NOVEL INVERSE AIRFOIL DESIGN

UTILIZING PARAMETRIC EQUATIONS

A Thesis
presented to
the Faculty of California Polytechnic State University,
San Luis Obispo

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in Aerospace Engineering

by
Kevin A. Lane
June 2010
COMMITTEE MEMBERSHIP

TITLE : NOVEL INVERSE AIRFOIL DESIGN
        UTILIZING PARAMETRIC EQUATIONS

AUTHOR: Kevin A. Lane

DATE SUBMITTED: June 2010

Dr. David D. Marshall
Advisor and Committee Chair
Aerospace Engineering Dept.

Dr. Rob A. McDonald
Committee Member
Aerospace Engineering Dept.

Dr. Jin Tso
Committee Member
Aerospace Engineering Dept.

Dr. Timothy Takahashi
Committee Member
Northrop Grumman Corp.
ABSTRACT

NOVEL INVERSE AIRFOIL DESIGN
UTILIZING PARAMETRIC EQUATIONS

Kevin A. Lane

The engineering problem of airfoil design has been of great theoretical interest for almost a century and has led to hundreds of papers written and dozens of methods developed over the years. This interest stems from the practical implications of airfoil design. Airfoil selection significantly influences the application’s aerodynamic performance. Tailoring an airfoil profile to its specific application can have great performance advantages. This includes considerations of the lift and drag characteristics, pitching moment, volume for fuel and structure, maximum lift coefficient, stall characteristics, as well as off-design performance.

A common way to think about airfoil design is optimization, the process of taking an airfoil and modifying it to improve its performance. The classic design goal is to minimize drag subject to required lift and thickness values to meet aerodynamic and structural constraints. This is typically an expensive operation depending on the selected optimization technique because several flow solutions are often required in order to obtain an updated airfoil profile. The optimizer requires gradients of the design space for a gradient-based optimizer, fitness values of the members of the population for a genetic algorithm, etc.

An alternative approach is to specify some desired performance and find the airfoil profile that achieves this performance. This is known as inverse airfoil design. Inverse design is more computationally efficient than direct optimization because changes in the geometry can be related to the required change in performance, thus requiring fewer flow solutions to obtain an updated profile. The desired performance for an inverse design method is specified as a pressure or velocity distribution over the airfoil at given flight conditions. The improved efficiency of inverse design comes at a cost. Designing a target pressure distribution is no trivial matter and has severe implications on the end performance. There is also no guarantee a specified pressure or velocity distribution can be achieved. However, if an obtainable pressure or velocity distribution can be created that reflects design goals and meets design constraints, inverse design becomes an attractive option over direct optimization.

Many of the available inverse design methods are only valid for incompressible flow. Of those that are valid for compressible flow, many require modifications to the method if shocks are present in the flow. The convergence of the methods are also greatly slowed by the presence of shocks. This paper discusses a series of novel inverse design methods that do not depend on the freestream Mach number. They can be
applied to design cases with and without shocks while not requiring modifications to the methods. Shocks also do not have a significant impact on the convergence of the methods. Airfoils are represented with parametric equations from the CST method to control shape changes and relate them to the required changes in the pressure or velocity distribution. To display the power of the methods, design cases are presented in the subsonic and transonic regimes. A circulation control design case is also presented using one of the methods to further show the robustness of the method.
ACKNOWLEDGMENTS

I would first like to thank my thesis advisor Dr. David Marshall for guiding me during my time at Cal Poly as a grad student. Your advice has been invaluable in completing my degree. You stuck with me throughout what seems like my many thesis topics and helped me to finally focus my graduate work to what it is here. Hopefully one day you can have a grad student do an adjoint optimizer. Your often asked question “So when are you graduating?” helped motivate me to finish my work.

To Dr. Rob McDonald, your knowledge of available resources has provided me with a great deal of reading and thorough knowledge of the existing literature relevant to my studies.

To Brenda Kulfan, your discussions regarding the CST method have given me insight into the power of the method and have led to its use in this study.

I also want to thank Craig Hange and Clif Horne for funding this work as the technical monitors for the NASA Research Announcement awarded under Contract #NNL07AA55C.

To my friends back in Roseville and the ones I have made while at Cal Poly, you are far too many to mention, but all deserve gratitude. You know who you are. Your timely, albeit occasionally frustrating, distractions have gone a long way to keeping me sane, particularly during the many late nights in the lab.

To all the guys in the lab, you warrant special mention. Thanks for always being flexible with the computers and FLUENT licenses. This work would have taken a lot longer with just one computer. Having you around to bounce questions and ideas off of has been immensely helpful as well. I’m so happy that some of us ended up in San Diego together.

Last but certainly not least, I want to thank my parents. Your constant support and continual acceptance of “just one more year” have gotten me through all 6 of my years at Cal Poly, not to mention the 12 years of schooling prior. Your guidance, lessons, and sacrifices over the past 24 years have gotten me where I am and made me who I am today. For that, I cannot thank you enough. Maybe one day I can get you that red sports car you’ve always wanted Mom. Perhaps that’s a start.
Table of Contents

List of Tables xi

List of Figures xii

Nomenclature xiii

1 Introduction 1
   1.1 The Need for Inverse Design 2
   1.2 History of Inverse Design Methods 3
      1.2.1 Foundations: 1930-1960 3
      1.2.2 Extensions and Practical Application: 1960-1990 7
      1.2.3 Recent Developments: 1990-Present 12
   1.3 Specifying Target Pressure Distributions 14
      1.3.1 Stratford Pressure Recovery 15
      1.3.2 Peaky Distribution 19
      1.3.3 Pressure Distribution Optimization 20

2 Inverse Airfoil Design 23
   2.1 Relation Between Target Airfoils and Pressure Distributions 23
   2.2 Normal Perturbation 25
   2.3 Pressure Gradient 27
      2.3.1 Degree Elevation 29
   2.4 Error Metric 29

3 Class/Shape Transformation (CST) 33
   3.1 General Form 34
      3.1.1 Class Function 34
      3.1.2 Shape Function 35
      3.1.3 Curve Representation 36
   3.2 Degree Elevation 38
   3.3 Modification to Leading Edge Representation 39
   3.4 Curve Fitting 40
      3.4.1 General Form 41
7 Final Remarks

7.1 Conclusions .......................................................... 116
7.2 Future Work .......................................................... 117

Appendices

A Residual Correction Method ........................................ 119
A.1 Updated Scale Factor .............................................. 119

B Governing Equations ............................................... 121
B.1 Navier-Stokes Equations .......................................... 122
B.1.1 Continuity Equation ........................................... 122
B.1.2 Momentum Equations .......................................... 122
B.1.3 Energy Equation ................................................ 126
B.1.4 Vector Formulation .............................................. 128
B.2 Reynolds Averaged Navier-Stokes Equations ............... 129
B.2.1 Time-Averaged Quantities .................................... 130
B.2.2 Favre-Averaged (Mass-Averaged) Quantities .......... 131
B.2.3 Continuity Equation ............................................ 131
B.2.4 Momentum Equations ......................................... 132
B.2.5 Energy Equation ................................................ 133

C Grid Spacing Calculation ........................................... 136

D Grid Convergence ...................................................... 138
D.1 Grid Convergence Index (GCI) ................................. 138
D.2 Least Squares GCI .................................................. 142
D.3 Two-Term Series Expansion .................................... 144

Bibliography .............................................................. 147
List of Tables

5.3.1  Lift coefficient convergence \((C_{l,\infty} = 0.7026, p = 2.02)\)  

5.3.2  Drag coefficient convergence \((C_{d,\infty} = 0.0122, p = 6.74)\)  

5.4.1  Comparison of CC airfoil grids  

6.1.1  Number of subsonic iterations to reach given geometry error  

6.2.1  Number of transonic iterations to reach given geometry error
# List of Figures

1.3.1 Pressure distributions with rooftops and Stratford pressure recoveries 19  
2.1.1 Example initial and target airfoils for inverse design . . . . . . . . . 24  
2.1.2 Example initial and target pressure distributions for inverse design 25  
3.1.1 General airfoil defined with the class function . . . . . . . . . . . . 35  
3.1.2 CST shape functions and basis airfoils . . . . . . . . . . . . . . . . 37  
3.3.1 CST shape functions and basis airfoils with LE term . . . . . . . . 40  
3.4.1 Airfoils parameterized using Bernstein coefficient optimization . . 42  
3.4.2 Error for RAE 2822 parameterized with and without class function 42  
3.4.3 Error for RAE 2822 parameterized with and without LE term . . . 44  
3.4.4 Airfoil, thickness, and camber distributions of an RAE 2822 . . . . 44  
3.4.5 Airfoil, thickness, and camber distributions of a NACA 6412 . . . . 45  
3.4.6 Error for NACA 6412 parameterized with and without LE term . . 45  
4.2.1 Velocity profile for a turbulent boundary layer . . . . . . . . . . . . 51  
5.1.1 Structured C-grid over an RAE 2822 airfoil - 25,000 cells . . . . . 59  
5.1.2 Lift and drag convergence histories showing Courant number effects 63  
5.2.1 GACC airfoil geometry with blown flap . . . . . . . . . . . . . . . . 64  
5.2.2 GACC flap and slot geometry . . . . . . . . . . . . . . . . . . . . . . 65  
5.2.3 Structured grid over GACC airfoil - 166,250 cells . . . . . . . . . 67  
5.3.1 Lift and drag coefficient convergence with cell count . . . . . . . . 72  
5.3.2 Structured C-grid over an RAE 2822 airfoil - 23,985 cells . . . . . 74  
5.3.3 Pressure distribution for an RAE 2822 airfoil in transonic flow . . 75  
5.4.1 Pressure distribution for the GACC airfoil - $C_{\mu} = 0.1$ . . . . . 77  
6.1.1 Subsonic design results - normal perturbation . . . . . . . . . . . . 80  
6.1.2 Subsonic pressure residuals - normal perturbation . . . . . . . . . 81  
6.1.3 Subsonic airfoil residuals - normal perturbation . . . . . . . . . . 81  
6.1.4 Subsonic design error - normal perturbation . . . . . . . . . . . . 82  
6.1.5 Subsonic Bernstein coefficients - normal perturbation . . . . . . . 83  
6.1.6 Subsonic design results - pressure gradients . . . . . . . . . . . . 85  
6.1.7 Subsonic pressure residuals - pressure gradients . . . . . . . . . . 86
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1.8</td>
<td>Subsonic airfoil residuals - pressure gradients</td>
<td>86</td>
</tr>
<tr>
<td>6.1.9</td>
<td>Subsonic design error - pressure gradients</td>
<td>87</td>
</tr>
<tr>
<td>6.1.10</td>
<td>Subsonic Bernstein coefficients - pressure gradients</td>
<td>88</td>
</tr>
<tr>
<td>6.1.11</td>
<td>Subsonic design results - degree elevation</td>
<td>90</td>
</tr>
<tr>
<td>6.1.12</td>
<td>Subsonic pressure residuals - degree elevation</td>
<td>91</td>
</tr>
<tr>
<td>6.1.13</td>
<td>Subsonic airfoil residuals - degree elevation</td>
<td>91</td>
</tr>
<tr>
<td>6.1.14</td>
<td>Subsonic design error - degree elevation</td>
<td>92</td>
</tr>
<tr>
<td>6.1.15</td>
<td>Subsonic Bernstein coefficients - degree elevation</td>
<td>93</td>
</tr>
<tr>
<td>6.2.1</td>
<td>Transonic design results - normal perturbation</td>
<td>97</td>
</tr>
<tr>
<td>6.2.2</td>
<td>Transonic pressure residuals - normal perturbation</td>
<td>98</td>
</tr>
<tr>
<td>6.2.3</td>
<td>Transonic airfoil residuals - normal perturbation</td>
<td>98</td>
</tr>
<tr>
<td>6.2.4</td>
<td>Transonic design error - normal perturbation</td>
<td>99</td>
</tr>
<tr>
<td>6.2.5</td>
<td>Transonic Bernstein coefficients - normal perturbation</td>
<td>100</td>
</tr>
<tr>
<td>6.2.6</td>
<td>Transonic design results - pressure gradients</td>
<td>102</td>
</tr>
<tr>
<td>6.2.7</td>
<td>Transonic pressure residuals - pressure gradients</td>
<td>103</td>
</tr>
<tr>
<td>6.2.8</td>
<td>Transonic airfoil residuals - pressure gradients</td>
<td>103</td>
</tr>
<tr>
<td>6.2.9</td>
<td>Transonic design error - pressure gradients</td>
<td>104</td>
</tr>
<tr>
<td>6.2.10</td>
<td>Transonic Bernstein coefficients - pressure gradients</td>
<td>105</td>
</tr>
<tr>
<td>6.2.11</td>
<td>Transonic design results - degree elevation</td>
<td>107</td>
</tr>
<tr>
<td>6.2.12</td>
<td>Transonic pressure residuals - degree elevation</td>
<td>108</td>
</tr>
<tr>
<td>6.2.13</td>
<td>Transonic airfoil residuals - degree elevation</td>
<td>108</td>
</tr>
<tr>
<td>6.2.14</td>
<td>Transonic design error - degree elevation</td>
<td>109</td>
</tr>
<tr>
<td>6.2.15</td>
<td>Transonic Bernstein coefficients - degree elevation</td>
<td>110</td>
</tr>
<tr>
<td>6.3.1</td>
<td>Initial and target pressure distributions for CC design - $C_\mu = 0.1$</td>
<td>112</td>
</tr>
<tr>
<td>6.3.2</td>
<td>CCW results - normal perturbation</td>
<td>115</td>
</tr>
</tbody>
</table>
# Nomenclature

## English Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a$</td>
<td>Speed of sound, $m/s$</td>
</tr>
<tr>
<td>$A$</td>
<td>Bernstein polynomial coefficient</td>
</tr>
<tr>
<td>$c$</td>
<td>Chord length</td>
</tr>
<tr>
<td>$C$</td>
<td>Class function</td>
</tr>
<tr>
<td>$C_l$</td>
<td>Section lift coefficient</td>
</tr>
<tr>
<td>$C_d$</td>
<td>Section drag coefficient</td>
</tr>
<tr>
<td>$C_p$</td>
<td>Conventional pressure coefficient</td>
</tr>
<tr>
<td>$C_{\bar{p}}$</td>
<td>Canonical pressure coefficient</td>
</tr>
<tr>
<td>$D$</td>
<td>2D CST geometry matrix</td>
</tr>
<tr>
<td>$i$</td>
<td>Index value</td>
</tr>
<tr>
<td>$j$</td>
<td>Index value</td>
</tr>
<tr>
<td>$k$</td>
<td>Thermal conductivity, $W/(m \cdot K)$ or index value</td>
</tr>
<tr>
<td>$k_0$</td>
<td>Reference thermal conductivity, $W/(m \cdot K)$</td>
</tr>
<tr>
<td>$K$</td>
<td>Binomial coefficient</td>
</tr>
<tr>
<td>$M$</td>
<td>Mach number</td>
</tr>
<tr>
<td>$\hat{n}$</td>
<td>Normal vector</td>
</tr>
<tr>
<td>$N$</td>
<td>Bernstein polynomial order or number of iterations</td>
</tr>
<tr>
<td>$N1$</td>
<td>First exponent in class function</td>
</tr>
<tr>
<td>$N2$</td>
<td>Second exponent in class function</td>
</tr>
<tr>
<td>$p$</td>
<td>Static pressure, $N/m^2$</td>
</tr>
<tr>
<td>$r$</td>
<td>Scale factor</td>
</tr>
<tr>
<td>$R$</td>
<td>Gas constant, $J/(kg \cdot K)$</td>
</tr>
<tr>
<td>$Re$</td>
<td>Reynolds number</td>
</tr>
<tr>
<td>$S$</td>
<td>Component shape function or Stratford condition</td>
</tr>
<tr>
<td>$S_k$</td>
<td>Sutherland thermal conductivity constant, $K$</td>
</tr>
<tr>
<td>$S_{\mu}$</td>
<td>Sutherland viscosity constant, $K$</td>
</tr>
<tr>
<td>$T$</td>
<td>Temperature, $K$</td>
</tr>
<tr>
<td>$u_\tau$</td>
<td>Friction velocity, $m/s$</td>
</tr>
<tr>
<td>$u^+$</td>
<td>Non-dimensional velocity</td>
</tr>
</tbody>
</table>
Velocity magnitude, \( m/s \)

Streamwise location or direction vector, \( m \)

Height above surface or vertical location, \( m \)

Non-dimensional wall distance

**Greek Symbols**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \alpha )</td>
<td>Angle of attack, ( deg )</td>
</tr>
<tr>
<td>( \gamma )</td>
<td>Specific-heat ratio</td>
</tr>
<tr>
<td>( \delta )</td>
<td>Boundary layer thickness, ( m )</td>
</tr>
<tr>
<td>( \Delta )</td>
<td>Displacement or increment</td>
</tr>
<tr>
<td>( \zeta )</td>
<td>Non-dimensional vertical location</td>
</tr>
<tr>
<td>( \mu )</td>
<td>Dynamic viscosity, ( kg/(m \cdot s) )</td>
</tr>
<tr>
<td>( \nu )</td>
<td>Kinematic viscosity, ( m^2/s )</td>
</tr>
<tr>
<td>( \rho )</td>
<td>Mass density, ( kg/m^3 )</td>
</tr>
<tr>
<td>( \tau_w )</td>
<td>Surface shear stress, ( N/m^2 )</td>
</tr>
<tr>
<td>( \psi )</td>
<td>Non-dimensional streamwise location</td>
</tr>
</tbody>
</table>

**Subscripts**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>( i )</td>
<td>Index value</td>
</tr>
<tr>
<td>( j )</td>
<td>Jet conditions or index value</td>
</tr>
<tr>
<td>( L )</td>
<td>Lower surface</td>
</tr>
<tr>
<td>( sep )</td>
<td>Separation value</td>
</tr>
<tr>
<td>( t )</td>
<td>Target value</td>
</tr>
<tr>
<td>( TE )</td>
<td>Trailing edge</td>
</tr>
<tr>
<td>( U )</td>
<td>Upper surface</td>
</tr>
<tr>
<td>( x )</td>
<td>Streamwise location</td>
</tr>
<tr>
<td>( z )</td>
<td>Vertical location</td>
</tr>
<tr>
<td>( \infty )</td>
<td>Freestream or extrapolated value</td>
</tr>
<tr>
<td>( 0 )</td>
<td>Reference conditions</td>
</tr>
</tbody>
</table>

**Superscripts**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>( + )</td>
<td>Pseudo-inverse</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

Aerodynamic design/optimization methods are traditionally classified as one of two formulations. These are the direct problem (analysis) and the inverse problem (design). The direct airfoil problem consists of finding the performance of a given airfoil and the inverse airfoil problem finds the airfoil shape that yields some specified performance. A direct airfoil optimization involves changing the airfoil shape to find the optimal performance subject to constraints, e.g. minimizing drag coefficient while holding lift coefficient constant and maintaining a specified thickness. The desired performance for an inverse airfoil design is typically specified in the form of a pressure or velocity distribution and the airfoil shape which generates that performance must be found. Inverse airfoil design methods employ some sort of iterative scheme that relates the required change in the pressure/velocity distribution to the required shape change. An initial airfoil is specified and modifications are made to get closer to the desired performance. The airfoil modifications are determined by the selected design method.
1.1 The Need for Inverse Design

Since direct optimization techniques attempt to find the optimal solution, they typically utilize some sort of search technique (gradient-based optimizer), genetic algorithm, or some other optimization method. These can be very computationally expensive because several flow solutions must be calculated to determine the direction of deepest descent, fitness of individuals in the population, etc. in order to determine the shape change.

Drela\(^1\) notes that a single-point optimization results in a large degradation in off-design performance, as shown in a low Reynolds number airfoil redesign. Increasing the number of design points reduces this degradation, but results in oscillations in the pressure distribution and requires many more flow solutions. The oscillations were caused by bumps that arose in the optimized airfoil. The bumps take in the shear layer so that less fluid needs to mix to get reattachment, yielding smaller mixing loss and pressure drag. The bumps correspond to the transition/reattachment locations for the sampled design points. Drela attributes this to the optimizer exploiting at the smallest physical scales. Increasing the number of design variables increases the small-scale changes in the airfoil and additional design points are needed to control this undesirable feature, further increasing the required number of flow solutions.

It is evident in a direct optimization process that the required number of flow solutions increases drastically with the number of design variables. However, since the desired performance is specified for the inverse problem, many techniques exist
to relate the required change in the pressure/velocity distribution to the required shape change. Therefore, shape changes can be determined with fewer flow solutions for inverse design techniques than for direct optimization methods. This results in inverse design methods typically being much more computationally efficient than direct optimization techniques, making them very attractive to be used in practice.

The main drawback to inverse design methods is that the designer is left with the task of creating pressure/velocity distributions that reflect the design goals and meet required aerodynamic characteristics.² It can be difficult to specify a pressure/velocity distribution that satisfies all the design goals. Also, one cannot guarantee that an arbitrarily prescribed pressure/velocity distribution will yield a closed and non-re-entrant airfoil.³

1.2 History of Inverse Design Methods

1.2.1 Foundations: 1930-1960

Airfoil design began with the direct problem and consisted of candidate airfoils being tested either in flight or in a wind tunnel. Modifications were made to the airfoil profile until the desired performance was reached. Performance was evaluated simply with the measurement of the macro parameters, namely lift, drag, angle of attack range, and maximum lift. Advancements in aerodynamic theory permitted aerodynamic analysis to be conducted outside of the wind tunnel. The exact solution for the direct problem in incompressible flow is straightforward using complex po-
potential theory and conformal mapping techniques. Theodorsen published an exact solution to the direct airfoil problem in 1932. The theory was well understood before the turn of the century, but the lack of computational resources greatly slowed its application to the airfoil problem. Goldstein later greatly improved the utility of the method through an approximation to Theodorsen’s theory that incurred only a minimal decrease in accuracy.

Inverse airfoil design has its roots in conformal mapping and dates back to the early days of NACA. Jacobs believed that Theodorsen’s airfoil analysis method could be applied to the problem of airfoil design by inverting the analysis process. Contributions by Theodorsen and Garrick proved Jacobs’ theory possible. This development, along with measurements of pressure distributions and rapidly developing boundary layer theory, lead to the development of the NACA 6-series airfoils by Jacobs, a series of low-drag laminar-flow airfoils where the airfoil geometry is defined by the superposition of camber and thickness distributions. This was the first of many inverse airfoil design techniques. Another early inverse design method based on a small-disturbance assumption was presented by Glauert. The inverse approach of Jacobs initially received significant skepticism at Langley and was never fully appreciated even after it proved successful. Abbott worked with Jacobs and later published the airfoil work done at NACA. Abbott is quoted as saying the following regarding the work into the inverse design method:

“We were told that even the statement of the problem was mathematical
nonsense with implications that it was our ignorance that encourages us.”

