

OPTIMAL SHAPE DESIGN USING AN
UNSTEADY CONTINUOUS ADJOINT APPROACH

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF
AERONAUTICS AND ASTRONAUTICS
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Thomas D. Economon

August 2014

© 2014 by Thomas David Economon. All Rights Reserved.
Re-distributed by Stanford University under license with the author.



This work is licensed under a Creative Commons Attribution-Noncommercial 3.0 United States License.

<http://creativecommons.org/licenses/by-nc/3.0/us/>

This dissertation is online at: <http://purl.stanford.edu/tg620qm8104>

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Juan Alonso, Primary Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Antony Jameson

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Robert MacCormack

Approved for the Stanford University Committee on Graduate Studies.

Patricia J. Gumport, Vice Provost for Graduate Education

This signature page was generated electronically upon submission of this dissertation in electronic format. An original signed hard copy of the signature page is on file in University Archives.

Abstract

Many practical flows of aerodynamic interest are unsteady in nature, and between increases in computing power and advanced algorithms, accurately predicting and designing for the performance of aerospace systems in an unsteady environment is becoming more tractable. Several examples of engineering applications that could immediately benefit from a truly time-accurate design approach are open rotors, rotorcraft, turbomachinery, wind turbines, maneuvering flight, or flapping flight, to name a few. An unsteady treatment of these flows will also directly enable multidisciplinary design, analysis, and optimization involving other time-dependent physics associated with these systems, such as their structural or acoustic responses. Consequently, new unsteady design methodologies will enable the design of next generation aerospace vehicles with reduced fuel burn, emissions, and noise or rotating machinery for meeting future propulsion and energy challenges.

This dissertation presents the development and application of a new, unsteady continuous adjoint formulation for optimal shape design. The arbitrary Lagrangian-Eulerian (ALE) form of the unsteady, compressible Reynolds-averaged Navier-Stokes (RANS) equations with a generic source term is considered, and from these governing flow equations, an adjoint formulation centered around finding surface sensitivities using shape calculus is derived. This surface formulation provides the gradient information necessary for performing gradient-based aerodynamic shape optimization. To analyze the effectiveness of the methodology, gradients provided by the continuous adjoint and finite differencing approaches are compared. Optimal shape design is demonstrated in both two and three dimensions for a range of pitching and rotating applications.

To my mother
Aileen Economon
1955 - 2009

Acknowledgement

I have been extremely fortunate to be supported through external fellowships throughout the majority of graduate school. These fellowships afforded me the opportunity to pursue the topics of most interest to me, which led me to the research problems treated in this dissertation. This type of academic freedom has been invaluable in building my skill set as an independent researcher, and I am grateful to the National Science Foundation and the Department of Defense for their support.

I would like to thank the members of my dissertation committee for sharing their time and insights: Prof. Michael Saunders, Prof. Sanjiva Lele, Prof. Robert MacCormack, and Prof. Antony Jameson. I had the pleasure of taking my first CFD course with Prof. MacCormack, and he sparked a love for the topic that propels me forward. Prof. Jameson is owed a special thanks for pioneering the adjoint-based methods for design in aeronautics upon which this dissertation is built.

Much of this dissertation took shape with the guidance and input of Dr. Francisco Palacios, and he has been a mentor and friend over the past several years. My advisor, Prof. Juan J. Alonso, has opened the door to new ideas and played a big role in my professional development. I have appreciated the freedom he granted me for investigating my academic interests, as well as the gentle nudges to keep me on track. I leave each meeting with Francisco and Juan feeling optimistic about research and energized: their infectious enthusiasm and unwavering support create a positive environment and inspire productivity. I hope for a long career full of collaboration and friendship with them.

It has been wonderful to call the Aerospace Design Lab my workplace while here at Stanford. The personalities of my lab mates and spirited conversations (work-related or otherwise) make for a daily atmosphere that is serious enough to make research contributions but light enough to laugh together. To my Stanford friends, many of whom I met in the first week here: we have had many great adventures while exploring the bay area together over the last several years, and I hope that we never stop exploring. My friends from Notre Dame have left me with my best memories during undergrad, and their friendship helps keep me motivated. I was too young to even remember how I met most of them, but my friends from home in St. Louis form a special group. I carry them with me wherever I go, and whenever I see them, we pick right back up where we left off. I can't wait to see what life has in store for all of us.

I have always looked up to my two older sisters. Undoubtedly, they have helped mold the person that I have become over the years, so in many ways, the words in this text are also their own. I

share this dissertation with them. My father has been the model of high integrity, perseverance, and strong work ethic. He has set a very high bar in that regard, one that I will always strive to meet. Finally, I am grateful to my mother, to whom this is dedicated, for telling me that life is short and reminding me to have some fun. I put my best effort toward things that I think would make her proud, and I believe this dissertation would have been one of them.

Contents

Abstract	v
Acknowledgement	vii
1 Introduction	1
1.1 Optimal Shape Design for Fluids	2
1.2 Sensitivity Analysis via the Adjoint Approach	5
1.2.1 Continuous Adjoint	5
1.2.2 Discrete Adjoint	7
1.3 Design Optimization in Unsteady Flows	8
1.4 Dissertation Layout	11
1.5 Contributions	11
2 Governing Equations for Fluid Dynamics	13
2.1 Compressible, Unsteady Navier-Stokes Equations	13
2.2 Turbulence Modeling	16
2.2.1 Spalart-Allmaras Model	16
2.3 Compressible, Unsteady Euler Equations	17
2.4 Governing Equations in a Rotating Reference Frame	18
3 The Unsteady Continuous Adjoint Approach	21
3.1 Optimal Shape Design Problem Statement	21
3.2 Variation of the Functional	22
3.2.1 Force-based Functionals	23
3.2.2 Temperature-based Functionals	26
3.3 The Adjoint Approach to Optimal Design	27
3.4 The Linearized Governing Equations	29
3.4.1 Linearized Navier-Stokes Equations in ALE Form	29
3.4.2 Linearized Boundary Conditions	30
3.4.3 Complete System of Linearized Equations	33
3.5 The Unsteady Continuous Adjoint Equations	34

3.5.1	Obtaining the Adjoint Equations	34
3.5.2	Evaluating the Adjoint Boundary Integrals	38
3.5.3	Complete System of Adjoint Equations and Surface Sensitivities	48
4	Numerical Implementation	57
4.1	Spatial Integration via the Finite Volume Method	58
4.1.1	Discretizing the Governing Flow Equations	60
4.1.2	Discretizing the Continuous Adjoint Equations	67
4.1.3	Evaluating Gradients of the Flow and Adjoint Variables	74
4.2	Time Integration	74
4.2.1	Time-marching Schemes for the Governing Flow Equations	75
4.2.2	Time-marching Schemes for the Continuous Adjoint Equations	78
4.2.3	Convergence Acceleration	79
4.2.4	Dynamic Meshes	79
4.3	Design Variable Definition	84
4.3.1	Gradient Projection using Surface Sensitivities	85
4.3.2	Bump Function Design Variables	85
4.3.3	Free-Form Deformation Variables	86
4.4	Mesh Deformation	87
4.4.1	Spring Analogy	87
4.4.2	Linear Elasticity	88
4.5	Optimization Framework	90
5	Optimal Shape Design Applications	91
5.1	Rotating Airfoil in Inviscid Flow	91
5.2	Pitching Airfoil in Inviscid Flow	94
5.3	Pitching Airfoil in Turbulent Flow	97
5.4	Rotor in Hover	99
5.5	Wind Turbine Design	102
5.6	Pitching Wing in Inviscid Flow	103
5.7	Pitching Wing in Turbulent Flow	108
6	Conclusions and Future Directions	113
A	Jacobians	115
	Bibliography	119

List of Tables

3.1	Number of physical (P) and numerical (N) boundary conditions required for the direct and adjoint problems in three dimensions (the state vector U has five components). .	45
5.1	Objective and constraint values for the baseline and final pitching wing design. The maximum thickness is denoted by $t_{y/b}$ where the subscript gives the spanwise location of the section as a percentage of span, b	108

List of Figures

1.1	Flow chart for a typical shape optimization problem. J is the objective function, and \vec{x} is the vector of design variables. J^* and \vec{x}^* represent an optimum.	3
1.2	Pressure coefficient and surface sensitivity contours (drag objective) on the upper surface of a pitching wing at the incidence of maximum drag.	6
1.3	Drag coefficient history and upper surface pressure contour comparison for a pitching wing design.	10
2.1	Notional schematic of the flow domain Ω the boundaries Γ_∞ and S , as well as the definition of the boundary surface normals.	14
3.1	An infinitesimal shape deformation in the local surface normal direction.	22
4.1	Flow chart for a typical shape optimization problem. J is the objective function, and \vec{x} is the vector of design variables. J^* and \vec{x}^* represent an optimum.	58
4.2	Schematics showing the geometry of the primal and dual meshes.	60
4.3	Hicks-Henne shape functions for a set of equally spaced bumps.	86
4.4	An example of the FFD technique applied to the ONERA M6 wing.	87
4.5	An example of volume mesh deformation using the linear elasticity equations.	89
5.1	Details for the 2D rotating airfoil numerical experiment, the computational mesh, and solutions for the baseline geometry.	92
5.2	Comparison studies between the continuous adjoint and finite differencing for the gradient of C_d . A set of 38 Hicks-Henne bump function variables (x_i) are along the x -axis.	93
5.3	Gradient verification and a comparison of the baseline and minimum drag airfoil designs.	94
5.4	Verification, validation, and optimization results for a pitching NACA 64A010.	95
5.5	Numerical grids for the pitching NACA 64A010 calculations.	96
5.6	Numerical results for a pitching NACA 64A010 in turbulent flow.	97
5.7	Force coefficient histories, shape comparison, and optimization history for the pitching airfoil design.	98
5.8	Mesh and FFD box details for the Caradonna and Tung numerical experiment.	99

5.9	C_p contours on the upper surface of the baseline rotor geometry along with a comparison to experiment at multiple span locations. The blade tip is on the right, which is rotating toward the bottom of the page. The surface sensitivity contours for a torque objective function are also shown. Note the high sensitivity to shape deformations in the vicinity of the shock.	100
5.10	Gradient verification using the FFD control point variables and optimization results.	101
5.11	Comparison of the baseline and optimized rotor geometries along with C_p contours. The strong shock has been removed due to a distinct change in the tip shape.	102
5.12	C_p distributions at multiple radial blade stations compared with experimental data.	103
5.13	Pressure coefficient, surface sensitivity, and shape modifications for the NREL Phase VI wind turbine.	104
5.14	Surface mesh for the inviscid case and FFD box for the pitching ONERA M6 numerical experiment.	105
5.15	Pressure coefficient (top row) and surface sensitivity contours (drag objective function, bottom row) for three time instances during one period of oscillation. The figures in the left column correspond to the minimum drag incidence, those in the middle column are near the mean angle of incidence, and those on the right correspond to the incidence of maximum drag.	106
5.16	Force coefficient histories, gradient comparison, and optimization history for the inviscid pitching wing design.	107
5.17	Pressure coefficient contour comparison between the ONERA M6 and the final design at the incidence of maximum drag for the inviscid case.	109
5.18	Shape comparison of the baseline (dotted) and final (solid) inviscid wing sections.	109
5.19	Surface mesh topology showing the wing surface and symmetry plane for the URANS case.	110
5.20	Force coefficient histories, section shape comparison, and optimization history for the pitching wing design in turbulent flow.	111
5.21	Pressure coefficient contour comparison between the ONERA M6 and the final design at the incidence of maximum drag for the URANS case.	112

Chapter 1

Introduction

The engineering design process provides a systematic procedure for the creation of a product that meets the user's specification or needs. Once a feasible design for a particular product or system has been achieved, a natural question is then how to make it better, or the best, for its intended purpose. The ultimate goal of engineering design problems is to create an optimized system.

Optimizing the performance of an engineering system is often accomplished by hand: designers iterate through prototypes, implement design changes through expert intuition, or test new ideas and measure performance using physical experiment. However, due to its increasing maturity and cost benefits over physical experiment, computational simulation is becoming more prevalent for analyzing these systems. As a direct consequence, it has enabled simulation-based design where computational tools are combined to automate the design process. This type of “automatic” design has tremendous potential, but it should be viewed as a tool to aid the designer when problems become complex enough to elude intuition.

The performance of an engineering system can often be evaluated based on its size and shape. The field of Optimal Shape Design (OSD) is concerned with finding the optimal shape for the object or surface that improves its performance. The performance metric will be defined by the objective of the designer, and this objective function will be minimized or maximized, subject to certain constraints, by changing the shape of the object. In the context of aeronautics, a typical example might be finding the shape of an aircraft wing with minimum drag while maintaining the required lift to stay aloft and obeying any geometric constraints on the wing (such as structural sizing or internal fuel volume requirements). In this example, the optimal shape of the wing is one with the minimum drag that satisfies the constraints.

Much of the progress in OSD can also be attributed to the aforementioned increased use and maturity of computational simulation and in particular, the numerical solution of Partial Differential Equations (PDEs) on computational grids. The development of Computational Fluid Dynamics (CFD) serves as an exceptional example. Over the last half-century, researchers have developed numerical methods for solving problems across the entire range of approximation for fluids in a continuum, starting from simpler models such as the potential flow equations all the way up to

the current state of the art using the Reynolds-averaged Navier-Stokes equations (RANS). Higher-fidelity methods, such as Large Eddy Simulation (LES) and Direct Numerical Simulation (DNS), are currently active research areas as well, although their computational cost is significantly higher. As they become available, these new CFD methods can be included in the OSD process and enable the treatment of more complex problems.

OSD is a broad and maturing field, appearing in multiple industries (aerospace, automotive, medical devices, to name a few) and spanning an array of disciplines, such as fluids, structures, or acoustics, for example. Today, shape design problems are mainly solved using automatic techniques on computers that leverage the latest advances in topics such as the numerical solution of governing equations describing particular physics, numerical optimization, sensitivity analysis in the case of gradient-based optimization, geometry and mesh definitions, and design variable parameterization.

OSD is simultaneously a mathematically- and application-rich topic. Mathematical rigor concerning many areas (optimal control theory, PDEs, numerical optimization, vector calculus, differential geometry) is required, but there is also a need for a translation from the purely mathematical to the applied, such that OSD can gain further penetration into industrial practice for realistic problems of engineering interest.

