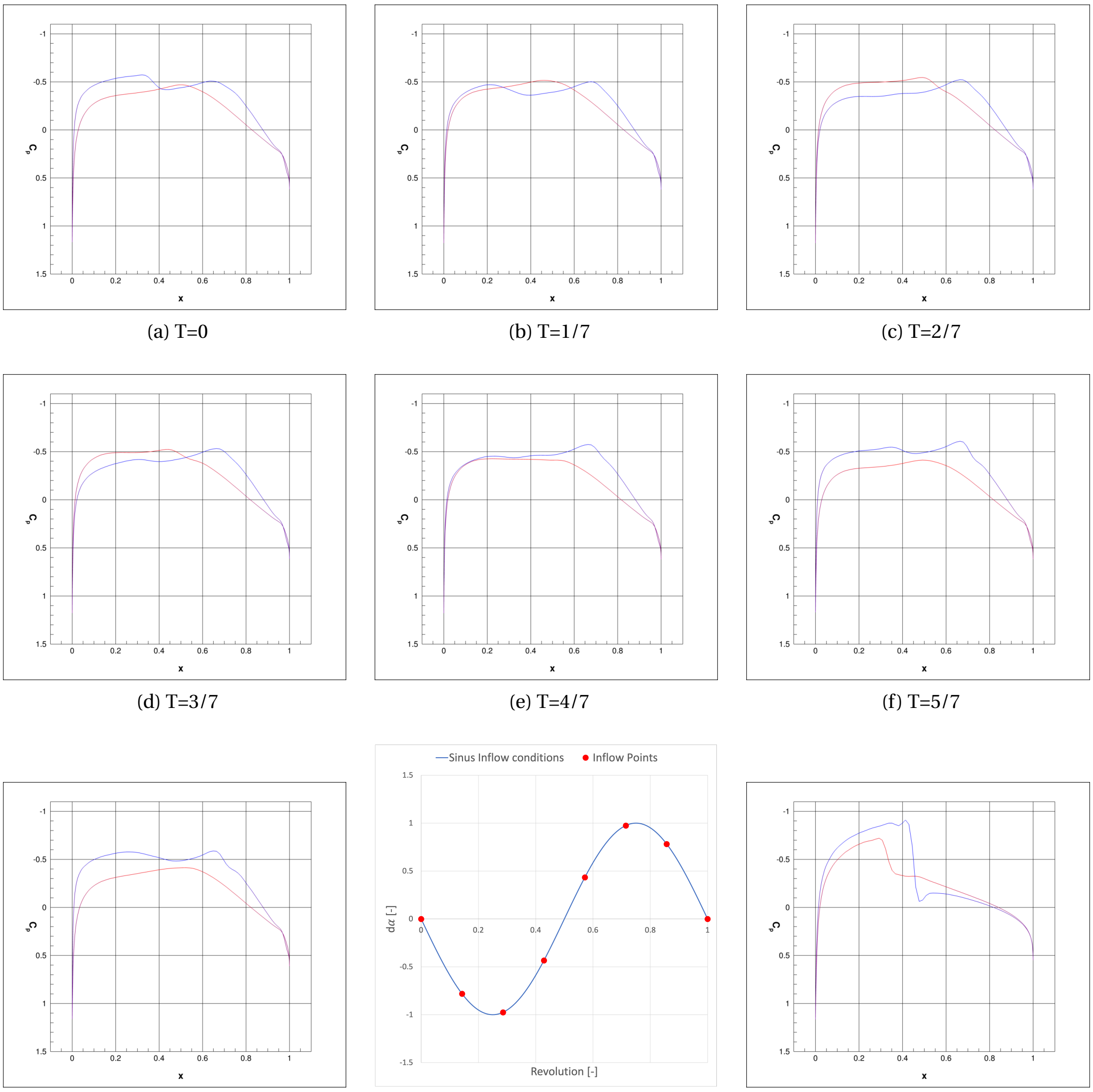
A
GitHub

As part of this research the optimisation software that has been used can be found in a GitHub repository. The steps to install SU2 can be found on the website. The source code that is up on GitHub still needs to be compiled since GitHub does not allow files of over 100 MB to be uploaded to their servers. The link for the repository can be found below