Theodorsen\textsuperscript{13} later extended the inverse method of Jacobs. However, by this time, more powerful conformal mapping techniques had begun to arise, most notably the work of Mangler\textsuperscript{14} and Lighthill.\textsuperscript{15} The significance of the methods of Mangler and Lighthill is that they showed that the specified pressure/velocity distribution could not be entirely arbitrary. Instead, three important integral constraints must be satisfied in order for a solution to exist. The pressure/velocity distribution must satisfy one constraint that guarantees uniform freestream flow at infinity. The additional two constrains ensure closure of the airfoil profile. These theories extinguished doubts about the theoretical soundness of the inverse approach to airfoil design and form the basis for most modern incompressible design methods.

The need to satisfy the integral constraints resulted in an additional problem. In order to satisfy the three integral constraints, the inverse method required the introduction of three free parameters into the specified pressure/velocity distribution. Many successful ways have been devised to do this. However, a difficulty arises when the values determined for the free parameters lead to unrealistic pressure/velocity distributions. This in turn corresponds to unrealistic airfoils, e.g. crossed airfoils or figure eights as discussed by Garrick.\textsuperscript{16} To account for this, essentially all practical inverse airfoil design methods employ some kind of iterative technique.

Practical application of an inverse technique was initially severely hindered by the computational requirements involved in obtaining the final airfoil shape. Widespread
use of inverse methods would not occur until the computational ability of computers increased. Thus, most of the early work done on the inverse problem following the theory of Mangler and Lighthill focused primarily on improving both the accuracy and efficiency of the numerical solution. This includes the work of Peebles, Glauert, Timman, and Peebles and Parkin. Due to the lack of numerical resources, linearized procedures were the most popular. Allen proposed a thin airfoil theory that considers both the direct and inverse problems. The user specifies either a modification to an existing airfoil or its corresponding pressure distribution and the method calculates the pressure distribution or the airfoil shape, respectively. Weber developed a second-order linearized analysis/design technique that did not require an existing airfoil geometry or pressure distribution to be specified. However, the method was only effective when applied to airfoils with low to moderate camber and thickness.

The last shortcoming of the inverse method pertains to the theory itself, not the application. Methods based on the theory of Mangler and Lighthill are regarded as single-point inverse airfoil design methods, meaning that the desired pressure/velocity distribution is prescribed at a single angle of attack. However, most airfoils must operate over a range of operating conditions, not just at a single point. If an airfoil is designed by a single-point method, it must be analyzed after the design at operating conditions of interest to determine whether or not the airfoil satisfies multipoint design requirements. Therefore, if multipoint design requirements are required, the
single-point design methods tend to be very tedious. Even though single-point inverse
design methods have been used to design many successful airfoils, a theory that has
the explicit capability of handling multipoint design requirements from the outset is
typically favored. An example of this is the theory for multipoint airfoil design by
Eppler.\textsuperscript{23}

Another formulation of the inverse design problem is the mixed inverse method.
Drela\textsuperscript{24} describes the mixed inverse method as a design where a portion of the airfoil
geometry is given and the desired velocity distribution is specified over the remainder
of the airfoil. An example of the mixed inverse method is the method by Woods,\textsuperscript{25,26}
who solves the mixed-inverse problem in a plane analytically mapped from the phys-
ical domain, similar to the conformal mapping techniques of Mangler and Lighthill.

1.2.2 Extensions and Practical Application: 1960-1990

Barger and Brooks\textsuperscript{27} and Barger\textsuperscript{28-30} further expanded on the Jacobs method
of inverse design in a series of NASA reports implementing a streamline curvature
relationship, but the work received little attention as methods based on the work of
Mangler and Lighthill had come into favor. Another method developed at NASA
is the direct iterative surface curvature (DISC) method by Campbell and Smith.\textsuperscript{31}
Beginning in the late 1960s, emphasis began to shift from increasing the efficiency of
the numerical techniques towards practical application through the use of computers.
This includes the work of Nonweiler,\textsuperscript{32} van Ingen,\textsuperscript{33} Arlinger,\textsuperscript{34} Strand,\textsuperscript{35} and Polito.\textsuperscript{36}

By the 1970s, the inverse approach had developed into a very powerful tool for
airfoil design. However, it was not without shortcomings, many of which still exist today. The integral constraints required to guarantee a solution are expressed in terms of the pressure/velocity distribution around the airfoil as a function of the angular coordinate around the circle used to map the airfoil, not as a function of arc length. Therefore, the desired velocity distribution is indirectly specified as a function of the circle angular coordinate, not as a function of arc length around the airfoil. However, iterative techniques were introduced by Arlinger and James that allowed the desired pressure/velocity distribution to be specified from the outset, subject to satisfying the integral constraints. The method by James does not require that the specified pressure/velocity distribution satisfy the integral constraints. Instead, it modifies the target pressure/velocity distribution to meet the constraints and returns it along with its corresponding airfoil. It also has the benefit that the target pressure/velocity distribution is specified as a function of arc length of the airfoil, not angular coordinates of the circle into which it is mapped as other methods require.

Since Eppler published his theory on multipoint inverse airfoil design, the method has been improved and made readily available as a computer program by Eppler and Somers and later by Eppler. The Eppler method implements a multipoint design by dividing the airfoil into a desired number of segments. The velocity distribution is prescribed over each segment along with the design angle of attack that achieves the desired velocity distribution. This allows multipoint design requirements to be satisfied during the actual design process, not iteratively through post-design
analysis. Even though the method proved quite versatile as a practical design tool, the actual theory has received very little attention. Exceptions to this are the work done by Miley and Ormsbee and Maughmer. Miley applied the Eppler inverse method to the design of low Reynolds number airfoils, and Ormsbee and Maughmer derived the necessary conditions and integral constraints in order to obtain finite trailing-edge pressure gradients.

The next wave of methods extended the incompressible inverse design techniques to compressible flows and displayed transonic solutions. Steger and Klineberg presented an inverse small-disturbance method valid for transonic flows. Tranen applied the circle-mapping formulation to transonic flows. The compressible nonconservative form of the potential equation is mapped into the interior of the unit circle and solved with a finite difference method subject to Dirichlet airfoil surface boundary conditions. Volpe and Melnik note that this formulation leads to a formally ill-posed Dirichlet problem that cannot converge in the general case because it does not address the first integral constraint requiring uniform freestream flow at infinity. Therefore, the speed on the airfoil and the freestream speed cannot both be arbitrarily specified.

Volpe and Melnik and Volpe extended the methods by Mangler and Lighthill to transonic flow. This was previously problematic due to the lack of closed-form expressions for the integral constraints at supercritical speeds. However, they were known to exist because the incompressible design problem is a subset of the more general compressible design problem. Volpe and Melnik first presented a for-
mulation of the inverse problem for airfoil design in transonic flow by either treating the freestream velocity as a free parameter or scaling the target surface velocity in order to satisfy the constraint of uniform flow at infinity. Volpe\textsuperscript{53} notes that these are equivalent options since the ratio of surface velocity to freestream velocity is the quantity of interest.

Daripa and Sirovich\textsuperscript{54} present a compressible inverse design technique. The full potential equation is transformed into the potential stream function plane. The flow angle and Prandtl-Meyer function are chosen as the dependent variables. This is then transformed into the interior of the unit circle. Boundary conditions on the Prandtl-Meyer function are used to solve the transformed full potential equation. Though valid for compressible flows, this method is limited to subsonic flows.

Another distinct class of compressible inverse design methods is based on the hodograph transformation, such as those of Bauer et al.\textsuperscript{55} and Boerstoel and Huizing.\textsuperscript{56} These methods implement a solution scheme for the potential and stream function based on integration along complex characteristics. Hodograph techniques are computationally efficient, but can be difficult to use because the required inputs are not directly related to physical speed and arc length on the airfoil. To satisfy consistency requirements, the method of Bauer et al. adjusts the freestream Mach number, which is a very undesirable feature for a transonic design where the freestream Mach number is critical to the design.

Carlson\textsuperscript{57,58} solved the mixed inverse problem in transonic flow on a Cartesian
mesh. The geometry near the leading edge is prescribed and the leading edge radius is adjusted to obtain a closed trailing edge. Volpe and Melnik\textsuperscript{48} point out this does not satisfy Lighthill’s first constraint. Also, achieving trailing edge closure by modifying the leading edge radius is a poor design practice since the leading edge radius greatly influences airfoil performance.

The ISES code,\textsuperscript{59,60} solves the mixed-inverse problem by modifying the specified target pressure distribution to meet the integral constraints. Curvature continuity between segments with prescribed pressures and geometry is ensured by enforcing a smoothness constraint. This added constraint further modifies the target pressure distribution.

The methods of Sobieczky et al.\textsuperscript{61} and Sobieczky and Seebass\textsuperscript{62} utilize a fictitious gas method that is utilized to redesign existing airfoils to achieve a transonic shock-free condition at a desired design point. These methods have not gained much popularity, which is likely due to the fact that airfoils that are shock free at their design point may have poor off-design performance.\textsuperscript{24}

In 1985, Takanashi\textsuperscript{63} reduced the inverse problem for transonic wing design to a Dirichlet boundary value problem solved with the aid of the transonic integral equation method. The method is based on an iterative residual-correction technique. Takanashi’s method utilizes the three-dimensional full potential equation written in terms of the perturbation velocity potential. Takanashi notes that the most common approach is to directly solve the partial differential equation that describes the
flow. However, the residual-correction method was adopted to develop a more general design tool that could be applied to a wide range of design problems.

Jameson’s work in inverse aerodynamic design utilizing control theory began in 1988. Jameson, Jameson and Reuther, and Jameson et al. have since published several papers throughout the 1990s extending the control theory approach to viscous designs. In the control theory method, a gradient-based technique is adopted to determine shape changes. The direction of descent is selected as the direction of steepest descent of the Fréchet derivative of a cost function obtained by solving an adjoint partial differential equation. Jameson notes that each design cycle of the control theory approach requires the numerical solution of both the flow and the adjoint equations. This leads to the computational cost being roughly equal to two flow solutions, but is kept low by using multigrid techniques and preconditioning to accelerate the convergence of the flow solutions.

1.2.3 Recent Developments: 1990-Present

In 1992, Selig and Maughmer extended the multi-point inverse design procedure of Eppler. The updated formulation allows the velocity over each airfoil segment to be prescribed as a function of arc length and allows for both cusped and finite trailing-edge-angle airfoils. The thickness, pitching moment, and other important design parameters may also be specified. These extensions to the Eppler theory create a powerful method for inverse airfoil design in incompressible potential flow.

Beginning in the late 1990s, Vicini and Quagliarella and Takanashi et al. applied...
plied multi-objective genetic algorithms to inverse airfoil design. Using a genetic algorithm easily permits design considerations from multiple disciplines such as aerodynamics, structures, and fuel storage to be implemented. It also is easily adapted to a multipoint inverse design. While genetic algorithms are accurate and do not have the pitfalls of traditional search algorithms, such as finding local minima, they are computationally intensive because they require multiple start locations. When an aerodynamic analysis such as CFD is included in the objective function, use of a genetic algorithm can be a very costly design choice.

During the same period, Kim and Rho\textsuperscript{74,75} combined the hybrid inverse optimization technique of Santos et al.\textsuperscript{76} with the target pressure optimization technique of Obayashi and Takanashi\textsuperscript{77} for designing transonic airfoils and wings. The hybrid inverse optimization technique correlates the difference between the computed and target pressures to changes in the geometry with the use of a modified Garabedian-McFadden ordinary differential equation given by Malone et al.\textsuperscript{78} The hybrid inverse optimization method is so called because it solves the inverse design problem with an optimization technique. They also extended the hybrid inverse optimization method to dual-point design.

Yu and Campbell\textsuperscript{79} and Campbell\textsuperscript{80,81} took the DISC method of Campbell and Smith\textsuperscript{31} and added constraints to create the constrained direct iterative surface curvature (CDISC) method. This was later extended by Milholen\textsuperscript{82} to include the surface curvature 2D (SC2D) method, a method similar to the approach by Barger and
Two other inverse design methods are those of Dulikravich and Baker and Yu et al. Dulikravich and Baker presented an inverse design method based on an analytical Fourier series solution to a pair of linear differential equations with interrelated boundary conditions. The method was successfully tested at subsonic and transonic speeds for both airfoils and wings. The method by Yu et al. changes the airfoil shape with a small-perturbation equation derived from the streamline momentum equations, continuity equation, and isentropic relations. The geometry similarity assumption of near streamlines near the airfoil surface is also utilized. This method was shown to work for subsonic and transonic flows.

1.3 Specifying Target Pressure Distributions

Selecting a target pressure distribution can be a difficult task for a designer. It can be difficult to ensure all design considerations with this selection. A common pressure distribution is a laminar or turbulent rooftop region with a Stratford pressure recovery for high lift. Another potential pressure distribution is a peaky leading edge region for low drag in transonic flow. Also, methods have been developed to optimize the target pressure distribution to relieve the designer of some of the burden of specifying the target pressure distribution.
1.3.1 Stratford Pressure Recovery

In his classic paper on high lift aerodynamics, A. M. O. Smith notes that the pressure coefficient is typically presented as

\[ C_p = \frac{p - p_\infty}{\frac{1}{2} \rho U_\infty^2} = 1 - \left( \frac{U}{U_\infty} \right)^2 \]  

(1.3.1)

This formulation makes it difficult to know much about a given pressure distribution’s tendency to separate. However, if two pressure distributions can be equated through scaling the \( x \) and \( C_p \) directions, the distributions are identical with the exception for the Reynolds number effect, which is weak. If an airfoil is analyzed at twice the velocity as an airfoil with twice the chord length, the velocity gradients are significantly different, but the flows are identical due to the same Reynolds numbers. Smith then argues that since it is the dimensionless shape that matters, it is convenient to scale out the velocity magnitude and the chord length. Smith describes this as a canonical pressure coefficient \( \bar{C}_p \) expressed as

\[ \bar{C}_p = \frac{p - p_0}{\frac{1}{2} \rho U_0^2} = 1 - \left( \frac{U}{U_0} \right)^2 \]  

(1.3.2)

where \( U \) is the surface velocity and the subscript \( o \) denotes reference conditions, values at the start of deceleration. In the canonical pressure distribution, \( \bar{C}_p = 0 \) represents the beginning of the pressure recovery and \( \bar{C}_p = 1 \) is the maximum possible corresponding to \( U = 0 \). This form is much more useful for separation analysis. Smith equates the conventional pressure coefficient to the canonical pressure coefficient with
the following relation
\[ C_p = 1 - \frac{1 - C_{p_{FE}}}{1 - C_{p_{sep}}} (1 - \bar{C}_p) \]  \hspace{1cm} (1.3.3)

where \( \bar{C}_{p_{sep}} \) is the value at which the canonical pressure distribution separates. This can be related to the velocity at which the flow separates. Stratford\textsuperscript{86} presents the following formula that predicts the point of separation in an arbitrary decelerating flow
\[ \frac{\bar{C}_p}{(10^{-6}Re)^{1/10}} = S \]  \hspace{1cm} (1.3.4)

where
\[ S = \begin{cases} 0.39 & \text{if } d^2p/dx^2 \geq 0 \\ 0.35 & \text{if } d^2p/dx^2 \leq 0 \end{cases} \]  \hspace{1cm} (1.3.5)

and \( \bar{C}_p \leq 4/7 \). This can also be used to describe a flow that is ready to separate everywhere. When the flow reaches the limiting value of \( S \), separation is said to occur. Smith notes that the study conducted by Cebeci et al.\textsuperscript{87} better predicted separation than Stratford’s method, but Smith accepted Stratford’s method due to its convenience. If \( S \) is held at the limiting value of 0.39 for \( d^2p/dx^2 \geq 0 \), Eq. 1.3.4 is an ordinary differential equation for \( \bar{C}_p(x) \). Stratford presents the following solutions to Eq. 1.3.4:

\[ \bar{C}_p(x) = 0.645 \left\{ 0.435R_0^{1/5} \left[ (x/x_0)^{1/5} - 1 \right] \right\}^{2/n} \]  \hspace{1cm} (1.3.6a)

for \( \bar{C}_p \leq \frac{n-2}{n+1} \)

and
\[ \bar{C}_p(x) = 1 - \frac{a}{(x/x_0 + b)^{1/2}} \]  \hspace{1cm} (1.3.6b)

for \( \bar{C}_p \geq \frac{n-2}{n+1} \)
Here $x_0$ is the location of the start of the pressure recovery and $n$ is a constant that Stratford gives to be approximately 6. The coefficients $a$ and $b$ are arbitrary constants and are selected to match values and slopes in Eq. 1.3.6 at the joining point $\bar{C}_p = (n - 2)/(n + 1)$. Since the value for $x_0$ to achieve a desired trailing edge pressure coefficient is not always known for the given flight conditions, it is convenient to solve for $a$ and $b$ to ensure the desired pressure distribution characteristics. If it is assumed that a pressure distribution is designed to separate at the trailing edge ($x = c$), then for a given $\bar{C}_{psep}$ value (Smith suggests 0.4), Eq. 1.3.6b can be rearranged to solve for $x_0$.

\[
x_0 = \frac{c}{\left[\frac{a}{(1 - \bar{C}_{psep})}\right]^2 - b} \tag{1.3.7}
\]

This can be substituted into Eq. 1.3.6a to find the location of the joining point $x_1$ by using $\bar{C}_p = (n - 2)/(n + 1)$.

\[
x_1 = \left[\frac{(\bar{C}_p/0.645)^{n/2}}{0.435R_0^{1/5} + 1}\right]^5 x_0 \tag{1.3.8}
\]

These values can be used to calculate the derivatives of Eq. 1.3.6 at the joining point. The derivatives are expressed as

\[
\frac{\partial \bar{C}_p}{\partial x}(x) = \frac{1.29 \left(0.435R_0^{1/5}\right)^{2/n} \left[\left(x/x_0\right)^{1/5} - 1\right]^{2/n - 1}}{5nx_0 \left(x/x_0\right)^{4/5}} \quad \text{for} \quad \bar{C}_p \leq \frac{n-2}{n+1} \tag{1.3.9a}
\]

and

\[
\frac{\partial \bar{C}_p}{\partial x}(x) = \frac{a}{2x_0 \left(x/x_0 + b\right)^{3/2}} \quad \text{for} \quad C_p \geq \frac{n-2}{n+1} \tag{1.3.9b}
\]
Equations 1.3.6-1.3.9 can be connected to an optimizer to find the values of \(a\) and \(b\) that minimize the difference between the values and slopes of Eq. 1.3.6. The MATLAB\textsuperscript{88} function \texttt{fminunc} is used to find the optimal values of \(a\) and \(b\). This is an unconstrained optimizer of a multivariate function. Smith notes that the Stratford pressure recovery is the shortest possible pressure recovery for a given flow. Nothing better can be done without boundary layer control. It is also the path of minimum drag connecting two pressure points.

Figure 1.3.1 shows both canonical and conventional pressure distributions with flat rooftops and Stratford pressure recoveries. All the pressure distributions were designed for \(C_p = 0.2\) at the trailing edge, which Smith notes comes from general airfoil theory.

Each distribution uses a different \(\bar{C}_{p_{sep}}\) value to force the optimizer to select different \(a\) and \(b\) values. This changes the values of both \(x_0\) and the height of the rooftop. With a high rooftop \(C_p\), \(x_0\) must be small because the required pressure recovery to the trailing edge pressure is greater. Conversely, for a low rooftop \(C_p\), \(x_0\) can be much greater because the pressure recovery is not as significant. The rooftop value of the canonical pressure distributions is 0 due to the transformation between the canonical and conventional pressure coefficients. As previously mentioned, \(\bar{C}_p = 0\) is the start of the pressure rise. None of the canonical pressure distributions reach \(\bar{C}_p = 1\), signifying 100% dynamic pressure recovery. It is theoretically possible, but
Figure 1.3.1: Pressure distributions with rooftops and Stratford pressure recoveries

an infinite distance is required.

One of the best examples of airfoil designs incorporating Stratford pressure recoveries is the work by Liebeck\textsuperscript{3,89-92} and Liebeck and Ormsbee.\textsuperscript{93} Liebeck used flat laminar and turbulent rooftops along with Stratford pressure recoveries to design a series of high lift airfoils in subsonic flow.

### 1.3.2 Peaky Distribution

For transonic flow, a pressure distribution exhibiting low drag is the peaky leading edge distribution.\textsuperscript{94,95} This began with the work of Pearcey\textsuperscript{96} who tried to experimentally obtain “essentially shock-free flow.” Pearcey followed the principle that
reducing the strength of shock waves has substantial benefits even if the flow is not entirely shock free. A peaky leading edge distribution creates a local supersonic region by forcing a weak shock wave to form near the leading edge, hence the name peaky leading edge. This concentrates the intensity of the expansion waves near the leading edge. The expansion waves reflected at the sonic lines come back to the airfoil surface as compression waves. This isentropically decelerates the local supersonic flow to subsonic speeds and results in low drag compared to an airfoil with a stronger shock further aft of the peaky leading edge shock.