1.1 Optimal Shape Design for Fluids

The shape optimization of systems whose performance is governed by the equations of fluid dynamics can be considered a subset of OSD. The goal in this scenario is to compute the forces or heat transfer on a body (wing, aircraft, automobile, etc.) from the numerical solution of the governing fluid equations on a computational grid occupying the volume surrounding the geometry and to use this information to improve the design in an iterative fashion. CFD is therefore a primary tool in the shape design process.

Efficient algorithms for CFD have allowed the aerospace industry to perform large numbers of simulations for rapidly evaluating candidate designs. Other developments, such as unstructured mesh technology, have enabled the analysis and design of increasingly complex geometries. Given the shorter turnaround times afforded by advances in CFD, applying OSD to large-scale problems of industrial interest with complex geometry or in other costly situations, such as for unsteady flows, is becoming more tractable.

The solution of the RANS equations is now an everyday occurrence in the aerospace industry, and RANS will likely remain the workhorse for analysis and design for the foreseeable future while higher-fidelity methods continue to mature along with further increases in computing power. Given current computational resources, including LES or DNS within the design process for practical problems is infeasible. While aerodynamic design using the steady RANS equations is fairly well established, there exists a growing interest in moving beyond steady problems, so that time-accurate physics can be taken into account during the design process. These problems may include moving surfaces at high Reynolds numbers, such as rotating rotor blades or maneuvering wings, and require the unsteady, compressible RANS equations in arbitrary Lagrangian-Eulerian (ALE) form for accurate

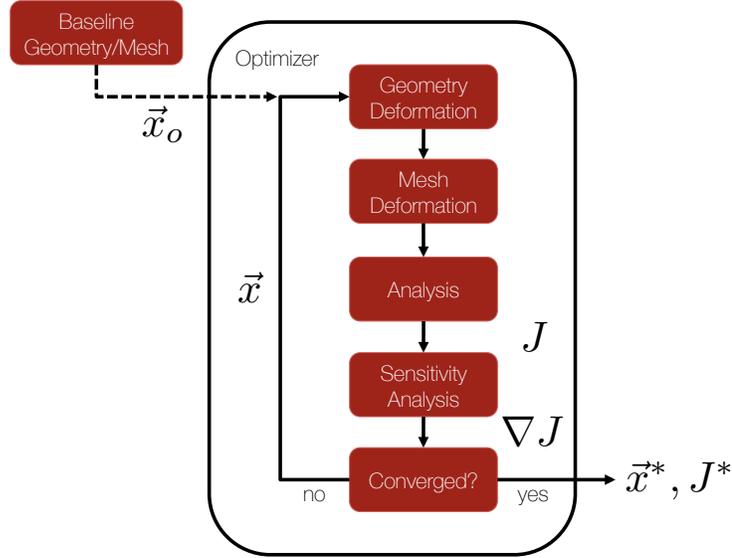


Figure 1.1: Flow chart for a typical shape optimization problem. J is the objective function, and \vec{x} is the vector of design variables. J^* and \vec{x}^* represent an optimum.

analysis. OSD techniques for unsteady and turbulent flows have been highlighted previously as unsolved problems [84], and although some progress has been made, there remains a need to derive and refine new, efficient techniques for unsteady problems with surfaces in motion.

Due to the high computational cost of solving the nonlinear flow equations, especially in unsteady situations, gradient-based optimization is an attractive option. As an example, consider again the design of a commercial aircraft wing for minimum drag with a lift constraint with respect to design parameters such as the sweep, taper, and twist of the wing. In this situation, a basic framework for optimal shape design might become:

1. Apply the current design variable state (sweep, twist, and taper) to the wing geometry.
2. Deform the existing volume mesh, or create a new mesh, around the updated surface shape.
3. Compute the objective and constraint values (drag and lift) by solving the flow equations.
4. Compute the gradient of the objective and constraints with respect to the design variables.
5. Provide the objective, constraint, and gradient values to a suitable optimization method for determining an updated design variable state.
6. Repeat previous steps until an optimum is found.

A representative flow chart for this process can be found in Fig. 4.1. The steps above form a type of segregated approach where the various components of the problem are executed independently

and placed in a design cycle with an optimization method, such as steepest descent or conjugate gradient. Note that other approaches where the flow, gradient, and optimality conditions are solved simultaneously, often called one-shot approaches [119], are possible.

Consider now the individual components required to perform the design cycle described above. Geometric representation and mesh generation are important components, but as these tools are also needed for basic analysis, many external packages, such as Computer-aided Design (CAD) programs and meshing software, are readily available. As they are used in many areas unrelated to fluids, numerous numerical optimization methods are also well established. Apart from needing to compute the objective function by solving the flow equations, we must perform sensitivity analysis by a suitable method in order to compute the gradients that are needed for optimization. The choice of method for sensitivity analysis is a crucial one that should garner much attention, as it will have a large impact on the overall efficiency of the shape optimization process.

One straightforward method for sensitivity analysis is the finite difference method. This brute-force approach for computing gradients can be applied in a simple fashion by taking the difference of the objective function value from a flow solution for the baseline geometric configuration and the new value of the objective function after applying perturbations to a chosen design variable, and finally dividing by the perturbation step size. In this fashion, the entire gradient vector can be computed by perturbing each design variable individually. This approach is costly for situations with a large design space, as computing the gradient for a problem with N design variables will require $N + 1$ flow solutions. Furthermore, the finite difference approach is sensitive to the choice of step size: a large step size may lead to unacceptable truncation error in the finite difference calculation, whereas a very small step size will suffer from numerical precision issues in the form of subtractive cancellation. The accuracy issues can be eliminated by formulating the finite difference problem in terms of a design variable perturbation in the complex plane [75]. However, the cost of the complex-step derivative approximation still depends on the number of design variables, and complex arithmetic can result in nearly double the overall cost of finite differencing. The exact gradients provided by the complex-step method have proven incredibly useful as a tool for verifying other gradient calculation techniques. An analogue for computing exact higher-order derivatives was recently developed by Fike [36], and it has been used to enhance optimization methods where injecting Hessian information proves useful.

Unfortunately, it is often the case in aerodynamic design that the design space is quite large. For instance, in order to suitably parameterize a full aircraft or vehicle configuration, a design space containing $\mathcal{O}(100)$ design variables (or more) is a regular occurrence. Given the high computational cost of function evaluations from CFD, a finite difference-type approach is not practical for these problems, and ideally, a sensitivity analysis technique that is independent of the number of design parameters would be employed. One such technique is the adjoint approach, and it is the main topic of this dissertation.

As an analytic method, the adjoint approach takes advantage of our knowledge of the governing equations, and more specifically their linearization, in order to extract sensitivity information. Obtaining the adjoint of a governing system is typically demonstrated through the use of Lagrange

multipliers, and the result is a linear problem of similar complexity to the original governing equations that, when solved, provides the sensitivity of a chosen objective function with respect to an arbitrary number of design parameters. Note that, while the cost of computing a gradient becomes independent of the number of design variables, a separate adjoint system must be solved for each objective function or constraint in a particular optimization problem.

1.2 Sensitivity Analysis via the Adjoint Approach

Formally, the groundwork for applying the adjoint approach can be traced back to the work of Lions [72] concerning the optimal control of systems governed by partial differential equations. Pironneau applied variational methods from optimal control to solve several drag minimization problems in incompressible, laminar flow [103] and other shape design problems for elliptic systems [104]. A framework for computing the gradient of an objective function analytically using the co-state, or adjoint, vector was demonstrated here in order to avoid the costly finite difference approach. Much of the theoretical basis for the adjoint method was also explored, including optimization techniques and optimality conditions, and consideration was given to numerical solutions on computers using the finite element method.

For problems in aeronautics, Jameson first introduced the adjoint approach for shape design in transonic flows to reduce shock-induced drag [50, 51]. Further pioneering work by Jameson et al. [55, 60] treated the Euler and Navier-Stokes equations in three dimensions, thus enabling the design of complete aircraft configurations [57].

Adjoint formulations as a means of sensitivity analysis have since become the subject of a rich volume of research literature. Many advances and extensions have been made, and the effectiveness of these formulations for use in aerodynamic design, especially for steady problems, is well established. While we are interested in shape optimization in this work, researchers continue to find new applications for adjoint sensitivities, such as error estimation, uncertainty quantification, or mesh adaptation [42, 35, 71, 99].

Adjoint formulations are now typically classified by their derivation as either continuous (the governing equations are first linearized then the result is discretized) or discrete (the governing equations are first discretized and the result is linearized). The merits for each approach have been discussed at length in literature, and they have been compared directly by several authors [66, 41, 87]. While it is generally agreed that both approaches are equally useful sensitivity analysis techniques for aerodynamic shape design, the details of their implementation result in advantages and disadvantages under different circumstances. A brief overview of the continuous and discrete adjoint approaches is given below.

1.2.1 Continuous Adjoint

The early work by Lions, Pironneau, and Jameson that is described above was continuous in nature, i.e., analytic expressions for the gradient of a functional were obtained through the use of variational

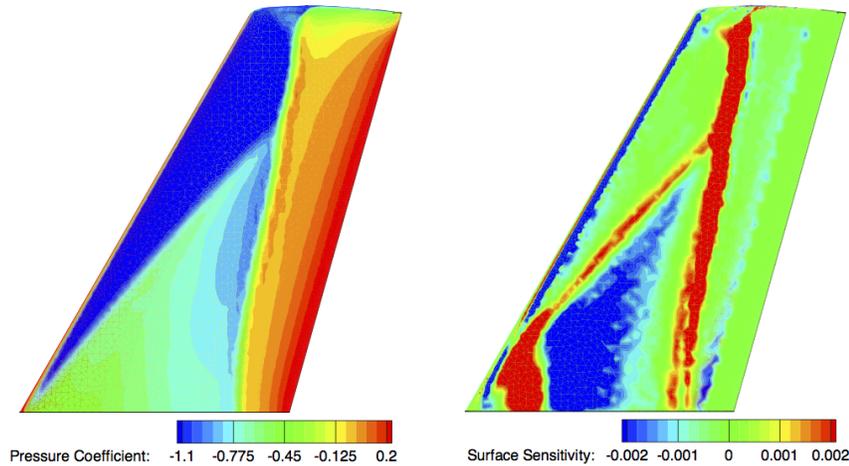


Figure 1.2: Pressure coefficient and surface sensitivity contours (drag objective) on the upper surface of a pitching wing at the incidence of maximum drag.

methods. Following the continuous approach results in a set of continuous PDEs and boundary conditions for the adjoint system that must be solved as an additional requirement in order to compute the gradient. The ability to study a continuous PDE for the adjoint can offer the advantage of physical insight into the character of the governing flow and adjoint systems, as well as flexibility in the choice of solution method. This insight can aid in composing well-behaved numerical methods that are tailored to the adjoint equations.

An additional advantage of the continuous approach is the ability to recover an analytic expression for the gradient of a functional as an integral over the surface to be designed, which is often called a *surface formulation* for computing gradients. A surface formulation of this type has no dependence on volume mesh sensitivities (in a continuous sense, movement of interior mesh nodes should not affect the behavior at the surface). After computing the adjoint solution in the domain, the gradient can be calculated with almost negligible computational cost by evaluating the analytic expression for the sensitivity, or the *surface sensitivity*, at each node and integrating over the surface. The surface sensitivity provides a measure of the variation of the objective function with respect to infinitesimal variations of the surface shape in the direction of the local surface normal.

As noted even in early work by Pironneau [104], the closed-form expression for sensitivity can also provide designer intuition by showing engineers exactly where it is best to modify the shape of their designs. Furthermore, these surface sensitivities can be mapped onto a surface to give a visual representation that can help designers make rapid changes or define appropriate design variables for further optimization. An example of a surface sensitivity map for a pitching wing in transonic flow is presented in Fig. 1.2.

While initial results for the continuous adjoint were calculated on structured meshes, numerical methods for the continuous adjoint equations were extended to unstructured meshes for complex

geometries. Anderson and Venkatakrishnan [3] demonstrated one of the first surface formulations for the Navier-Stokes equations on unstructured meshes, but they noted that higher-order derivative terms appearing in the adjoint equations caused issues related to implementation and accuracy. Jameson and Kim [58] arrived at a similar surface formulation for the Euler equations by eliminating volume integral terms from the gradient formula in the continuous limit. Castro et al. [18] derived continuous adjoints for inviscid and laminar flows on unstructured grids with a special emphasis on simplifications and the reduction of higher-order derivative terms with the aid of differential geometry formulas (shape calculus) [115, 118]. The resulting surface formulation alleviated many of the issues related to implementation and accuracy for unstructured meshes, and it has been successfully applied on three-dimensional meshes for a range of applications [97, 112]. Bueno-Orovio et al. [14] later extended the formulation to turbulent flows by including the Spalart-Allmaras (S-A) turbulence model, which represented the first such formulation for compressible flows.

Despite many advantages, continuous adjoint approaches can suffer from issues related to their derivation and formulation. Depending on the form of the governing equations and the choice of objective functions, the mathematical manipulations required to arrive at the continuous adjoint system may be quite complicated or even impossible. The continuous approach does not allow for the choice of arbitrary functionals, as only certain terms can be canceled on the solid boundary surface in order to arrive at a consistent adjoint system [4]. Special considerations may be required for handling discontinuities such as shocks [5] or other non-differentiable terms. Deriving consistent boundary conditions and expressions for the surface sensitivity that accompany the continuous adjoint equations can also be difficult, and unfortunately, clear strategies for their derivation are less prevalent in the literature [40, 46]. Once derived, however, the continuous adjoint approach can be easily implemented within existing solvers while leveraging many of the same numerical methods used for the flow problem.

1.2.2 Discrete Adjoint

In the discrete adjoint approach, the flow equations and objective function are first discretized and then linearized in order to form the adjoint equations. Therefore, the result of a discrete adjoint derivation is a linear system for the unknown adjoint state that involves the Jacobian of the discretized governing equations and a right-hand side based on the chosen objective function. The discrete adjoint can be derived in several ways, but the concept of Lagrange multipliers is often used, which is similar to the continuous approach.

