Spring 1-1-2018

Isogeometric Analysis of Subsonic Aerodynamic Flows with Application to Shape Optimization

Ansel Rothstein-Dowden
University of Colorado at Boulder, anro6767@colorado.edu

Isogeometric Analysis of Subsonic Aerodynamic Flows with Application to Shape Optimization

by

A. Rothstein-Dowden

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Master of Science

Ann and H. J. Smead Department of Aerospace Engineering Sciences

2018

This thesis entitled:
Isogeometric Analysis of Subsonic Aerodynamic Flows with Application to Shape Optimization
written by A. Rothstein-Dowden
has been approved for the Ann and H. J. Smead Department of Aerospace Engineering Sciences

_____

Dr. John A. Evans

_____

Dr. Brian Argrow

_____

Dr. John Farnsworth

Date _____

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Rothstein-Dowden, A. (M.S., Aerospace Engineering Sciences)

Isogeometric Analysis of Subsonic Aerodynamic Flows with Application to Shape Optimization

Thesis directed by Dr. John A. Evans

The preliminary design phase of any engineering project is characterized by computationally efficient low-fidelity predictive modeling to inform early-stage design choices. It is essential to have fast, efficient tools for preliminary analysis in order to maximize the amount of time that can be allocated to the detailed design phase, where high-fidelity models must be used to determine design parameters as accurately as possible.

Much of the relevant analysis applicable to engineering systems is described by partial differential equations, to which the analytical solution is known for a very small class of problems. Instead we resort to compuational methods. One of the most common ways of approximating the solution to a partial differential equation in engineering applications is the finite element method. However, this method and other like it introduce a substantial computational bottleneck in the design process because of their indirect link with the geometric representation of engineering designs. The recently developed field of isogeometric analysis shows great promise due to its ability to circumvent the design-to-analysis bottleneck inherent in conventional finite element methods. More recently still, isogeometric analysis has been utilized to perform shape optimization, providing the potential to eliminate the design-analysis loop altogether. Low-fidelity automatic optimization routines offer to preliminary design the distinct advantage of replacing heuristic human-guided design iteration with mathematically informed iteration toward the optimal design for a system.

In this thesis we propose a novel method for the analysis of an airfoil in high Reynolds number sub-sonic flow as a tool for early stage aircraft design. The geometry of the airfoil is left arbitrary and the airfoil may consist of multiple distinct bodies. We investigate the validity of the method and apply the method to a surrogate optimization problem.

Dedication

For Lily and Kiki, whose silent support helped make this thesis paws-ible.

Acknowledgements

The author wishes to acknowledge Dr. John Evans for his invaluable support and guidance throughout this learning process. He is an inspiration and without him this thesis would not have been possible. The author also wishes to acknowledge all those in the Computational Mechanics and Geometry Laboratory who have offered their thoughts and opinions. At the top of this list are Joe Benzaken (for all the "math questions"), Luke Engvall, Eric Peters, and Corey Wetterer-Nelson. Finally, the author wishes to acknowledge Syndey Richardson for her patience and for all her non-technical support.

Contents

Chapter

List of Tables

Table

List of Figures

Figure

Chapter 1

Introduction

This chapter serves to introduce the reader to the topics covered in the body of this thesis. This will include the motivation behind the work done, the accomplishments of this thesis, and a guide to aid the reader in navigating the contents and nomenclature of the text.

1.1 Motivation

One of the most well-known open problems in modern mathematics is the existence and uniqueness of solutions to the Navier-Stokes equations. The Navier-Stokes equations are a set of partial differential equations (PDEs) which govern the properties of a continuum model of a fluid. The applicability of the solution to these equations is widespread in the field of engineering. The role of fluids is crucial in thermodynamic heat exchange systems, hydraulic systems, propulsion systems, atmospheric affects on spacecraft systems, and, important to this thesis, aircraft systems (to name a few). These examples help show the range of properties which are relevant to the study of fluid flow, all of which encompassed by the Navier-Stokes equations.

Because this problem is unsolved, immense effort in the scientific community has been devoted to the derivation of numerical solution techniques for approximating the behavior of fluids in various settings. Perhaps some of the most well-known numerical methods for solving the Navier-Stokes equations and, in fact, PDEs in general, are the Finite Difference Method, the Finite Volume Method, and the Finite Element Method (FEM), or Finite Element Analysis (FEA). FEA is a method for solving PDEs in which the variational or weak form of the PDE is approximated

using weighted residual methods, or, more specifically, Bubnov-Galerkin and Petrov-Galerkin approximation methods [20]. In classical FEA, piecewise linear functions (the finite element 'basis functions') are used to approximate to the solution to the weak form of a given PDE over the domain of definition. In this sense the solution geometry is imposed by the analysis discretization, and thus classical FEA often misrepresents the true geometry of the domain. Accuracy losses are therefore two-fold, in that the solution as well as the geometry are in general not well-approximated by piecewise linear functions. Even application of higher-order finite element bases which may more accurately approximate the solution to a PDE still impose erroneous geometry as an approximation to even the simplest of engineering designs, but this issue will be addressed more thoroughly in a later section.

Not only does FEA fundamentally misrepresent most engineering design geometries, but in doing so it also creates a substantial bottle-neck in the engineering design process. A study at Sandia National Laboratories suggested that approximately 70% of the engineering design-analysis cycle is devoted to FEA model construction and solutions [11]. This computational cost is rooted in the indirect link between design geometry and analysis. In recent years, concern over this issue has brought about the field of Isogeometric Analysis (IGA), first proposed in 2005 in [11, 21], wherein the basis functions used to describe geometry are themselves used to perform analysis. This embeds the exact geometry of the domain of a given PDE into the solution automatically. IGA is now a rapidly growing field which has shown great promise in outperforming classical FEA in many engineering applications. Chief among its advantages is its ability to exactly represent geometries generated in many commonly used Computer Aided Design (CAD) packages. IGA is useful not only in its ability to accurately reflect the geometry, but also in circumventing the domain decomposition required for classical FEA. In this thesis we will adopt concepts from IGA to perform analysis on aerodynamic configurations.

Of interest in this thesis are efforts that have been made to quantify the flow properties of air around arbitrarily shaped aerodynamic bodies in a two-dimensional setting. This particular application of the Navier-Stokes equations is relevant to the preliminary design process of aircraft,

from radio-controlled hobby planes to United States Air Force X-Planes to commercial passenger aircraft [3]. Of considerable interest in the early stages of aircraft design is wing design, as the wing provides the majority of the lifting force which allows any aircraft to remain airborne. Cross-sections of the wing, known as airfoils, can be studied independently of the full wing to approximate the lifting capability of the full wing by neglecting 3D effects. This thesis will examine a new method for characterization of aerodynamic properties of airfoils in the subsonic regime.

There has been a great deal of work done to address the problem of subsonic aerodynamic flows around airfoils spanning a variety of avenues for analysis. These range from conformal mappings, described in [2], to panel methods, used in [12] and described in [24] and [3], to classical FEA applied in [5] [17], [51], to IGA [16, 19, 28, 34]. Closest to our goal in this thesis is MIT's open-source software package XFOIL [12]. This program is designed to perform low-fidelity airfoil analysis using coupled inviscid and viscid flow solvers at low Reynolds number. It is capable of modeling some limited flow separation regimes and transition from laminar to turbulent flow. It is also capable of targeting desired airfoil characteristics. In this thesis we propose an IGA-based airfoil flow solver which lays the foundation for an isogeometric counterpart to the XFOIL software package.

To derive a formulation for our IGA-based flow solver, we turn to Boundary Element Methods (BEM) [40] which offer several advantages in practical analysis, including a dramatic dimensionality reduction, consequently mitigating computational expense as well as precluding an otherwise necessary volume parametrization. These methods are applied to strongly elliptic problems where solutions are desired on the domain boundary rather than the entire solution domain over which the PDE is posed. This is particularly relevant to the problem of the flow around an airfoil because as we will see in a later section, the aerodynamic properties of interest are independent of the flow conditions at any finite distance from the surface of the body. In the most general terms, the method of boundary elements is performed by restating the strong form of the (elliptic) PDE in boundary integral form, then applying the (Petrov-) Galerkin method to the resulting integral equation. If, in the application of the Galerkin method to integral equations, splines [11] are used

in lieu of Lagrange polynomials [20] for the trial space, an isogeometric analog to classical BEM is conceived. It is typical in such methods to also employ collocation methods [4] because of the resulting reduced computational cost. Collocation-based isogeometric BEM have been successfully implemented in a variety of cases, such as for Stokes flow in [18], the Helmholtz equations in [36], and elastostatics in [29, 41, 43] and [42], and to the Laplace equation in [48].

Because of the direct link between geometry and analysis provided by IGA, a recent avenue of exploration has been in isogeometric shape optimization because of the ability it gives to calculate analytic sensitivities to geometric variation. Most of these papers focus on elastostatics [10, 38, 52] with the exception of [27] which studied vibrational modes in the wave equation, but some have studied shape optimization in a fluid dynamics context [23, 35]. To the author's knowledge, combination of analytic sensitivities from IGA and dimensionality reduction from BEM has seen little study [25]. It is desirable to develop fast, efficient methods for shape optimization because such methods offer the ultimate, automatic closure of the engineering design cycle. While isogeometric analysis obviates the need for geometry decomposition, Isogeometric shape optimization eliminates the need for any human intervention altogether, requiring only a reasonable initial design from a human user.

In this thesis we propose a collocation-based isogeometric boundary element method for determining airfoil properties in an inviscid, incompressible subsonic flow. We explore opportunities for shape optimization made possible by this solution technique.

1.2 Accomplishments

In completing this thesis, the following have been accomplished.

(1) We have proposed a novel solution routine for inviscid, incompressible flow in two dimensions. This routine is capable of predicting the velocity field at the surface of an arbitrary number of arbitrarily shaped airfoils.

(2) We have confirmed the accuracy of the solutions produced by comparison with analytical

results for the specific case of a cylinder in a uniform flow field.

(3) We have demonstrated a per-degree-of-freedom superiority of the method's accuracy when compared to existing solution techniques.

(4) We have computed the formulae for the shape sensitivities necessary for gradient-based optimizations which rely on the solutions generated by the proposed method.

(5) We have used the solution technique to perform shape optimization on an example problem.

1.3 Thesis Structure

This thesis will address the following subjects. First, we examine the mathematical primitives of isogeometric analysis in Chapter 2, outlining the basic tools necessary for understanding what it means to perform isogeometric analysis. We will also derive the specific analysis tools for use in our solution procedure. Next in Chapter 3, the we will present the mathematical problem under study in its continuous form as well as in its discretized form. We will then examine the necessary considerations for successful implementation of the method. To conclude Chapter 3 we will present and discuss numerical results. In Chapter 4 we will discuss an example optimization problem and present the numerical results. Shape sensitivities for optimization are included. Finally we conclude in Chapter 5 with an overview of the work presented and suggest avenues for future work.

A substantial amount of mathematical notation is introduced throughout the body of this thesis. The most common or important notational devices are listed here for reference in Table 1.1. Some notation has been omitted from this list because of its explicit definition in the text.

$B_{i,p}$	The i^{th} Bernstein basis function of polynomial degree p
C	Bezier extraction operator
C_P	Coefficient of pressure
$\mathbf{c}$	The location of the center of an ideal vortex
$\mathbf{C}$	A curve in physical space mapped from parametric space
$\mathbf{d}$	A set of design variables
$\hat{\mathbf{e}}_\theta^{c_i}$	The angular basis vector of a polar coordinate system centered at $\mathbf{c}_i$
$\mathcal{F}$	An objective function
g	The number of bodies in a flow
G	Green's function
h	Normal derivative of the Green's function
$\mathbf{J}$	Jacobian of the mapping from parametric space to physical space
$\tilde{\mathbf{J}}$	Jacobian of the mapping from reference space to physical space
L	Lift force
n	Number of basis functions
$\mathbf{n}$	Outward normal to a closed curve
$N_{i,p}$	The i^{th} B-spline basis function of polynomial degree p
$\mathbf{P}_i$	The i^{th} control point defining a parametric curve
$R_{i,p}$	The i^{th} NURBS basis function of polynomial degree p
$\mathbf{t}$	Tangent to a curve
V	Flow speed
$\mathbf{V}$	Velocity vector
$x_{\alpha i}$	$(\mathbf{P}_i)_\alpha$
$\mathbf{x}$	Location in phyisical space
$(\cdot)_j$	The j^{th} component of $(\cdot)$
Γ	Physical domain boundary
ξ	Parametric domain independent variable
$\tilde{\xi}$	Reference domain independent variable
Ξ	Knot vector
ν	Angle of attack
ρ	Density
τ	Greville abscissa
ϕ	Disturbance potential corresponding to uniform flow
ψ	disturbance potential corresponding to an ideal vortex
Ω	Physical domain
Subscript	
∞	Free stream value
d	Dirichlet
n	Neumann
Superscript	
b	Values local to an element
e	Element number
h	Discretized approximation

Table 1.1: Notational convention used in this thesis

Chapter 2

Isogeometric Analysis

In the conventional engineering design process, design geometries are created with computer-aided design (CAD) packages and analyzed using finite element analysis (FEA). There is currently a substantial gap limiting fluidity between these processes, at the root of which is a mismatch between the underlying mathematical description of geometry within CAD packages and the mathematical tools used to perform analysis in conventional FEA. Isogeometric analysis (IGA) is a technique in computational mechanics which leverages a unifying mathematical framework to define object geometry and to perform analysis, bridging the aforementioned gap between design and analysis. For a visual clarification of this concept, see Figure 2.1. IGA obviates the computationally expensive process of geometric decomposition and object meshing which are necessary for application of classical finite element methods.

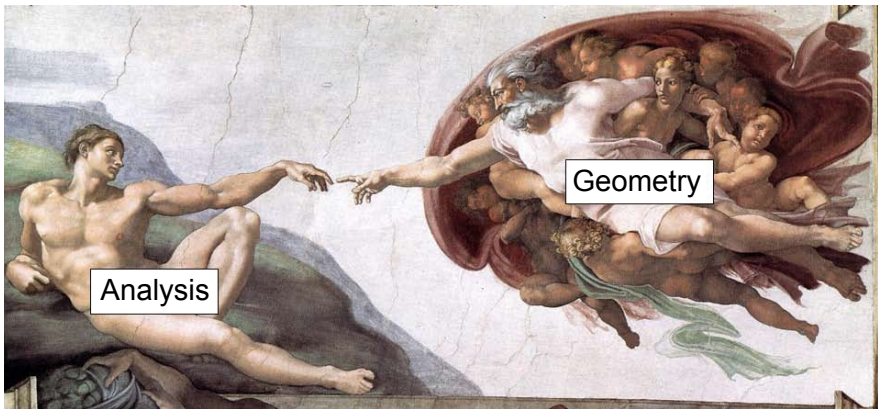

Figure 2.1: Isogeometric analysis provides a direct connection between geometry and analysis. Original painting by Michelangelo, *The Creation of Adam*. Image borrowed from [11].

In many modern CAD packages, the functions used to describe geometry are non-uniform rational B-splines (NURBS). When used for analysis in place of the standard finite element basis functions, these NURBS functions offer the advantage of higher order accuracy and robustness per degree of freedom than the typical polynomial finite element basis, as well as eliminate the design-to-analysis bottleneck caused by the costly process of finite element mesh construction [21].

The direct correlation between analysis and geometry is particularly advantageous in the context of shape optimization. In IGA the basis functions for analysis and geometry share a parametrization, and thus sensitivities of analysis objectives with respect to shape variables can be calculated directly. This paves the way for gradient-based optimization techniques which close the design loop.

To begin any discussion of shape optimization, however, we must first discuss the nature of the mathematics of IGA formally. We will discuss the basis functions that make IGA unique as well as explore the notion of refinement in the isogeometric context.

2.1 The Distinction Between FEA and IGA made Formal

Our previous discussions have emphasized the ability of IGA to exactly capture geometry and insisted upon its superiority over typical FEM. What exactly is the difference between the two? In this section we will briefly formalize the distinction with a motivating example.

To perform FEA or IGA, we aim to approximate a PDE for a function $u : \Omega \to \mathbb{R}^m$ defined over some region of space $\Omega \subset \mathbb{R}^n$ with boundary $\Gamma = \overline{\Gamma_d \cup \Gamma_n}$. For motivational purposes we will simplify the problem by assuming m=1 and n=3, and that the PDE under study is linear with even degree k. This problem can in general be stated as follows.

$$\mathcal{L}u(\mathbf{x}) = f(\mathbf{x}), \qquad \mathbf{x} \in \Omega \tag{2.1}$$

$$\mathcal{B}_d u(\mathbf{x}) = g(\mathbf{x}), \qquad \mathbf{x} \in \Gamma_d \tag{2.2}$$

$$\mathcal{B}_n u(\mathbf{x}) = h(\mathbf{x}), \qquad \mathbf{x} \in \Gamma_h \tag{2.3}$$

where $\mathcal{L}$ is a k^{th} order linear differential operator, $\mathcal{B}_d u$ is the Dirichlet (or "essential") boundary condition, and $\mathcal{B}_n u$ is the Neumann (or "natural") boundary condition. It is in general difficult to solve 2.3 in this form. Instead, we define a space of "trial solutions"

$$\mathcal{S} = \{u \mid u \in H^{k/2}(\Omega),\ \mathcal{B}_d u(\mathbf{x}) = g(\mathbf{x})\ \forall \mathbf{x} \in \Gamma_d\} \tag{2.4}$$

where $H^k(\Omega)$ is a Sobolev space. We also define as a set of "weighting functions"

$$\mathcal{V} = \{w \mid w \in H^{k/2}(\Omega),\ \mathcal{B}_d w(\mathbf{x}) = 0\ \forall \mathbf{x} \in \Gamma_d\} \tag{2.5}$$

We then proceed to apply a weighting function to 2.1 and integrate over the domain as follows

$$\int_\Omega w\mathcal{L}u d\Omega = \int_\Omega w f d\Omega \tag{2.6}$$

We then seek to rearrange 2.6 into the form

$$a(w, u) = L(w) \tag{2.7}$$

where $a(\cdot, \cdot)$ is a symmetric bilinear operator, and $L(\cdot)$ is a linear operator. This task is accomplished through a combination of $k/2$ applications of integration by parts and application of the definitions of $\mathcal{S}$ and $\mathcal{V}$ to simplify terms that appear. Assuming such a manipulation is possible, the problem has been reformulated such that we wish to find $u \in \mathcal{S}$ such that

$$a(w,u) = L(w) \qquad \forall w \in \mathcal{V} \tag{2.8}$$

The problem is now in a form where we can apply Galerkin's method, which is to discretize the solution space. This is accomplished by assuming that the test and trial functions can be relegated to a particular finite-dimensional subspace (dimension n) of the actual test and trial space. Our objective is then to find $u^h \in \mathcal{S}^h \subset \mathcal{S}$ such that

$$a(w^h,u^h) = L(w^h) \qquad \forall w^h \in \mathcal{V}^h \subset \mathcal{V} \tag{2.9}$$

Without loss of generality, if we assume $\mathcal{V}^h = \text{span}\{N_i\}_{i=1}^n$, then it is possible to show that the solution for $u^h = \sum_{i=1}^{n} c_i N_i$ can be obtained via a system of n linear equations for $\{c_i\}$ independent of the choice of w^h, and thus an approximation to u has been found [15, 20].