1.3.3 Pressure Distribution Optimization

Obayashi and Takanashi\textsuperscript{77} and Kim and Rho\textsuperscript{74,75} present a method of optimizing transonic pressure distributions with the use of a genetic algorithm. Obayashi and Takanashi represent pressure distributions with B-splines where the control points are the design variables. Kim and Rho utilize shape functions presented by van Egmond.\textsuperscript{97} The objective function minimizes $C_d$ subject to a specified $C_l$ and airfoil thickness $t$. The airfoil thickness is estimated by relating the pressure distribution to the freestream Mach number according to the expression given by van Egmond.

$$t = -\sqrt{1 - M_\infty^2} \frac{1}{2} \int_0^1 \frac{C_{p,u} + C_{p,l}}{2} dx$$

(1.3.10)

Additional constraints include that the $C_p$ at the suction peak must be less than or equal to the sonic $C_p$ and

$$\left(\frac{dC_{p,u}}{dx}\right)_{0.1 \leq x \leq 0.5} \approx 0$$

(1.3.11)
These constraints are intended to generate the sonic plateau pressure distributions exhibited by the NASA supercritical airfoils given by Harris. Also, separation of the boundary layer is avoided by enforcing

\[
\frac{dC_{p,u}}{dx} \leq 2.5
\]  

(1.3.12)

which is taken from observations of pressure distributions given by Harris. The viscous drag is estimated with the Squire-Young relation

\[
C_{dw} = 2\theta_{te} \left( \frac{U_{e,te}}{U_{\infty}} \right)^{3.2}
\]  

(1.3.13)

where \(\theta_{te}\) is the momentum thickness at the trailing edge and \(U_{e,te}\) is the potential velocity at the trailing edge. When a shock wave is present, Obayashi and Takanashi estimate the wave drag as

\[
C_{dw} = K \left( M^2 U^2 - 1 \right)^{4.4}
\]  

(1.3.14)

where \(K\) is a constant that Inger gives to be about 0.02 by applying Oswatitsch’s theorem to calculate the drag generated by a shock wave over supercritical airfoils. Kim and Rho calculate the wave drag according to the following expression by Campbell

\[
C_{dw} = \frac{0.04}{(t/c)^{1.5}} (M_{us} - 1)^4
\]  

(1.3.15)

where \(M_{us}\) signifies the Mach number just upstream of the shock. Additional relations are introduced to handle the location and strength of the shock. Obayashi and Takanashi note that a typical pressure jump across a shock on a supercritical airfoil
is about 70% of the Rankine-Hugoniot jump. They initially set the pressure jump to 60% as

$$\Delta C_p = C_{p,ds} - C_{p,us} = 0.6 \left( C_{p,us} + \frac{1}{0.7M_\infty^2} \right) \frac{7 \left( M_{us}^2 - 1 \right)}{6}$$ (1.3.16)

where $ds$ represents the value slightly downstream of the shock. Kim and Rho approximate the pressure jump across the shock according to the modified Rankine-Hugoniot relation given by Mabey

$$\frac{p_{ds}}{p_{us}} = 1 + \frac{2\gamma}{\gamma + 1} A \left( M_{us}^2 - 1 \right) \quad 0.55 < A < 0.75$$ (1.3.17)

where the range of $A$ is determined from experimental data in Mabey.
Chapter 2
Inverse Airfoil Design

For inverse airfoil design, a target pressure or velocity distribution and an initial airfoil shape are typically specified. The pressure or velocity distribution of the initial airfoil is computed and compared to the target distribution, and modifications are made to the initial airfoil according to the selected inverse design method. These modifications result in an airfoil that has a pressure or velocity distribution more closely resembling the target distribution. Inverse design methods are commonly tested by specifying the target pressure distribution as that of a known airfoil at a desired Mach number, Reynolds number, and angle of attack. As the pressure distribution converges to the target distribution, the airfoil should also converge to the target airfoil.

2.1 Relation Between Target Airfoils and Pressure Distributions

By examining initial and target airfoils and their corresponding pressure distributions, one can note a connection between the pressure distributions and how the
initial airfoil must change to match the target airfoil. Figure 2.1.1 shows an example of initial and target airfoils where the initial airfoil is a NACA 0012 and the target airfoil is an RAE 2822. To reach the target airfoil, the leading edge and lower aft region of the initial airfoil must be perturbed with a negative outward facing normal and the upper aft region must be perturbed with a positive outward facing normal.

Figure 2.1.1: Example initial and target airfoils for inverse design

Figure 2.1.2 shows initial and target pressure distributions for different Mach numbers and angles of attack. The initial and target airfoils are again a NACA 0012 and an RAE 2822 respectively. The pressures at the leading edge and lower aft region of the initial airfoil are all less than those of the target airfoil. Also, the pressures at the upper aft region of the initial airfoil are greater than those of the target airfoil. By comparing Figures 2.1.1 and 2.1.2, it is evident that wherever the pressure is greater than the target pressure ($C_p - C_{p,t} > 0$), the airfoil surface must be perturbed with a positive outward facing normal vector. Conversely, wherever the pressure is lower...
than the target pressure \( C_p - C_{p,t} < 0 \), the airfoil must be perturbed with a negative outward facing normal vector.

## 2.2 Normal Perturbation

The relation discussed in Section 2.1 provides a straightforward method to modify the airfoil shape in an inverse design process as proposed by Lane and Marshall.\(^{105}\) At each design iteration, the pressure residuals can be used to control the normal vector along which each point is perturbed. As previously mentioned, the pressure residuals and the required normal perturbation direction have the same sign. Therefore, the perturbation direction is given by the sign of the pressure residuals. Since the direction is known, the magnitude each point is moved along its normal vector simply needs to be scaled appropriately. The airfoil is perturbed according to

\[
\Delta \hat{n}(x) = r [C_p(x) - C_{p,t}(x)]
\]
where \( r \) is a scale factor between 0 and 1 to control convergence and ensure the airfoil is not perturbed enough to yield a non-realistic airfoil. Given the magnitudes of the required pressure and geometry changes, \( r \) is on the order of 0.05. Each airfoil coordinate is then perturbed by its corresponding magnitude along its normal vector to yield the updated airfoil. This is given by

\[
x_{i+1} = x_i + \hat{n}_{x,i} \cdot \Delta \hat{n}_i
\]

\[
y_{i+1} = y_i + \hat{n}_{y,i} \cdot \Delta \hat{n}_i
\]

This formulation is not strictly true, meaning the locations the initial and target pressures cross are not necessarily at the same locations as the initial and target airfoils. However, it provides the proper general trend for a wide range of design cases. Since this method is not guaranteed to result in a smooth airfoil, a smoothing process must be implemented. The updated airfoil coordinates are parameterized using the class/shape transformation (CST) method, which is discussed in Chapter 3. This produces a smooth airfoil to be used in the next iteration of the inverse design. This method will be referred to as the normal perturbation (NP) method. Since this method simply examines pressure distributions to obtain the updated airfoil, no assumptions are made regarding the fidelity of the model and the solver can be kept as a “black box.” Also, no assumptions are made about the flowfield. This method can be applied to any design case where the required pressure and geometry changes follow the general trend described by this method. This includes compressible viscous flows,
while many of the early methods were only valid for incompressible inviscid flows. This method builds on earlier work by Lane and Marshall\textsuperscript{106} where CST was used in a direct airfoil optimization scheme. The above method is simple to implement, but requires user interaction to get the best efficiency. The ideal scale factor depends on the flight conditions. After several iterations the airfoil may begin to oscillate around the target airfoil and the error ceases to reduce. Reducing the scale factor reduces this oscillations, but also slows convergence. What is needed is a more versatile method that determines not only the required direction of the shape change, but also the magnitude. Another potential formulation of this method is discussed in Appendix A where the scale factor is updated throughout the design to attempt to achieve better convergence.

\textbf{2.3 Pressure Gradient}

The required change in $C_p$ at a given point on the airfoil can be estimated as a first order series expansion of the derivative of $C_p$ with respect to the Bernstein coefficients used to define a CST curve.

$$\frac{dC_p}{dA} = \frac{\partial C_p}{\partial A_0} dA_0 + \frac{\partial C_p}{\partial A_1} dA_1 + \cdots \tag{2.3.1}$$

The Bernstein coefficients given by $A$ are the parameters used to define a CST curve and are discussed in the following chapter. The derivative of $C_p$ with respect to a given Bernstein coefficient is approximated as a first order finite difference.

$$\frac{\partial C_p}{\partial A_i} = \frac{C_{p,A_i+\Delta A_i} - C_{p,A_i}}{\Delta A_i} \tag{2.3.2}$$
The airfoil is perturbed according to $\Delta A_i$. Changing one of the Bernstein coefficients modifies the surface and influences the pressure distribution. The pressure distribution is then compared to the unperturbed airfoil pressure distribution ($C_{p,A_i}$). These derivatives are calculated at every point on the airfoil for each Bernstein coefficient used to represent the airfoil. These derivatives are then put into an aerodynamic derivative matrix. Since the change in $C_p$ due to $\Delta A_i$ is calculated at every point on the airfoil, the matrix is an $M_U + M_L$ by $N_U + N_L + 2$ matrix where $M$ is the number of airfoil points at which $C_p$ is measured and $N$ is the order of the Bernstein polynomial.

\[
\frac{\partial C_p}{\partial A} = \begin{bmatrix}
\frac{\partial C_{p0,U}}{\partial A_{0,U}} & \cdots & \frac{\partial C_{p0,U}}{\partial A_{0,L}} & \cdots & \frac{\partial C_{p0,U}}{\partial A_{N,0,U}} & \cdots \\
\vdots & \ddots & \vdots & & \vdots & \cdots \\
\frac{\partial C_{PM,U}}{\partial A_{0,0,U}} & \cdots & \frac{\partial C_{PM,U}}{\partial A_{0,0,L}} & \cdots & \frac{\partial C_{PM,U}}{\partial A_{PM,0,U}} & \cdots \\
\vdots & \ddots & \vdots & & \vdots & \cdots \\
\frac{\partial C_{PM,L}}{\partial A_{0,0,U}} & \cdots & \frac{\partial C_{PM,L}}{\partial A_{0,0,L}} & \cdots & \frac{\partial C_{PM,L}}{\partial A_{PM,0,U}} & \cdots \\
\end{bmatrix} \tag{2.3.3}
\]

Equation 2.3.1 must be solved at every point on the airfoil to find the change in the Bernstein coefficients that results in the required change in $\vec{C}_p$. This system of equations is put in matrix form to solve for the required increment to the set of Bernstein coefficients. Since this is an overdetermined rectangular system, the pseudo inverse is used to minimize the least squared error of the system. This is given as

\[
d\vec{A} = \left(\frac{\partial C_p}{\partial A}\right)^+ \cdot d\vec{C}_p \tag{2.3.4}
\]

where

\[
d\vec{C}_p = \vec{C}_{p,t} - \vec{C}_p \tag{2.3.5}
\]
This increment to the Bernstein coefficients is applied to the airfoil to obtain the airfoil for the next iteration. This process is repeated until the solution reaches a desired tolerance or a specified number of iterations. Like the previous inverse design method, this is independent of the flow solver and can be used with any solution technique. It is more computationally expensive because the gradients need to be calculated, but it is a more robust method and requires less user interaction. This method will subsequently be referred to as the pressure gradient (PG) method.

2.3.1 Degree Elevation

A low order CST curve can be used initially to quickly calculate pressure gradients and get the airfoil close to the target airfoil. Degree elevation can then be used in subsequent iterations to increase the control on the airfoil shape. This reduces the required number of flow solutions over the course of the entire design process because of the reduced number of flow solutions early in the design process. This will be referred to as the pressure gradient degree elevation (PGDE) method. Refer to Section 3.2 for details on CST degree elevation.

2.4 Error Metric

In order to compare how close two airfoils or pressure distributions are to one another, the square of the area between them is calculated. For two curves \( f_1 \) and \( f_2 \) with endpoints \( a \) and \( b \), this equates to the integral of the square of the difference
between the two curves over the interval \([a,b]\).

\[
A^2 = \int_a^b [f_1(x) - f_2(x)]^2 dx \quad (2.4.1)
\]

The area is squared so that when the curves cross each other the area is not reduced. If only the difference between the curves is integrated, the area reduces when \(f_2\) is greater than \(f_1\) because the difference is negative. This would report a lower error than is actually present. The integral can be approximated numerically using one of the closed Newton-Cotes formulas,\(^{107}\) which encompass a method of quadrature. These are a series of numerical integration formulas that work by dividing the interval \([a,b]\) into \(n\) equal parts and using a polynomial of order \(n - 1\) to approximate the area under the curve. The polynomials are found using the Lagrange interpolating polynomial, which is the polynomial of degree \(n - 1\) that passes through the \(n\) points given by \((x_i, y_i = f(x_i))\). The Lagrange polynomial takes the form\(^{108}\)

\[
P(x) = \sum_{i=1}^{n} nP_i(x) \quad (2.4.2)
\]

where

\[
P_i(x) = y_i \prod_{j=1, j \neq i}^{n} \frac{x - x_j}{x_i - x_j} \quad (2.4.3)
\]

The formula was first published by Waring in 1779, rediscovered by Euler in 1783, and published by Lagrange in 1795. The most common Newton-Cotes formulas include the trapezoid rule,\(^{107}\)

\[
\int_a^b f(x) dx \approx \frac{1}{2} h (f_0 + f_1) - \frac{1}{12} h^3 f''(\xi) \quad (2.4.4)
\]
Simpson’s rule,

$$\int_{a}^{b} f(x)dx \approx \frac{1}{3} h(f_0 + 4f_1 + f_2) - \frac{1}{90} h^5 f^{(4)}(\xi) \quad (2.4.5)$$

Simpson’s 3/8 rule,

$$\int_{a}^{b} f(x)dx \approx \frac{3}{8} h(f_0 + 3f_1 + 3f_2 + f_3) - \frac{3}{80} h^5 f^{(4)}(\xi) \quad (2.4.6)$$

and Boole’s rule,\(^{109}\)

$$\int_{a}^{b} f(x)dx \approx \frac{2}{45} h(7f_0 + 32f_1 + 12f_2 + 32f_3 + 7f_4) - \frac{8}{945} h^7 f^{(6)}(\xi) \quad (2.4.7)$$

Here, \(h = (b - a)/n\) is the interval spacing and \(f_i = f(x_i)\) where \(x_i = a + ih\). This satisfies the constant spacing requirement of these equations. The last term on the right-hand side represents the error of the methods. The exponent of the \(h\) term is the order of accuracy of the method. If the error is \(O(h^n)\), the method is said to be an order \(n\) approximation. This gives an indication of how the error varies with \(h\). For example, the error for a first order method is proportional to \(h\). Many other higher-order Newton-Cotes formulas exist that use additional points and generally follow the analytic expression\(^{110}\)

$$\int_{a}^{b} f(x)dx \approx h \sum_{i=1}^{n} H_{n,i}f_i \quad (2.4.8)$$

where

$$H_{n,r+1} = \frac{(-1)^{n-r}}{r!(n-r)!} \int_{0}^{1} t(t-1) \cdots (t-r+1)(t-r-1) \cdots (t-n)dt \quad (2.4.9)$$
Since these formulations apply to the entire interval \([a, b]\), the interval size is often too large to be very accurate. Therefore, the selected integration technique can be applied to many subintervals on \([a, b]\) and summed together to yield the approximation of the integral over the entire interval \([a, b]\). This is known as a composite Newton-Cotes formula.\textsuperscript{107} To account for data that is not uniformly spaced, the Newton-Cotes formulas can be derived without the assumption of uniform spacing. Following this derivation for Simpson’s rule yields\textsuperscript{1}

\[
\int_a^b f(x)dx \approx \frac{(N-1)/2}{6} \sum_{i=1}^{N} \left( \frac{x_{2i+1} - x_{2i}}{2} \right) \left( 1 + \frac{1}{\Delta} \right) \left[ \Delta (2 - \Delta) f_{2i+1} \right. \\
+ \left( 1 + \Delta \right)^2 f_{2i} + (2\Delta - 1) f_{2i-1} \right] 
\]

where \(N\) is the index of the last odd point and the interval \(\Delta\) is given by

\[
\Delta = \frac{x_{2i} - x_{2i-1}}{x_{2i+1} - x_{2i}} 
\]

If an even number of points are used, the last segment must be added and Eq. 2.4.10 becomes

\[
\int_a^b f(x)dx \approx \frac{(N-1)/2}{6} \sum_{i=1}^{N} \left( \frac{x_{2i+1} - x_{2i}}{2} \right) \left( 1 + \frac{1}{\Delta} \right) \left[ \Delta (2 - \Delta) f_{2i+1} \right. \\
+ \left( 1 + \Delta \right)^2 f_{2i} + (2\Delta - 1) f_{2i-1} \right] \\
+ \frac{x_N - x_{N-1}}{6} \left[ \left( \frac{2 + 3\Delta}{1 + \Delta} \right) f_N \\
+ \left( 3 + \frac{1}{\Delta} \right) f_{N-1} - \left( \frac{1}{\Delta + \Delta^2} \right) f_{N-2} \right] 
\]

\textsuperscript{1}\textsc{D. D. Marshall, Private Communication, April, 2010.}
Chapter 3

Class/Shape Transformation (CST)

The class/shape transformation (CST) method developed by Kulfan\textsuperscript{111–113} is a powerful parameterization technique for modeling both two-dimensional and three-dimensional shapes. It defines a basic profile with a class function and modifies it with a shape function to create the specific shape within its class. The CST method can efficiently model any smooth airfoil in the entire design space. This capability makes it attractive for aerodynamic design and is why CST was selected as the airfoil representation method for this study. Since this study focuses on airfoil design, the discussion of the CST method will be limited to the two-dimensional equations.

A smooth airfoil can be represented with the two-dimensional CST equations by beginning with the general airfoil defined by the class function and modifying it with the shape function to reach the desired airfoil. The only thing that differentiates two curves within the same class in the CST method are an array of coefficients in the defining equation. These coefficients control the deviation from the class function curve. This method of parameterization captures the entire design space of smooth
airfoils and is therefore useful for applications requiring a smooth airfoil.

3.1 General Form

The general form of the CST method uses linear superposition of basis functions to define a shape, much like the formulation of Bézier curves. Each basis function takes the form

$$\zeta_i(\psi) = C_{N_1}^{N_2}(\psi) \cdot S_i(\psi)$$

(3.1.1)

where $C$ and $S$ represent the class function and shape function respectively and will be discussed in this section. The following non-dimensional coordinates are used:

$$\psi = x/c \quad \zeta = y/c$$

(3.1.2)

3.1.1 Class Function

The class function generalizes the shape into a specific class. This defines the initial profile to be somewhat close to the desired shape before any coefficients are used. The class function is expressed as

$$C_{N_2}^{N_1}(\psi) = \psi^{N_1} \cdot (1 - \psi)^{N_2}$$

(3.1.3)

The $\psi$ and $1 - \psi$ terms in the class function cause all curves in the CST method to be defined between $\psi$ values of 0 and 1. Therefore, the coordinates must be scaled by the chord length. The values of $N_1$ and $N_2$ define the specific class into which a shape falls. Many classes exist including an elliptic airfoil ($C_{0.5}^{0.5}0$, biconvex airfoil ($C_{1.0}^{1.0}$), or a Sears-Haack body ($C_{0.75}^{0.75}$). For a general NACA-type airfoil with a round nose and
sharp trailing edge, $N_1$ is 0.5 and $N_2$ is 1.0. This classifies the final shape as being within the airfoil geometry class, which forms the basis of CST airfoil representation. Figure 3.1.1 displays the NACA-type airfoil represented by the class function mirrored about the x-axis to produce a symmetric airfoil.

All airfoils represented by the CST method are derived from the class function airfoil. This is because the superposition of all the unscaled shape functions equals unity everywhere, which is discussed in the following section. Therefore, the airfoil that results from unscaled shape functions is equivalent to that given by the class function.

3.1.2 Shape Function

The shape function modifies the profile obtained from the class function to produce the final curve. The shape function is represented by a Bernstein polynomial, shown below,

$$S_i(\psi) = K^N_i \cdot \psi^i \cdot (1 - \psi)^{N-i}$$

(3.1.4)
where $N$ is the order of the Bernstein polynomial and $K$ is the binomial coefficient, which yields the coefficient of the $x^i$ term in the polynomial expansion of the binomial power $(1 + x)^N$. The binomial coefficient is defined as

$$K^N_i \equiv \binom{n}{i} = \frac{n!}{i!(n-i)!} \quad (3.1.5)$$

### 3.1.3 Curve Representation

Now that the class and shape functions have been defined, they must be superimposed to produce a CST curve. As previously mentioned, a CST curve is represented as a summation of scaled CST basis functions (Eq. 3.1.1), shown below,

$$\zeta(\psi) = \sum_{i=0}^{N} A_i \cdot \zeta_i(\psi) + \psi \cdot \Delta \zeta \quad (3.1.6)$$

where $\Delta \zeta$ is the trailing edge spacing to account for curves that have non-zero trailing edge positions. This is required because all curves defined with the class and shape functions have a trailing edge position of zero. $A$ is the set of Bernstein coefficients to represent a given curve. Each CST basis function is scaled by its corresponding Bernstein coefficient, which determines the specific curve shape. Equation 3.1.6 fully describes any smooth curve given the correct Bernstein coefficients. These coefficients can be optimized to represent a known curve. Since Eq. 3.1.6 models a single curve, an airfoil is represented by two CST curves, one for the upper surface and one for the lower surface. Having an airfoil parameterized by the CST method gives an equation for the upper and lower surfaces. This allows points to be added at desired locations to refine areas such as the leading edge of an airfoil that has high curvature,
or allows the Bernstein coefficients to be used as design variables in an aerodynamic optimization study.

Figure 3.1.2 displays the shape functions (Eq. 3.1.4) and corresponding basis functions (Eq. 3.1.1) for a Bernstein polynomial order of 4. Figure 3.1.2(b) uses \( N1 = 0.5 \) and \( N2 = 1.0 \) in the class function. Therefore, the basis functions fall into the airfoil class and can also be referred to as basis airfoils. The blue line on

![Figure 3.1.2: CST shape functions and basis airfoils](image)

Figure 3.1.2(a) represents the sum of all the shape functions. A property of Bernstein polynomials is that their sum for a given order is unity everywhere. The blue line on Figure 3.1.2(b) again represents the sum of all the basis functions. Since this is equivalent to the sum of the products of the class function and each shape function, the blue line is simply the airfoil represented by the class function.
3.2 Degree Elevation

Any CST curve can be represented exactly with a CST curve of a higher order. Since the class function is independent of the Bernstein coefficients, degree elevation of a CST curve is analogous to degree elevation of a Bernstein polynomial.

A degree $N$ polynomial can be exactly represented by a polynomial of degree $N + 1$ since increasing the degree of the polynomial simply adds an additional degree of freedom. This is expressed as

$$\zeta(\psi) = \sum_{i=0}^{N} A_i \cdot \zeta_i(\psi) + \psi \cdot \Delta \zeta = \sum_{i=0}^{N+1} A^*_i \cdot \zeta_i(\psi) + \psi \cdot \Delta \zeta \quad (3.2.1)$$

where $A^*$ represents the set of Bernstein coefficients for the higher order CST curve. In order to perform this process on a Bernstein polynomial, an algorithm from Rogers\textsuperscript{114} is used. To start, the first and last coefficients must be equal so that the endpoints of the two curves are equal, therefore,

$$A^*_0 = A_0 \quad (3.2.2a)$$

$$A^*_{N+1} = A_N \quad (3.2.2b)$$

The remaining coefficients are related by the following,

$$A^*_i = \alpha_i A_{i-1} + (1 - \alpha_i) A_i \quad \text{for} \quad 1 \leq i \leq N \quad (3.2.3a)$$

where

$$\alpha_i = \frac{i}{N + 1} \quad (3.2.3b)$$
This raises a CST curve by an order of one and exactly represents the lower order CST curve. Iterating on this procedure can raise a CST curve to an arbitrary order.