The sensitivity information provided by the discrete adjoint is numerically exact, i.e., it gives the exact gradient of the discretized objective function. In this regard, discrete adjoint sensitivities are compatible with finite difference or complex-step sensitivities. In fact, the complex-step method is regularly used in literature to verify discrete adjoint sensitivities to near machine precision. This numerical consistency between the objective function and the gradient has the potential to help optimizers find minima more easily, as opposed to the continuous adjoint approach, which offers an inexact gradient of the discrete functional.

Soon after the work of Jameson on the continuous adjoint for compressible flows, numerous

researchers began investigating discrete adjoint formulations. Frank and Shubin [37] first applied the discrete adjoint approach for quasi-one-dimensional flow. Burgreen and Baysal [15] and also Elliott and Peraire [33] performed shape design using discrete sensitivities from the Euler equations in three dimensions. A discrete adjoint for turbulent flows on unstructured meshes in two dimensions was presented by Anderson and Bonhaus [2] for airfoil design. Nadarajah [86] developed the discrete adjoint for the Euler and Navier-Stokes equations based on the schemes of Jameson.

Unfortunately, the discrete approach can become increasingly complex for higher-order schemes with large stencils, and the memory required for storing the discrete Jacobians can also be a bottleneck. Many strategies for avoiding memory overhead or for simplifying the linearization by excluding terms at the cost of accuracy in the resulting gradient have been studied [38, 91, 39, 95, 26, 80]. A more comprehensive review of the various strategies employed for alleviating performance issues and memory requirements for discrete adjoints has been prepared by Peter and Dwight [101].

Algorithmic differentiation (AD) is an approach for automatically computing derivatives using software tools that operate directly on the source code representation of the discretized equations [44, 20]. In recent years, the use of AD has become more widespread, and it has been demonstrated as another option for obtaining the Jacobians of the discretized governing equations that are needed to form the discrete adjoint [73]. AD has also been applied to entire flow solvers for obtaining a discrete adjoint solver rather than in a piecemeal fashion for the various Jacobian components. However, The application of AD in this manner can result in prohibitively large memory requirements or code runtime, unless given particular attention [17].

1.3 Design Optimization in Unsteady Flows

Many practical flows of aerodynamic interest are unsteady in nature, and with the increasing power of computational resources and advanced algorithms, accurately predicting and designing for the performance of aerospace systems in an unsteady environment is becoming more tractable and more of a necessity. Several examples of engineering applications that could immediately benefit from a truly time-accurate design approach are open rotors, rotorcraft, turbomachinery, wind turbines, maneuvering flight, or flapping flight, to name a few. An unsteady treatment of these flows will also directly enable multidisciplinary design, analysis, and optimization involving other time-dependent physics associated with these systems, such as their structural or acoustic responses. Consequently, these new unsteady methodologies will enable the design of next-generation aerospace vehicles with reduced fuel burn, emissions, and noise or rotating machinery for meeting future propulsion and energy challenges.

Computational cost is paramount for design in unsteady flows. Due to the increased cost of time-accurate simulations, efficient methods for computing sensitivity information are a must. The adjoint approach is again an appealing option, as its computational cost is independent of the number of design variables. However, adjoint formulations for unsteady problems are less common and more challenging due to the potentially prohibitive storage requirements associated with managing the time-accurate solution data that is needed for the solution of the corresponding unsteady adjoint

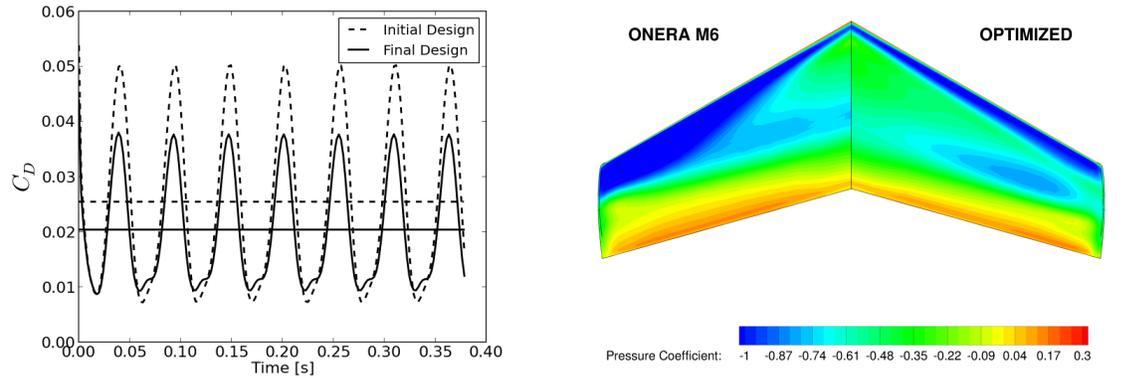
equations.

Moreover, the engineering applications mentioned above also involve moving aerodynamic surfaces, and this motion must be taken into account by the governing flow equations (including the boundary conditions) and subsequently, by the adjoint equations. As previously mentioned, solving the governing equations in ALE form addresses this issue, but it adds an additional layer of complexity to the problem, as the motion of the surface and surrounding volume mesh must be treated in time. The derivation of a continuous adjoint based on the ALE form of the equations requires consideration of the dynamic surfaces and meshes, and the terms related to the motion of the domain appear explicitly in the adjoint system, boundary conditions, and expressions for the surface sensitivities.

Despite the challenges, recent work demonstrating the viability of unsteady adjoint approaches across a range of applications suggests a growing interest in and capability for design in unsteady flows. Nadarajah and Jameson [88] performed shape design for pitching airfoils using the Euler equations with both continuous and discrete adjoints and compared the unsteady approach to multi-point design. Rumpfkiel and Zingg [108] used a discrete adjoint formulation for the control of unsteady flows in two dimensions, including drag minimization for flows past bluff bodies and inverse design of a multi-element airfoil for noise. Mavriplis and Mani [81, 74] formulated an unsteady, discrete adjoint for turbulent flows on dynamically deforming unstructured meshes in both two and three dimensions. More recently, Nielsen et al. [93, 92] have demonstrated an unsteady, discrete adjoint approach for design in turbulent flows on dynamic, possibly overset, deforming meshes. Economon et al. [29] investigated an unsteady continuous adjoint for inviscid flows around pitching airfoils on meshes with sliding mesh interfaces.

In certain situations, complementary approaches are available to help reduce the cost or complexity of the problem. For some rotating applications, the governing flow equations can be recast into a rotating frame of reference moving with the body. This transformation allows for the steady solution of a problem that was unsteady in the inertial frame, and consequently, it can considerably reduce the computational cost of these calculations. However, the rotational speed of the surface and volume mesh must still be accounted for in this formulation, along with an additional source term in the momentum equations. Several publications have addressed adjoint-based shape design using this form of the equations. Lee and Kwon [68] presented a continuous adjoint formulation for inviscid, hovering rotor flows on unstructured meshes. Discrete adjoint formulations for the RANS equations in a rotating frame have been shown by Nielsen et al. [94] with the Spalart-Allmaras turbulence model on unstructured meshes and by Dumont et al. [24] with the $k - \omega$ turbulence model and the shear stress transport correction on structured meshes. Economon et al. [28, 30] have shown both inviscid and viscous continuous adjoint formulations in a rotating frame, which form a subset of the results within this dissertation.

Another complementary approach for unsteady problems with inherent time-periodicity is the time-spectral method [43], or similarly non-linear frequency domain methods [45, 83]. These approaches allow for the solution of a periodic steady state directly by introducing the periodicity explicitly for the discretization of the time derivative term in the flow equations. Trading memory



(a) C_D history for the initial and final pitching wing designs. The average values are also shown as horizontal lines. The average drag is greatly reduced for the final design.

(b) Pressure coefficient comparison between the ONERA M6 and the final design at the incidence of maximum drag for the URANS case.

Figure 1.3: Drag coefficient history and upper surface pressure contour comparison for a pitching wing design.

cost for calculation time, the time-spectral methods greatly reduces the overall computational cost for achieving a periodic steady state. Adjoint approaches for these periodic methods have been effectively employed for design, including the design of helicopter rotors [19, 90, 89].

As detailed above, a large amount of the previous work related to unsteady adjoints has been discrete in nature, and while a discrete adjoint approach can often be more straightforward to implement, especially if AD is available, this dissertation presents advances in the continuous approach. Flow unsteadiness, the motion of solid walls, or the presence of source terms in the governing equations can complicate matters, but the appeal of obtaining a surface formulation for shape design gradients (without a dependence on volume mesh sensitivities) and the ability to tailor numerical solution methods for the adjoint equations (to help mitigate numerical stiffness and other convergence issues while avoiding memory overhead) make the continuous adjoint approach particularly attractive.

This dissertation provides detailed derivations of the continuous adjoint equations, their admissible boundary conditions, and the expressions for surface sensitivity in a formulaic structure for multiple objective functions and governing equation systems. In particular, the goal is to derive and present a new continuous adjoint surface formulation that is widely applicable by treating the compressible, unsteady RANS equations while allowing for dynamic surfaces and the possibility of source terms. In this manner, a new set of interesting engineering problems in unsteady flows can be addressed using the continuous adjoint. In particular, shape design problems for rotating and

pitching surfaces are investigated in this dissertation, including the design of a pitching wing in transonic, turbulent flow for minimum drag with lift and section thickness constraints. The continuous adjoint formulation is successfully demonstrated for these test cases, and a sample of the results for the pitching wing design are shown in Fig. 1.3.

From the general scenario of viscous, unsteady flow, the corresponding adjoint formulations for inviscid, rotating frame, or even steady problems can be immediately recovered from the general framework. Moreover, as the unsteady continuous adjoint equations are a system of PDEs, they can be discretized in time using any approach (just as in space), which offers even more flexibility. For example, the equations can be immediately discretized with a time-spectral operator to give a time-spectral adjoint approach. This and other discretization strategies are currently under investigation and are included as future work.

1.4 Dissertation Layout

Chapter 2 briefly overviews the unsteady, compressible RANS equations, including the accompanying boundary conditions and turbulence modeling. Chapter 3 contains a detailed derivation of the unsteady continuous adjoint formulation for computing surface sensitivities. Chapter 4 details the numerical implementation of the components needed for automatic shape design: numerical methods for PDE analysis, geometry parameterization (design variable definition), mesh deformation, and the optimization framework. Chapter 5 presents results for two- and three-dimensional optimal shape design demonstrations for a range of problems governed by the non-inertial or unsteady flow equations. Lastly, Chapter 6 summarizes the main conclusions of the dissertation and discusses avenues for future research related to the unsteady adjoint formulation.

1.5 Contributions

The contributions of this dissertation are centered around the detailed derivation, implementation, and application of a new unsteady continuous adjoint formulation for aerodynamic design on dynamic meshes. A general framework for the derivation enables the immediate recovery of adjoint formulations for different scenarios, such as for design in unsteady inviscid flow, flow in rotating reference frames, or steady flow.

More specifically and to the author's knowledge, the first continuous adjoint surface formulation based on shape calculus for the unsteady, compressible RANS equations in ALE form with a generic source term is presented. Emphasis is placed on the simplification of terms using differential geometry, vector calculus, and information from the original governing equations such that the resulting expressions can be easily implemented numerically. The derivation is complete with the accompanying boundary conditions and surface sensitivity expressions for various functionals and adjustments to the governing equations. As a direct consequence, the first viscous continuous adjoint surface formulation for the governing equations expressed in a rotating frame is also recovered.

A shape design framework has been implemented within an open-source software suite for the numerical solution of PDEs and PDE-constrained optimization problems on general, unstructured meshes. The core of the suite is a three-dimensional, finite volume solver within which the ALE form of the governing equations (for solving unsteady flows on dynamic meshes) and the new unsteady adjoint formulation have been implemented. Additional elements needed for shape design, such as mesh deformation based on the linear elasticity equations, have also been incorporated into the suite. Finally, the shape design framework is demonstrated through a collection of optimal shape design examples for rotating and pitching problems in both two and three dimensions.

Chapter 2

Governing Equations for Fluid Dynamics

In aeronautics, aerodynamic performance depends on the shape of the vehicle or body, and we can predict the performance using a physical model for the fluid behavior in the presence of the vehicle. The physical model is often a set of nonlinear PDEs that govern the motion for a fluid in a continuum. By satisfying these governing equations, an engineer can predict aerodynamic performance in terms of quantities such as integrated forces on the vehicle.

Due to the fact that integrated forces are typical choices for the objective and constraints within an optimal shape design problem, we will need to consider the governing equations both for evaluating the objective or constraints and for sensitivity analysis. Therefore, the flow equations are briefly overviewed in this chapter before their use within the continuous adjoint derivation that follows in Chapter 3.

2.1 Compressible, Unsteady Navier-Stokes Equations

Consider an aerodynamic surface S immersed in a fluid represented by a domain $\Omega \subset \mathbb{R}^3$ with a disconnected boundary that is divided into a far-field component Γ_∞ and the solid wall boundary S , as seen in Fig. 2.1. The surface S represents the outer mold line of an aerodynamic body, such as a wing or a full aircraft configuration. Throughout the domain in both space and time, the behavior of the fluid is physically modeled by a particular set of coupled, nonlinear PDEs, represented by $\mathcal{R}(U) = 0$, where $U = U(\vec{x}, t)$ is the state of the fluid at a point in Ω at a given instance in time. In general, the positions of S and Ω may vary with time, which enables the analysis of aerodynamic surfaces in arbitrary motion through the fluid.

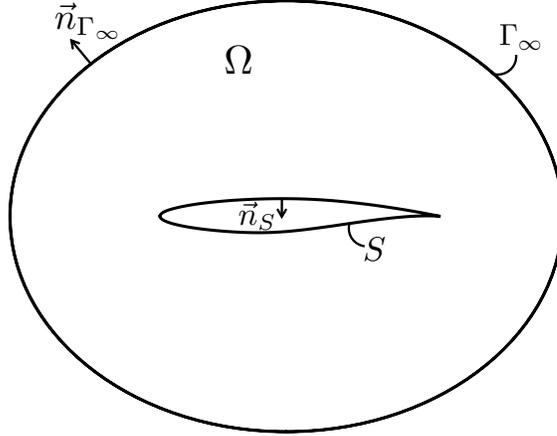


Figure 2.1: Notional schematic of the flow domain Ω the boundaries Γ_∞ and S , as well as the definition of the boundary surface normals.