We can now note that the distinction between classical FEA and IGA lies in the choice of $\mathcal{V}^h$. The set of basis functions $\{N_i\}$ used in classical FEA are designed to interpolate points in acceptable proximity to Ω. The choice of interpolating polynomials is an intuitive one for representing an approximate curve, but comes with the undesirable consequence that the geometry of the basis is imposed on the geometry of the solution. Since the geometry of the solution is already defined by the problem, the more robust approach would be to allow the pre-existing geometry to define the solution behavior. IGA accomplishes this by defining basis functions used to approximate the solution as those that define the geometry of Ω, and as such the true geometry of the domain is preserved identically. We will explore the properties of these basis functions in the coming sections.

2.2 B-Splines and NURBS

In conventional CAD packages, splines are used ubiquitously to define geometry. If we wish to preserve geometric information during analysis, our previous discussion dictates the use of these

splines as a basis. In this section we will examine the definition and properties of two commonly used types of splines – B-splines and NURBS.

2.2.1 B-Splines

This section aims to provide an overview of the fundamental concepts necessary for fluency with and the implementation of isogeometric analysis. We begin our discussion with the definition of a B-spline curve. To define a B-spline, it is necessary to define a dimension d for the resulting curve, a polynomial degree of the basis functions p, the number of basis functions n, and the knot vector $\Xi = [\xi_1 \ \xi_2 \ \cdots \ \xi_{n+p+1}]$, which among other things defines the parametric domain. The elements of Ξ are necessarily monotonically increasing and are referred to as knots. The i^{th} B-spline basis function which we will denote $N_{i,p}(\xi)$ is defined in the parametric space $\xi \in [\xi_1, \ \xi_{n+p+1}]$ and is a recursive function of lower polynomial order B-spline basis functions. The recursion relation is given below [21].

$$N_{i,p}(\xi) = \left(\frac{\xi - \xi_i}{\xi_{i+p} - \xi_i}\right) N_{i,p-1}(\xi) + \left(\frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}}\right) N_{i+1,p-1}(\xi)$$
$$N_{i,0}(\xi) = \begin{cases} 1, & \xi \in [\xi_i, \xi_{i+1}) \\ 0, & \xi \notin [\xi_i, \xi_{i+1}) \end{cases} \tag{2.10}$$

We note here that in the case where the fractions have denominator 0, the fractions themselves are set to 0.

B-splines have a number of useful properties which make them better-suited to use for analysis than the standard finite element basis. Notably, if a knot has multiplicity k, then the basis functions are C^{p-k} continuous at that knot. Additionally, the B-spline basis functions have the property of local support, i.e. $N_{i,p}(\xi) \neq 0 \iff \xi \in [\xi_i, \ \xi_{i+p+1})$ [21].

To form a parametric curve in $\mathbb{R}^d$ from these basis functions, we simply define the set $\{\mathbf{P}_i\} \in \mathbb{R}^d$, $i \in [1, \ n] \cap \mathbb{Z}$ to be the so-called control points, which define the geometry. These control points

are blended in the parametric domain to form a curve $\mathbf{C}$ of the form

$$\mathbf{C}(\xi) = \sum_{i=1}^{n} \mathbf{P}_i N_{i,p}(\xi) \tag{2.11}$$

An example is shown below in Figure 2.2 for $d = 2$, $p = 2$, $n = 8$, and $\Xi = [0\,0\,0\,1\,2\,3\,4\,4\,5\,5\,5]$ Note

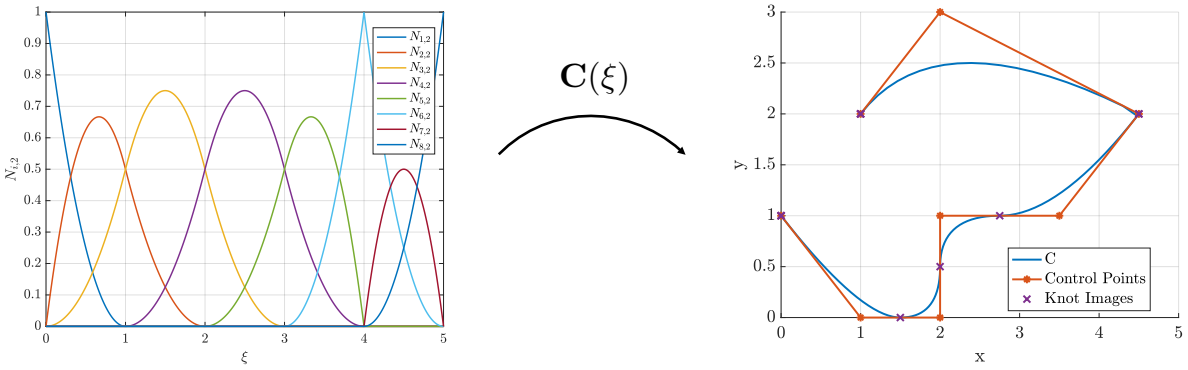

Figure 2.2: The parametric basis functions are pictured on the left. The resulting curve in physical space is shown on the right. The knot images correspond to integer values of ξ as per the definition of Ξ.

in the figure that because of the repetition of knots 0, 4, and 5, the resulting curve is interpolatory at the corresponding knot images.

The discussion here is limited to one-dimensional parametric descriptions. Higher-dimensional basis functions are not used in the body of this paper, but extension to higher dimensions is commonplace in application (see [21] for mathematical details).

2.2.2 NURBS

We are now in a position to define NURBS curves, which are capable of exactly describing a variety of geometries inaccessible with B-splines and the conventional finite element basis, namely conic sections. The relevance of such curves to engineering applications and thus the ability to capture them exactly cannot be understated. NURBS curves are constructed as a B-spline curve in $\mathbb{R}^{d+1}$, then projected onto the unit plane in the $d+1^{\text{th}}$ dimension. We denote the projective space

control points $\{\mathbf{P}_i^w\}$, and the projected control points $\{\mathbf{P}_i\}$. To project these control points into $\mathbb{R}^d$, we simply divide by the $d+1^{\text{th}}$ element, i.e. $(\mathbf{P}_i)_j = (\mathbf{P}_i^w)_j/(\mathbf{P}_i^w)_{d+1}, \forall j \in [1,d]\cap\mathbb{Z}$. The projection of the resulting curve is accounted for in the definition of the NURBS basis functions and is derived analogously. Because the $d+1^{\text{th}}$ elements of the projective control points are quantitatively important but only conceptually important in the derivation, we rebrand these quantities the NURBS 'weights' $w_i \equiv (\mathbf{P}_i^w)_{d+1}$. This process can is visualized in Figure 2.3 below.

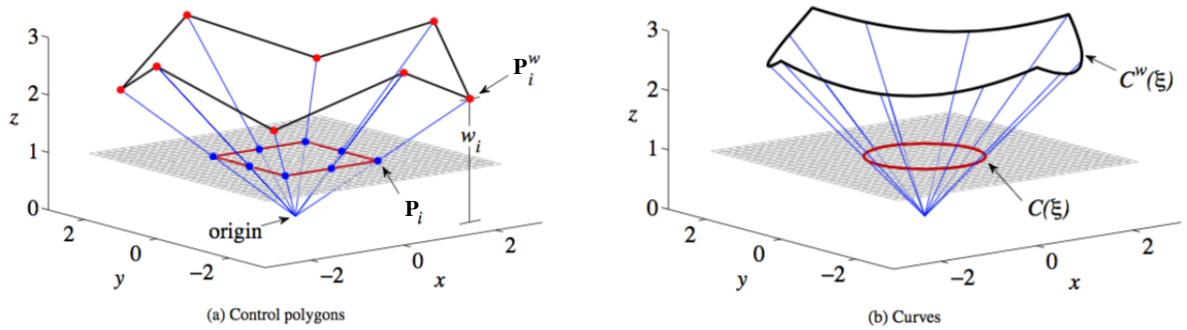

Figure 2.3: In (a) the projective control points $\mathbf{P}_i^w$ are projected onto the plane z=1 to obtain $\mathbf{P}_i$. In (b), the corresponding B-spline curve defined in $\mathbb{R}^3$ by $\mathbf{P}_i^w$ is projected onto z=1 to create the NURBS curve. The projection of the curve itself is accounted for in the basis functions $R_{i,p}$. Image borrowed from [21].

To define a NURBS curve practically we first prescribe a set of weights $w_i \in \mathbb{R}^+$. In one parametric dimension, the index i simply enumerates $[1,\, n]\cap\mathbb{Z}$. We define the i^{th} degree p NURBS basis function

$$R_{i,p}(\xi) = \frac{w_i N_{i,p}(\xi)}{w(\xi)} \tag{2.12}$$

where $w(\xi)$ is the weighting function

$$w(\xi) = \sum_{i=1}^{n} w_i N_{i,p}(\xi) \tag{2.13}$$

In exactly the same manner as before these basis functions can be used to construct a curve $\mathbf{C}$

from control net $\{\mathbf{P}_i\}$.

$$\mathbf{C}(\xi) = \sum_{i=1}^{n} \mathbf{P}_i R_{i,p}(\xi) \tag{2.14}$$

It should be noted that in the case where all NURBS weights are set to 1, the resulting parametrization is exactly equivalent to a B-spline. For comparison, Figure 2.4 below shows a parametrization identical to that shown in Figure 2.2 except that w_7=4. Other weights are set to 1 to imitate B-splines.

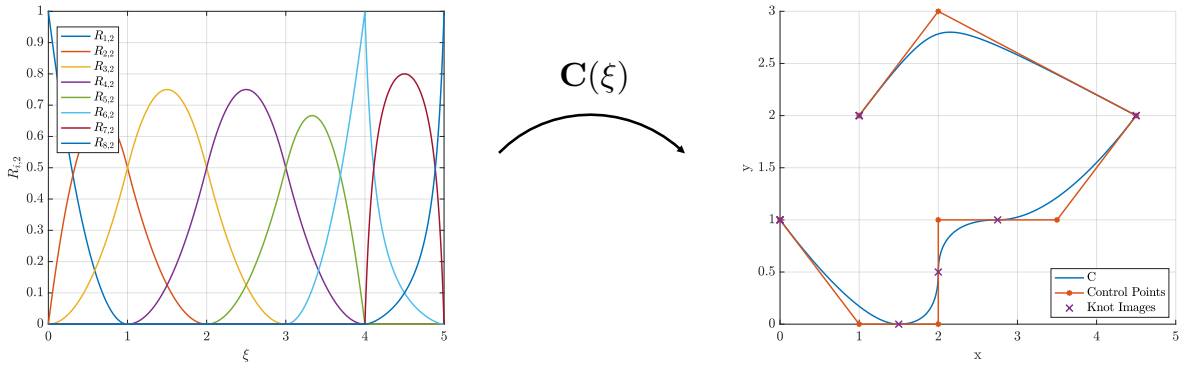

Figure 2.4: The parametric basis functions are pictured on the left. Note that $R_{7,2}$ has been enlarged because of w_7, and how the resulting curve is pulled toward the 7th control point at $(2, 3)$.

2.3 Bézier Extraction and *h*-refinement

It is desireable in the IGA framework to have a notion of h and p refinement, as in traditional finite elements. Such refinement techniques allow for convergence of numerical solutions to the true solution to a given PDE. In IGA, the notion of p-refinement is conceptually straight-forward: given a curve $\mathbf{C}$ represented by parametric basis functions of degree p, one performs manipulations that result in an equivalent representation with degree p+1 basis functions. The machinery of such manipulations is not strictly necessary for implementation of the methods presented in this work and thus will not be derived here. A full treatment can be found in [37].

More relevant to this work is the notion of h-refinement, or the shrinking of geometric pieces

or 'elements' which comprise the solution domain. However, our current description of a parametric curve does not provide an explicit notion of an element. Our discussion here will outline the process for obtaining a set of elements from a general isogeometric parametrization. This process, called Bézier extraction, can be found in full detail in [7].

The principle of Bézier extraction relies on the property that if two sequential knots are both repeated with multiplicity $p+1$ (e.g. $\Xi = [\xi_1 \ \cdots \ 1 \ 1 \ 1 \ 2 \ 2 \ 2 \ \cdots \ \xi_{n+p+1}]$ for p=2), the resulting B-spline basis functions which have support between them (are non-zero) have support nowhere else (are zero everywhere else) and are the only basis functions with support between those knots, effectively isolating that region of parametric space. The resulting basis functions between the repeated knots will take the form of Bernstein polynomials [26]

$$N_{i,p} = B_{i,p}(x) = \binom{i}{p} x^p (1-x)^{i-p} \tag{2.15}$$

and their mapping to physical space is known as a Bézier curve. In the case of NURBS basis functions being used in the original parametrization, the resulting curve is a rational Bézier curve.

To perform Bézier extraction on a given parametric representation, *all* knots are repeated with multiplicity p+1, parametrically isolating each knot span. These knot spans can then be thought of as independent elements. Borden et al. present a procedure in [7] for deriving the new control points and weights that produce an equivalent curve given the new knot vector with the repeated knots. These new 'local' control points and weights are denoted $w^{b,e}$ and $\mathbf{P}^{b,e}$ for corresponding element e.

The transformation between global (original) and local control points for a B-spline curve is achieved by the linear extraction operator C, where

$$\mathbf{P^b} = (\mathbf{W}^b)^{-1} C^T \mathbf{W} \mathbf{P} \tag{2.16}$$

where the $\mathbf{W}$ and $\mathbf{W}^b$ matrices are diagonal matrices containing the global and local weights,

respectively. The local weights are calculated as

$$\mathbf{w}^b = C^T \mathbf{w} \tag{2.17}$$

and we define the local weighting function

$$w^e(\tilde{\xi}) = \sum_{a=1}^{p+1} w_a^{b,e} B_{a,p}(\tilde{\xi}), \qquad \tilde{\xi} \in [0,1] \tag{2.18}$$

Once in element form, the process of knot insertion together with Bézeir extraction can be utilized to create n_{el} arbitrarily small elements. This process serves as $h-$refinement in the isogeometric setting.

To translate between local and global quantities, we construct a connectivity array called the IEN, which maps local indices with element number to a global index

$$IEN(a,e) = i \quad \leftrightarrow \quad \mathbf{P}_a^{b,e} = \mathbf{P}_i. \tag{2.19}$$

In one parametric dimension, construction of the IEN is simply $IEN(a,e) = (l+a)-(p+1)$ when element e exists in the knot span $[\xi_l, \xi_{l+1})$, with $\xi_l \neq \xi_{l+1}$.

Using this connectivity array, we can identify submatrices of C which map a particular subset of $\mathbf{P}_i$ to $\mathbf{P}_a^{b,e}$ for a specific e. We will denote these matrices C^e. The particular subset of C is given by the IEN, where

$$[C^e]_{ab} = [C]_{ij}, \tag{2.20}$$

$$i = IEN(a,e), \quad j = IEN(b,e), \qquad a,b \in [1,\ p+1] \cap \mathbb{Z} \tag{2.21}$$

2.3.1 Line Integration Over a NURBS Curve

Because we will be computing boundary integrals, it is important to have a notion of integration in the isogeometric context. This section describes the process of integrating over a domain

defined by a NURBS object. We limit our discussion here to one-dimensional parametrizations embedded in physical space of dimension $d \geq 1$.

For a given level of refinement, having calculated the local points and weights, it is possible to rewrite the curve $\mathbf{C}$ in terms of its elements

$$\mathbf{C} = \bigcup_{e=1}^{n_{el}} \mathbf{C}^e = \bigcup_{e=1}^{n_{el}} \sum_{i=1}^{p+1} \frac{w_i^{b,e} \mathbf{P}_i^{b,e} B_{i,p}(\tilde{\xi})}{w^e(\tilde{\xi})}, \qquad \tilde{\xi} \in [0,1) \tag{2.22}$$

We will refer to the location of a point on the curve $\mathbf{C}$ as $\mathbf{x}$ with components $[x_1,\ x_2, \cdots, x_d]$. If we now wish to integrate a function $f(\mathbf{x})$ over $\mathbf{C}$, we can simply sum the integrals over each element

$$\int_{\mathbf{C}} f(\mathbf{x}) ds = \sum_{e=1}^{n_{el}} \left(\int_{\mathbf{C}^e} f(\mathbf{x}) ds \right) \tag{2.23}$$

In this local setting we can compute $\mathbf{x}$ in terms of the local points and weights,

$$\mathbf{x}(\tilde{\xi}) = \sum_{i=1}^{p+1} \frac{w_i^{b,e} \mathbf{P}_i^{b,e} B_{i,p}(\tilde{\xi})}{w^e(\tilde{\xi})}, \qquad \tilde{\xi} \in [0,1). \tag{2.24}$$

To compute ds we must compute the derivatives of the mappings from ξ, $\tilde{\xi}$ to $\mathbf{x}$. These mappings are referred to as the Jacobian, $\mathbf{J}$, $\tilde{\mathbf{J}}^e$ and are computed as

$$\mathbf{J} = \left[\frac{dx_1}{d\xi},\ \frac{dx_2}{d\xi}, \cdots, \frac{dx_d}{d\xi} \right], \qquad \tilde{\mathbf{J}}^e = \left[\frac{dx_1}{d\tilde{\xi}},\ \frac{dx_2}{d\tilde{\xi}}, \cdots, \frac{dx_d}{d\tilde{\xi}} \right] \quad \text{on element } e \tag{2.25}$$

where

$$\frac{dx_\alpha}{d\xi} = \sum_{i=1}^{n} (\mathbf{P}_i)_\alpha \frac{dR_i}{d\xi} = \sum_{i=1}^{n} w_i (\mathbf{P}_i)_\alpha \frac{w \frac{dN_i}{d\xi} - N_i \frac{dw}{d\xi}}{w^2}, \qquad \alpha \in \{1,\ 2, \cdots, d\} \tag{2.26}$$

and similarly

$$\frac{dx_\alpha}{d\tilde{\xi}} = \sum_{i=1}^{p+1} w_i^{b,e} (\mathbf{P}_i^{b,e})_\alpha \frac{w^e \frac{dB_i}{d\tilde{\xi}} - B_i \frac{dw^e}{d\tilde{\xi}}}{(w^e)^2}, \qquad \alpha \in \{1,\ 2, \cdots, d\}. \tag{2.27}$$

For the remainder of this thesis we will use Greek subscripts $\{\alpha,\ \beta\ \cdots\}$ to refer to a specific

cartesian coordinate, i.e. the α^{th} direction, and Roman subscripts to indicate the order in which a value appears in a list, e.g. $(\mathbf{P}_k)_\alpha$ would be the component of the k^{th} control point corresponding to the x_α direction. If a value has two subscripts, the order will be $x_{\alpha k}$.

To compute the value of the expressions in 2.23 in parametric space we can equivalently compute

$$\int_{\mathbf{C}} f(\mathbf{x})||\mathbf{J}||d\xi \tag{2.28}$$

or, more usefully,

$$\sum_{e=1}^{n_{el}} \left(\int_0^1 f(\mathbf{x})||\tilde{\mathbf{J}}^e||d\tilde{\xi} \right). \tag{2.29}$$

where $||\cdot||$ denotes the Euclidean norm in $\mathbb{R}^d$. Since the Jacobian points in the direction of the local tangent to $\mathbf{C}$ [47], to find the unit tangent vector to the curve $\mathbf{C}$ we simply normalize the Jacobian

$$\mathbf{t} = \frac{\mathbf{J}}{||\mathbf{J}||} \tag{2.30}$$

with the same holding true for $\tilde{\mathbf{J}}^e$. In the case where $d = 2$ which will be prevalent in the airfoil example, we can compute the outward normal to a simple closed curve in $\mathbb{R}^2$ by rotating the tangent vector through an angle $\pi/2$ in the opposite sense of the curve's parametrization. For example, if the perimeter of an airfoil is parametrized in the clockwise sense, to compute the outward normal we rotate the tangent vector counterclockwise

$$\mathbf{n} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \mathbf{t} = \frac{1}{||\mathbf{J}||} \begin{bmatrix} -(\mathbf{J})_2 \\ (\mathbf{J})_1 \end{bmatrix}. \tag{2.31}$$

Chapter 3

Airfoil Problem

This chapter will discuss the problem of solving for the flow of air around an arbitrary set of airfoils in a low Mach number subsonic flow at high Reynolds number. In Section 3.1, the derivation for the solution method is presented. Then in Section 3.2 numerical considerations are addressed for computational implementation of the method. In Section 3.3, results will be compared to readily available experimental data. For the remainder of this thesis, we will refer to the solution method as IGAirfoil.