### 3.3 Modification to Leading Edge Representation

Kulfan\(^1\) proposed a modification to the CST method that models airfoils with significant leading edge camber with fewer Bernstein coefficients than the original CST method. It can also have a slight improvement in modeling airfoils without significant leading edge camber. The modification consists of adding a term to Eq. 3.1.6 that is equivalent to a mirror image of the last basis function. First, the CST basis function (Eq. 3.1.1) must be expanded.

\[
\zeta_i(\psi) = \psi^{N1} \cdot (1 - \psi)^{N2} \cdot \frac{N!}{i!(N - i)!} \cdot \psi^i \cdot (1 - \psi)^{N-i} \tag{3.3.1}
\]

For \(i = N\), this reduces to:

\[
\zeta_N(\psi) = \psi^{N+N1} \cdot (1 - \psi)^{N2} \tag{3.3.2}
\]

This represents the equation of the last basis function. To mirror it, the exponents are simply switched. This yields the additional basis function.

\[
\zeta_{N+1}(\psi) = \psi^{N2} \cdot (1 - \psi)^{N+N1} \tag{3.3.3}
\]

Dividing by the class function (Eq. 3.1.3) yields the additional shape function.

\[
S_{N+1}(\psi) = \psi^{N2-N1} \cdot (1 - \psi)^{N+N1-N2} \tag{3.3.4}
\]

\(^1\)B. M. Kulfan, Private Communication, Nov. 2009.
Adding a scaled version of Eq. 3.3.3 to Eq. 3.1.6 completes the modification.

\[ \zeta(\psi) = \sum_{i=0}^{N} A_i \cdot \zeta_i(\psi) + A_{N+1} \cdot \zeta_{N+1}(\psi) + \psi \cdot \Delta \zeta \]  

(3.3.5)

Figure 3.3.1 repeats Figure 3.1.2 and displays the additional shape function (Eq. 3.3.4) and basis airfoil (Eq. 3.3.3 with \( N1 = 0.5, N1 = 1.0 \)) respectively as the dashed black lines. The additional basis airfoil improves the fit for airfoils with significant leading edge camber.

Figure 3.3.1: CST shape functions and basis airfoils with LE term

### 3.4 Curve Fitting

In order to fit a CST curve to a given curve, a system of equations must be solved to find the Bernstein coefficients that minimize the least squared error between the CST curve and the given curve. To accomplish this, the CST equations must be put into matrix form.
3.4.1 General Form

In order to solve for the Bernstein coefficients to represent a given airfoil, Eq. 3.1.6 must be put in matrix form. This is accomplished by creating a function that isolates \( \vec{A} \) since it contains the Bernstein coefficients to represent a given airfoil. Therefore, all the terms multiplied by \( \vec{A} \) must be combined into a single matrix. These are simply the CST basis functions, which are built up as a CST geometry matrix shown below.

\[
B = \begin{bmatrix}
\zeta_0(\psi_0) & \cdots & \zeta_N(\psi_0) \\
\vdots & \ddots & \vdots \\
\zeta_0(\psi_M) & \cdots & \zeta_N(\psi_M)
\end{bmatrix}
\]  

(3.4.1)

Now the matrix math formulation of Eq. 3.1.6 can easily be created.

\[
\vec{\zeta} = B \cdot \vec{A} + \vec{\psi} \cdot \Delta \zeta
\]  

(3.4.2)

Since \( \vec{A} \) is the only unknown term for a given airfoil, Eq. 3.4.2 can be rearranged to solve for \( \vec{A} \). Since Eq. 3.4.2 is an overdetermined rectangular system unless the order of the Bernstein polynomial equals one less the number of points used to represent the airfoil, the pseudo inverse is used to solve for the Bernstein coefficients. This minimizes the least squared error between the given \( \zeta \) values and the calculated ones.

Figure 3.4.1 displays some examples of parameterized airfoils to display the power of the CST method. The circles show the airfoil coordinates and the lines correspond to the CST airfoil surfaces calculated using optimized Bernstein coefficients.

To evaluate the effectiveness of the class function, the CST method was used to parameterize an RAE 2822 airfoil with and without the class function. The airfoil
Figure 3.4.1: Airfoils parameterized using Bernstein coefficient optimization was parameterized with a CST curve using the airfoil class \((N1 = 0.5, N2 = 1.0)\) and again without the class function \((N1 = 0, N2 = 0)\). Figure 3.4.2 shows the error of a CST curve modeling an RAE 2822 airfoil for varying Bernstein polynomial orders. The error is quantified as the square of the area between the CST curve and the exact airfoil coordinates as discussed in Section 2.4.

Figure 3.4.2: Error for RAE 2822 parameterized with and without class function

The CST curve defined with the airfoil class function clearly has a lower \(A^2\) value for every Bernstein polynomial order. Using the class function to define a profile close to the desired shape always reduces the number of Bernstein coefficients required to
meet a desired tolerance.

### 3.4.2 Modified Form

To fit a CST curve to a given curve using the modified CST formulation (Eq. 3.3.5), the additional basis function (Eq. 3.3.3) must be put into matrix form. This simply consists of creating a vector of $\vec{\zeta}_{N+1}$ by evaluating $\zeta_{N+1}$ for each $\psi$ value.

$$\vec{\zeta}_{N+1} = \begin{bmatrix} \zeta_{N+1}(\psi_0) \\ \vdots \\ \zeta_{N+1}(\psi_M) \end{bmatrix} \quad \text{(3.4.3)}$$

This is then added as an additional column to the CST geometry matrix (Eq. 3.4.1) and solved in the same fashion as Eq. 3.4.2.

$$B = \begin{bmatrix} \zeta_0(\psi_0) & \cdots & \zeta_N(\psi_0) & \zeta_{N+1}(\psi_0) \\ \vdots & \ddots & \vdots & \vdots \\ \zeta_0(\psi_M) & \cdots & \zeta_N(\psi_M) & \zeta_{N+1}(\psi_M) \end{bmatrix} \quad \text{(3.4.4)}$$

Figure 3.4.3 shows the $A^2$ values for an RAE 2822 airfoil parameterized with and without the additional leading edge term. It shows that adding the extra term has little effect on the error, yet still reduces the error slightly for each Bernstein polynomial order.

This is due to the RAE 2822 airfoil having very little leading edge camber. Thickness and camber are calculated as:

$$\zeta_{\text{thickness}}(\psi) = \frac{\zeta_U(\psi) - \zeta_L(\psi)}{2} \quad \text{(3.4.5a)}$$

$$\zeta_{\text{camber}}(\psi) = \frac{\zeta_U(\psi) + \zeta_L(\psi)}{2} \quad \text{(3.4.5b)}$$

Figure 3.4.4 displays the thickness and camber distributions for the RAE 2822 airfoil. It has very little camber until about 40% of the chord.
Figure 3.4.3: Error for RAE 2822 parameterized with and without LE term

Figure 3.4.4: Airfoil, thickness, and camber distributions of an RAE 2822

To show the power of the additional leading edge term, a NACA 6412 was parameterized in the same manner. Figure 3.4.5 shows the thickness and camber distributions for the NACA 6412 airfoil. It has significantly more leading edge camber than the RAE 2822 airfoil.

Figure 3.4.6 shows the $A^2$ values for the NACA 6412 airfoil parameterized with and without the additional leading edge term. It shows that the CST curve with the extra term has a lower $A^2$ value than the CST curve without the extra term for
all Bernstein polynomial orders. Since the modified CST method has an additional coefficient compared to the original formulation for the same Bernstein polynomial order, the error of the CST curve of order $N$ with the additional leading edge term should be compared to the error of the CST curve of order $N + 1$ without the extra term. However, this comparison still shows that the additional leading edge term enhances the CST method for fitting curves with significant leading edge camber.
3.5 Piecewise CST Curves

To represent geometries with corners that would not be well represented with a single CST curve, the curve is segmented at each corner and each segment is fit with a CST curve. This requires a method to detect corners and increases the number of Bernstein coefficients used to represent the curve, but greatly improves the fit.
Chapter 4
Flow Solution Technique

The flow solution technique used for this study is the Reynolds averaged Navier-Stokes (RANS) computational fluid dynamics (CFD) solver FLUENT. Using Reynolds averaged quantities removes the small scale and high-frequency effects of turbulent flows, which is advantageous since these effects cannot be efficiently modeled.

4.1 Governing Equations

FLUENT solves the conservation equations for mass and momentum using the RANS formulations. When a compressible solver is used, it also solves the conservation equation for energy. When a turbulent solver is used, a turbulence model must also be solved. The equations in this section are taken from the FLUENT user manual. Further discussion on the governing equations is located in Appendix B.
4.1.1 Continuity Equation

The continuity equation satisfies conservation of mass and for an inertial (non-accelerating) reference frame can be written as

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{U}) = S_m$$  \hspace{1cm} (4.1.1)

where the source term $S_m$ represents the mass added to the continuous phase from the dispersed second phase, such as due to the vaporization of liquid droplets. This is a general form valid for incompressible and compressible flows.

4.1.2 Momentum Equation

Conservation of momentum in an inertial (non-accelerating) reference frame is described by the momentum equation.

$$\frac{\partial}{\partial t} \left( \rho \vec{U} \right) + \nabla \cdot \left( \rho \vec{U} \vec{U} \right) = -\nabla p + \nabla \cdot (\tilde{T}) + \rho \vec{g} + \vec{F}$$  \hspace{1cm} (4.1.2)

where $p$ is the static pressure, $\tilde{T}$ is the stress tensor, and $\rho \vec{g}$ and $\vec{F}$ are the gravitational and external body forces, respectively. The stress tensor $\tilde{T}$ is defined as

$$\tilde{T} = \mu \left[ \left( \nabla \vec{U} + \nabla \vec{U}^T \right) - \frac{2}{3} \nabla \cdot \vec{U} \right]$$  \hspace{1cm} (4.1.3)

4.1.3 Energy Equation

The energy equation solved by FLUENT is written as

$$\frac{\partial}{\partial t} (\rho E) + \nabla \cdot \left[ \vec{U} (\rho E + p) \right] = \nabla \cdot \left[ k_{eff} \nabla T - \sum_j h_j \vec{J}_j + (\tilde{T}_{eff} \cdot \vec{U}) \right] + S_h$$  \hspace{1cm} (4.1.4)
where \( k_{\text{eff}} \) is the effective thermal conductivity \((k + k_t)\) and \( k_t \) is the turbulent thermal conductivity defined according to the selected turbulence model. The term \( \vec{J}_j \) is the diffusion flux of species \( j \) and \( S_h \) represents any heat due to chemical reactions and any defined sources. The first three terms on the right-hand side define the energy transfer due to conduction, species diffusion, and viscous dissipation, respectively.

The energy term \( E \) is represented as

\[
E = h - \frac{p}{\rho} + \frac{v^2}{2} \quad (4.1.5)
\]

where for ideal gases, the sensible enthalpy \( h \) is defined as

\[
h = \sum_j Y_j h_j \quad (4.1.6)
\]

The term \( Y_j \) represents the mass fraction of species \( j \) and is given as

\[
h_j = \int_{T_{\text{ref}}}^{T} c_{p,j} dT \quad (4.1.7)
\]

where \( T_{\text{ref}} = 298.15 \) K.

### 4.2 Law of the Wall

The law of the wall is an empirically-derived relationship for turbulent flows near a solid boundary. It comes from measurements of external and internal flows that show the streamwise velocity of the flow near the boundary varies logarithmically with the distance from the boundary. Wilcox\textsuperscript{116} states that the mean velocity gradient can be correlated as

\[
\frac{\partial U}{\partial y} = \frac{u_T}{y} F \left( u_T y / \nu \right) \quad (4.2.1)
\]
where \( F(u_\tau y/\nu) \) is assumed to be a universal function and \( u_\tau \) is the friction velocity, which is expressed as
\[
    u_\tau = \sqrt{\frac{\tau_w}{\rho}} \tag{4.2.2}
\]
Experimental data, such as that by Coles and Hirst,\textsuperscript{117} indicates the following
\[
    F(u_\tau y/\nu) \to \frac{1}{\kappa} \quad \text{as} \quad u_\tau y/\nu \to \infty \tag{4.2.3}
\]
where \( \kappa \) is Kármán’s constant. The fact that the function \( F \) approaches a constant follows the notion that viscous effects are negligible far from the surface. The flow approaches freestream conditions. Integrating over \( y \) yields the law of the wall,
\[
    \frac{U}{u_\tau} = \frac{1}{\kappa} \ln \frac{u_\tau y}{\nu} + C \tag{4.2.4}
\]
where \( C \) is simply a dimensionless integration constant. Measurements by Kline et al.\textsuperscript{118} give \( C \approx 5.0 \) for smooth surfaces and \( \kappa \approx 0.41 \) for smooth and rough surfaces.

A common practice is to define a dimensionless velocity \( u^+ \) as
\[
    u^+ = \frac{U}{u_\tau} \tag{4.2.5}
\]
and a dimensionless distance \( y^+ \) as
\[
    y^+ = \frac{u_\tau y}{\nu} \tag{4.2.6}
\]
Therefore, Eq. 4.2.4 can be rewritten as
\[
    u^+ = \frac{1}{\kappa} \ln y^+ + C \tag{4.2.7}
\]
Figure 4.2.1\textsuperscript{116} displays a typical velocity profile for a turbulent boundary layer where the solid line represents the typical velocity profile. It shows that for low $y^+$ values (within the viscous sublayer), the law of the wall is a poor approximation for the actual velocity profile. Therefore, to properly capture viscous effects, a mesh must contain cells within the viscous sublayer. A common goal for this is to achieve a $y^+$ value of 1 throughout the grid.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{velocity_profile.png}
\caption{Velocity profile for a turbulent boundary layer}
\end{figure}

\section{4.3 Turbulence Modeling}

As previously mentioned, when a turbulent solver is used, a turbulence model is introduced. Turbulence models solve for the extra quantities in turbulent flows that arise due to the process of Reynolds averaging.
4.3.1 The $k - \epsilon$ Model

The $k - \epsilon$ model is one of the most popular two-equation models, which allows the turbulent velocity and length scales to be independently determined. It is robust, efficient, and reasonably accurate for a wide range of turbulent flows. The standard model was first proposed by Launder and Spalding.\textsuperscript{119} Improvements have since been made to the model, including the RNG $k - \epsilon$ model\textsuperscript{120} and the realizable $k - \epsilon$ model.\textsuperscript{121} The standard model includes transport equations for the turbulence kinetic energy ($k$) and its dissipation rate ($\epsilon$). These equations are given as

\begin{align*}
\frac{\partial}{\partial t} (\rho k) + \frac{\partial}{\partial x_i} (\rho k u_i) &= \frac{\partial}{\partial x_j} \left[ \left( \mu + \frac{\mu_t}{\sigma_k} \right) \frac{\partial k}{\partial x_j} \right] + G_k + G_b - \rho \epsilon - Y_M + S_k \quad (4.3.1a) \\
\frac{\partial}{\partial t} (\rho \epsilon) + \frac{\partial}{\partial x_i} (\rho \epsilon u_i) &= \frac{\partial}{\partial x_j} \left[ \left( \mu + \frac{\mu_t}{\sigma_\epsilon} \right) \frac{\partial \epsilon}{\partial x_j} \right] + C_1 \frac{\epsilon}{k} (G_k + C_3 G_b) \\
&\quad - C_2 \rho \frac{\epsilon^2}{k} + S_\epsilon \quad (4.3.1b)
\end{align*}

where $G_k$ is the generation of turbulence kinetic energy due to mean velocity gradients, defined as

\begin{equation}
G_k = -\rho u_i u_j \frac{\partial u_j}{\partial x_i} \quad (4.3.2)
\end{equation}

The production of turbulence kinetic energy can also be written as

\begin{equation}
G_k = \mu_t S^2 \quad (4.3.3)
\end{equation}

where $S$ is the modulus of the mean rate-of-strain tensor $S_{ij}$

\begin{equation}
S = \sqrt{2S_{ij}S_{ij}} \quad (4.3.4)
\end{equation}
In Eq. 4.3.1, $G_b$ represents the generation of turbulence kinetic energy due to buoyancy, $Y_M$ is the contribution of the fluctuating dilation in compressible turbulence to the overall dissipation rate and $\mu_t$ is the turbulent or eddy viscosity. The terms $C_{1\epsilon}, C_{2\epsilon},$ and $C_{3\epsilon}$ are simply model constants and $\sigma_k$ and $\sigma_\epsilon$ are the turbulent Prandtl numbers for $k$ and $\epsilon$, respectively. Lastly, $S_k$ and $S_\epsilon$ are user-defined source terms for $k$ and $\epsilon$, respectively.

4.3.2 The $k-\omega$ Model

The $k-\omega$ turbulence model developed by Wilcox\cite{wilcox1998} is well validated and is the model of choice for the sublayer of the boundary layer. It contains modifications for low-Reynolds-number effects, compressibility, and shear flow spreading. It provides accurate free shear spreading rates for far wakes, mixing layers, and plane, round, and radial jets. The original model has since been modified by Wilcox\cite{wilcox1999} to include an additional closure coefficient and a modification to the dependence of eddy viscosity on turbulence properties. The resulting model better predicts boundary layers and free shear flows. The turbulence properties have very little sensitivity to the finite freestream boundary conditions. The model includes transport equations for the turbulence kinetic energy ($k$) and the specific dissipation rate ($\omega$), given as

$$\frac{\partial}{\partial t} (\rho k) + \frac{\partial}{\partial x_i} (\rho k u_i) = \frac{\partial}{\partial x_j} \left( \Gamma_k \frac{\partial k}{\partial x_j} \right) + G_k - Y_k + S_k \quad (4.3.5a)$$

$$\frac{\partial}{\partial t} (\rho \omega) + \frac{\partial}{\partial x_i} (\rho \omega u_i) = \frac{\partial}{\partial x_j} \left( \Gamma_\omega \frac{\partial \omega}{\partial x_j} \right) + G_\omega - Y_\omega + S_\omega \quad (4.3.5b)$$
where \( G_\omega \) represents the generation of \( \omega \), and \( \Gamma_k \) and \( \Gamma_\omega \) are the effective diffusivity of \( k \) and \( \omega \), respectively. The terms \( Y_k \) and \( Y_\omega \) represent the dissipation of \( k \) and \( \omega \) due to turbulence. Lastly, the \( S_\omega \) is a user-defined source term for \( \omega \). Any terms in common with the standard \( k - \epsilon \) model have the same definition.

### 4.3.3 The Shear Stress Transport (SST) \( k - \omega \) Model

Menter\(^{124}\) developed the shear-stress transport (SST) \( k - \omega \) model to blend the robust and accurate nature of the \( k - \omega \) model in the near-wall region with the freestream independence of the \( k - \epsilon \) model in the far field. This is achieved by transforming the \( k - \epsilon \) model into a \( k - \omega \) formulation, multiplying both the \( k - \omega \) and transformed \( k - \epsilon \) models by a blending function, and adding the models together. The blending function activates the \( k - \omega \) model in the near-wall region and activates the \( k - \epsilon \) model away from the surface. This formulation is more accurate than the standard \( k - \omega \) model and reliable for a wider class of flows including adverse pressure gradient flows, flows over airfoils, and flows including transonic shock waves. Since a transonic case is included in the selected design cases, the SST \( k - \omega \) model is used for all flow simulations. The transport equations for the SST \( k - \omega \) model are similar to those of the standard \( k - \omega \) model. They are expressed as

\[
\frac{\partial}{\partial t} (\rho k) + \frac{\partial}{\partial x_i} (\rho k u_i) = \frac{\partial}{\partial x_j} \left( \Gamma_k \frac{\partial k}{\partial x_j} \right) + \tilde{G}_k - Y_k + S_k \tag{4.3.6a}
\]

\[
\frac{\partial}{\partial t} (\rho \omega) + \frac{\partial}{\partial x_i} (\rho \omega u_i) = \frac{\partial}{\partial x_j} \left( \Gamma_\omega \frac{\partial \omega}{\partial x_j} \right) + G_\omega - Y_\omega + D_\omega + S_\omega \tag{4.3.6b}
\]
where $\tilde{G}_k$ represents the generation of turbulence kinetic energy due to mean velocity gradients. The remaining terms use the same definitions of the standard $k-\omega$ model.

### 4.3.4 Boussinesq Approximation

When the RANS equations are used, the Reynolds stresses must be appropriately modeled. The $k-\epsilon$ and $k-\omega$ turbulence models, among others, employ the Boussinesq approximation, which relates the Reynolds stresses to the mean velocity gradients as

$$-\rho u'_i u'_j = 2\mu_T S_{ij} - \frac{2}{3} \delta_{ij} \left( \mu_T \frac{\partial u_k}{\partial x_k} + \rho \tilde{k} \right)$$

(4.3.7)

where $S_{ij}$ represents the mean rate-of-strain tensor, defined as

$$S_{ij} = \frac{1}{2} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right)$$

(4.3.8)

The Boussinesq approximation makes the assumption that $\mu_t$ is an isotropic scalar quantity, which is not strictly true. However, this is a computationally efficient formulation and works well for a variety of flows.
Chapter 5

Flow Solution Automation

In order to conduct iterative inverse airfoil design, the flow solution techniques must be automated in order to avoid any user interaction. The programming language MATLAB is used to generate journal files that are read by the flow solution programs. A journal file simply contains commands that a program reads and carries out. MATLAB is then used to start the automation by calling the software packages with their respective journal files.

5.1 CFD Automation - Conventional Airfoils

In order to conduct CFD simulations in an iterative process with no user interaction, all steps must be automated, which is much more complicated than automating a panel code such as XFOIL.\textsuperscript{125} The steps include pre-processing, solving, and post-processing. GAMBIT\textsuperscript{126} is used as the pre-processing software to create the grids and FLUENT\textsuperscript{115} is used as the solver and post-processing software.
5.1.1 Operating Conditions

The operating conditions \((T, P, \rho, \mu, k)\) are found by one of two ways based on the user inputs. The user must specify a Mach number, an angle of attack, and either an altitude or a Reynolds number. The user may also specify a temperature if a desired Reynolds number is given. If the user specifies a Mach number and altitude, the temperature, pressure, and density are calculated at that altitude based on the standard atmosphere equations. If the user specifies a Mach number and Reynolds number, the altitude is found that produces the temperature, pressure, and density that yield the desired Reynolds number at the specified Mach number. This is done with the MATLAB function \texttt{fminbnd}, which is a 1-D minimizer on a fixed interval. It is used to minimize the square of the difference between the desired Reynolds number and that calculated using the values obtained from the standard atmosphere calculator at the altitude selected by the optimizer. Altitude is bounded by the limits of the standard atmosphere calculator in order to obtain valid results.