The time-accurate, viscous, compressible flow around S is governed by the Navier-Stokes equations, which are statements of conservation for mass, momentum, and energy in the fluid. These conservation equations, along with a generic source term \mathcal{Q} , can be expressed in an arbitrary Lagrangian-Eulerian (ALE) [23] differential form as

$$\begin{cases} \mathcal{R}(U) = \frac{\partial U}{\partial t} + \nabla \cdot \vec{F}_{ale}^c - \nabla \cdot (\mu_{tot}^1 \vec{F}^{v1} + \mu_{tot}^2 \vec{F}^{v2}) - \mathcal{Q} = 0 & \text{in } \Omega & t > 0 \\ \vec{v} = \vec{u}_\Omega & \text{on } S \\ \partial_n T = 0 & \text{on } S \\ (W)_+ = W_\infty & \text{on } \Gamma_\infty \end{cases} \quad (2.1)$$

where the conservative variables are given by

$$U = \begin{Bmatrix} \rho \\ \rho \vec{v} \\ \rho E \end{Bmatrix}, \quad (2.2)$$

and the convective fluxes, viscous fluxes, and source term are

$$\vec{F}_{ale}^c = \begin{Bmatrix} \rho(\vec{v} - \vec{u}_\Omega) \\ \rho \vec{v} \otimes (\vec{v} - \vec{u}_\Omega) + \bar{\bar{I}} p \\ \rho E(\vec{v} - \vec{u}_\Omega) + p \vec{v} \end{Bmatrix}, \quad \vec{F}^{v1} = \begin{Bmatrix} \cdot \\ \bar{\bar{\tau}} \\ \bar{\bar{\tau}} \cdot \vec{v} \end{Bmatrix}, \quad \vec{F}^{v2} = \begin{Bmatrix} \cdot \\ \cdot \\ c_p \nabla T \end{Bmatrix}, \quad \mathcal{Q} = \begin{Bmatrix} q_\rho \\ \bar{q}_\rho \vec{v} \\ q_{\rho E} \end{Bmatrix}, \quad (2.3)$$

ρ is the fluid density, $\vec{v} = \{v_1, v_2, v_3\}^\top \in \mathbb{R}^3$ is the flow speed in a Cartesian system of reference, \vec{u}_Ω is the velocity of a moving domain (mesh velocity after discretization), E is the total energy per

unit mass, p is the static pressure, c_p is the specific heat at constant pressure, T is the temperature, the viscous stress tensor can be written in vector notation as

$$\bar{\tau} = \nabla \vec{v} + \nabla \vec{v}^T - \frac{2}{3} \bar{I} (\nabla \cdot \vec{v}), \quad (2.4)$$

and

$$\bar{\sigma} = \mu_{tot}^1 \bar{\tau}. \quad (2.5)$$

Assuming a perfect gas with a ratio of specific heats, γ , and gas constant, R , the pressure is determined from

$$p = (\gamma - 1) \rho \left[E - \frac{1}{2} (\vec{v} \cdot \vec{v}) \right], \quad (2.6)$$

the temperature is given by

$$T = \frac{p}{\rho R}, \quad (2.7)$$

and

$$c_p = \frac{\gamma R}{(\gamma - 1)}. \quad (2.8)$$

The second line in the equation system (2.1) represents the no-slip condition at a solid wall, the third line represents an adiabatic condition at the wall, and the final line represents a characteristic-based boundary condition at the far-field where the fluid state at the boundary is updated using the state at infinity (free-stream conditions) depending on the sign of the eigenvalues [48]. Here, W represents the characteristic variables.

For turbulent flows, we are also concerned with obtaining solutions of the unsteady Reynolds-averaged Navier-Stokes equations (URANS), which will require the inclusion of a suitable turbulence model. In accord with the standard approach to turbulence modeling based upon the Boussinesq hypothesis [132], which states that the effect of turbulence can be represented as an increased viscosity, the total the viscosity is divided into laminar and turbulent components, or μ_{dyn} and μ_{tur} , respectively. In order to close the system of equations, the dynamic viscosity μ_{dyn} is assumed to satisfy Sutherland's law [131] as a function of temperature alone, or $\mu_{dyn} = \mu_{dyn}(T)$, and the turbulent viscosity μ_{tur} is computed via a selected turbulence model.

Turbulence and the mean flow become coupled by replacing the dynamic viscosity in the momentum and energy equations in the Navier-Stokes equations with

$$\mu_{tot}^1 = \mu_{dyn} + \mu_{tur}, \quad \mu_{tot}^2 = \frac{\mu_{dyn}}{Pr_d} + \frac{\mu_{tur}}{Pr_t}, \quad (2.9)$$

where Pr_d and Pr_t are the dynamic and turbulent Prandtl numbers, respectively. Here, μ_{tot}^2 represents the effective thermal conductivity. The viscous fluxes, viscosity, and effective thermal conductivity are decomposed and expressed in a nonstandard notation in order to obtain reduced expressions in the calculus during the forthcoming continuous adjoint treatment.

The ALE form of the equations can be identified by the inclusion of domain movement in the form of adjustments to the convective flux terms in \bar{F}_{ale}^c and boundary conditions that involve the velocity of the domain or surface \vec{u}_Ω . For example, this form of the equations enables the analysis of problems with pitching, plunging, or rotating surfaces, to name a few options. The surface motion can be prescribed or calculated in time by using the integrated loads on the surface coupled with an additional set of equations governing the dynamics of the body. Numerical strategies for simulations with dynamic meshes are discussed in Chapter 4. Note that, for problems on fixed grids (i.e., $\vec{u}_\Omega = 0$), the system in (2.1) reduces to a purely Eulerian formulation.

Apart from those given in (2.1), additional boundary conditions are possible and later discussed. In particular, we will also consider an isothermal no-slip condition on S , characteristic-based inlet boundaries (stagnation, mass flow, or supersonic conditions prescribed), and characteristic-based outlet boundaries (back pressure prescribed). Note that all boundary conditions takes into account any motion of the boundaries.

For unsteady problems, the temporal conditions will be problem dependent, and in this dissertation, we are interested in time-periodic flows where the initial and terminal conditions do not affect the time-averaged behavior over the time interval of interest, such as prescribed pitching, plunging, or rotational motion of the domain at constant frequencies. Therefore, we use the free-stream fluid state as the initial condition for the mean flow in conjunction with integration over multiple periods of oscillation (to remove transient effects by reaching a periodic steady state), and this is a typical practice in external aerodynamics.

2.2 Turbulence Modeling

The turbulent viscosity, μ_{tur} , is obtained from a turbulence model dependent on the flow state and a set of new state variables for turbulence, $\hat{\nu}$, i.e., $\mu_{tur} = \mu_{tur}(U, \hat{\nu})$. Here, we assume that $\hat{\nu}$ is a single scalar variable obtained from a one-equation turbulence model. The Spalart-Allmaras (S-A) model is one of the most common and widely used turbulence models for the analysis and design of engineering applications affected by turbulent flows, especially for applications in external aerodynamics. The S-A model is used for all of the turbulent calculations in this dissertation, and a brief description of it is given below.

2.2.1 Spalart-Allmaras Model

In the case of the one-equation Spalart-Allmaras [117] turbulence model, the turbulent viscosity is computed as

$$\mu_{tur} = \rho \hat{\nu} f_{v1}, \quad f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3}, \quad \chi = \frac{\hat{\nu}}{\nu}, \quad \nu = \frac{\mu_{dyn}}{\rho}. \quad (2.10)$$

The new variable $\hat{\nu}$ is obtained by solving the following transport equation in conjunction with the mean flow equations:

$$\begin{cases} \mathcal{R}_{\hat{\nu}}(U, \hat{\nu}) = \frac{\partial \hat{\nu}}{\partial t} + \nabla \cdot \vec{F}^c - \nabla \cdot \vec{F}^v - \mathcal{Q} = 0 & \text{in } \Omega \quad t > 0 \\ \hat{\nu} = 0 & \text{on } S \\ \hat{\nu} = \sigma_{\infty} \hat{\nu} & \text{on } \Gamma_{\infty} \end{cases} \quad (2.11)$$

where the convective, viscous, and source terms are given by

$$\vec{F}^c = \vec{v} \hat{\nu}, \quad \vec{F}^v = -\frac{\nu + \hat{\nu}}{\sigma} \nabla \hat{\nu}, \quad \mathcal{Q} = c_{b1} \hat{S} \hat{\nu} - c_{w1} f_w \left(\frac{\hat{\nu}}{d_S} \right)^2 + \frac{c_{b2}}{\sigma} |\nabla \hat{\nu}|^2, \quad (2.12)$$

the production term \hat{S} is defined as $\hat{S} = |\vec{\omega}| + \frac{\hat{\nu}}{\kappa^2 d_S^2} f_{v2}$, $\vec{\omega} = \nabla \times \vec{v}$ is the fluid vorticity, d_S is the distance to the nearest wall, and $f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}}$. The function f_w is computed as $f_w = g \left[\frac{1 + c_{w3}}{g^6 + c_{w3}^6} \right]^{1/6}$, where $g = r + c_{w2}(r^6 - r)$ and $r = \frac{\hat{\nu}}{\hat{S} \kappa^2 d_S^2}$. Finally, the set of closure constants for the model is given by

$$\sigma = 2/3, \quad c_{b1} = 0.1355, \quad c_{b2} = 0.622, \quad \kappa = 0.41, \quad c_{w1} = \frac{c_{b1}}{\kappa^2} + \frac{1 + c_{b2}}{\sigma}, \quad c_{w2} = 0.3, \quad c_{w3} = 2, \quad c_{v1} = 7.1. \quad (2.13)$$

The physical meaning of the far-field boundary condition for the turbulent viscosity is the imposition of some fraction of the laminar viscosity at the far-field. On viscous walls, $\hat{\nu}$ is set to zero, corresponding to the absence of turbulent eddies very near to the wall.

2.3 Compressible, Unsteady Euler Equations

It is often useful to ignore viscous effects when investigating and developing numerical methods for the integration of convective fluxes or implementation of boundary conditions, especially when dealing with the ALE form of the equations. Similarly, treating the inviscid form of the equations also aids in the development of suitable numerical methods and boundary conditions for the continuous adjoint, as we will see in later chapters.

The Navier-Stokes equations simplify to the Euler equations with the omission of the viscous fluxes and an adjustment to the solid wall boundary condition, and these equations can be written in an ALE differential form as

$$\begin{cases} \mathcal{R}(U) = \frac{\partial U}{\partial t} + \nabla \cdot \vec{F}_{ale}^c - \mathcal{Q} = 0 & \text{in } \Omega \quad t > 0 \\ (\vec{v} - \vec{u}_{\Omega}) \cdot \vec{n} = 0 & \text{on } S \\ (W)_+ = W_{\infty} & \text{on } \Gamma_{\infty} \end{cases} \quad (2.14)$$

where

$$U = \begin{Bmatrix} \rho \\ \rho \vec{v} \\ \rho E \end{Bmatrix}, \quad \vec{F}_{ale}^c = \begin{Bmatrix} \rho(\vec{v} - \vec{u}_\Omega) \\ \rho \vec{v} \otimes (\vec{v} - \vec{u}_\Omega) + \bar{I} p \\ \rho E(\vec{v} - \vec{u}_\Omega) + p \vec{v} \end{Bmatrix}, \quad \mathcal{Q} = \begin{Bmatrix} q_\rho \\ \vec{q}_{\rho \vec{v}} \\ q_{\rho E} \end{Bmatrix}, \quad (2.15)$$

which require fewer boundary conditions than the Navier-Stokes equations. In particular, the condition on temperature is unnecessary, and the condition on the velocity at a solid wall has become a flow tangency condition. Without viscosity, there is no mechanism to enforce a no-slip condition; therefore, a no penetration condition is imposed.

2.4 Governing Equations in a Rotating Reference Frame

For the simulation of flow past certain aerodynamic bodies that operate under an imposed steady rotation, including many turbomachinery, propeller, and rotor applications, it can be advantageous to transform the system of governing equations into a non-inertial reference frame that rotates with the body of interest [48, 49].

With this transformation, a flow field that is unsteady when viewed from the inertial frame can be solved for in a steady manner, and thus much more efficiently, without the need for time-accurate integration or grid motion. This can be viewed as a simplification of the general unsteady formulation above with the following applied to the systems in (2.1) or (2.14):

$$\frac{\partial U}{\partial t} = 0, \quad \vec{u}_\Omega = \vec{\omega} \times \vec{r}, \quad \mathcal{Q} = \begin{Bmatrix} \cdot \\ -\rho(\vec{\omega} \times \vec{v}) \\ \cdot \end{Bmatrix}, \quad (2.16)$$

where $\vec{\omega} = \{\omega_x, \omega_y, \omega_z\}^T$ is the steady angular velocity of the reference frame and \vec{r} is the position vector pointing from a specified rotation center (x_o, y_o, z_o) to a point (x, y, z) in the flow domain, or $\vec{r} = \{(x - x_o), (y - y_o), (z - z_o)\}^T$. In this case, \vec{u}_Ω is the velocity due to rotation, sometimes called the whirl velocity. After including the terms in (2.16), the result is the absolute velocity formulation of the non-inertial governing equations. A form based on the relative velocity \vec{v}_r , where $\vec{v}_r = \vec{v} - \vec{u}_\Omega$, is also possible but not considered here.

It is important to note that not all rotating applications can benefit from this solution approach. The flow field must be steady in the rotating frame, and some conditions or geometric features, such as relative surface motion, can cause unsteadiness for rotating bodies. For instance, if the incoming flow velocity is not parallel to the axis of rotation, the conditions are no longer axisymmetric, and the blades would not see a steady field during rotation. However, if desired, the non-inertial governing equations can also be solved in an unsteady fashion. Typical applications for the non-inertial equations might include hovering helicopter rotors, propellers in forward flight, or horizontal axis wind turbines in isolation.