3.1 Mathematical Formulation

3.1.1 Continuous Formulation

To determine the aerodynamic properties of an arbitrarily shaped airfoil, it is in general necessary to know the flow properties immediately surrounding the body. In the continuous setting, the equations which describe such a flow field are the well-known Navier-Stokes equations. It is in general impossible to solve these equations because of the unresolved scales present in the full solution. As such, we introduce a number of assumptions – namely that the flow is inviscid, steady, and incompressible. Such assumptions are most readily applicable to the flow outside the boundary layer of an airfoil placed in a flow at Mach 0.3 or lower with sufficiently small angle of attack (generally less than about 10°) [3]. We will investigate an airfoil composed of g distinct closed contours flying at speed V_∞, inclined to the free stream at an angle ν. Under these assumptions, the Navier-Stokes equations to reduce to the following partial differential equation for the velocity

field $V : \mathbb{R}^2 \to \mathbb{R}^2$ [3].

$$\begin{aligned} \nabla \times \mathbf{V}(\mathbf{x}) &= 0, \qquad \mathbf{x} \in \mathbb{R}^2 \setminus \bigcup_{i=1}^{g} \Omega_i \\ \nabla \cdot \mathbf{V}(\mathbf{x}) &= 0, \qquad \mathbf{x} \in \mathbb{R}^2 \setminus \bigcup_{i=1}^{g} \Omega_i \\ \mathbf{V} \cdot \mathbf{n} &= 0, \qquad \mathbf{x} \in \bigcup_{i=1}^{g} \Gamma_i \end{aligned} \tag{3.1}$$

$$\text{and} \quad ||\mathbf{V}|| \to V_\infty \langle \cos\nu, \sin\nu \rangle \quad \text{as} \quad ||\mathbf{x}|| \to \infty$$

where Ω_i are the bodies present in the flow, Γ_i are the corresponding boundaries of said bodies, and $\mathbf{n}$ is the outward normal to a given Γ_i. This PDE states that the velocity field must be free of vorticity, be incompressible, be tangential to the surface of a solid body present in the flow, and appear like a uniform flow as distance from the bodies increases. A diagram showing the regions of interest is shown below in Figure 3.1.

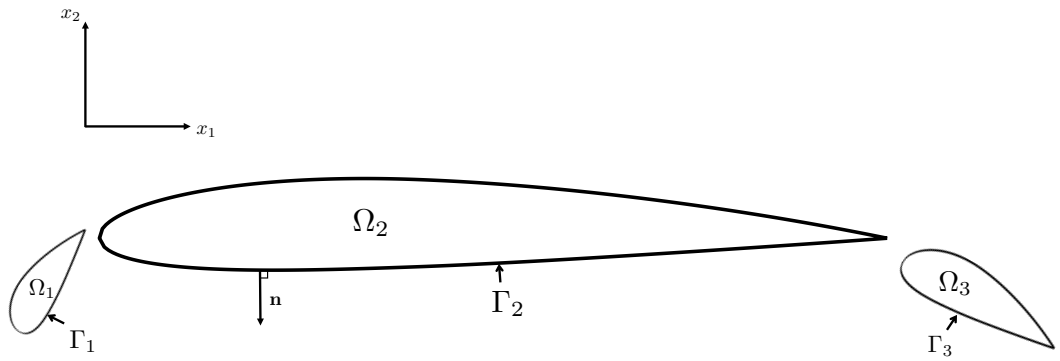

Figure 3.1: The domains of interest are marked on this example airfoil. In this case $g = 3$. This approximately models an airfoil with a leading slat and a trailing flap.

For inviscid, incompressible flows it is possible to decompose the velocity field into constituent components with specific properties. This is accomplished through the Helmholtz-Hodge decomposition [6] as follows

$$\mathbf{V}(\mathbf{x}) = \nabla\hat{\phi}(\mathbf{x}) + \nabla \times \mathbf{w}(\mathbf{x}) + \sum_{i=1}^{g} \beta_i \hat{\mathbf{V}}_i(\mathbf{x}), \qquad \mathbf{x} \in \mathbb{R}^2 \setminus \bigcup_{i=1}^{g} \Omega_i. \tag{3.2}$$

This expression states that the flow outside the bodies can is composed of the gradient of a potential function $\hat{\phi}$, the curl of a vector function $\mathbf{w}(\mathbf{x})$, and the linear combination of g velocity fields known as harmonic forms. The gradient term is identically curl-free, the curl term is identically divergence free, and the final term is the sum of curl and divergence free harmonic forms. In order to satisfy 3.1, we set $\mathbf{w} = 0$ so that $\nabla \times \mathbf{V} = 0$ and proceed to seek the remaining $g + 1$ fields and g linear coefficients from the remaining equations.

Due to the linearity of 3.1, we can solve for $\hat{\phi}$ and the harmonic forms separately, then superimpose their solutions to produce a physical solution. We can transform 3.2 by noting that each of the $\hat{\mathbf{V}}_i$ can be expressed as the gradient of a scalar function by their curl-free property. We define

$$\nabla\hat{\psi}_i = \hat{\mathbf{V}}_i, \qquad i \in [1,\ g] \cap \mathbb{Z} \tag{3.3}$$

and ultimately our task will be to find $\hat{\phi}$ and $\hat{\psi}_i$. Drawing inspiration from the lifting cylinder problem [3], we choose to express both of these potentials as the sum of an ideal flow potential and a disturbance potential. We denote these disturbance potentials ϕ and ψ_i. We construct $\hat{\phi}$ as the sum of ϕ and a uniform flow field with the desired inclination ν and we construct each $\hat{\psi}_i$ as the sum of ψ_i and an ideal vortex (irrotational away from the center) of unit strength placed at a location $\mathbf{c}_i$ inside Ω_i so that vorticity will remain identically 0 in the domain of the solution.

With this formulation, we can express the velocity fields due to $\hat{\phi}$ and $\hat{\psi}_i$ which we will denote $\mathbf{V}_\phi$ and $\mathbf{V}_{\psi,i}$ respectively in terms of the ideal velocity contributions as well as the gradients of

their corresponding disturbances

$$\begin{aligned} \mathbf{V}_\phi &= V_\infty \langle \cos\nu, \sin\nu \rangle + \nabla\phi \\ \mathbf{V}_{\psi,i} &= \beta_i \left(\frac{-1}{2\pi||\mathbf{x}-\mathbf{c}_i||} \hat{\mathbf{e}}_\theta^{c_i} + \nabla\psi_i \right) \end{aligned} \tag{3.4}$$

where $\mathbf{c}_i$ is the vortex center corresponding to body i and $\hat{\mathbf{e}}_\theta^{c_i}$ is the unit angular basis vector of a polar coordinate system originated at $\mathbf{c}_i$, i.e. a vortex is placed at each $\mathbf{c}_i$, rotating in a clockwise sense. Now, since our expressions for $\mathbf{V}$ are identically curl-free we are left with the following g+1 PDEs in terms of ϕ and ψ_i when we substitute each of 3.4 into 3.1.

$$\begin{aligned} \Delta\phi = \Delta\psi_i = 0, \qquad & \mathbf{x} \in \mathbb{R}^2 \setminus \bigcup_{j=1}^{g} \Omega_j, \quad \forall i \in [1,g] \cap \mathbb{Z} \\ \frac{\partial\phi}{\partial\mathbf{n}} = -V_\infty \langle \cos\nu, \sin\nu \rangle \cdot \mathbf{n}, \qquad & \mathbf{x} \in \bigcup_{j=1}^{g} \Gamma_j \\ \frac{\partial\psi_i}{\partial\mathbf{n}} = -\frac{-1}{2\pi||\mathbf{x}-\mathbf{c}_i||} \hat{\mathbf{e}}_\theta^{c_i} \cdot \mathbf{n}, \qquad & \mathbf{x} \in \bigcup_{j=1}^{g} \Gamma_j, \quad \forall i \in [1,g] \cap \mathbb{Z} \\ \text{and} \quad ||\nabla\phi|| \to 0, \quad ||\nabla\psi_i|| \to 0 \quad \text{as} \quad ||\mathbf{x}|| \to \infty \ \forall i \in [1,g] \cap \mathbb{Z} & \end{aligned} \tag{3.5}$$

There are several things we note at this point. First, the β_i scalings have vanished. These will be found later through imposition of the Kutta condition [3]. Next, we note that although we have treated each body's harmonic independently, the boundary conditions are still enforced over the union of *all* Γ_i. In other words, the velocity field composed of the ideal vortex inside a given body and the gradient of its corresponding potential must satisfy the no-penetration condition on every body in the flow. We also note that each potential function solves the Laplace problem on unbounded $\mathbb{R}^2$ and thus have a known Green's function $G(\mathbf{x},\mathbf{y})$. For this type of problem we have $G(\mathbf{x},\mathbf{y}) = \frac{1}{2\pi}\ln||\mathbf{x}-\mathbf{y}||$ [9]. From here on our derivation will examine only ϕ, but our process will apply to each potential function (with the appropriate corresponding boundary condition imposed). It is now convenient to define $\mathbf{V}_{BC} \equiv V_\infty \langle \cos\nu, \sin\nu \rangle \cdot \mathbf{n}$ for notational simplicity[1]. We now apply

[1] Were we to solve for one of the ψ_i, we would simply define $\mathbf{V}_{BC} \equiv \frac{\partial\psi_i}{\partial\mathbf{n}} = -\frac{-1}{2\pi||\mathbf{x}-\mathbf{c}_i||}\hat{\mathbf{e}}_\theta^{c_i} \cdot \mathbf{n}$ and proceed identically. The implications of this will be discussed later.

the identity from integration by parts [30]

$$\int_{\Omega} (w\Delta u - u\Delta w)d\Omega = \int_{\Gamma} \left(w\frac{\partial u}{\partial \mathbf{n}} - u\frac{\partial w}{\partial \mathbf{n}} \right) d\Gamma \tag{3.6}$$

and allow $w = \phi(\mathbf{x})$, $u = G(\mathbf{x}, \mathbf{y})$, noting that $-\Delta G(\mathbf{x}, \mathbf{y}) = \delta(\mathbf{x} - \mathbf{y})$ by definition, where δ is the Dirac delta. Rearranging and applying 3.5, we find

$$\int_{\Omega} \delta(\mathbf{x} - \mathbf{y})\phi(\mathbf{x})d\Omega_{\mathbf{x}} + \int_{\Gamma} \phi(\mathbf{x})\frac{\partial G(\mathbf{x}, \mathbf{y})}{\partial \mathbf{n}_{\mathbf{x}}} d\Gamma_{\mathbf{x}} = -\int_{\Gamma} G(\mathbf{x}, \mathbf{y})\mathbf{V}_{BC} d\Gamma_{\mathbf{x}}. \tag{3.7}$$

Note that the problem is now in terms of the interior of the aerodynamic bodies and their boundary. This is in fact advantageous, as velocity away from the surface of the airfoil has no impact on its aerodynamic properties. Our ultimate goal is to solve the above equation on $\bigcup_{i=1}^{g} \Gamma_i$.

3.1.2 Discrete Formulation

Rather than solve 3.7 analytically (which would be difficult for all but the simplest of geometries) it is advantageous to reduce the dimensionality of the problem by discretizing the solution. This section describes the discretization process. We assume for this process that each of the g bodies is parametrically defined by a closed NURBS curve, each of identical polynomial degree p, a knot vector Ξ_i, with n_i basis functions, for $i \in [1,\ g] \cap \mathbb{Z}$.

Because we are concerned only with the boundary of the airfoil, we wish to eliminate the area integral in 3.7. To do this, we collocate $n = \sum_{i=1}^{g} n_i$ points along the boundary in the $\mathbf{y}$ variable. Thus assuming the boundary of the airfoils are at least C^1-continuous at the collocation points[2], the Dirac delta acting against boundary points is integrated in a single-sided manner and simplifies to the jump term $\phi(\mathbf{y})/2$. Additionally we define $h(\mathbf{x}, \mathbf{y}) \equiv \frac{\partial G(\mathbf{x}, \mathbf{y})}{\partial \mathbf{n}_{\mathbf{x}}}$. We now have n collocated

[2]This assumption is, for reasonable airfoils, applicable everywhere except the trailing edge. We will see later that our collocation scheme will never place a collocation point at the trailing edge for reasonable parametrizations.

equations of the form

$$\frac{1}{2}\phi(\mathbf{y}_i) + \int_\Gamma \phi(\mathbf{x})h(\mathbf{x},\mathbf{y}_i)d\Gamma_\mathbf{x} = -\int_\Gamma G(\mathbf{x},\mathbf{y}_i)\mathbf{V}_{BC}d\Gamma_\mathbf{x}, \qquad i \in \{1,2,...,n\}. \tag{3.8}$$

To reduce the dimensionality of our solution to n as well, we project ϕ onto the isogeometric basis, that is we assert

$$\phi(\mathbf{x}) \approx \phi^h(\mathbf{x}) = \sum_{i=1}^{n} \phi_i R_i(\mathbf{x}(\xi)) \tag{3.9}$$

where $n = \sum_{i=1}^{n} n_i$. Substituting this projection into the previous expression and simplifying, we obtain the final system of equations for the variables ϕ_i.

$$\sum_{j=1}^{n}\left[\frac{1}{2}R_j(\mathbf{y}_i) + \int_\Gamma R_j(\mathbf{x})h(\mathbf{x},\mathbf{y}_i)d\Gamma_\mathbf{x}\right]\phi_j = -\int_\Gamma G(\mathbf{x},\mathbf{y}_i)\mathbf{V}_{BC}(\mathbf{x})d\Gamma_\mathbf{x} \tag{3.10}$$

which can be rewritten in matrix vector form

$$\begin{aligned} K\vec{\phi} &= F \\ [K]_{ij} &= \frac{1}{2}R_j(\mathbf{y}_i) + \int_\Gamma h(\mathbf{x},\mathbf{y}_i)R_j(\mathbf{x})d\Gamma_x \\ [F]_i &= -\int_\Gamma G(\mathbf{x},\mathbf{y}_i)\mathbf{V}_{BC}(\mathbf{x})d\Gamma_\mathbf{x} \end{aligned} \tag{3.11}$$

where $\vec{\phi} = [\phi_1,\ \phi_2, \cdots, \phi_n]^T$ are the control variables corresponding to ϕ^h. We now have a system of n equations and n unknowns, with the i^{th} equation corresponding to the i^{th} collocation point. Thus the rows of K and F can be assembled independently from other rows, however it is important to note that because of the integral terms the matrix system will not be sparse because the integral has access to the entire parametric domain and thus receives the support of every basis function. Another important facet of this solution method is that the computational cost to assemble the stiffness matrix given above depends roughly quadratically on g (quadratic dependence is exact when each body is parametrized with the same number of basis functions), so adding airfoils can

increase computation time significantly.

This procedure can be followed for each potential function, and the observant reader will note that the matrix K depends only on the geometry of the airfoil and not on the boundary condition, and can thus be re-used for each of the g+1 potential functions. For the vortex-based disturbance potentials ψ_i we will adopt the notation ψ_i^j for the i^{th} control variable of the potential corresponding to $\mathbf{c}_j$. With this method we can obtain the values of all of the potential functions on all of Γ_i, and the only remaining task is to find the combination coefficients β_i. These can be determined by the Kutta condition, i.e. enforcing zero velocity at each trailing edge. This yields a system of g equations and g unknowns which will be developed in the next section.

3.2 Numerical Considerations for Implementation

In calculating the values of the integrals in Equation 3.11 there are a number of nuances that must be addressed. They are: handling projective quantities, the selection of the collocation points, selection of an integration scheme, placement of the ideal vortex, computation of singular and discontinuous integrals, computation of near-singular integrals, and post-solution analysis of the results.

3.2.1 Projective Quantities

It was found in implementation that solution accuracy would suffer when using the true value of $\mathbf{V}_{BC}$ in 3.11. This was because the potential functions were being represented by low polynomial degrees (<6) but being forced by functions of effectively infinite polynomial degree, and the resulting curve demonstrated spurious oscillations near areas of high curvature in $\mathbf{V}_{BC}$.

To combat this, the boundary conditions can be projected onto the isogeometric basis via the same collocation scheme as the corresponding potential function. Collocating the boundary

condition at n points $\mathbf{x}_i$ yields a square system for a set of control variables u_j as follows

$$\sum_{j=1}^{n} u_j R_j(\mathbf{x}_i) = \mathbf{V}_{BC}(\mathbf{x}_i), \qquad \mathbf{x}_i \in \Gamma \tag{3.12}$$

Which can be rewritten in matrix vector form

$$\begin{aligned} D\vec{u} &= U \\ [D]_{ij} &= R_j(\mathbf{x}_i) \\ [U]_i &= \mathbf{V}_{BC}(\mathbf{x}_i) \end{aligned} \tag{3.13}$$

where $\vec{u} = [u_1,\ u_2,\ \cdots, u_n]^T$.

Having solved for the coefficients u_j, we can substitute 3.12 into 3.7, and a new matrix system can be derived in terms of these coefficients, namely

$$\begin{aligned} K\vec{\phi} &= \Lambda\vec{u} \\ [\Lambda]_{ij} &= -\int_{\Gamma} G(\mathbf{x}, \mathbf{y}_i) R_j(\mathbf{x}) d\Gamma_x \end{aligned} \tag{3.14}$$

Again, the matrix formed in this process depends only on the geometry of the airfoil and *not* on the boundary conditions. This means that the angle of attack can be changed with relatively little computational effort when the problem is formulated in terms of K and Λ.

3.2.2 Vortex Placement

In the previous section our discussion was left at the mathematical requirement that the i^{th} vortex be placed in the interior of Ω_i. Here we bring further specificity to this statement. Because of the shape of a typical airfoil, the resulting $\mathbf{V}_{BC}$ profiles can have a high curvature depending on the exact location of the vortex inside Ω_i. If the vortex is placed at the center, for example, there is an abrupt change in $\mathbf{V}_{BC}(\xi)$ as $\mathbf{x}(\xi)$ passes the center, because the dot product must flip signs. This results in a boundary condition that is difficult for the NURBS basis to capture in the

projection described above. The best location that was found inside the airfoil is just aft of the leading edge at the center of the osculating circle to the leading edge [1]. This provides the most gradual change of sign, as the circular nature of the ideal vortex blends more smoothly into the nearly circular contour of the leading edge. In general it is possible to calculate the radius of this circle by computing the curvature of the airfoil at its leading edge [47].

3.2.3 Collocation Points

In the section above only the number of collocation points was specified. The choice of the location of the collocation points can greatly affect the accuracy of the solution. Some possible choices for collocation methods are the Greville abscissae, the Demko abscissae, uniformly spaced points, and basis function maxima [4]. The collocation scheme chosen for this method is the Greville abscissae because of their successful application in literature, simplistic computation, and parametric definition.

