The Pennsylvania State University
The Graduate School
College of Engineering

AUTOMATED DESIGN AND EVALUATION OF AIRFOILS FOR ROTORCRAFT APPLICATIONS

A Thesis in
Aerospace Engineering
by
Jason Stanko

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

August 2017

The thesis of Jason Stanko was reviewed and approved* by the following:

Sven Schmitz
Associate Professor of Aerospace Engineering
Thesis Advisor

Mike P. Kinzel
Assistant Professor of Aerospace Engineering
Research Associate, Applied Research Laboratory

James G. Coder
Assistant Professor of Mechanical, Aerospace & Biomedical Engineering
University of Tennessee Knoxville
Special Signatory

George A. Lesieutre
Professor of Aerospace Engineering
Head of the Department of Aerospace Engineering

*Signatures are on file in the Graduate School.

Abstract

In this work, a methodology is presented and implemented for the automated design and optimization of airfoil sections. The airfoil optimization is conducted using the CMA-ES genetic algorithm with constraints applied to the airfoil's area and pitching moment coefficient. The design variables are formed through a class shape transformation with orthogonal basis modes, allowing for decreased levels of multi-collinearity in higher-order design spaces, while still maintaining the completeness of lower-order spaces. A Python framework is developed to automate the generation of airfoil performance tables using the RANS CFD solver OVERFLOW 2.2i allowing the optimization methodology to be extended to rotorcraft applications. An empirical maximum lift coefficient criteria is developed and incorporated into the table generation process to overcome inaccuracies associated with stall prediction in CFD-based methods. The methodology presented is used to perform a single point shape optimization on the tip airfoil of a UH-60A baseline rotor. The new tip section shows substantial improvements in forward-flight performance in exchange for a small reduction in the rotor's stall margin. The airfoil optimization and table generation routine shows to be effective for the single design point investigated. This holds promise that the technology developed can later be successfully extended to higher-fidelity automated routines.

Table of Contents

List of Figures

List of Tables

List of Symbols

α	angle of attack
c_l	sectional lift coefficient
c_d	sectional drag coefficient
c_m	sectional pitching moment coefficient
c_p	pressure coefficient
C_T	thrust coefficient
C_P	power Coefficient
C_w	weight Coefficient
ξ	non-dimensional chordwise airfoil coordinate
ζ	non-dimensional vertical airfoil coordinate
η	non-dimensional chordwise airfoil coordinate in unwrapped domain
P_n	Legendre polynomial of order n
$B_{i,n}$	i^{th} Bernstein polynomial of order n
ϕ_i	i^{th} basis mode
C	covariance matrix
μ_p	penalty parameter
ϵ	population size

σ	standard deviation of candidate solutions
$y_m^{(k)}$	mean design variable vector at iteration k
$\mathcal{N}(y_m, C)$	Gaussian distribution centered at y_m with variance C
u_i	Cartesian velocity component
x_i	Cartesian coordinate
ρ	fluid density
E	total fluid energy
e	internal energy
h	specific enthalpy
γ	ratio of specific heats
p	pressure
v_i	velocity at rotor disk
Q	rotor torque
T	rotor thrust
R	rotor radius
A	rotor area
I_b	rotor blade inertia about flapping axis
β	rotor blade flapping angle
ψ	rotor azimuth position
θ_{1s}	longitudinal cyclic pitch
θ_{1c}	lateral cyclic pitch
θ_0	collective pitch
U_T	flow velocity tangential to rotor blade
U_P	flow velocity perpendicular to rotor blade

U_R	flow velocity radial to rotor blade
λ_i	non-dimensional inflow velocity
μ	advance ratio
Re	Reynolds number
Re_{δ_2}	momentum thickness Reynolds number
Re_{δ_2}	displacement thickness Reynolds number
c_f	skin friction coefficient

Acknowledgments

First and foremost I would like to thank my Fiancée Sarah for her continued support throughout the years, without whom I likely would not have pursued my Masters degree in the first place.

I would also like to humbly thank Dr. Jim Coder for accepting me into his research and providing me the opportunity to work on this project; along with sharing his expertise on aerodynamics, computational fluid dynamics, and for guiding me through this process.

I would like to thank Dr. Mike Kinzel for his feedback and for sharing his knowledge on the subject matter.

I would like to thank Dr. Sven Schmitz for the insight that he provided me, and for his continuous feedback over the course of this project.

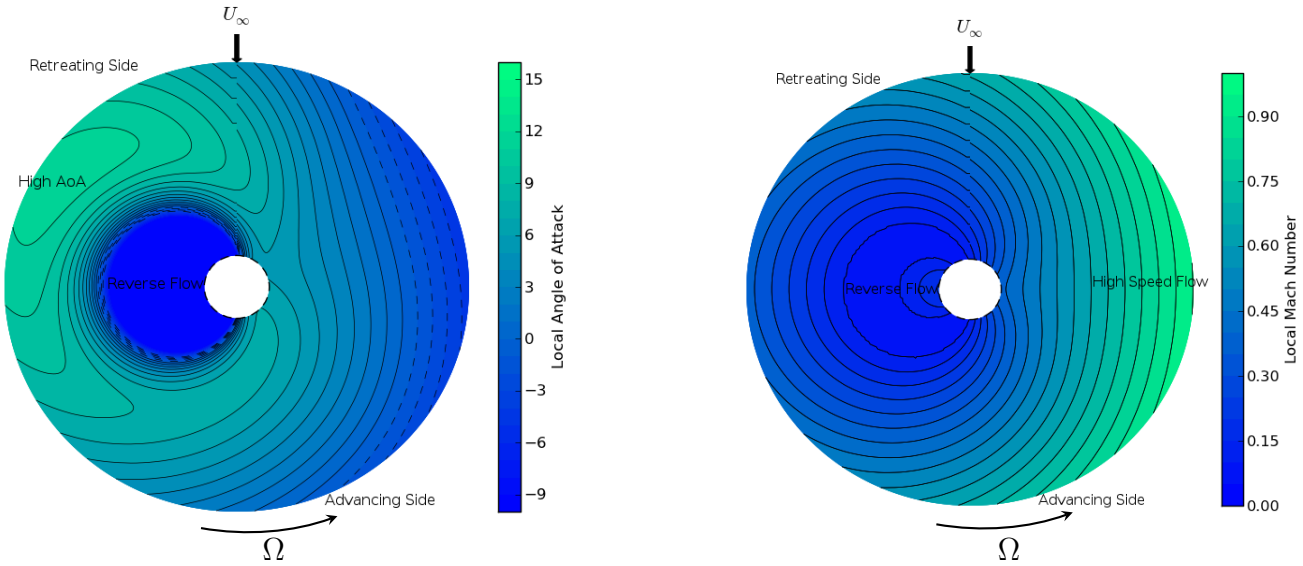
Finally, I would like to thank all my friends and family who have supported me and helped make this possible.

This effort was sponsored by the US Government under Other Transaction number W15QKN-10-9-0003 between Vertical Lift Consortium, Inc. and the Government, project NRTC-FY15-C-02, technical monitor Dr. Mark Calvert. The US Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the US Government.

Chapter 1
Introduction

The design and optimization of airfoils has long been a subject of research in the rotorcraft industry. Proper choice in airfoils for rotorcraft blades is central to the performance of the vehicle. However, the extreme operating environment of rotorcraft blades has historically made this a difficult process. The performance of early rotorcraft was hampered by high pitching moments resulting from the wide variations in angles-of-attack experienced by rotorcraft blade sections. Limitations in construction techniques and building materials with poor torsional stiffness during the 1940's restricted early rotorcraft to the almost universal usage of symmetric airfoils [2] such as the NACA0012 whose center of pressure location is often less sensitive to changes in angles-of-attack.

In the 1950-60s, advances in manufacturing techniques coupled with developments in computational methods lead to the use of cambered airfoil sections allowing for improved lift-to-drag ratios [2]. The performance of these airfoils, however, was still limited by conflicting design requirements between blade sections. In forward flight, retreating blade sections experience low-speed flow and large angles-of-attack, which favors thicker airfoils with high maximum lift coefficients at low Mach numbers to mitigate stall. In contrast, the onset of compressibility effects at the advancing blade tip acts as a primary drag source in high-speed forward flight, favoring blades with thin profiles and high drag divergence Mach numbers. Figure 1.1 shows examples of typical rotor flow distributions in forward flight; on the right side of 1.1b, a high-speed region exists at the advancing blade tip while in Figure 1.1a, a region of high angles-of-attack is visible on the retreating blade.



(a) angle-of-attack (b) Mach number

Figure 1.1: Examples of rotor disk flow distributions in forward flight, $\mu = 0.45$

Additional improvements in manufacturing techniques during the 1970's helped to alleviate problems caused by conflicting design requirements by allowing for the use of multiple airfoil sections across the blade. The UH-60A, for example, initially employed a single SC1095 airfoil section over the length of the blade; however, during testing it was revealed that the rotor was unable to meet the Army's maneuverability requirements. A leading-edge cap was added to the SC1095 to overcome this. The new airfoil's mean chordline was rotated nose-down by 1 deg and a higher maximum lift coefficient. The modified airfoil was named the SC1094R8 and was implemented between 47% - 85% of the rotor radius [1].

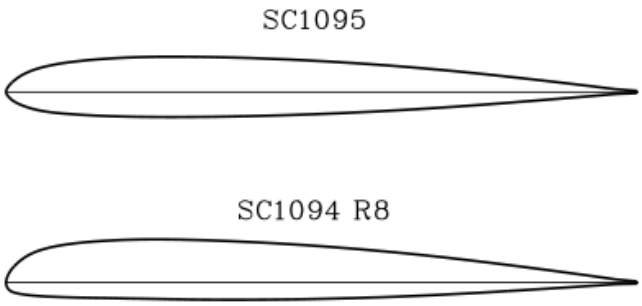


Figure 1.2: SC1095 and SC1094R8 airfoil sections (Ref. [1])

Other attempts to overcome these conflicting design requirements have been made through the use of coaxial vehicles such as the X2 [10]. This vehicle is capable of advance ratios of up to 0.5 with as much as 50% of the retreating blade operating in reverse flow. By balancing the left and right-side lift production between the two rotors, the vehicle can obtain significantly higher airspeeds as the vehicle's controllability is less impacted by the large roll moments generated resulting from loss of lift on the retreating blade. The vehicle's performance, however, is still subject to high power requirements and large vibrations caused by wave drag at the advancing blade tips.

Most current generation aircraft employ multiple airfoils across the length of their rotor blades. The design of these sections, however, is often expensive to perform manually. Some of this burden can be reduced by implementing automated routines into the design process. However, extending these routines to rotorcraft airfoil design is less straightforward than for fixed-wing aircraft. Rotorcraft have large operational envelopes which requires performance evaluations be made across multiple flight conditions. This generally prohibits the use of high-fidelity Reynolds-averaged Navier-Stokes (RANS) frameworks such as HELIOS [11], as these methods are often too computationally expensive to be used for early-stage design routines [12–14].

An alternative, design-oriented approach is taken by comprehensive analysis tools such as RCAS [15] and CAMRAD II [16]. Rather than simulate the entire flow volume around the rotor, these tools generally avoid resolving all near-surface flow detail for each trim condition and rotor configuration. Instead, reduced-order aerodynamic models, such as lifting-line theory, are used to estimate the three-dimensional flowfield at the rotor. These methods are then supplemented by quasi-steady, two-dimensional airfoil lookup tables containing pre-computed airfoil data at a range of Mach numbers and angles-of-attack. Unsteady effects are incorporated through empirical corrections, and structural dynamics are often included as well through reduced-order beam bending models. Comprehensive analysis tools currently act as a major workhorse of the rotorcraft industry, and with proper tuning sufficient engineering accuracy can be achieved [17].

The use of pre-calculated tables in comprehensive analysis tools decouples the airfoil flow physics from the induced-inflow environment of the rotor, providing significant time savings allowing for much larger numbers of rotor configurations to be evaluated for given computing resources. As such, many studies have been carried

out using comprehensive analysis tools for a variety of multi-disciplinary objectives [18–22]. Commonly, these optimization methods alter taper, twist, or control device deployment; however, the integration of automated airfoil optimization into these routines is often computationally cumbersome. At each design iteration, data tables must be re-generated for the new airfoil, and the table generation process itself is often non-trivial and expensive. Mayda and van Dam (Ref. [23]) laid out a successful methodology for generating airfoil performance tables using the implicit thin-layer Navier-Stokes solver ARC2D [24]. However, CFD tends to over-predict the airfoil maximum sectional lift coefficient, a behavior which has been noted by Refs. [7, 25, 26] and others. This deficiency, if left uncorrected, creates a weakness that can be readily exploited by the optimization routine. The optimizer may attempt to reduce the rotor profile power by decreasing solidity and driving blades towards thinner planforms than the aircraft's stall margin would physically allow, if $c_{l,max}$ been accurately predicted. This ultimately results in an under-performing rotor that fails to meet expectations.

A variety of automated shape optimization strategies have been developed over the last four decades to automate the design of these airfoil sections. Genetic algorithms (GA) generally require more function evaluations. However, they have recently grown in popularity [27, 28] due to their robustness and ease of implementation. Genetic algorithms are stochastic methods which attempt to simulate natural selection in a population of candidate solutions generated at each design iteration. Genetic algorithms evaluate the fitness of each candidate solution and update the optimization path depending on the best performing individuals of the generation. By averaging search paths over multiple candidate designs, GA optimizations have a reduced susceptibility to objective function discontinuities and can better avoid premature convergence at local minima [29]. Because the fitness of each candidate is evaluated independently, these algorithms have a straightforward parallelization strategy, thus making them well suited for computational fluid dynamics problems where design point evaluations can often take long periods of time.

The efficiency and effectiveness of such methods, however, is subject to the design variables being optimized. For airfoil shape optimization problems, these design variables take the form of an airfoil parameterization method, where the airfoil surface is defined by a set of prescribed functions. A multitude of ways to define these

functions have been developed, including the popular Hicks-Henne sine functions [30], PARSEC [31], and orthogonal mode decomposition [5, 32, 33]. In addition, several surveys of parameterization methods have also been conducted [34–36]. Of the developed methods, the class shape transformation (CST) developed by Kulfan [6] has become a popular choice for automated design routines due to its ability to represent a large number of airfoils using relatively few design variables. The Bernstein polynomial basis modes originally employed by Kulfan, however, are not orthogonal and can be prone to issues in higher-dimensional design spaces. A study by Vassberg et al. [37] analyzed the extension of airfoil optimization techniques to higher-order design spaces using the degree raising property of Bernstein polynomials, they noted an initial decrease in optimization performance, despite the lower-order design space being exactly contained in the higher-order-space. Although they were able to overcome this by extending the bounds of the design variables, this presents an undesirable coupling between the number of design variables and the optimization surface topology.

1.1 Goals and Research Approach

The overarching goal of this work is to develop an airfoil shape optimization method and extend it to rotorcraft applications. The airfoil optimization was driven by the CMA-ES algorithm with constraints applied to the airfoil cross-sectional area and pitching-moment coefficient through an augmented Lagrange penalty function. The airfoil was parameterized using a CST-based method. A set of orthogonal polynomials was chosen to represent the basis modes to mitigate cross-talk between design variables and maintain efficiency in higher-order design spaces.

The optimization was then extended to rotorcraft by the inclusion of an automated performance table generation method. This was accomplished through Python framework which was used to drive the Reynolds-averaged Navier-Stokes computational fluid dynamics code OVERFLOW. In order to account for the over-prediction of the maximum sectional lift coefficient, an empirical $c_{l,max}$ criteria was developed, and a lift coefficient correction routine was implemented as a post-processing step.

The methodology developed is applied to a UH-60A planform to serve as the baseline rotor due to the large volume of experimental and computational data

available. The optimized airfoil performance was investigated through the use of a blade-element momentum theory (BEMT) based rotor analysis code. To demonstrate the effectiveness of the tool for airfoil design, a design point was chosen to minimize the change in trim state while still producing substantial performance gains. In forward flight, the rotor tip acts primarily as a drag source while often producing negative lift [28]. By optimizing the tip airfoil to minimize drag at a constant lift coefficient, the optimization methodology was capable of reducing the power required for forward flight with minimal impact on hover performance.

In summary, the objectives of this thesis are as follows:

- Implement a constrained airfoil shape optimization method
- Develop a parameterization method with reduced cross-talk between basis modes
- Implement an automated airfoil performance table generation method
- Develop and implement a maximum lift criterion
- Demonstrate the effectiveness of the tool developed on a representative baseline rotor

Chapter 2
Background

In this chapter, the relevant aerodynamic theory to simulate a helicopter main rotor in forward flight is developed using blade-element momentum theory (BEMT). The equations formed here create the basis for the helicopter performance code ROTOR, which is used to investigate the optimized airfoil and lift coefficient correction influence on rotor performance. The second section introduces the governing equations of fluid dynamics, the Navier-Stokes equations, which form the basis for the OVERFLOW computational fluid dynamics (CFD) code used to drive the airfoil table generation and the optimization routine. Finally, a variety of airfoil parameterization methods are introduced. These methods are essential to airfoil optimization as they define the design variables being optimized.

2.1 Rotorcraft Aeromechanics

2.1.1 Momentum theory

The earliest developments in rotorcraft aerodynamics were made by Rankine in 1865, who developed momentum theory for marine propeller applications. In momentum theory, the rotor is treated as a disk with uniform inflow that produces thrust by discontinuously compressing the fluid passing through it. Consider a rotor in axial flow (eg. hover or climb), as shown in Figure 2.1.

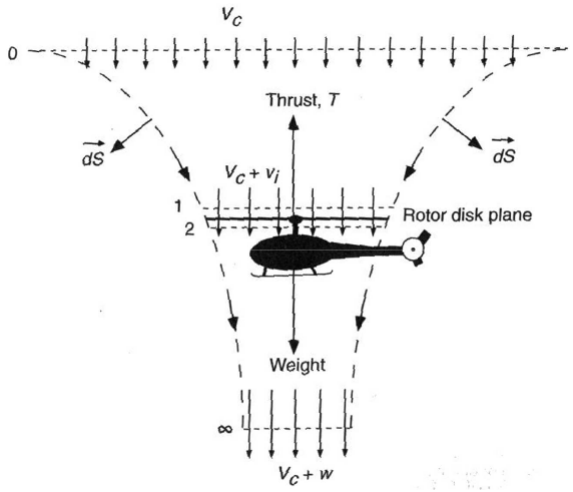


Figure 2.1: Momentum theory control volume for a hovering rotor (Ref [2])

If the pressure jump across the rotor disk is uniform, then the thrust T produced by the rotor is

$$T = (p_2 - p_1)A \tag{2.1}$$

where A is the rotor area, p_1 and p_2 are the pressures at positions 1 and 2, respectively. The power required in axial flow can be expressed as the work per unit time applied to the fluid traveling through the rotor

$$P = T(v_i) \tag{2.2}$$

Unlike the pressure, the velocity does not jump discontinuously across the disk. Assuming a uniform inflow distribution and applying a mass balance on either side of the rotor disk it can be shown that

$$\dot{m}_1 = \rho v_{i1} A_1$$
$$\dot{m}_2 = \rho v_{i2} A_2$$
$$A_1 = A_2$$
$$\therefore v_{i1} = v_{i2}$$

To find the velocity at the rotor disk in hover (v_h), a conservation of momentum balance between the thrust on the rotor and the change in momentum of fluid in

the far rotor wake is first applied as:

$$T = \dot{m}w \tag{2.3}$$

Similarly, applying a conservation of energy balance between the rotor thrust and the change in energy of the fluid at the rotor disk yields

$$Tv_h = \frac{1}{2}\dot{m}w^2 \tag{2.4}$$

Combining Equations (2.3) and (2.4), the flow induced at the rotor disk in hover is related to the velocity in the far wake through

$$v_h = \frac{1}{2}w \tag{2.5}$$

Next, Bernoulli's equation is applied twice; first between the far upstream flow and the rotor disk upper surface, and secondly between the locations below the rotor disk and in the far wake such that

$$\begin{aligned} p_\infty &= p_1 + \frac{1}{2}\rho v_h \\ p_2 + \frac{1}{2}\rho v_h &= p_\infty + \frac{1}{2}\rho w^2 \\ \frac{T}{A} &= p_2 - p_1 = \frac{1}{2}\rho w^2 \end{aligned}$$

Substituting Eq. (2.5) yields the velocity induced at the rotor disk for a hovering rotor

$$v_h = \sqrt{\frac{T}{2\rho A}} \tag{2.6}$$

Finally, the power required for a hovering rotor becomes

$$P = v_h T = \sqrt{\frac{T^{3/2}}{2\rho A}} \tag{2.7}$$

In axial flow, there is an additional velocity component at the rotor disk v_c. This addition modifies the power required through

$$P = T(v_c + v_i) \tag{2.8}$$

where the induced velocity can be found through similar applications of Bernoulli's equation and control volume analysis. Additionally, in forward flight, the inflow velocity becomes skewed, and the ideal power required becomes

$$P = T(V_\infty \sin\alpha + v_i) \tag{2.9}$$

The inflow distribution, however, becomes nonuniform, and its prediction is not provided through pure momentum theory. Additionally, momentum theory does not provide any insight as to how the forces are generated by the airfoil sections along the blade. Instead, a modified version of momentum theory, blade-element momentum theory (BEMT) will be used in this work.

2.1.2 Blade-Element and Momentum Theory

In blade-element momentum theory, the rotor is decomposed into a set of discrete radial dr and azimuth stations $d\psi$. At each station, force and moment calculations are made by looking up pre-computed sectional force and moment coefficients from airfoil performance tables. Figure 2.2 shows a diagram of the forces and moments acting on a discrete blade-element

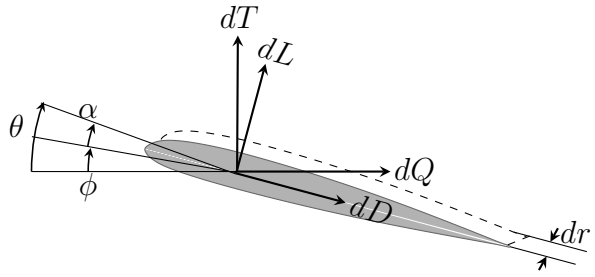


Figure 2.2: Blade-element diagram

where θ is the geometric pitch angle relative to the rotor plane, α is the effective local angle of attack, and ϕ is the relative angle between the rotor plane and the local flow. The sectional forces along the blade can be then written as

$$dL = \frac{1}{2}\rho c(r)U(r,\phi)^2 c_l(\alpha, U)dr \tag{2.10}$$

$$dD = \frac{1}{2}\rho c(r)U(r,\phi)^2 c_d(\alpha, U)dr \tag{2.11}$$

The net forces and moments can be calculated by integrating across the radius and around the azimuth

$$T = \frac{1}{2\pi}\int_0^{2\pi}\int_0^R (dL\cos\phi + dD\sin\phi)drd\psi \tag{2.12}$$

$$Q = \frac{1}{2\pi}\int_0^{2\pi}\int_0^R (dL\sin\phi + dD\cos\phi)rdrd\psi \tag{2.13}$$

The air density ρ is a function of altitude and temperature, and the local chord $c(r)$ is a geometric parameter depending on the blade planform. The calculations of the local flow velocity U, the sectional lift coefficient c_l, and sectional drag coefficient c_d are more complicated to resolve as they depend on the physical motion of the blade; their formulations will be discussed in the following subsections.

2.1.2.1 Local Flow Velocity

As the rotor moves through the air, the relative angle between the blade velocity and the freestream velocity results in a nonuniform velocity across the rotor. On the retreating side, sections where the freestream and blade velocities are in the same direction have a lower effective local velocity. Here a reverse flow region develops centered at $3\pi/2$. On the advancing side, the rotor and freestream velocities are in opposite directions, which increases the local velocity and creates a high-speed region at the rotor tip (See Figure 1.1b). The tangential velocity for a rotor in forward flight becomes

$$U_T(r,\psi) = \Omega r + V_\infty \sin\psi \tag{2.14}$$

and the radial or crossflow component flow component becomes

$$U_R(r,\psi) = \mu\Omega R\cos\psi \tag{2.15}$$

Blade Motion

The asymmetric flow caused by the blade and freestream relative velocities causes an unsteady loading across the blade. Helicopters commonly employ a hinge allowing for out-of-plane motion to compensate for this, which induces a cyclic flapping motion (Figure 2.3) as the blade travels around the azimuth. This hinge is commonly offset by a non-dimensional distance e. A diagram of the flapping motion is given in Figure 2.3 where β is the flapping angle.

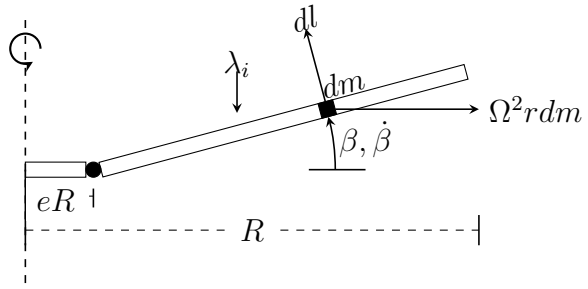


Figure 2.3: Rotor flapping motion (Ref. [2])

This flapping motion will induce two new additional sources of velocity perpendicular to the rotor plane. The first of these is a direct result of the flapping motion $\dot{\beta}$, while the second is a result of the relative angle between the flapping blade and the freestream velocity. The out-of-plane velocity becomes

$$U_P(r,\psi) = \underbrace{\Omega R \lambda_i(r,\psi)\cos\beta(\psi)}_{\text{inflow velocity}} + \underbrace{r\dot{\beta}(\psi)}_{\text{flapping motion}} + \underbrace{\mu\Omega R\beta(\psi)\cos(\psi)}_{\text{relative angle}} \tag{2.16}$$

To determine the new local velocity, the flapping motion must be resolved. To accomplish this, a force and moment balance can be taken about the flapping hinge. The centrifugal force acts as a restoring force about the hinge.

$$\begin{aligned} F_C &= \int_{eR}^{R} \Omega^2 r(r-eR)\beta(\psi)dm(r)dr \\ &= \Big[\Omega^2 \int_{eR}^{R} r(r-eR)dm(r)dr\Big]\beta(\psi) \end{aligned} \tag{2.17}$$

The blade inertial force acts about the hinge, resisting changes in motion through

$$\begin{aligned} F_I &= \int_{eR}^{R} m(y-eR)^2 \ddot{\beta(\psi)}dr \\ &= \ddot{\beta}(\psi)\int_{eR}^{R} m(y-eR)^2 dr \\ &= I_b\ddot{\beta}(\psi) \end{aligned} \tag{2.18}$$

Lift acts as the perturbing force about the hinge.

$$F_L = \int_{eR}^{R} L(r-eR)dr \tag{2.19}$$

Combining the equation of motion for the flapping blade becomes

$$I_b\ddot{\beta}(\psi) + \Big[\Omega^2 \int_{eR}^{R} r(r - eR)dm(r)dr\Big]\beta(\psi) = \int_{eR}^{R} L(r - eR)dr \tag{2.20}$$

Assuming periodicity, a substitution can be made to remove the time dependency of the differential

$$\frac{d\beta}{dt} = \frac{d\psi}{dt}\frac{d\beta}{d\psi} = \Omega\frac{d\beta}{d\psi}$$

$$\frac{d^2\beta}{dt^2} = \Big(\frac{d\psi}{dt}\Big)^2\frac{d^2\beta^2}{d\psi^2} + \frac{d\beta}{d\psi}\frac{d^2\psi}{dt^2}$$

$$\Omega^2 I_b\frac{d^2\beta^2}{d\psi^2} + \Omega^2\Big[\int_{eR}^{R} r(r - eR)dm(r)dr\Big]\beta = \frac{1}{\int}_{eR}^{R} L(r - eR)dr$$

Finally, the equation of motion for a rigid flapping rotor in forward flight becomes

$$I_b\frac{d^2\beta^2}{d\psi^2} + I_b\omega^2\beta = \frac{1}{\Omega^2}\int_{eR}^{R} L(r - eR)dr \tag{2.21}$$

Inflow Velocity

The asymmetric flow causes the induced velocity at the rotor disk to become asymmetric as well, meaning that the uniform inflow assumption used in momentum theory no longer holds. Unlike for momentum theory in hover, the induced velocity cannot be determined *a priori* as it depends on the rotor wake. The rotor wake is difficult to resolve as it depends on the blade motion, as well as the thrust distribution and trim state. To overcome this, BEMT uses a semi-empirical estimate of the rotor inflow distribution based on adaptations from momentum theory. Multiple inflow models have been proposed [38]. A commonly used model is the linear inflow model first proposed by Glauert [39]

$$\lambda_i(r, \psi) = \lambda_0(1 + k_x r\cos\psi) \tag{2.22}$$

where k_x is commonly chosen to be a static 1.2; however, in practice it should depend on the advance ratio. Here λ_0 is the non-dimensional inflow velocity for a hovering rotor

$$\lambda_0 = \frac{v_{i_0}}{\Omega R} = \frac{v_i}{v_{\text{tip}}} \tag{2.23}$$

2.1.2.2 Local Angle of Attack

The flapping motion induces a perpendicular velocity that changes the local angle of attack along the blade. The local angle of attack becomes a function of both radial and azimuth location according to

$$\begin{aligned} \alpha(r,\psi) &= \theta - \phi \\ &= \theta - \arctan\frac{U_T(r,\psi)}{U_P(r,\psi)} \end{aligned} \tag{2.24}$$

where

$$U_T(r,\psi) = \Omega r + V_\infty \sin\psi$$
$$U_P(r,\psi) = \Omega R\lambda_i(r,\psi)\cos\beta(\psi) + r\dot{\beta(\psi)} + \mu\Omega R\beta(\psi)\cos(\psi)$$

The pitch angle θ changes radially due to blade twist θ_t and around the azimuth due to the lateral cyclic control input θ_{1c} and the longitudinal cyclic θ_{1s} according to

$$\theta(r,\psi) = \theta_0 + \theta_t(r) + \theta_{1c}\cos\psi + \theta_{1s}\sin\psi \tag{2.25}$$

where θ_0 is the collective control input. This represents a common control configuration; however higher-harmonic pitch control inputs are also possible.