Due to their inclusion of domain velocity terms and the reduced computational cost for steadily

rotating problems, the non-inertial governing equations offer an excellent intermediate stage in the development of flow and adjoint solvers for unsteady flows on dynamic meshes. Grid velocities, source terms, and the continuous adjoint formulation can all be tested with a steady problem in a rotating frame in order to avoid the added cost of time-accurate integration, which can help speed up the implementation and verification of the solvers. This was an approach taken for the development, implementation, and application of the work within this dissertation.

Chapter 3

The Unsteady Continuous Adjoint Approach

This chapter presents the main contributions of the dissertation: the derivation of the unsteady continuous adjoint approach using differential geometry for problems with dynamic surfaces. Detailed derivations of the continuous adjoint equations, their admissible boundary conditions, and the expressions for surface sensitivity in a formulaic structure for multiple objective functions and governing equation systems are demonstrated below.

3.1 Optimal Shape Design Problem Statement

A typical shape optimization problem seeks the minimization of a cost function $J(S)$ as chosen by the designer, with respect to changes in the shape of the boundary S . For the present methodology, the choice of cost function is not an arbitrary one, and important consequences of this will be discussed later. Initially, we will concentrate on a generic functional defined as a time-averaged, integrated quantity on the solid surface that depends on a scalar j evaluated at each point on S as a function of the force on the surface, surface temperature, or heat flux through the surface.

We note that any changes to the shape of S will result in perturbations in the fluid state U in the domain, and that these variations in the state are constrained to satisfy the system of governing equations, i.e., $\mathcal{R}(U) = 0$ must be satisfied for any candidate shape of S . Therefore, the optimal shape design problem can be formulated as a PDE-constrained optimization problem:

$$\begin{aligned} \min_S J(S) &= \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S j(\vec{f}, T, \partial_n T, \vec{n}) ds dt \\ &\text{subject to: } \mathcal{R}(U) = 0, \end{aligned} \tag{3.1}$$

where $\mathbb{T} = t_f - t_o$ is the time interval of interest, $\vec{f} = (f_1, f_2, f_3)$ is the time-dependent force on the surface, T is the temperature, and \vec{n} is the outward-pointing unit normal vector to the surface S .

As we are in infinite-dimensional, functional space, a suitable space must be chosen for the admissible shape parameterization. In this work, S is assumed to be continuously differentiable (C^1), and the local shape perturbations applied to S can be described by

$$S' = \{\vec{x} + \delta S(\vec{x}) \vec{n}(\vec{x}) : \vec{x} \in S\}, \quad (3.2)$$

where S has been deformed to a new surface shape S' by applying an infinitesimal profile deformation δS in the local normal direction \vec{n} at a point on the surface \vec{x} , as shown in Fig. 3.1.

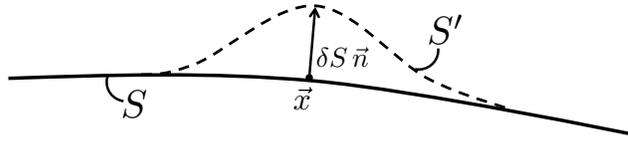


Figure 3.1: An infinitesimal shape deformation in the local surface normal direction.

The minimization (3.1) can be considered a problem in optimal control theory where the behavior of the governing flow equation system is controlled by the shape of the boundary S . As we are interested in gradient-based optimization, the goal is to compute the first variation of $J(S)$ caused by multiple, small perturbations of the surface and to use this sensitivity information to drive our geometric changes in order to find an optimal shape for S .

3.2 Variation of the Functional

The first step is to evaluate the gradient of the functional in infinite-dimensional space with respect to the infinitesimal boundary perturbations, also called the first variation when viewed from the perspective of multidimensional calculus of variations, which gives

$$\delta J = \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_{\delta S} j(\vec{f}, T, \partial_n T, \vec{n}) ds dt + \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \delta j(\vec{f}, T, \partial_n T, \vec{n}) ds dt, \quad (3.3)$$

where we have introduced $\int_{\delta S}(\cdot) ds = \int_{S'}(\cdot) ds - \int_S(\cdot) ds$ as a shorthand. Note that taking the variation results in two separate terms: the first term depends on the variation of the geometry and the value of the scalar function in the original state, while the second term depends on the original geometry and the variation of the scalar function caused by the deformation.

Eqn. 3.3 can be further simplified by using formulas from differential geometry and expressing the first variation more explicitly in terms of the independent variables of the functional. It can be shown [114, 104] that $\int_{\delta S} j ds = \int_S (\partial_n j - 2H_m j) \delta S ds$, where H_m is the mean curvature of S computed as $(\kappa_1 + \kappa_2)/2$, where (κ_1, κ_2) are curvatures in two orthogonal directions on the surface.

Using this relationship, we find that the first term on the right-hand side of Eqn. 3.3 becomes

$$\begin{aligned} \int_{\delta S} j(\vec{f}, T, \partial_n T, \vec{n}) ds &= \int_S (\partial_n j - 2H_m j) \delta S ds \\ &= \int_S \left(\frac{\partial j}{\partial \vec{f}} \cdot \partial_n \vec{f} + \frac{\partial j}{\partial T} \partial_n T + \frac{\partial j}{\partial(\partial_n T)} \partial_n^2 T - 2H_m j \right) \delta S ds, \end{aligned} \quad (3.4)$$

where from the chain rule and our functional definition,

$$\partial_n j = \vec{n} \cdot \nabla j(\vec{f}, T, \partial_n T, \vec{n}) = \frac{\partial j}{\partial \vec{f}} \cdot \partial_n \vec{f} + \frac{\partial j}{\partial T} \partial_n T + \frac{\partial j}{\partial(\partial_n T)} \partial_n^2 T. \quad (3.5)$$

The second term on the right-hand side of Eqn. 3.3 can also be further manipulated by focusing on δj :

$$\delta j(\vec{f}, T, \partial_n T, \vec{n}) = \frac{\partial j}{\partial \vec{f}} \cdot \delta \vec{f} + \frac{\partial j}{\partial T} \delta T + \frac{\partial j}{\partial(\partial_n T)} \delta(\partial_n T) - \frac{\partial j}{\partial \vec{n}} \cdot \nabla_S(\delta S), \quad (3.6)$$

where we have used $\delta \vec{n} = -\nabla_S(\delta S)$, which holds for small deformations [115]. Here, ∇_S represents the tangential gradient operator on S . Combining results from Eqns. 3.4 and 3.6 and introducing them into Eqn. 3.3 gives an intermediate expression for the variation of the functional:

$$\begin{aligned} \delta J &= \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \left(\frac{\partial j}{\partial \vec{f}} \cdot \partial_n \vec{f} + \frac{\partial j}{\partial T} \partial_n T + \frac{\partial j}{\partial(\partial_n T)} \partial_n^2 T - 2H_m j \right) \delta S ds dt \\ &\quad + \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \left(\frac{\partial j}{\partial \vec{f}} \cdot \delta \vec{f} + \frac{\partial j}{\partial T} \delta T + \frac{\partial j}{\partial(\partial_n T)} \delta(\partial_n T) - \frac{\partial j}{\partial \vec{n}} \cdot \nabla_S(\delta S) \right) ds dt. \end{aligned} \quad (3.7)$$

3.2.1 Force-based Functionals

At this point, we might choose a force-based objective function that depends only on \vec{f} in the following way:

$$j(\vec{f}) = \vec{d} \cdot \vec{f}, \quad (3.8)$$

such that

$$\frac{\partial j}{\partial \vec{f}} = \vec{d}, \quad \frac{\partial j}{\partial T} = 0, \quad \frac{\partial j}{\partial(\partial_n T)} = 0, \quad \frac{\partial j}{\partial \vec{n}} = \vec{0}, \quad (3.9)$$

where $\vec{d} = \vec{d}(\vec{x}, t)$ is the force projection vector, which can be chosen to relate the force on the surface \vec{f} to a desired quantity of interest. For unsteady problems, the force projection vector can be a function of both space and time. The local normal vector \vec{n} could also be chosen for \vec{d} , but additional terms involving $\frac{\partial j}{\partial \vec{n}}$ would arise. For many typical aerodynamic applications, the force

projection vector is constant, and some likely candidates are

$$\vec{d} = \begin{cases} \left(\frac{1}{C_\infty} \right) (\cos \alpha \cos \beta, \sin \alpha \cos \beta, \sin \beta) & C_D \quad \text{Drag} \\ \left(\frac{1}{C_\infty} \right) (-\sin \alpha, \cos \alpha, 0) & C_L \quad \text{Lift} \\ \left(\frac{1}{C_\infty} \right) (-\sin \beta \cos \alpha, -\sin \beta \sin \alpha, \cos \beta) & C_{SF} \quad \text{Side-force} \\ \left(\frac{1}{C_\infty C_D} \right) \left(-\sin \alpha - \frac{C_L}{C_D} \cos \alpha \cos \beta, -\frac{C_L}{C_D} \sin \beta, \cos \alpha - \frac{C_L}{C_D} \sin \alpha \cos \beta \right) & \frac{C_L}{C_D} \quad \text{L/D} \\ \left(\frac{1}{C_\infty} \right) (0, 0, 1) & C_{f_z} \quad \text{Z-force} \\ \left(\frac{1}{C_\infty L_{ref}} \right) (-(y - y_o), (x - x_o), 0) & C_{m_z} \quad \text{Z-moment,} \end{cases} \quad (3.10)$$

where $C_\infty = \frac{1}{2} \rho_\infty v_\infty^2 A_z$, ρ_∞ is the free-stream density, v_∞ is the freestream velocity, A_z is the reference area, L_{ref} is a reference length for computing moments, α is the free-stream angle of attack, and β is the side-slip angle. In practice for a three-dimensional surface, the sum of all positive components in the z -direction of the surface normal vectors is used for the projection A_z . A pre-specified reference area can also be used in a similar fashion, and this is an established procedure in applied aerodynamics.

Care must be taken so that any constants used in the non-dimensionalization, such as the chord lengths or projected areas that might appear in (3.10), are not exploited by the optimizer to reduce the objective function. Changes to these values due to shape changes should be explicitly included in the objective function during optimization, or rather, the design variable parameterization and/or geometric constraints should strictly forbid changes to these quantities.

After choosing a force-based functional and imposing the relationships in Eqn. 3.9 above, we can simplify the variation of the functional in Eqn. 3.7 to

$$\delta J = \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \left(\vec{d} \cdot \partial_n \vec{f} - 2H_m(\vec{d} \cdot \vec{f}) \right) \delta S \, ds \, dt + \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \vec{d} \cdot \delta \vec{f} \, ds \, dt, \quad (3.11)$$

where f will take a different form depending on whether viscous or inviscid flow is governing.

Viscous Flow

For viscous flows, the force on the surface is composed of a pressure component along with a component due to viscous stresses. It can be expressed as $\vec{f} = (\bar{I}p - \bar{\sigma}) \cdot \vec{n}$, and therefore,

$$\delta \vec{f} = \delta \left[(\bar{I}p - \bar{\sigma}) \cdot \vec{n} \right] = (\bar{I}\delta p - \delta \bar{\sigma}) \cdot \vec{n} - (\bar{I}p - \bar{\sigma}) \cdot \nabla_S(\delta S), \quad (3.12)$$

where we have again used $\delta \vec{n} = -\nabla_S(\delta S)$. By introducing Eqn. 3.12, we can rearrange Eqn. 3.11 as

$$\begin{aligned} \delta J = & \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \vec{d} \cdot (\bar{I}\delta p - \delta \bar{\sigma}) \cdot \vec{n} \, ds \, dt + \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \left[\vec{d} \cdot \partial_n \vec{f} - 2H_m(\vec{f} \cdot \vec{d}) \right] \delta S \, ds \, dt \\ & - \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \vec{d} \cdot (\bar{I}p - \bar{\sigma}) \cdot \nabla_S(\delta S) \, ds \, dt. \end{aligned} \quad (3.13)$$

The final term of Eqn. 3.13 can be readily integrated by parts to give

$$\begin{aligned}
\delta J &= \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \vec{d} \cdot (\bar{I}\delta p - \delta\bar{\sigma}) \cdot \vec{n} \, ds \, dt + \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \left[\vec{d} \cdot \partial_n \vec{f} - 2H_m(\vec{f} \cdot \vec{d}) \right] \delta S \, ds \, dt \\
&\quad - \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \nabla_S \cdot \left[\vec{d} \cdot (\bar{I}p - \bar{\sigma}) \delta S \right] \, ds \, dt + \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \nabla_S \cdot \left[\vec{d} \cdot (\bar{I}p - \bar{\sigma}) \right] \delta S \, ds \, dt \\
&= \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \vec{d} \cdot (\bar{I}\delta p - \delta\bar{\sigma}) \cdot \vec{n} \, ds \, dt \\
&\quad + \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \left\{ \nabla_S \cdot \left[\vec{d} \cdot (\bar{I}p - \bar{\sigma}) \right] + \vec{d} \cdot \partial_n \vec{f} - 2H_m(\vec{f} \cdot \vec{d}) \right\} \delta S \, ds \, dt, \tag{3.14}
\end{aligned}$$

where we have rearranged and used the identity $\int_S \nabla_S \cdot (\cdot) \, ds = 0$ on a closed surface [114] in going from the first to second lines. Further simplifications can be made in the braced portion of the integrand in the final term of Eqn. 3.14:

$$\begin{aligned}
&\nabla_S \cdot \left[\vec{d} \cdot (\bar{I}p - \bar{\sigma}) \right] + \vec{d} \cdot \partial_n \vec{f} - 2H_m(\vec{f} \cdot \vec{d}) \\
&= \nabla_S \cdot \left[\vec{d} \cdot (\bar{I}p - \bar{\sigma}) \right] + \partial_n (\vec{d} \cdot \vec{f}) - 2H_m(\vec{d} \cdot \vec{f}) - \vec{f} \cdot \partial_n \vec{d} \\
&= \nabla_S \cdot \left[\vec{d} \cdot (\bar{I}p - \bar{\sigma}) \right] + \partial_n \left[\vec{d} \cdot (\bar{I}p - \bar{\sigma}) \cdot \vec{n} \right] - 2H_m \left[\vec{d} \cdot (\bar{I}p - \bar{\sigma}) \cdot \vec{n} \right] - (\bar{I}p - \bar{\sigma}) \cdot \vec{n} \cdot \partial_n \vec{d} \\
&= \nabla \cdot \left[\vec{d} \cdot (\bar{I}p - \bar{\sigma}) \right] - (\bar{I}p - \bar{\sigma}) \cdot \vec{n} \cdot \partial_n \vec{d}, \tag{3.15}
\end{aligned}$$

where we have used the product rule in going from the first to second lines and the divergence operator written in local coordinates, or $\nabla \cdot \vec{q} = \nabla_S \cdot \vec{q} + \partial_n (\vec{q} \cdot \vec{n}) - 2H_m(\vec{q} \cdot \vec{n})$ with \vec{q} being an arbitrary vector that, in this case, is described by $\vec{d} \cdot (\bar{I}p - \bar{\sigma})$. Substituting the result of Eqn. 3.15 back into Eqn. 3.14 gives a near final expression for the variation of the functional,

$$\delta J = \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \vec{d} \cdot (\bar{I}\delta p - \delta\bar{\sigma}) \cdot \vec{n} \, ds \, dt + \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \left\{ \nabla \cdot \left[\vec{d} \cdot (\bar{I}p - \bar{\sigma}) \right] - (\bar{I}p - \bar{\sigma}) \cdot \vec{n} \cdot \partial_n \vec{d} \right\} \delta S \, ds \, dt. \tag{3.16}$$