`https://github.com/maxvansplunteren/HB_RMSE_SU2_MOD`

B

Pressure Distributions

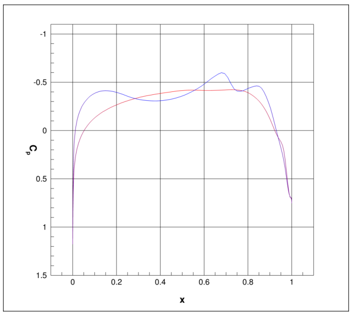


(a) T=0 (b) T=1/7 (c) T=2/7

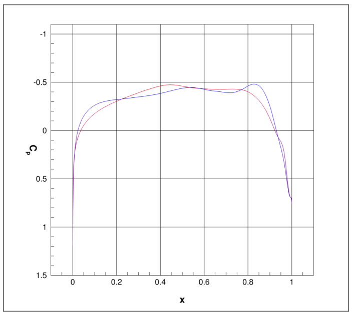
(d) T=3/7 (e) T=4/7 (f) T=5/7

(g) T=6/7 (h) Time point overview (i) NACA 0010 T=0 comparison

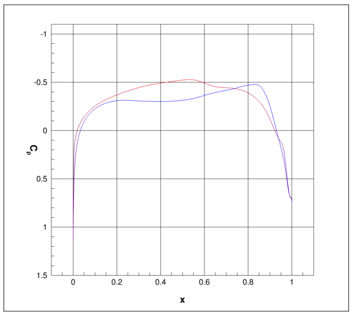
Figure B.1: Pressure distributions of the NACA 0010 baseline optimisation for the simple sinusoid inflow condition. All time instances computed with the adjoint harmonic balance method are presented. The suction side is represented by the blue line the pressure side by the red line.