The Greville abscissae τ_i are defined as the knot averages as shown below.

$$\tau_i = \frac{1}{p} \sum_{j=i+1}^{i+p} \xi_j \tag{3.15}$$

This scheme brings the advantage of sampling more densely in regions of lower continuity since continuity is determined by knot repetition. For a Greville abscissus to be located at a point of low continuity, that point would have to have at least p repeated knots, corresponding to a point of C^0 continuity, or a cusp. For practical applications, points other than the trailing edge of the airfoil will not have a cusp, and therefore have parametrizations with fewer than p knots everywhere other than the first and last knot.

To handle the first and last knot, however, special care must be taken. In this work the foil is parametrized in a clockwise sense, originating and terminating at the trailing edge of the airfoil, with p+1 repeated knots in both locations. To move the first and last abscissae off the trailing edge, the first and last p+2 knots are averaged, respectively.

3.2.4 Integration

Before we discuss the nuances of the integrands in the previously described integrals, we must first have a way of computing integrals. It is unreasonable to compute analytic solutions to all the integrals for arbitrary geometries. Therefore we adopt a numerical scheme with which to approximate these integral quantities. Like collocation, there are many schemes available. The method selected for this work is Gaussian quadrature [8]. We elect to use at minimum $p+1$ quadrature points in the hopes of capturing the most information for the least computational expense, given that polynomials of degree p can be captured exactly with $p+1$ Gaussian quadrature points [8]. Because all elements are integrated over $0 \leq \tilde{\xi} \leq 1$, the quadrature points and weights can be computed only once and reused.

3.2.5 Integrating Through Discontinuities

In order to compute the integrals in 3.11, some elements require special treatment because of the properties of h and G. Namely, when $\mathbf{x} = \mathbf{y}_i$, h is discontinuous and G is weakly singular. To compute the integrals in K, it is sufficient to simply split the element at $\mathbf{y}_i$ and integrate each side separately. To compute integrals in Λ requires a bit more care. To accomplish this, the Telles transformation [50] is used. This strategy employs a change of variables via a weighting function whose Jacobian vanishes at the point of the singularity and renders the integrand continuous and therefore integrable.

Suppose the function $f(x)$ is weakly singular at some $x = c \in [a, b]$ but we wish to compute $\int_a^b f(x)dx$. The Telles transformation employs a cubic change of variables, $\gamma(t) = At^3 + Bt^2 + Ct + D$ under the constraints $\gamma(a) = a$, $\gamma(b) = b$, and $\gamma'(t_0) = 0$, where t_0 corresponds to the singularity. A fourth constraint is necessary to determine all the coefficients, so $\gamma''(t_0) = 0$ is chosen. Using an even polynomial degree would violate bijectivity and render the change of variables useless, so the extra coefficient is necessary. These constraints serve two purposes. First, to hold the bounds of integration constant under transformation, and second, to negate the effect of the singularity at c

via the jacobian term that appears due to the mapping from γ to x.

Substituting this change of variables into the integral we find

$$\int_a^b f(\gamma)d\gamma = \int_a^b f(\gamma(t))\gamma'(t)dt \tag{3.16}$$

Rather than solve the nonlinear system of equations which results from ensuring that $\gamma(t_0) = c$, it is convenient to simply split the integral at the singularity and perform two separate transformations.

Suppose on collocation point y_i lives on element e and our goal is to compute

$$\int_0^1 R_j(\mathbf{x}(\tilde{\xi}))G(\mathbf{x}(\tilde{\xi}), \mathbf{y}_i)||\tilde{\mathbf{J}}^e||d\tilde{\xi}. \tag{3.17}$$

If y_i is located at $\xi^* \in [0,\ 1]$ then we compute the left and right integrals

$$\int_0^{\xi^*} R_j(\mathbf{x}(\tilde{\xi}))G(\mathbf{x}(\tilde{\xi}), \mathbf{y}_i)||\tilde{\mathbf{J}}^e||d\tilde{\xi} + \int_{\xi^*}^1 R_j(\mathbf{x}(\tilde{\xi}))G(\mathbf{x}(\tilde{\xi}), \mathbf{y}_i)||\tilde{\mathbf{J}}^e||d\tilde{\xi}. \tag{3.18}$$

Applying the Telles transformation to the left integral, we require that $\gamma(0) = 0$, $\gamma(\xi^*) = \xi^*$, $\gamma'(\xi^*) = \gamma''(\xi^*) = 0$. This yields the linear system of equations

$$\begin{bmatrix} (\xi^*)^3 & (\xi^*)^2 & \xi^* & 1 \\ 3(\xi^*)^2 & 2\xi^* & 1 & 0 \\ 6\xi^* & 2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} = \begin{bmatrix} \xi^* \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.19}$$

The right integral is handled similarly, with $\gamma(1) = 1$ replacing $\gamma(0) = 0$. This alters the last row of 3.19 in a straight-forward way.

When this method is applicable, the quadrature scheme is modified through two linear transformations mapping $[0,\ 1]$ to $[0,\ \xi^*]$ and $[\xi^*,\ 1]$ for each integral.

3.2.6 Integrating Near Singularities

Elements near the singularities discussed in the previous section also pose a challenge in integration. While the integrand over an element may be finite, the integral is not well-approximated by quadrature rules because quadrature is designed to integrate polynomials rather than exponentials (or logarithms). In this case adaptive refinement is necessary. On an element where a singularity is too close, the element is split until the element is sufficiently small to provide a smooth integrand which can be captured with quadrature. Quantitatively, we declare that refinement is needed on an element e when $d_y < 2\gamma$, where d_y is the distance from e to the singular point and γ is the square root of the ratio of the length of e to the perimeter of the airfoil [49]. To compute d_y, a sub-element is approximated as a straight line locus between its end-points, and d_y can be computed as the minimum distance over the locus. This refinement is applied recursively until all elements satisfy the inequality.

3.2.7 Post-Solution Analysis

From a practical standpoint it is not useful to know only the potential function along the surface of an airfoil. We desire the velocity profile along the surface at minimum, and ultimately wish to know the bulk aerodynamic properties of the airfoil. In this section we will explore the process of deriving useful information once the potential functions have been calculated.

By superposition we can compute the total potential Φ as the sum $\Phi = \hat{\phi} + \sum_{i=1}^{g} \beta_i \hat{\psi}_i$, where β_i must be determined by the imposition of the Kutta condition. The Kutta condition states that the velocity at the trailing edge of each body in the flow must be zero. Thus if TE_j represents the trailing edge location of the j^{th} body we have g equations of the form

$$\sum_{i=1}^{g} \beta_i (\nabla \hat{\psi} \cdot \mathbf{t}) \mid_{TE_j} = -(\nabla \hat{\phi} \cdot \mathbf{t}) \mid_{TE_j}, \qquad j \in [1,\, g] \cap \mathbb{Z} \tag{3.20}$$

and can solve for the g unknown β_i values.

Because the numerator of the ideal vortex potential was left as unity, the quantities β_i are

now direct measures of vorticity, i.e. $\oint_{C_c} \nabla \times \mathbf{V} = \beta$ where C_c is any closed contour which encloses only $\mathbf{c}_i$ and no other vortex centers. Then the total vorticity produced by all g bodies is simply the sum $\beta = \sum_{i=1}^{g} \beta_i$. Via the Kutta-Jukowski theorem of lift it can be shown that the lift per unit span $L' = \rho_\infty V_\infty \beta$ [3]. We will discuss the computation of β shortly after we have shown how to compute the flow velocity.

In addition to lift, we can also calculate the pressure contour along the surface of the airfoil. The nondimensional pressure coefficient C_P is defined

$$C_P = \frac{p - p_\infty}{\frac{1}{2}\rho_\infty V_\infty^2} \tag{3.21}$$

but in the steady, inviscid setting where Bernoulli's equation applies, this formula can be reduced to

$$C_P = 1 - \left(\frac{V}{V_\infty}\right)^2. \tag{3.22}$$

It is now useful to define

$$\tilde{\mathbf{V}}_u = V_\infty \langle \cos\nu, \sin\nu \rangle \cdot \mathbf{t} \tag{3.23}$$

and

$$\tilde{\mathbf{V}}_{v,i} = -\frac{1}{2\pi||\mathbf{x} - \mathbf{c}_i||} \hat{\mathbf{e}}_\theta^{c,i} \cdot \mathbf{t} \tag{3.24}$$

and perform a collocation-based projection of these quantities. However, in this case we will project onto a basis of degree $p-1$ rather than p as before, as we intend to include these quantities in equations with the gradient of potential which is a lower polynomial degree than the potential itself by virtue of the derivative. The lower order basis functions will be denoted $\bar{R}_i$ for $i \in \{1,\ 2,\ \cdots, n-1\}$. The projection operation is analogous to 3.13 with a new set of $n-1$ collocation

points.

$$\begin{aligned} \tilde{D}\vec{\tilde{u}} &= \tilde{U} \\ [\tilde{D}]_{ij} &= \frac{1}{||\mathbf{J}||}\bar{R}_j(\mathbf{x}_i) \\ [\tilde{U}]_i &= \tilde{\mathbf{V}}_u(\mathbf{x}_i) \end{aligned} \tag{3.25}$$

and for ψ_i

$$\begin{aligned} \tilde{D}\vec{\tilde{v}} &= \tilde{V} \\ [\tilde{V}]_j &= \tilde{\mathbf{V}}_{v,i}(\mathbf{x}_j) \end{aligned} \tag{3.26}$$

where $\vec{\tilde{u}} = [\tilde{u}_1,\ \tilde{u}_2,\ \cdots, \tilde{u}_{n-1}]^T$ and $\vec{\tilde{v}} = [\tilde{v}_1,\ \tilde{v}_2,\ \cdots, \tilde{v}_{n-1}]^T$.

We will adopt the same subscript-superscript notation for $\tilde{v}$ as for ψ. To find the velocity at the surface of the airfoil, we compute the directional derivative of the total potential along the tangent to Γ, recalling that our no-penetration boundary conditions forbid velocity in any direction other than the local tangent. Thus the directional derivative is the gradient.

$$\begin{aligned} V &= \nabla\Phi(\mathbf{x}) = \nabla\Phi(\mathbf{x})\cdot\mathbf{t} \\ &= \left(\nabla\phi(\mathbf{x}) + \sum_{i=1}^{g}\beta_i\nabla\psi_i(\mathbf{x})\right)\cdot\mathbf{t} + \left(V_\infty\langle\cos\nu,\sin\nu\rangle - \sum_{i=1}^{g}\frac{\beta_i}{2\pi||\mathbf{x}-\mathbf{c}_i||}\hat{\mathbf{e}}_\theta^{c,i}\right)\cdot\mathbf{t} \\ &\approx \nabla\left(\sum_{i=1}^{n}\left(\phi_i + \sum_{j=1}^{g}\beta_j\psi_i^j\right)R_i(\mathbf{x})\right)\cdot\mathbf{t} + \sum_{i=1}^{n-1}\frac{1}{||\mathbf{J}||}\left(\tilde{u}_i + \sum_{j=1}^{g}\beta_j\tilde{v}_i^j\right)\bar{R}_i(\mathbf{x}) \\ &= \sum_{i=1}^{n}\left(\phi_i + \sum_{j=1}^{g}\beta_j\psi_i^j\right)(\nabla R_i(\mathbf{x})\cdot\mathbf{t}) + \sum_{i=1}^{n-1}\frac{1}{||\mathbf{J}||}\left(\tilde{u}_i + \sum_{j=1}^{g}\beta_j\tilde{v}_i^j\right)\bar{R}_i(\mathbf{x}) \\ &= \sum_{i=1}^{n}\left(\phi_i + \sum_{j=1}^{g}\beta_j\psi_i^j\right)\frac{dR_i(\mathbf{x})}{d\Gamma} + \sum_{i=1}^{n-1}\frac{1}{||\mathbf{J}||}\left(\tilde{u}_i + \sum_{j=1}^{g}\beta_j\tilde{v}_i^j\right)\bar{R}_i(\mathbf{x}) \\ &= \left(\sum_{i=1}^{n}\left(\phi_i + \sum_{j=1}^{g}\beta_j\psi_i^j\right)\frac{dR_i(\mathbf{x})}{d\xi} + \sum_{i=1}^{n-1}\left(\tilde{u}_i + \sum_{j=1}^{g}\beta_j\tilde{v}_i^j\right)\bar{R}_i(\mathbf{x})\right)\frac{1}{||\mathbf{J}||} \end{aligned} \tag{3.27}$$

Using the expressions derived here we can also compute the value of β. Because of the

discontinuity at the trailing edge, the tangent vector is undefined. Thus rather than satisfy 3.20 directly, we instead impose the condition

$$\frac{1}{2}\left(\mathbf{V}|_{TE^-} + \mathbf{V}|_{TE^+}\right) = 0 \tag{3.28}$$

on each airfoil where we define

$$\begin{aligned} TE_i^+ &\equiv \sum_{j=1}^{n_i} \mathbf{P}_j R_{j,p}(\xi_1 + \varepsilon) \\ TE_i^- &\equiv \sum_{j=1}^{n_i} \mathbf{P}_j R_{j,p}(\xi_{n+p+1} - \varepsilon) \end{aligned} \tag{3.29}$$

for the j^{th} body and its corresponding parametrization, and require $\varepsilon \ll \xi_{n+p+1} - \xi_1$. Numerical results presented in this thesis utilize a value of $\epsilon = 10^{-5}$ and $\xi_{n+p+1} - \xi_1 = 2$. Because we expect the velocities along the surface above and below the trailing edge to continuously approach 0 when the value of β is correct, solving 3.28 asymptotically approaches the correct value of β as ε becomes increasingly small.

To compute β we expand 3.28 at each airfoil to obtain the following system of linear equations which can be solved to obtain all values of β_i.

$$\begin{aligned} A\vec{\beta} &= B \\ [A]_{ij} &= \sum_{k=1}^{n} \psi_k^j \left(\left.\frac{dR_k(\mathbf{x})}{d\xi}\right|_{TE_i^+} + \left.\frac{dR_k(\mathbf{x})}{d\xi}\right|_{TE_i^-} \right) + \sum_{k=1}^{n-1} \tilde{v}_k^j \left(\bar{R}_k(\mathbf{x})\big|_{TE_i^+} + \bar{R}_k(\mathbf{x})\big|_{TE_i^-} \right) \\ [\vec{\beta}]_i &= \beta_i \\ [B]_i &= -\sum_{k=1}^{n} \phi_k \left(\left.\frac{dR_k(\mathbf{x})}{d\xi}\right|_{TE_i^+} + \left.\frac{dR_k(\mathbf{x})}{d\xi}\right|_{TE_i^-} \right) - \sum_{k=1}^{n-1} \tilde{u}_k \left(\bar{R}_k(\mathbf{x})\big|_{TE_i^+} + \bar{R}_k(\mathbf{x})\big|_{TE_i^-} \right) \end{aligned} \tag{3.30}$$

We noted earlier that β is a measure of lift. We have not discussed drag, however. This is because in the current formulation, it is impossible to predict drag. Because we assumed that the flow was inviscid ($\nabla \times \mathbf{V} = 0$), it can be shown [3] that the drag force acting on any body in the

flow is identically zero. We address this shortcoming in Chapter 5.

3.3 Results

In this section numerical results from the IGAirfoil will be presented. Solutions for various test cases will be compared to analytical solutions, wind tunnel data, and lower-order methods.

3.3.1 Convergence

In the special case of of cylindrical geometry with a radius of 1, an analytic solution to 3.1 is available [3], namely

$$\mathbf{V}(r,\theta)|_{\Gamma} = 2V_{\infty}\sin\theta. \tag{3.31}$$

We can compare the solutions generated by IGAirfoil to this known analytic solution for a sequence of parametrizations generated through the h-refinement process, each with smaller elements and more degrees of freedom than the previous. This is achieved through the process of uniform knot insertion, where knots are inserted at the average value of existing sequential unique knot values. It is important to note here that even for the simplest of test cases it has already become necessary to represent conic sections exactly.

A plot of the resulting L^2 norm of the solution error is shown below in Figure 3.2

It is clear from Figure 3.2 that optimal convergence rates have not been achieved. However, obvious patterns are present. As polynomial degree increases, the asymptotic convergence rate remains the same. The pre-asymptotic behavior of the solutions, however, appears to align with the optimal rates before trailing off. This consistency implies that an approximation error is slowing convergence rather than the solution method itself. We observe in Figure 3.3 where only the boundary condition is compared to the exact solution that the achieved convergence rates are in fact optimal.

This plot shows that when a pure projection operation is performed, the collocation and refinement schemes are able to capture the correct behavior numerically. Since the projected

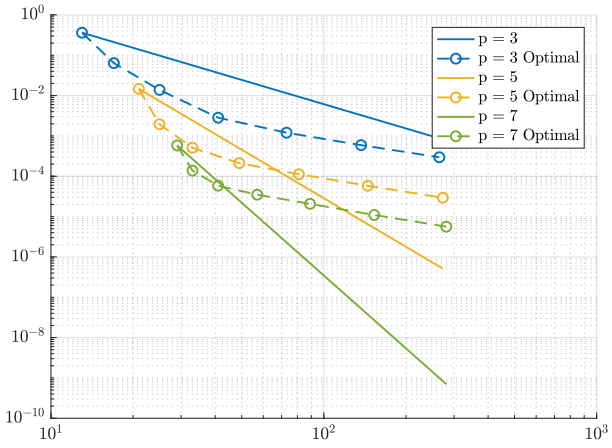

Figure 3.2: Convergence of the velocity profile on the surface of the cylinder

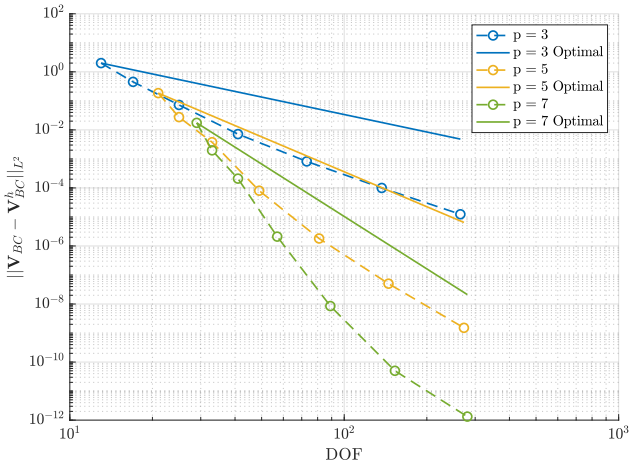

Figure 3.3: Convergence of the projected boundary condition

boundary condition and the disturbance potential share a collocation scheme, we can infer that the problem must lie elsewhere. It is possible that the Gaussian quadrature scheme fails to approximate the integrals in 3.11 under refinement. An investigation into more accurate quadrature schemes is the subject of ongoing work, discussed in Chapter 5.

3.3.2 Comparison to Existing Data

It is helpful to know that in addition to providing converging solutions for the simple problem of flow around a cylinder. However, from a practical design perspective it is important to know that IGAirfoil can predict the properties of actual aerodynamic bodies as well. We desire for the solver to be able to predict not only the net lifting force, but also determine the pressure contour at the surface of the foil. Below in Figure 3.4 we compare IGAirfoil to theoretical and experimental results for a NACA 0012 airfoil at an angle of attack of 0°, parametrized with 32 boundary elements.