Lastly, the final term in Eqn. 3.16 can be simplified for easier calculation by using information from the Navier-Stokes equations. In this manner, we can remove higher order derivative terms that can lead to implementation issues. Expanding the divergence term gives

$$\begin{aligned}
\nabla \cdot \left[\vec{d} \cdot (\bar{I}p - \bar{\sigma}) \right] &= \nabla \vec{d} : (\bar{I}p - \bar{\sigma}) + \vec{d} \cdot \nabla \cdot (\bar{I}p - \bar{\sigma}) \\
&= \nabla \vec{d} : (\bar{I}p - \bar{\sigma}) + \vec{d} \cdot (\nabla p - \nabla \cdot \bar{\sigma}) \\
&= \nabla \vec{d} : (\bar{I}p - \bar{\sigma}) + \vec{d} \cdot [\vec{q}_{\rho\bar{\sigma}} - \partial_t(\rho\vec{v})], \tag{3.17}
\end{aligned}$$

where in going from the third to fourth line of Eqn. 3.17, we have used the momentum equation written on the surface (including unsteadiness and source term effects). The variation of the functional

can then be written concisely as

$$\begin{aligned} \delta J = & \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \vec{d} \cdot (\bar{I} \delta p - \delta \bar{\sigma}) \cdot \vec{n} \, ds \, dt \\ & + \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \left\{ \vec{d} \cdot [\bar{q}_{\rho \vec{v}} - \partial_t(\rho \vec{v})] + \nabla \vec{d} : (\bar{I} p - \bar{\sigma}) - (\bar{I} p - \bar{\sigma}) \cdot \vec{n} \cdot \partial_n \vec{d} \right\} \delta S \, ds \, dt. \end{aligned} \quad (3.18)$$

Note that, for a steady problem without source terms and with a constant force projection vector, the second integral in Eqn. 3.18 vanishes.

Inviscid Flow

For inviscid flows, the expression for the force on the surface reduces to $\vec{f} = \bar{I} p \cdot \vec{n}$, as the lack of viscosity has removed any viscous forces on the surface, which is consistent with the flow tangency boundary condition on the solid wall, or $\vec{v} \cdot \vec{n} = 0$. Noting this change in the definition of \vec{f} , we recover the following for its variation:

$$\delta \vec{f} = \delta (\bar{I} p \cdot \vec{n}) = \bar{I} \delta p \cdot \vec{n} - \bar{I} p \cdot \nabla_S (\delta S). \quad (3.19)$$

Starting from Eqn. 3.11 and using Eqn. 3.19, we can perform a nearly identical procedure as in the viscous case (see Eqns. 3.13–3.16) to arrive at the following reduced form:

$$\delta J = \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \vec{d} \cdot \bar{I} \delta p \cdot \vec{n} \, ds \, dt + \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \left[\nabla \cdot (\vec{d} \cdot \bar{I} p) - \bar{I} p \cdot \vec{n} \cdot \partial_n \vec{d} \right] \delta S \, ds \, dt. \quad (3.20)$$

Lastly, the final integral in Eqn. 3.20 can again be simplified by expanding the divergence term,

$$\nabla \cdot (\vec{d} \cdot \bar{I} p) = \nabla \vec{d} : \bar{I} p + \vec{d} \cdot \nabla p = p \nabla \cdot \vec{d} + \vec{d} \cdot \nabla p, \quad (3.21)$$

and the variation of the functional can then be written concisely as

$$\delta J = \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \vec{d} \cdot \bar{I} \delta p \cdot \vec{n} \, ds \, dt + \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \left(\vec{d} \cdot \nabla p + p \nabla \cdot \vec{d} - \bar{I} p \cdot \vec{n} \cdot \partial_n \vec{d} \right) \delta S \, ds \, dt. \quad (3.22)$$

3.2.2 Temperature-based Functionals

By nature, temperature functionals only make sense in the context of viscous flows where viscous effects cause heat transfer at the surface. Indeed, only for the Navier-Stokes equations is a boundary condition for the temperature on the surface even required. Typically, one of two boundary conditions is chosen for the temperature on the surface: an imposition of the heat flux (related to the normal gradient of the temperature, $\partial_n T$), or an imposition of the surface temperature directly (an isothermal condition). In the next section, we will present the results for a heat flux functional.

Heat Flux Design

In order to perform shape design for controlling the heat flux on the surface (minimizing the total integrated heat flux, for instance), an objective function that depends on $\partial_n T$ can be chosen as

$$j(\partial_n T) = c \partial_n T = c(\vec{n} \cdot \nabla T), \quad (3.23)$$

such that the other terms in the intermediate expression for the functional variation in Eqn. 3.7 become

$$\frac{\partial j}{\partial \vec{f}} = 0, \quad \frac{\partial j}{\partial T} = 0, \quad \frac{\partial j}{\partial(\partial_n T)} = c, \quad \frac{\partial j}{\partial \vec{n}} = c \nabla T, \quad (3.24)$$

where c is an arbitrary constant.

From the relationships in Eqn. 3.24, the variation of the functional in Eqn. 3.7 is simplified to

$$\delta J = \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S (c \partial_n^2 T - 2H_m(c \partial_n T)) \delta S ds dt + \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S (c \delta(\partial_n T) - c \nabla T \cdot \nabla_S(\delta S)) ds dt, \quad (3.25)$$

and the final term in Eqn. 3.25 can be integrated by parts to give

$$\begin{aligned} \delta J &= \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S (c \partial_n^2 T - 2H_m(c \partial_n T)) \delta S ds dt \\ &\quad + \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S (c \delta(\partial_n T) - \nabla_S \cdot (c \nabla T \delta S) + \nabla_S \cdot (c \nabla T) \delta S) ds dt \\ &= \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S c \delta(\partial_n T) ds dt + \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S (\nabla_S \cdot (c \nabla T) + c \partial_n^2 T - 2H_m(c \partial_n T)) \delta S ds dt, \end{aligned} \quad (3.26)$$

where we have rearranged and used the identity $\int_S \nabla_S \cdot (\cdot) ds = 0$ on a closed surface [114] in going from the first to second lines. Recognizing that the divergence operator in local coordinates can be applied, or $\nabla \cdot \vec{a} = \nabla_S \cdot \vec{a} + \partial_n(\vec{a} \cdot \vec{n}) - 2H_m(\vec{a} \cdot \vec{n})$ with $\vec{a} = c \nabla T$ in this case, yields a simplified form for the variation of the functional:

$$\delta J = \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S c \delta(\partial_n T) ds dt + \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \nabla \cdot (c \nabla T) \delta S ds dt. \quad (3.27)$$

3.3 The Adjoint Approach to Optimal Design

As we are interested in gradient-based methods, the shape optimization process will search for a stationary point of the functional variation, or $\delta J = 0$, which represents a first-order optimality condition. In typical practice, finding a minimum involves both computing the value and gradient of the functional and supplying this information to a suitable optimization method that will search the design space with appropriate step sizes until satisfying the optimality condition. The functional gradient can be obtained through a number of sensitivity analysis techniques.

In the case of a force-based functional in viscous flow, for instance, the integrated force on the surface in Eqn. 3.8 and the functional variation in Eqn. 3.18 are required. While many of the

quantities required to evaluate J and δJ are readily available from a single solution of the governing equations in the domain (e.g., the pressure and viscous stress tensor on the surface), perturbations of certain flow quantities are also present, and obtaining the variation of these variables for multiple, arbitrary surface deformations δS is not as straightforward. In fact, as expressed in Eq. 3.18, calculating δJ for a number N of shape deformations would require N solutions of the linearized governing equations in order to compute the values of δp and $\delta \bar{\sigma}$ that correspond to each deformation. This is a direct approach, and it is analogous to the finite difference approach to computing gradients discussed in Chapter 1, especially in terms of the computational cost.

Ideally, the explicit dependence on δp and $\delta \bar{\sigma}$ in the variation of the functional would be removed, so that the variation due to an arbitrary number of deformations can be computed in a much more efficient manner. This suggests the use of an analytic method for sensitivity analysis that can eliminate these perturbations and leads to the adjoint method.

Following the adjoint approach to optimal design, Eqn. 3.1 can be transformed into an unconstrained optimization problem by including the inner product of an adjoint state vector, or $\Psi = \Psi(\vec{x}, t)$, and the governing equations integrated over the domain (space and time) to the original functional in order to form the Lagrangian:

$$\mathcal{J} = \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S j(\vec{f}, T, \partial_n T, \vec{n}) ds dt - \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_{\Omega} \Psi^T \mathcal{R}(U) d\Omega dt, \quad (3.28)$$

where we have introduced the adjoint variables, which operate as Lagrange multipliers and are defined as

$$\Psi = \begin{Bmatrix} \psi_\rho \\ \psi_{\rho v_1} \\ \psi_{\rho v_2} \\ \psi_{\rho v_3} \\ \psi_{\rho E} \end{Bmatrix} = \begin{Bmatrix} \psi_\rho \\ \vec{\varphi} \\ \psi_{\rho E} \end{Bmatrix}. \quad (3.29)$$

The adjoint state vector contains an co-state variable for each conservative variable in the original governing equations. Note that, because the flow equations must be satisfied in the domain, or $\mathcal{R}(U) = 0$, the original functional in Eqn. 3.1 and the Lagrangian in Eqn. 3.28 are equivalent. Moreover, because it is equal to zero, it is equivalent to add or subtract the second term in the Lagrangian, and we will choose to subtract for convenience in signs. To find the gradient information needed to minimize the objective function, we repeat the process of taking the first variation of Eqn. 3.28:

$$\delta \mathcal{J} = \delta J - \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_{\Omega} \Psi^T \delta \mathcal{R}(U) d\Omega dt, \quad (3.30)$$

where the variation of the original functional δJ remains unchanged from the expressions derived above and a new term involving the linearized governing equations $\delta \mathcal{R}(U)$ has appeared. The goal

now is to perform manipulations involving the additional analytic sensitivity information provided by the linearized equations (along with linearized forms of the boundary conditions) that will remove any dependence on variations of the flow variables. In this manner, the cost of evaluating both J and $\delta\mathcal{J}$ will become independent of the number of surface perturbations (design variables), thus offering an efficient method for sensitivity analysis in a large design space.

3.4 The Linearized Governing Equations

The second term on the right-hand side of Eqn. 3.30 can be expanded by including the version of the governing equations that has been linearized with respect to the small perturbations of the surface, or $\delta\mathcal{R}(U)$. The deformation of the surface will induce perturbations in the solution, or δU , as well as in the gradient of the solution, or $\delta(\nabla U)$. To complete the linearized system of equations, the boundary conditions corresponding to the original governing system must also be linearized.

3.4.1 Linearized Navier-Stokes Equations in ALE Form

First, consider a perturbation to the unsteady, compressible Navier-Stokes equations in ALE form:

$$\begin{aligned}
\delta\mathcal{R}(U, \nabla U) &= \delta \left(\frac{\partial U}{\partial t} + \nabla \cdot \vec{F}_{ale}^c - \nabla \cdot \mu_{tot}^k \vec{F}^{vk} - \mathcal{Q} \right) \\
&= \delta \left[\frac{\partial U}{\partial t} + \nabla \cdot \vec{F}^c - \nabla \cdot (U \vec{u}_\Omega) - \nabla \cdot \mu_{tot}^k \vec{F}^{vk} - \mathcal{Q} \right] \\
&= \frac{\partial}{\partial t}(\delta U) + \nabla \cdot \delta \vec{F}^c - \nabla \cdot \delta(U \vec{u}_\Omega) - \nabla \cdot \delta(\mu_{tot}^k \vec{F}^{vk}) - \delta \mathcal{Q} \\
&= \frac{\partial}{\partial t}(\delta U) + \nabla \cdot \delta \vec{F}^c - \nabla \cdot \delta(U \vec{u}_\Omega) - \nabla \cdot \left(\vec{F}^{vk} \delta \mu_{tot}^k \right) - \nabla \cdot \left(\mu_{tot}^k \delta \vec{F}^{vk} \right) - \delta \mathcal{Q} \\
&= \frac{\partial}{\partial t}(\delta U) + \nabla \cdot \left(\frac{\partial \vec{F}^c}{\partial U} \delta U \right) - \nabla \cdot \left[\frac{\partial (U \vec{u}_\Omega)}{\partial U} \delta U \right] - \nabla \cdot \vec{F}^{vk} \frac{\partial \mu_{tot}^k}{\partial U} \delta U \\
&\quad - \nabla \cdot \mu_{tot}^k \left[\frac{\partial \vec{F}^{vk}}{\partial U} \delta U + \frac{\partial \vec{F}^{vk}}{\partial (\nabla U)} \delta(\nabla U) \right] - \frac{\partial \mathcal{Q}}{\partial U} \delta U \\
&= \frac{\partial}{\partial t}(\delta U) + \nabla \cdot \left(\vec{A}^c - \bar{I}_5 \vec{u}_\Omega - \mu_{tot}^k \vec{A}^{vk} \right) \delta U - \nabla \cdot \vec{F}^{vk} \frac{\partial \mu_{tot}^k}{\partial U} \delta U \\
&\quad - \nabla \cdot \mu_{tot}^k \vec{D}^{vk} \delta(\nabla U) - \frac{\partial \mathcal{Q}}{\partial U} \delta U,
\end{aligned} \tag{3.31}$$

where we have introduced the following Jacobian matrices that can be found in the appendix:

$$\left. \begin{aligned} \vec{A}^c &= (A_x^c, A_y^c, A_z^c), & A_i^c &= \left. \frac{\partial \vec{F}_i^c}{\partial U} \right|_{U(x,y,z)} \\ \vec{A}^{vk} &= (A_x^{vk}, A_y^{vk}, A_z^{vk}), & A_i^{vk} &= \left. \frac{\partial \vec{F}_i^{vk}}{\partial U} \right|_{U(x,y,z)} \\ \bar{D}^{vk} &= \begin{pmatrix} D_{xx}^{vk} & D_{xy}^{vk} & D_{xz}^{vk} \\ D_{yx}^{vk} & D_{yy}^{vk} & D_{yz}^{vk} \\ D_{zx}^{vk} & D_{zy}^{vk} & D_{zz}^{vk} \end{pmatrix}, & D_{ij}^{vk} &= \left. \frac{\partial \vec{F}_i^{vk}}{\partial (\partial_j U)} \right|_{U(x,y,z)} \end{aligned} \right\} \quad i, j = 1 \dots 3, \quad k = 1, 2. \quad (3.32)$$