For the control configuration given in Equation (2.25), the collective control input acts to increase the pitch on all blades simultaneously, while θ_{1c} and θ_{1s} act to increase the blade pitch on a 1/rev basis with maximums at 0 deg and 90 deg respectively. For forward flight, increasingly negative inputs of θ_{1s} are used to tilt the rotor disk forward. A positive input of θ_{1c} is also required to balance the blades laterally due to the asymmetry produced by the forward motion.

2.1.2.3 Force and Moment Coefficients

At each station, force and moment calculations are made by looking up precalculated sectional force and moment coefficients through the use of airfoil performance tables. The range of flow speeds experienced by the airfoil sections requires that these tables contain flow solutions for Mach numbers ranging from 0.0 to

1.0 . In addition, angles-of-attack from -180 deg to 180 deg must be included to capture the reverse flow region on the retreating blade. Ideally these tables would be generated experimentally however the high costs prohibit this approach. Instead, this data is often generated computationally using a CFD solver or splined in from existing data tables such as the NACA 0012.

Generating usable data tables requires that an appropriate solver is used. The solver must be capable of characterizing the range of flow conditions experienced by the rotor. Varying flow speeds across the rotor disk necessitate a solver capable of accurately resolving both incompressible and compressible flow regimes. In addition, the large angles-of-attack range experienced by the rotor requires a solver able to accurately resolve viscous and turbulence effects, particularly in the stall and post-stall regimes. The choice of solver and table generation procedure will be further covered in the next chapter.

2.2 Governing Equations

The Navier-Stokes equations are mass, momentum and energy conservation equations that describe the behavior of a fluid flowfield. This section derives the conservation form of the Navier-Stokes equations that are used by the CFD solver to calculate airfoil sectional properties.

2.2.1 Conservation of Mass

Considering a control volume V with a surface S and unit normal vector $\vec{n}$.

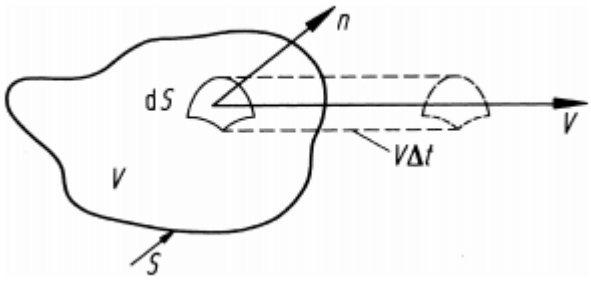


Figure 2.4: Control volume diagram (Ref. [3])

The time rate of change of mass inside the control volume can be expressed as

$$\frac{d}{dt}\int_V \rho dV = 0 \tag{2.26}$$

Applying the Reynolds Transport Theorem, the equation can be split into its volume and surface integrals

$$\frac{d}{dt}\int_V \rho dV = \int_V \frac{\partial \rho}{\partial t} + \int_S (u_i n_i)\rho dA \tag{2.27}$$

Applying the divergence theorem, the two integrals can be combined into a single volume integral

$$\int_V \Big[\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{u})\Big] dV = 0 \tag{2.28}$$

If the integral is equal to zero, then the integrand must also be equal to zero. Then the conservative form of the mass conservation equation becomes

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i}(\rho u_i) = 0 \tag{2.29}$$

2.2.2 Conservation of Momentum

From Newton's second law, a fluid element's time rate of change of momentum can be expressed as a sum of the external forces acting on it

$$\frac{d}{dt}\int_V \rho u_i dV = \int_S T_i dS + \int_V \rho B_i dV \tag{2.30}$$

where T_i are surface forces per unit area and B_i are the body forces per unit volume acting on the fluid element. The surface forces in the i direction can be decomposed into its isotropic portion $p\delta_{ki}$ and deviatoric component τ_{ki}

$$T_{ij} = \sigma_{ki} n_i = (-p\delta_{ki} + \tau_{ki}) n_i \tag{2.31}$$

where τ_{ki} is the viscous stress tensor. Following Stokes' hypothesis gives

$$\tau_{ij} = \mu\Big[\frac{\partial u_k}{\partial x_i} + \frac{\partial u_i}{\partial x_k}\Big] - \frac{2\mu}{3}\frac{\partial u_j}{\partial x_j}\delta_{ij} \tag{2.32}$$

By applying the divergence theorem to Equation (2.30), the surface force contribution can be arranged to

$$\int_S T_i dS = \int_V \frac{\partial}{\partial x_i}(-p\delta_{ki} + \tau_{ki})dV \tag{2.33}$$

Following the same methodology from the mass conservation, the time rate of change in momentum can be transformed into a volume integral as well. The momentum conservation equation becomes

$$\frac{\partial \rho u}{\partial t} + \frac{\partial}{\partial x_i}(\rho u_k u_i + p\delta_{ki} - \tau_{ki}) = \rho B_i \tag{2.34}$$

2.2.3 Conservation of Energy

The time rate of change of total energy inside a fluid volume is a sum of the energy flux into the element, the rate of work done on the element by external forces $\vec{T}_i$ and $\vec{B}_i$, and the heat flux q_i through the surface S. The conservation of energy equation becomes

$$\frac{d}{dt}\int_V \rho(e + \frac{u^2}{2})dV = \int_S \vec{T}_i u_i dS + \int_V \rho \vec{B}_i u_k dV - \int_S q_i n_i dS \tag{2.35}$$

By similar application of the Reynolds transport theorem and the divergence theorem as before, the conservation of energy equation can be rearranged to

$$\frac{\partial(\rho E)}{\partial t} + \frac{\partial}{\partial x_i}(\rho E u_i + (p\delta_{ki} - \tau_{ki})u_i - q_i) = \rho \vec{B}_i u_i \tag{2.36}$$

2.2.4 Equation of State

Using an ideal, calorically perfect gas, the pressure is commonly found through the equation of state

$$p = (\gamma - 1)[E - \frac{1}{2}(u_i u_i)] \tag{2.37}$$

where E is the total energy

$$E = (e + \frac{u^2}{2}) \tag{2.38}$$

and here γ is the ratio of specific heats, e is the internal energy, and h is the specific enthalpy

$$\gamma = \frac{c_p}{c_v} \tag{2.39}$$

$$e = c_v T \tag{2.40}$$

$$h = c_p T \tag{2.41}$$

2.2.5 The Navier-Stokes Equations

Combining the equations for conservation of mass, momentum and energy, the three-dimensional Navier-Stokes equations in conservative form can be written as

$$\frac{\partial \vec{q}}{\partial t} + \frac{\partial}{\partial x_i}\vec{F_i} = 0 \tag{2.42}$$

where q is the vector of conserved quantities

$$\vec{q} = \begin{bmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_2 \\ E \end{bmatrix} \tag{2.43}$$

The flux vector F_i can be decomposed into the convective and viscous flux vectors through $\vec{F_i} = \vec{f_i^c} - \vec{f_i^v}$ where

$$\vec{f_i^c} = \begin{bmatrix} \rho u_i \\ \rho u_i u_1 + p\delta_{i1} \\ \rho u_i u_2 + p\delta_{i2} \\ \rho u_i u_3 + p\delta_{i3} \\ (E\rho + p)u_i \end{bmatrix} \qquad \vec{f_i^v} = \begin{bmatrix} 0 \\ \tau_{i1} \\ \tau_{i2} \\ \tau_{i3} \\ u_i\tau_{ki} + q_i \end{bmatrix} \tag{2.44}$$

Extension of the Navier-Stokes equations to include turbulence effects in RANS approaches requires an additional equation to model the Reynolds stress tensor τ_{ki} and close the system of equations.

2.3 Airfoil Shape Parameterization

Airfoil parametrization is a key part of optimization as it allows for the mathematical description of an airfoil geometry through a set of design variables. In this way, the airfoil parameterization plays an important part in defining the cost surface of the optimization problem. The following sections aim to introduce common methods of forming these design variables.

2.3.1 Local Parameterization

Local parameterization methods use singular control points to form the design variables [40]. These methods do not require a fixed set of shape parameters which allows them to be easily extended to a wide range of applications. These points can be individual discrete surface points; however, this may introduce high-frequency noise. Instead, semi-discrete approaches are commonly taken where a set of control points are distributed over the airfoil surface with a smoothing function defined through them, so that movement of an individual surface point has an area of effect (Figure 2.5).

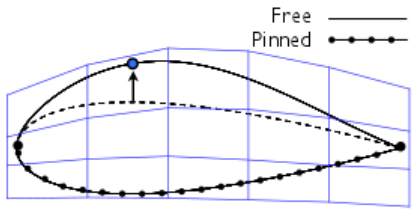


Figure 2.5: Semi-discrete local parameterization (Ref. [4])

2.3.2 Geometric

In order to make design variables more intuitive, other parameterization methods such as PARSEC [31] describe an airfoil through a set of real geometric airfoil parameters as design variables. In PARSEC, the design variables are a set of 11 basic airfoil geometric parameters (Figure 2.6).

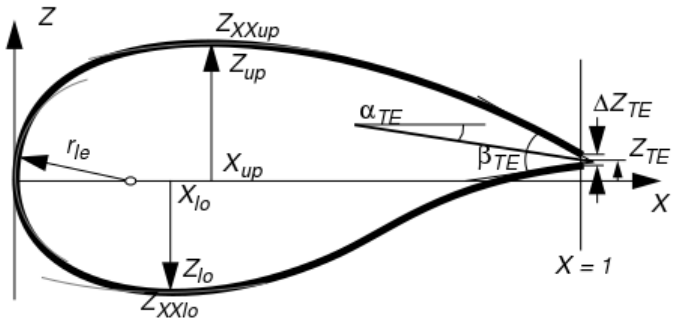


Figure 2.6: PARSEC geometric parameters

These parameters are used to define a set of polynomial coefficients that describe the airfoil surface through

$$z = \sum_{i=1}^{n} a_i X^{i-0.5} \tag{2.45}$$

where the polynomial coefficients a_i and b_i are found by solving two systems of linear equations, see below.

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ X_u^{1/2} & X_u^{3/2} & X_u^{5/2} & X_u^{7/2} & X_u^{9/2} & X_u^{11/2} \\ 1/2 & 3/2 & 5/2 & 7/2 & 9/2 & 11/2 \\ \frac{1}{2}X_u^{-1/2} & \frac{3}{2}X_u^{3/2} & \frac{5}{2}X_u^{5/2} & \frac{7}{2}X_u^{7/2} & \frac{9}{2}X_u^{9/2} & \frac{11}{2}X_u^{11/2} \\ -\frac{1}{4}X_u^{-3/2} & \frac{3}{4}X_u^{-1/2} & \frac{15}{4}X_u^{1/2} & \frac{35}{4}X_u^{3/2} & \frac{63}{4}X_u^{5/2} & \frac{99}{4}X_u^{7/2} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{bmatrix} = \begin{bmatrix} Z_{te} + \frac{1}{2}\Delta Z_{te} \\ \sqrt{r_l e} \\ Z_{up} \\ \tan(2\alpha_{te} - \beta_{te})/2 \\ 0 \\ Z_{xxup} \end{bmatrix} \tag{2.46}$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ X_{lo}^{1/2} & X_{lo}^{3/2} & X_{lo}^{5/2} & X_{lo}^{7/2} & X_{lo}^{9/2} & X_{lo}^{11/2} \\ 1/2 & 3/2 & 5/2 & 7/2 & 9/2 & 11/2 \\ \frac{1}{2}X_{lo}^{-1/2} & \frac{3}{2}X_{lo}^{3/2} & \frac{5}{2}X_{lo}^{5/2} & \frac{7}{2}X_{lo}^{7/2} & \frac{9}{2}X_{lo}^{9/2} & \frac{11}{2}X_{lo}^{11/2} \\ -\frac{1}{4}X_{lo}^{-3/2} & \frac{3}{4}X_{lo}^{-1/2} & \frac{15}{4}X_{lo}^{1/2} & \frac{35}{4}X_{lo}^{3/2} & \frac{63}{4}X_{lo}^{5/2} & \frac{99}{4}X_{lo}^{7/2} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \end{bmatrix} = \begin{bmatrix} Z_{te} + \frac{1}{2}\Delta Z_{te} \\ \sqrt{r_{le}} \\ Z_{lo} \\ \tan(2\alpha_{te} - \beta_{te})/2 \\ 0 \\ Z_{xxlo} \end{bmatrix} \tag{2.47}$$

$$\vec{a} = [X_{up}]^{-1} \vec{z_{up}}$$

$$\vec{b} = [X_{lo}]^{-1}\vec{z_{lo}}$$

2.3.3 Analytic

For fully autonomous optimization, purely analytic methods are commonly used, these methods offer a wide range of fidelity depending on the number of design variables specified. These methods can be constructive like PARSEC where an airfoil surface is assembled through a linear combination of shape functions, ϕ_i, that are defined over the airfoil's chord

$$\zeta(\xi, a_i) = \sum a_i \phi_i(\xi) \tag{2.48}$$

where $\xi = x/c$ and $\zeta = z/c$ are the non-dimensional chordwise and vertical directions, respectively. Similarly, deformative methods can be used where a set of basis functions is used to perturb an existing baseline airfoil geometry through

$$\zeta(\xi, a_i) = \zeta_{\text{baseline}} + \sum_{i=0}^{n} a_i \phi_i(\xi) \tag{2.49}$$

An advantage of deformative methods is that that they can be made to start from within the feasible domain by choosing a baseline airfoil that satisfies design constraints.

2.3.3.1 Hicks-Henne Bump Functions

A very popular set of deformative basis functions are the Hicks-Henne bump functions [30] defined as

$$\phi_i = \sin^{t_i}(\pi \xi^{mi}) \tag{2.50}$$

$$m_i = \ln(0.5)/ln(x_{M_i}) \tag{2.51}$$

Here, t_i acts to control the width of the bump function, while x_{M_i} controls the bumps location. A set of bump functions can be seen in Figure 2.7. In order to keep the parameterization linear with respect to the design variables, Wu et al. [41] held fixed x_{M_i} and t_i, only using the coefficients a_i as design variables (Equations (2.52), (2.53)).

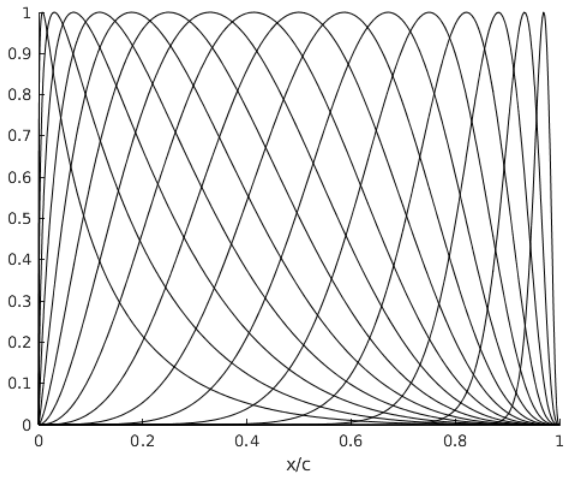


Figure 2.7: Hicks-Henne basis bump modes

$$t_i = 4 \tag{2.52}$$

$$x_{M_i} = \frac{1}{2}\Big[1 - \cos(\frac{i\pi}{n+1})\Big] \tag{2.53}$$

2.3.3.2 Bernstein Polynomials

The Bernstein polynomials are commonly used for a variety of deformative and constructive purposes due to having a number of useful properties on the interval [0;1] defined as

$$B_{i,n}(\xi) = \sum_{i=0}^{n} \binom{\xi}{i} (\xi)^i (1-\xi)^{n-i} \tag{2.54}$$

The set of 4th-order Bernstein polynomials is shown in Figure 2.8.

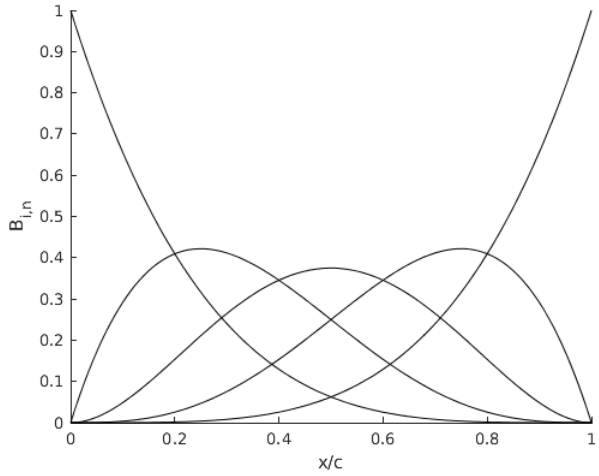


Figure 2.8: Bernstein polynomial basis modes

The degree raising property of Bernstein polynomials permits that a Bernstein polynomial of degree $n-1$ can be expressed exactly by a linear combination of polynomials of degree n through the relation

$$B_{i,n-1}(\xi) = B_{i_n}(\xi) + B_{i+1,n}(\xi) \tag{2.55}$$

allowing for the exact recovery of lower-order spaces. Desideri et al. [42] used this degree elevation as a basis for their multi-level parameterization. In addition, Vassberg et al. [37] used this property in their investigation of higher-order design spaces and showed that higher-order design spaces were guaranteed to permit a better airfoil as lower-order design spaces are exactly contained in higher-order spaces.

2.3.3.3 Orthogonal Mode Decomposition

A more novel approach to developing the basis modes is through the use of proper orthogonal decomposition [5, 32, 33]. In this method, a training set of airfoils is decomposed into a set of orthogonal modes equal to the number of airfoils in the training set. The dominant modes across the training set are then selected to form the parameterization basis modes. An example of the mode shapes generated using this approach is shown in Figure 2.9, where several libraries of training airfoils were analyzed. This allows for large numbers of airfoils to be represented by the modes;

however, the completeness of such a parameterization is also subject to the training set used.

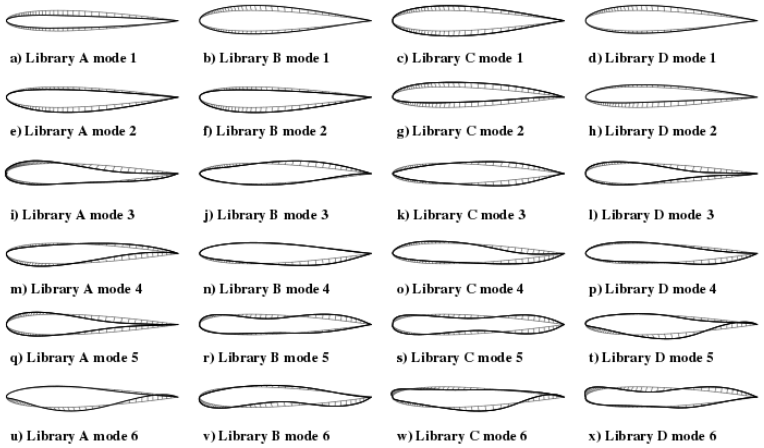


Figure 2.9: Dominant mode shapes from orthogonal mode decomposition (Ref. [5])

2.3.4 Class Shape Transformation

The Class Shape Transformation (CST) was developed by Kulfan [6] and allows one to overcome challenges posed by the infinite slope at the leading edge by enforcing geometric constraints onto the basis modes ϕ through the introduction of a class transformation C_{N2}^{N1}. The transformation is of the form

$$\zeta_{upper} = C_{N2}^{N1}(\xi)\phi(\xi) + (\xi)\Delta\zeta_{upper} \tag{2.56}$$

$$\zeta_{lower} = C_{N2}^{N1}(\xi)\phi(\xi) + (\xi)\Delta\zeta_{lower} \tag{2.57}$$

where $\Delta\zeta$ is the non-dimensional trailing-edge thickness. This addition causes $\zeta = 0$ at $\xi = 1$,removing the trailing-edge discontinuity and allowing convergence with the airfoil shape through the Stone-Weierstrass theorem. The class transformation is defined as

$$C_{N2}^{N1} = (\xi)^{N1}(1-\xi)^{N2} \tag{2.58}$$

where $N1$ and $N2$ define the shape of the leading edge and trailing edge of the airfoil, respectively. For a round leading edge and a sharp-trailing edge airfoil, such as the NACA 4-digit airfoils, values of $N1 = 0.5$ and $N2 = 1$ are appropriate.

The class shape allows the shape functions to overcome the difficulty posed by the infinite slope at the leading-edge. Different class functions and their associated modification on the unitary function can be seen in Figure 2.10.

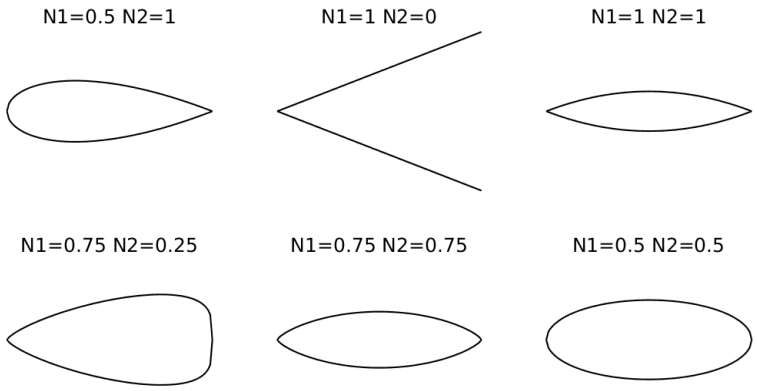


Figure 2.10: CST influence on unit function [6]

This allows for a wider range of shapes to be described by a single set of basis modes as the geometric constraint enforcement isn't a direct result of the mode shapes themselves. This also results in a reduction of the number of design variables required to sufficiently resolve the space, which is further investigated in Chapter 4.

Chapter 3
Computational Methods

In this work, performance calculations were made through the use of a main rotor (MR) analysis code known as ROTOR. ROTOR is based on a blade-element momentum theory model that uses tabular airfoil performance tables to estimate trim and performance calculations for forward-flight analysis. This code began as an autogyro performance code originally known as HELI (Ref. [43]) and was later extend to rotorcraft by Hartwich [44], before finally being modified by Kinzel [45] to include a trimming algorithm. ROTOR is used to evaluate the optimized design's performance, as well as analyze the influence of the lift coefficient correction.

The operating environment of rotorcraft airfoils necessitates a computational method capable of accurately capturing viscous effects, as well as incompressible and compressible flow for Mach numbers into the transonic regime. For this reason, the data table generation and airfoil optimization were conducted using OVERFLOW 2.2i [46].

OVERFLOW is developed and maintained by NASA. It is an implicit, structured, three-dimensional, RANS CFD code, capable of running on overset grids. Although lower-fidelity methods such as XFOIL [47] or MSES [48] are attractive, XFOIL is a potential-flow panel method with an integral boundary layer and is unable to resolve compressible flow fields required for rotorcraft applications. MSES is an Euler solver coupled with an integral boundary-layer method. While very efficient in its operation, MSES can be prone to robustness and convergence issues which require manual intervention.

CFD-based methods, however, are limited in their ability to accurately predict an airfoil's maximum lift coefficient, despite the maximum lift coefficient being a critical parameter in airfoil design. CFD solvers have difficulty predicting the

onset and growth of separated flow. Coder and Maughmer [25] analyzed the ability of several CFD solvers for single-element airfoils and noted the tendency for CFD to over-predict $c_{l,max}$. Rumsey and Ying [26] compiled results from several researchers and analyzed the ability of CFD to predict high-lift flows for multi-element airfoils and high-lift devices. These flaws are readily exploited during design and optimization loops and can ultimately result in an aircraft that fails to meet design criteria, if not properly corrected. To overcome this, a semi-empirical lift-coefficient correction routine is implemented, described in Section 3.3.4.

3.1 ROTOR Performance Evaluation Tool

ROTOR makes trim and performance calculations at a specified gross weight and forward airspeed based on the helicopter main rotor aerodynamics laid out in the previous chapter. The section lift and drag coefficients used are derived from an airfoil performance table, the generation of which is outlined in the next section. For this work, the rotor was discretized into 100 radial stations and 150 azimuth elements. To account for tip losses, ROTOR employs a 97% effective rotor radius where lift generated beyond $r/R \geq 0.97$ is neglected.

The blade motion is described by the flapping equation for a rotor in forward flight

$$I_b \frac{d^2\beta^2}{d\psi^2} + I_b\omega^2\beta = \frac{1}{\Omega^2}\int_{R_0}^{0.97R} L(r - eR)dr \tag{3.1}$$

where R_0 is the root cut-out and e is the rotor-blade hinge offset. The main rotor is trimmed by iterative adjustment of the collective θ_{1c}, lateral cyclic pitch θ_1, and longitudinal cyclic pitch θ_{1s} until the aircraft is in steady level flight i.e.

$$\begin{aligned} F_x &= \text{Drag} \\ F_y &= 0 \\ F_z &= \text{Weight} \end{aligned}$$

where F_x is in the flight direction, F_y is to the aircraft's right, and F_z is in the vertical direction.

3.1.1 Aircraft Parameters

The ROTOR code was applied to a UH-60A blade planform to serve as a baseline for analysis due to the large amount of experimental data available. The UH-60A main rotor is comprised of two different airfoils, the SC1094-R8 and the SC1095 with two main transition points. The first transition takes place at approximately $0.47R$ and the second takes place at approximately $0.85R$. The blade also has 20 deg tip sweep which begins at approximately $.92R$.

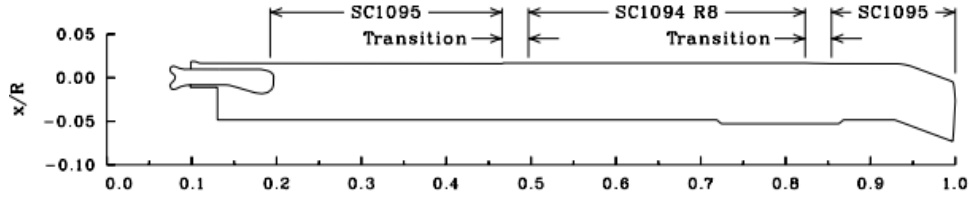


Figure 3.1: UH-60A Blade composition (Ref. [1])

The UH-60A blade includes a twist of approximately -16 deg which can be seen in Figure 3.2. There is a ± 1-degree difference in the twist between transition sections as the SC1094R8 mean chordline is rotated by 1 deg relative to the SC1095.