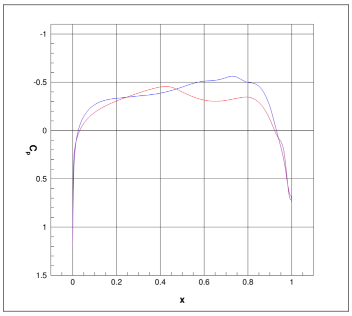
(a) T=0



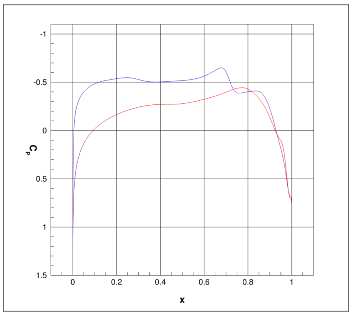
(b) T=1/7



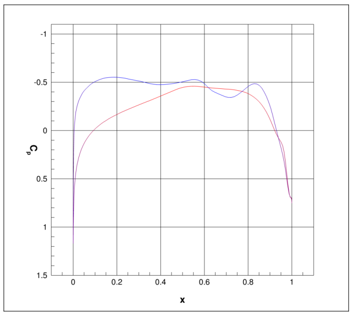
(c) T=2/7



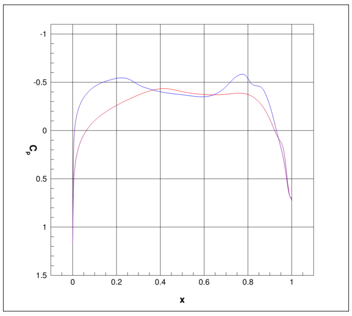
(d) T=3/7



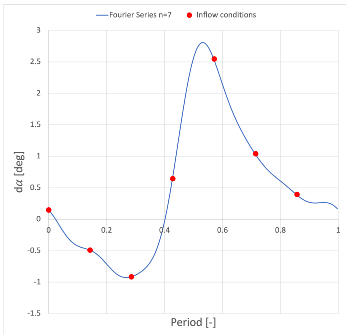
(e) T=4/7



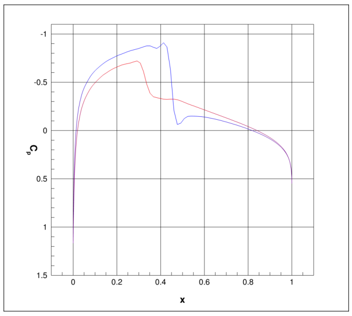
(f) T=5/7



(g) T=6/7

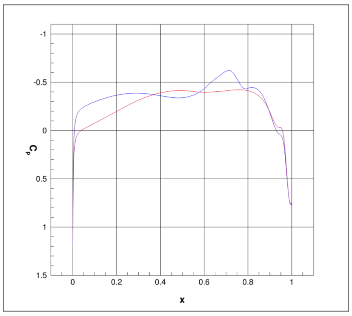


(h) Time point overview

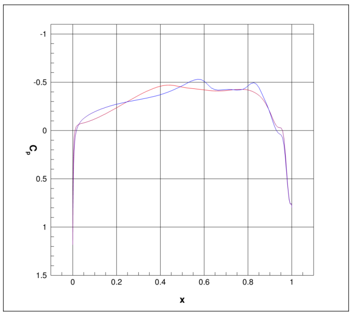


(i) NACA 0010 T=0 comparison

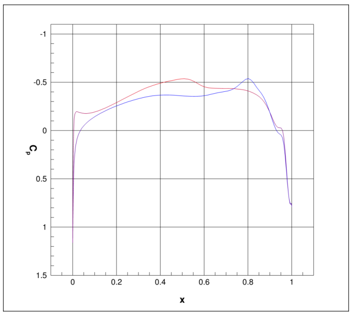
Figure B.2: Pressure distributions of the NACA 0010 baseline optimisation for the real inflow condition. All time instances computed with the adjoint harmonic balance method are presented. The suction side is represented by the blue line the pressure side by the red line.