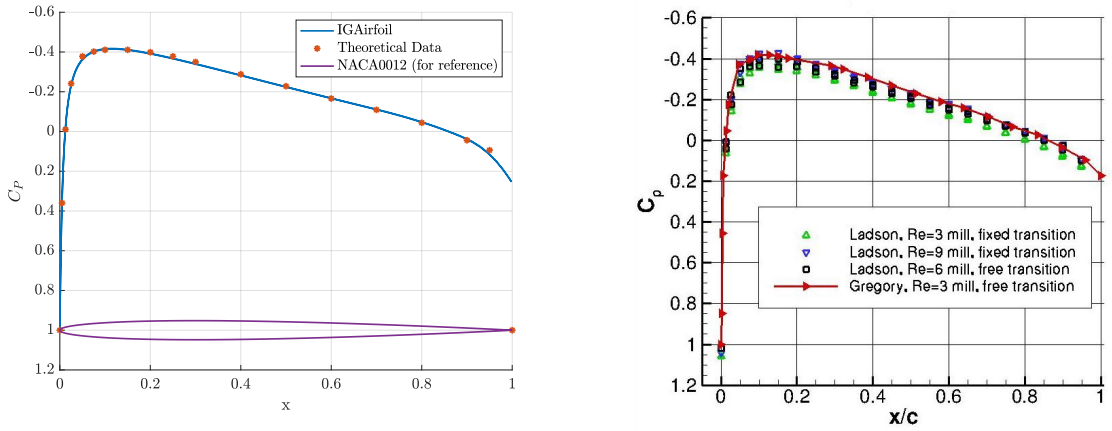

Figure 3.4: On the left, the IGAirfoil results are plotted against theoretical data from [1] and agreement is good. On the right, experimental data from [39] are shown.

Overall there is excellent agreement between all models. It should be noted that the axis scales on both plots are identical, and visual comparison reveals that to within the differences between the experiments the IGAirfoil result matches the data. Next we will compare similar quantities for an angle of attack of 10°, again for a 32-element parametrization. The results are shown in Figure 3.5.

Discrepancy is noticeable here. Because IGAirfoil is a potential model, flow can experience arbitrary pressure gradients without adverse effects. In this case, both experimental data and Reynolds-Averaged Navier-Stokes (RANS) [14] data using the Spalart-Allmaras (SA) turbulence model [46] predict a higher pressure on the upper side of the leading edge. This is the location

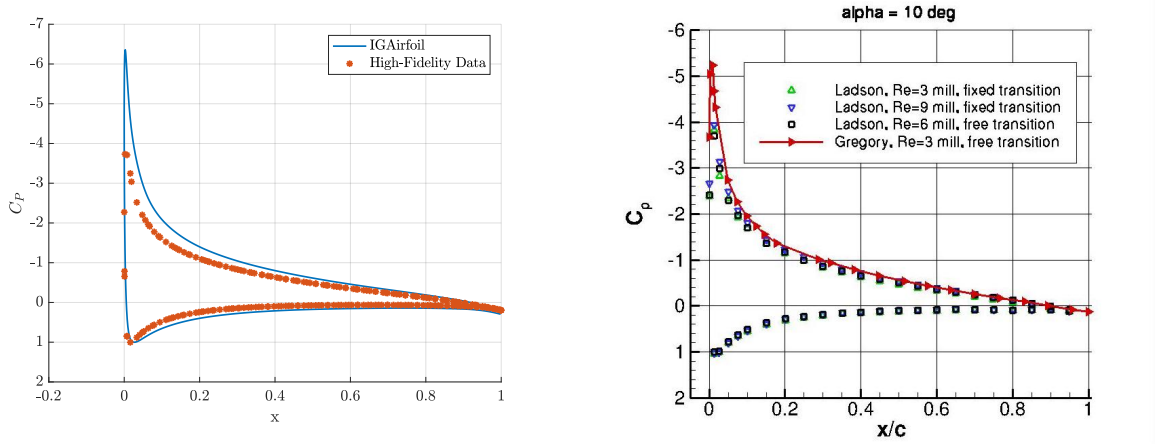

Figure 3.5: On the left, the IGAirfoil results are plotted against RANS data generated in PHASTA. On the right, experimental data from [39] are shown.

where pressure gradients are greatest, and thus dissipative effects present in both experiment and SA prohibit the flow from achieving the 'ideal' pressure contour given by the pure potential model. Since the lift per unit span of an airfoil is given by the area inside the C_P curve [3], we note that IGAirfoil will over-predict lift in this case. The tendency to over-predict is an inherent flaw with potential models because of the unphysical response to adverse pressure gradients.

3.3.3 Comparison with Low-Fidelity Methods

In addition to comparison with high-fidelity models and experiment which provide us with a measure of confidence in the performance of IGAirfoil, it is also beneficial to compare the method with alternative low-fidelity methods. In this case, we will examine the results of the source panel and vortex panel methods [1].

These two methods are similar. They each approximate an airfoil as a series of connected line segments and use fundamental flows to construct a flow field. In the source panel method, each line segment is designated as a line source, and in the vortex panel method each segment is a vortex line. The strengths of these lines can vary linearly, but must be continuous. Therefore to describe the strength along the airfoil the strengths at the intersections are all that is needed. To calculate these strengths, a no-penetration condition is set at the midpoint of each panel and the

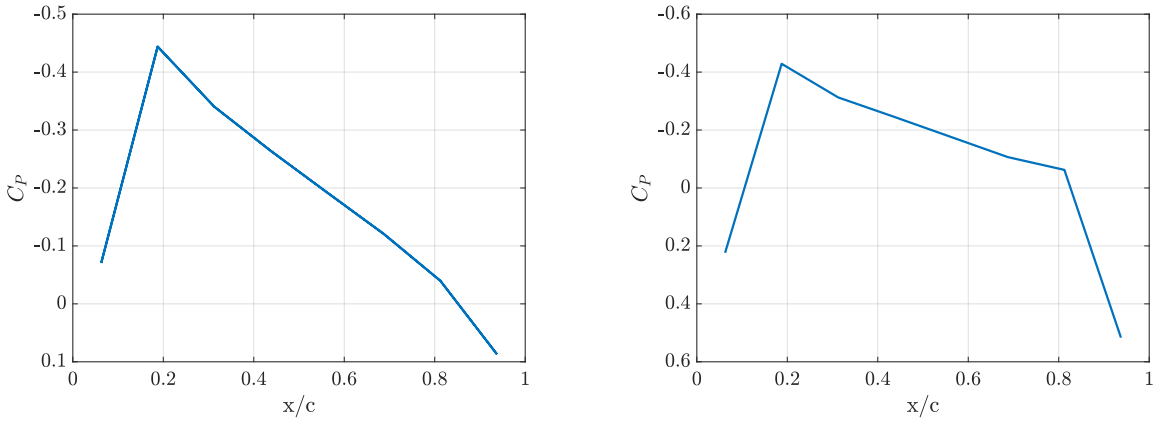

Figure 3.6: The results for a 16-DOF source panel method are shown on the left and the results for a 16-DOF vortex panel method are shown on the right.

Kutta condition is enforced, which produces a linear system of equations.

Using the same 32-element parametrization of the IGA airfoil, the resulting number of degrees of freedom in the global system is 16. The results for an equal number of degrees of freedom in both the source panel and vortex panel methods is shown below in Figure 3.6

The difference between Figures 3.6 and 3.4 is striking. Per degree of freedom, IGAirfoil exhibits a drastically improved result over conventional panel methods. It is clear that these panel methods fall short in this regard, but more importantly for our purposes, they are not geometrically informed. Geometry is crudely approximated and while the solution does depend on geometry, analysis and geometry are not analytically tied and thus cannot be used for shape optimization. IGAirfoil overcomes both these shortcomings as a preliminary design tool.

Chapter 4

Airfoil Shape Optimization

Having described a novel method for the computation of the lifting force produced by a set of aerodynamic bodies, we now turn our attention to shape optimization. In general, our goal is to minimize an objective function $\mathcal{F}$. This objective function will depend on the geometry of the problem, which is encapsulated in some set of design variables $\mathbf{d}$. It will also depend on a set of auxiliary variables $\mathbf{u}(\mathbf{d})$ which depend on the design variables. In the case of IGAirfoil these auxiliary variables are the solution to the PDE and the projected quantities which are relative to the specific choice of $\mathcal{F}$. We wish to minimize $\mathcal{F}$ over the space of design variables.

Given the capability of IGAirfoil, the most logical choice of $\mathcal{F}$ is a function that targets a specific value of lift, L_0. To place the minimum of $\mathcal{F}$ at L_0, we can simply construct the function

$$\mathcal{F}(\mathbf{d}, \mathbf{u}) = (L(\mathbf{d}, \mathbf{u}) - L_0)^2. \tag{4.1}$$

so that standard minimization techniques can be used. More complex objective functions may be utilized given expanded solver capability. This will be addressed in Chapter 5.

To limit the dimensionality of the problem, the following design parameters are to be utilized in this surrogate optimization problem. First, it should be noted that this optimization problem will fix a value of $g = 2$ (with the exception of the first and last cases) and will optimize an airfoil with a trailing flap. Both foils will be NACA 4-digit series foils [1], and thus their geometry can be described completely by 3 numbers, m_i, p_i, and t_i, for $i \in \{1, 2\}$. The NACA 4-digit series is a

family of airfoil shapes whose profile is generated via a formula which uses the digits in a specific way. Generally speaking, these numbers provide the amount of camber, the along-chord location of the maximum thickness, and the maximum thickness of the airfoil respectively. To generate an airfoil from the 4-digit series, the following formulae are used [1].

$$y_t = 5t(0.2969\sqrt{x} - 0.1260x - 0.3516x^2 + 0.2843x^3 - 0.1036x^4) \tag{4.2}$$

$$y_c = \begin{cases} \frac{m}{p^2}\left(2p\frac{x}{c} - \left(\frac{x}{c}\right)^2\right), & 0 \le x \le pc \\ \frac{m}{(1-p)^2}\left((1-2p) + 2p\frac{x}{c} - \left(\frac{x}{c}\right)^2\right), & pc \le x \le c \end{cases} \tag{4.3}$$

where y_c is the mean camber line and y_t is the airfoil thickness centered at y_c. The quantities m, p, and t are defined such that $100m$ is the first of the 4 digits, $10p$ is the second, and $100t$ are the final two.

In our optimization it is assumed that the orientation of the primary body is not to change with respect to the free stream because we could equivalently change the angle of attack, so we can simply fix the orientation of this body. The location and orientation of the flap may remain variables. To describe the location of the flap, we define d to be the distance between the trailing edge of the main body and the leading edge of the flap, and we define δ to be the angle between horizontal and the line between these edges. The orientation of the flap is described by its inclination from horizontal γ. The chord length of both bodies is allowed to vary, and are denoted c_1 and c_2 for the main body and the flap, respectively. The diagram below depicts these variables graphically.

The resulting optimization depends on a total of 11 design variables (recall that the four NACA digits are representative of only three independent numbers), including three NACA digits and a chord length for each body as well as the distance separating the bodies and the two orientation angles of the flap. To parametrize the problem in the IGA setting, an L^2 projection (see Appendix B for a brief derivation of this method) is performed to project the exact NACA foil geometry onto a B-spline basis of degree 4 to within a tolerance of 10^{-5} as measured by the L^2

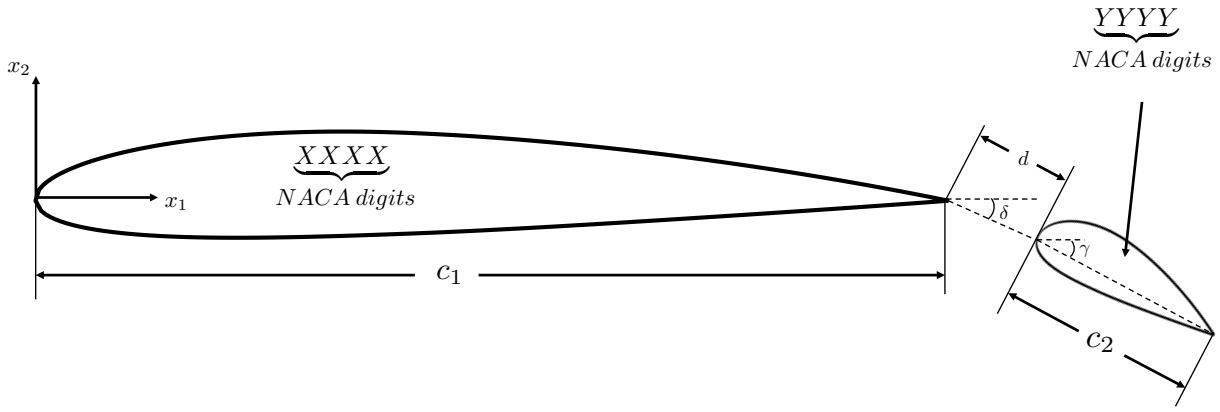

Figure 4.1: The optimization design variables. Angles are measured positive clockwise.

norm. This is process is performed independently for each airfoil. Given the tolerance specified for the projection, the resulting fidelity of the parametrization (and thus the number of basis functions) will vary between the two bodies.

It is apparent that there are many ways to arrive at a given value of lift with all 11 design variables at one's disposal. Thus we will examine a variety of sub-problems in which we constrain the design space and target a value of lift. In these sub-problems an interior-point minimization routine [31] is used with a step tolerance of 10^{-14} and a first-order optimality tolerance [33] of 10^{-3}. These operations are captured by the MATLAB function `fmincon`.These tolerance values were selected based on the level of refinement used in the optimization procedure so that numerical errors would not stall the minimization routine. In application, physical tolerances and design requirement flowdown would dictate the necessary values for these tolerances and thus the necessary level of refinement used. To produce the results given in this section, two uniform knot insertions were performed on airfoils after their initial construction. The minimization was given constant upper and lower bounds on each of the design variables. These bounds were chosen heuristically on a case-by-case basis and were selected such that solutions would be physically relevant from a design perspective. Bounds worthy of note are explicitly justified.

The first problem we will examine is the simplest. Our objective is to optimize the shape of

only the main body until a lift of 0 is produced. We fix the chord length $c_1 = 1$ and the angle of attack $\nu = 0$ and allow the NACA digits to vary until the lift produced by the foil is satisfactorily low. The initial airfoil profile is a NACA 4412. The optimization variables are given below in 4.1 and the resulting profiles are shown in Figure 4.2.

Design Variable	Initial Value	Lower Bound	Upper Bound	Final Value
m_1	0.04	0	0.1	0.0022995
p_1	0.4	0.1	0.9	0.42063
t_1	0.12	0.03	0.20	0.13374

Table 4.1: Optimization values for targeting L=0 with a single foil.

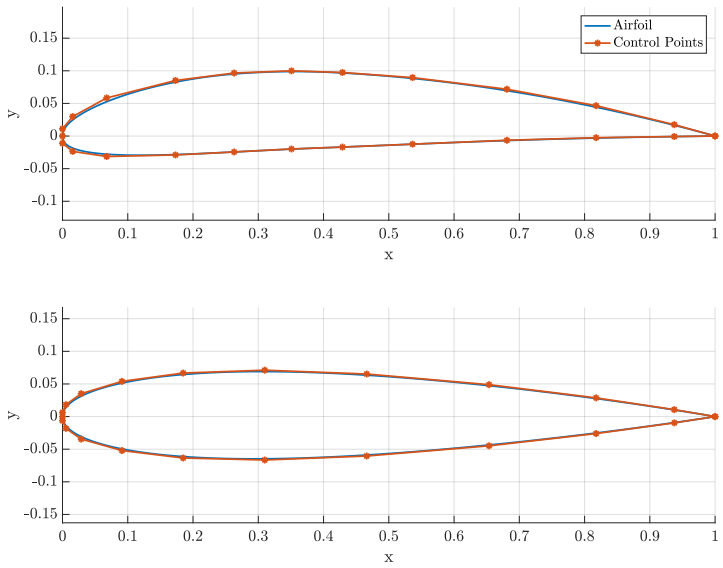

Figure 4.2: The initial airfoil is pictured at top with the optimized design below. The axes share a scale and which is equal in the x and y directions.

The constraints for m_1 and t_1 given in Table 4.1 were chosen heuristically to provide a reasonable design space. The constraints on p_1 were chosen because of an exponential increase in the number of degrees of freedom necessary to capture the geometry of designs where p_1 is either very close to 0 or 1. Limiting p_1 to within 0.1 of these values maintains a reasonable design space while avoiding a trade-off between accuracy and computational expense.

Visually, the optimization performed as expected. We anticipate a symmetric airfoil producing 0 lift in a uniform flow. The final values for the NACA digits decrease by an order of magnitude. Using a value of $V_\infty = 25$ and $\rho = 1.225$, the lift force produced in the original configuration is $L = 150.42$, whereas in the final configuration the lift force is on the order 10^{-6}, a drastic reduction. The main quantity of interest is m_1 which determines how much camber is present in the design. A value of 0 would indicate a perfectly symmetric airfoil. When $m_1 = 0$ the value of p_1 is irrelevant. The value of m_1 decreased by an order of magnitude, converging to a value representing a 0.2%-chord camber profile.

Our next example will try to place a trailing flap in the optimal location. We now fix the digits of both bodies such that both airfoils are NACA 0012 profiles. We fix both chord lengths to $c_1 = 1$, $c_2 = 0.25$, and we fix $d = 0.1$ to mimic a hinged connection between the bodies. The optimization values are given in Table 4.2 and visualized in Figure 4.3 below.

Design Variable	Initial Value	Lower Bound	Upper Bound	Final Value
δ	20°	0°	90°	0.29699°
γ	20°	0°	90°	0.0083202°

Table 4.2: Optimization values for targeting L=0 with a single foil.

In this optimization problem we expect that the flap align itself with the main body so that both foils produce zero lift. Visually this is what occurred. It is notable from Table 4.2 that the inclination of the foil itself was driven to 0°by two orders of magnitude more than its relative orientation to the main body. This is because the lift generated by the flap depends much more heavily on its individual angle of attack than it does on the orientation of the two bodies. While it is true that two symmetric bodies each aligned with the incoming flow can experience a lift force because of the interaction between the two bodies, in this case that effect is much less prevalent than the role of γ. In the final configuration, the flap is translated a distance of 0.000524 from the x axis which is small compared to the other distances in the problem. In the original configuration, the bodies produced a lift force of 498.18267, and in the final configuration the lift force produced was 0.00083073.

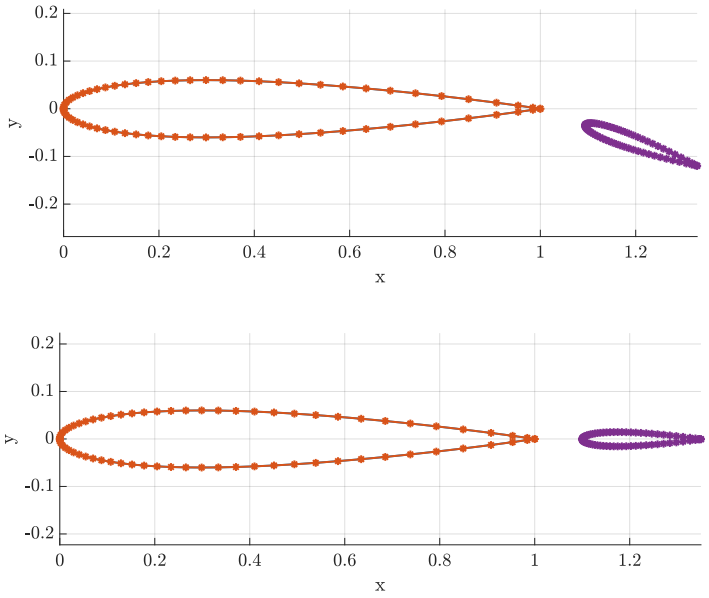

Figure 4.3: In the top panel is the original configuration with the flap inclined to the flow and off the x axis. On the bottom is the final configuration with the flap aligned.