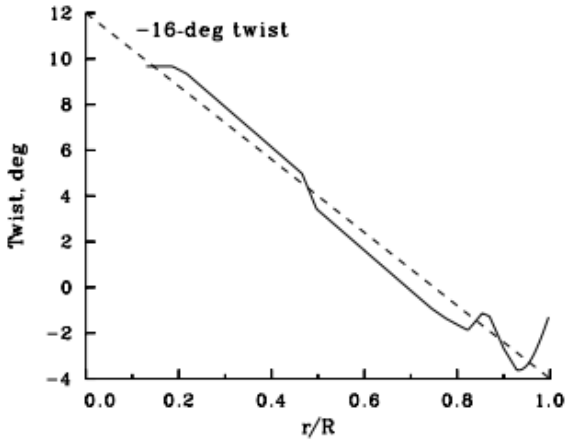


Figure 3.2: UH-60A blade twist (Ref. [1])

The gross weight of the UH-60A varies between 12000 and 24000 lbs. For this model, a weight of 16,000 lbs was chosen to correspond to the utility configuration UH60A used during collection of flight test data by [8]. Table 3.1 gives the dimensional parameters used in this work.

MR speed (rad/sec)	27.0
MR radius (ft)	26.83
MR chord (ft)	1.73
MR weight (lbs) (Ref. [49])	207.8
Density Altitude (ft^2)	5,250
Gross Weight (lbs)	16,000
Equivalent flat plate area (ft^2)	28.14

Table 3.1: UH-60A Parameters used in ROTOR

3.2 OVERFLOW CFD Code

OVERFLOW solves the Navier-Stokes equations in generalized curvilinear coordinates which are represented as

$$\frac{\partial \vec{q}}{\partial t} + \frac{\partial \vec{E}}{\partial \xi} + \frac{\partial \vec{F}}{\partial \eta} + \frac{\partial \vec{G}}{\partial \zeta} = 0 \tag{3.2}$$

where $\vec{q}$ is the conserved variable vector and E, F, G are the flux vectors containing the convective and viscous fluxes.

For many cases, the solver can be used in non time-accurate mode. However, in the stall/post-stall regime, the flow becomes unsteady and a dual-time stepping method is used. The time-accurate implementation along with low-mach preconditioning is further described by Pandya et al. in Ref. [50]. For time accuracy, dual-time stepping can be employed by recasting the equations as

$$\Gamma \frac{\partial \vec{q}}{\partial \tau} + \frac{\partial \vec{q}}{\partial t} + \frac{\partial \vec{E}}{\partial \xi} + \frac{\partial \vec{F}}{\partial \eta} + \frac{\partial \vec{G}}{\partial \zeta} = 0 \tag{3.3}$$

where τ is a pseudo-time variable added to improve convergence, and Γ is a preconditioning vector added to overcome numerical stiffness at low Mach numbers. At each physical time step n, the solution is driven to steady state in τ by iterating over the non-physical time step m, i.e. $\frac{\partial \vec{q}^{n+1}}{\partial \tau} = 0$ or $q^{m+1,n+1} = q^{n+1} + error$.

The linearized, unfactored, Euler implicit form of Equation (3.2) with dual-time

stepping is given as

$$\begin{aligned} \underbrace{\Big[I + \frac{\Delta\tau}{S_D}(\delta_x iA + \delta\eta B + \delta_\zeta C) \Big]}_{LHS} \Delta q^{n+1,m+1} \\ = \underbrace{-\frac{1+\theta}{S_D \Delta t}(q^{n+1,m} - q^n) - \frac{\theta\Delta\tau}{S_D\Delta t}\Delta q^n + \frac{\Delta\tau}{S_D} RHS^{n+1,m}}_{RHS} \end{aligned} \tag{3.4}$$

where

$$A = \frac{\partial \vec{E}}{\partial \vec{q}}, B = \frac{\partial \vec{F}}{\partial \vec{q}}, C = \frac{\partial \vec{G}}{\partial \vec{q}} \tag{3.5}$$

$$RHS^n = \frac{\partial \vec{E}^n}{\partial \xi} + \frac{\partial \vec{F}^n}{\partial \eta} + \frac{\partial \vec{G}^n}{\partial \zeta} \tag{3.6}$$

$$\tag{3.7}$$

3.2.1 Implicit Factorization

Solving Equation (3.4) requires inversion of the left-hand-side (LHS). The LHS matrix is often sparse and difficult to invert efficiently. As such, several approximate factorization techniques are implemented in OVERFLOW to reduce the computational expense of this inversion. The commonly used ARC3D solver (`ILHS=0`) Beam-Warming Alternating Direction Implicit (ADI) scheme [51] factors Equation (3.4) into a block tridiagonal form for more efficient solving. This comes at the expense of an additional factorization error. The Pulliam and Chaussee [52] pentadiagonal solver (`ILHS=2`) further decouples the system of equations into a scalar pentadiagonal matrix. This avoids the successive inversions of the block tridiagonal matrix, producing a very computationally and memory efficient discretization. Unfortunately, numerical experimentation revealed, the scalar-pentadiagonal solver was deemed too unstable for automation purposes given the high angle-of-attack solutions that must be evaluated.

For most of this work, the diagonalized diagonally-dominant ADI scheme (D3ADI) developed by Klopfer, et al. [53] was used. This method is an extension of the Diagonally Dominant ADI (DDADI) scheme of Bardina and Lombard [54]. The DDADI scheme is diagonalized by applying the Pulliam-Chaussee methodology. With the addition of Huang sub-iterations [55], the factorization error is removed,

providing a reasonable trade-off between robustness and speed (Ref. [56]).

For cases that repeatedly produce negative pressures and densities, the unfactored SSOR [57] solver (`ILHS=6`) is employed. By solving an unfactored system, the factorization error can be avoided and great robustness can be achieved. The updated Δq^{n+1} is obtained by iteratively inverting the LHS at each time step through the combination of backwards and forwards successive over relaxation sweeps. In exchange, $2.2 * \min(jd, kd, ld)$ times more memory is required than a tridiagonal ADI solver (Ref. [57]), and this solver is significantly more robust than other factored methods.

A summary of relative LHS timings is given in Table 3.2. These timings were evaluated on a 51x51x651 grid and are sensitive to processor and grid size, where 1 is the fastest relative time. A full list of solver options is available in Ref. [46].

BW (Central) ILHS=0	F3D ILHS=1	Diag ILHS=2	LU-SGS ILHS=3	D3ADI ILHS=4	(Upwind) (Upwind) ILHS=5	SSOR ILHS=6	SSOR ILHS=7
3.94	5.31	1.27	1.00	3.41	4.08	9.38	9.85

Table 3.2: Relative Solver timings (Ref. [9])

3.2.2 Spatial Discretization

For this work, the RHS was discretized using a Roe upwind scheme [58] with a 3rd-order MUSCL reconstruction (`IRHS=4, FSO=3`). This choice allows for the use of low-Mach preconditioning while maintaining both numerical accuracy and robustness. In addition, when using the SSOR or D3ADI LHS, the solver is able to converge without the need for artificial dissipation (`DIS2=0.0, DIS4=0.0`). The Roe upwind scheme defines the interface flux through

$$F_{i+1/2} = \frac{1}{2}(F^L_{i+1/2} + F^R_{i+1/2}) - \frac{1}{2}|\bar{A}|_{i+1/2}(Q^R_{i+1/2} - Q^L_{i+1/2}) \tag{3.8}$$

Higher-order accuracy is obtained through the application of an MUSCL interpolation scheme to reconstruct the left (L) and right (R) fluxes. Several flux limiters are available to switch the algorithm to first order in regions of strong gradients, allowing for better shock capturing and preventing overshoots. For this

work, a 3rd-order MUSCL reconstruction scheme (`FSO=3`) with the Koren limiter (`ILIMIT=1`) was used.

3.2.3 Turbulence Closure

Closure was obtained using the one-equation linear eddy-viscosity model of Spalart-Allmaras [59].

$$\begin{aligned}\frac{D\tilde{\nu}}{Dt} = c_{b1}\tilde{S}\tilde{\nu}(1-f_{t2}) - \Big[c_{w1}f_2 - \frac{c_{b1}}{\kappa^2}f_{t2}\Big]\Big(\frac{\tilde{\nu}}{d}\Big)^2 + \\ \frac{1}{\sigma}\Big[\frac{\partial}{\partial x_j}\Big((\nu+\tilde{\nu})\frac{\partial\tilde{\nu}}{\partial x_j}\Big) + c_{b2}\Big(\frac{\partial\tilde{\nu}}{\partial x_j}\Big)^2\Big]\end{aligned} \tag{3.9}$$

To account for the influence of laminar-turbulent transition, the Amplification Factor Transport transition (AFT) model of Coder and Maughmer [60] was employed as well.

$$\frac{\partial(\rho\tilde{n})}{\partial t} + \frac{\partial(\rho u_j\tilde{n})}{\partial x_i} = \rho\Omega F_{crit}F_{growth}\frac{d\tilde{n}}{dR_{\delta 2}} + \frac{\partial}{\partial x_j}\Big[(\mu+\frac{\mu_t}{\sigma_n})\frac{\partial\tilde{n}}{\partial x_j}\Big] \tag{3.10}$$

3.2.4 Grid Generation

The grids used in this work were generated in accordance with *Best Practices in Overset Grid Generation* (Ref. [61]). Two types of grids were used in this work. For airfoils with a sharp trailing edge, a C-grid was used. For finite-thickness trailing-edge airfoils, an O-grid topology was employed. Surface grids were first generated using an in-house algorithm using natural spline interpolation with a clustering of points near the leading edge and trailing edge of the airfoil. For C-grids, a wake cut was made using the Chimera Grid Tools (CGT) [62] utility `WKCUT`.

The volume grids were generated using CGT through the hyperbolic grid generation tool `HYPGEN`. An exponential stretching function was employed in the wall normal direction. Initial and ending spaces were chosen along with the number of points to maintain a stretching ratio of less than 1.2 to avoid large jumps in grid spacing. The stretching ratio is the ratio between the current and the previous grid increment

$$\frac{\max\left[\Delta s_j, \Delta s_{j+1}\right]}{\min\left[\Delta s_j, \Delta s_{j+1}\right]} \tag{3.11}$$

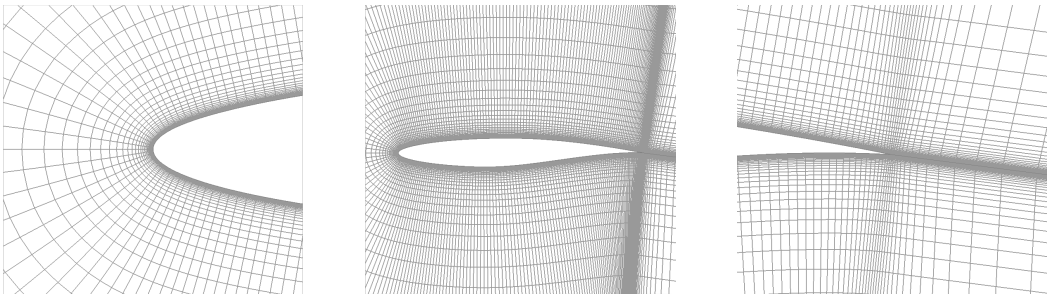
The first off-body grid point was chosen to maintain a y^+ value of approximately 1 to improve viscous drag measurements. The first off-body-grid point is estimated through

$$y = \frac{y^+}{Re\sqrt{c_f/2}} \tag{3.12}$$

where c_f is the approximate skin friction coefficient for a turbulent flat plate given by

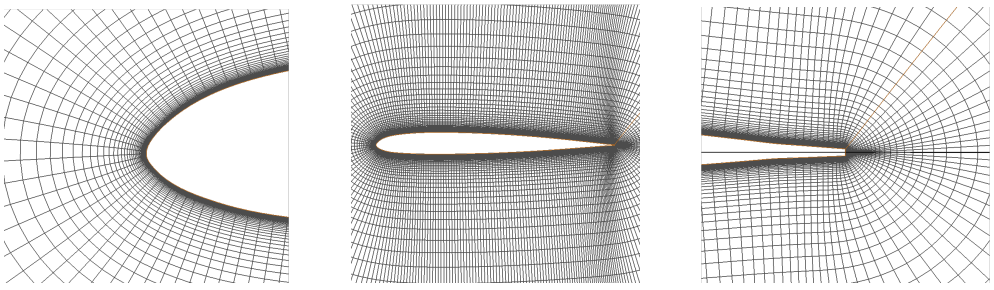
$$c_f \approx \frac{0.455}{\ln^2(0.06\mathrm{Re}_x)} \tag{3.13}$$

with a reference length of 10% of an airfoil chord. A total of 129 grid points were used in the wall normal direction, and a total of 259 points were used on the airfoil surface. For C-grids, a total of 98 grid points were placed in the wake. For O-grid topologies, 20 grid points were placed on the airfoil trailing edge.



(a) leading-edge (b) C-grid (c) trailing-edge

Figure 3.3: C-grid around RAE 2822 airfoil



(a) leading-edge (b) C-grid (c) trailing-edge

Figure 3.4: O-grid around SC1095 airfoil

3.3 Performance Table Generation

A Python framework named overPy was developed to conduct the performance table generation. Python programming was used to handle the high-level logic, while the procedural computation and OVERFLOW output file interfaces were programmed using FORTRAN. The third-party Python library Numpy was used as well. NumPy contains a set of efficient numerical tools for Python, as well as a module F2Py which allows for the generation of a Python interface with FORTRAN compiled shared object files. This allows for Python functions to interface with FORTRAN subroutines as though they were native objects eliminating the need for reading and writing of intermediate text files to exchange information between subroutines.

The automation routines were used to link various utilities to perform the preprocessing, job submission, and postprocessing of OVERFLOW jobs. This methodology allows for the development of a more complex set of run logic than available purely through the OVERFLOW solver.

3.3.1 Preprocessing

In the preprocessing routine, a directory tree is generated to manage each case. The required files for an OVERFLOW run are then scattered to each directory. The input files include

- Suitable grid file
- Input file for OVERFLOW
- Queue manager job script
- Input file for a FOMOCO preprocessing utility

Examples of these input files can be found in Appendix A. Calculation of force and moments also require initialization of integration surfaces before starting OVERFLOW. For this work, the pre-/post-processing tool USURP [63] was used; however, alternatives can be used provided they produce the required `panel_weights.dat` files. The preprocessing procedure is as follows

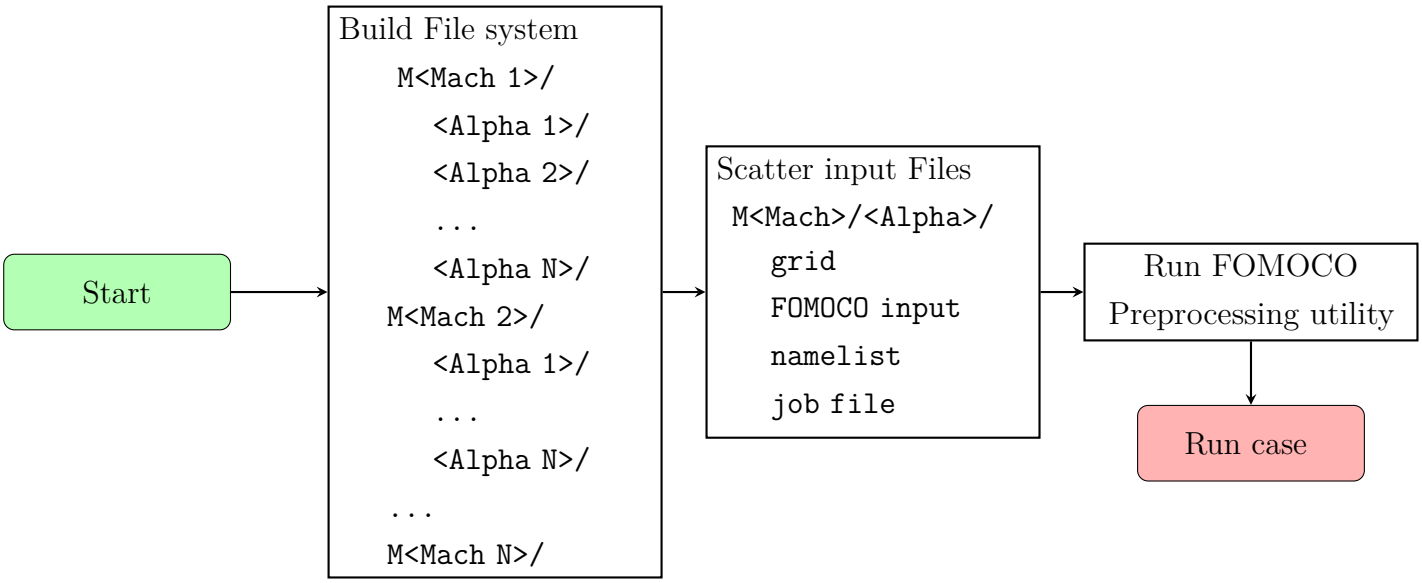


Figure 3.5: Airfoil performance table preprocessing routine

3.3.2 Run Process

The run routine navigates to each case directory and submits the job script to the queue manager via the `qsub` command. A user-specified maximum number of concurrent jobs are managed at any given time. Cases are cycled to avoid overloading the queue manager with job submissions. To determine a job's completion, the queue manager is pinged periodically using the `qstat` command for each submitted job until the job can no longer be located.

After job completion, output files are verified and a set of logic is employed to improve convergence and recover from unsuccessful runs. If a negative pressure or density is found in the output file, then the run is deemed unsuccessful. To attempt a recovery, an angle-of-attack sequencing (ASEQ) routine is first employed. By reducing the airfoil's angle until convergence is achieved and then re-sequencing to the original angle, often times convergence can be achieved within a few degrees.

After a successful run, a restart routine is employed to improve convergence. The force and moment coefficient history is read from the `fomoco.out` file. If the lift, drag, or moment standard deviations exceed a set of tolerances, the case is restarted in an attempt to improve convergence. The default convergence tolerances chosen are 1×10^{-4}, 1×10^{-5}, and 1×10^{-5} for the lift, drag, and pitching moment

coefficients standard deviations, respectively.

Finally, if the ASEQ routine is unable to succeed or if the case fails during a restart, a final failure routine is employed. An alternative set of more robust, albeit slower, NAMELIST options are then employed in hopes of gaining convergence. A flow chart of the run routine can be seen in Figure 3.6

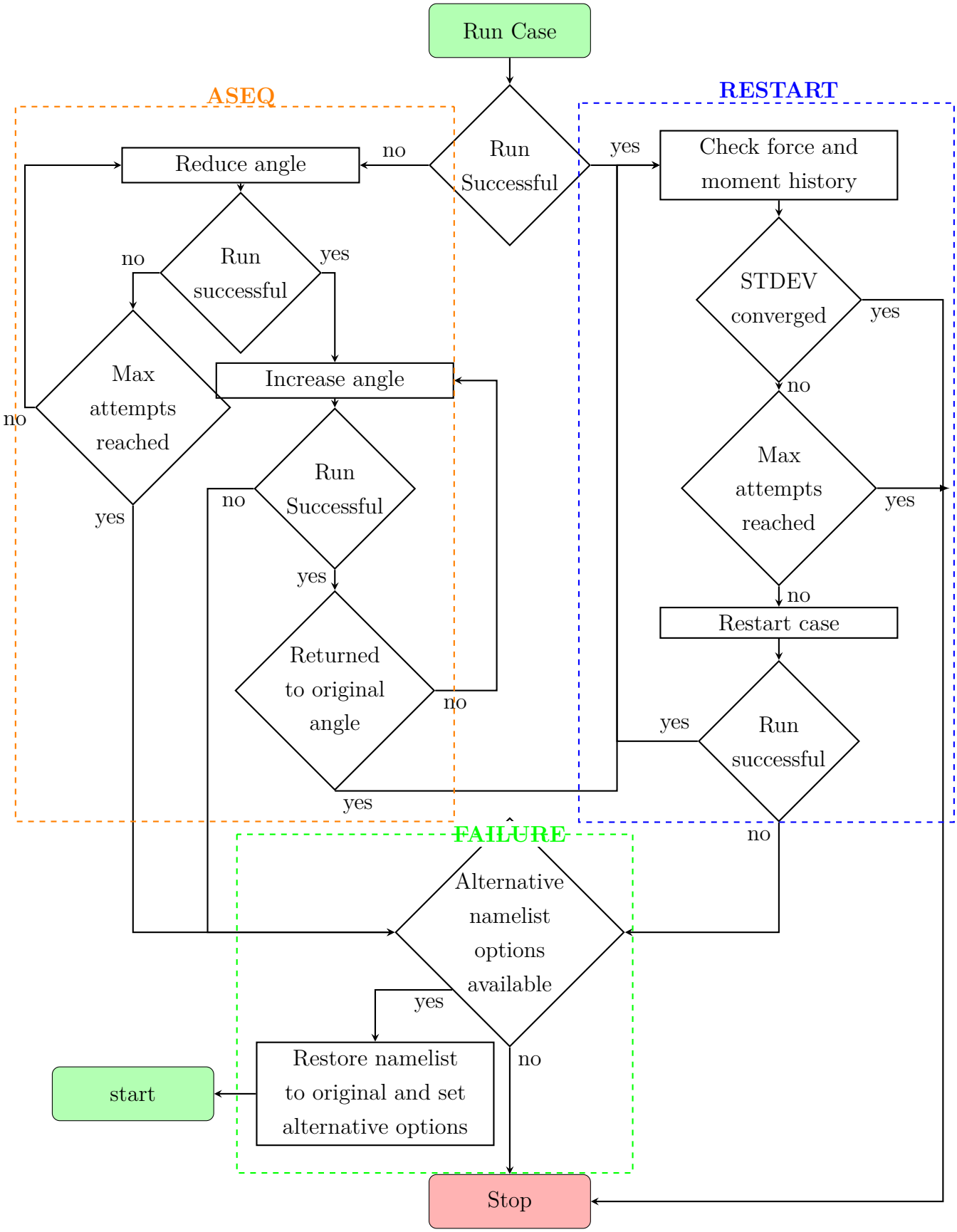


Figure 3.6: Airfoil performance table run routine

3.3.3 Postprocessing

During the postprocessing routine lift, drag, and moment coefficients are read from the force and moment history file `fomoco.out`. Each case directory is cross-validated with the flow solution file (`q.save`) to make sure that the correct Mach number and angle-of-attack values are represented. C81 tables require the computation of incompressible flow solutions (Mach=0.0). OVERFLOW is incapable of running this condition, the incompressible Mach behaviors are estimated using the Prandtl-Glauert correction with the lowest available Mach as a reference.

$$c_{l_{M_0}} = c_{l_{M_{min}}} \sqrt{1 - M_{min}^2} \tag{3.14}$$

$$c_{d_{M_0}} = c_{d_{M_{min}}} \sqrt{1 - M_{min}^2} \tag{3.15}$$

$$c_{m_{M_0}} = c_{m_{M_{min}}} \sqrt{1 - M_{min}^2} \tag{3.16}$$

To offload some of the computational expense, experimentally generated NACA0012 data are used to supplement CFD generated results for high angles-of-attack ($|\alpha| \geq 30$ deg) [64]. At these angles-of-attack, the assumption of two-dimensional CFD begins to break down as three-dimensional effects begin to take place. In addition, the airfoil shape begins to have less influence on the flow characteristics as it begins to behave more similar to a wall than an airfoil. Smith et al. [65] showed that this behavior in the deep-stall regime was similar between airfoils regardless of camber.

Finally, a lift-coefficient correction routine is implemented, the specifics of which are discussed in the next chapter. The post-processing procedure is as follows:

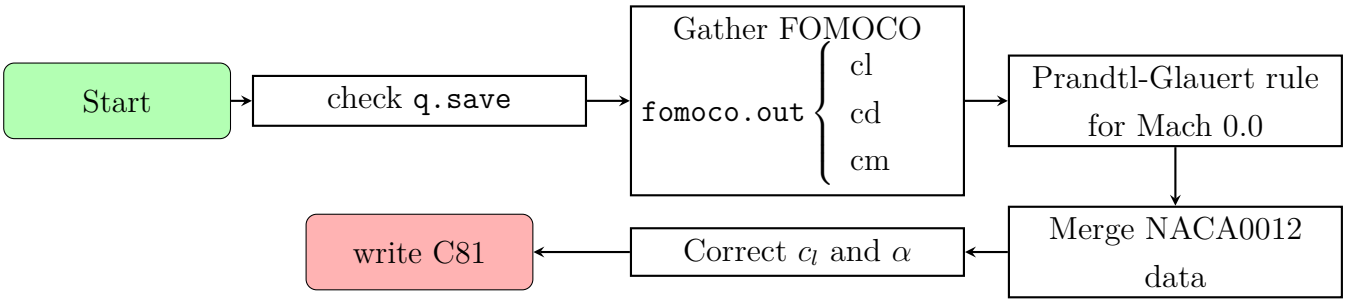


Figure 3.7: Airfoil performance table postprocessing routine

3.3.4 Lift Coefficient Correction

The inability for CFD to accurately predict the maximum lift coefficient creates a flaw that may be exploited by optimization loops potentially producing unrealizable performance results. A $c_{l,max}$ criterion and correction routine were then implemented to overcome this.

3.3.4.1 Stall Criterion

Several maximum lift critera have been proposed that are traditionally based on boundary-layer development on the upper surface. A.M.O Smith [66] proposed a maximum lift coefficient criterion based on the maximum pressure coefficient and Mach number. Smith proposed placing a limit on the leading-edge suction peak of

$$c_{p,max} = \frac{1}{0.7M_\infty^2}\{[(1+0.2M_\infty^2)/1.2]^3.5 - 1\} \tag{3.17}$$

Valarezo and Chin [7] proposed a surface pressure criterion for high-lift systems that included both Mach number and Reynolds number influences. The criterion is based on the pressure difference between the suction peak near the leading-edge and the trailing-edge pressure.

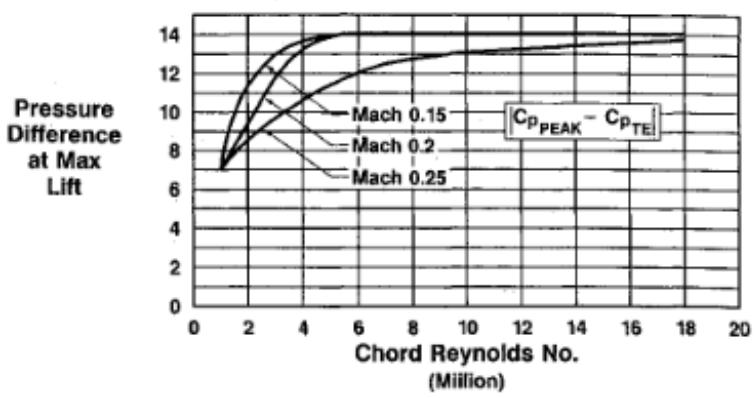


Figure 3.8: Pressure difference rule for maximum lift (Ref. [7])

A multitude of other methods for maximum lift prediction have been proposed; however, these methods often depend on the evaluation method used. In order to develop a stall criterion that will work across a variety of methods, a stall criterion based explicitly on boundary-layer properties was implemented. The

boundary-layer properties of interest are the displacement thickness δ_1 and the momentum thickness δ_2.

$$\delta_1 = \int_0^\infty (1 - \frac{\rho u}{\rho_e U_e}) dz \tag{3.18}$$

$$\delta_2 = \int_0^\infty \frac{\rho u}{\rho_e U_e} (1 - \frac{u}{U_e}) dz \tag{3.19}$$

These quantities were calculated using OVERFLOW generated flow solutions using the methodology described by Coder and Maughmer in Ref. [67]. A baseline set of airfoil cases for which high-quality experimental data are available were evaluated at lift coefficients equal to the experimentally observed maximum lift coefficients.

The behavior of the displacement-thickness Reynolds number (Re_{δ_1}), momentum-thickness Reynolds number (Re_{δ_2}), upper-surface drag contribution, and the upper-surface velocity ratio between the suction peak and trailing-edge for the base case of airfoils are plotted in Figures 3.9 through 3.11, respectively, for varying chord Reynolds number.