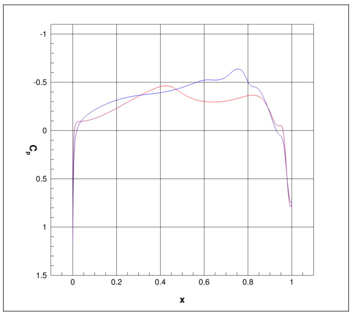
(a) T=0



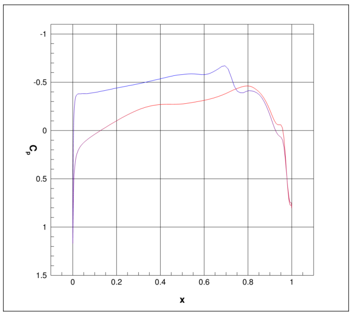
(b) T=1/7



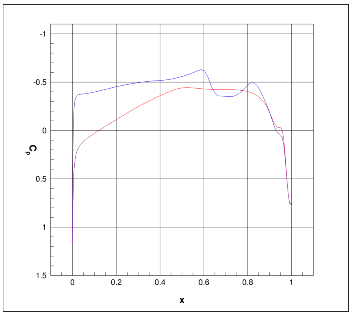
(c) T=2/7



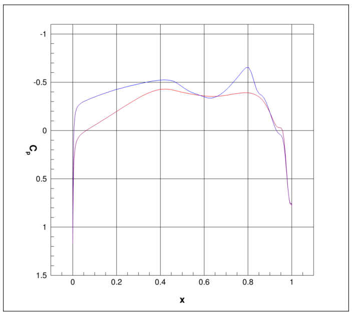
(d) T=3/7



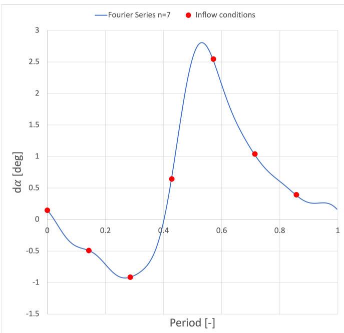
(e) T=4/7



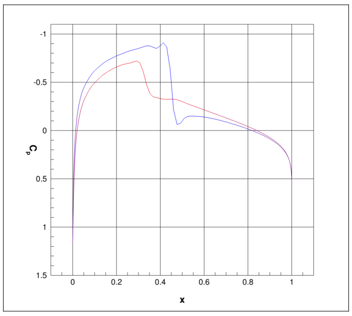
(f) T=5/7



(g) T=6/7

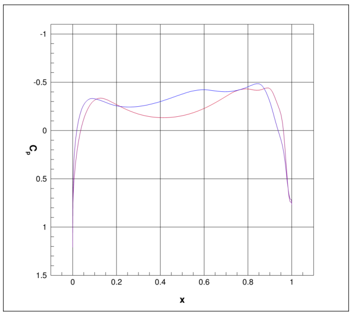


(h) Time point overview

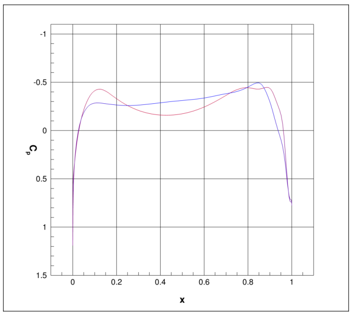


(i) NACA 0010 T=0 comparison

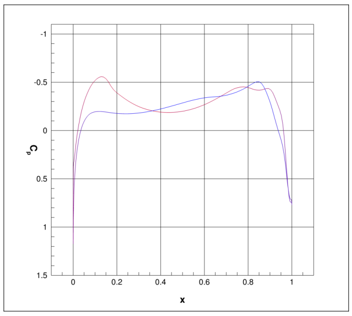
Figure B.3: Pressure distributions of the NACA 0010 optimisation with a constraint on the $C_{L_{RMSE}}$ value for the real inflow condition. All time instances computed with the adjoint harmonic balance method are presented. The suction side is represented by the blue line the pressure side by the red line.



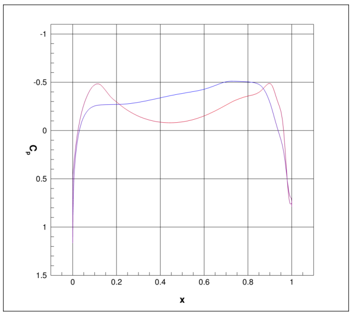
(a) T=0



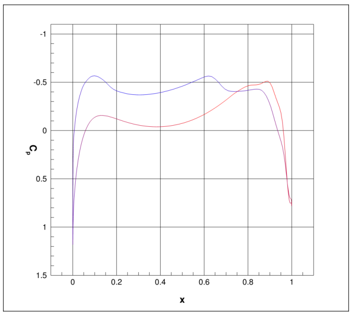
(b) T=1/7



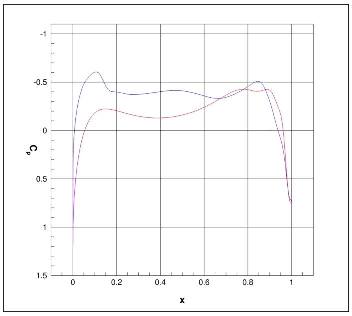
(c) T=2/7



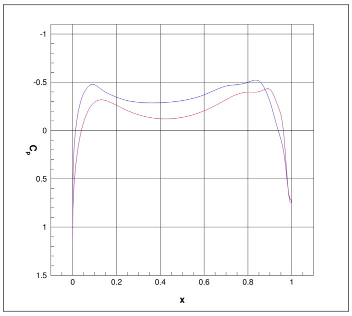
(d) T=3/7



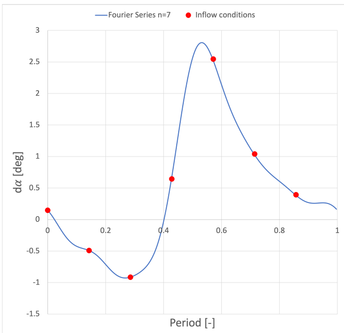
(e) T=4/7



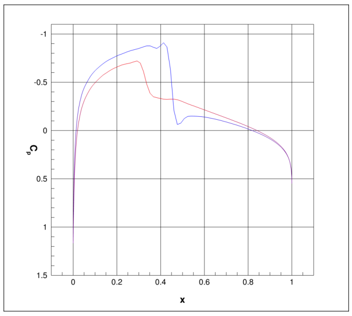
(f) T=5/7



(g) T=6/7



(h) Time point overview



(i) NACA 0010 T=0 comparison

Figure B.4: Pressure distributions of the NACA 0010 optimisation with an alternative thickness constraint for the real inflow condition. All time instances computed with the adjoint harmonic balance method are presented. The suction side is represented by the blue line the pressure side by the red line.