For our final test case, we wish to optimize the body profiles of both airfoils in a flow at an angle of attack $\nu = -7.5°$with both bodies aligned with the x axis. By targeting a lift force of 0 this mimics the process of obtaining a desired zero-lift angle of attack of -7.5°. In this case we will hold $c_1 = 1$, $c_2 = 0.25$, $d = 1$, $\delta = 0°$and γ=0°fixed. The optimization variables are shown below in Table 4.3 and visualized in Figure 4.4.

This trial also behaved more or less as expected. Both airfoils were given a fairly aggressive camber profile close to their respective upper bounds. Both foils were given a relatively slim profile, which is to be expected since at the earliest steps of the minimization process, decreasing the thickness would have produced less lift in the negative direction, so initially both thicknesses would decrease. As we noted earlier, there tend to be multiple solutions to these problems. If a different thickness were desired, a new optimization process could be formulated where the objective function is some cost function in terms of the thickness, and the old cost function $\mathcal{F}$ could be applied as a nonlinear constraint in an attempt to traverse the contour in the design space where the original

Design Variable	Initial Value	Lower Bound	Upper Bound	Final Value
m_1	0	0	0.10	0.099756
p_1	0.5	0.2	0.6	0.59667
t_1	0.12	0.10	0.30	0.105185
m_2	0	0	0.05	0.049688
p_2	0.5	0.2	0.6	0.59763
t_2	0.12	0.05	0.20	0.052212

Table 4.3: Optimization values for targeting L=0 with two foils in a fixed orientation at a negative angle of attack. The bounds for p_1 and p_2 have been shrunk to produce more reasonable designs.

cost function is satisfied. Originally, a lift of -429.95 was produced, and after optimization the lift force had a magnitude of -0.0005083. The reader should note that the initial values of p_1 and p_2 were set to 0.5 because when the first digit is 0, this value has no effect, but as soon as optimization steps are taken the constraints would be violated if these values were 0 as well.

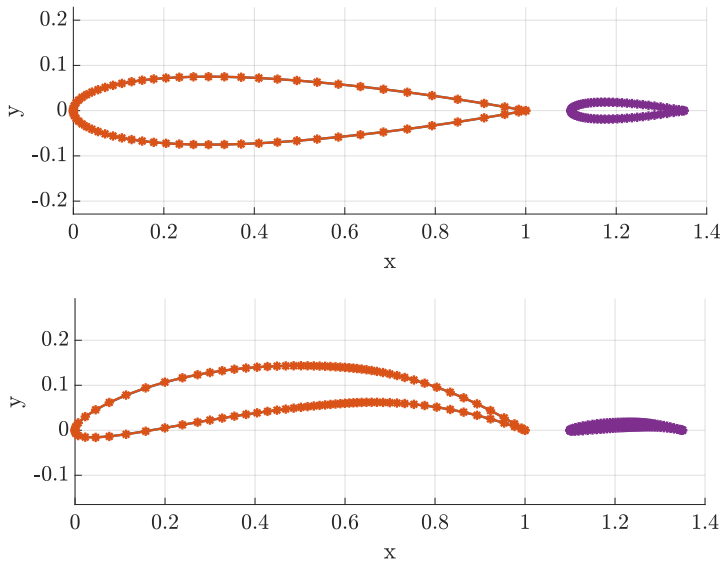

Figure 4.4: The top panel shows the initial body and flap both symmetric and aligned with the x axis. In the bottom panel they have been endowed with a camber profile capable of negating the negative angle of attack.

In the final optimization problem, we aim to find an airfoil geometry at zero angle of attack capable of matching the lift produced by a symmetric airfoil at a positive angle of attack. We

will elect to match the lift produced by a NACA 0015 airfoil inclined at 5°angle of attack, which corresponds to an approximate value of lift of 230. The parameters and results are shown below in Table 4.4 and Figure 4.5. The initial design is a symmetric airfoil. Note again that since we expect a nonzero final value of m_1, the initial value of p_1 is set to 0.5 to allow for a natural physical progression upon minimization iteration.

Design Variable	Initial Value	Lower Bound	Upper Bound	Final Value
m_1	0	0	0.20	0.065217
p_1	0.5	0.2	0.6	0.30247
t_1	0.12	0.03	0.30	0.068945

Table 4.4: Optimization values for targeting L=230 with a single foil at 0 angle of attack.

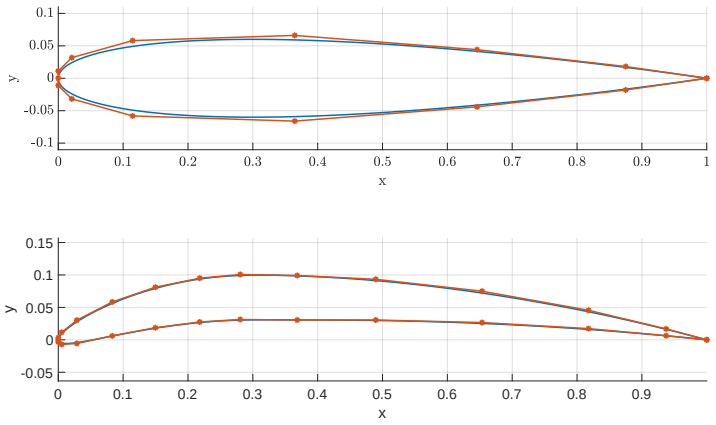

Figure 4.5: The top panel shows the initial symmetric body. In the bottom panel the airfoil has been given a cambered profile so that it produces the desired value of lift at 0 angle of attack.

In this problem, the resulting airfoil is close to a NACA 6306 profile. The optimization process has given it an aggressive camber and a maximum camber near the front of the airfoil. These are two easy ways to produce a large amount of lift. In a more sophisticated routine, there are a number of ways that these could be constrained beyond simple upper and lower bounds. This is true of each of the optimization problems we have described. In general, an approach similar to [27] can be taken, where the optimization process is posed in terms of an auxiliary objective

function and the actual objective function is posed as a constraint. This would be useful once an initial solution is found like the ones above. If the auxiliary function is used to target certain characteristics, for example the minimum camber capable of producing the desired lift, the result is a traversal of viable solutions within the design space. Similarly the optimization process can be informed by further modeling capability such as a viscosity model which would forbid solutions that produced unsuitably strong adverse pressure gradients. This is discussed further in Chapter 5.

4.1 Analytical Gradients

In previous sections we have stated the advantages which IGA presents in the shape optimization context. In this section we will formalize the notion of analytic shape sensitivities and present the technique for a gradient-based shape optimization. Referring again to our generic cost function $\mathcal{F}(\mathbf{d}, \mathbf{u}(\mathbf{d}))$, we wish to compute the gradient with respect to the design variables $\vec{d}$. To accomplish this, we simply invoke the chain rule.

$$\nabla_{\mathbf{d}}\mathcal{F} = \bar{\nabla}_{\mathbf{d}}\mathcal{F} + \nabla_{\mathbf{u}}\mathcal{F}\frac{\partial \mathbf{u}}{\partial \mathbf{d}} \tag{4.4}$$

Here we have introduced a symbol $\bar{\nabla}$ to take the place of the partials with respect to design variables which one calculates directly, as opposed to the partials denoted by ∇ which represent the "total" derivatives with respect to design variables. Since there are multiple design variables, these are technically partial derivatives as well, so to distinguish between these two types of partials we adopt the $\bar{\nabla}$ notation.

For many optimization functions, such as lift, the gradients of $\mathcal{F}$ are simple to compute. The calculation of the sensitivities of the auxiliary variables with respect to geometry are the main concern. To calculate this quantity, we examine the relationship between the variables that is known. In general, this is the discretized system which approximates PDE under study, and we

will represent it with a generic residual equation.

$$R(\mathbf{d}, \mathbf{u}) = \mathbf{0}. \tag{4.5}$$

Taking the partials of 4.5 in a similar fashion to $\mathcal{F}$, we find

$$\nabla_{\mathbf{d}} R = \bar{\nabla}_{\mathbf{d}} R + \nabla_{\mathbf{u}} R \frac{\partial \mathbf{u}}{\partial \mathbf{d}}. \tag{4.6}$$

In this case, however, the left hand side is 0 identically and thus we can rearrange to find an expression for the partial

$$\frac{\partial \mathbf{u}}{\partial \mathbf{d}} = -(\nabla_{\mathbf{u}} R)^{-1} \bar{\nabla}_{\mathbf{d}} R. \tag{4.7}$$

Since R is a known function of $\mathbf{u}$ and $\mathbf{d}$, this quantity can be calculated analytically. This has been completed, but for readability the tedium of partial derivatives has been relegated to Appendix A for reference. The reader should note that the derivatives in Appendix A are given for a set of design variables $\mathbf{d}$ simply equal to the control points. This set of design variables is the most fundamental and can be utilized no matter what design variables are chosen. In general, if an arbitrary set $\mathbf{d}'$ is selected, to apply the above equations it is necessary only to compute $\partial \mathbf{d}/\partial \mathbf{d}'$ which varies on a case-specific basis. In the case of the optimization problem stated above, calculation of this quantity would necessarily be performed through a finite difference scheme [8] since NACA profiles are not analytically related to B-splines.

Chapter 5

Concluding Remarks

5.1 Summary of Completed Work

In this thesis, we have motivated the need for fast solvers applicable to the preliminary design process. The need for efficient and accurate low-fidelity models stems from a desire to focus resources on detailed design. Current computational methods like FEA are not motivated by geometry and cause a bottleneck in the design-analysis process. However, isogeometric analysis offers an elegant solution to this issue. IGA provides a direct connection between geometry and analysis and in doing so not only circumvents this bottleneck but also allows for the calculation of analytic sensitivities which aid the optimization process. Low-fidelity geometric optimization routines give the possibility of eliminating the design-analysis cycle altogether.

In this thesis we have derived a novel potential-based method for computing velocity and pressure distributions around groups aerodynamic bodies of arbitrary shape. We have demonstrated the accuracy of the method. Finally, we have provided initial steps toward a robust shape optimization framework built upon the solution method by demonstrating its applicability on surrogate optimization problems and providing the analytic shape sensitivities for gradient-based optimization. Lastly we have discussed avenues for improvement on the work that has been completed and suggested avenues for new, related research topics.

5.2 Ongoing and Future Work

Based on the results in Figure 3.2, a critical area of work is improving convergence rates. To accomplish this a wider breadth of literature on quadrature schemes and integration techniques for boundary elements is being taken into consideration. Alternative methods to the refinement scheme in use currently exist [22, 53] and await implementation in IGAirfoil. Utilization of an alternative technique for integration may also greatly decrease computation time, as the adaptive refinement scheme accounts for the majority of computation time currently.

One of the other major thrusts of ongoing work on this project is to incorporate drag prediction capability into IGAirfoil. This goal is mathematically impossible with a pure potential flow model [3]. Rather than re-imagine the model, however, it is more useful to couple the existing solve to a viscous boundary layer model such as the one presented in [13]. This is in fact the model used by the aforementioned XFOIL. To predict the characteristics of a boundary layer using this formulation, it is necessary to enforce as a boundary condition the flow properties outside the boundary condition. As we stated earlier, the true role of potential models is to predict flow outside the boundary layer. Thus the viscosity model is informed by the potential model, and the potential model is reliant on the shape given by the boundary layer thickness calculated by the viscosity model. The viscosity model produces a boundary layer whose boundary condition is the velocity predicted by the potential model, and the boundary layer thickness influences the geometry imposed on the potential model. To ensure agreement between the models, a Newton-Raphson residual-based root-finding method [8] can be used until the solutions agree. Implementation of this model is ongoing.

Finally, extension of the methods presented here to more complex or difficult problems is a viable avenue for further work. An isogeometric optimization problem for the path of a satellite performing a finite burn in the presence of a massive body has been posed but no results are yet available. This problem does not involve BEM. A viable extension of BEM-based isogeometric optimization lies in electromagnetics and acoustics. To this author's knowledge, there is no published

material investigating this topic. Material exists exploring isogeometric BEM for electromagnetics and acoustics such as [36, 45, 44] as well as isogeometric shape optimization of such problems [32], but synthesis between the two remains an open area for investigation.

Bibliography

[1] Ira Herbert Abbott and Albert Edward Von Doenhoff. Theory of wing sections, including a summary of airfoil data. Courier Corporation, 1959.

[2] Mark J Ablowitz and Athanassios S Fokas. Complex variables: introduction and applications. Cambridge University Press, 2003.

[3] John David Anderson Jr. Fundamentals of aerodynamics. Tata McGraw-Hill Education, 2010.

[4] F Auricchio, L Beirao Da Veiga, TJR Hughes, A_ Reali, and G Sangalli. Isogeometric collocation methods. Mathematical Models and Methods in Applied Sciences, 20(11):2075–2107, 2010.

[5] Francesco Bassi and Stefano Rebay. A high-order accurate discontinuous finite element method for the numerical solution of the compressible navier–stokes equations. Journal of computational physics, 131(2):267–279, 1997.

[6] Peter S Bernard. Fluid Dynamics. Cambridge University Press, 2015.

[7] Michael J Borden, Michael A Scott, John A Evans, and Thomas JR Hughes. Isogeometric finite element data structures based on bézier extraction of nurbs. International Journal for Numerical Methods in Engineering, 87(1-5):15–47, 2011.

[8] R Burden and J Faires. Numerical analysis 9th edn (boston: Brooks/cole). 2011.

[9] AH-D Cheng, Heinz Antes, and Norbert Ortner. Fundamental solutions of products of helmholtz and polyharmonic operators. Engineering analysis with boundary elements, 14(2):187–191, 1994.

[10] Seonho Cho and Seung-Hyun Ha. Isogeometric shape design optimization: exact geometry and enhanced sensitivity. Structural and Multidisciplinary Optimization, 38(1):53, 2009.

[11] J Austin Cottrell, Thomas JR Hughes, and Yuri Bazilevs. Isogeometric analysis: toward integration of CAD and FEA. John Wiley & Sons, 2009.

[12] Mark Drela. Xfoil: An analysis and design system for low reynolds number airfoils. In Low Reynolds number aerodynamics, pages 1–12. Springer, 1989.

[13] Mark Drela and Michael B Giles. Viscous-inviscid analysis of transonic and low reynolds number airfoils. AIAA journal, 25(10):1347–1355, 1987.

[14] Paul A Durbin and BA Pettersson Reif. Statistical theory and modeling for turbulent flows. John Wiley & Sons, 2011.

[15] John A. Evans. Lecture notes on isogeometric analysis, 2014.

[16] E Gillebaart and R De Breuker. Low-fidelity 2d isogeometric aeroelastic analysis and optimization method with application to a morphing airfoil. Computer Methods in Applied Mechanics and Engineering, 305:512–536, 2016.

[17] Ralf Hartmann and Paul Houston. Adaptive discontinuous galerkin finite element methods for the compressible euler equations. Journal of Computational Physics, 183(2):508 – 532, 2002.

[18] Luca Heltai, Marino Arroyo, and Antonio DeSimone. Nonsingular isogeometric boundary element method for stokes flows in 3d. Computer Methods in Applied Mechanics and Engineering, 268:514–539, 2014.

[19] Ming-Chen Hsu, Ido Akkerman, and Yuri Bazilevs. High-performance computing of wind turbine aerodynamics using isogeometric analysis. Computers & Fluids, 49(1):93–100, 2011.

[20] Thomas J. R. Hughes. The finite element method : linear static and dynamic finite element analysis. New York : Dover publication, Inc, 2007.

[21] Thomas JR Hughes, John A Cottrell, and Yuri Bazilevs. Isogeometric analysis: Cad, finite elements, nurbs, exact geometry and mesh refinement. Computer methods in applied mechanics and engineering, 194(39-41):4135–4195, 2005.

[22] Peter R Johnston. Application of sigmoidal transformations to weakly singular and near-singular boundary element integrals. International journal for numerical methods in engineering, 45(10):1333–1348, 1999.

[23] KV Kostas, AI Ginnis, CG Politis, and PD Kaklis. Ship-hull shape optimization with a t-spline based bem–isogeometric solver. Computer Methods in Applied Mechanics and Engineering, 284:611–622, 2015.

[24] Arnold M Kuethe and CY Chow. Foundations of aerodynamics-bases of aerodynamic design, 1998.

[25] Kang Li and Xiaoping Qian. Isogeometric analysis and shape optimization via boundary integral. Computer-Aided Design, 43(11):1427–1437, 2011.

[26] George G Lorentz. Bernstein polynomials. American Mathematical Soc., 2012.

[27] Nguyen Dang Manh, Anton Evgrafov, Allan Roulund Gersborg, and Jens Gravesen. Isogeometric shape optimization of vibrating membranes. Computer Methods in Applied Mechanics and Engineering, 200(13-16):1343–1353, 2011.

[28] Andrea Manzoni, Filippo Salmoiraghi, and Luca Heltai. Reduced basis isogeometric methods (rb-iga) for the real-time simulation of potential flows about parametrized naca airfoils. Computer Methods in Applied Mechanics and Engineering, 284:1147–1180, 2015.

[29] Benjamin Marussig, Jürgen Zechner, Gernot Beer, and Thomas-Peter Fries. Fast isogeometric boundary element method based on independent field approximation. Computer Methods in Applied Mechanics and Engineering, 284:458–488, 2015.

[30] William Charles Hector McLean. Strongly elliptic systems and boundary integral equations. Cambridge university press, 2000.

[31] Arkadi S Nemirovski and Michael J Todd. Interior-point methods for optimization. Acta Numerica, 17:191–234, 2008.

[32] Dang Manh Nguyen, Anton Evgrafov, and Jens Gravesen. Isogeometric shape optimization for electromagnetic scattering problems. Progress In Electromagnetics Research, 45:117–146, 2012.

[33] Jorge Nocedal and Stephen J Wright. Numerical optimization 2nd, 2006.

[34] Knut Nordanger, Runar Holdahl, Arne Morten Kvarving, Adil Rasheed, and Trond Kvamsdal. Implementation and comparison of three isogeometric navier–stokes solvers applied to simulation of flow past a fixed 2d naca0012 airfoil at high reynolds number. Computer Methods in Applied Mechanics and Engineering, 284:664–688, 2015.

[35] Peter Nørtoft and Jens Gravesen. Isogeometric shape optimization in fluid mechanics. Structural and Multidisciplinary Optimization, 48(5):909–925, 2013.

[36] Michael J Peake, Jon Trevelyan, and Graham Coates. Extended isogeometric boundary element method (xibem) for two-dimensional helmholtz problems. Computer Methods in Applied Mechanics and Engineering, 259:93–102, 2013.

[37] Les Piegl and Wayne Tiller. The NURBS book. Springer Science & Business Media, 2012.

[38] Xiaoping Qian. Full analytical sensitivities in nurbs based isogeometric shape optimization. Computer Methods in Applied Mechanics and Engineering, 199(29-32):2059–2071, 2010.

[39] Christopher Rumsey. 2dn00: 2d naca 0012 airfoil validation case, 2018.

[40] Stefan A Sauter and Christoph Schwab. Boundary Element Methods, volume 39 of Springer Series in Computational Mathematics. Springer Berlin, 2010.