If the user specifies a Mach number, Reynolds number, and temperature, enough is known to calculate the remaining unknowns. Once the temperature is known, the dynamic viscosity and thermal conductivity are calculated using Sutherland’s formulas:

\[ \mu = \mu_0 \left( \frac{T}{T_0} \right)^{3/2} \frac{T_0 + S_\mu}{T + S_\mu} \]  \hspace{1cm} (5.1.1a)

\[ k = k_0 \left( \frac{T}{T_0} \right)^{3/2} \frac{T_0 + S_k}{T + S_k} \]  \hspace{1cm} (5.1.1b)
where \( \mu_0 \) is the reference viscosity, \( k_0 \) is the reference thermal conductivity, \( T_0 \) is the reference temperature, and \( S_\mu \) and \( S_k \) are Sutherland’s constants for viscosity and thermal conductivity respectively for a given gas. For air at moderate temperatures, \( \mu_0 = 1.716 \times 10^{-5} \text{ kg/(m \cdot s)} \), \( k_0 = 0.0241 \text{ W/(m \cdot K)} \), \( T_0 = 273.11 \text{ K} \), \( S_\mu = 110.56 \text{ K} \), and \( S_k = 194 \text{ K} \). The specified temperature also allows the speed of sound to be calculated using the relation

\[
a = \sqrt{\gamma RT}
\]

where \( \gamma \) is the ratio of specific heats and \( R \) is the gas constant, which are 1.4 and approximately \( 287 \text{ J/(kg-K)} \) for air respectively. This yields the flow velocity using the specified Mach number.

\[
U = Ma
\]

If a Reynolds number is provided, the density is calculated using the airfoil chord length, the specified Reynolds number, and the definition of Reynolds number.

\[
Re = \frac{\rho Uc}{\mu}
\]

Finally the pressure is calculated with the ideal gas equation of state.

\[
P = \rho RT
\]

### 5.1.2 Pre-Processing

MATLAB is used to write journal files containing commands that GAMBIT reads and executes to produce a structured C-grid over an arbitrary airfoil geometry. Figure
5.1.1(a) shows a C-grid generated with the GAMBIT automation over an RAE 2822 airfoil with 25,000 cells.

The airfoil is split slightly downstream of the leading edge on both the upper and lower surfaces. This permits better node clustering around the leading edge to better capture the suction peak of the airfoil. The same node spacing is used at the end of the leading edge segment and the beginning of the main segment to provide a satisfactory transition between the node spacing used on each segment. Figure 5.1.1(b) shows the same grid as Figure 5.1.1(a) zoomed in around the airfoil to better

Figure 5.1.1: Structured C-grid over an RAE 2822 airfoil - 25,000 cells
show the leading edge node clustering.

The user has control of the number of nodes on each airfoil segment (leading edge and main body for the upper and lower surfaces), the height and length of the wake region of the mesh, the number of nodes on each wake segment, as well as the desired wall $y^+$ value, which is used to calculate the height of the first cell in the boundary layer. Figure 5.1.1(c) shows the grid around the leading edge of the airfoil to show the cell growth away from the wall. This was generated using a $y^+$ value of 1 to ensure cells are within the viscous sublayer. The $y^+$ calculation is based on the method used by Jones,\textsuperscript{128} which is discussed in Appendix C.

### 5.1.3 Solution Process

The solution automation consists of three steps: case setup, first order iteration, and second order iteration. The case setup stage simply sets the solver settings, operating conditions, boundary conditions, turbulence model, and initializes the flow field. An implicit density-based solver is used where density is calculated with the ideal gas law and viscosity with the three-coefficient Sutherland’s law. Since density is calculated with the ideal gas law, all domain boundaries are given pressure far-field boundary conditions. The $k - \omega$ SST turbulence model is selected for its superiority over both the standard $k - \epsilon$ and standard $k - \omega$ turbulence models.\textsuperscript{124}

The first order stage sets the solution discretization to first order upwind and solves for about 100 iterations. This is intended to provide a base solution before switching to second order upwind to enhance stability. Beginning the simulation in
second order upwind discretization could potentially cause the solution to diverge. After completing the first order solution, the solution discretization is switched to second order upwind. While in second order discretization, the Courant number is gradually increased up to 100 in order to obtain rapid convergence.

The Courant number plays an important role in the Courant-Friedrichs-Lewy (CFL) condition,\textsuperscript{129} which is a necessary condition for numerically solving partial differential equations (PDEs) with time-marching schemes. It is commonly prescribed for the advection terms in the PDEs being solved. The CFL condition for two dimensions is given by

\begin{equation}
\frac{U \cdot \Delta t}{\Delta x} < C
\end{equation}

where $U$ is the velocity, $\Delta t$ is the time step, $\Delta x$ is the length interval, and the constant $C$ is dependent on the PDE being solved. The Courant number ($CFL$) is simply the left side of Eq. 5.1.6.

\begin{equation}
CFL = \frac{U \cdot \Delta t}{\Delta x}
\end{equation}

Increasing the Courant number throughout the solution yields the same lift and drag coefficients as using the default Courant number of 5 for the entire simulation. However, increasing the Courant number converges in significantly fewer iterations than using the default value. This is discussed further in Section 5.1.5.
5.1.4 Post-Processing

The post-processing automation simply consists of writing data to files to determine how to change the airfoil shape for the subsequent design iteration. A file containing the pressure coefficient distribution over the airfoil surface is written in order to compare the current pressure distribution to the target pressure distribution. Also, the lift and drag coefficients are written to a file to compare to the target values.

The automation also writes an interpolation file. This allows the solution to be interpolated onto the grid of another airfoil. This results in accelerated convergence for the same flight conditions because the simulation begins with a converged solution similar to its end result. This was included as an added feature of the automation, but is not used in the inverse design process because it can be slightly unstable. High gradients can appear in the solution if the new boundary is inside the old boundary because flow information is not available within the old boundary. To account for this, the under-relaxation factors and Courant number must be reduced to help prevent divergence. Solution interpolation can be used to good effect when the user is in control, but can be unpredictable when used in an automated process, particularly when using high Courant numbers. This drawback is the reason solution interpolation is not implemented in the inverse design process for conventional airfoils, particularly given the computational efficiency of increasing the Courant number.
5.1.5 Efficiency of Courant Number Ramping

To evaluate the efficiency of increasing the Courant number, the FLUENT automation was used to simulate a case using both an increasing Courant number and the default value of 5. Figure 5.1.2 shows the lift and drag coefficient histories for an RAE 2822 airfoil in transonic flow using Courant number ramping and the default value of 5. The operating conditions are a freestream Mach number of 0.729, angle of attack of $2.3^\circ$, and Reynolds number of $6.5 \times 10^6$ based on a chord length of one foot and a static temperature of 460 Rankine. This case was selected because it was used to validate the FLUENT automation, which is discussed in Section 5.3. The case simulated with the Courant number set to 5 was run for 10,000 iterations and the case simulated with Courant number ramping was run for 1,000 iterations. The iteration number in the figure is divided by the maximum iteration number to show both on the same scale. The lift and drag in each case converge to roughly the same

![Figure 5.1.2: Lift and drag convergence histories showing Courant number effects](image)

63
values at approximately an $N/N_{max}$ value of 0.4. However, this corresponds to 10 times more iterations to converge using a constant Courant number of 5 than using Courant number ramping. This results in significantly reduced runtime for each flow solution, which saves a substantial amount of computational time when used in an iterative scheme such as inverse design. While this is not a universal practice and will cause the solution to diverge for more complicated flows, it has proven robust enough for flow over conventional airfoils.

5.2 CFD Automation - CC Airfoils

To adapt the CFD automation to circulation control airfoils, the gridding process must be altered to handle the slot. Figure 5.2.1 shows the general aviation circulation control (GACC) airfoil developed by Jones et al.\textsuperscript{130} with an attached flap. From far away, the GACC airfoil resembles a conventional airfoil with a flap.

![GACC airfoil geometry with blown flap](image)

Figure 5.2.1: GACC airfoil geometry with blown flap

However, a closer look at the trailing edge in Figure 5.2.2 reveals a small slot at the beginning of the flap. High velocity flow is blown out of the slot and over the curved flap. This takes advantage of the Coandă effect, which is the tendency
of a fluid to stay attached to an adjacent curved surface.\textsuperscript{131} Early development into circulation control technology consisted of an add-on device\textsuperscript{132} to convert a wing flap’s sharp trailing edge into a Coandă surface, which is a rounded trailing edge. The original GACC airfoil implemented such a trailing edge. Englar\textsuperscript{133} notes that this configuration experiences high drag at cruise. To remedy this, Englar and Huson\textsuperscript{134} and Englar et al.\textsuperscript{135} implemented a dual radius curved flap with a sharp trailing edge, similar to that used on the version of the GACC airfoil shown here. The high speed flow expelled from the slot remains attached over the flap and dramatically increases lift. This also has the effect of moving the stagnation point on the leading edge and the flow separation point on the trailing edge. The extent of this movement is primarily a function of the jet momentum coefficient $C_\mu$, which is a measure of the jet momentum relative to the freestream momentum. The jet momentum coefficient is commonly expressed as

\[
C_\mu = \frac{\dot{m}U_j}{q_\infty S} = 2 \left( \frac{h_j w_j}{S} \right) \left( \frac{\rho_j}{\rho_\infty} \right) \left( \frac{U_j}{U_\infty} \right)^2
\]  

(5.2.1)
where $U_j$ is the jet velocity out of the slot, $\rho_j$ is the density at the jet, and $h_j$ and $w_j$ are the jet height and width, respectively.

### 5.2.1 Pre-Processing

The flow aft of the slot is much more complicated than the flow near the trailing edge of a conventional airfoil. This requires grids over CC airfoils to include significantly more cells, particularly aft of the slot in order to resolve the flow around the flap, around the leading edge to capture the stagnation point movement, and in the wake region to capture recirculation. However, not as many cells are required over the main portion of the lower surface since the flow is near stagnation conditions over the majority of the lower surface.

The gridding automation is based on the structured grid developed by Storm and Marshall\textsuperscript{136} for studying the $v^2-f$ turbulence model applied to circulation control. Great care was taken to ensure gridline orthogonality to reduce numerical error. A $y^+$ value of 1 is used along the airfoil to ensure that the first cell is within the laminar sublayer (generally $y^+ < 5$). Since the jet velocity is not known prior to the simulation, a $y^+$ value corresponding to a Mach number of 1 is used on the flap. The grid also has sufficient leading edge discretization to capture the stagnation point movement to accurately predict lift and enough resolution in the wake region to capture recirculation. The grid contains 166,250 cells and is shown in Figure 5.2.3. The unique farfield configuration shown in Figure 5.2.3(a) is caused by a combination of the grid orthogonality requirement and the large flap deflection. Figures 5.2.3(b)-
5.2.3(d) display the cell clustering around the airfoil and flap. To accomplish this, a boundary normal to the airfoil and flap is created and a small structured mesh is generated within this region. This mesh is joined with a structured mesh generated within the outer region to complete the CC airfoil mesh generation.

![Structured grid over GACC airfoil - 166,250 cells](image)

(a) Domain  
(b) Airfoil  
(c) Flap  
(d) Slot

**Figure 5.2.3: Structured grid over GACC airfoil - 166,250 cells**

### 5.2.2 Solution Process

The FLUENT automation for CC airfoils is identical to that for conventional airfoils with the exception of setting the boundary condition at the slot. The slot
is set as a mass flow inlet due to the mass flow rate $\dot{m}$ term in calculating the jet momentum coefficient (Eq. 5.2.1). The difficulty with this boundary condition is that for a mass flow inlet, the user has control of the mass flow rate and the temperature at the boundary. The velocity is unknown until the simulation is conducted. Therefore, $C_\mu$ is not known at the beginning of a given simulation. In order to estimate the $\dot{m}$ required to yield a desired $C_\mu$, the definition of mass flow rate is used.

$$\dot{m} = \rho AU$$  \hspace{1cm} (5.2.2)

Rearranging for velocity and multiplying by $\dot{m}$ yields

$$\dot{m}U = \frac{\dot{m}^2}{\rho A}$$  \hspace{1cm} (5.2.3)

Using the ideal gas law for density yields

$$\dot{m}U = \frac{\dot{m}^2RT}{PA}$$  \hspace{1cm} (5.2.4)

The temperature and pressure are assumed to be at freestream values for the purposes of estimating the required $\dot{m}$ used in the slot boundary condition. Also, the slot area $A_j$ can be replaced with $bh_j$ where the jet width is assumed to equal the span length and $h_j$ is the height of the jet.

$$\dot{m}U_j = \frac{\dot{m}^2RT_\infty}{P_\infty bh_j}$$  \hspace{1cm} (5.2.5)

Also, the dynamic pressure can be rearranged using the ideal gas law and the
relation for the speed of sound.

\[ q_\infty = \frac{1}{2} \rho_\infty U_\infty^2 \]
\[ = \frac{1}{2} \frac{P_\infty}{RT_\infty} (M_\infty a_\infty)^2 \]
\[ = \frac{1}{2} \frac{P_\infty}{RT_\infty} \left( M_\infty \sqrt{\gamma RT_\infty} \right)^2 \]
\[ = \frac{1}{2} \gamma P_\infty M_\infty^2 \] (5.2.6)

These relations can be substituted into Eq. 5.2.1 and rearranged to solve for \( \dot{m} \) where the reference area \( S = bc \).

\[ \dot{m} = M_\infty P_\infty b \sqrt{\frac{C_\mu \gamma ch_j}{2RT_\infty}} \] (5.2.7)

Using the freestream conditions as the jet values is a rough estimation, but gets close to the desired \( C_\mu \) without knowing the jet velocity at the beginning of the solution. For example, using this method to set \( \dot{m}_j \) for the case simulated in Section 5.4 to validate the CC solution process yielded a \( C_\mu \) of 0.0982 when the desired value was 0.1. Once the jet velocity is obtained from the CFD solution, the mass flow rate can be updated to get closer to the desired \( C_\mu \). From experience, this typically only works for low \( C_\mu \) values. For higher \( C_\mu \) values, the blowing cannot be modeled solely with the slot; the plenum must be accounted for as well. Since only low \( C_\mu \) values were tested, the plenum was ignored.

Since the flow over a CC airfoil is much more complex than the flow over a conventional airfoil, Courant number ramping cannot be utilized. To reduce the solution time at each iteration of the inverse design process, the solution from the previous
iteration is interpolated onto the grid of the current iteration. This is much more computationally efficient, particularly since the flap geometry and blowing coefficient are held constant throughout the design. The flow over the flap is already developed at the beginning of each iteration, which comprises a significant portion of the computational burden. The only changes in the flow are due to the changes in the main body forward of the flap and slot.

5.3 CFD Verification and Validation - Conventional Airfoils

Since CFD simulations of the “same” problem can result in widely different answers due to differences in grids and solution techniques, it is important to validate the methods used. This includes performing a grid convergence study and comparing the methods used against experimental data and/or previously validated CFD studies. A grid convergence study displays that the CFD solution is not a function of the number of cells in the grid and is therefore converged. Adding additional cells will not significantly change the answer. Also, comparing against experimental data and other CFD studies shows that the solution is “correct.”

5.3.1 Verification - Grid Convergence Study

When doing CFD simulations, one must ensure that the grids being used have satisfactory refinement. It is important that the simulations conducted are in the asymptotic region, meaning that the solutions are independent of the number of cells
in the grid. If this is not done, the methods used come into question because it is unclear if adding additional cells to the grid will significantly change the solution. The necessary grid refinement is evaluated by performing a grid convergence study. One well-tested and widely accepted method for performing a grid convergence study is the grid convergence index (GCI),\textsuperscript{137} which is a measure of the discretization error. The solution is then said to be confident within $\pm$GCI. The method also calculates the extrapolated solution if an infinite number of cells were used. The techniques used to calculate the grid convergence index and the extrapolated solution are discussed in Appendix D.

The FLUENT and GAMBIT automations were used to simulate transonic flow over an RAE 2822 airfoil using several grids with different cell counts in order to determine the coarsest grid that could safely be used in the inverse design process because fewer cells directly translates to reduced runtime. The cell count was roughly doubled for each grid refinement. The operating conditions used are a freestream Mach number of 0.729, angle of attack of 2.3°, and Reynolds number of $6.5 \times 10^6$ based on a chord length of one foot and a static temperature of 460 Rankine. This case was selected because experimental data\textsuperscript{138} and other CFD solutions\textsuperscript{139} exist for this case, which are used to further validate the CFD automation in Section 5.3.2. Figure 5.3.1 shows the lift and drag coefficient convergence with cell count. The circles represent the solution obtained from the CFD simulation and the vertical bars surrounding each solution correspond to the GCI for each solution. The horizontal
line represents the extrapolated solution if an infinite number of cells were used. Both the lift and drag coefficients converge fairly quickly as evident by the rapid decrease of the GCI as the cell count increases.

![Graphs showing lift and drag coefficients convergence](image)

**Figure 5.3.1: Lift and drag coefficient convergence with cell count**

Tables 5.3.1 and 5.3.2 display the results of the grid convergence study with GCI values for both the lift and drag coefficients. The subscript $\infty$ represents the extrapolated value and $p$ is the order of accuracy the data appears to exhibit. The $GCI(\%)$ value is the percentage the $GCI$ is of the solution. The error is the percent difference from the extrapolated solution. The high $p$ value for $C_d$ is caused by the large change in the solution between the two coarsest grids. The grid with 25,000 cells is the coarsest grid that has $GCI$ values less than one percent of the solution and has solution values less than one percent from the extrapolated values. Therefore, it was selected as the grid to use for the design of conventional airfoils because it represents a good compromise between computational efficiency and numerical accuracy.
Table 5.3.1: Lift coefficient convergence ($C_{l,\infty}$ = 0.7026, $p$ = 2.02)

<table>
<thead>
<tr>
<th>$N$</th>
<th>$C_l$</th>
<th>$GCI$</th>
<th>$GCI(%)$</th>
<th>$C_{l,\infty}$ Error(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>200,000</td>
<td>0.7007</td>
<td>0.0025</td>
<td>0.35</td>
<td>-0.2804</td>
</tr>
<tr>
<td>100,000</td>
<td>0.7021</td>
<td>0.0007</td>
<td>0.10</td>
<td>-0.0769</td>
</tr>
<tr>
<td>52,500</td>
<td>0.7021</td>
<td>0.0007</td>
<td>0.10</td>
<td>-0.0801</td>
</tr>
<tr>
<td>25,000</td>
<td>0.6988</td>
<td>0.0048</td>
<td>0.68</td>
<td>-0.5418</td>
</tr>
<tr>
<td>12,528</td>
<td>0.6929</td>
<td>0.0121</td>
<td>1.75</td>
<td>-1.3810</td>
</tr>
<tr>
<td>6,256</td>
<td>0.6859</td>
<td>0.0209</td>
<td>3.05</td>
<td>-2.3837</td>
</tr>
</tbody>
</table>

Table 5.3.2: Drag coefficient convergence ($C_{d,\infty}$ = 0.0122, $p$ = 6.74)

<table>
<thead>
<tr>
<th>$N$</th>
<th>$C_d$</th>
<th>$GCI \times 10^3$</th>
<th>$GCI(%)$</th>
<th>$C_{d,\infty}$ Error(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>200,000</td>
<td>0.0121</td>
<td>0.6073</td>
<td>0.50</td>
<td>-0.40</td>
</tr>
<tr>
<td>100,000</td>
<td>0.0122</td>
<td>0.0102</td>
<td>0.50</td>
<td>-0.07</td>
</tr>
<tr>
<td>52,500</td>
<td>0.0122</td>
<td>0.0341</td>
<td>0.28</td>
<td>0.22</td>
</tr>
<tr>
<td>25,000</td>
<td>0.0122</td>
<td>0.0676</td>
<td>0.55</td>
<td>0.44</td>
</tr>
<tr>
<td>12,528</td>
<td>0.0123</td>
<td>0.1893</td>
<td>1.54</td>
<td>1.24</td>
</tr>
<tr>
<td>6,256</td>
<td>0.0138</td>
<td>2.0738</td>
<td>15.00</td>
<td>13.64</td>
</tr>
</tbody>
</table>

5.3.2 Validation - Comparison to Experimental Data

Once it is known that the CFD solution does not depend on the number of cells in the grid, the automation must then be compared to experimental data and/or previously validated CFD studies to ensure the “correct” answer is obtained. Slater\textsuperscript{139} conducted a study of an RAE 2822 airfoil in transonic flow using the codes NPARC and WIND. The solutions were compared against experimental data obtained by Cook et al.\textsuperscript{138} The study was conducted using a freestream Mach number of 0.729, angle of attack of 2.3°, and Reynolds number of $6.5 \times 10^6$ based on a chord length of one foot and a static temperature of 460 Rankine. The grid used for Slater’s CFD simulations was a 369 x 65 C-grid, shown in Figure 5.3.2(a).

The experimental pressure distribution is shown in Figure 5.3.3 along with the
pressure distributions obtained from NPARC, WIND, and the FLUENT automation developed for the present study. Slater notes that no sensitivity studies were conducted, so it is unclear if the solution is independent of the grid resolution. However, the NPARC and WIND pressure distributions are fairly close to the experimental pressure distribution using a grid with only 23,985 cells. Both nearly match the pressures on the lower surface and aft shock region. NPARC captures the suction peak well while WIND slightly over predicts it. Also, both under predict the shock location, but WIND predicts the it better than NPARC. Both codes also under predict
the pressures forward of the shock by a small amount. The FLUENT automation was used to simulate the same case in order to further validate the automation. FLUENT predicts the suction peak approximately as well as NPARC and the shock location predicted by FLUENT is on par with that of WIND. FLUENT slightly under predicts the pressures forward of the shock, but they are roughly as close to the experimental data as those predicted by NPARC and WIND. The pressures over the remainder of the airfoil are roughly the same for all three codes. This correlation with experimental data and two validated CFD codes suggests that the FLUENT automation can be used with confidence to predict the flow over an airfoil.
5.4 CFD Validation - CC Airfoils

In the interest of computational efficiency, particularly in an iterative inverse design process, an attempt was made to reduce the cell count of the grid by Storm and Marshall without sacrificing numerical accuracy. A grid was developed using the same farfield configuration, but with only 166,250 cells, a reduction of 143,190 cells or 46.3%. The two grids were simulated using FLUENT at $M_\infty = 0.1$, $\alpha = 0^\circ$, and $C_\mu = 0.1$. Table 5.4.1 displays the lift and drag coefficients for both simulations, as well as the percent difference from the grid by Storm and Marshall. Both the lift and drag coefficients are within 2% of the values obtained using the grid by Storm and Marshall. For a 46.3% decrease in cell count, this is a minimal difference in lift and drag. However, the significantly reduced cell count contributes to greatly reduced simulation times.

Table 5.4.1: Comparison of CC airfoil grids

<table>
<thead>
<tr>
<th>Grid</th>
<th>Cells</th>
<th>$C_l$</th>
<th>$C_d$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storm and Marshall</td>
<td>309,440</td>
<td>4.42</td>
<td>0.115</td>
</tr>
<tr>
<td>Lane and Marshall</td>
<td>166,250</td>
<td>4.48</td>
<td>0.117</td>
</tr>
<tr>
<td>Difference (%)</td>
<td>-46.3</td>
<td>1.38</td>
<td>1.43</td>
</tr>
</tbody>
</table>
Figure 5.4.1 displays the pressure distribution over the GACC airfoil using the present grid and the grid by Storm and Marshall. Comparing the $C_p$ values between the two grids shows that the present grid captures the flow over the GACC airfoil well including the pressure spike aft of the slot.