Bibliography

[1] WG ACARE. Strategic Research and Innovation Agenda (SRIA) Vol. 1, Realizing Europe's vision for aviation. *Challenge*, 5(5.6):132–137, 2012.

[2] Daniel Acevedo, Tom van Cranenburgh, Joaquin Exalto, Luuk Halsema, Bob van der Heijden, Yoram Hoogterp, Eliisabet Jahilo, Antonio Jorge, Yair Naaman, Martin van Schie, and Manouk van der Zwan. A320 Auxiliary Propulsion and Power Unit: DSE Report, 2020. 2020.

[3] Pedro Alves, Miguel Silvestre, and Pedro Gamboa. Aircraft propellers—is there a future? *Energies*, 13(16): 4157, 2020.

[4] Nitish Anand, Antonio Rubino, Piero Colonna, and Matteo Pini. Adjoint-based aeroelastic design optimization using a harmonic balance method. In *Turbo Expo: Power for Land, Sea, and Air*, volume 84089, page V02CT35A054. American Society of Mechanical Engineers, 2020.

[5] Nando van Arnhem. Design and Analysis of an Installed Pusher Propeller with Boundary Layer Inflow. Master's thesis, TU Delft, 2015.

[6] Air Transport Action Group (ATAG). The right flightpath to reduce aviation emissions. *UNFCCC Climate Talks*, 2011.

[7] Mohammadreza Azimi, Fathollah Ommi, and Naghmeh Jamshidi Alashti. Using acoustic liner for fan noise reduction in modern turbofan engines. *International Journal of Aeronautical and Space Sciences*, 15(1):97–101, 2014.

[8] Antonio Baeza, Carlos Castro, Francisco Palacios, and Enrique Zuazua. 2-d euler shape design on nonregular flows using adjoint rankine-hugoniot relations. *AIAA journal*, 47(3):552–562, 2009.

[9] Stefan Banach. On the operations in abstract sets and their application to the quations. 3(1):133–181, 1922.

[10] Brennan T Blumenthal, Alaa A Elmiligui, Karl A Geiselhart, Richard L Campbell, Mark D Maughmer, and Sven Schmitz. Computational investigation of a boundary-layer-ingestion propulsion system. *Journal of aircraft*, 55(3):1141–1153, 2018.

[11] Alfonso Bueno-Orovio, Carlos Castro, Francisco Palacios, and Enrique Zuazua. Continuous adjoint approach for the spalart-allmaras model in aerodynamic optimization. *AIAA journal*, 50(3):631–646, 2012.

[12] Carlo E Carcangiu, Jens N Sørensen, Francesco Cambuli, and Natalino Mandas. Cfd–rans analysis of the rotational effects on the boundary layer of wind turbine blades. In *Journal of physics: conference series*, volume 75, page 012031. IOP Publishing, 2007.

[13] Andrew Chapman, Yousef Saad, and Larry Wigton. High-order ilu preconditioners for cfd problems. *International Journal for numerical methods in fluids*, 33(6):767–788, 2000.

[14] Alexandre Joel Chorin. A numerical method for solving incompressible viscous flow problems. *Journal of computational physics*, 135(2):118–125, 1997.

[15] William T Cousins, Dmytro Voytovych, Gregory Tillman, and Eric Gray. Design of a distortion-tolerant fan for a boundary-layer ingesting embedded engine application. In *53rd AIAA/SAE/ASEE Joint Propulsion Conference*, page 5042, 2017.

[16] John De Young. Propeller at high incidence. *Journal of Aircraft*, 2(3):241–250, 1965.

[17] Sydney Lawrence Dixon and Cesare Hall. *Fluid mechanics and thermodynamics of turbomachinery*. Butterworth-Heinemann, 2013.