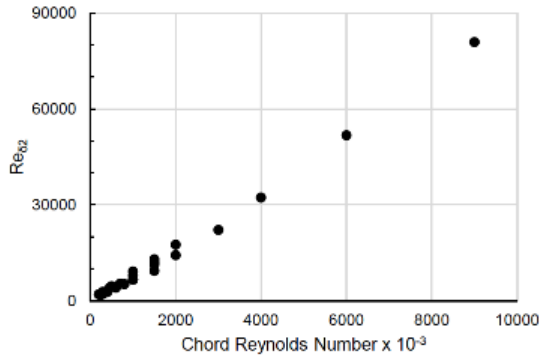


Figure 3.9: Predicted upper-surface, trailing-edge Re_{δ_2} at $c_{l,max}$

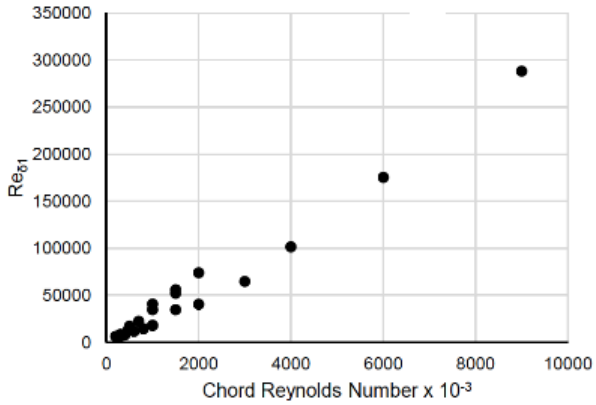


Figure 3.10: Predicted upper-surface, trailing-edge Re_{δ_1} at $c_{l,max}$

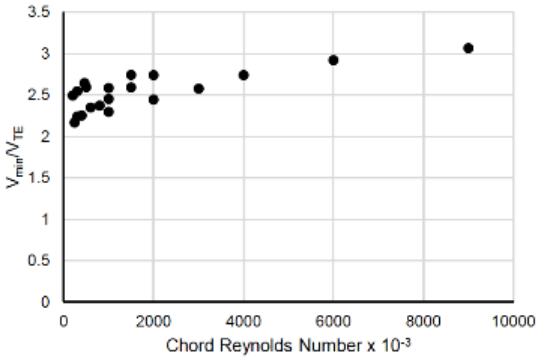


Figure 3.11: Predicted velocity recovery ratio at $c_{l,max}$

Of these quantities, the momentum-thickness Reynolds number shows the strongest correlation with Reynolds number. The displacement-thickness Reynolds number and the upper-surface profile drag show the presence of two behavioral branches, which is an undesirable non-uniqueness in an empirical correlation. The upper-surface velocity ratio shows scatter; however, there appears to be a distinct upper limit to the velocity ratio that may be achieved for maximum lift at a given Reynolds number confirming Valarezo and Chin's observations.

The maximum lift criterion is then chosen as the sectional lift coefficient for which the upper-surface, trailing-edge momentum-thickness Reynolds number first satisfies

$$Re_{\delta_2,TE} = 8760 \frac{Re}{1 \times 10^6} \tag{3.20}$$

The momentum thickness at the trailing-edge is reflective of the momentum

losses in the boundary layer, and the underlying physics of the correction reflects the notion that a boundary layer can only recover so much pressure in a given distance before it separates.

3.3.4.2 Correction Implementation

The lift coefficient criteria implemented acts as a mapping between the original coefficient space $c_l(\alpha)$ and the corrected coefficient space $c_l'(\alpha')$. A similar mapping was implemented by Coder in Ref. [68]; however, in the present work, additional care was given to preserve zero-lift angle-of-attack (α_0) as well as the lift curve slope ($\frac{dC_l}{d_\alpha}$), where CFD has been shown to accurately predict these values for quasi-steady airfoils by Smith et al. [69], and the same trend was also observed in the present work. The correction is applied through a linear mapping that acts as a constant scaling centered about α_0.

$$c_l' = \kappa c_l$$
$$\alpha' = \kappa(\alpha - \alpha_0) + \alpha_0$$

where the correction factor κ is defined as

$$\kappa = \frac{c_{l,max,experimental}}{c_{l,max,CFD}}$$

An example of this mapping is shown in Figure 3.12 where $\frac{dc_l}{d_\alpha}$ and α_0 are maintained exactly throughout the mapping, while the linear departure angle and stall characteristics are scaled by the correction factor κ.

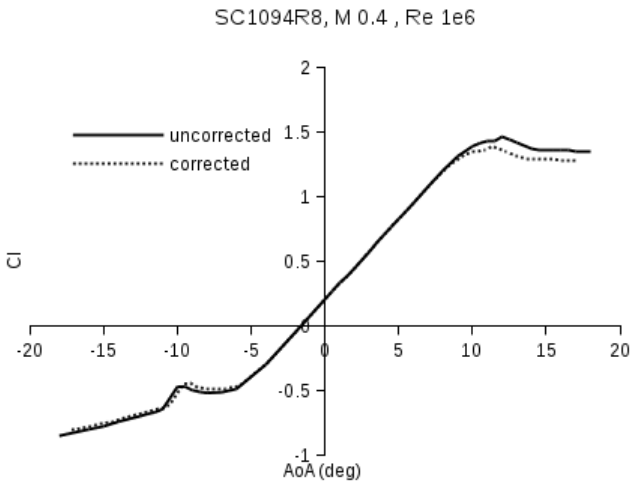


Figure 3.12: Influence of lift coefficient correction on raw CFD results, SC1095, $Re = 6 \times 10^6$, $M_\infty = 0.4$

Further Enhancement

A piece-wise stall criterion was also introduced to correct both the positive stall and negative stall independently. Rather than only using the upper surface stall, the same criterion can be implemented on the negative stall regime by integrating the momentum thickness Reynolds number on the lower surface. The final form of the correction becomes

$$c_l' = \begin{cases} \kappa_1 c_l & \text{if } c_l \geq 0 \\ \kappa_2 c_l & \text{if } c_l < 0 \end{cases} \tag{3.21}$$

$$\alpha' = \begin{cases} \kappa_1(\alpha - \alpha_0) + \alpha_0 & \text{if } c_l \geq 0 \\ \kappa_2(\alpha - \alpha_0) + \alpha_0 & \text{if } c_l < 0 \end{cases} \tag{3.22}$$

where

$$\kappa_1 = \frac{c_{l,max,experimental}}{c_{l,max,CFD}} \tag{3.23}$$

$$\kappa_2 = \frac{c_{l,min,experimental}}{c_{l,min,CFD}} \tag{3.24}$$

Should the correction fail and $\kappa_1 > 1$, or $\kappa_2 > 1$, then the original, uncorrected table values are used.

Chapter 4
Airfoil Optimization

4.1 Design Variables

The design variables in this work, were constructed through a CST-based approach, although Kulfan [6] originally suggested the use of Bernstein polynomials basis modes. The Bernstein polynomials suffer from multicollinearity issues which are investigated in the next section. Instead, a set of orthogonal basis modes are used, namely the Legendre polynomials. Although Farouki [70] developed a set of Bernstein-to-Legendre basis transformations with decent conditioning of the transformation matrix for use in finit- element schemes, the original Legendre polynomials themselves are used in this work, defined by Rodrigues' formula as

$$P_n(\eta) = \frac{1}{2^n} \sum_r (-1)^r \frac{(2n-3r)!}{r!(n-r)(n-2r)!} \eta^{n-2r} \tag{4.1}$$

The Legendre polynomials satisfy the orthogonality condition

$$\int_{-1}^{1} P_m(\eta) P_n(\eta) d\eta = \frac{2}{2n+1} \delta_{mn} \tag{4.2}$$

The first 4 Legendre modes are shown in Figure 4.1

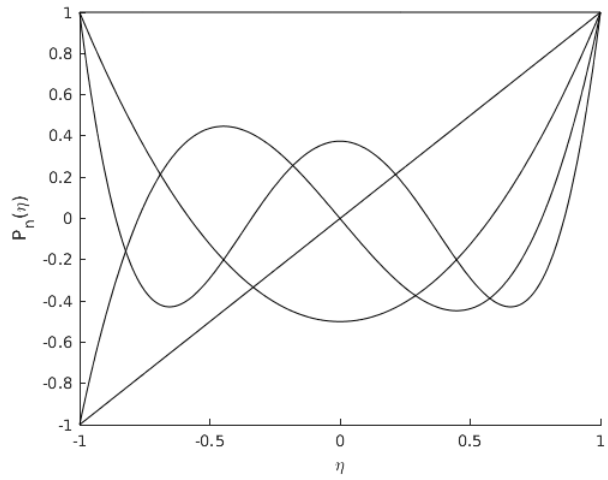


Figure 4.1: Legendre polynomial modes

The parameterization was implemented as a deformative method by first "unwrapping" the baseline airfoil about the leading edge and evaluating the Legendre polynomials on the interval [-1;1]. This unwrapped domain is denoted η in this work where the original, non-dimensional, chordwise space is denoted ξ. The basis modes were wrapped to the original interval and the CST was applied in the original airfoil domain of [0;1]. Figure 4.2 shows the first four Legendre modes in the unwrapped domain with the inclusion of the CST.

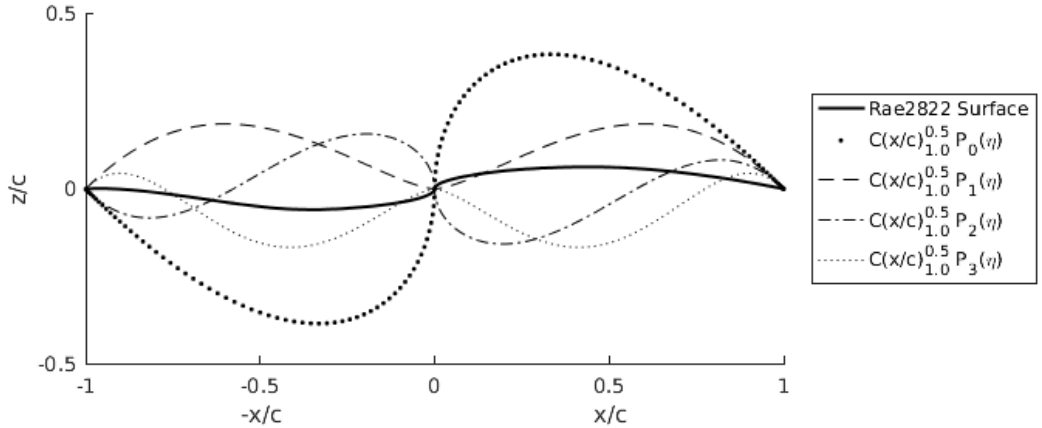


Figure 4.2: Unwrapped RAE 2822 compared with parameterization basis modes

As a result of the unwrapping, the even-numbered Legendre modes are symmetric about the leading edge, causing them to influence thickness. Similarly, odd-

numbered Legendre modes are antisymmetric about the leading edge, causing them to influence camber. This can be visualized in Figure 4.3 where the airfoil is transformed back to the original domain

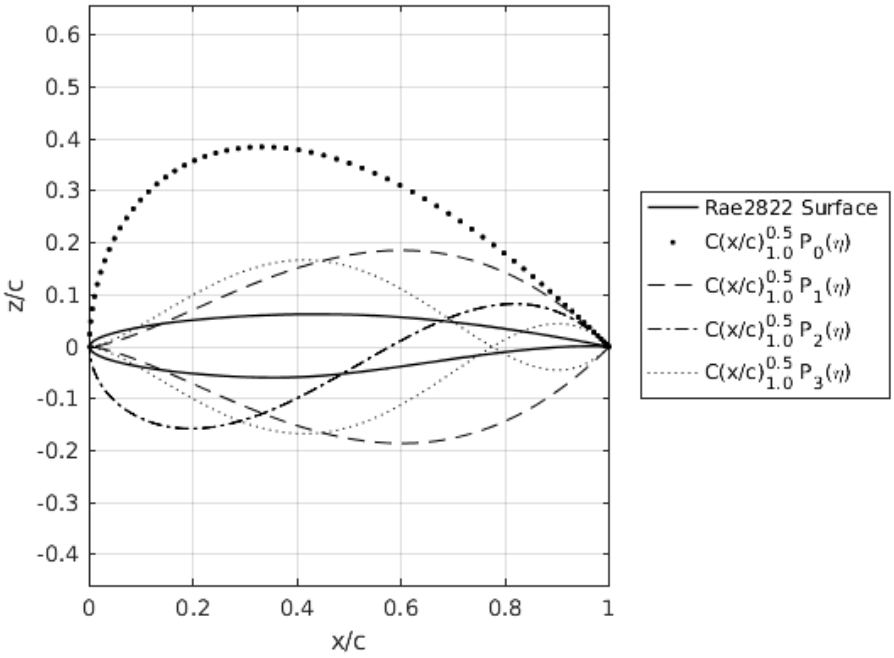


Figure 4.3: Wrapped airfoil parameterization

The even-numbered modes are mirrored across the x-axis, while the camber modes overlap each other. In addition, CST forces the thickness modes to maintain the infinite slope required at the leading edge for a round-nosed airfoil. Defining the parameterization in this method allows for a single set of modes to define the upper and lower surfaces. In summary, the deformative parameterization used in this work, including the unwrapping is

$$\zeta_{upper} = \zeta_{upper,base} + C_{N2}^{N1}(\xi) \sum_{i=1}^{n} a_i P_i(\eta) + (\eta)\Delta\zeta_{upper} \tag{4.3}$$

$$\zeta_{lower} = \zeta_{upper,base} + C_{N2}^{N1}(\xi) \sum_{i=1}^{n} a_i P_i(\eta) + (\eta)\Delta\zeta_{lower} \tag{4.4}$$

4.1.1 Multicollinearity

Multicollinearity is a result of the increasing similarity between basis mode shapes as their order increases. In a mathematical sense, this causes the design-matrix to become sparse and ill-conditioned as non-unique solutions begin to develop. In an optimization sense, this results in a damping of the design space where large changes in design variables may result in small net geometric perturbations. This makes it difficult for the optimizer to efficiently navigate the design space as the optimization path length becomes highly dependent on the number of design variables.

When investigating the extension of Bernstein polynomial parameterizations to higher-orders, Vassberg et al. [37] noted an initial decrease in performance, despite the lower-order design space being exactly contained in the higher-order-space. While they were able to overcome this by extending the bounds of the design space, this suggests that the problem may be a result of multicollinearity between the basis modes of the Bernstein polynomials.

As an extreme case, consider the monomial basis $a_0 + a_1x + a_2x^2 + ...a_nx^{n-1}$. This basis produces a highly correlated set of design variables. Looking at Figure 4.4, as the order increases the basis mode shapes become increasingly similar until eventually, there is very little difference between individual modes and linear independence is lost.

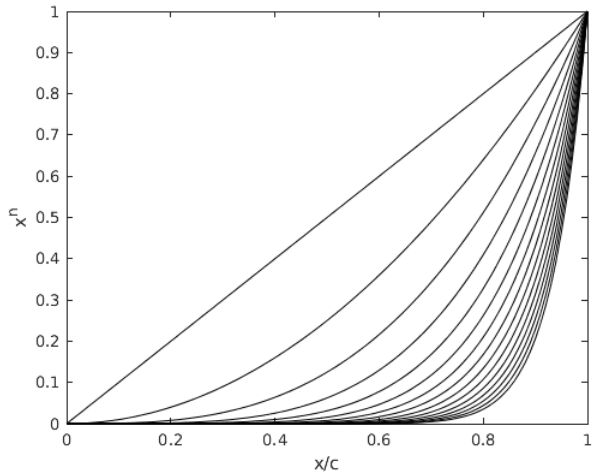


Figure 4.4: Monomial basis modes

One of the most common measures of multicollinearity is the condition index or

condition number, developed by Belsey [71], defined by

$$c = \sqrt{\frac{\lambda_{max}}{\lambda_{min}}} \tag{4.5}$$

where the λ_{max} and λ_{min} are the largest and smallest eigenvalues, respectively. The condition number gives insight as to the invertability of a matrix. In this case, the matrix being measured is the design or regressor matrix that defines the least squares regression problem, evaluating the shape functions on the interval [0;1] at m discrete points. A high condition number means that large perturbations in the design coefficients are required to produce small perturbations in the resultant geometry and vice versa here

$$[z] = [X]\vec{a} \tag{4.6}$$

$$\begin{bmatrix} z(\xi_0) \\ z(\xi_1) \\ \vdots \\ z(\xi_m) \end{bmatrix} = \begin{bmatrix} X_0(\xi_0) & X_1(\xi_0) & \dots & X_n(\xi_0) \\ X_0(\xi_1) & X_1(\xi_1) & \dots & X_n(\xi_1) \\ \vdots & \ddots & \vdots & \\ X_0(\xi_m) & X_1(\xi_m) & \dots & X_n(\xi_m) \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} \tag{4.7}$$

where for CST-based methods

$$X_{i,j} = C_{N2}^{N1}(\xi_j) S_i(\xi_j) \tag{4.8}$$

A summary of shape functions used for this analysis can be found in Table 4.1. For comparison, the shifted Legendre Polynomials $P_n = P_n(2\xi - 1)$ were used to be orthogonal on the interval [0, 1] rather than their standard [-1;1]. When using CST methods, $N1$ and $N2$ were chosen to be 0.5 and 1, respectively.

Bernstein Polynomials	$X_i(\xi_j) = \sum_{i=0}^{n} \binom{n}{i} (\xi_j)^i (1-\xi_j)^{n-i}$
Hicks-Henne Bumps	$X_i(\xi_j) = \sin^4(\pi \xi_j^{m_i})$ $m_i = \frac{\ln(0.5)}{\ln(\frac{1}{2}\left[1-\cos(\frac{i\pi}{n+1})\right])}$
Legendre Polynomials	$X_i(\xi_j) = \frac{1}{2^i i!} \frac{d^i}{d\xi^i} (\xi_j^2 - 1)^i$

Table 4.1: Summary of parameterization investigation methods

The following figure compares the condition values of several analytic parame-

terization methods over the interval [0,1]. A total 250 points were generated across the interval.

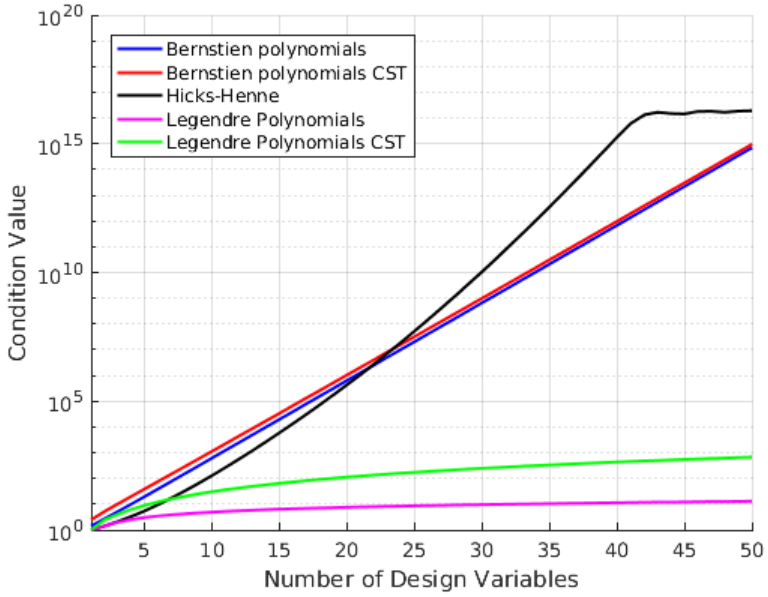


Figure 4.5: Condition value comparison

Generally, a condition value of 30 is considered to have high degrees of multicollinearity while condition values over 100 are considered severe [72]. On the extreme end, condition values of up to 10,000 are numerically manageable. Beyond that point, the design matrix becomes too sparse to invert normally and a pseudo inverse must be calculated instead. For the Hicks-Henne and Bernstein polynomial shape functions, this puts a numerical limit on the maximum number of design variables near 30. The Legendre polynomial design matrices remained invertible even up to 50 design variables.

Looking at Figure 4.6 it can be seen that although the class transformation adds a considerable amount of multicollinearity to the Legendre polynomials, it greatly reduces the error in approximating the upper surface of the RAE2822 airfoil by allowing the Legendre basis modes to overcome difficulties posed by the infinite leading edge slope. Although this does increase the condition number greatly, it acts to damp solutions that would likely be unsuccessful anyway. It also reduces the overall number of design variables required to sufficiently resolve the design space.

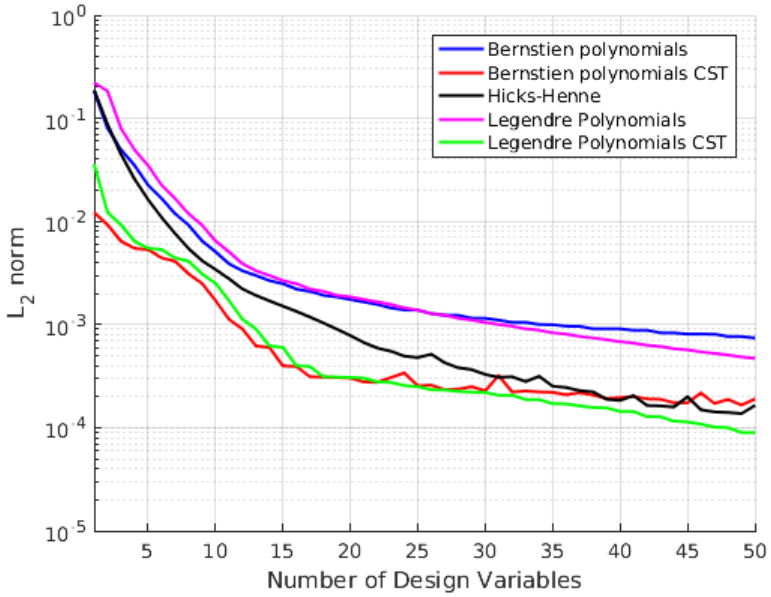


Figure 4.6: RAE2822 Upper Surface Error

The Bernstein polynomials begin to exhibit fluctuations at higher-order design spaces. This is a result of the increasing condition number making inversion of the design matrix difficult and unstable, despite using a More-Penrose pseudo inverse to perform the calculation.

4.1.2 Completeness

Another important property of airfoil parameterization is the completeness of the design space. A parameterization method should be able to represent a large number of airfoils. To analyze this, the UIUC airfoil database [73] was analyzed using each of the parameterization methods from before. The resulting L_2 error norm was calculated for each airfoil. The solid lines in Figure 4.7 show the averaged $\log_{10}(L_2 error$ norm).

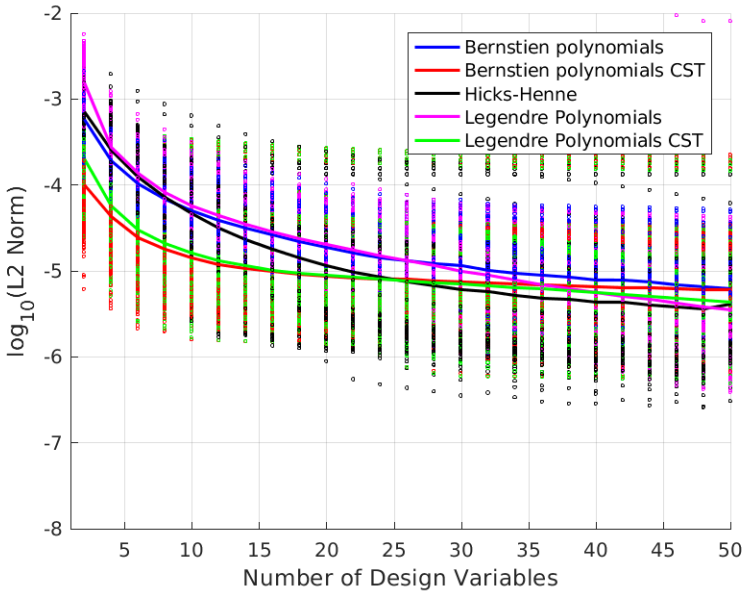


Figure 4.7: Parameter comparison average $\log_{10}$ of L_2 error norm

To remove any issues from variance in airfoil point distribution, each airfoil was smoothed with a cubic spline and interpolated with 150 points on each surface using a cosine distribution. The upper and lower surfaces were fit independently, approximately 500 airfoils of 1500 in the UIUC database were randomly selected and analyzed.

The Legendre-CST parameterization performs very closely to the Bernstein polynomial-CST approach but without the large levels of multicollinearity that are present in the Bernstein polynomials. Both methods do a fair job at covering the design space with as few as 5-10 design variables. This completeness evaluation was applied on the interval [0;1] and therefore only represents the ability of the parameterization to fit the upper and lower surfaces independently, i.e. twice the number of design variables are required to fit both surfaces.

The actual parameterization is applied on the interval [-1;1] so an additional completeness test was carried out on that interval. Comparing the ability of the Legendre polynomials and the shifted Bernstein polynomials to parameterize an entire airfoil, the same point distribution as before was used with a total of 300 points on the combined upper and lower surfaces.

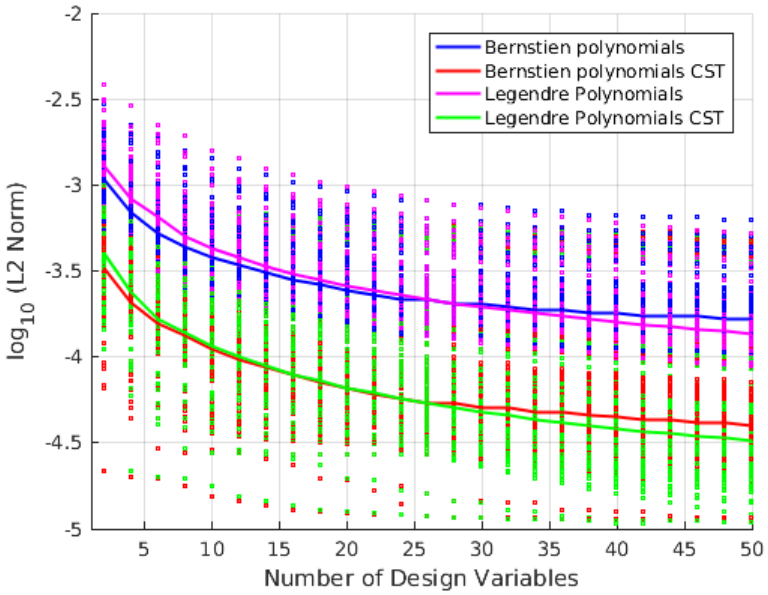


Figure 4.8: Unwrapped parameterization comparison average $\log_{10}$ of L_2 error norm

An investigation of the error locations across the chordwise direction yields that the CST reduces the residual regression across the entire chord with a particularly large decrease in error at the leading edge (Figure 4.9). For this comparison, 10 design variables were used to fit the same 500 randomly selected airfoils from the UIUC airfoil database, and the vertical axis again averages the $\log_{10}$ normalized residual errors.