![Pressure distribution for the GACC airfoil - $C_p$ = 0.1](image)

*Figure 5.4.1: Pressure distribution for the GACC airfoil - $C_p$ = 0.1*
Chapter 6

Results

6.1 Subsonic Design

To first test the methods, a simple subsonic case is selected. The target airfoil for the subsonic design case is an RAE 2822 and the initial airfoil is a NACA 0012. The operating conditions are $M_\infty = 0.3$, $Re = 1.0 \times 10^7$, and $\alpha = 4.0^\circ$. The target airfoil is modeled with 5th order CST curves. Since the airfoil is modeled with a CST curve throughout the design process, using a parameterized airfoil as the target gives the potential for the designed airfoil to exactly represent the target airfoil.

6.1.1 Normal Perturbation

For the normal perturbation method, a scale factor of 0.04 is used and the airfoil is initially fit with 2nd order CST curves. If the area between the current and previous airfoils (Eq. 2.4.1) is sufficiently small, the order is increased by one up to a maximum order of 5. Beginning with high orders result in unrealistic airfoils when residuals are high. The increased degrees of freedom compared to low-ordered curves yield
undesirable bumps in the airfoil surface. Once the residuals are small, the airfoil can be parameterized with higher order curves. The inverse design is executed for 20 iterations to obtain a good representation of its behavior.

Figure 6.1.1 shows the pressure distribution and airfoil throughout the first 6 iterations of the inverse design process. The airfoil and pressure distribution very closely match the target airfoil and pressures after just 6 iterations. For a method this straightforward and simple to implement, these are encouraging results.

Figure 6.1.2 shows the pressure error at the same iterations as Figure 6.1.1. Here, negative $x/c$ values correspond to the lower surface. There is a spike in the pressure error at the leading edge that is not eliminated. This is a problem present in many inverse design techniques. The residuals over the remainder of the airfoil dampen out in relation to the leading edge. The pressure error at the leading edge is about 0.25 after 6 iterations and negligible over the rest of the airfoil.

Figure 6.1.3 shows the airfoil coordinate residuals at the same 4 iterations. Again, negative $x/c$ values correspond to the lower surface. The geometry error keeps a similar distribution throughout the design. However, the magnitude decreases throughout the design. By iteration 6 all geometry residuals are within $2 \times 10^{-3}$. 

79
Figure 6.1.1: Subsonic design results - normal perturbation
Figure 6.1.2: Subsonic pressure residuals - normal perturbation

Figure 6.1.3: Subsonic airfoil residuals - normal perturbation
Figure 6.1.4 shows the pressure and airfoil geometry error convergence history. The error decreases very rapidly for the first few iterations. However, after about iteration 6, the error ceases to decrease. This could be a byproduct of using a non-optimal scale factor. This would infer that the minimum error shown is the best the scale factor used can achieve for the given flight conditions.

Figure 6.1.4: Subsonic design error - normal perturbation
Figure 6.1.5 shows the upper and lower surface Bernstein coefficients throughout the design. The coefficients vary during the first 5 or 6 iterations as the airfoil undergoes the majority of the shape change. After this point, the Bernstein coefficients do not change much as the airfoil shape is mostly converged. This is also shown by the minimal change in the pressure and geometry error after 5 or 6 iterations in Figure 6.1.4. The circles represent the Bernstein coefficients of the 5th order CST curves representing the RAE 2822 airfoil. Therefore, these are the target values. The final Bernstein coefficients are slightly off the desired values, which can be caused by a non-optimal scale factor.
6.1.2 Pressure Gradients

For the pressure gradient method, the airfoil is modeled with 5th order CST curves at each design iteration to correspond to the order used for the target airfoil. In order to calculate the pressure gradients with respect to the Bernstein coefficients, each coefficient is perturbed by 1% of its value.

Figure 6.1.6 displays the inverse design results for the pressure gradient method at the same iterations presented for the normal perturbation method. The pressures and airfoil are very close after just 2 iterations and nearly exact after 6 iterations.

Figure 6.1.7 shows the pressure error at the same iterations as Figure 6.1.6. Again, there is a spike in the pressure error at the leading edge that is not eliminated throughout the design. The residuals over the remainder of the airfoil dampen out in relation to the leading edge. The pressure error at the leading edge is less than 0.05 after 6 iterations and negligible over the rest of the airfoil. This is significantly less than the pressure error for the normal perturbation method, but required more flow solutions.

Figure 6.1.8 shows the airfoil coordinate residuals throughout the design. The geometry error keeps a similar distribution throughout the first 4 iterations. However, by iteration 6 the geometry error dampens out with the exception of a spike at the leading edge. By iteration 6 all geometry residuals are within 5×10^{-4}, which is also an improvement over the normal perturbation method.
Figure 6.1.6: Subsonic design results - pressure gradients
Figure 6.1.7: Subsonic pressure residuals - pressure gradients

Figure 6.1.8: Subsonic airfoil residuals - pressure gradients
Figure 6.1.9 shows the pressure and airfoil geometry error convergence history. The error decreases very rapidly for the first few iterations. However, after about iteration 5 or 6, the error ceases to decrease. It is unclear why this is the case with the possibility of obtaining an exact solution. It is most likely due to the error caused by the grid generation and CFD solution. There is an expected lower bound on the error due to these sources of error. Despite the introduced error, the area between the designed and target airfoils is less than $1 \times 10^{-9}$ for both the upper and lower surfaces.

![Figure 6.1.9: Subsonic design error - pressure gradients](image-url)
Figure 6.1.10 shows the upper and lower surface Bernstein coefficients throughout the pressure gradient design. The coefficients initially have a greater range than in the normal perturbation design case. However, they display similar behavior as the previous case. They converge after 5 or 6 iterations, but converge to slightly different values than the Bernstein coefficients from the normal perturbation design. The final Bernstein coefficients match up better to the target values than those from the normal perturbation design.

![Graph showing Bernstein coefficients for upper and lower surfaces](image)

Figure 6.1.10: Subsonic Bernstein coefficients - pressure gradients
6.1.3 Degree Elevation Pressure Gradients

In the degree elevation pressure gradient method, the initial airfoil is parameterized with 1st order CST curves. If the area between the current and previous airfoils is sufficiently small, the degree is elevated by 1 up to a maximum of 5. This reduces the number of flow solutions compared to the pressure gradient method. Each Bernstein coefficient is perturbed by 1% of its value to calculate the pressure gradients.

Figure 6.1.11 shows the inverse design results for the degree elevation pressure gradient method. The pressures and airfoil do not match the target values as well as the pressure gradient method in the first few iterations. This is expected since low-order CST curves are used. However, after iteration 6 there is almost no difference.

Figure 6.1.12 shows the pressure error at the same iterations as Figure 6.1.11. The residuals exhibit fewer oscillations than the pressure gradient method due to lower-order CST curves being used in the early stages of the design. The pressure error at the leading edge is less than 0.05 after 6 iterations and negligible over the rest of the airfoil. This is approximately the same as the pressure gradient method, but required fewer flow solutions to achieve this accuracy.

Figure 6.1.13 shows the airfoil coordinate residuals throughout the design. Again, the geometry error keeps a similar distribution throughout the first 4 iterations. However, by iteration 6 the geometry error dampens out with the exception of the spike at the leading edge. By iteration 6 all geometry residuals are within $5 \times 10^{-4}$, which is about the same performance as the pressure gradient method.
Figure 6.1.11: Subsonic design results - degree elevation
Figure 6.1.12: Subsonic pressure residuals - degree elevation

Figure 6.1.13: Subsonic airfoil residuals - degree elevation
Figure 6.1.14 shows the pressure and airfoil geometry error convergence history. Again, the error decreases very rapidly in the first few iterations and ceases to decrease after about iteration 5 or 6. At that point the pressure and geometry error is approximately the same as that for the pressure gradient method.

![Figure 6.1.14: Subsonic design error - degree elevation](image-url)
Figure 6.1.15 shows the upper and lower surface Bernstein coefficients throughout the degree elevate design. Degree elevation is evident by coefficients beginning after iteration 1. For example, at iteration 1, only $A_0$ and $A_1$ have values because the airfoil is initially parameterized with a 1st order CST curve. By the end of the design, all coefficients are present. The last Bernstein coefficient $A_5$ does not change after it is introduced. It also does not have an effect on the other coefficients; they do not change after the last coefficient is introduced. Therefore, the orders of the CST curves used are higher than necessary to achieve the minimum error displayed in Figure 6.2.14. The final coefficients are roughly the same as those from the pressure gradient design, which are very close to the target values.

![Subsonic Bernstein coefficients - degree elevation](image1)

(a) Upper Surface  
(b) Lower Surface

**Figure 6.1.15: Subsonic Bernstein coefficients - degree elevation**
6.1.4 Comparison of Methods

Table 6.1.1 shows a comparison of the performance of the three inverse design methods for the subsonic case. The second row shows the number of flow solutions required for each iteration of the different methods. The remaining rows show how many iterations are required to reach a given airfoil $A^2$ value. The left-hand values correspond to the upper surface and the right-hand values to the lower surface. The normal perturbation method can achieve $A^2$ values of $1 \times 10^{-5}$ and $1 \times 10^{-6}$ in far fewer flow solutions than the pressure gradient and degree elevation methods. However, the normal perturbation method does not achieve an error lower than this, signified by the dashes for errors lower than $1 \times 10^{-6}$. To be more accurate one of the other two methods is required. The degree elevation method achieves the best error in the same or fewer number of iterations as the pressure gradient method.

<table>
<thead>
<tr>
<th>Solns/Iter</th>
<th>Normal Perturbation</th>
<th>Pressure Gradient</th>
<th>Degree Elevation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A^2 = 1 \times 10^{-5}$</td>
<td>2/2</td>
<td>1/2</td>
<td>2/4</td>
</tr>
<tr>
<td>$A^2 = 1 \times 10^{-6}$</td>
<td>7/5</td>
<td>5/3</td>
<td>4/4</td>
</tr>
<tr>
<td>$A^2 = 1 \times 10^{-7}$</td>
<td>-</td>
<td>5/3</td>
<td>4/4</td>
</tr>
<tr>
<td>$A^2 = 1 \times 10^{-8}$</td>
<td>-</td>
<td>5/5</td>
<td>5/5</td>
</tr>
<tr>
<td>$A^2 = 1 \times 10^{-9}$</td>
<td>-</td>
<td>5/6</td>
<td>5/5</td>
</tr>
<tr>
<td>$A^2 = 1 \times 10^{-10}$</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
6.2 Transonic Design

A transonic case was selected to see how the methods handle the presence of a shock. The target airfoil for the transonic design case is an RAE 2822 and the initial airfoil is a NACA 0012. The operating conditions are $M_\infty = 0.715$, $Re = 1.0 \times 10^7$, and $\alpha = 2.3^\circ$. Again, the target pressure distribution is obtained using 5th order CST curves to model the target airfoil.

6.2.1 Normal Perturbation

As with the subsonic design, the settings used are a scale factor of 0.04, an initial CST order of 2, and a maximum CST order of 5. The inverse design is again executed for 20 iterations to obtain a good representation of its behavior.

Figure 6.2.1 shows the pressure distribution and airfoil geometry throughout the inverse design process. As expected, the presence of a shock greatly slows convergence of the pressures and airfoil geometry. After iteration 5, the shock strength is close, but the location is too far aft. However, the lower surface pressures and pressures aft of the shock are almost exact. By iteration 10 the shock location and pressures forward of the shock are closer than iteration 5. At iteration 20 the shock strength and location match well and the pressures forward of the shock are nearly exact.

Figure 6.2.2 shows the pressure error at the same iterations as Figure 6.2.1. There is a spike in the pressure error at the leading edge and at the shock throughout the design. The residuals over the remainder of the airfoil dampen out and are negligible.
compared to the leading edge and shock. After 20 iterations, the pressure error at the leading edge is about 0.07 and about 0.05 at the shock.

Figure 6.2.3 shows the airfoil coordinate residuals at the same 4 iterations. The geometry error keeps a similar distribution throughout the design. However, the magnitude decreases throughout the design. By iteration 20 all geometry residuals are within about $5 \times 10^{-4}$. 
Figure 6.2.1: Transonic design results - normal perturbation
Figure 6.2.2: Transonic pressure residuals - normal perturbation

Figure 6.2.3: Transonic airfoil residuals - normal perturbation
Figure 6.2.4 shows the pressure and airfoil geometry error convergence history. The error decreases much slower than for the subsonic design due to the shock. Also, the lower surface converges faster than the upper surface. The more oscillatory nature of the convergence for the upper surface is caused by the designed shock location oscillating around the target shock location.

Figure 6.2.4: Transonic design error - normal perturbation
Figure 6.2.5 shows the upper and lower surface Bernstein coefficients throughout the design. The upper surface coefficients oscillate much more than the lower surface coefficients due to the presence of the shock on the upper surface. They also require more iterations for the oscillations to dampen than the lower surface coefficients. As with the subsonic normal perturbation design, the Bernstein coefficients converge to values slightly off of the target values, most likely caused by a non-optimal scale factor.
6.2.2 Pressure Gradients

As for the subsonic design, 5th order CST curves are used to model the airfoil and the pressure gradients are calculated by perturbing each Bernstein coefficient by 1% of its value.

Figure 6.2.6 displays the inverse design results for the pressure gradient method at the same iterations presented for the normal perturbation method. The shock strength and location are very close after just 1 iteration. However, a second shock is generated slightly aft of the leading edge. After 2 iterations the pressures nearly match the target values including the strength and location of the shock.

Figure 6.2.7 shows the pressure error at the same iterations as Figure 6.2.6. There are residual spikes at the leading edge and shock throughout the first 5 iterations. By iteration 10 the spike at the shock is almost completely dampened out and is absent by iteration 20. This signifies that the pressure gradient method nearly exactly captures the shock, which is a great improvement over the normal perturbation method.

Figure 6.2.8 shows the airfoil coordinate residuals throughout the design. By iteration 10 the residuals are dampened out compared to the error at the leading edge. By iteration 20 all geometry residuals are within $4 \times 10^{-4}$, which is also an improvement over the normal perturbation method.
Figure 6.2.6: Transonic design results - pressure gradients
Figure 6.2.7: Transonic pressure residuals - pressure gradients

Figure 6.2.8: Transonic airfoil residuals - pressure gradients
Figure 6.2.9 shows the pressure and airfoil geometry error convergence history. As with the subsonic design, the error decreases very rapidly for the first few iterations, but ceases to decrease after about iteration 6 or 7. The oscillatory convergence present in the normal perturbation method is completely absent with the exception of the error increasing at one iteration very early in the design.

Figure 6.2.9: Transonic design error - pressure gradients
Figure 6.2.10 shows the upper and lower surface Bernstein coefficients throughout the design. As in the subsonic design, the coefficients initially have a greater range than in the normal perturbation design case. The oscillations displayed in the normal perturbation case are not present in this design. The shock is resolved much quicker than in the previous design. The final Bernstein coefficients are much closer to the target values than those from the normal perturbation design.

Figure 6.2.10: Transonic Bernstein coefficients - pressure gradients
6.2.3 Degree Elevation Pressure Gradients

As for the subsonic design, 1st order CST curves are initially used with a maximum order of 5 and the pressure gradients are calculated by perturbing each Bernstein coefficient by 1% of its value.

Figure 6.2.11 displays pressure distribution and airfoil throughout the inverse design for the degree elevation pressure gradient method. After 1 iteration the pressures are not as close as for the pressure gradient method, particularly on the lower surface. However, after 5 iterations the pressures and geometry nearly match the target values and after 10 iterations there is no difference between the pressure gradient and degree elevation methods.

Figure 6.2.12 shows the pressure error at the same iterations as Figure 6.2.11. As with the pressure gradient method, there are residual spikes at the leading edge and shock early in the design. By iteration 10 the spike at the shock is gone and the error is negligible except at the leading edge. The pressure at the leading edge gets to within 0.02 of the target value.

Figure 6.2.13 shows the airfoil coordinate residuals throughout the design. By iteration 10 the residuals are dampened out compared to the error at the leading edge and all geometry residuals are within $2 \times 10^{-4}$, which is actually less than the pressure gradient method.
Figure 6.2.11: Transonic design results - degree elevation
Figure 6.2.12: Transonic pressure residuals - degree elevation

Figure 6.2.13: Transonic airfoil residuals - degree elevation
Figure 6.2.14 shows the pressure and airfoil geometry error convergence history. As with the subsonic design, the error decreases very rapidly for the first few iterations, but ceases to decrease after about iteration 6 or 7. After the first few iterations and the CST curves are at the maximum allotted order, the convergence and error is similar to the pressure gradient method.

Figure 6.2.14: Transonic design error - degree elevation
Figure 6.2.15 shows the upper and lower surface Bernstein coefficients throughout the design. Degree elevation is evident by coefficients beginning after iteration 1. For example, at iteration 1, only $A_0$ and $A_1$ have values because the airfoil is initially parameterized with a 1st order CST curve. By the end of the design, all coefficients are present. The last Bernstein coefficient $A_5$ on the upper surface does change one iteration after being introduced. However, it does not change after this point. None of the Bernstein coefficients on the lower surface change after the last coefficient is introduced. Therefore, the orders of the CST curves used are higher than necessary to achieve the minimum error displayed in Figure 6.2.14. As with the pressure gradient design, the final Bernstein coefficients converge to very near the target values.
### 6.2.4 Comparison of Methods

Table 6.2.1 shows a comparison of the performance of the three inverse design methods for the transonic case. Again, the second row shows the number of flow solutions required for each iteration of the different methods and the remaining rows show how many iterations are required to reach a given airfoil $A^2$ value. All three methods required more iterations in the transonic design than in the subsonic design to achieve the same $A^2$ values. This is due to the slowed convergence caused by the presence of the shock.

As in the subsonic design, the normal perturbation method does not attain an error lower than $1 \times 10^{-6}$. The degree elevation method requires more iterations to achieve the same error as the pressure gradient method, but requires fewer flow solutions. Therefore, it is still more computationally efficient.

**Table 6.2.1: Number of transonic iterations to reach given geometry error**

<table>
<thead>
<tr>
<th>Solns/Iter</th>
<th>Normal Perturbation</th>
<th>Pressure Gradient</th>
<th>Degree Elevation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A^2 = 1 \times 10^{-5}$</td>
<td>1/2</td>
<td>1/2</td>
<td>1/4</td>
</tr>
<tr>
<td>$A^2 = 1 \times 10^{-6}$</td>
<td>6/4</td>
<td>2/2</td>
<td>4/5</td>
</tr>
<tr>
<td>$A^2 = 1 \times 10^{-7}$</td>
<td>-</td>
<td>4/5</td>
<td>5/6</td>
</tr>
<tr>
<td>$A^2 = 1 \times 10^{-8}$</td>
<td>-</td>
<td>5/6</td>
<td>7/7</td>
</tr>
<tr>
<td>$A^2 = 1 \times 10^{-9}$</td>
<td>-</td>
<td>6/7</td>
<td>7/8</td>
</tr>
<tr>
<td>$A^2 = 1 \times 10^{-10}$</td>
<td>-</td>
<td>-/7</td>
<td>8/9</td>
</tr>
</tbody>
</table>
6.3 Circulation Control Airfoil

The CC design case is different from the previous designs in that there is no target airfoil. The initial airfoil is the GACC airfoil. The target pressure distribution is a redesign of the GACC airfoil pressure distribution. Figure 6.3.1 displays the initial and target pressure distributions for the CC airfoil design case. The red line indicates how the initial pressure distribution of the GACC airfoil is modified to the target distribution.

![Initial and target pressure distributions for CC design - $C_\mu = 0.1$](image)

The main portion of the upper surface is modified to have a flat rooftopt region with a Stratford pressure recovery. While this may imply the airfoil is designed for laminar flow, a turbulent rooftopt is used since a turbulence model is used to analyze...
the flow. In airfoil designs and discussions on high-lift aerodynamics, both Liebeck\(^3\) and Smith\(^85\) present pressure distributions with turbulent rooftops. The height of the rooftop is \(C_p = -6\) to provide a value that the peak of the designed pressure distribution should be able to attain. A rooftop pressure at the suction peak of the initial airfoil was found to be unrealistic in a preliminary study. The pressure recovery ends at the initial pressure forward of the slot and the target distribution follows the initial distribution for the remainder of the upper surface. The lower surface is modified to have \(C_p = 1\) across the entire chord. The operating conditions are \(M_\infty = 0.1\), \(\alpha = 0.0^\circ\), and \(C_\mu = 0.1\). The mass flow rate for this jet momentum coefficient is estimated using Eq. 5.2.7.

Since there is such a significant area change between the initial and target pressure distributions, a cap of 20\% of the maximum thickness is placed on the magnitude of the normal perturbation to prevent an airfoil being selected that is thick enough to generate a large separated region and cause a substantial drop in lift. Also, it is assumed the prescribed pressure distribution does not meet the constraints necessary for a realizable solution. This design case is selected to display the performance of the normal perturbation inverse design method applied to a complex flow like that over a CC airfoil, as well as to show the effects of having an unrealizable solution.

Figure 6.3.2 displays the results of the CC airfoil inverse design. The suction peak at the leading edge decreases in magnitude at each iteration as the height of the main portion of the airfoil increases to reach the rooftop pressure, which is matched well.
The aft pressures on the upper surface match the bottom of the Stratford recovery, but do not increase to match the initial pressure. This may be due to the fact that the flow over the trailing edge of the upper surface is not parallel to the flow leaving the slot due to the large change in slope at the trailing edge. The airfoil begins to resemble a Liebeck high lift airfoil designed to generate a pressure distribution with a small rooftop region followed by a Stratford recovery.
Figure 6.3.2: CCW results - normal perturbation
Chapter 7
Final Remarks

7.1 Conclusions

Three novel inverse design methods were presented that are accurate, efficient, and simple to implement. They were shown to work for a subsonic case and a transonic case. A high lift subsonic case implementing circulation control was also presented where the target pressure distribution was not realizable. The designed pressure distribution for this case reduced the error in the pressure distribution and the designed airfoil resembled results by Liebeck without circulation control.

The normal perturbation (NP) method simply relates the pressure residuals to the required change in airfoil shape. The airfoil is then perturbed by a given fraction of the pressure residuals.