[18] Jean Donea, Antonio Huerta, J-Ph Ponthot, and Antonio Rodríguez-Ferran. Arbitrary l agrangian–e ulerian methods. *Encyclopedia of computational mechanics*, 2004.

[19] Thomas Economon, Francisco Palacios, and Juan Alonso. Optimal shape design for open rotor blades. In *30th AIAA Applied Aerodynamics Conference*, page 3018, 2012.

[20] Thomas D Economon, Francisco Palacios, and Juan J Alonso. Unsteady continuous adjoint approach for aerodynamic design on dynamic meshes. *AIAA Journal*, 53(9):2437–2453, 2015.

[21] Thomas D Economon, Francisco Palacios, Sean R Copeland, Trent W Lukaczyk, and Juan J Alonso. Su2: An open-source suite for multiphysics simulation and design. *Aiaa Journal*, 54(3):828–846, 2016.

[22] Kivanc Ekici, Kenneth C Hall, and Earl H Dowell. Computationally fast harmonic balance methods for unsteady aerodynamic predictions of helicopter rotors. *Journal of Computational Physics*, 227(12):6206–6225, 2008.

[23] Saeed Farokhi. *Aircraft propulsion*. John Wiley & Sons, 2014.

[24] Christophe Geuzaine and Jean-François Remacle. Gmsh: A 3-d finite element mesh generator with built-in pre-and post-processing facilities. *International journal for numerical methods in engineering*, 79(11):1309–1331, 2009.

[25] Michael B Giles and Niles A Pierce. An introduction to the adjoint approach to design. *Flow, turbulence and combustion*, 65(3):393–415, 2000.

[26] Arathi Gopinath and Antony Jameson. Application of the time spectral method to periodic unsteady vortex shedding. In *44th AIAA Aerospace Sciences Meeting and Exhibit*, page 449, 2006.

[27] Arathi Gopinath, Edwin Van Der Weide, Juan Alonso, Antony Jameson, Kivanc Ekici, and Kenneth Hall. Three-dimensional unsteady multi-stage turbomachinery simulations using the harmonic balance technique. In *45th AIAA Aerospace Sciences Meeting and Exhibit*, page 892, 2007.

[28] Brandon Graver, Kevin Zhang, and Dan Rutherford. emissions from commercial aviation, 2018. 2019.

[29] DB Hanson. Sound from a propeller at angle of attack: a new theoretical viewpoint. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, 449(1936):315–328, 1995.

[30] L HEIDELBERG and M Nallasamy. Unsteady blade pressure measurements for the sr-7a propeller at cruise conditions. In *13th Aeroacoustics Conference*, page 4022, 1990.

[31] Iván Herráez, Buşra Akay, Gerard JW van Bussel, Joachim Peinke, and Bernhard Stoevesandt. Detailed analysis of the blade root flow of a horizontal axis wind turbine. *Wind Energy Science*, 1(2):89–100, 2016.

[32] Raymond M Hicks and Preston A Henne. Wing design by numerical optimization. *Journal of Aircraft*, 15 (7):407–412, 1978.

[33] Antony Jameson and Kui Ou. Optimization methods in computational fluid dynamics. *Encyclopedia of aerospace engineering*, 2010.

[34] Antony Jameson, Luigi Martinelli, and Niles A Pierce. Optimum aerodynamic design using the navier–stokes equations. *Theoretical and computational fluid dynamics*, 10(1):213–237, 1998.

[35] Jack L Kerrebrock. Small disturbances in turbomachine annuli with swirl. *Aiaa Journal*, 15(6):794–803, 1977.

[36] DW Kurtz and JE Marte. A review of aerodynamic noise from propellers, rotors, and lift fans. 1970.

[37] Kai Li, Jiaqing Kou, and Weiwei Zhang. Unsteady aerodynamic reduced-order modeling based on machine learning across multiple airfoils. *Aerospace Science and Technology*, 119:107173, 2021.

[38] David J Lucia, Philip S Beran, and Walter A Silva. Reduced-order modeling: new approaches for computational physics. *Progress in aerospace sciences*, 40(1-2):51–117, 2004.

[39] Peijian Lv and Arvind G Rao. Conceptual analysis of boundary-layer ingestion towards aircraft propulsion integration. In *ISABE conference*, pages 2013–1436, 2013.