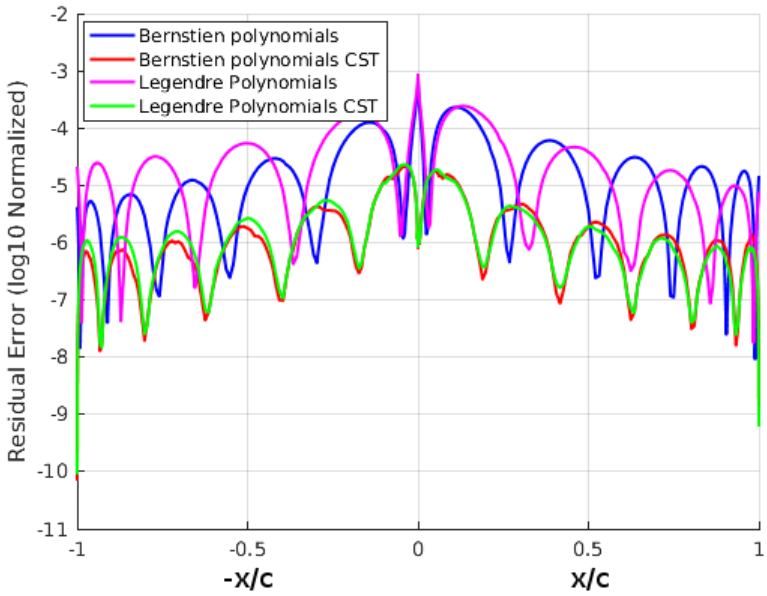


Figure 4.9: Unwrapped parameterization surface residual comparison

4.2 Optimization Method

4.2.1 Covariance Matrix Adaptation

The optimization algorithm used to perform the shape optimization was the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [29, 74, 75]. This method was designed as a black box optimization approach for large non-linear problems. It handles ill-conditioning better than gradient-based methods. In addition, it is easily parallelized and well suited for problems with small sample populations. This makes it a good choice for academic CFD problems with limited computational power as the objective functions take a long time to evaluate. It also allows for the optimization of nosier objective functions as it does not rely on the explicit computation of the objective function's gradients.

As the name suggests, the Covariance Matrix Adaptation algorithm attempts to find a global optimum by successively updating the covariance matrix at each design iteration. This covariance matrix is used to generate candidate solutions based on multivariate normal distributions. Given a mean design vector $y^{(k)}_{mean}$, search points are sampled as a normal distribution with variance $(\sigma^{(k)})^2 C^{(k)}$ of

population size ϵ

$$y_i^{(k)} \sim \mathcal{N}(y_{mean}^{(k)}, (\sigma^{(k)})^2 C^{(k)}) \tag{4.9}$$

where $y_{mean}^{(k)}$ is the current favorite solution vector at iteration k, and C is the covariance matrix which governs the shape of the distribution ellipsoid. The candidate solutions are then evaluated by the loss function $L(y)$. The solutions are weighted, and only the best ν solutions are chosen to influence the new search path. Here

$$y_{mean}^{(k+1)} = \sum_{i=1}^{\nu} w_i y_i$$

where

$$L(y_1) < L(y_2) < L(y_3)...L(y_\epsilon)$$
$$\nu \leq \epsilon$$

The covariance matrix is updated as a weighted combination of the old covariance matrix, the optimum search path (rank 1 update), and the weighted sum of the best ν candidate solutions (rank ν update). The evolution path is updated according to

$$p_c^{(k+1)} = (1 - c_c) p_c^{(k)} + \sqrt{c_c(2 - c_c)\nu_{eff}} \frac{y_{mean}^{(k+1)} - y_{mean}^{(k)}}{\sigma} \tag{4.10}$$

where

$$\nu_{eff} = \frac{1}{\sum_{i=1}^{\nu} w_i^2} \tag{4.11}$$

and $(y_{mean}^{(k+1)} - y_{mean}^{(k)})/\sigma^{(k)}$ defines the new evolution path in the direction of the new y_{mean}, and $\sqrt{c_c(2 - c_c)\nu_{eff}}$ is a normalization factor. If $c_c = 1$ and $\nu_{eff} = 1$, the path reduces to $p_c = (y_{mean}^{(k+1)} - y_{mean}^{(k)})/\sigma^{(k)}$ which is simply the difference between the last and updated y_{mean}, scaled by the path step size $\sigma^{(k)}$. The covariance matrix is updated according to

$$\begin{aligned} C^{(k+1)} &= (1 - c_1 - c_\nu) C^{(k)} + c_1 p_c^{(k+1)} (p_c^{(k+1)})^T \\ &+ c_\nu \sum_{i=1}^{\nu} w_i \frac{(y_i^{(k+1)} - y_{mean})}{\sigma^{(k)}} \frac{(y_i^{(k+1)} - y_{mean})^T}{\sigma^{(k)}} \end{aligned} \tag{4.12}$$

where c_1 and c_ν are scaling parameters that control how much the rank-one and rank-ν updates will influence the updated covariance matrix, respectively. The rank-one update is driven by the updated evolution path, while the rank ν update is a result of the best ν solutions.

Although not necessary, CMA-ES includes a step-size control method for whose evolution path p_σ is updated independently of the covariance matrix with the intent of increasing the step size σ when cumulative steps are in similar directions and decreasing steps when the search paths are in opposite directions.

$$p_\sigma = (1 - c_\sigma) + \sqrt{1 - (1 - c_\sigma)^2}\sqrt{\nu_w}(C^{(k)})^{-1/2}\frac{y_{mean}^{(k+1)} - y_{mean}^{(k)}}{\sigma^{(k)}} \tag{4.13}$$

The key operator here is $C^{-1/2}$ which acts as a whitening transformation for the updated path decorrelating $(y_{mean}^{(k+1)} - y_{mean}^{(k)})$. This transforms the update path vector from a sampled vector in the distribution $\mathcal{N}(0, C)$ to a vector in the distribution $\mathcal{N}(0, I)$ through the relation

$$\begin{aligned} y_i \sim & y_{mean} + \sigma\mathcal{N}(0, C) && (4.14)\\ \sim & y_{mean} + \sigma C^{1/2}\mathcal{N}(0, I) && (4.15)\\ C^{-1/2}y_i \sim & C^{-1/2}y_{mean} + \sigma\mathcal{N}(0, I) && (4.16)\\ & && (4.17) \end{aligned}$$

Therefore, the step-size evolution path exists in the decorrelated space

$$p_\sigma \sim \mathcal{N}(0, I) \tag{4.18}$$

Finally, the step size is updated according to

$$\sigma^{(k+1)} = \sigma^{(k)} \exp\Big[\frac{c_\sigma}{d_\sigma}\frac{||p_\sigma||}{E||\mathcal{N}(0, I)||} - 1\Big] \tag{4.19}$$

where $E||\mathcal{N}(0, I)||$ is the expectation of the distribution $\mathcal{N}(0, I)$. This equation acts to scale $p_\sigma \sim \mathcal{N}(0, I)$ with its expected value. When $||p_\sigma||$ is larger than the normal expectation, the step size σ is increased and when it is less, it is decreased. Therefore, when $p_\sigma^{(k+1)}$ and $p_\sigma^{(k)}$ are in similar directions, the paths cumulate and the

step size increases. When they are in opposite directions the paths annihilate and the step size decreases. Fairly straightforward to implement, the CMA-ES algorithm provides a vehicle to the search the design space, particularly for nonlinear objective functions.

4.2.2 Constraint handling

Constraint handling in genetic algorithms is still an area of active research. Several novel approaches have been implemented with respect to CMA-based algorithms [76, 77]; however, these approaches can often be objective function dependent. Consider the constrained optimization problem

$$\begin{aligned} \underset{y}{\text{minimize}} \quad & f(y) \\ \text{subject to} \quad & h_i(y) = 0,\ i = 1, \ldots, m. \\ & g_j(y) \geq 0,\ i = 1, \ldots, m. \end{aligned}$$

where y is the vector of design variables. In death penalty methods, a point that lies outside of the feasible domain is ignored entirely. Unfortunately, this method is particularly inefficient for CFD problems as objective functions take a long time to evaluate as no information of the violation is relayed back to search algorithm. In restoration methods [78], a point (y) that does not satisfy constraints is projected back to a point in the space $(\tilde{y})$ by estimating the constraint sensitivities to the design variables, through the linear approximation

$$g_j \approx g_j(y) + \nabla g_j(\tilde{y} - y) \tag{4.20}$$

This, however, might interfere with the CMA algorithm and force it to move tangent to the constraint boundary. To allow the algorithm to more freely navigate, a penalty method was implemented. In penalty methods, a loss function is used where a term is added to the objective function to include a penalty as a result of violating constraints.

$$L(y) = f(y) + P(y)$$

4.2.2.1 Penalty Functions

One of the most common penalty functions is the quadratic penalty function

$$P(y) = \frac{1}{2\mu_p} \sum h(y)^2 + \frac{1}{2\mu_p} \sum g_(y)^{-2} \tag{4.21}$$

Here, decreasing the penalty parameter μ_p increases the penalty severity, and in dynamic methods μ_p can be driven to 0 with each iteration, thereby driving the constraint to zero as well. However, this is known to cause the Hessian $\nabla^2_{yy} L$ to become ill conditioned. In addition, the inequality constraint can create a discontinuity in the event that $g_j(y) = 0$. Many penalty functions have been developed as it remains one of the most popular methods of constrained optimization; however, one common deficiency of penalty based methods is the reliance on heuristic hyperparameters to ensure convergence, particularly with regard to the penalty parameter [79].

4.2.2.2 Augmented Lagrange Approach

To overcome this strong dependence between the penalty parameter and constraint enforcement, several successful methods based on the augmented Lagrangian approach have been implemented by [80, 81]. Consider the equality constrained optimization problem

$$\begin{aligned} \underset{y}{\text{minimize}} \quad & f(y) \\ \text{subject to} \quad & h_i(y) = 0,\ i = 1, \ldots, m. \end{aligned}$$

In the penalty constraint method, μ_p needed to be driven to zero to ensure that the constraints are enforced at convergence. However, this had the adverse result of causing the objective function to become ill-conditioned. Powell [82] and Hestenes [83] independently developed the augmented Lagrange approach method to overcome these issues. In the augmented Lagrange approach, the objective function is modified to avoid the need to drive μ_p to zero thus removing the constraints dependence on μ_p. The Lagrangian for the equality constrained problem is defined as

$$L(y, \lambda) = f(y) + \lambda_i h(y) \tag{4.22}$$

The Hestenes-Powell augmented Lagrangian for a quadratic penalty function is

$$L(y, \lambda) = f(y) + \lambda_i h(y) + \frac{1}{2\mu_p} \sum h(y)^2 \tag{4.23}$$

Here, the difference between the two is the addition of the penalty term. In the augmented Lagrange approach, the multiplier is chosen to help advance the solution to the optimal by estimating the optimal Lagrange multiplier at each iteration. Powell [82] showed that a good estimate for this value is

$$\lambda^{(k+1)} = \frac{h(y)^{(k)}}{\mu_p^{(k)}} \tag{4.24}$$

For the inequality constrained optimization problem, the problem can be reformulated as an equality constraint with the addition of the slack variable s_i.

$$s_j^{(k)} = \max\{g_j(y^{(k)}) - \mu_p^{(k)} \lambda_j^{(k)}, 0\} \tag{4.25}$$

The problem can be then reformulated as a combination of equality constraints from

$$g_j(y) \geq 0$$

to

$$g_j(y) - s_j = 0$$

The updated Lagrange multiplier estimate λ_i for the inequality constrained problem becomes

$$\lambda_j^{(k+1)} = \max\{\lambda_j^{(k)} - \frac{g_j(y^{(k)})}{(k)}, 0\} \tag{4.26}$$

4.3 Computational Approach

For the airfoil optimization problem, two inequality constraints have been applied. The airfoil area was constrained to prevent the airfoil from growing increasingly slimmer. A constraint on the pitching moment was applied as to avoid compromising the pitching moment in an effort to maintain the lift coefficient. The augmented

loss function becomes

$$L(y^{(k)}) = c_d(y^{(k)}) + \lambda_m(M(y^{(k)}) - s_m^{(k)}) + \frac{1}{2\mu_p}(M(y^{(k)}) - s_m^{(k)})^2 + \lambda_a(A(y^{(k)}) - s_a^{(k)}) + \frac{1}{2\mu_p}(A(y^{(k)}) - s_a^{(k)})^2 \tag{4.27}$$

where $M(y^{(k)})$ and $A(y^{(k)})$ are the pitching moment and area constraints

$$M(y^{(k)}) = (c_d^{(k=1)}/c_{m_{targ}})(c_{m_{targ}} - c_m(y^{(k)})) \tag{4.28}$$

$$A(y) = (c_d^{(k=1)}/\text{area}_{targ})(\text{area}(y^{(k)}) - \text{area}_{targ}) \tag{4.29}$$

The terms $c_d^{(k=1)}/\text{area}_{targ}$ and $c_d^{(k=1)}/c_{m_{targ}}$ are added to scale the constraints to the same magnitude as $c_d(y^{(k)})$ as well as keep the pitching moment constraint positive. The Lagrange multiplier estimates for the pitching moment and area constraints are initialized to 0 and their updates respectively are defined by

$$\lambda_m^{(k+1)} = \max\{\lambda_m^{(k)} - \frac{M(y^{(k)})}{\mu_p^{(k)}}, 0\} \tag{4.30}$$

$$\lambda_a^{(k+1)} = \max\{\lambda_a^{(k)} - \frac{A(x)}{\mu_p^{(k)}}, 0\} \tag{4.31}$$

The relaxation variables for the area and pitching moment constraints s_a and s_m are defined as

$$s_m^{(k)} = \max\{M(y^{(k)}) - \mu_p\lambda_m^{(k)}, 0\} \tag{4.32}$$

$$s_a^{(k)} = \max\{A(y^{(k)}) - \mu_p\lambda_a^{(k)}, 0\} \tag{4.33}$$

An initial value of the penalty parameter μ_p was chosen to keep the penalties at the same order of magnitude as the objective function. Because M and A are small

$$\mu_p^{(k=1)} = \frac{1}{2}\sqrt{A(y^{(k=1)})^2 + M(y^{(k=1)})^2} \tag{4.34}$$

At each iteration, the penalty function is estimated at the new y_{mean} by estimating the sensitivities and projecting the constraint violations at the new point

through an estimate of the constraint sensitivities to the design variables.

$$dN_i = y_{mean}^{(k)} \vec{1}_\epsilon^T - [y_1^{(k)}, y_2^{(k)} \cdots y_\epsilon^{(k)}]$$
$$dG_j = [g_j(y_1^{(k)}), g_j(y_2^{(k)}) ... g_j(y_\epsilon^{(k)})] - g_j(y_{mean}^{(k)}) \vec{1}_\epsilon$$
$$\frac{dG_j}{dN} = (N^T N)^{-1} NG$$
$$g_j(y_{mean}^{(k+1)}) \approx g(y_{mean}^{(k)}) + [y_{mean}^{(k+1)} - y_{mean}^{(k)}] \frac{dG_j}{dN}$$

Here $\vec{1}_\epsilon$ is a vector of 1's of length ϵ. If the penalty is below a tolerance τ_k, then the multipliers λ_a and λ_m are re-evaluated to advance the solution. If the penalty is above the tolerance, then the penalty parameter μ_p is decreased to coerce the solution back into the feasible region. Either way, τ is increased at each iteration.

The algorithm used for combining the CMA, CST, and augmented Lagrange approach is inspired by the Evolian algorithm of Myung and Kim [84] and is as follows:

Algorithm 1 CMA, Augmented Lagrange

```
Initalize parameters
while Not converged do
    Generate sample population of design vectors y_1, y_2...y_ε
    for Each candidate y_i do
        evaluate L(y_i)
    end for
    Sort population by L(y)
    Weight and recombine y_mean^(k+1)
    Estimate constraint sensitivities
    Project for new constraint violations at y_mean^(k+1)
    if P(y_mean^(k+1)) < τ then
        Update λ_j^(k+1)
    else
        Decrease μ_p^(k+1)
    end if
    Update evolution paths p_σ, p_c
    Adapt covariance matrix C^(k+1)
    Update step size σ^(k+1)
end while
```

Chapter 5
Results and Discussion

5.1 Baseline Rotor

The baseline rotor considered in this work was that of a UH-60A operating at a gross weight of 16,000 lbs at a density altitude of 5250 ft. These conditions correspond to a weight coefficient of $C_w = 0.0116$ on the baseline rotor. To verify the model, calculations made using ROTOR were compared with steady level flight experimental data contained in Ref. [8]. The control positions required to trim the rotor and the power required across a range of forward airspeeds are represented in Figures 5.1 and 5.2, respectively, where the solid lines represent the calculations made by ROTOR, and the unconnected points represent the experimental data in Ref. [8].

At low advance ratios, there is a significant discrepancy between the model and experimental results. This is in part due to difficulty in trimming the aircraft at these airspeeds during the experimental data collection (Ref. [8]). Also, no influences from rotor downwash on the airframe were included in the model. It should also be noted that all solutions were run quasi-steady. No dynamic stall model was used in this work, which can cause inaccuracies at high advance ratios. Nevertheless, ROTOR supplies a reasonably accurate platform to comparatively analyze the performance of the lift coefficient correction and optimization routines.

To perform the calculations, airfoil performance tables were first generated for the SC1095 and SC1094R8 airfoils using OVERFLOW 2.2i, following the methodology laid out in Chapter 4. High angle-of-attack points were supplemented by experimentally generated NACA0012 data as illustrated in Figures 5.3 - 5.5 for

the SC1095 airfoil, where the gray regions highlight the supplemented NACA0012 data. Low angle-of-attack regime details for the SC1095 and SC1094R8 are shown in Figures 5.6 - 5.8. All tables were constructed using a Reynolds number of 6×10^6.

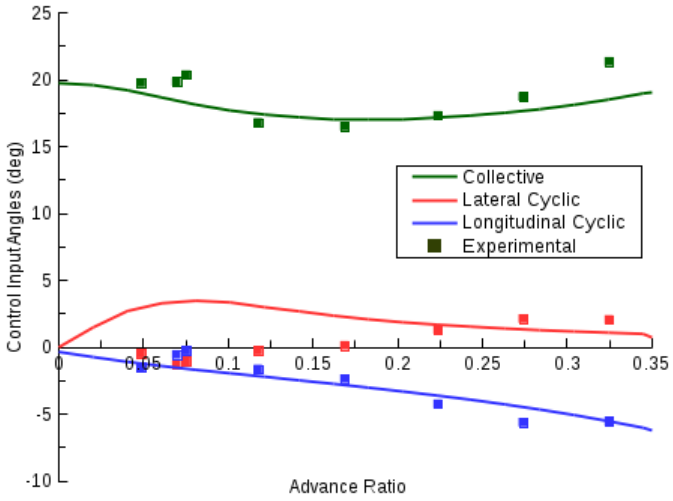


Figure 5.1: UH60 control inputs with varying airspeed, (Ref. [8]), $C_w = 0.0116$

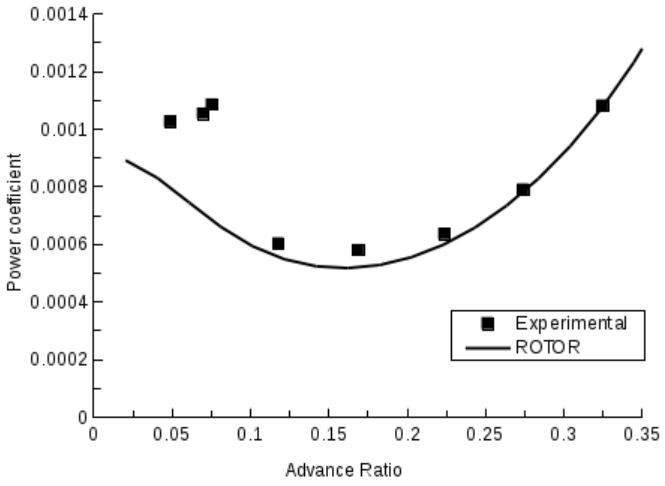


Figure 5.2: UH60 main rotor power vs airspeed, (Ref. [8]), $C_w = 0.0116$

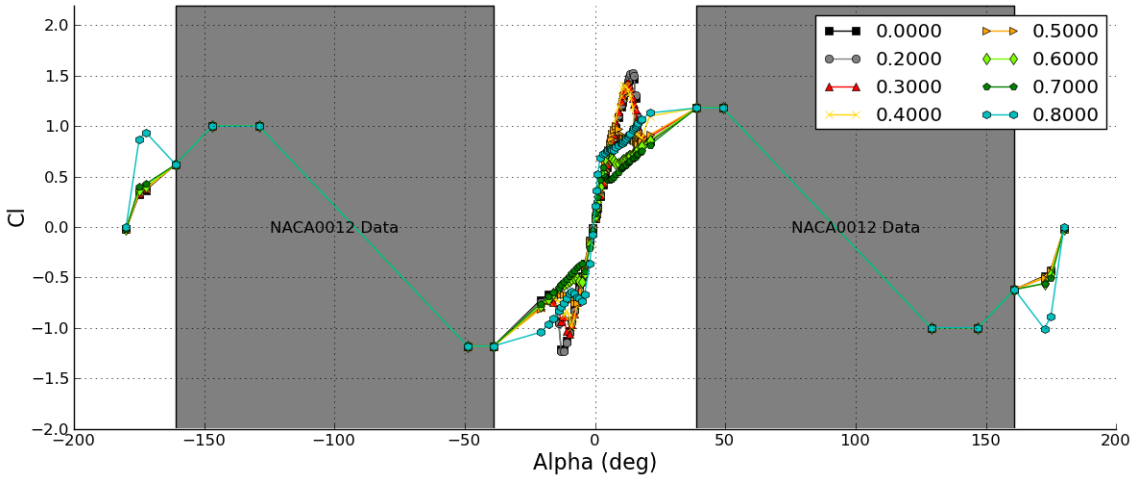


Figure 5.3: Performance table lift coefficient SC1095, $Re = 6 \times 10^6$

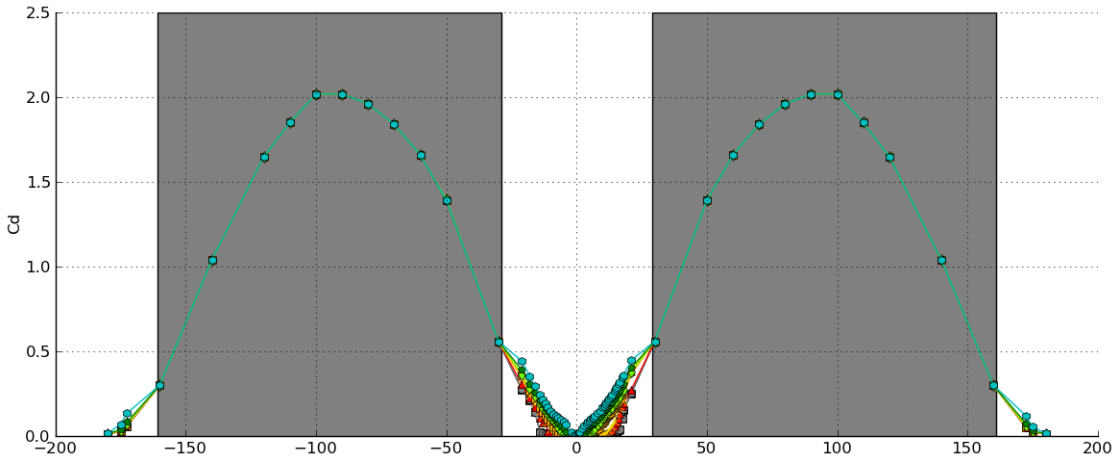


Figure 5.4: drag coefficient SC1095 C81 table, $Re = 6 \times 10^6$

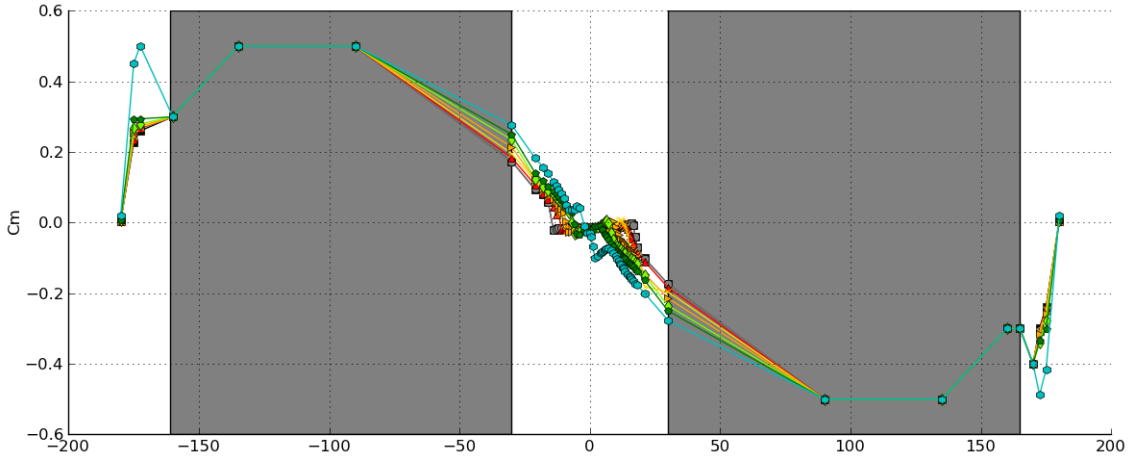
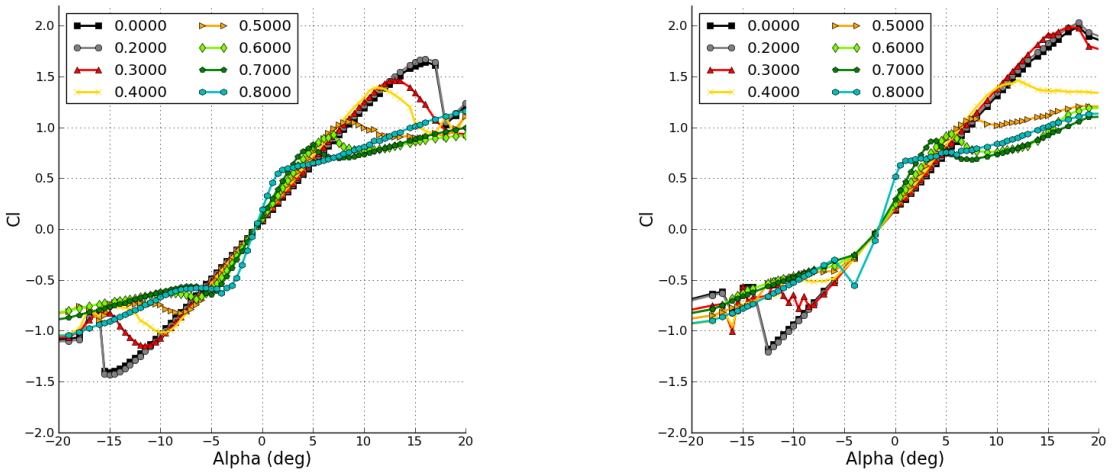


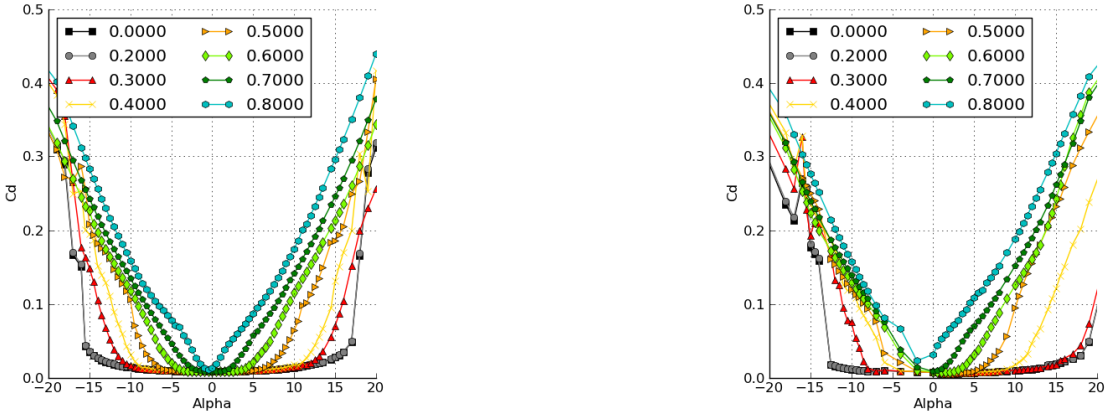
Figure 5.5: Performance table pitching moment coefficient SC1095, $Re = 6 \times 10^6$



(a) SC1095

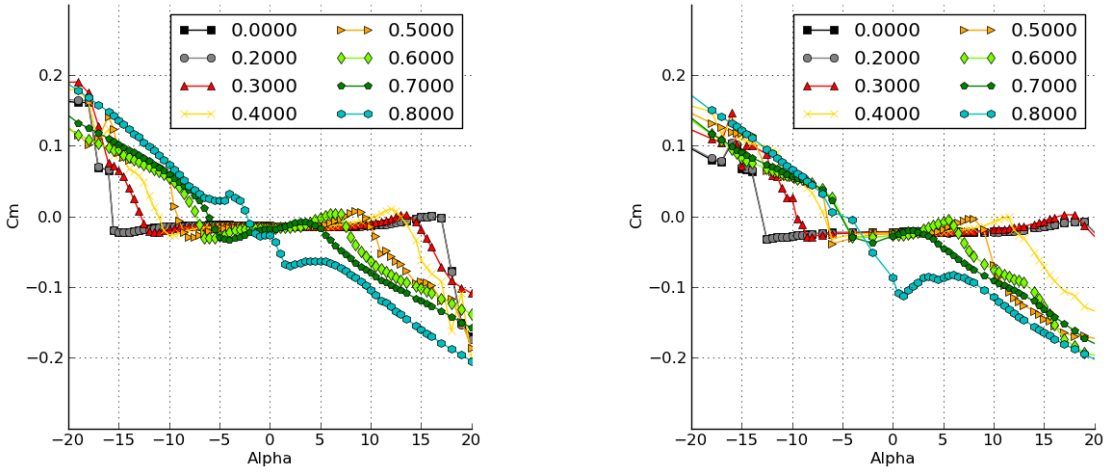
(b) SC1094R8

Figure 5.6: Performance table lift curve slopes, $Re = 6 \times 10^6$



(a) SC1095 (b) SC1094R8

Figure 5.7: Performance table drag coefficient, $Re = 6 \times 10^6$



(a) SC1095 (b) SC1094R8

Figure 5.8: Performance table pitching moment coefficient, $Re = 6 \times 10^6$

5.1.1 Lift Coefficient Correction

To more accurately predict the maximum sectional lift coefficient, a lift coefficient correction was applied to the performance tables using the methodology developed in Chapter 3. The positive and negative stall regions were corrected independently by evaluating the momentum thickness Reynolds number (Re_{δ_2}) on the upper and lower surfaces, respectively. Figures 5.9 and 5.10 show the trailing-edge Re_{δ_2}

distribution across a range of angles-of-attack for the SC1095 upper and lower surfaces. The momentum thickness Reynolds number generally increases past the empirical criterion to a peak which roughly coincides with the CFD predicted stall. As the separated region rapidly grows in the CFD solution, the airfoil stalls and Re_{δ_2} decreases as the airfoil moves into the post-stall regime. The momentum thickness Reynolds number does not monotonically increase; which means that it cannot be used on an isolated flow solution to determine if an airfoil should be in the post-stall regime. Instead, the correction methodology must be used contextually and is restricted as a post-processing step.