[41] Michael A Scott, Robert N Simpson, John A Evans, Scott Lipton, Stephane PA Bordas, Thomas JR Hughes, and Thomas W Sederberg. Isogeometric boundary element analysis using unstructured t-splines. Computer Methods in Applied Mechanics and Engineering, 254:197–221, 2013.

[42] RN Simpson, SPA Bordas, Haojie Lian, and J Trevelyan. An isogeometric boundary element method for elastostatic analysis: 2d implementation aspects. Computers & Structures, 118:2–12, 2013.

[43] Robert N Simpson, Stéphane PA Bordas, Jon Trevelyan, and Timon Rabczuk. A two-dimensional isogeometric boundary element method for elastostatic analysis. Computer Methods in Applied Mechanics and Engineering, 209:87–100, 2012.

[44] Robert N Simpson, Zhaowei Liu, Ráfael Vazquez, and John A Evans. An isogeometric boundary element method for electromagnetic scattering with compatible b-spline discretizations. arXiv preprint arXiv:1704.07128, 2017.

[45] Robert N Simpson, Michael A Scott, Matthias Taus, Derek C Thomas, and Haojie Lian. Acoustic isogeometric boundary element analysis. Computer Methods in Applied Mechanics and Engineering, 269:265–290, 2014.

[46] PRaA Spalart and S1 Allmaras. A one-equation turbulence model for aerodynamic flows. In 30th aerospace sciences meeting and exhibit, page 439, 1992.

[47] James Stewart. Essential calculus. Cengage Learning, 2010.

[48] Toru Takahashi and Toshiro Matsumoto. An application of fast multipole method to isogeometric boundary element method for laplace equation in two dimensions. Engineering Analysis with Boundary Elements, 36(12):1766–1775, 2012.

[49] Matthias F. Taus. Isogeometric Analysis for Boundary Integral Equations. PhD thesis, University of Texas at Austin, 2015.

[50] J. C. F. Telles. A self-adaptive co-ordinate transformation for efficient numerical evaluation of general boundary element integrals. International Journal for Numerical Methods in Engineering, 24(5):959–973.

[51] Jacobus JW van der Vegt and H Van der Ven. Space–time discontinuous galerkin finite element method with dynamic grid motion for inviscid compressible flows: I. general formulation. Journal of Computational Physics, 182(2):546–585, 2002.

[52] Wolfgang A Wall, Moritz A Frenzel, and Christian Cyron. Isogeometric structural shape optimization. Computer methods in applied mechanics and engineering, 197(33-40):2976–2988, 2008.

[53] Wenjing Ye. A new transformation technique for evaluating nearly singular integrals. Computational Mechanics, 42(3):457, Apr 2008.

Appendix A

Shape Derivatives

One of the distinct advantages of IGA is the opportunity for analytic shape sensitivities to aid the optimization process. Having developed the equations for the potential functions, we can identify our design and auxiliary variables for optimization. The design variables are simply the components of the geometric control variables, as this is the most general description of geometry. We define $x_{1k} = (\mathbf{P}_k)_{x_1}$ and $x_{2k} = (\mathbf{P}_k)_{x_2}$ and construct $\mathbf{d}$ as follows.

$$\mathbf{d} = [x_{11}\ x_{12}\ \cdots\ x_{1n}\ x_{21}\ x_{22}\ \cdots\ x_{2n}]^T \tag{A.1}$$

To reduce the complexity of the notation, we will assume there is only one body in the flow. This is not the most general form, but the derivatives that appear are identical to within indexing differences. Now to define $\mathbf{u}$ we assemble all potential function and projection control variables as well as $\beta_1 = \beta$. We will extend the notation used in 3.12 such that $\vec{u}$ is the set of control variables for the uniform flow boundary condition, and $\vec{v}$ is the set of control variables for the vortex flow boundary condition, however we will denote the right hand side V rather than U to distinguish between boundary conditions.

$$\mathbf{u} = [\vec{\phi}^T\ \vec{u}^T\ \vec{\psi}^T\ \vec{v}^T\ \vec{\tilde{u}}^T\ \vec{\tilde{u}}^T\ \beta]^T \tag{A.2}$$

From our derivations, the first $6n-2$ residual equations for $\mathbf{u}$ satisfy

$$\begin{bmatrix} K & -\Lambda & & & & & 0 \\ 0 & D & & & & & \\ & & K & -\Lambda & & & \\ & & 0 & D & & & \\ & & & & \tilde{D} & 0 & \\ 0 & & & & 0 & \tilde{D} & \end{bmatrix} \begin{bmatrix} \vec{\phi} \\ \vec{u} \\ \vec{\psi} \\ \vec{v} \\ \vec{\tilde{u}} \\ \vec{\tilde{v}} \end{bmatrix} - \begin{bmatrix} 0 \\ U \\ 0 \\ V \\ \tilde{U} \\ \tilde{V} \end{bmatrix} = 0. \tag{A.3}$$

The final equation comes directly from 3.30.

$$R_\beta = -\frac{\sum_{i=1}^{n} \phi_i \left(\frac{dR_i}{d\xi}\Big|_{TE^-} + \frac{dR_i}{d\xi}\Big|_{TE^+} \right) + \sum_{i=1}^{n-1} \tilde{u}_i \left(\bar{R}_i|_{TE^-} + \bar{R}_i|_{TE^+} \right)}{\sum_{i=1}^{n} \psi_i \left(\frac{dR_i}{d\xi}\Big|_{TE^-} + \frac{dR_i}{d\xi}\Big|_{TE^+} \right) + \sum_{i=1}^{n-1} \tilde{v}_i \left(\bar{R}_i|_{TE^-} + \bar{R}_i|_{TE^+} \right)} - \beta = 0 \tag{A.4}$$

Taking partials with respect to the auxiliary variables we find

$$\begin{aligned}
\frac{\partial R_\beta}{\partial \phi_k} &= -\frac{\frac{dR_k}{d\Gamma}\Big|_{TE^-} + \frac{dR_k}{d\xi}\Big|_{TE^+}}{\sum_{i=1}^{n} \psi_i \left(\frac{dR_i}{d\xi}\Big|_{TE^-} + \frac{dR_i}{d\xi}\Big|_{TE^+} \right) + \sum_{i=1}^{n-1} \tilde{v}_i \left(\bar{R}_i|_{TE^-} + \bar{R}_i|_{TE^+} \right)} \\
\frac{\partial R_\beta}{\partial \tilde{u}_k} &= -\frac{\bar{R}_k|_{TE^-} + \bar{R}_k|_{TE^+}}{\sum_{i=1}^{n} \psi_i \left(\frac{dR_i}{d\xi}\Big|_{TE^-} + \frac{dR_i}{d\xi}\Big|_{TE^+} \right) + \sum_{i=1}^{n-1} \tilde{v}_i \left(\bar{R}_i|_{TE^-} + \bar{R}_i|_{TE^+} \right)} \\
\frac{\partial R_\beta}{\partial \psi_k} &= \frac{\sum_{i=1}^{n} \phi_i \left(\frac{dR_i}{d\xi}\Big|_{TE^-} + \frac{dR_i}{d\xi}\Big|_{TE^+} \right) + \sum_{i=1}^{n-1} \tilde{u}_i \left(\bar{R}_i|_{TE^-} + \bar{R}_i|_{TE^+} \right)}{\left[\sum_{i=1}^{n} \psi_i \left(\frac{dR_i}{d\xi}\Big|_{TE^-} + \frac{dR_i}{d\xi}\Big|_{TE^+} \right) + \sum_{i=1}^{n-1} \tilde{v}_i \left(\bar{R}_i|_{TE^-} + \bar{R}_i|_{TE^+} \right) \right]^2} \left(\frac{dR_k}{d\xi}\Big|_{TE^-} + \frac{dR_k}{d\xi}\Big|_{TE^+} \right) \\
\frac{\partial R_\beta}{\partial \tilde{v}_k} &= \frac{\sum_{i=1}^{n} \phi_i \left(\frac{dR_i}{d\xi}\Big|_{TE^-} + \frac{dR_i}{d\xi}\Big|_{TE^+} \right) + \sum_{i=1}^{n-1} \tilde{u}_i \left(\bar{R}_i|_{TE^-} + \bar{R}_i|_{TE^+} \right)}{\left[\sum_{i=1}^{n} \psi_i \left(\frac{dR_i}{d\xi}\Big|_{TE^-} + \frac{dR_i}{d\xi}\Big|_{TE^+} \right) + \sum_{i=1}^{n-1} \tilde{v}_i \left(\bar{R}_i|_{TE^-} + \bar{R}_i|_{TE^+} \right) \right]^2} \left(\bar{R}|_{TE^-} + \bar{R}|_{TE^+} \right) \\
\frac{\partial R_\beta}{\partial \beta} &= -1.
\end{aligned} \tag{A.5}$$

It is now useful to define two quantities $\mathbb{K}$ and $\mathbb{F}$.

$$\mathbb{K} = \begin{bmatrix} K & -\Lambda & & & & & 0 \\ 0 & D & & & & & \\ & & K & -\Lambda & & & \\ & & 0 & D & & & \\ & & & & \tilde{D} & 0 & \\ 0 & & & & 0 & \tilde{D} & \\ \frac{\partial R_\beta}{\partial \vec{\phi}}^T & 0 & \frac{\partial R_\beta}{\partial \vec{\psi}}^T & 0 & \frac{\partial R_\beta}{\partial \vec{u}}^T & \frac{\partial R_\beta}{\partial \vec{v}}^T & \frac{\partial R_\beta}{\partial \beta} \end{bmatrix}, \qquad \mathbb{F} = \begin{bmatrix} 0 \\ U \\ 0 \\ V \\ \tilde{U} \\ \tilde{V} \\ 0 \end{bmatrix} \tag{A.6}$$

We now define our residual equation $R(\mathbf{d}, \mathbf{u})$ as the concatenation of A.3 with R_β and we immediately note the following

$$\begin{aligned} \nabla_{\mathbf{u}} R &= \mathbb{K} \\ -\bar{\nabla}_{\mathbf{d}} R &= \frac{\partial \mathbb{F}}{\partial \mathbf{d}} - \frac{\partial (\mathbb{K}\mathbf{u})}{\partial \mathbf{d}} \end{aligned} \tag{A.7}$$

These comprise objective-independent derivatives for the airfoil problem. The elements of $\mathbb{K}$ have already been derived in the main body. The remainder of the shape derivatives will be calculated here at the element level. First, there are several commonly occurring derivatives that

are useful to compute first.

$$
\begin{aligned}
\frac{dx_\alpha}{dx_{\alpha k}} = R_k \frac{d(\mathbf{P}_i^{b,e})_\alpha}{dx_{\alpha k}} &= \frac{1}{w_i^{b,e}} C_{k,i}^e w_k (\mathbf{P}_k)_\alpha, \qquad \alpha \in \{1,2\},\ \ \mathbf{P}_k = \mathbf{P}_i^{b,e} \\
\frac{dx_\alpha}{d\tilde{\xi}} &= \sum_{i=1}^{p+1} w_i^{b,e} (\mathbf{P}_i^{b,e})_\alpha \frac{w^e \frac{dB_i}{d\tilde{\xi}} - B_i \frac{dw^e}{d\tilde{\xi}}}{(w^e)^2}, \qquad \alpha \in \{1,2\} \\
\frac{d}{dx_{\alpha k}} \left(\frac{dx_\alpha}{d\tilde{\xi}} \right) &= \sum_{i=1}^{p+1} w_i^{b,e} \frac{d}{dx_{\alpha k}} (\mathbf{P}_i^{b,e})_\alpha \frac{w^e \frac{dB_i}{d\tilde{\xi}} - B_i \frac{dw^e}{d\tilde{\xi}}}{(w^e)^2} = w_k \sum_{i=1}^{p+1} C_{k,i} \frac{w^e \frac{dB_i}{d\tilde{\xi}} - B_i \frac{dw^e}{d\tilde{\xi}}}{(w^e)^2} \\
\frac{d||\tilde{\mathbf{J}}^e||}{dx_{\alpha k}} &= \frac{d}{dx_{\alpha k}} \sqrt{\left(\frac{dx_1}{d\tilde{\xi}} \right)^2 + \left(\frac{dx_2}{d\tilde{\xi}} \right)^2} = \frac{\frac{d}{dx_{\alpha k}} \left(\frac{dx_\alpha}{d\tilde{\xi}} \right)}{||\tilde{\mathbf{J}}^e||} \\
\frac{d(\mathbf{n})_1}{dx_{1k}} &= \frac{\frac{dx_1}{d\tilde{\xi}}}{||\tilde{\mathbf{J}}^e||^2} \frac{d}{dx_{1k}} \left(\frac{dx_1}{d\tilde{\xi}} \right) \\
\frac{d(\mathbf{n})_1}{dx_{2k}} &= -\frac{||\tilde{\mathbf{J}}^e|| \frac{d}{dx_{2k}} \left(\frac{dx_2}{d\tilde{\xi}} \right) - \frac{dx_2}{d\tilde{\xi}} \frac{d||\tilde{\mathbf{J}}^e||}{dx_{2k}}}{||\tilde{\mathbf{J}}^e||^2} \\
\frac{d(\mathbf{n})_2}{dx_{1k}} &= \frac{||\tilde{\mathbf{J}}^e|| \frac{d}{dx_{1k}} \left(\frac{dx_1}{d\tilde{\xi}} \right) - \frac{dx_1}{d\tilde{\xi}} \frac{d||\tilde{\mathbf{J}}^e||}{dx_{1k}}}{||\tilde{\mathbf{J}}^e||^2} \\
\frac{d(\mathbf{n})_2}{dx_{2k}} &= -\frac{\frac{dx_2}{d\tilde{\xi}}}{||\tilde{\mathbf{J}}^e||^2} \frac{d}{dx_{2k}} \left(\frac{dx_2}{d\tilde{\xi}} \right)
\end{aligned}
\tag{A.8}
$$

To populate the $\frac{d\mathbf{R}}{d\mathbf{d}}$ matrix we begin by calculating the components of $\frac{d\mathbb{F}}{d\mathbf{d}}$. Trivially, $[\mathbb{F}]_{ik} = 0\ \forall i \in \{[1,n] \cup [2n+1, 3n] \cup 6n-1\} \cap \mathbb{Z}$. For $j \in [n+1, 2n] \cap \mathbb{Z}$ we will have

$$
[\mathbb{F}]_{jk} = \begin{cases} \frac{dU_i}{dx_{1k}} & k \leq n \\ \frac{dU_i}{dx_{2(k-n)}} & k > n \end{cases}, \qquad i = j - n \tag{A.9}
$$

where

$$
\begin{aligned}
\frac{dU_i}{dx_{\alpha k}} &= \frac{\partial U_i}{\partial (\mathbf{n})_1} \frac{d(\mathbf{n})_1}{dx_{\alpha k}} + \frac{\partial U_i}{\partial (\mathbf{n})_2} \frac{d(\mathbf{n})_2}{dx_{\alpha k}} \\
&= V_\infty \cos\nu \frac{d(\mathbf{n})_1}{dx_{\alpha k}} + V_\infty \sin\nu \frac{d(\mathbf{n})_2}{dx_{\alpha k}}, \qquad k \in \{1,\ 2,\ \cdots, n\}.
\end{aligned}
\tag{A.10}
$$

Next, for $j \in [3n+1, 4n] \cap \mathbb{Z}$ we have

$$[\mathbb{F}]_{jk} = \begin{cases} \frac{dV_i}{dx_{1k}} & k \leq n \\ \\ \frac{dV_i}{dx_{2(k-n)}} & k > n \end{cases}, \qquad i = j - 3n \tag{A.11}$$

where

$$\begin{aligned} \frac{dV_i}{dx_{\alpha k}} &= \frac{dV_i}{d(\mathbf{x})_\alpha}\frac{d(\mathbf{x})_\alpha}{dx_{\alpha k}} + \frac{\partial V_i}{\partial(\mathbf{n})_1}\frac{d(\mathbf{n})_1}{dx_{\alpha k}} + \frac{\partial V_i}{\partial(\mathbf{n})_2}\frac{d(\mathbf{n})_2}{dx_{\alpha k}} \\ &= \frac{dV_i}{d(\mathbf{x})_\alpha}R_k + \frac{(\mathbf{x})_2 - (\mathbf{c})_2}{2\pi||\mathbf{x}-\mathbf{c}||^2}\frac{d(\mathbf{n})_1}{dx_{\alpha k}} - \frac{(\mathbf{x})_1 - (\mathbf{c})_1}{2\pi||\mathbf{x}-\mathbf{c}||^2}\frac{d(\mathbf{n})_2}{dx_{\alpha k}}, \qquad k \in \{1,\ 2, \cdots, n\}, \\ \frac{dV_i}{d(\mathbf{x})_\alpha} &= \begin{cases} \frac{1}{2\pi}\frac{-(\mathbf{n})_2||\mathbf{x}-\mathbf{c}||^2 - 2[(\mathbf{n})_1((\mathbf{x})_2-(\mathbf{c})_2) - (\mathbf{n})_2((\mathbf{x})_1-(\mathbf{c})_1)]((\mathbf{x})_1-(\mathbf{c})_1)}{||\mathbf{x}-\mathbf{c}||^4} & \alpha = 1 \\ \frac{1}{2\pi}\frac{(\mathbf{n})_1||\mathbf{x}-\mathbf{c}||^2 - 2[(\mathbf{n})_1((\mathbf{x})_2-(\mathbf{c})_2) - (\mathbf{n})_2((\mathbf{x})_1-(\mathbf{c})_1)]((\mathbf{x})_2-(\mathbf{c})_2)}{||\mathbf{x}-\mathbf{c}||^4} & \alpha = 2 \end{cases}. \end{aligned} \tag{A.12}$$

Next for $j \in [4n+1, 5n-1] \cap \mathbb{Z}$ we have

$$[\mathbb{F}]_{jk} = \begin{cases} \frac{d\tilde{U}_i}{dx_{1k}} & k \leq n \\ \\ \frac{d\tilde{U}_i}{dx_{2(k-n)}} & k > n \end{cases}, \qquad i = j - 4n \tag{A.13}$$

where

$$\begin{aligned} \frac{dU_i}{dx_{\alpha k}} &= \frac{\partial U_i}{\partial(\mathbf{t})_1}\frac{d(\mathbf{t})_1}{dx_{\alpha k}} + \frac{\partial U_i}{\partial(\mathbf{t})_2}\frac{d(\mathbf{t})_2}{dx_{\alpha k}} \\ &= V_\infty \cos\alpha \frac{d(\mathbf{t})_1}{dx_{\alpha k}} + V_\infty \sin\nu \frac{d(\mathbf{t})_2}{dx_{\alpha k}} \\ &= -V_\infty \cos\nu \frac{d(\mathbf{n})_2}{dx_{\alpha k}} + V_\infty \sin\alpha \frac{d(\mathbf{n})_1}{dx_{\alpha k}}, \qquad k \in \{1,\ 2, \cdots, n\}. \end{aligned} \tag{A.14}$$