The pressure gradient (PG) method perturbs the CST coefficients used to parameterize the airfoil in order to determine their influence on the pressure distribution. These distributions can be superimposed to yield an updated airfoil that has a pressure distribution more closely resembling the target pressure distribution.
The pressure gradient degree elevation (PGDE) method is identical to the PG method except that a low order CST curve is used initially. Once the solution does not significantly improve, the order of the curve is increased. This is more computationally efficient because fewer gradients are calculated in the early part of the design.

When an unrealizable pressure distribution is prescribed, the methods attempt to find the best possible solution. In the case with circulation control, the designed airfoil reached the rooftop pressure over a short portion of the chord and decreased to match the bottom of the Stratford pressure recovery.

### 7.2 Future Work

If the scale factor in the normal perturbation method can be related to the flight conditions, the burden on the user would be greatly reduced and the computational efficiency of the method is improved. One could investigate including the angle of attack as a degree of freedom and parameterizing the thickness and camber distributions of the airfoil instead of the upper and lower surfaces. Also, a study could be conducted into extending the methods to a multi-point inverse design.

One problem with the method is that having an excessively high order curve can lead to oscillations in the solution or lead to unrealistic airfoils that contain many bumps. A study must be conducted to determine the effects of the order of the curve on the solution and determine if a maximum possible order exists.

The methods presented here can be improved by eliminating the spike in pressure
error at the leading edge that exists in all the methods. This could potentially be achieved by parameterizing the leading edge with a separate curve and blending it with a curve for the rest of the airfoil. Perhaps the CST method could be modified to allow the Bernstein coefficients to have a leading edge bias. Some methods implement what is referred to as a “leading edge patch” to handle this, which is a similar idea.

The methods may also be extended to supersonic flow by modifying the perturbation method. Due to the differences in subsonic and supersonic flow, the methods presented are not valid for supersonic flow. A potential method to perturb the airfoil surface is to relate the shocks and expansions to how the airfoil shape must change.

Future cases with circulation control should include the flap geometry in the inverse design. This allows the flow at the trailing edge of the airfoil to be parallel to the flow blown out of the slot. This would resolve the issue present with a constant flap geometry where the flow at the trailing edge of the airfoil is at a different angle than the flow blown out of the slot.
Appendix A

Residual Correction Method

A.1 Updated Scale Factor

The initial scale factor can be used in conjunction with the previous and updated pressure distributions to produce a better estimate of the scale factor. Knowing how the change in the airfoil shape changed the pressure distribution permits an updated scale factor to be calculated. This reduces the burden on the user because the scale factor is improved throughout the design process. The updated scale factor is expressed as

\[ r_i(x) = \frac{C_{p,i}(x) - C_{p,t}(x)}{|C_{p,i}(x) - C_{p,i-1}(x)|} \]  \hspace{1cm} (A.1.1)

The perturbations are now expressed as a correction to the perturbations from the previous iteration.

\[ \Delta \hat{n}_i = r_i |\Delta \hat{n}_{i-1}| \]  \hspace{1cm} (A.1.2)

The user specifies the scale factor for the first iteration since there is no previous iteration to update the normal perturbation. The normal perturbation for the first iteration is identical to the previous method. Subsequent iterations use normal per-
turbations calculated according to Eqs. A.1.1 and A.1.2. While this does improve the method, it still has the oscillation problem of the previous method. A better use of the CST method can rectify this problem. Instead of using CST curves as a smoothing process, the parameterizations can be built into the perturbation method to calculate the best airfoil to use as the updated shape, not just a shape closer to the target shape.
Appendix B

Governing Equations

The governing equations for viscous fluid flow consist of the Navier-Stokes equations. The Navier-Stokes equations include the continuity, momentum, and energy equations, which are explained in this section. The governing equations will not be formally derived in this section. Schlichting\textsuperscript{140} provides a thorough derivation using the phenomenological approach where the equations are derived from the conservation laws using relations between stress, strain rate, heat flux, and the temperature gradient. Hirschfelder et al.\textsuperscript{141} handles the kinetic theory approach, where the fluid dynamic equations are derived using integral relations for the transport coefficients. The following presentation follows the approach by Tannehill et al.\textsuperscript{127}
B.1 Navier-Stokes Equations

B.1.1 Continuity Equation

The law of conservation of mass applied to a fluid passing through an infinitesimal, fixed control volume yields the following form of the continuity equation.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{U}) = 0 \quad (B.1.1)$$

The first term represents the rate of increase of density in the control volume. The second term represents the rate of mass flux leaving the control surface per unit volume. For a Cartesian coordinate system, Eq. B.1.1 can be written in terms of the components of the velocity vector ($u, v, w$ in the $x, y, z$ directions).

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x} (\rho u) + \frac{\partial}{\partial y} (\rho v) + \frac{\partial}{\partial z} (\rho w) = 0 \quad (B.1.2)$$

This can be expressed more generally by using Einstein notation yielding

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_j} (\rho u_j) = 0 \quad (B.1.3)$$

B.1.2 Momentum Equations

Newton’s second law applied to a fluid passing through an infinitesimal, fixed control volume yields following form of the momentum equation.

$$\frac{\partial}{\partial t} \left( \rho \mathbf{U} \right) + \nabla \cdot \rho \mathbf{U} \mathbf{U} = \rho \mathbf{f} + \nabla \cdot \mathbf{Π} \quad (B.1.4)$$

The first term represents the rate of increase of momentum per unit volume in the control volume. The second term represents the rate of momentum lost by convection.
per unit volume through the control surface. The second term can be expanded as

$$\nabla \cdot \rho \vec{U} \vec{U} = \rho \vec{U} \cdot \nabla \vec{U} + \vec{U} \left( \nabla \cdot \rho \vec{U} \right)$$  \hspace{1cm} (B.1.5)

When this is substituted into Eq. B.1.4, the resulting equation can be simplified with the continuity equation yielding

$$\rho \frac{D \vec{U}}{Dt} = \rho \vec{f} + \nabla \cdot \vec{\Pi}_{ij}$$  \hspace{1cm} (B.1.6)

where the total or substantial derivative is defined as

$$\frac{D ( \ )}{Dt} \equiv \frac{\partial ( \ )}{\partial t} + \vec{U} \cdot \nabla ( \ )$$  \hspace{1cm} (B.1.7)

The first term on the right-hand side of Eq. B.1.6 represents the body force per unit volume. The second term on the right-hand side is a stress tensor and represents the surface forces per unit volume, which consist of normal and shearing stresses. This formulation is general and applies to both continuum and non-continuum flows. It only loses its generality when approximate expressions for the shear-stress tensor are used. For a Newtonian fluid (stresses are linearly dependent on the strain rates), the stress tensor can be related to the pressure and velocity components with a general deformation law\(^{140}\) as

$$\vec{\Pi}_{ij} = -p\delta_{ij} + \mu \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) + \delta_{ij}\mu' \frac{\partial u_k}{\partial x_k} \hspace{1cm} i, j, k = 1, 2, 3$$  \hspace{1cm} (B.1.8)

where \(\delta_{ij}\) is the Kronecker delta function

$$\delta_{ij} = \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases}$$  \hspace{1cm} (B.1.9)
and \( \mu' \) is the second coefficient of viscosity, which is related to dynamic viscosity through the coefficient of bulk viscosity \( \kappa \) with the expression

\[
\kappa = \frac{2}{3} \mu + \mu' \quad \text{(B.1.10)}
\]

In practice it is typically assumed that \( \kappa \) is negligible. Therefore, the second coefficient of viscosity becomes

\[
\mu' = -\frac{2}{3} \mu \quad \text{(B.1.11)}
\]

permitting the stress tensor to be written as

\[
\Pi_{ij} = -p \delta_{ij} + \mu \left[ \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \delta_{ij} \frac{\partial u_k}{\partial x_k} \right] \quad i, j, k = 1, 2, 3 \quad \text{(B.1.12)}
\]

The viscous stresses are commonly separated from the pressures in the stress tensor in the following manner

\[
\Pi_{ij} = -p \delta_{ij} + \tau_{ij} \quad \text{(B.1.13)}
\]

and the viscous stress tensor \( \tau_{ij} \) is simply the second term of the right-hand side of the stress tensor (Eq. B.1.12).

\[
\tau_{ij} = \mu \left[ \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \delta_{ij} \frac{\partial u_k}{\partial x_k} \right] \quad i, j, k = 1, 2, 3 \quad \text{(B.1.14)}
\]

Substituting Eq. B.1.12 into Eq. B.1.6 yields the Navier-Stokes equation

\[
\rho \frac{D\bar{U}}{Dt} = \rho \bar{f} - \nabla p + \frac{\partial}{\partial x_j} \left[ \mu \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \delta_{ij} \mu \frac{\partial u_k}{\partial x_k} \right] \quad \text{(B.1.15)}
\]
For a Cartesian coordinate system, Eq. B.1.15 can be written in conservative form by utilizing Eq. B.1.4

\[
\frac{\partial}{\partial t} (\rho u_i) + \frac{\partial}{\partial x_j} (\rho u_i u_j + p \delta_{ij} - \tau_{ij}) = \rho f_i \tag{B.1.16}
\]

This can be expanded as

\[
\frac{\partial}{\partial t} (\rho u) + \frac{\partial}{\partial x} (\rho u^2 + p - \tau_{xx}) + \frac{\partial}{\partial y} (\rho u v - \tau_{xy}) + \frac{\partial}{\partial z} (\rho u w - \tau_{xz}) = \rho f_x
\]

\[
\frac{\partial}{\partial t} (\rho v) + \frac{\partial}{\partial x} (\rho u v - \tau_{xy}) + \frac{\partial}{\partial y} (\rho v^2 + p - \tau_{yy}) + \frac{\partial}{\partial z} (\rho v w - \tau_{yz}) = \rho f_y
\]

\[
\frac{\partial}{\partial t} (\rho w) + \frac{\partial}{\partial x} (\rho u w - \tau_{xz}) + \frac{\partial}{\partial y} (\rho v w - \tau_{yz}) + \frac{\partial}{\partial z} (\rho w^2 + p - \tau_{zz}) = \rho f_z
\]

The components of the viscous stress tensor \(\tau_{ij}\) are expressed using Eq. B.1.14 by entering the corresponding coordinates.

\[
\tau_{xx} = \frac{2}{3} \mu \left( \frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} - \frac{\partial w}{\partial z} \right)
\]

\[
\tau_{yy} = \frac{2}{3} \mu \left( \frac{\partial v}{\partial y} - \frac{\partial u}{\partial x} - \frac{\partial w}{\partial z} \right)
\]

\[
\tau_{zz} = \frac{2}{3} \mu \left( \frac{\partial w}{\partial z} - \frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} \right)
\]

\[
\tau_{xy} = \mu \left( \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) = \tau_{yx}
\]

\[
\tau_{xz} = \mu \left( \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \right) = \tau_{zx}
\]

\[
\tau_{yz} = \mu \left( \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right) = \tau_{zy}
\]

125
B.1.3 Energy Equation

The first law of thermodynamics applied to a fluid passing through an infinitesimal, fixed control volume yields the following form of the energy equation

$$\frac{\partial E_t}{\partial t} + \nabla \cdot E_t \vec{U} = \frac{\partial Q}{\partial t} - \nabla \cdot \vec{q} + \rho \vec{f} \cdot \vec{U} + \nabla \cdot \left( \Pi_{ij} \cdot \vec{U} \right)$$  \hspace{1cm} (B.1.19)

where $E_t$ is the total energy per unit volume and is expressed as

$$E_t = \rho \left( e + \frac{U^2}{2} + \text{potential energy} + \cdots \right)$$  \hspace{1cm} (B.1.20)

and $e$ is the internal energy per unit mass. The first term on the left-hand side of Eq. B.1.19 is the rate of increase of total energy in the control volume. The second term on the left-hand side represents the rate of total energy lost by convection per unit volume through the control surface. The first term on the right-hand side is the rate of heat production and the second term is the rate heat is lost by conduction per unit volume through the control surface. The heat transfer $\vec{q}$ can be expressed in terms of the temperature and the coefficient of thermal conductivity $k$ by assuming Fourier’s law for heat transfer by conduction.

$$\vec{q} = -k \nabla T$$  \hspace{1cm} (B.1.21)

The third term on the right-hand side of Eq. B.1.19 is equivalent to the work done on the control volume per unit volume by body forces. The last term on the right-hand side represents the work done on the control volume per unit volume by the surface forces. For a Cartesian coordinate system, Eq. B.1.19 can be rewritten...
in conservative form as

\[
\frac{\partial E_t}{\partial t} - \frac{\partial Q}{\partial t} - \rho (f_x u + f_y v + f_z w) \\
+ \frac{\partial}{\partial x} (E_t u + pu - u \tau_{xx} - v \tau_{xy} - w \tau_{xz}) \\
+ \frac{\partial}{\partial y} (E_t v + pv - u \tau_{xy} - v \tau_{yy} - w \tau_{yz}) \\
+ \frac{\partial}{\partial z} (E_t w + pw - u \tau_{xz} - v \tau_{yz} - w \tau_{zz}) = 0
\] (B.1.22)

where the components of the stress tensor are calculated using Eq. B.1.18. The continuity equation can be used to replace the left-hand side of Eq. B.1.19 with

\[
\rho D \frac{(E_t/\rho)}{Dt} = \frac{\partial E_t}{\partial t} + \nabla \cdot E_t \mathbf{U}
\] (B.1.23)

If only internal and kinetic energy are considered significant, this can also be written as

\[
\rho \frac{D (E_t/\rho)}{Dt} = \rho \frac{De}{Dt} + \rho \frac{D (U^2/2)}{Dt}
\] (B.1.24)

\[
\rho \frac{D \mathbf{U}}{Dt} \cdot \mathbf{U} = \rho \mathbf{f} \cdot \mathbf{U} - \nabla p \cdot \mathbf{U} + (\nabla \cdot \tau_{ij}) \cdot \mathbf{U}
\] (B.1.25)

Substituting Eqs. B.1.23-B.1.25 into B.1.19 yields

\[
\rho \frac{De}{Dt} + p (\nabla \cdot \mathbf{U}) = \frac{\partial Q}{\partial t} - \nabla \cdot \mathbf{q} + \nabla \cdot (\tau_{ij} \cdot \mathbf{U}) - (\nabla \cdot \tau_{ij}) \cdot \mathbf{U}
\] (B.1.26)

The last two terms on the right-hand side can be combined as

\[
\tau_{ij} \frac{\partial u_i}{\partial x_j} = \nabla \cdot (\tau_{ij} \cdot \mathbf{U}) - (\nabla \cdot \tau_{ij}) \cdot \mathbf{U}
\] (B.1.27)

This is typically referred to as the dissipation function \( \Phi \), which represents the rate at which mechanical energy is expended due to viscosity deforming the fluid.
Equation B.1.26 is now expressed as

\[ \rho \frac{De}{Dt} + p (\nabla \cdot \vec{U}) = \frac{\partial Q}{\partial t} - \nabla \cdot \vec{q} + \Phi \]  

(B.1.28)

Using the continuity equation and the following definition of enthalpy,

\[ h = e + \frac{p}{\rho} \]  

(B.1.29)

Eq. B.1.28 is reformulated as

\[ \rho \frac{Dh}{Dt} = \frac{Dp}{Dt} + \frac{\partial Q}{\partial t} - \nabla \cdot \vec{q} + \Phi \]  

(B.1.30)

Using a Cartesian coordinate system and \( \mu' = - \frac{2}{3} \mu \) so the dissipation function is always positive, the dissipation function becomes

\[ \Phi = \mu \left[ 2 \left( \frac{\partial u}{\partial x} \right)^2 + 2 \left( \frac{\partial v}{\partial y} \right)^2 + 2 \left( \frac{\partial w}{\partial z} \right)^2 + \left( \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right)^2 + \left( \frac{\partial w}{\partial y} + \frac{\partial v}{\partial z} \right)^2 \right. \]

\[ + \left. \left( \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right)^2 - \frac{2}{3} \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right)^2 \right] \]  

(B.1.31)

**B.1.4 Vector Formulation**

Before solving the Navier-Stokes equations with a numerical algorithm, it is convenient to combine them in a vector formulation. The compressible Navier-Stokes equations in Cartesian coordinates are represented by Eqs. B.1.2, B.1.16, and B.1.22. They can be further reduced assuming there are no body forces, mass diffusion, finite-rate chemical reactions, or external heat addition. These assumptions yield the following form of the Navier-Stokes equations

\[ \frac{\partial \vec{v}}{\partial t} + \frac{\partial \vec{F}}{\partial x} + \frac{\partial \vec{G}}{\partial y} + \frac{\partial \vec{H}}{\partial z} = 0 \]  

(B.1.32)
where $\vec{V}$, $\vec{F}$, $\vec{G}$, and $\vec{H}$ are given by Eqs. B.1.2, B.1.16, and B.1.22 after applying the aforementioned assumptions.

$$
\vec{V} = \begin{bmatrix}
\rho \\
\rho u \\
\rho v \\
\rho w \\
E_t
\end{bmatrix}
$$

$$
\vec{F} = \begin{bmatrix}
\rho u \\
\rho u^2 + p - \tau_{xx} \\
\rho u v - \tau_{xy} \\
\rho u w - \tau_{xz} \\
(E_t + p) u - u \tau_{xx} - v \tau_{xy} - w \tau_{xz} + q_x
\end{bmatrix}
$$

$$
\vec{G} = \begin{bmatrix}
\rho v \\
\rho v u - \tau_{xy} \\
\rho v^2 + p - \tau_{yy} \\
\rho v w - \tau_{yz} \\
(E_t + p) v - u \tau_{xy} - v \tau_{yy} - w \tau_{yz} + q_y
\end{bmatrix}
$$

$$
\vec{H} = \begin{bmatrix}
\rho w \\
\rho w u - \tau_{xz} \\
\rho w v - \tau_{yz} \\
\rho w^2 + p - \tau_{zz} \\
(E_t + p) w - u \tau_{xz} - v \tau_{yz} - w \tau_{zz} + q_z
\end{bmatrix}
$$

The components of the stress tensor are again calculated using Eq. B.1.18. The first row of Eq. B.1.32 is simply the continuity equation, while the second, third, and forth rows correspond to the momentum equations and the fifth row is the energy equation.

## B.2 Reynolds Averaged Navier-Stokes Equations

FLUENT is one of what are known as RANS solvers. This means they solve the Reynolds averaged Navier-Stokes (RANS) equations. To obtain the RANS equations, time averaged and Favre averaged (mass averaged) quantities must first be introduced.
These relationships are used to transform the Navier-Stokes equations into the RANS equations.

### B.2.1 Time-Averaged Quantities

FLUENT and other CFD solvers can only capture low-frequency effects, not the high-frequency effects. Therefore, it becomes necessary to introduce time-averaged quantities that filter out the high-frequency effects. If a given flow variable is expressed as $f$, then the time-averaged quantity $\overline{f}$ is defined as

$$\overline{f} = \frac{1}{\Delta t} \int_t^{t+\Delta t} f dt \quad (B.2.1)$$

This requires that $\Delta t$ be large with respect to the period of the random turbulence fluctuations (high-frequency effects), but small compared to the time constant for slow variations in the flow field normally associated with unsteady flows (low-frequency effects). According to conventional Reynolds decomposition, the randomly changing flow variables are replaced with their time averages plus its fluctuation about the average. This is expressed as

$$f = \overline{f} + f' \quad (B.2.2)$$

where $f'$ represents the fluctuating quantity. By definition, the time average of a fluctuating quantity is zero.

$$\overline{f'} = \frac{1}{\Delta t} \int_{t_0}^{t_0+\Delta t} f' dt \equiv 0 \quad (B.2.3)$$

Given these definitions, the following relations hold for any flow variables $f$ and $g$.

$$\overline{fg'} = 0 \quad \overline{f}g = \overline{fg} \quad \overline{f + g} = \overline{f} + \overline{g} \quad (B.2.4)$$
Also, even though the time average of a fluctuating quantity ($\bar{f}$) is zero, the time average of the product of two fluctuating quantities is, in general, not zero ($\bar{f}g' \neq 0$).