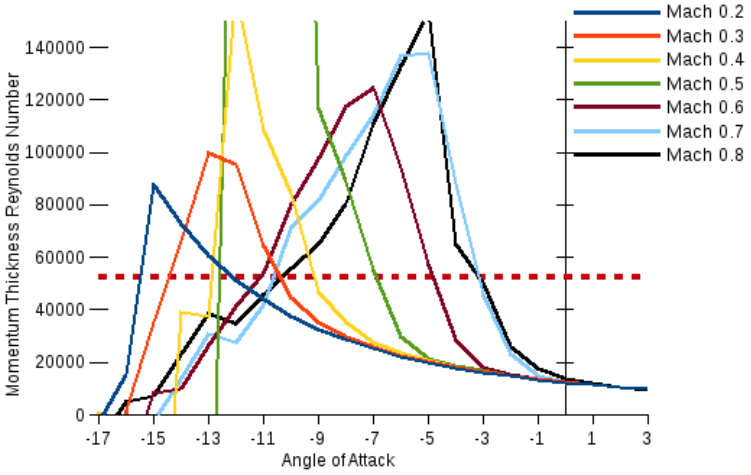


Figure 5.9: SC1095 lower surface trailing-edge Re_{δ_2}, $Re = 6 \times 10^6$

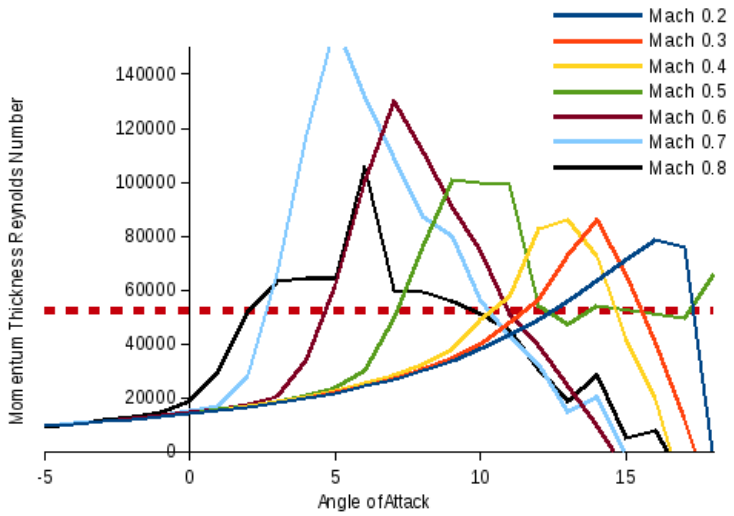
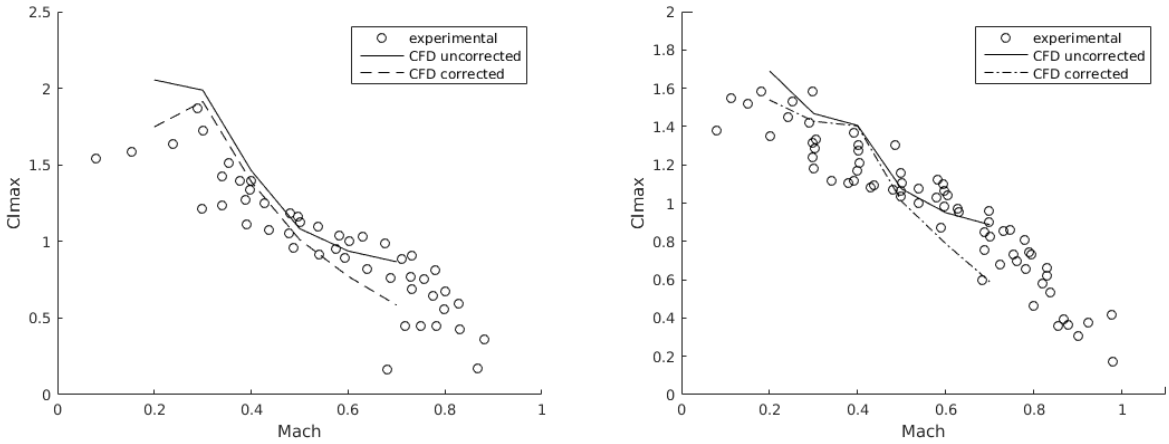


Figure 5.10: SC1095 upper surface trailing-edge Re_{δ_2}, Re=6×10^6

The corrected lift coefficients are compared against experimental results from Ref. [1] in Figure 5.11. The correction appears to be most effective at lower Mach numbers (from 0.2 to 0.5). Perhaps most notable is that the the corrected low Mach number solutions (Mach < 0.3) show an initial increase in maximum lift coefficient with increasing Mach number, following the same trend for the SC1094R8 as the experimental data (Figure 5.11a), whereas the raw CFD values incorrectly over predict the low-Mach number $c_{l,max}$.

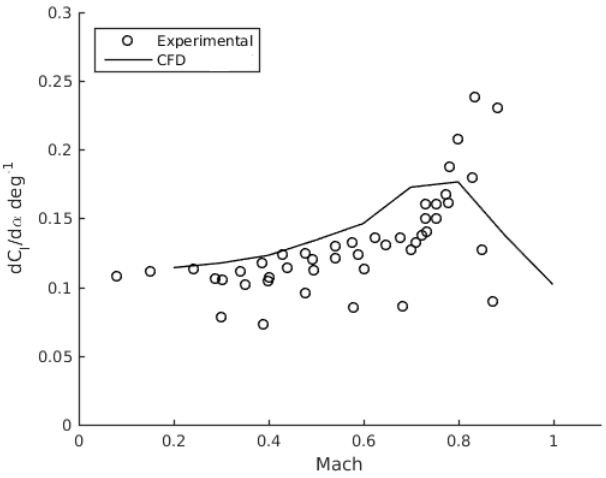
The CFD predictions are better able to predict $c_{l,max}$ in the transonic regime, as stall becomes increasingly driven by compressibility effects and shockwave formation, which mitigates the errors of RANS turbulence modeling.



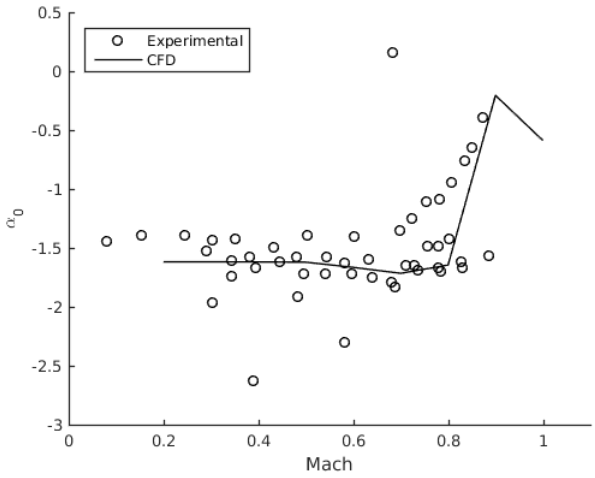
(a) SC1094R8 (b) SC1095

Figure 5.11: $c_{l_{max}}$ correction comparison $Re = 6 \times 10^6$ (Ref. [1])

The raw lift slope and zero-lift angle for both the SC1095 and SC1094R8 agree well with the experimental data collected by Ref. [1] (Figures 5.12 and 5.13), which confirms the assumption that these values should remain unchanged through the correction mapping. It should be noted, though, that high Mach number experimental data are error-prone for two-dimensional cases as shock reflections develop in the test section. Although the drag coefficients were not modified in this work, it can be seen in Figures 5.14 and 5.15 that the zero-lift drag coefficients (c_{d_0}) also agree reasonably well with experimental results.

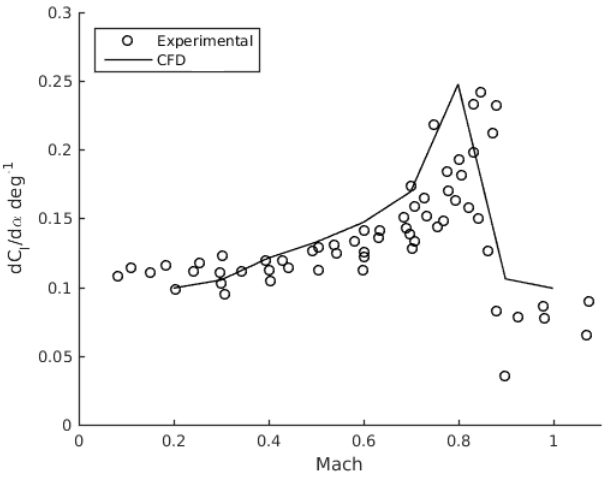


(a) SC1094R8 $\frac{dC_l}{d\alpha}$

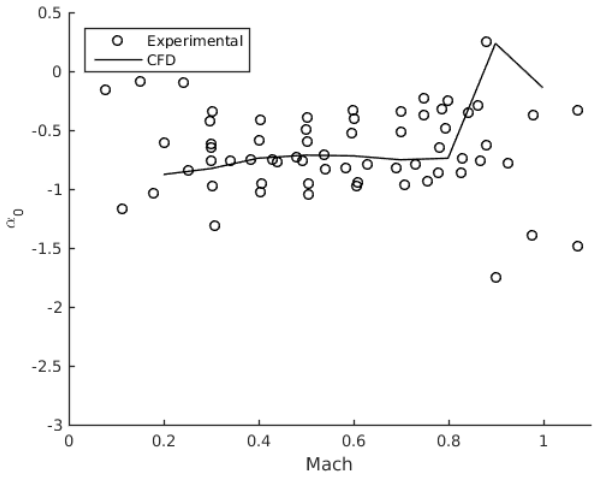


(b) SC1094R8 α_0

Figure 5.12: SC1094R8 polar parameter comparison, $Re = 6 \times 10^6$ (Ref. [1])



(a) SC1095 $\frac{dC_l}{d\alpha}$



(b) SC1095 α_0

Figure 5.13: SC1095 polar parameter comparison, $Re = 6 \times 10^6$ (Ref. [1])

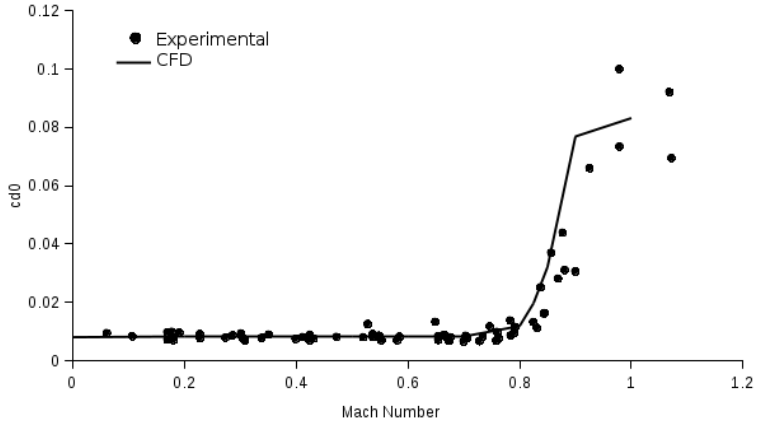


Figure 5.14: SC1095 c_{d_0} comparison, $Re = 6 \times 10^6$ (Ref. [1])

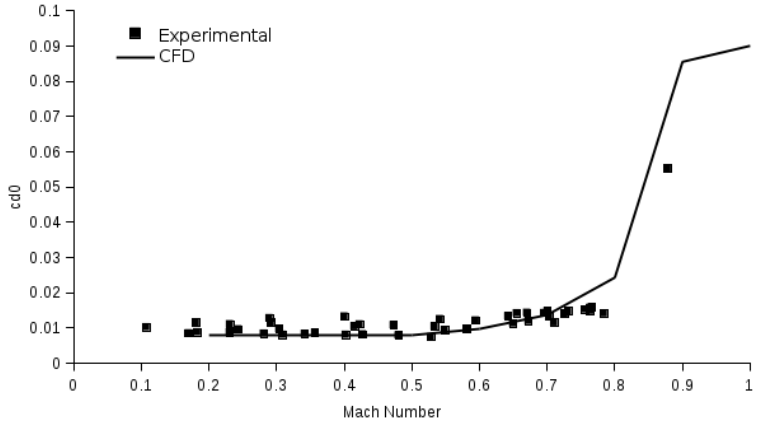


Figure 5.15: SC1094R8 c_{d_0} comparison, $Re = 6 \times 10^6$ (Ref. [1])

5.1.1.1 Corrected Airfoil Performance

The influence of the lift coefficient correction on rotor performance was investigated. At the baseline gross weight, the correction has little impact on the overall performance of the rotor as it operates mostly below the stall margin of the airfoils. ROTOR was used to compute the Mach number and angle-of-attack combinations experienced by the rotor for advance ratios of $0 \leq \mu \leq 0.4$. The ROTOR computed values are compared against the stall boundary for the SC1095 and SC1094R8 airfoils as computed with OVERFLOW in Figures 5.16 and 5.17. These values were calculated at 150 evenly spaced azimuth stations.

For the baseline case ($C_w = 0.0116$), neither of the airfoils experience angles-

of-attack beyond $\alpha_{c_{l,max}}$. Tip sections with the SC1095 airfoil, however, do exceed the negative stall boundary. This will be further explored during the optimization routine, as this result implies increasing the negative stall margin at the rotor tip and can help to increase performance at higher advance ratios.

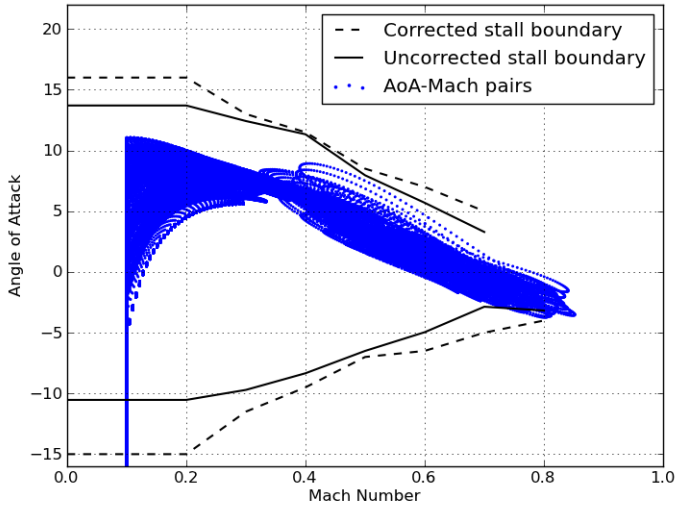


Figure 5.16: Calculated Mach numbers and angles-of-attack for SC1095, $Re = 6 \times 10^6$, $0 \leq \mu \leq 0.4$, $C_w = 0.0116$

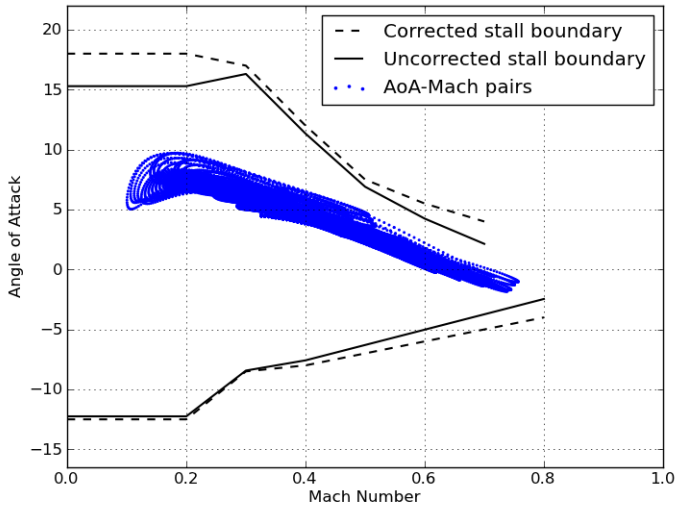


Figure 5.17: Calculated Mach numbers and angles-of-attack for SC1094R8, $Re = 6 \times 10^6$, $0 \leq \mu \leq 0.4$, $C_w = 0.0116$

The influence of the lift coefficient correction becomes most apparent when analyzing the performance limitations of the rotor, particularly with regards to the stall margin. Weight coefficient versus the power coefficient data for the UH-60A baseline rotor in hover is plotted in Figure 5.18. The lift-corrected performance table exhibits a maximum weight coefficient of 6.8% lower than that of the uncorrected table. This deficiency emphasizes the importance of the lift coefficient correction, particularly during rotor design. For the baseline case, this corresponds to an 1801-lb maximum take-off weight deficit between the corrected and uncorrected performance tables. Designing a rotor with an inaccurate lift coefficient could lead to an over-prediction of the rotorcraft's performance, resulting in a final design with a lower stall margin than intended.

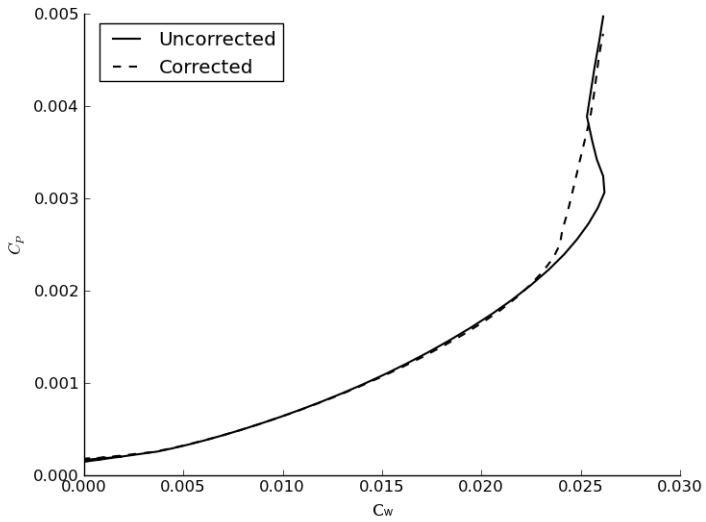


Figure 5.18: Corrected performance table stall margin comparison in hover

5.2 Optimization

The airfoil shape was optimized using the genetic algorithm methodology described in Chapter 4. A total of 6 design variables and a population size of 10 was used to conduct the optimization. An advance ratio of $\mu = 0.30$ was chosen to act as the design point for the UH-60A baseline rotor. This advance ratio corresponds to a forward airspeed of approximately 128.6 kts. For this optimization, only changes to the the tip airfoil (SC1095) were considered between $0.85 \leq \frac{r}{R} \leq 1.0$. In order to

locate an optimization point that would likely provide the largest decrease in power required, a line search algorithm was employed to integrate the torque produced by various Mach number and angle-of-attack combinations over the rotor disk. Although the actual optimization was essentially a point optimization, the new airfoil was assumed to have a lower drag coefficient for points within 0.25 degrees of the optimization angle, and Mach numbers within 0.025 of the design Mach number. The target point for airfoil optimization was chosen to be a Mach number of 0.80 and a lift coefficient of -0.551, as this epicenter was determined to produce the majority of the power required by the rotor at the design speed. At the design speed, this region produces approximately 2.9% of the total rotor torque, however, the actual affected region may be larger or smaller. The highlighted area in Figure 5.19 shows the region assumed to be benefited by the optimization at the design airspeed.

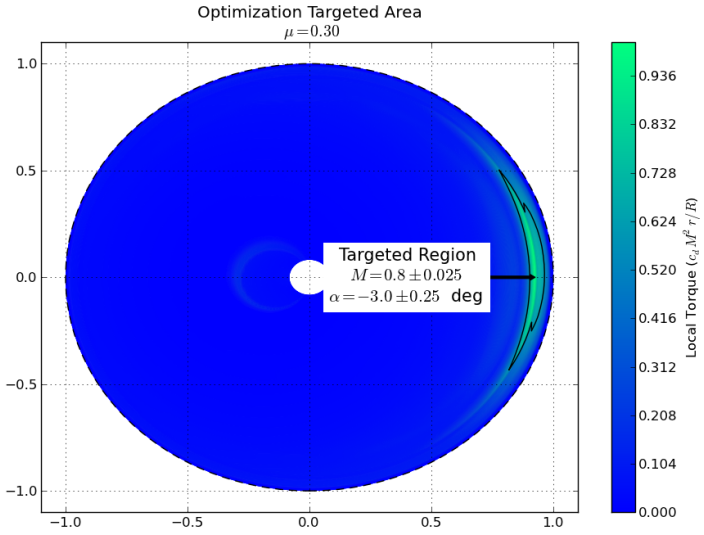


Figure 5.19: Rotor torque distribution
baseline rotor, $C_w = 0.0116$, $\mu = 0.3$

5.2.1 Single Point Design Optimization

Two optimizations were performed in order to cross-validate the optimum solution. The first optimization used the SC1095 airfoil as a starting point. This optimization acts as an interior method as the starting point was inside of the feasible region, ie.

the design constraints were satisfied at the starting point. In order to verify the results, a second optimization starting from a point exterior to the feasible region was carried out. For the exterior design point, the starting airfoil was inverted so that the pitching moment coefficient constraint was not immediately satisfied at the starting position. This allows for a validation of the resultant airfoil as well as a validation of the constraint handling technique.

5.2.1.1 Convergence

The exterior point drag convergence experiences a very dramatic drag increase as it attempts to balance the design constraint enforcement with the objective function minimization. The interior starting point exhibits a much more monotonic decrease towards its final optimum, except similar to the exterior starting point, the drag coefficient exhibits several oscillations as the design approaches the final drag coefficient value.

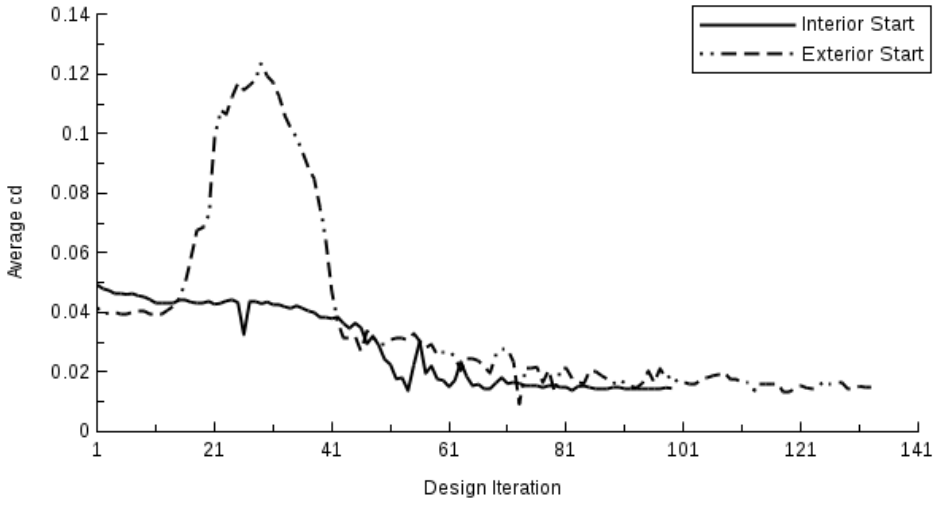


Figure 5.20: Single design point optimization drag convergence, $c_l = -0.551$, $M_\infty = 0.8$, $Re = 6 \times 10^6$

This is partially due to the path-length control implemented in the CMA-ES algorithm. When successive paths are in similar directions, the optimization path length grows longer; when paths are in opposite directions, the search path length decreases. This means that when the algorithm initially encounters an optimal solution, it continues past the solution for a few steps. It then recognizes that the

solution is getting worse and changes directions until it continues past the solution again. As the path length decreases the oscillations dampen and the algorithm begins to converge to a final solution. In a physical sense, this is analogous to the optimizer having a momentum as it navigates the cost surface. This phenomenon can be seen by analyzing the standard deviation (ie. path length) of the generated population over the optimization process in Figure 5.21 where it is shown that strong oscillations tend to occur near changes in path length size as the algorithm changes direction on the cost surface.

It was also observed in this work that the early performance of the optimization is subject to the initial choice of σ. If this value is too large the routine has a tendency to exhibit severe oscillations early on as the distribution ellipse only contains information from a limited number of previous iterations.

It should also be noted that Figure 5.20 only represents the objective function c_d, and does not include the influence from the added penalty. The loss function, with the inclusion of the penalty function, may exhibit fewer oscillations and appear to decrease more monotonically than just the objective function. This needs to be further explored, however, before any conclusion can be drawn.

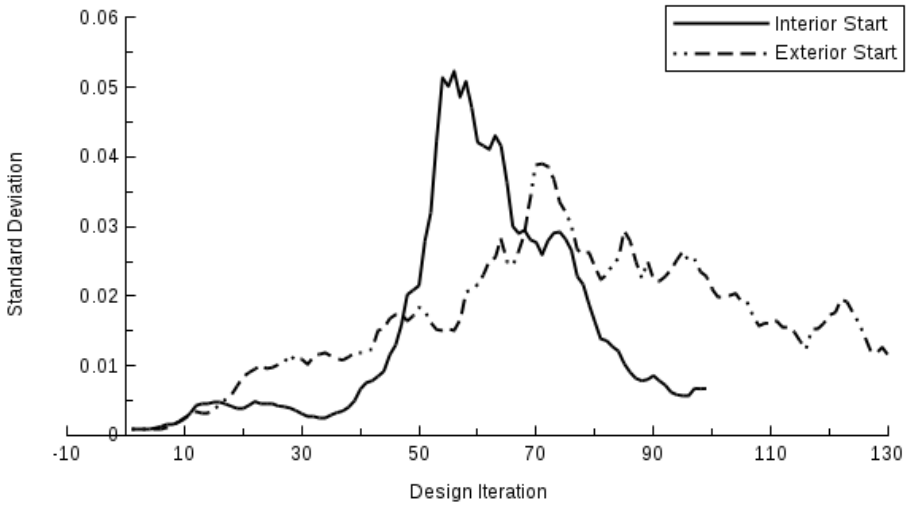


Figure 5.21: Single design point optimization design variable standard deviation, $c_l = -0.551$, $M_\infty = 0.8$, $Re = 6 \times 10^6$

5.2.1.2 Optimized Airfoil Sectional Properties

The interior and exterior starting points appear to be approaching a similar final shape and drag coefficient; however, the design spaces are not completely interchangeable. Both airfoils have similar leading-edge radii, except that the interior start pushed some of the thickness to the upper surface, while the exterior start maintained the thickness on the lower surface. Neither of the starting points completely converged to $\sigma = 0$, even though the drag coefficients appear to have significantly leveled off. The interior point does exhibit significantly fewer oscillations so the performance evaluation will be completed using the airfoil resulting from the interior optimization. The coordinates of the optimized airfoil are given in the Appendix.