In the second line of Eqn. 3.31, the terms involving the domain velocity \vec{u}_Ω have been separated from the traditional inviscid convective fluxes, \vec{F}^c . In this work, we assume that the perturbations of the surface do not affect the value of the viscosity, or $\delta\mu_{tot}^k = 0$. This is known as the constant, or frozen, viscosity assumption, and it is commonly used with the adjoint approach in order to reduce the complexity inherent in including sensitivity information for the viscosity. This approximation removes the difficulty of treating a turbulence model in the formulation at the cost of some accuracy in the resulting sensitivities. The validity of this assumption is problem-dependent, but in a wide variety of situations, it leads to accurate sensitivity information.

Under the frozen viscosity assumption, the linearized governing equations become

$$\delta\mathcal{R}(U, \nabla U) = \frac{\partial}{\partial t}(\delta U) + \nabla \cdot \left(\vec{A}^c - \bar{I}_5 \vec{u}_\Omega - \mu_{tot}^k \vec{A}^{vk} \right) \delta U - \nabla \cdot \mu_{tot}^k \bar{D}^{vk} \delta(\nabla U) - \frac{\partial \mathcal{Q}}{\partial U} \delta U = 0 \text{ in } \Omega. \quad (3.33)$$

3.4.2 Linearized Boundary Conditions

The linearized boundary conditions of the governing flow equations are also be required in order to remove any dependence on flow variations. For viscous flow, the no-slip and temperature boundary conditions are treated, and for inviscid flow, the flow tangency boundary condition is considered. In all cases, a characteristic-based condition is imposed at far-field, inlet, and outlet boundaries. The details for linearizing each of these boundary conditions are given in the following sections.

Linearized No-Slip Wall Boundary Condition

For viscous flows, we start with the no-slip boundary condition for a surface in arbitrary motion:

$$(\vec{v} - \vec{u}_\Omega) = 0 \text{ on } S, \quad (3.34)$$

where \vec{v} is the absolute flow velocity and \vec{u}_Ω is the local velocity of the domain in motion. Consider linearization of (3.34) with respect to small perturbations in the surface δS , which gives

$$(\vec{v} - \vec{u}_\Omega)' = (\vec{v} - \vec{u}_\Omega) + \delta(\vec{v} - \vec{u}_\Omega) + \partial_n(\vec{v} - \vec{u}_\Omega)\delta S, \quad (3.35)$$

where the second term on the right hand side of Eqn. 3.35 represents the change in the flow solution induced by the deformation and the third term represents the change due solely to the geometry of the deformation. Keeping in mind that the linearized boundary condition must also equal zero, we can rearrange Eqn. 3.35 to give a useful result for the continuous adjoint derivation:

$$\delta\vec{v} = -\partial_n(\vec{v} - \vec{u}_\Omega)\delta S \text{ on } S, \quad (3.36)$$

where, in order to simplify, we have used the original boundary condition (3.34) and $\delta\vec{u}_\Omega = 0$.

Linearized Flow Tangency Boundary Condition

For inviscid flows, we start with the flow tangency boundary condition:

$$(\vec{v} - \vec{u}_\Omega) \cdot \vec{n} = 0 \text{ on } S, \quad (3.37)$$

where \vec{v} is again the absolute flow velocity, \vec{u}_Ω is the local velocity of the domain in motion, and \vec{n} is the local normal of the surface. Consider linearization with respect to small perturbations in the surface δS for both the velocity and the normal terms separately:

$$(\vec{v} - \vec{u}_\Omega)' = (\vec{v} - \vec{u}_\Omega) + \delta(\vec{v} - \vec{u}_\Omega) + \partial_n(\vec{v} - \vec{u}_\Omega)\delta S \quad (3.38)$$

and

$$(\vec{n})' = \vec{n} + \delta\vec{n}, \quad (3.39)$$

where the second term on the right hand side of Eqn. 3.38 represents the change in the flow solution caused by the deformation and the third term represents the change due solely to the geometry of the deformation. The normal in Eqn. 3.39 does not involve any flow variables, so the change is due to the deformation alone. The complete linearized flow tangency boundary condition can then be obtained by taking the dot product of the two linearized components:

$$\begin{aligned} (\vec{v} - \vec{u}_\Omega)' \cdot (\vec{n})' &= \{(\vec{v} - \vec{u}_\Omega) + \delta\vec{v} - \delta\vec{u}_\Omega + \partial_n(\vec{v} - \vec{u}_\Omega)\delta S\} \cdot (\vec{n} + \delta\vec{n}) \\ &= (\vec{v} - \vec{u}_\Omega) \cdot \delta\vec{n} + \delta\vec{v} \cdot \vec{n} + \partial_n(\vec{v} - \vec{u}_\Omega)\delta S \cdot \vec{n}, \end{aligned} \quad (3.40)$$

where, in order to simplify, we have used the original boundary condition (3.37), $\delta\vec{u}_\Omega = 0$, and the approximation that any products of variations are negligible. Since the linearized version of the boundary condition must also equal zero, Eqn. 3.40 can be rearranged as

$$\delta\vec{v} \cdot \vec{n} = -(\vec{v} - \vec{u}_\Omega) \cdot \delta\vec{n} - \partial_n(\vec{v} - \vec{u}_\Omega)\delta S \cdot \vec{n}. \quad (3.41)$$

Finally, we apply the relationship $\delta\vec{n} = -\nabla_S(\delta S)$, which holds for small deformations, and the result is a form more directly related to the imposed surface perturbations:

$$\delta\vec{v} \cdot \vec{n} = (\vec{v} - \vec{u}_\Omega) \cdot \nabla_S(\delta S) - \partial_n(\vec{v} - \vec{u}_\Omega)\delta S \cdot \vec{n} \text{ on } S. \quad (3.42)$$

Linearized Heat Flux Wall Boundary Condition

Given its similar form, the linearization of the heat flux boundary condition will mirror that of the flow tangency condition. Recall the original heat flux condition, or

$$\partial_n T = \vec{n} \cdot \nabla T = q_n \text{ on } S, \quad (3.43)$$

where T is the temperature, \vec{n} is the local normal of the surface, and q_n is the imposed heat flux in the normal direction ($q_n = 0$ corresponds to an adiabatic wall). Consider linearization with respect to small perturbations in the surface δS for both the temperature and the normal terms separately:

$$(\nabla T)' = \nabla T + \delta(\nabla T) + \partial_n(\nabla T)\delta S \quad (3.44)$$

and

$$(\vec{n})' = \vec{n} + \delta\vec{n}, \quad (3.45)$$

where the second term on the right-hand side of Eqn. 3.44 represents the change in the flow solution caused by the deformation and the third term represents the change due solely to the geometry of the deformation. The normal in Eqn. 3.45 does not involve any flow variables, so the change is due to the deformation alone. The complete linearized boundary condition can be obtained by taking the dot product of the two linearized components:

$$\begin{aligned} (\nabla T)' \cdot (\vec{n})' &= [\nabla T + \delta(\nabla T) + \partial_n(\nabla T)\delta S] \cdot (\vec{n} + \delta\vec{n}) \\ &= \nabla T \cdot \delta\vec{n} + \delta(\nabla T) \cdot \vec{n} + \partial_n^2(T)\delta S, \end{aligned} \quad (3.46)$$

where, in order to simplify, we have used the original boundary condition (3.43) and the approximation that any products of variations are negligible. As the linearized version of the boundary condition must also equal the imposed normal heat flux q_n , Eqn. 3.46 can be rearranged as

$$\delta(\nabla T) \cdot \vec{n} = -(\nabla T) \cdot \delta\vec{n} - \partial_n^2(T)\delta S. \quad (3.47)$$

Again, using the relationship $\delta\vec{n} = -\nabla_S(\delta S)$, which holds for small deformations, and the fact that in a continuum $\delta(\nabla T) = \nabla(\delta T)$, gives

$$\partial_n(\delta T) = \nabla T \cdot \nabla_S(\delta S) - \partial_n^2(T)\delta S \text{ on } S. \quad (3.48)$$

Linearized Isothermal Wall Boundary Condition

The typical isothermal wall boundary condition can be expressed as

$$T = T_o \text{ on } S, \quad (3.49)$$

where T is the temperature and T_o is the imposed wall temperature for the isothermal condition. Consider linearization with respect to small perturbations in the surface δS :

$$T' = T + \delta T + \partial_n(T)\delta S, \quad (3.50)$$

where the second term on the right-hand side of Eqn. 3.50 represents the change in the flow solution caused by the deformation and the third term represents the change due solely to the geometry of the deformation. As the linearized version of the boundary condition must also equal the imposed isothermal condition, Eqn. 3.50 can be rearranged to give a useful result for the continuous adjoint derivation:

$$\delta T = -\partial_n(T)\delta S \text{ on } S, \quad (3.51)$$

where, in order to simplify, we have used the original boundary condition (3.49).

Linearized Characteristic-based Boundary Condition

At far-field, inlet, or outlet boundaries, a typical characteristic-based boundary condition is applied in the direct problem, where the solution state is updated based on the sign of the eigenvalues from the convective flux Jacobian. As seen in the original governing equations (2.1), the boundary state is imposed for any incoming characteristics. Therefore, for the linearized problem, we impose

$$(\delta W)_+ = 0 \text{ on } \Gamma_\infty, \quad (3.52)$$

where $(\delta W)_+$ represents the incoming characteristics at the boundary. The treatment of characteristic-based boundary conditions for the adjoint problem is discussed in detail later in this chapter.

3.4.3 Complete System of Linearized Equations

Here, we collect the final expressions above in order to summarize the complete systems of linearized equations for the viscous and inviscid problems. Given a perturbation in the surface δS , these equations can be solved for the resulting perturbation in the flow solution δU .

Linearized Navier-Stokes Equations

By grouping the results from the linearization of the Navier-Stokes equations in ALE form and their boundary conditions from (3.33), (3.36), (3.48), and (3.52), one obtains

$$\begin{cases} \delta\mathcal{R} = \frac{\partial}{\partial t}(\delta U) + \nabla \cdot \left(\vec{A}^c - \bar{I}_5 \vec{u}_\Omega - \mu_{tot}^k \vec{A}^{vk} \right) \delta U - \nabla \cdot \mu_{tot}^k \bar{D}^{vk} \delta(\nabla U) - \frac{\partial \mathcal{Q}}{\partial U} \delta U = 0 & \text{in } \Omega \quad t > 0 \\ \delta \vec{v} = -\partial_n(\vec{v} - \vec{u}_\Omega) \delta S & \text{on } S \\ \partial_n(\delta T) = \nabla T \cdot \nabla_S(\delta S) - \partial_n^2(T) \delta S & \text{on } S \\ (\delta W)_+ = 0 & \text{on } \Gamma_\infty, \end{cases} \quad (3.53)$$

where we have assumed the use of an adiabatic wall condition.

Linearized Euler Equations

The linearized Euler equations can be easily recovered by removing the viscous terms from the linearized Navier-Stokes equations and applying the appropriate flow tangency boundary condition from (3.42):

$$\begin{cases} \delta\mathcal{R}(U) = \frac{\partial}{\partial t}(\delta U) + \nabla \cdot \left(\vec{A}^c - \bar{I}_5 \vec{u}_\Omega \right) \delta U - \frac{\partial \mathcal{Q}}{\partial U} \delta U = 0 & \text{in } \Omega \\ \delta \vec{v} \cdot \vec{n} = (\vec{v} - \vec{u}_\Omega) \cdot \nabla_S(\delta S) - \partial_n(\vec{v} - \vec{u}_\Omega) \delta S \cdot \vec{n} & \text{on } S \\ (\delta W)_+ = 0 & \text{on } \Gamma_\infty. \end{cases} \quad (3.54)$$

3.5 The Unsteady Continuous Adjoint Equations

This section contains a detailed derivation of the time-accurate, viscous, continuous adjoint equations along with their suitable boundary conditions. The process will involve introducing the results from the sections concerning the variation of the functional and the linearized governing equations.

3.5.1 Obtaining the Adjoint Equations

After linearizing the governing equations, we can now substitute Eqn. 3.33 into Eqn. 3.30 to produce

$$\begin{aligned} \delta\mathcal{J} = \delta J - \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_{\Omega} \Psi^\top \frac{\partial}{\partial t}(\delta U) d\Omega dt - \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_{\Omega} \Psi^\top \nabla \cdot \left(\vec{A}^c - \bar{I}_5 \vec{u}_\Omega - \mu_{tot}^k \vec{A}^{vk} \right) \delta U d\Omega dt \\ + \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_{\Omega} \Psi^\top \nabla \cdot \mu_{tot}^k \bar{D}^{vk} \delta(\nabla U) d\Omega dt + \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_{\Omega} \Psi^\top \frac{\partial \mathcal{Q}}{\partial U} \delta U d\Omega dt. \end{aligned} \quad (3.55)$$

If we can remove any dependence on variations of the flow variables, or δU , the variation of the objective function for multiple surface deformations can be found *without* the need for multiple solutions of the governing equations. As the flow perturbations are by definition arbitrary (non-zero), the general strategy for their removal is to isolate and factor these terms out of any expressions such that their multiplier can be set to zero, thereby forcing any terms involving δU to vanish. We now perform manipulations to remove this dependence, and integration by parts is the tool that enables

isolation of the flow perturbations by moving any derivatives off of these terms. We will consider each of the major terms in Eqn. 3.55 separately first and then combine the resulting expressions.