### B.2.2 Favre-Averaged (Mass-Averaged) Quantities

For treatment of compressible flows, mass-averaged quantities are typically used. Mass-averaged quantities are defined as the time average of the product of density and the given flow variable divided by the time average of density.

$$\tilde{f} = \frac{\bar{\rho f}}{\bar{\rho}}$$  \hspace{1cm} (B.2.5)

This is substituted into the conservative form by defining new fluctuating quantities.

$$f = \tilde{f} + f''$$  \hspace{1cm} (B.2.6)

where $f''$ represents the new fluctuating quantity. These new fluctuating quantities will be referred to as secondary fluctuating quantities. In general, the time averages of the secondary fluctuating quantities are not zero unless $\rho' = 0$. However, the time average of the product of density and a secondary fluctuating quantity is zero.

$$\bar{\rho f''} \equiv 0$$  \hspace{1cm} (B.2.7)

Also, the time average of a secondary fluctuating quantity can be shown to equal

$$\bar{f''} = -\frac{\rho' \bar{f}}{\bar{\rho}}$$  \hspace{1cm} (B.2.8)

### B.2.3 Continuity Equation

In order to transform the continuity equation into the RANS formulation, Eqs. B.2.2 and B.2.6 are used for density and velocity, respectively. This yields $\rho = \bar{\rho} + \rho'$
and \( u_i = \tilde{u}_i + u_i'' \). These quantities are substituted into the continuity equation in Cartesian coordinates and the result is time averaged. Therefore, Eq. B.1.3 becomes

\[
\frac{\partial}{\partial t} (\bar{\rho} + \rho') + \frac{\partial}{\partial x_j} \left( (\bar{\rho} + \rho') \left( \tilde{u}_j + u_j'' \right) \right) = 0 \quad \text{(B.2.9)}
\]

Using the properties of Eqs. B.2.3 and B.2.4, it can be shown that

\[
\bar{\rho} + \rho' = \bar{\rho} + \rho = \rho
\]

and by Eqs. B.2.2, B.2.3, and B.2.7

\[
(\bar{\rho} + \rho') \left( \tilde{u}_j + u_j'' \right) = \bar{\rho} \tilde{u}_j + \rho' \tilde{u}_j + \bar{\rho} u_j'' + \rho u_j''
\]

\[
= \bar{\rho} \tilde{u}_j + \rho' \tilde{u}_j + \rho u_j''
\]

\[
= \bar{\rho} \tilde{u}_j + \rho' \tilde{u}_j + \rho u_j''
\]

This yields the RANS formulation of the continuity equation.

\[
\frac{\partial \bar{\rho}}{\partial t} + \frac{\partial}{\partial x_j} \left( \bar{\rho} \tilde{u}_j \right) = 0 \quad \text{(B.2.10)}
\]

**B.2.4 Momentum Equations**

Similar to the continuity equation, the momentum equation in Cartesian coordinates (Eq. B.1.16) is transformed into the RANS formulation by substituting \( \rho = \bar{\rho} + \rho' \), \( u = \tilde{u} + u'' \), and \( p = \bar{p} + p' \) and taking a time average of the result. After introducing these time and mass-averaged values and neglecting body forces, the momentum equation becomes

\[
\frac{\partial}{\partial t} \left[ (\bar{\rho} + \rho') (\tilde{u}_i + u_i'') \right] + \frac{\partial}{\partial x_j} \left[ (\bar{\rho} + \rho') \left( \tilde{u}_i + u_i'' \right) \left( \tilde{u}_j + u_j'' \right) \right]
\]

\[
+ (\bar{\rho} + \rho') \delta_{ij} - \tau_{ij} = 0 \quad \text{(B.2.11)}
\]
Time averaging the unsteady term and reducing using Eqs. B.2.3 and B.2.7 yields

\[
\begin{align*}
\overline{(\rho + \rho') (\tilde{u}_i + u''_i)} &= \rho \tilde{u}_i + \rho' \tilde{u}_i + \rho u''_i \\
&= \rho \tilde{u}_i + \rho' \tilde{u}_i + \rho' 0 \\
&= \rho \tilde{u}_i
\end{align*}
\] (B.2.12)

The time average of the convective term \(\rho u_i u_j\) can be reduced by

\[
\begin{align*}
\overline{(\rho + \rho') (\tilde{u}_i + u''_i) (\tilde{u}_j + u''_j)} &= \rho \tilde{u}_i \tilde{u}_j + \rho' \tilde{u}_i \tilde{u}_j + \rho u''_i \tilde{u}_j + \rho \tilde{u}_i \rho' u''_j + \rho u''_i \rho' u''_j \\
&= \rho \tilde{u}_i \tilde{u}_j + \rho' \tilde{u}_i \tilde{u}_j + \tilde{u}_j \rho' \tilde{u}_i + \tilde{u}_i \rho' \tilde{u}_j + \rho u''_i \rho' u''_j \\
&= \rho \tilde{u}_i \tilde{u}_j + \rho u''_i \rho' u''_j
\end{align*}
\] (B.2.13)

The time average of the pressure term \(\overline{(\rho + \rho') \delta_{ij}}\) simply becomes \(\rho \delta_{ij}\).

\[
\begin{align*}
\overline{\rho + \rho'} &= \rho + \rho' 0 \\
&= \rho
\end{align*}
\] (B.2.14)

The viscous tensor \(\tau_{ij}\) is given by Eq. B.1.14. Using mass-averaged velocities, time averaging, and assuming \(\mu' \approx 0\) and \(\mu \approx \overline{\rho}\) yields

\[
\tau_{ij} = \mu \left[ \frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i} - \frac{\delta_{ij}}{3} \frac{\partial \tilde{u}_k}{\partial x_k} \right] + \mu \left[ \frac{\partial u''_i}{\partial x_j} + \frac{\partial u''_j}{\partial x_i} - \frac{2}{3} \delta_{ij} \frac{\partial u''_k}{\partial x_k} \right]
\] (B.2.15)

Equations B.2.11-B.2.15 can be combined to give the RANS formulation of the momentum equation.

\[
\frac{\partial}{\partial t} (\overline{\rho \tilde{u}_i}) + \frac{\partial}{\partial x_j} (\overline{\rho \tilde{u}_i \tilde{u}_j}) = -\frac{\partial}{\partial x_i} \overline{\rho} + \overline{\rho u''_i \tilde{u}_j} (\tau_{ij} - \rho u''_i u''_j)
\] (B.2.16)

### B.2.5 Energy Equation

If the total energy is assumed to be composed of only internal energy and kinetic energy, heat generation is neglected, and \(E_i\) is replaced with \(\rho h - p\), Eq. B.1.19 can
be written as

\[ \frac{\partial}{\partial t} (\rho h) + \frac{\partial}{\partial x_j} (\rho u_j h + q_j - u_i \tau_{ij}) = \frac{\partial p}{\partial t} \]  

(B.2.17)

Using Eq. B.2.2 for the dependent variables and time averaging the result yields

\[ \frac{\partial}{\partial t} (\bar{p} \bar{h} + \rho' \bar{h}') + \frac{\partial}{\partial x_j} \left( \bar{p} \bar{u}_j \bar{h} + \rho' \bar{u}_j' \bar{h}' + \bar{u}_j \rho' \bar{h}' - k \frac{\partial \bar{T}}{\partial x_j} \right) \]

\[ = \frac{\partial \bar{p}}{\partial t} + \frac{\partial}{\partial x_j} \left[ \bar{u}_i \left( -\frac{2}{3} \mu \delta_{ij} \frac{\partial \bar{u}_k}{\partial x_k} \right) \right] + \mu \bar{u}_i \left( \frac{\partial \bar{u}_j}{\partial x_i} + \frac{\partial \bar{u}_i}{\partial x_j} \right) \]

(B.2.18)

Expressing Eq. B.1.30 in conservative form and letting \( h = c_p T \) yields

\[ \frac{\partial}{\partial t} (\rho c_p T) + \frac{\partial}{\partial x_j} \left( \rho c_p u_j T - k \frac{\partial T}{\partial x_j} \right) = \frac{\partial \bar{p}}{\partial t} + u_j \frac{\partial \bar{p}}{\partial x_j} + \Phi \]  

(B.2.19)

The dissipation function can be written in terms of velocity components as

\[ \Phi = \tau_{ij} \frac{\partial u_i}{\partial x_j} = \mu \left[ -\frac{2}{3} \left( \frac{\partial u_k}{\partial x_k} \right)^2 + \frac{1}{2} \left( \frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j} \right)^2 \right] \]  

(B.2.20)

Similar to the process for the continuity and momentum equations, the density, pressure, and temperature in Eq. B.2.19 are replaced with time-averaged values according to Eq. B.2.2. Also, the velocity and enthalpy are replaced with mass-averaged quantities according to Eq. B.2.6. The result is then time averaged. Eliminating terms known to be zero yields the Reynolds-averaged energy equation.

\[ \frac{\partial}{\partial t} (\bar{p} \bar{h}) + \frac{\partial}{\partial x_j} \left( \bar{p} \bar{u}_j \bar{h} + \rho' \bar{u}_j' \bar{h}' - k \frac{\partial \bar{T}}{\partial x_j} \right) = \frac{\partial \bar{p}}{\partial t} + \frac{\partial}{\partial x_j} \left( \bar{u}_j \tau_{ij} + \bar{u}_i' \tau_{ij} \right) \]  

(B.2.21)
This can be written in terms of static temperature by letting \( \hat{h} = c_p \hat{T} \).

\[
\frac{\partial}{\partial t} \left( \rho c_p \hat{T} \right) + \frac{\partial}{\partial x_j} \left( \rho c_p \hat{T} \hat{u}_j \right) = \frac{\partial \bar{p}}{\partial t} + \hat{u}_j \frac{\partial \bar{p}}{\partial x_j} + u''_j \frac{\partial p}{\partial x_j} + \frac{\partial}{\partial x_j} \left( \frac{k}{c_p} \frac{\partial \hat{T}}{\partial x_j} + k \frac{\partial T''}{\partial x_j} - c_p \rho T'' u''_j \right) + \Phi \tag{B.2.22}
\]

where

\[
\Phi = \tau_{ij} \frac{\partial \hat{u}_i}{\partial x_j} = \tau_{ij} \frac{\partial \hat{u}_i}{\partial x_j} + \tau_{ij} \frac{\partial u''_i}{\partial x_j} \tag{B.2.23}
\]
Appendix C

Grid Spacing Calculation

The inputs to the grid spacing calculator include Reynolds number, velocity, density, kinematic viscosity, and the desired $y^+$ value. Jones$^{128}$ implements a Newton-Raphson method$^{107}$ to solve for the friction coefficient $C_f$ and defines the initial value of $C_f$ to be 0.02. In the Newton-Raphson solver, an updated $C_f$ value is calculated according to

$$C_{f,i+1} = C_{f,i} - \frac{F(C_f)}{F'(C_f)} \quad (C.0.1)$$

The function $F(C_f)$ is defined by Jones as

$$F = 4.15 \sqrt{C_f \log_{10}(Re \cdot C_f)} + 1.7 \sqrt{C_f} - 1 \quad (C.0.2)$$

where $F'(C_f)$ is equivalent to $\partial F/\partial C_f$ and is written as

$$F'(C_f) = \frac{4.15 \log_{10}(e) + 2.075 \log_{10}(Re \cdot C_f) + 0.85}{\sqrt{C_f}} \quad (C.0.3)$$

Jones places a restriction on the updated $C_f$ value according to

$$C_{f,i+1} = \begin{cases} 0.5C_{f,i} & \text{if } C_{f,i+1} \leq 0 \\ C_{f,i+1} & \text{if } C_{f,i+1} > 0 \end{cases} \quad (C.0.4)$$
Once the solver converges, a final value for $C_f$ is calculated as

$$C_f = \left[ \frac{1}{4.15 \log_{10} (Re \cdot C_f) + 1.7} \right]^2$$

(C.0.5)

This is used to estimate the wall shear stress as

$$\tau_w = C_f q$$

(C.0.6)

where $q$ is simply the dynamic pressure. The wall shear stress is used to calculate the friction velocity according to

$$u_\tau = \sqrt{\frac{\tau_w}{\rho}}$$

(C.0.7)

This is used along with the following definition of $y^+$ to calculate the height of the first cell in the boundary layer.

$$y^+ = \frac{u_\tau y}{\nu}$$

(C.0.8)
Appendix D

Grid Convergence

D.1 Grid Convergence Index (GCI)

Roache\textsuperscript{137} discusses a grid convergence index (GCI) used to determine if a grid has satisfactory refinement meaning the solution is not a function of the number of cells in the grid. The GCI is a measure of the discretization error of a particular solution variable where said variable is confident to within ±GCI. A similar method that is based on Roache’s method is presented in Celik et al.\textsuperscript{142} The GCI is based on a generalization of the Richardson extrapolation (Richardson\textsuperscript{143} and Richardson and Gaunt\textsuperscript{144}). The Richardson extrapolation is defined as a Taylor series expansion

\[ f = f_\infty + \sum_{i=1}^{n} \alpha_i h^{p_i} \]  

(D.1.1)

and the Richardson error as

\[ \delta_{RE} = f - f_\infty \]  

(D.1.2)

where \( f \) is some flow variable critical to the study, \( f_\infty \) is the extrapolated value, \( h \) is the grid spacing, \( p_i \) is related to the order of accuracy of the method, and \( \alpha_i \) is a
constant. This is typically approximated with a one-term expansion where only the leading term of the expansion is retained and the remaining terms are grouped as “higher order terms.” Equation D.1.1 then becomes

\[ f = f_\infty + \alpha h^p + \text{H.O.T.} \]  

(D.1.3)

When \( h \) becomes sufficiently small, the higher order terms are negligible compared to the leading term \( h^p \) and are ignored, yielding

\[ f \approx f_\infty + \alpha h^p \]  

(D.1.4)

This simplification comes with the assumption that the solutions are in the asymptotic range of convergence. The Richardson extrapolation was derived assuming a second-order accurate solution (\( p = 2 \)) and is expressed as

\[ f_\infty = \frac{h_2^2 f_1 - h_1^2 f_2}{h_2^2 - h_1^2} + \text{H.O.T.} \]  

(D.1.5)

where subscripts 1 and 2 refer to the fine grid and coarse grid, respectively. The exponent 2 comes from the assumption of a second-order accurate method. A grid refinement ratio \( r = h_2/h_1 \) can be introduced to express the extrapolated solution as a correction to the fine grid solution

\[ f_\infty \approx f_1 + \frac{f_1 - f_2}{r^2 - 1} \]  

(D.1.6)

where the higher order terms have been dropped. The GCI method involves generalizing the Richardson extrapolation to \( p \)-th order methods by replacing the exponent.
2 with the observed order p. The extrapolated solution in the GCI method can be expressed as any of the following equivalent statements

$$f_\infty \approx f_1 + \frac{f_1 - f_2}{r^p - 1}$$

$$\approx f_2 + \frac{r^p f_1 - f_2}{r^p - 1}$$

$$\approx \frac{r^p f_1 - f_2}{r^p - 1}$$

(D.1.7)

This is taken as the “exact” solution if an infinite number of cells were used in the grid. The first statement is identical to the Richardson extrapolation with an observed order p of 2. The observed order is the order the data appears to exhibit. It is calculated to compare to the theoretical order or when the theoretical order is unknown. The theoretical order is the nominal order of the method used (e.g. 2 for a second-order upwind CFD solution). An observed order roughly equal to the theoretical order is a good indicator that one may proceed with confidence.

The standard formulation of the GCI requires flow solutions on three grids since with different levels of refinement. The third grid is required due to the added unknown by introducing p to the Richardson extrapolation. First, a representative cell size h is defined for each grid. For a two-dimensional grid, the average length of the side of a cell is used and is calculated as

$$h = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (\Delta A_i)}$$

(D.1.8)

where $\Delta A_i$ is the area of the $i^{th}$ cell and N is the total number of cells in the grid. Celik et al. note that the grid refinement factor ($r = h_{\text{coarse}}/h_{\text{fine}}$) should be greater than
1.3. This value comes from experience rather than formal derivation. Simulations are then conducted on each grid to obtain the key flow variables $f$. Once the solutions are completed, grid refinement factors $r_{12} = h_2/h_1$ and $r_{23} = h_3/h_2$ are calculated in order to find the observed order. For a constant domain area, these reduce to $r_{12} = \sqrt{N_1/N_2}$ and $r_{23} = \sqrt{N_2/N_3}$. Here subscript 1 refers to the fine grid and subscript 3 the coarse grid, therefore $h_1 < h_2 < h_3$. The observed order is the obtained by solving the expression

$$\frac{\epsilon_{23}}{\epsilon_{12}} = r_{12}^p \left( \frac{r_{23}^p - 1}{r_{12}^p - 1} \right),$$

where $\epsilon_{ij} = f_j - f_i$. Equation D.1.9 is solved for $p$ with the MATLAB function \texttt{fzero}, which finds the root of a continuous function of one variable. Once the observed order is known, the extrapolated flow variable $f_\infty$ is calculated according to Eq. D.1.7. Finally, the GCI values are determined. The GCI for the fine grid is defined as

$$GCI_{FINE} = F_s \frac{|\epsilon|}{r^p - 1} \quad (D.1.10)$$

and for the coarse grid as

$$GCI_{COARSE} = r^p GCI_{FINE}$$

$$= GCI_{FINE} + F_s |\epsilon|$$

$$= F_s |\delta_{RE}|$$
where $F_s$ is a “factor of safety” over the Richardson error. Roach uses $F_s = 3$, but notes this may be overly conservative, particularly when the quality of the grid convergence study increases. Eça and Hoekstra\textsuperscript{145} use $F_s = 1.25$ and is adopted here.

Equations D.1.10 and D.1.11 are applied to a pair of grids. The fine and medium grids are compared to each other as are the medium and coarse grids. Therefore, the GCI of the fine grid is obtained by using $\epsilon_{12}$ and $r_{12}$ or $f_1$ in Eq. D.1.10. The GCI for the medium grid can be found by using $\epsilon_{23}$ and $r_{23}$ or $f_2$ in Eq. D.1.10 because the medium grid represents the fine grid between the medium grid and the coarse grid. Alternatively, $\epsilon_{12}$ and $r_{12}$ can be used in Eq. D.1.11 because the medium grid represents the coarse grid between the fine grid and medium grid. Lastly, the GCI of the fine grid is calculated using $\epsilon_{23}$ and $r_{23}$ or $f_3$ in Eq. D.1.11. These GCI values are used to create error bars around each solution. If the value of $f_\infty$ falls within the error bar of any of the coarser solutions, it may be used in place of the fine solution. This can save computational time if many solutions are required.

## D.2 Least Squares GCI

When more than three solutions can feasibly be obtained, it can be advantageous to use the least squares formulation of the GCI, developed by Eça and Hoekstra.\textsuperscript{145} This allows more than three grids to be used and can therefore yield a more accurate error estimate, particularly when the data exhibits some degree of scatter. The least
squares GCI is based on a one-term series expansion of the discretization error

$$\delta_{RE} = f_i - f_\infty \cong \alpha h_i^p$$  \hspace{1cm} (D.2.1)

The least squares formulation is based on minimizing the function

$$S(f_\infty, \alpha, p) = \sqrt{\sum_{i=1}^{N_g} [f_i - (f_\infty + \alpha h_i^p)]^2}$$  \hspace{1cm} (D.2.2)

Setting the derivatives of $S$ with respect to $f_\infty$, $\alpha$, and $p$ equal to zero yields

$$\sum_{i=1}^{N_g} f_i h_i^p \log(h_i) - f_\infty \sum_{i=1}^{N_g} h_i^p \log(h_i) - \alpha \sum_{i=1}^{N_g} h_i^{2p} \log(h_i) = 0$$  \hspace{1cm} (D.2.3)

where

$$f_\infty = \frac{1}{N_g} \left( \sum_{i=1}^{N_g} f_i - \alpha \sum_{i=1}^{N_g} h_i^p \right)$$  \hspace{1cm} (D.2.4a)

$$\alpha = \frac{N_g \sum_{i=1}^{N_g} f_i h_i^p - \left( \sum_{i=1}^{N_g} f_i \right) \left( \sum_{i=1}^{N_g} h_i^p \right)}{N_g \sum_{i=1}^{N_g} h_i^{2p} - \left( \sum_{i=1}^{N_g} h_i^p \right) \left( \sum_{i=1}^{N_g} h_i^p \right)}$$  \hspace{1cm} (D.2.4b)

The observed order is calculated by solving Eq. D.2.3. In the case that the observed order is greater than the theoretical order (super-convergence), Eça and Hoekstra\textsuperscript{145} note that GCI values calculated with the observed order are not reliable and proposes limiting the observed order to the theoretical order. They also introduce a measure of uncertainty in the error estimate caused by scatter in the data.

$$U_s = \sqrt{\frac{\sum_{i=1}^{N_g} [f_i - (f_\infty + \alpha h_i^p)]^2}{N_g - 1}}$$  \hspace{1cm} (D.2.5)

This is added to the GCI to yield the discretization uncertainty. Eça and Hoekstra\textsuperscript{146} put further criteria on calculating the numerical uncertainty with the least
squares approach. For a nominally second-order accurate method with monotonic convergence defined by \((f_2 - f_1) \times (f_3 - f_2) > 0\)

\[
U_d = \begin{cases} 
1.25\delta_{RE} + U_s & \text{if } 0.95 \leq p < 2.05 \\
\min (1.25\delta_{RE} + U_s, 1.25\Delta M) & \text{if } 0 < p < 0.95 \\
\max (1.25\delta'_{RE} + U_s, 1.25\Delta M) & \text{if } p \geq 2.05 
\end{cases} \tag{D.2.6}
\]

where \(\Delta M\) is the maximum difference between all the solutions available and \(\delta'_{RE}\) is the Richardson error calculated using the theoretical order of 2.

\[
\delta'_{RE} = f_i - f_\infty = \alpha h_i^2 
\tag{D.2.7}
\]

If monotonic convergence is not observed, the discretization uncertainty is expressed simply as three times the range of the solutions.

\[
U_d = 3\Delta M \tag{D.2.8}
\]

**D.3 Two-Term Series Expansion**

Salas\textsuperscript{147} suggests that a one-term expansion on the discretization error may not always be valid and a two-term expansion is sometimes required. A two-term expansion can be represented as

\[
f_\infty = f + \alpha h_x^p + \beta h_y^p + \text{H.O.T.} \tag{D.3.1}
\]

where \(h_x\) and \(h_y\) are the grid spacing in the \(x\) and \(y\) directions, respectively. For \(h_x\) and \(h_y\) sufficiently small this can be simplified to

\[
f_\infty = f + \alpha h_x^p + \beta h_y^p \tag{D.3.2}
\]

The \(x\) and \(y\) grid spacings are commonly combined and used in the one-term expansion by defining \(h = \sqrt{h_x h_y}\). This defines the grid spacing as the side of a
square with area equal to a rectangle with sides $h_x$ and $h_y$. Salas notes $h$ could also be defined as the diagonal $h = \sqrt{h_x^2 + h_y^2}$. Using the one-term expansion has the advantage of there being three unknowns instead of four as in the two-term expansion. However, this advantage comes with a caveat. Using a one-term expansion requires that the grid spacings be related by

$$\frac{h_{y,i}}{h_{x,i}} = r \quad \text{for} \quad \forall i, i = 1, 2, 3, \ldots \quad (D.3.3)$$

where the ratio $r$ is constant over all grids, which is not always convenient to ensure.

Therefore, a least squares formulation of the two-term expansion can be solved to account for non-constant grid refinement. This can be accomplished with the Gauss-Newton method, which minimizes the sum of the squares

$$S(x) = \sum_{i=1}^{m} y_i^2(x) \quad (D.3.4)$$

Starting with an initial guess $x_0$, $x$ is updated according to

$$x_{k+1} = x_k + \Delta \quad (D.3.5)$$

where the increment $\Delta$ is the solution to the normal equations

$$(J^TJ) \Delta = -J^Ty \quad (D.3.6)$$

and $J$ is the Jacobian matrix of $y$ with respect to $x$. In order to solve the two-term expansion of the discretization error, $x$ is a vector of the four unknowns $f_\infty$, $p$, $\alpha$, and $\beta$. The function $y$ can be expressed as

$$y_i = f_i + \alpha h_{x,i}^p + \beta h_{y,i}^p - f_\infty \quad (D.3.7)$$
which is equivalent to setting Eq. D.3.2 to zero. The Jacobian matrix is then comprised of the partial derivatives of $y$ with respect to the four unknowns for each solution.

\[
\begin{align*}
\frac{\partial y_i}{\partial f_\infty} &= -1 \\
\frac{\partial y_i}{\partial p} &= \alpha h_{x,i}^p \ln (h_{x,i}) + \beta h_{y,i}^p \ln (h_{y,i}) \\
\frac{\partial y_i}{\partial \alpha} &= h_{x,i}^p \\
\frac{\partial y_i}{\partial \beta} &= h_{y,i}^p
\end{align*}
\] (D.3.8)

Iterating on this procedure will yield the extrapolated value $f_\infty$ and the observed order $p$. The discretization error is then calculated as before.
Bibliography


sign Concepts and Optimization in Engineering Sciences, edited by G. S. Duklikravich, University of Texas at Austin, Austin, TX, 1987, pp. 487–505.


Collection of Technical Papers. Pt. 2, No. A95-36501, AIAA, Washington, DC,
Fluid Dynamics Conference and Open Forum, San Diego, CA, June 19-22.

AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and
Optimization, Panama City Beach, FL, Sept. 7-9.


pp. 1162–1170.


164