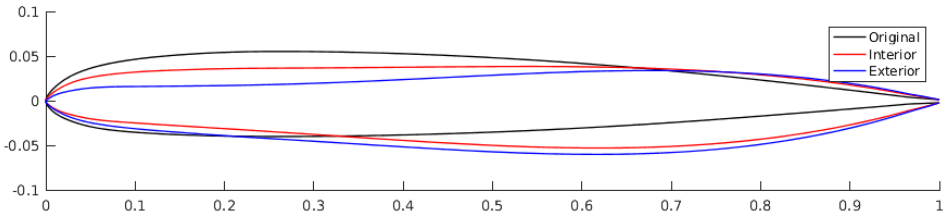


Figure 5.22: Single design point optimization resultant shape comparison,$c_l = -0.551$, $M_\infty = 0.8$, $Re = 6 \times 10^6$

Table 5.1: Single point optimization results, $M_\infty = 0.8$,$Re = 6 \times 10^6$

	Original Interior	Original Exterior	Interior	Exterior
Airfoil Area	0.0651	0.0651	0.06815	0.06542
c_m	0.02173	0.0681	0.02082	0.0217
c_l	-0.551	-0.551	-0.551	-0.551
c_d	0.0492	0.0415	0.0145	0.0136

The optimization routine attempted to move the low-drag region towards the design point by changing the camber of the airfoil. The negative lift coefficient design point results in an airfoil with a negative camber. The drag polar for the optimized airfoil at the design Mach number shows that the minimum drag location has been shifted to a lower lift coefficient. In addition, by flattening the lower

surface and decreasing the leading-edge radius, the optimization routine was able to delay the shock-wave formation and weaken its strength at the design lift coefficient. Both of these features can be seen in the surface pressure coefficient shown in Figure 5.23 and the Mach contours shown in Figure 5.24. The airfoil experiences a lower flow acceleration around the leading edge due to its decreased radius. It also sees the shock wave pushed back from around $0.55c$ to $0.64c$, while the pressure increase across the shock is also significantly reduced.

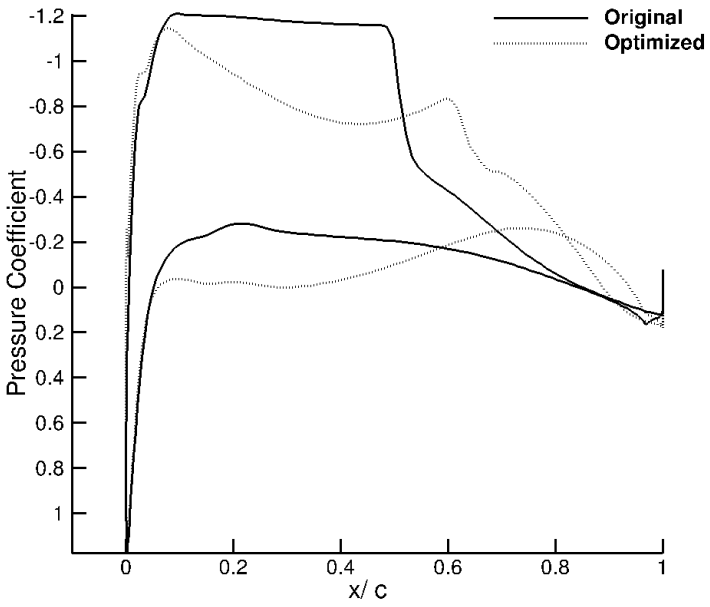
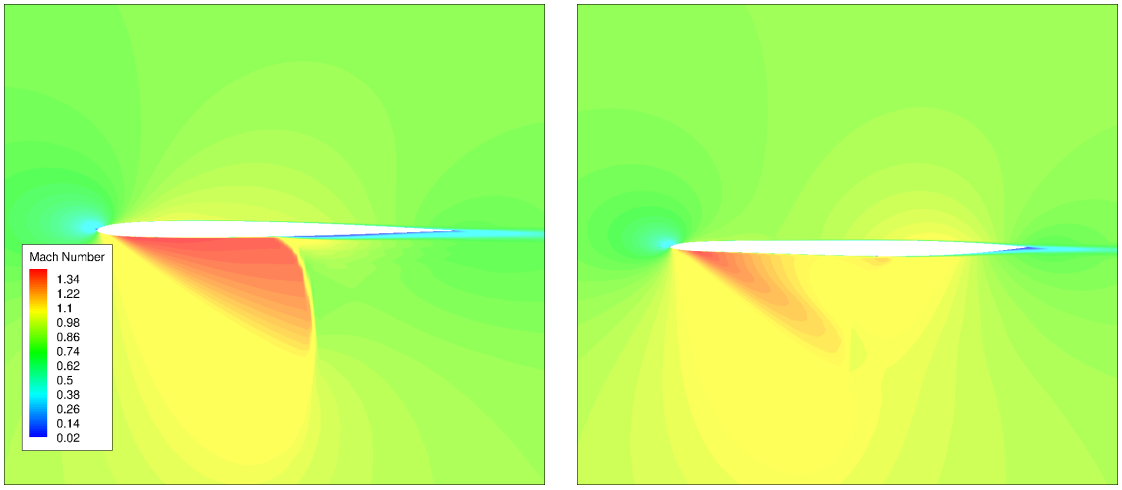


Figure 5.23: Single design point optimization surface pressure coefficient comparison (Interior Start), $Re = 6 \times 10^6$, $c_l = -0.551$, $M_\infty = 0.8$



(a) original (b) optimized

Figure 5.24: Single design point optimization Mach number contour comparison (Interior Start), $Re = 6 \times 10^6$, $c_l = -0.551$

The optimized airfoil presents an almost global improvement in drag at the design Mach number (Figure 5.25); however, the drag rise is much sharper than the original airfoil, a quality that may cause large vibrations as the rotor passes through the region. Also, the optimized airfoil shows slightly higher drag divergence and shows a lower increase in drag at higher Mach numbers. Figure 5.26 shows the change in the drag coefficient at the zero-lift angle-of-attack for the original and optimized airfoils.

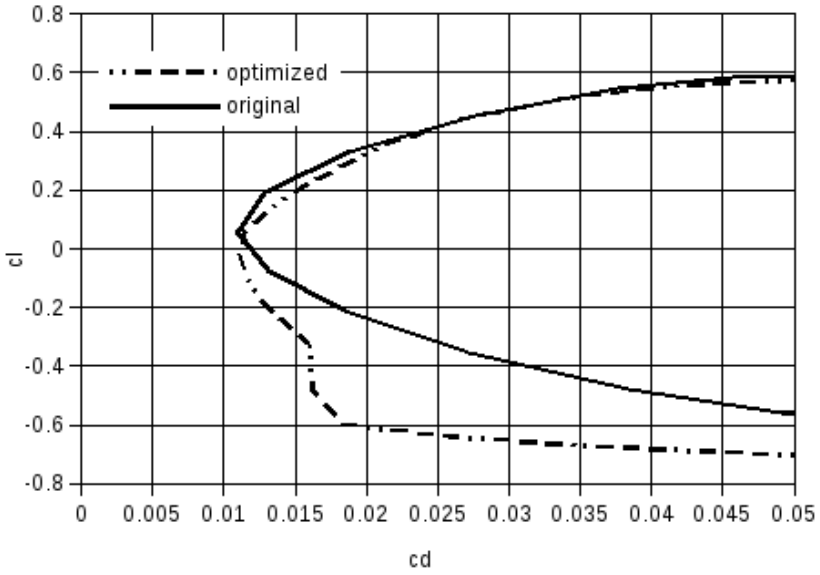


Figure 5.25: Single design point optimization drag polar comparison (Interior Start), $Re = 6 \times 10^6$, $M_\infty = 0.8$

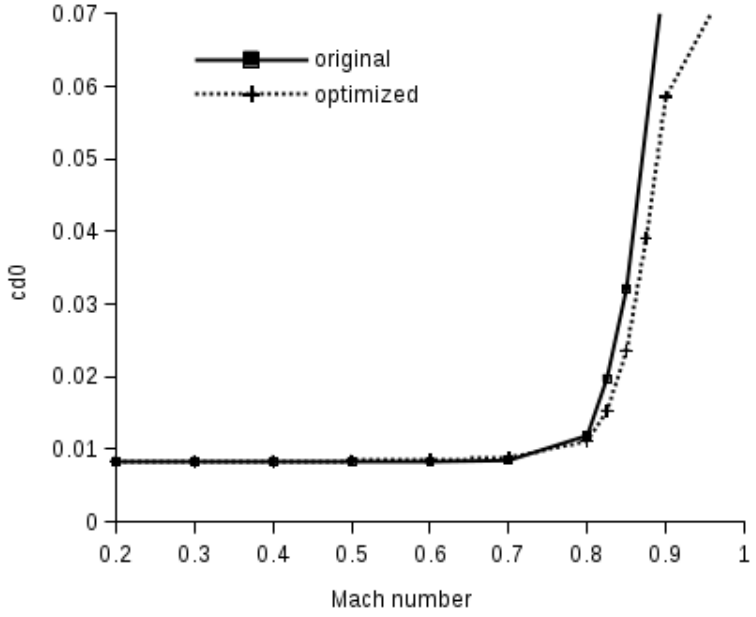


Figure 5.26: Single design point optimization c_{d_0} comparison (Interior Start), $Re = 6 \times 10^6$

5.2.1.3 Optimized Airfoil Performance

The difference in the change in local rotor torque between the optimized and original airfoils at the design speed is shown in Figure 5.27. Here, the optimized airfoil was used to replace the outboard 8% of the rotor radius and a negative value corresponds to a decrease in local torque. The resulting power required at various airspeeds and gross weights are compared in Figure 5.28. The optimized tip airfoil did not substantially change the trim state at the design speed, allowing the airfoil to significantly reduce the torque on the advancing blade with a marginal performance reduction of the inboard sections.

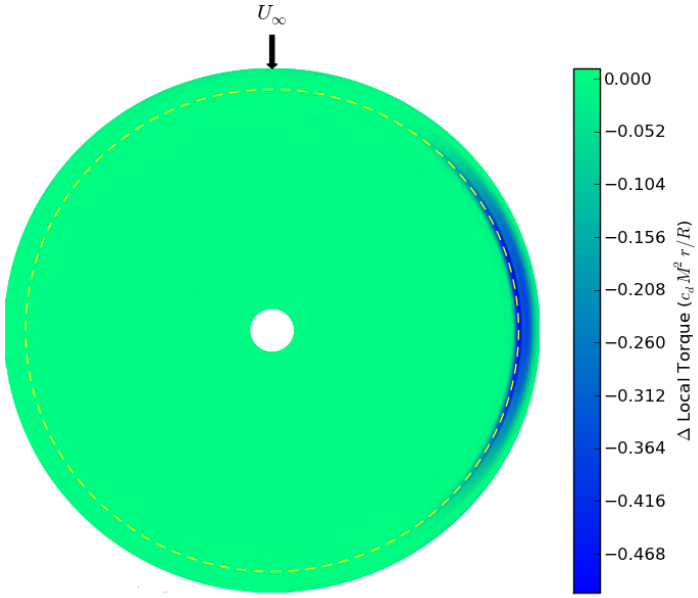


Figure 5.27: Difference in local rotor torque contours ($T_{\text{optimized}}$ - T_{original})
$\mu = 0.30$, 8% optimized

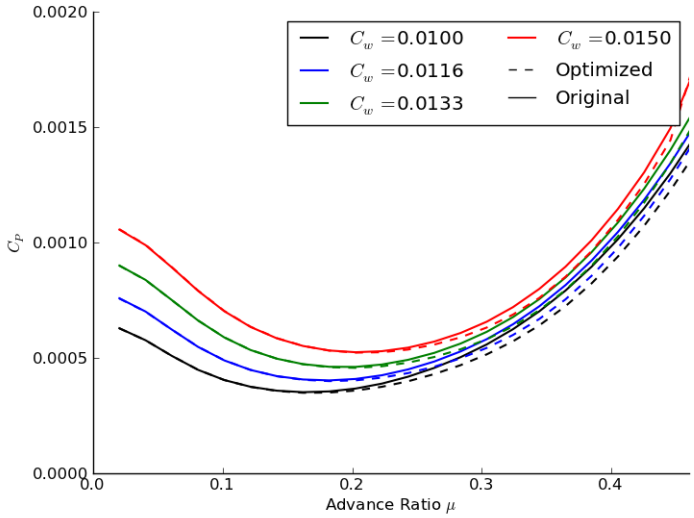


Figure 5.28: Single design point optimization blade profile power required, 8% optimized

The Mach number and angle-of-attack combinations that are experienced by the rotor tip airfoil are shown in Figure 5.29. At the design Mach number (0.8), the stall margin has been expanded, and the rotor no longer experiences angles-of-attack beyond negative stall. This allows the optimized rotor to delay the sharp drag increase that comes with the stall and post-stall regimes, reducing the power required in forward flight. The new tip airfoil reduces the required power by $5-9\%$ depending on the gross weight. The ratio of the optimized to baseline power required is shown in Figure 5.30 for various advance ratios. These values are based on the rotor power required to overcome the rotor torque and do not include the power required to overcome air-frame drag, as this is counted as parasitic power and is not affected by rotor changes. Then the total main rotor power is reduced by approximately 3.6

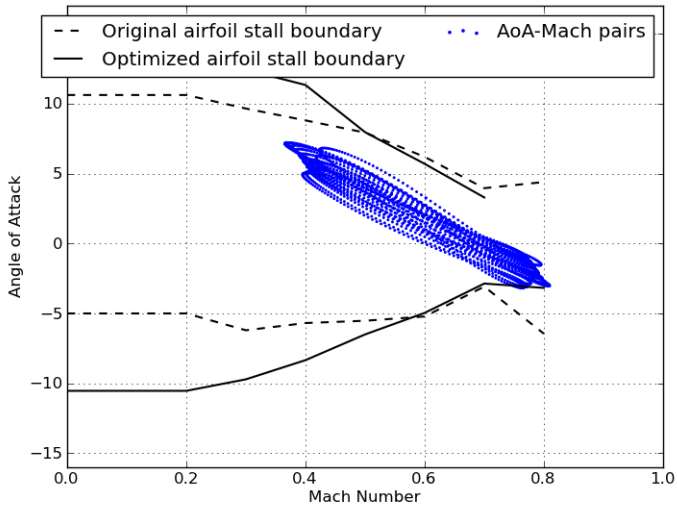


Figure 5.29: Calculated Mach numbers and angles-of-attack for tip airfoil section, $Re = 6 \times 10^6$, $\mu = 0.3$, $C_w = 0.0116$

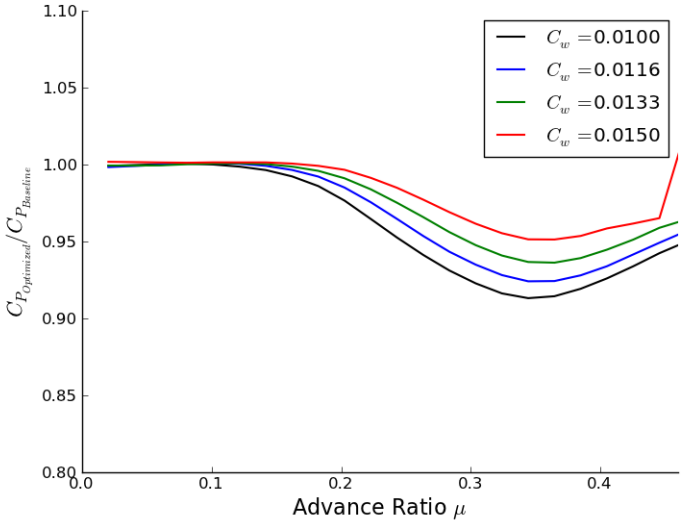


Figure 5.30: Single-point-optimization blade profile power ratio ($\frac{C_{P_{\text{optimized}}}}{C_{P_{\text{baseline}}}}$) with varying gross weight

The power required replacing various lengths of the rotor blade with the optimized airfoil section is compared in Figure 5.31. Replacing the outboard 5% of the rotor blade provides very little increase in performance, and this is partially

due to the twist distribution of the rotor blade. The UH-60A rotor blade sees a sharp increase in twist (more negative) starting at $0.85R$. The twist peaks at $0.92R$ and decreases along the rest of the blade (See Figure 3.2). This causes the most outboard portions of the rotor blade to experience more positive angles-of-attack than at more inboard tip sections. The blade also has a 20 degree tip sweep that begins at approximately $0.92R$ which reduces the effective Mach number of the most outboard blade sections. Both of these factors limit the exposure of the most outboard tip sections to the design point. Instead, most of the benefit is concentrated near $0.90 \leq \frac{r}{R} \leq 0.95$. Using the optimized airfoil on the outboard 15% provides worse performance compared to the 8% results as the more inboard blade sections operate at angles beyond the stall margin of the optimized airfoil section for the Mach numbers experienced. However, at large advance ratios ($\mu \geq 0.40$), the 15% R results begin to perform better as further inboard sections begin to experience higher Mach numbers, allowing them to benefit from the new airfoil.

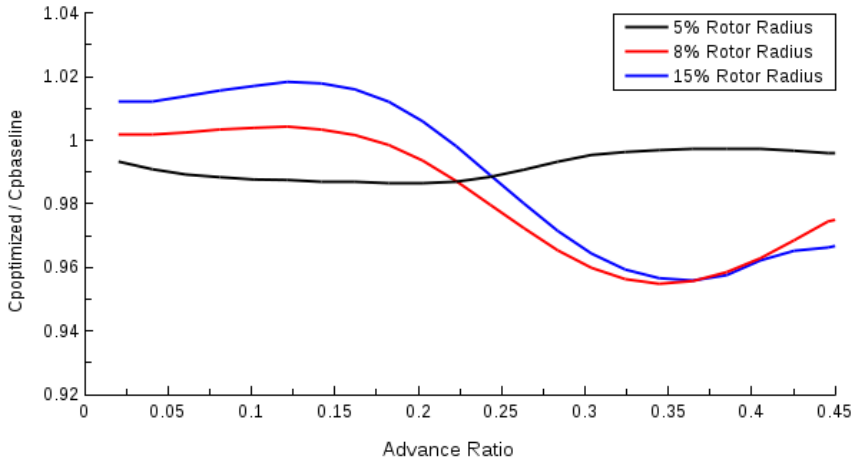


Figure 5.31: Ratio of total power required for steady level forward flight ($\frac{C_{P_{\text{optimized}}}}{C_{P_{\text{baseline}}}}$) with optimized airfoil sections of varying length along blade span, $C_w = 0.0116$

As the gross weight increases, higher collective-pitch values are required to trim the rotor, causing higher angles-of-attack across the blade, shifting the angle-of-attack and Mach number pairs in Figure 5.29 upwards. For Mach numbers below the design Mach number, the tip sections begin to operate beyond the positive stall boundary, and the optimized rotor performance begins to degrade. This effect can be seen in Figure 5.30 where the $C_w = 0.0150$ case begins to diverge at

$\mu = 0.44$ as the optimized rotor begins to stall. This can be further observed in Figure 5.32 where replacing the last 5%, 8%, and 15% of the blade length with the optimized airfoil section reduces the maximum weight coefficient in hover by 0.60%, 2.21%, and 4.24%, respectively. Although the tip region itself is not a major thrust source in hover, the stall of the blade tip dramatically increases drag and therefore the power required by the rotor. In addition, the loss of lift at the blade tip also requires the more inboard sections to operate at higher lift coefficients to compensate, which then feeds back and requires even higher collective-pitch inputs and further aggravates the problem.

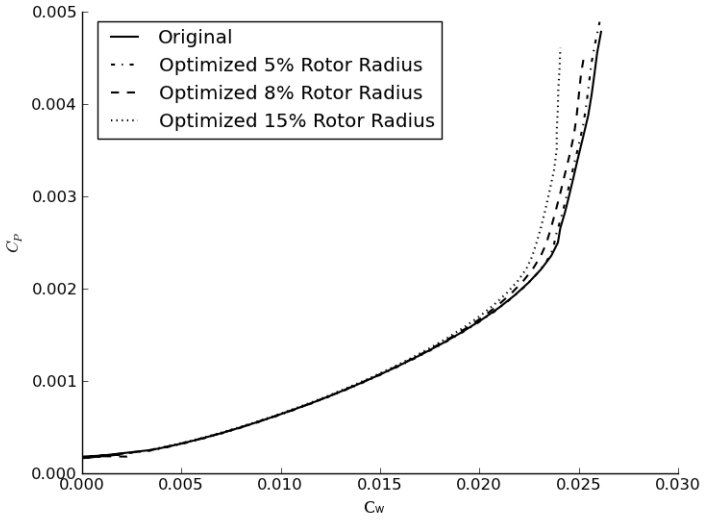


Figure 5.32: Single-point-optimization hover stall margin comparison with optimized airfoil sections of varying length along blade span

5.2.2 Multi design point optimization

The airfoil resulting from the single design point optimization looks atypical. This could potentially be the result of the optimization routine exploiting weaknesses in the OVERFLOW solver and turbulence model at these conditions. It could also potentially be a result of an over-optimization for the design point causing large performance reductions at off-design points. In order to soften the design point, two multi-point clusters were used. A Mach number cluster was employed where the objective function was computed as a weighted average between three design

points spanning $M_\infty = 0.8 \pm 0.05$ as

$$c_d(y) = \frac{1}{4}c_{d_{M=0.75}}(y) + \frac{1}{2}c_{d_{M_\infty=0.80}}(y) + \frac{1}{4}c_{d_{M_\infty=0.85}}(y) \tag{5.1}$$

A similar three-point lift coefficient cluster was also conducted using a range of 0.1 ie.

$$c_d(y) = \frac{1}{4}c_{d_{c_l=-0.501}}(y) + \frac{1}{2}c_{d_{c_l=-0.551}}(y) + \frac{1}{4}c_{d_{c_l=-0.601}}(y) \tag{5.2}$$

For both design clusters, the constraints were not averaged across the points. Instead, they were evaluated at the original design point. Figure 5.33 shows the drag convergence history of the clusters. The vertical axis represents the weighted drag coefficient represented by Equations (5.1) and (5.2). The clusters represent different design points, meaning that the optimization will converge towards different objective function minima even if the resulting shapes are very similar. The single design point optimization and multi-point clusters all converged towards similar shapes and results are for the most part indistinguishable (Figure 5.34). This suggests that the resultant shape may be a valid global optimum.

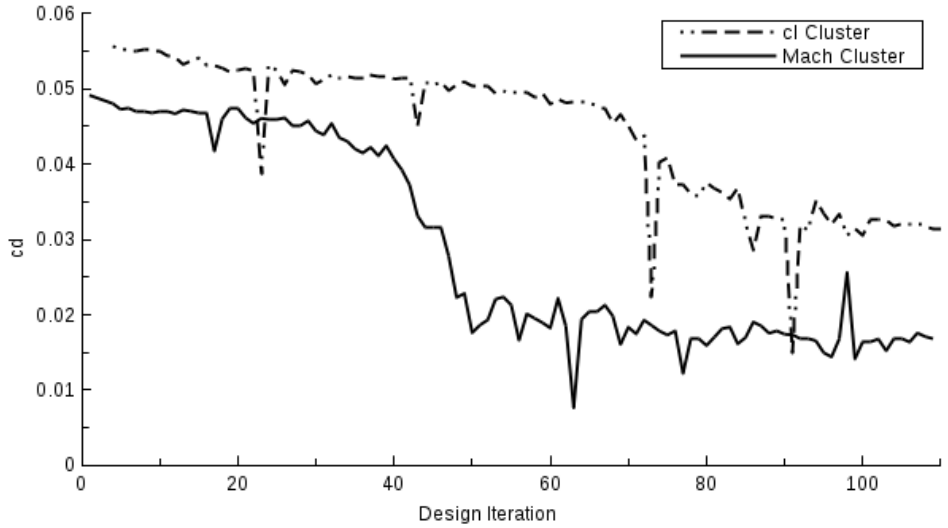


Figure 5.33: Multi-point optimization drag convergence history, c_l and Mach number clusters, $Re = 6 \times 10^6$

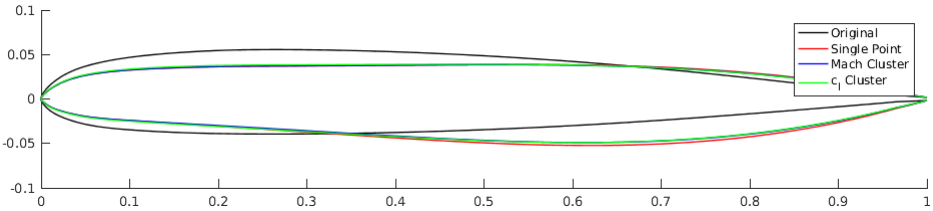


Figure 5.34: Multi-point optimization resultant airfoil shape comparison, c_l and Mach number clusters, $Re = 6 \times 10^6$

Table 5.2: Multi- point optimization results, $M_\infty = 0.8$,$Re = 6 \times 10^6$

	Original	Single-point (Interior)	Mach cluster	c_l cluster
Airfoil area	0.0651	0.0681	0.0658	0.0661
c_m	0.0217	0.02082	0.0213	0.02081
c_l	-0.551	-0.551	-0.551	-0.551
c_d	0.04908	0.0145	0.01478	0.0152

Chapter 6
Conclusion

An airfoil shape optimization method was developed and implemented by using the CMA-ES genetic algorithm to drive the OVERFLOW 2.2i CFD code. The optimization design variables were prescribed through a CST-based airfoil parameterization using the orthogonal Legendre polynomials to form the basis modes. The optimization was extended to rotorcraft applications by coupling a Python framework with OVERFLOW to automate the generation of airfoil performance tables. The generated performance tables were then supplied to a BEMT-based analysis code known as ROTOR in order to simulate the UH-60A rotor used as the baseline case in this work.

A multicollinearity investigation of the parameterization method was conducted using the condition number. The Legendre polynomial basis modes showed significantly better conditioning over the original Bernstein polynomials. An additional completeness study was conducted where the ability of the parameterization to describe a range of airfoils in the UIUC database was investigated. The Legendre polynomials performed similarly compared to the original Bernstein polynomials and were capable of representing a large number of airfoil shapes with a minimal number of design variables.

The performance tables showed good agreement between the OVERFLOW predictions and experimental results for the airfoils' zero-lift angles-of-attack, lift-curve slopes, and zero-lift drag coefficients. The maximum lift coefficient, however, is known to often be over-predicted by CFD-based methods. To overcome this, a $c_{l,max}$ criteria was developed based on the momentum thickness Reynolds number at the airfoil trailing-edge. The criteria was then incorporated into a post-processing correction routine, which acted as a scaling of $c_{l,max}$, while maintaining α_0, $\frac{dc_l}{d\alpha}$,

and the stall characteristics. The criteria proved most accurate at lower Mach numbers where stall prediction is more dependent on the accuracy of the underlying turbulence model used in the solver. The influence of the correction routine was investigated for the baseline rotor. During standard operating conditions, the lift-corrected table presented little change in performance over the raw CFD results. However, an investigation of the rotor performance during hover showed that the lift-corrected table presented a substantial reduction in the rotor's maximum weight coefficient, suggesting that a rotor designed using the raw CFD results could have a lower stall margin than intended.