Finally for $j \in [5n, 6n-2] \cap \mathbb{Z}$,

$$[\mathbb{F}]_{jk} = \begin{cases} \frac{d\tilde{V}_i}{dx_{1k}} & k \leq n \\ \\ \frac{d\tilde{V}_i}{dx_{2(k-n)}} & k > n \end{cases}, \qquad i = j - (5n-1) \tag{A.15}$$

where

$$\begin{aligned}\frac{d\tilde{V}_i}{dx_{\alpha k}} &= \frac{d\tilde{V}_i}{d(\mathbf{x})_\alpha}\frac{d(\mathbf{x})_\alpha}{dx_{\alpha k}} + \frac{\partial\tilde{V}_i}{\partial(\mathbf{t})_1}\frac{d(\mathbf{t})_1}{dx_{\alpha k}} + \frac{\partial\tilde{V}_i}{\partial(\mathbf{t})_2}\frac{d(\mathbf{t})_2}{dx_{\alpha k}} \\ &= \frac{d\tilde{V}_i}{d(\mathbf{x})_\alpha}R_k - \frac{(\mathbf{x})_2-(\mathbf{c})_2}{2\pi||\mathbf{x}-\mathbf{c}||^2}\frac{d(\mathbf{n})_2}{dx_{\alpha k}} - \frac{(\mathbf{x})_1-(\mathbf{c})_1}{2\pi||\mathbf{x}-\mathbf{c}||^2}\frac{d(\mathbf{n})_1}{dx_{\alpha k}}, \qquad k\in\{1,\ 2,\ \cdots,n\}.\end{aligned} \tag{A.16}$$

Next, since the auxiliary variables have no direct dependence on the design variables, to take $\frac{\partial(\mathbb{K}\mathbf{u})}{\partial\mathbf{d}}$ we need only take the partials of the components of $\mathbb{K}$

$$[\mathbb{K}\mathbf{u}]_i = \sum_{j=1}^{6n-1}\mathbb{K}_{ij}\mathbf{u}_j \quad\Rightarrow\quad \left[\frac{\partial(\mathbb{K}\mathbf{u})}{\partial\mathbf{d}}\right]_i = \sum_{j=1}^{6n-1}\frac{\partial[\mathbb{K}]_{ij}}{\partial\mathbf{d}}\mathbf{u}_j \tag{A.17}$$

so it will suffice to compute the partials of the submatrices of $\mathbb{K}$. The partials of K are computed as follows

$$\begin{aligned}\frac{d}{dx_{\alpha k}}[K]_{ij} &= \frac{d}{dx_{\alpha k}}\left(\frac{1}{2}R_j(\mathbf{y}(\tau_i)) + \int_\Gamma h(\mathbf{x},\mathbf{y}(\tau_i))R_j(\mathbf{x})d\Gamma_x\right) \\ &= \cancelto{0}{\frac{d}{dx_{\alpha k}}\left(\frac{1}{2}R_j(\tau_i)\right)} + \frac{d}{dx_{\alpha k}}\left(\sum_{e=1}^{n_{el}}\int_0^1 h(\mathbf{x},\mathbf{y}(\tau_i))R_j(\mathbf{x})||\tilde{\mathbf{J}}^e||d\tilde{\xi}\right) \\ &= \sum_{e=1}^{n_{el}}\left(\int_0^1\frac{dh(\mathbf{x},\mathbf{y}(\tau_i))}{dx_{\alpha k}}R_j(\mathbf{x})||\tilde{\mathbf{J}}^e||d\tilde{\xi} + \int_0^1 h(\mathbf{x},\mathbf{y}(\tau_i))R_j(\mathbf{x})\frac{d||\tilde{\mathbf{J}}^e||}{dx_{\alpha k}}d\tilde{\xi}\right)\end{aligned} \tag{A.18}$$

where

$$\begin{aligned}\frac{dh(\mathbf{x},\mathbf{y}(\tau_i))}{dx_{\alpha k}} &= \frac{dh(\mathbf{x},\mathbf{y}(\tau_i))}{d(\mathbf{x})_\alpha}\frac{d(\mathbf{x})_\alpha}{dx_{\alpha k}} + \frac{dh(\mathbf{x},\mathbf{y}(\tau_i))}{d(\mathbf{y}_i)_\alpha}\frac{d(\mathbf{y}_i)_\alpha}{dx_{\alpha k}} \\ &\quad + \frac{dh(\mathbf{x},\mathbf{y}(\tau_i))}{d(\mathbf{n})_1}\frac{d(\mathbf{n})_1}{dx_{\alpha k}} + \frac{dh(\mathbf{x},\mathbf{y}(\tau_i))}{d(\mathbf{n})_2}\frac{d(\mathbf{n})_2}{dx_{\alpha k}} \\ &= \frac{dh(\mathbf{x},\mathbf{y}(\tau_i))}{d(\mathbf{x})_\alpha}R_k(\xi) + \frac{dh(\mathbf{x},\mathbf{y}(\tau_i))}{d(\mathbf{y}_i)_\alpha}R_k(\tau_i) \\ &\quad + \frac{dh(\mathbf{x},\mathbf{y}(\tau_i))}{d(\mathbf{n})_1}\frac{d(\mathbf{n})_1}{dx_{\alpha k}} + \frac{dh(\mathbf{x},\mathbf{y}(\tau_i))}{d(\mathbf{n})_2}\frac{d(\mathbf{n})_2}{dx_{\alpha k}},\end{aligned} \tag{A.19}$$

with

$$\begin{aligned}
\frac{dh(\mathbf{x},\mathbf{y}(\tau_i))}{d(\mathbf{x})_\alpha} &= \frac{1}{2\pi}\frac{||\mathbf{x}-\mathbf{y}_i||^2(\mathbf{n})_\alpha - 2((\mathbf{x})_\alpha-(\mathbf{y}_i)_\alpha)(\mathbf{x}-\mathbf{y}_i\cdot\mathbf{n})}{||\mathbf{x}-\mathbf{y}_i||^4},\\
\frac{dh(\mathbf{x},\mathbf{y}(\tau_i))}{d(\mathbf{y}_i)_\alpha} &= -\frac{dh(\mathbf{x},\mathbf{y}(\tau_i))}{d(\mathbf{x})_\alpha},\\
\frac{dh(\mathbf{x},\mathbf{y}(\tau_i))}{d(\mathbf{n})_1} &= \frac{(\mathbf{x})_1-(\mathbf{y}_i)_1}{2\pi||\mathbf{x}-\mathbf{y}_i||^2},\\
\frac{dh(\mathbf{x},\mathbf{y}(\tau_i))}{d(\mathbf{n})_2} &= \frac{(\mathbf{x})_2-(\mathbf{y}_i)_2}{2\pi||\mathbf{x}-\mathbf{y}_i||^2}.
\end{aligned} \tag{A.20}$$

The partials of Λ are computed as follows

$$\begin{aligned}
\frac{d}{dx_{\alpha k}}[\Lambda]_{ij} &= \frac{d}{dx_{\alpha k}}\left(-\int_\Gamma G(\mathbf{x},\mathbf{y}(\tau_i))R_j(\mathbf{x})d\Gamma_x\right)\\
&= \frac{d}{dx_{\alpha k}}\left(-\sum_{e=1}^{n_{el}}\int_0^1 G(\mathbf{x},\mathbf{y}(\tau_i))R_j(\mathbf{x})||\tilde{\mathbf{J}}^e||d\tilde{\xi}\right)\\
&= -\sum_{e=1}^{n_{el}}\left(\int_0^1 \frac{dG(\mathbf{x},\mathbf{y}(\tau_i))}{dx_{\alpha k}}R_j(\mathbf{x})||\tilde{\mathbf{J}}^e||d\tilde{\xi} + \int_0^1 G(\mathbf{x},\mathbf{y}(\tau_i))R_j(\mathbf{x})\frac{d||\tilde{\mathbf{J}}^e||}{dx_{\alpha k}}d\tilde{\xi}\right),
\end{aligned} \tag{A.21}$$

where

$$\begin{aligned}
\frac{dG(\mathbf{x},\mathbf{y}(\tau_i))}{dx_{\alpha k}} &= \frac{dG(\mathbf{x},\mathbf{y}(\tau_i))}{d(\mathbf{x})_\alpha}\frac{d(\mathbf{x})_\alpha}{dx_{\alpha k}} + \frac{dG(\mathbf{x},\mathbf{y}(\tau_i))}{d(\mathbf{y}_i)_\alpha}\frac{d(\mathbf{y}_i)_\alpha}{dx_{\alpha k}}\\
&= \frac{dG(\mathbf{x},\mathbf{y}(\tau_i))}{d(\mathbf{x})_\alpha}R_k(\xi) + \frac{dG(\mathbf{x},\mathbf{y}(\tau_i))}{d(\mathbf{y}_i)_\alpha}R_k(\tau_i)
\end{aligned} \tag{A.22}$$

with

$$\begin{aligned}
\frac{dG(\mathbf{x},\mathbf{y}(\tau_i))}{d(\mathbf{x})_\alpha} &= \frac{1}{2\pi}\frac{(\mathbf{x})_\alpha-(\mathbf{y}_i)_\alpha}{||\mathbf{x}-\mathbf{y}_i||^2},\\
\frac{dG(\mathbf{x},\mathbf{y}(\tau_i))}{d(\mathbf{y}_i)_\alpha} &= -\frac{1}{2\pi}\frac{(\mathbf{x})_\alpha-(\mathbf{y}_i)_\alpha}{||\mathbf{x}-\mathbf{y}_i||^2}.
\end{aligned} \tag{A.23}$$

Because the collocation points are defined in parametric space, a change in mapping to physical space has no affect on the quantity $R_j(\mathbf{x}_i)$ and thus

$$\frac{d}{dx_{\alpha k}}[D]_{ij} = 0 \tag{A.24}$$

In the case of $\tilde{D}$, however, because of the dependence on the Jacobian there will be a non-zero

shape derivative

$$\frac{d}{dx_{\alpha k}}[\tilde{D}]_{ij} = -\frac{\frac{d||\mathbf{J}||}{dx_{\alpha k}}}{||\mathbf{J}||^2}\bar{R}_j(\mathbf{x}_i) \tag{A.25}$$

where we can compute the shape derivative of the global Jacobian by expressing the global Jacobian in terms of the local Jacobian. From 2.23 we can relate the two via

$$\begin{aligned} ||\mathbf{J}|| &= ||\tilde{\mathbf{J}}^e||\frac{d\tilde{\xi}}{d\xi} \\ &= \frac{||\tilde{\mathbf{J}}^e||}{\xi_{l+1} - \xi_l} \end{aligned} \tag{A.26}$$

Then we can express the projection matrix derivatives in terms of known quantities

$$\frac{d}{dx_{\alpha k}}[\tilde{D}]_{ij} = -\frac{(\xi_{l+1} - \xi_l)\frac{d||\tilde{\mathbf{J}}^e||}{dx_{\alpha k}}}{||\tilde{\mathbf{J}}^e||^2}\bar{R}_j(\mathbf{x}_i) \tag{A.27}$$

Finally, we count ourselves lucky by noting that none of A.5 depends on geometry and we can state simply

$$\frac{d}{dx_{\alpha k}}[\mathbb{K}]_{6n-1,j} = 0, \qquad j \in \{1\ \ 2\ \cdots\ 6n-1\} \tag{A.28}$$

The derivation of the shape derivatives agnostic to the objective function is complete. We will examine application to an optimization problem.

A.1 Example Airfoil Optimization Problem

We will briefly examine the full process of deriving the shape sensitivities for a specific objective function

$$\mathcal{F}(\mathbf{d}, \mathbf{u}) = \int_\Gamma (C_P - C_P^h)^2 d\Gamma \tag{A.29}$$

where C_P is a target pressure distribution and C_P^h is the pressure distribution generated by the airfoil defined by $\mathbf{d}$. We note this function is positive definite with respect to its integrand and is therefore minimized at $\mathcal{F} = 0$. To compute the total gradient with respect to the control points, we first take the partial derivatives of the function with respect to the design and auxiliary variables.

To take the shape derivative of $\mathcal{F}$, we put the integral in parametric form

$$\begin{aligned}\mathcal{F} &= \int_{\xi_1}^{\xi_{n+p+1}} (C_P - C_P^h)^2 ||\mathbf{J}|| d\xi \\ &= \sum_{e=1}^{n_{el}} \left(\int_0^1 (C_P - C_P^h)^2 ||\mathbf{J}^e|| d\tilde{\xi} \right)\end{aligned} \tag{A.30}$$

and we see immediately that

$$\frac{\partial \mathcal{F}}{\partial x_{\alpha k}} = \sum_{e=1}^{n_{el}} \int_0^1 (C_P - C_P^h)^2 \frac{d||\mathbf{J}^e||}{dx_{\alpha k}} d\tilde{\xi}. \tag{A.31}$$

yields the elements of $\bar{\nabla}_{\mathbf{d}} \mathcal{F}$. To take partials with respect to the auxiliary variables, we first refer to 3.22 and 3.27 to compute the partials of C_P. We find

$$\begin{aligned}
\frac{\partial C_P^h}{\partial V} &= -\frac{2V}{V_\infty^2} \\
\frac{\partial V}{\partial \phi_j} &= \frac{1}{||\mathbf{J}||} \frac{dR_j(\mathbf{x})}{d\xi} \\
\frac{\partial V}{\partial \tilde{u}_k} &= \frac{1}{||\mathbf{J}||} \bar{R}_k(\mathbf{x}) \\
\frac{\partial V}{\partial \psi_j} &= \frac{\beta}{||\mathbf{J}||} \frac{dR_j(\mathbf{x})}{d\xi} \\
\frac{\partial V}{\partial \tilde{v}_k} &= \frac{\beta}{||\mathbf{J}||} \bar{R}_k(\mathbf{x}), \qquad j \in \{1,\ 2,\ \cdots, n\},\quad k \in \{1,\ 2,\ \cdots, n-1\} \\
\frac{\partial V}{\partial \beta} &= \left(\sum_{i=1}^{n} \psi_i \frac{dR_i(\mathbf{x})}{d\xi} + \sum_{i=1}^{n-1} \tilde{v}_i \bar{R}_i(\mathbf{x}) \right) \frac{1}{||\mathbf{J}||}
\end{aligned} \tag{A.32}$$

We can take the partial of $\mathcal{F}$ with respect to C_P^h

$$\frac{\partial \mathcal{F}}{\partial C_P^h} = \sum_{e=1}^{n_{el}} - \int_0^1 2(C_P - C_P^h) ||\mathbf{J}^e|| d\tilde{\xi} \tag{A.33}$$

then combine A.33 with A.32 and invoke the chain rule to find

$$
\begin{aligned}
\frac{\partial \mathcal{F}}{\partial \phi_j} &= \sum_{e=1}^{n_{el}} \frac{4}{V_\infty^2} \int_0^1 (C_P - C_P^h) V \frac{dR_j}{d\xi} \frac{||\mathbf{J}^e||}{||\mathbf{J}||} d\tilde{\xi} \\
\frac{\partial \mathcal{F}}{\partial \tilde{u}_k} &= \sum_{e=1}^{n_{el}} \frac{4}{V_\infty^2} \int_0^1 (C_P - C_P^h) V \bar{R}_k \frac{||\mathbf{J}^e||}{||\mathbf{J}||} d\tilde{\xi} \\
\frac{\partial \mathcal{F}}{\partial \psi_j} &= \sum_{e=1}^{n_{el}} \frac{4\beta}{V_\infty^2} \int_0^1 (C_P - C_P^h) V \frac{dR_j}{d\xi} \frac{||\mathbf{J}^e||}{||\mathbf{J}||} d\tilde{\xi} \\
\frac{\partial \mathcal{F}}{\partial \tilde{v}_k} &= \sum_{e=1}^{n_{el}} \frac{4\beta}{V_\infty^2} \int_0^1 (C_P - C_P^h) V \bar{R}_k \frac{||\mathbf{J}^e||}{||\mathbf{J}||} d\tilde{\xi}, \qquad j \in \{1,\ 2,\ \cdots, n\},\quad k \in \{1,\ 2,\ \cdots, n-1\} \\
\frac{\partial \mathcal{F}}{\partial \beta} &= \sum_{e=1}^{n_{el}} \frac{4}{V_\infty^2} \int_0^1 (C_P - C_P^h) V \left(\sum_{i=1}^{n} \psi_i \frac{dR_i(\mathbf{x})}{d\xi} + \sum_{i=1}^{n-1} \tilde{v}_i \bar{R}_i(\mathbf{x}) \right) \frac{||\mathbf{J}^e||}{||\mathbf{J}||} d\tilde{\xi}
\end{aligned}
\tag{A.34}
$$

Appendix B

L^2 Projection of a Function onto a Basis

In the body of this thesis we discussed projecting continuous functions onto a spline basis via collocation. This method of projection is relatively simple and can be derived heuristically. In this appendix we will derive the linear system that allows for a different variation on function projection.

To perform L^2 optimization, we begin with the formal definition of the problem. Suppose we have a function $f(x)$, $x \in \Omega \subset \mathbb{R}$ which we wish to project into a finite-dimensional function space $\mathcal{Q}$, spanned by n functions $N_i(x)$, i.e.

$$f^h(x) = \sum_{i=1}^{n} f_i N_i(x)$$

We ultimately wish to find the set of f_i that make f^h as 'close' as possible to the original f. To do this we must have a notion of distance between functions. We elect the L^2 norm over Ω, where

$$||g(x)||_{L^2_\Omega} \equiv \sqrt{\int_\Omega g^2(x) d\Omega}.$$

Our problem is to find the values of f_i when the distance is minimized between the projected and original functions. We note that this is equivalent to finding the minimum of the squared distance,

and thus we can ignore the square root in the above equation. We define the value

$$L = ||f^h(x) - f(x)||^2_{L^2_\Omega} = \int_\Omega (f^h(x) - f(x))^2 d\Omega$$

for convenience. Thus we can state the problem as determining

$$arg\,min_{\{f_i\}} L.$$

To do this, we simply take the gradient $\nabla_{f_i} L$ and set to zero in typical calculus fashion. Expanding L, we find that

$$\begin{aligned} L &= \int_\Omega \left(\sum_{i=1}^n f_i N_i(x) - f(x) \right)^2 d\Omega \\ &= \int_\Omega \left(\sum_{i=1}^n \sum_{j=1}^n f_i f_j N_i(x) N_j(x) - f_i N_i(x) f(x) \right) d\Omega \\ &= \sum_{i=1}^n \sum_{j=1}^n \left(\int_\Omega f_i f_j N_i(x) N_j(x) d\Omega - \int_\Omega f_i N_i(x) f(x) d\Omega \right) \end{aligned}$$

and so we can calculate

$$\frac{\partial L}{\partial f_i} = 2 \left[\sum_{j=1}^n \left(\int_\Omega f_j N_i(x) N_j(x) d\Omega \right) - \int_\Omega N_i(x) f(x) d\Omega \right] = 0$$

Thus setting the gradient equal to zero has yielded a linear system of n equations, namely

$$\sum_{j=1}^n \left(\int_\Omega f_j N_i(x) N_j(x) d\Omega \right) = \int_\Omega N_i(x) f(x) d\Omega, \qquad \forall i \in [1\ n] \cap \mathbb{Z}.$$

We can write this system as a matrix vector equation $\mathbf{L}x = \mathbf{F}$ where

$$[\mathbf{L}]_{ij} = \int_{\Omega} N_i(x) N_j(x) d\Omega$$

$$[x]_i = f_i$$

$$[\mathbf{F}]_i = \int_{\Omega} N_i(x) f(x) d\Omega$$

Assuming linearly independent basis functions, we can simply invert the system to solve for the coefficients f_i. Thus we have determined the closest approximation to f in $\mathcal{Q}$. A similar process could be derived for a norm other than $||\cdot||_{L^2_\Omega}$, however in the context of shape optimization it makes the most sense to use this norm as it provides a global measure of Euclidean separation from the target airfoil which is desirable.