[40] Zhoujie Lyu, Zelu Xu, and JRRA Martins. Benchmarking optimization algorithms for wing aerodynamic design optimization. In *Proceedings of the 8th International Conference on Computational Fluid Dynamics, Chengdu, Sichuan, China*, volume 11, 2014.

[41] Benoit Marinus, Michel Roger, and René Van Den Braembussche. Aeroacoustic and aerodynamic optimization of aircraft propeller blades. In *16th AIAA/CEAS aeroacoustics conference*, page 3850, 2010.

[42] Siva Nadarajah and Antony Jameson. A comparison of the continuous and discrete adjoint approach to automatic aerodynamic optimization. In *38th Aerospace Sciences Meeting and Exhibit*, page 667, 2000.

[43] M Nallasamy. Unsteady blade pressures on a propfan-predicted and measured compressibility effects. *Journal of Aircraft*, 31(3):730–736, 1994.

[44] Olivier Petit, Fabiola Costa, Jesuino Takachi Tomita, and Cleverson Bringhenti. Aerodynamic influence of airframe boundary layer on propeller performance.

[45] A Rubino, Matteo Pini, Piero Colonna, T Albring, S Nimmagadda, T Economon, and J Alonso. Adjoint-based fluid dynamic design optimization in quasi-periodic unsteady flow problems using a harmonic balance method. *Journal of Computational Physics*, 372:220–235, 2018.

[46] Max Sagebaum, Tim Albring, and Nicolas R Gauger. High-performance derivative computations using codipack. *ACM Transactions on Mathematical Software (TOMS)*, 45(4):1–26, 2019.

[47] R Singh, G Ameyugo, and F Noppel. Jet engine design drivers: past, present and future. In *Innovation in Aeronautics*, pages 56–82. Elsevier, 2012.

[48] Tomas Sinnige. AE4130 Aircraft Aerodynamics/ AE4135 Rotor-Wake Aerodynamics : Propeller Integration Lecture Notes. TU Delft, 2020.

[49] Tomas Sinnige and Leo L Veldhuis. Pylon trailing edge blowing effects on the performance and noise production of a pusher propeller. In *52nd Aerospace Sciences Meeting*, page 0566, 2014.

[50] Tomas Sinnige, Daniele Ragni, Anwar Malgoezar, Georg Eitelberg, and Leo LM Veldhuis. Apian-inf: an aerodynamic and aeroacoustic investigation of pylon-interaction effects for pusher propellers. *CEAS Aeronautical Journal*, 9(2):291–306, 2018.

[51] S Slavik. Preliminary determination of propeller aerodynamic characteristics for small aeroplanes. *Acta Polytechnica*, 44(2), 2004.

[52] Leroy H Smith Jr. Wake ingestion propulsion benefit. *Journal of Propulsion and Power*, 9(1):74–82, 1993.

[53] Tom CA Stokkermans, Nando Van Arnhem, Tomas Sinnige, and Leo LM Veldhuis. Validation and comparison of rans propeller modeling methods for tip-mounted applications. *AIAA Journal*, 57(2):566–580, 2019.

[54] Eli Turkel. Preconditioned methods for solving the incompressible and low speed compressible equations. *Journal of computational physics*, 72(2):277–298, 1987.

[55] DX Wang, Li He, YS Li, and RG Wells. Adjoint aerodynamic design optimization for blades in multistage turbomachines—part ii: Validation and application. 2010.

[56] Selwyn Edgar Wright. Sound radiation from a lifting rotor generated by asymmetric disk loading. *Journal of Sound and Vibration*, 9(2):223–240, 1969.

[57] Anil Yildirim, Justin S Gray, Charles A Mader, and Joaquim RRA Martins. Aeropropulsive design optimization of a boundary layer ingestion system. In *AIAA Aviation 2019 Forum*, page 3455, 2019.

[58] Weiwei Zhang, Bobin Wang, Zhengyin Ye, and Jingge Quan. Efficient method for limit cycle flutter analysis based on nonlinear aerodynamic reduced-order models. *AIAA journal*, 50(5):1019–1028, 2012.