The optimization methodology was applied to the baseline rotor's tip airfoil (SC1095), with design constraints applied to the airfoil cross-sectional area and pitching moment coefficients using an augmented Lagrangian penalty function. The penalty method was able to enforce the design constraints for both interior and exterior starting points. Although the design spaces were not completely interchangeable between the two starting points, the optimizer approached similar final designs by reducing the leading-edge radius in an attempt to mitigate the shock wave. The optimizer also shifted the airfoil's camber, moving the low-drag region towards the design lift coefficient. The optimization routine was capable of substantially reducing the airfoil's drag at the design condition.

The effectiveness of the optimized airfoil was investigated by using the new airfoil to replace the outboard 5-15% of the baseline rotor radius. At the design Mach number, the increased stall margin of the new airfoil allowed the rotor to mitigate the sharp increase in drag produced during the stall and post-stall regimes, thereby reducing the required power in forward flight. At off-design Mach numbers, however, the optimized airfoil exhibited a lower stall margin, causing a reduction in the rotorcraft's maximum take-off weight.

An additional multi-point optimization was conducted using a weighted Mach number and lift coefficient three-point cluster. The shape resulting from the interior single design point optimization and multi-point clusters were nearly indistinguishable, suggesting that the resultant airfoil may be a valid global optimum.

Future Work

Although the Multicollinearity of the design matrices was investigated using the condition index, it would be helpful to additionally demonstrate how the use of orthogonal polynomials influences the optimization process. The CMA-ES

algorithm inherently provides a metric for evaluating this, as the algorithm uses a standard deviation variable to govern the size of the search ellipse when generating candidate solutions. The orthogonal polynomials should cause the magnitude of the standard deviation to be relatively insensitive to changes in design space dimensions when compared to other parameterization methods.

In the future, a drag coefficient correction should also be implemented into the table generation routine. The prediction of drag is strongly correlated with the lift coefficient, particularly during the stall regime where the airfoil generally experiences a large drag increase as a result of the formation and growth of the separation region.

The airfoil optimization and table generation routines presented in this work showed to be useful for the single design point investigated. This holds promise that the technology developed can later be extended to multi-point optimization loops, allowing for the modification of multiple airfoil sections along the blade span.

Appendix A
Example Input Files/File Format

A.1 C81 Table Format

```
                                                    Read/Write Format
---------------------------------------          -----------------
AIRFOIL_NAME          ML,NL,MD,ND,MM,NM           A30,6I2,6I2
       M(1)      ...     ...    M(ML)             7X,9F7.0
AL(1)  CL(1,1)   ...     ...    CL(1,NL)          10F7.0/(7X,9F7.0)
.      .                        .                 .
.      .                        .                 .
.      .                        .                 .
AL(NL) CL(NL,1)  ...     ...    CL(NL,ML)         10F7.0/(7X,9F7.0)
       M(1)      ...     ...    M(MD)             7X,9F7.0
AD(1)  CD(1,1)   ...     ...    CD(1,ND)          10F7.0/(7X,9F7.0)
.      .                        .                 .
.      .                        .                 .
.      .                        .                 .
AD(ND) CD(ND,1)  ...     ...    CD(ND,MD)         10F7.0/(7X,9F7.0)
       M(1)      ...     ...    M(MM)             7X,9F7.0
AM(1)  CM(1,1)   ...     ...    CM(1,NM)          10F7.0/(7X,9F7.0)
.      .                        .                 .
.      .                        .                 .
.      .                        .                 .
AM(NM) CM(NM,1)  ...    ...    CM(NM,MM)         10F7.0/(7X,9F7.0)

AL = Lift coefficient angles of attack
AD = Drag coefficient angles of attack
AM = Pitching moment coefficient angles of attack
ML = Number of lift coefficient Machs
NL = Number of lift coefficient alphas
MD = Number of drag coefficient Machs
```

```
ND = Number of drag coefficient alphas
MM = Number of moment coefficient Machs
NM = Number of moment coefficient alphas
```

A.2 USURP Input File

```
0, 500, 500, -1, 0, 0   FSMACH,ALPHA,BETA,REY,GAMINF,TINF
1                       NREF
1., 1.0, 0.25, 0., 0.   REFL,REFA,XMC,YMC,ZMC

1                       NSURF

1, 1                    NSUB ,IREFS
1, 3, 1, -1,  1, -1,  1,  1     NG,IBDIR,JS,JE,KS,KE,LS,LE
0                       NPRI

1                       NCOMP

TARGET
1, 1                    NIS,IREFC
1
```

A.3 HYPGEN Input File

```
grd.srf               INPUT SURFACE GRID
grid.in               OUTPUT VOLUME GRID
0                     IFORM(0/1)
1, 1, 1               IZSTRT(1/2/-1),NZREG,KLAYER
129, 1e3, 4e-6, 0.0    NPZREG(),ZREG(),DZ0(),DZ1()
10, 10, 2, 2            IBCJA,IBCJB,IBCKA,IBCKB
1, 0.01, 5, 10        IVSPEC(1/2),EPSSS,ITSVOL,NSUB
2, 0.5                IMETH(0/2/3),SMU2
0.0, 0.0              TIMJ,TIMK
1, 0.3, 0.3           IAXIS(1/2),EXAXIS,VOLRES
```

A.4 OVERFLOW Input File

```
 $GLOBAL
    NSTEPS=14000, RESTRT=.F., NSAVE=500, NQT=302, FMG=.T., MULTIG=.T.,
    NGLVL=2, FMGCYC=1000
```

```
   $END

$FLOINP
   REY=6E6, TINF=518.7, ALPHA=0.0, FSMACH=0.4
   $END

$VARGAM $END
$GRDNAM
   NAME='AIRFOIL'
   $END

$NITERS $END
$METPRM
   IRHS=4, ILHS=4, IDISS=4, ILIMIT=3
   $END

$TIMACU
   DT=0.1, ITIME=1, CFLMIN=5.0
   $END

$SMOACU
   DIS2=0.00, DIS4=.0, ISPEC=2, SMOO=1.0
   $END

$VISINP
   CFLT=1.0, VISC=.T.
   $END

$BCINP
   IBTYP= 5, 21, 47, 10,
   IBDIR= 3,  2, -3,  1,
   JBCS=  1,  1,  1,  1,
   JBCE= -1, -1, -1,  1,
   KBCS=  1,  1,  1,  1,
   KBCE= -1,  1, -1, -1,
   LBCS=  1,  1, -1,  1,
   LBCE=  1, -1, -1, -1
   $END

$SCEINP $END
```

Appendix B
Extended Results

B.1 Optimized Airfoil Coordinates

Upper Surface		Lower Surface	
x	z	x	z
1.00000	0.00173	0.00000	0.00000
0.99986	0.00175	0.00021	-0.00097
0.99937	0.00184	0.00095	-0.00208
0.99854	0.00198	0.00220	-0.00319
0.99737	0.00218	0.00375	-0.00445
0.99404	0.00274	0.00780	-0.00733
0.99189	0.00310	0.01037	-0.00877
0.98942	0.00351	0.01332	-0.01017
0.98664	0.00396	0.01662	-0.01152
0.98356	0.00446	0.02026	-0.01278
0.98018	0.00500	0.02423	-0.01397
0.97252	0.00616	0.03308	-0.01627
0.96827	0.00680	0.03793	-0.01739
0.95898	0.00849	0.04850	-0.01941
0.94862	0.01042	0.06019	-0.02102
0.94305	0.01139	0.06642	-0.02170
0.93723	0.01239	0.07290	-0.02231
0.93115	0.01341	0.07962	-0.02286
0.91148	0.01649	0.10115	-0.02437
0.90446	0.01752	0.10876	-0.02487
0.89721	0.01856	0.11658	-0.02539
0.88975	0.01959	0.12460	-0.02591
0.88207	0.02061	0.13282	-0.02644
0.86610	0.02263	0.14982	-0.02751
0.84934	0.02459	0.16754	-0.02862
0.83184	0.02647	0.18594	-0.02976

0.82282	0.02737	0.19538	-0.03035
0.81364	0.02825	0.20497	-0.03095
0.80429	0.02910	0.21471	-0.03156
0.77531	0.03146	0.24474	-0.03348
0.76536	0.03218	0.25501	-0.03415
0.75527	0.03286	0.26539	-0.03482
0.74506	0.03351	0.27590	-0.03552
0.73472	0.03412	0.28650	-0.03622
0.70300	0.03571	0.31892	-0.03839
0.68134	0.03657	0.34098	-0.03987
0.67037	0.03694	0.35212	-0.04061
0.65932	0.03727	0.36333	-0.04136
0.64818	0.03757	0.37461	-0.04210
0.63697	0.03783	0.38595	-0.04283
0.62570	0.03805	0.39734	-0.04357
0.61436	0.03824	0.40879	-0.04429
0.60297	0.03840	0.42027	-0.04500
0.56850	0.03870	0.45493	-0.04702
0.54534	0.03876	0.47816	-0.04825
0.53372	0.03877	0.48979	-0.04883
0.52209	0.03875	0.50143	-0.04937
0.49878	0.03866	0.52471	-0.05034
0.48713	0.03860	0.53633	-0.05076
0.47548	0.03853	0.54794	-0.05113
0.46384	0.03844	0.55953	-0.05146
0.44061	0.03825	0.58261	-0.05196
0.42903	0.03815	0.59410	-0.05212
0.41748	0.03805	0.60554	-0.05223
0.40597	0.03795	0.61693	-0.05227
0.38308	0.03774	0.63954	-0.05216
0.37172	0.03764	0.65075	-0.05201
0.34917	0.03746	0.67293	-0.05150
0.33800	0.03737	0.68390	-0.05115
0.32690	0.03729	0.69477	-0.05072
0.30496	0.03714	0.71623	-0.04966
0.27275	0.03695	0.74756	-0.04755
0.26223	0.03690	0.75776	-0.04672
0.25182	0.03684	0.76783	-0.04582
0.24152	0.03677	0.77776	-0.04486
0.23136	0.03668	0.78754	-0.04384
0.22132	0.03658	0.79717	-0.04277
0.21142	0.03645	0.80664	-0.04164
0.19205	0.03611	0.82509	-0.03923
0.17330	0.03565	0.84285	-0.03664
0.16417	0.03539	0.85146	-0.03529

0.15520	0.03510	0.85987	-0.03390
0.14641	0.03479	0.86809	-0.03248
0.13781	0.03447	0.87611	-0.03104
0.12938	0.03412	0.88392	-0.02958
0.12116	0.03375	0.89152	-0.02810
0.11313	0.03333	0.89890	-0.02661
0.09029	0.03177	0.91970	-0.02213
0.08312	0.03112	0.92617	-0.02064
0.07619	0.03041	0.93240	-0.01916
0.06949	0.02962	0.93838	-0.01770
0.06304	0.02875	0.94411	-0.01626
0.04526	0.02553	0.95975	-0.01210
0.03991	0.02421	0.96443	-0.01080
0.03486	0.02277	0.96887	-0.00967
0.03013	0.02119	0.97306	-0.00868
0.02574	0.01949	0.97697	-0.00775
0.02169	0.01770	0.98058	-0.00685
0.01798	0.01584	0.98390	-0.00600
0.01462	0.01394	0.98692	-0.00522
0.00892	0.01020	0.99206	-0.00387
0.00657	0.00841	0.99416	-0.00331
0.00456	0.00669	0.99595	-0.00283
0.00288	0.00509	0.99742	-0.00243
0.00152	0.00362	0.99857	-0.00212
0.00057	0.00224	0.99938	-0.00190
0.00010	0.00099	1.00000	-0.00173

References

[1] Bousman, W. G. (2003) *Aerodynamic characteristics of SC1095 and SC1094 R8 airfoils*, *Tech. rep.*, DTIC Document, ADA480634.

[2] Leishman, G. J. (2006) *Principles of helicopter aerodynamics with CD extra*, Cambridge university press.

[3] Anderson, J. D. and J. Wendt (1995) *Computational fluid dynamics*, vol. 206, Springer.

[4] Anderson, G., M. Aftosmis, and M. Nemec (2012) "Constraint-based shape parameterization for aerodynamic design," in *7th International Conference on Computational Fluid Dynamics, Big Island, Hawaii*.

[5] J. Toal, D. J., N. W. Bressloff, A. J. Keane, and C. M. E. Holden (2010) "Geometric filtration using proper orthogonal decomposition for aerodynamic design optimization," *AIAA journal*, **48**(5), pp. 916–928.

[6] Kulfan, B. M. (2008) "Universal parametric geometry representation method," *Journal of Aircraft*, **45**(1), pp. 142–158.

[7] Valarezo, W. O. and V. D. Chin (1994) "Method for the prediction of wing maximum lift," *Journal of Aircraft*, **31**(1), pp. 103–109.

[8] Ballin, M. G. and A. R. Center. (1987) *Validation of a real-time engineering simulation of the UH-60A helicopter*, National Aeronautics and Space Administration, Ames Research Center.

[9] Buning, P. G. and R. H. Nichols (2010) "OVERFLOW 2 Training Class," *University of Alabama and NASA Langley Research Center*.

[10] Blackwell, R. and T. Millott (2008) "Dynamics Design Characteristics of the Sikorsky X2 TechnologyâĎć Demonstrator Aircraft," in *ANNUAL FORUM PROCEEDINGS-AMERICAN HELICOPTER SOCIETY*, vol. 64, AMERICAN HELICOPTER SOCIETY, INC, p. 886.

[11] Sankaran, V., A. Wissink, A. Datta, J. Sitaraman, M. Potsdam, B. Jayaraman, A. Katz, S. Kamkar, B. Roget, D. Mavriplis, et al. (2011) "Overview of the Helios version 2.0 computational platform for rotorcraft simulations," in *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, p. 1105.

[12] Smith, M. J., J. W. Lim, B. G. van der Wall, J. D. Baeder, R. T. Biedron, D. D. Boyd Jr, B. Jayaraman, S. N. Jung, and B.-Y. Min (2013) "The HART II international workshop: an assessment of the state of the art in CFD/CSD prediction," *CEAS Aeronautical Journal*, **4**(4), pp. 345–372.

[13] Slotnick, J., A. Khodadoust, J. Alonso, D. Darmofal, W. Gropp, E. Lurie, and D. Mavriplis (2014) *CFD vision 2030 study: a path to revolutionary computational aerosciences*, *Tech. rep.*, CR-2014-218178 Washington, DC: NASA.

[14] Ganguli, R. (2004) "A survey of recent developments in rotorcraft design optimization," *Journal of Aircraft*, **41**(3), pp. 493–510.

[15] Saberi, H., M. Khoshlahjeh, R. A. Ormiston, and M. J. Rutkowski (2004) "Overview of RCAS and application to advanced rotorcraft problems," in *American Helicopter Society 4th Decennial Specialists' Conference on Aeromechanics, San Francisco, CA*.

[16] Johnson, W. (1994) "CAMRAD II," *Comprehensive Analytical Model of Rotorcraft Aerodynamics and Dynamics. Johnson Aeronautics, Palo Alto, CA*.

[17] van der Wall, B. G., J. W. Lim, M. J. Smith, S. N. Jung, J. Bailly, J. D. Baeder, and D. D. Boyd Jr (2013) "The HART II International Workshop: an assessment of the state-of-the-art in comprehensive code prediction," *CEAS Aeronautical Journal*, **4**(3), pp. 223–252.

[18] Kody, F., E. Corle, M. D. Maughmer, and S. Schmitz (2015) "Higher-Harmonic Deployment of Trailing-Edge Flaps for Rotor-Performance Enhancement and Vibration Reduction," *Journal of Aircraft*.

[19] Chattopadhyay, A. and J. L. Walsh (1990) "Application of optimization methods to helicopter rotor blade design," *Structural optimization*, **2**(1), pp. 11–22.

[20] Miura, H. (1984) "Applications of numerical optimization methods to helicopter design problems: A survey," in *15th European Rotorcraft Forum*, NASA Ames Research Center.

[21] Adelman, H. M. and W. R. Mantay (1989) *Integrated multidisciplinary design optimization of rotorcraft*, NASA, Langley Research Center.

[22] Celi, R. (1999) "Recent applications of design optimization to rotorcraft- a survey," *Journal of Aircraft*, **36**(1), pp. 176–189.

[23] Mayda, E. and C. van Dam (2005) "Automated Generation of Airfoil Performance Tables Using a Two-Dimensional Navier-Stokes Solver," *Journal of the American Helicopter Society*, **50**(4), pp. 338–348.

[24] Pulliam, T. H. (1986) "Efficient solution methods for the Navier-Stokes equations," in *In Von Karman Inst. for Fluid Dynamics Numerical Techniques for Viscous Flow Calculations in Turbomachinery Bladings 104 p (SEE N86-30988 22-34)*.

[25] Maughmer, M. D. and J. G. Coder (2010) *Comparisons of theoretical methods for predicting airfoil aerodynamic characteristics*, *Tech. rep.*, DTIC Document, ADA532502.

[26] Rumsey, C. L. and S. X. Ying (2002) "Prediction of high lift: review of present CFD capability," *Progress in Aerospace Sciences*, **38**(2), pp. 145–180.

[27] Quagliarella, D. and A. Della Cioppa (1995) "Genetic algorithms applied to the aerodynamic design of transonic airfoils," *Journal of aircraft*, **32**(4), pp. 889–891.

[28] Jones, B. R., W. A. Crossley, and A. S. Lyrintzis (2000) "Aerodynamic and aeroacoustic optimization of rotorcraft airfoils via a parallel genetic algorithm," *Journal of Aircraft*, **37**(6), pp. 1088–1096.

[29] Hansen, N. (2016) "The CMA evolution strategy: A tutorial," *arXiv preprint arXiv:1604.00772*.

[30] Hicks, R. M. and P. A. Henne (1978) "Wing design by numerical optimization," *Journal of Aircraft*, **15**(7), pp. 407–412.

[31] Sobieczky, H. (1999) "Parametric airfoils and wings," in *Recent Development of Aerodynamic Design Methodologies*, Springer, pp. 71–87.

[32] Poole, D. J., C. B. Allen, and T. C. Rendall (2015) "Metric-based mathematical derivation of efficient airfoil design variables," *AIAA Journal*, **53**(5), pp. 1349–1361.

[33] Ghoman, S., Z. Wang, P. Chen, and R. Kapania (2012) "A POD-based reduced order design scheme for shape optimization of air vehicles," in *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 20th AIAA/ASME/AHS Adaptive Structures Conference 14th AIAA*, p. 1808.

[34] Ceze, M., M. Hayashi, and E. Volpe "A study of the CST parameterization characteristics," in *27th AIAA Applied Aerodynamics Conference*, p. 3767.

[35] Sripawadkul, V., M. Padulo, and M. Guenov (2010) "A comparison of airfoil shape parameterization techniques for early design optimization," in *13th AIAA/ISSMO multidisciplinary analysis optimization conference*, p. 9050.

[36] Nadarajah, S., P. Castonguay, and A. Mousavi (2007) "Survey of shape parameterization techniques and its effect on three-dimensional aerodynamic shape optimization," in *18th AIAA computational fluid dynamics conference*, p. 3837.

[37] Vassberg, J., N. Harrison, D. Roman, and A. Jameson (2011) "A systematic study on the impact of dimensionality for a two-dimensional aerodynamic optimization model problem," in *29th AIAA Applied Aerodynamics Conference*, p. 3176.

[38] Chen, R. T. (1989) "A survey of nonuniform inflow models for rotorcraft flight dynamics and control applications," in *15th European Rotorcraft Forum*, NASA Ames Research Center.

[39] Glauert, H. (1927) "The theory of the autogyro," *Journal of the Royal Aeronautical Society*, **31**(198), pp. 483–508.

[40] Jameson, A. (1989) "Aerodynamic design via control theory," in *Recent advances in computational fluid dynamics*, Springer, pp. 377–401.

[41] Wu, H.-Y., S. Yang, F. Liu, and H.-M. Tsai (2003) "Comparisons of Three Geometric Representations of Airfoils for Aerodynamic Optimization," in *16th AIAA computational fluid dynamics conference*, p. 4095.

[42] Désidéri, J.-A. and A. Janka (2004) "Multilevel Shape Parameterization for Aerodynamic Optimization: Application to Drag and Noise Reduction of Transonic/Supersonic Business Jet," in *Proceedings of the European Congress on Computational Methods in Applied Sciences and Engineering, ECCOMAS 2004*.

[43] McCormick, B. (2002) "A numerical analysis of autogyro performance," in *2002 Biennial International Powered Lift Conference and Exhibit*, p. 5950.

[44] Hartwich, R. (1999), "Parametric Study for the Design of Advanced Rotorcraft Airfoils," .

[45] Kinzel, M. P. (2004), "Miniature Trailing-Edge Effectors for Rotorcraft Applications," .

[46] Nichols, R. H. and P. G. Buning (2008) "User's Manual for OVERFLOW 2.1," *University of Alabama and NASA Langley Research Center*.

[47] Drela, M. (1989) "Xfoil: An analysis and design system for low reynolds number airfoils," *University of Notre Dame*.

[48] Drela, M. and M. B. Giles (1987) "Viscous-inviscid analysis of transonic and low Reynolds number airfoils," *AIAA journal*, **25**(10), pp. 1347–1355.

[49] Bielawa, R. L. and K. Frederickson (1995) *Development of a System for Improved Helicopter Blade Tracking.*, *Tech. rep.*, DTIC Document, ADA315089.

[50] Pandya, S., S. Venkateswaran, and T. Pulliam (2003) "Implementation of preconditioned dual-time procedures in overflow," in *41st Aerospace Sciences Meeting and Exhibit*, p. 72.

[51] Beam, R. M. and R. F. Warming (1976) "An implicit finite-difference algorithm for hyperbolic systems in conservation-law form," *Journal of computational physics*, **22**(1), pp. 87–110.

[52] Pulliam, T. H. and D. Chaussee (1981) "A diagonal form of an implicit approximate-factorization algorithm," *Journal of Computational Physics*, **39**(2), pp. 347–363.

[53] Klopfer, G., C. Hung, R. Van der Wijngaart, and J. Onufer "A diagonalized diagonal dominant alternating direction implicit (D3ADI) scheme and subiteration correction," in *29th AIAA, Fluid Dynamics Conference*, p. 2824.

[54] Bardina, J. and C. Lombard (1987) "Three dimensional hypersonic flow simulations with the CSCM implicitupwind Navier-Stokes method," in *8th Computational Fluid Dynamics Conference*, p. 1114.

[55] Huang, P., T. Coakley, and P. Buning (1998) "On Iterative Diagonally Dominant ADI Algorithm," *To be published as a NASA TM*.

[56] Pulliam, T., R. MacCormack, and S. Venkateswaran (1998) "Convergence characteristics of approximate factorization methods," in *Sixteenth International Conference on Numerical Methods in Fluid Dynamics*, Springer, pp. 409–414.

[57] Nichols, R., R. Tramel, and P. Buning (2006) "Solver and turbulence model upgrades to OVERFLOW 2 for unsteady and high-speed applications," in *24th AIAA Applied Aerodynamics Conference*, p. 2824.

[58] Kandula, M. and P. Buning "Implementation of LU-SGS algorithm and Roe upwinding scheme in OVERFLOW thin-layer Navier-Stokes code," in *Fluid Dynamics Conference*, p. 2357.

[59] Spalart, P. and S. Allmaras (1992) "A one-equation turbulence model for aerodynamic flows," in *30th aerospace sciences meeting and exhibit*, p. 439.

[60] Coder, J. and M. Maughmer (2013) "A CFD-compatible transition model using an amplification factor transport equation," in *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, p. 253.

[61] Chan, W., R. Gomez, S. Rogers, and P. Buning (2002) "Best practices in overset grid generation," in *32nd AIAA Fluid Dynamics Conference and Exhibit*, p. 3191.

[62] Chan, W. M., S. E. Rogers, S. M. Nash, P. G. Buning, and R. Meakin (2003) "User's manual for Chimera grid tools, version 1.8," *NASA Ames Research Center*.

[63] Boger, D. and J. Dreyer (2006) "Prediction of hydrodynamic forces and moments for underwater vehicles using overset grids," in *44th AIAA Aerospace Sciences Meeting and Exhibit*, p. 1148.

[64] Johnson, W. (1998) "Rotorcraft Aerodynamics Models for a Comprehensive Analysis," in *American Helicopter Society Forum*, Johnson Aeronautics.

[65] Smith, M. J., N. D. Liggett, and B. C. Koukol (2011) "Aerodynamics of airfoils at high and reverse angles of attack," *Journal of Aircraft*, **48**(6), pp. 2012–2023.

[66] O. Smith, A. (1975) "High-lift aerodynamics," *Journal of Aircraft*, **12**(6), pp. 501–530.

[67] Coder, J. G. and M. D. Maughmer (2014) "Numerical Validation of the Squire–Young Formula for Profile-Drag Prediction," *Journal of Aircraft*, **52**(3), pp. 948–955.

[68] Coder, J. G., M. D. Maughmer, and P. B. Martin (2011) "CFD investigation of unsteady rotorcraft airfoil aerodynamics: Mites and dynamic stall," in *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, p. 1125.

[69] Smith, M. J., M. Potsdam, T.-C. Wong, J. D. Baeder, and S. Phanse (2006) "Evaluation of Computational Fluid Dynamics to Determine Two-Dimensional Airfoil Characteristics for Rotorcraft Applications," *Journal of the American Helicopter Society*, **51**(1), pp. 70–79.

[70] Farouki, R. T. (2000) "Legendre–Bernstein basis transformations," *Journal of Computational and Applied Mathematics*, **119**(1), pp. 145–160.

[71] Belsley, D. A., E. Kuh, and R. E. Welsch (2005) *Regression diagnostics: Identifying influential data and sources of collinearity*, vol. 571, John Wiley & Sons.

[72] Hair, J. F., W. C. Black, B. J. Babin, R. E. Anderson, R. L. Tatham, et al. (1998) *Multivariate data analysis*, vol. 5, Prentice hall Upper Saddle River, NJ.

[73] Selig, M. (2000) "UIUC airfoil coordinates database," *University of Illinois at Urbana-Champaign*.

[74] Hansen, N. and A. Ostermeier (2001) "Completely derandomized self-adaptation in evolution strategies," *Evolutionary computation*, **9**(2), pp. 159–195.

[75] Hansen, N., S. D. Müller, and P. Koumoutsakos (2003) "Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES)," *Evolutionary computation*, **11**(1), pp. 1–18.

[76] Kramer, O., A. Barthelmes, and G. Rudolph (2009) "Surrogate constraint functions for CMA evolution strategies," in *Annual Conference on Artificial Intelligence*, Springer, pp. 169–176.

[77] Arnold, D. V. and N. Hansen (2012) "A (1+ 1)-CMA-ES for constrained optimisation," in *Proceedings of the 14th annual conference on Genetic and evolutionary computation*, ACM, pp. 297–304.

[78] Rosen, J. B. (1960) "The gradient projection method for nonlinear programming. Part I. Linear constraints," *Journal of the Society for Industrial and Applied Mathematics*, **8**(1), pp. 181–217.

[79] Coello, C. A. C. (2002) "Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art," *Computer methods in applied mechanics and engineering*, **191**(11), pp. 1245–1287.

[80] Myung, H. and J.-H. Kim (1998) "Hybrid interior-lagrangian penalty based evolutionary optimization," in *Evolutionary Programming VII*, Springer, pp. 85–94.

[81] Adeli, H. and N.-T. Cheng (1994) "Augmented Lagrangian genetic algorithm for structural optimization," *Journal of Aerospace Engineering*, **7**(1), pp. 104–118.

[82] Powell, M. J. (1978) "Algorithms for nonlinear constraints that use Lagrangian functions," *Mathematical programming*, **14**(1), pp. 224–248.

[83] Hestenes, M. R. (1969) "Multiplier and gradient methods," *Journal of optimization theory and applications*, **4**(5), pp. 303–320.

[84] Myung, H. and J.-H. Kim (1997) "Evolian: evolutionary optimization based on Lagrangian with constraint scaling," in *Evolutionary Programming VI*, Springer, pp. 177–187.