As an aside, the correct procedure for integration by parts in any dimension can always be deduced by returning to the basic concept behind it: form the derivative of a product, apply the appropriate product rule to maintain the order of the resulting tensor (i.e., scalars remain scalars, vectors remain vectors, etc.), integrate both sides of the resulting equation, and rearrange the result. The familiar formula for integration by parts is recovered in one dimension. However, in multiple dimensions, the Divergence theorem will be used to form a surface integral (assuming a smooth integrand). For clarity in the adjoint derivation, we are explicit about this procedure below.

First, consider the integral involving the time derivative in Eqn. 3.55. The linearization procedure for the governing equations has caused δU to appear within a time derivative, so integration by parts is used to isolate the perturbations. The integration by parts procedure gives

$$\begin{aligned} \frac{\partial}{\partial t} (\Psi^\top \delta U) &= \frac{\partial \Psi^\top}{\partial t} \delta U + \Psi^\top \frac{\partial}{\partial t} (\delta U) \\ \int_{\Omega} [\Psi^\top \delta U]_{t_o}^{t_f} d\Omega &= \int_{\Omega} \int_{t_o}^{t_f} \frac{\partial \Psi^\top}{\partial t} \delta U dt d\Omega + \int_{\Omega} \int_{t_o}^{t_f} \Psi^\top \frac{\partial}{\partial t} (\delta U) dt d\Omega \\ \int_{t_o}^{t_f} \int_{\Omega} \Psi^\top \frac{\partial}{\partial t} (\delta U) d\Omega dt &= \int_{\Omega} [\Psi^\top \delta U]_{t_o}^{t_f} d\Omega - \int_{t_o}^{t_f} \int_{\Omega} \frac{\partial \Psi^\top}{\partial t} \delta U d\Omega dt, \end{aligned} \quad (3.56)$$

where we have formed the product rule, integrated in space and time in going from the first to second lines, and changed the order of integration and rearranged in going to the third line. The time derivative has been successfully moved from the flow perturbations to the adjoint state. However, a new term involving the initial and final states of Ψ and δU has emerged. Note that the integration by parts procedure has also produced a negative sign in front of the integral involving a time derivative of the adjoint state, and the implications of this are discussed later.

The third term on the right-hand side of Eqn. 3.55 contains the Jacobians of the convective terms from the original system along with an additional term involving the Jacobian of the viscous fluxes with respect to the conservative variables. The integration by parts procedure for this term results

in the following:

$$\begin{aligned}
\nabla \cdot \left[\Psi^\top \left(\vec{A}^c - \bar{I}_5 \vec{u}_\Omega - \mu_{tot}^k \vec{A}^{vk} \right) \delta U \right] &= \nabla \Psi^\top \cdot \left(\vec{A}^c - \bar{I}_5 \vec{u}_\Omega - \mu_{tot}^k \vec{A}^{vk} \right) \delta U \\
&\quad + \Psi^\top \nabla \cdot \left(\vec{A}^c - \bar{I}_5 \vec{u}_\Omega - \mu_{tot}^k \vec{A}^{vk} \right) \delta U \\
\int_{t_o}^{t_f} \int_{\Omega} \nabla \cdot \left[\Psi^\top \left(\vec{A}^c - \bar{I}_5 \vec{u}_\Omega - \mu_{tot}^k \vec{A}^{vk} \right) \delta U \right] d\Omega dt &= \int_{t_o}^{t_f} \int_{\Omega} \nabla \Psi^\top \cdot \left(\vec{A}^c - \bar{I}_5 \vec{u}_\Omega - \mu_{tot}^k \vec{A}^{vk} \right) \delta U d\Omega dt \\
&\quad + \int_{t_o}^{t_f} \int_{\Omega} \Psi^\top \nabla \cdot \left(\vec{A}^c - \bar{I}_5 \vec{u}_\Omega - \mu_{tot}^k \vec{A}^{vk} \right) \delta U d\Omega dt \\
\int_{t_o}^{t_f} \int_{\Omega} \Psi^\top \nabla \cdot \left(\vec{A}^c - \bar{I}_5 \vec{u}_\Omega - \mu_{tot}^k \vec{A}^{vk} \right) \delta U d\Omega dt &= \int_{t_o}^{t_f} \int_{\partial\Omega} \Psi^\top \left(\vec{A}^c - \bar{I}_5 \vec{u}_\Omega - \mu_{tot}^k \vec{A}^{vk} \right) \delta U \cdot \vec{n} ds dt \\
&\quad - \int_{t_o}^{t_f} \int_{\Omega} \nabla \Psi^\top \cdot \left(\vec{A}^c - \bar{I}_5 \vec{u}_\Omega - \mu_{tot}^k \vec{A}^{vk} \right) \delta U d\Omega dt,
\end{aligned} \tag{3.57}$$

where we have formed the product rule, integrated in space and time in going from the first to second lines, and used the divergence theorem and rearranged in going to the third line. The spatial derivative has been successfully moved from the flow perturbations to the adjoint state vector, but a surface integral over the disconnected boundary of the domain has also appeared.

Lastly, the fourth term on the right-hand side of Eqn. 3.55 containing the Jacobian of the viscous fluxes with respect to the gradient of the flow variables is treated. Due to the appearance of a second derivative on δU , isolating the perturbation requires integrating by parts twice. The first integration by parts procedure gives

$$\begin{aligned}
\nabla \cdot \left[\Psi^\top \mu_{tot}^k \bar{D}^{vk} \cdot \delta(\nabla U) \right] &= \Psi^\top \nabla \cdot \left[\mu_{tot}^k \bar{D}^{vk} \cdot \delta(\nabla U) \right] + \nabla \Psi^\top \cdot \left[\mu_{tot}^k \bar{D}^{vk} \cdot \delta(\nabla U) \right] \\
\int_{t_o}^{t_f} \int_{\Omega} \nabla \cdot \left[\Psi^\top \mu_{tot}^k \bar{D}^{vk} \cdot \delta(\nabla U) \right] d\Omega dt &= \int_{t_o}^{t_f} \int_{\Omega} \Psi^\top \nabla \cdot \left[\mu_{tot}^k \bar{D}^{vk} \cdot \delta(\nabla U) \right] d\Omega dt \\
&\quad + \int_{t_o}^{t_f} \int_{\Omega} \nabla \Psi^\top \cdot \left[\mu_{tot}^k \bar{D}^{vk} \cdot \delta(\nabla U) \right] d\Omega dt \\
\int_{t_o}^{t_f} \int_{\Omega} \Psi^\top \nabla \cdot \left[\mu_{tot}^k \bar{D}^{vk} \cdot \delta(\nabla U) \right] d\Omega dt &= \int_{t_o}^{t_f} \int_{\partial\Omega} \left[\Psi^\top \mu_{tot}^k \bar{D}^{vk} \cdot \delta(\nabla U) \right] \cdot \vec{n} ds dt \\
&\quad - \int_{t_o}^{t_f} \int_{\Omega} \nabla \Psi^\top \cdot \left[\mu_{tot}^k \bar{D}^{vk} \cdot \delta(\nabla U) \right] d\Omega dt,
\end{aligned} \tag{3.58}$$

where we have again formed the product rule, integrated in space and time in going from the first to second lines, and used the divergence theorem and rearranged in going to the third line. Integration by parts is required once more for the final term in Eqn. 3.58. Noting that $\delta(\nabla U) = \nabla(\delta U)$ in a

continuum and repeating the same procedure yields

$$\begin{aligned}
\nabla \cdot \left(\nabla \Psi^\top \cdot \mu_{tot}^k \bar{\bar{D}}^{vk} \delta U \right) &= \nabla \cdot \left(\nabla \Psi^\top \cdot \mu_{tot}^k \bar{\bar{D}}^{vk} \right) \delta U + \left(\nabla \Psi^\top \cdot \mu_{tot}^k \bar{\bar{D}}^{vk} \right) \cdot \nabla (\delta U) \\
\int_{t_o}^{t_f} \int_{\Omega} \nabla \cdot \left(\nabla \Psi^\top \cdot \mu_{tot}^k \bar{\bar{D}}^{vk} \delta U \right) d\Omega dt &= \int_{t_o}^{t_f} \int_{\Omega} \nabla \cdot \left(\nabla \Psi^\top \cdot \mu_{tot}^k \bar{\bar{D}}^{vk} \right) \delta U d\Omega dt \\
&\quad + \int_{t_o}^{t_f} \int_{\Omega} \left(\nabla \Psi^\top \cdot \mu_{tot}^k \bar{\bar{D}}^{vk} \right) \cdot \nabla (\delta U) d\Omega dt \\
\int_{t_o}^{t_f} \int_{\Omega} \left(\nabla \Psi^\top \cdot \mu_{tot}^k \bar{\bar{D}}^{vk} \right) \cdot \nabla (\delta U) d\Omega dt &= \int_{t_o}^{t_f} \int_{\partial\Omega} \left(\nabla \Psi^\top \cdot \mu_{tot}^k \bar{\bar{D}}^{vk} \delta U \right) \cdot \vec{n} ds dt \\
&\quad - \int_{t_o}^{t_f} \int_{\Omega} \nabla \cdot \left(\nabla \Psi^\top \cdot \mu_{tot}^k \bar{\bar{D}}^{vk} \right) \delta U d\Omega dt. \tag{3.59}
\end{aligned}$$

Now that we have considered each of the integrals involving derivatives of perturbations in Eqn. 3.55, the results can be collected from Eqns. 3.56, 3.57, 3.58, and 3.59 and substituted back into the original expression for the variation of the objective function:

$$\begin{aligned}
\delta \mathcal{J} &= \delta J - \frac{1}{\mathbb{T}} \int_{\Omega} [\Psi^\top \delta U]_{t_o}^{t_f} d\Omega - \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_{\partial\Omega} (B_1 - B_2 + B_3) ds dt \\
&\quad - \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_{\Omega} \left[-\frac{\partial \Psi^\top}{\partial t} - \nabla \Psi^\top \cdot \left(\vec{A}^c - \bar{\bar{I}}_5 \vec{u}_\Omega - \mu_{tot}^k \vec{A}^{vk} \right) - \nabla \cdot \left(\nabla \Psi^\top \cdot \mu_{tot}^k \bar{\bar{D}}^{vk} \right) - \Psi^\top \frac{\partial \mathcal{Q}}{\partial U} \right] \delta U d\Omega dt, \tag{3.60}
\end{aligned}$$

where, as a shorthand,

$$B_1 = \Psi^\top \left(\vec{A}^c - \bar{\bar{I}}_5 \vec{u}_\Omega \right) \delta U \cdot \vec{n}, \tag{3.61}$$

$$B_2 = \Psi^\top \mu_{tot}^k \vec{A}^{vk} \delta U \cdot \vec{n} + \Psi^\top \mu_{tot}^k \bar{\bar{D}}^{vk} \cdot \nabla (\delta U) \cdot \vec{n}, \tag{3.62}$$

and

$$B_3 = \nabla \Psi^\top \cdot \mu_{tot}^k \bar{\bar{D}}^{vk} \delta U \cdot \vec{n}. \tag{3.63}$$

At this point, the flow perturbations have been removed from derivatives and isolated, but $\delta \mathcal{J}$ will not be readily computable until all terms containing δU are eliminated. For instance, the final integral in Eqn. 3.60 can be eliminated by forcing to zero the expression in brackets throughout the entire domain (space and time). This is the first appearance of the system of PDEs known as the adjoint equations, and their satisfaction becomes a requirement for computing the variation of the Lagrangian. However, the admissible boundary conditions needed for satisfying the adjoint system have not yet been specified, and this topic is discussed in the following sections.

3.5.2 Evaluating the Adjoint Boundary Integrals

Suitable boundary conditions for the adjoint system can be found by considering the remaining boundary integrals in Eqn. 3.60 while introducing the linearized boundary conditions from the governing equations where necessary. The evaluation of the remaining boundary integrals is detailed below.

Detailed Evaluation of the Integral Involving Time

By assuming an unsteady flow with time-periodic behavior (the cost function does not depend on t_f), we can impose

$$\Psi(\vec{x}, t_o) = 0 \quad (3.64)$$

and

$$\Psi(\vec{x}, t_f) = 0, \quad (3.65)$$

which forces the second term on the right-hand side of Eqn. 3.60 to vanish. While we are treating problems featuring time-periodicity in this work, this is not a requirement. Due to a reversal of the characteristic speeds in the adjoint problem (reverse time integration), we can impose an initial zero value for the adjoint state. This corresponds to a zero state at the final physical time, or $\Psi(\vec{x}, t_f) = 0$. However, the term involving $[\Psi^T \delta U]_{t_o}$ in Eqn. 3.60 must be included or forced to vanish with an admissible choice for $\Psi(\vec{x}, t_o)$.

Detailed Evaluation of Boundary Integrals Involving B_1 on S

The expression for B_1 in Eqn. 3.61 contains the flow perturbations δU , and given our knowledge of the convective flux Jacobian, it can be evaluated explicitly. For clarity, the manipulations are performed starting in two dimensions, but then the resulting expressions are immediately generalized to three dimensions using vector notation. Recalling also that this evaluation occurs on the surface, we impose the solid wall boundary condition for simplification. With $a_0 = (\gamma - 1)$ and $\phi = (\gamma - 1) \frac{|\vec{v}|^2}{2}$

defined for convenience, the expression can be evaluated and simplified as follows:

$$\begin{aligned}
B_1 &= \Psi^\top \left(\vec{A}^c - \bar{I}_5 \vec{u}_\Omega \right) \delta U \cdot \vec{n} \\
&= \Psi^\top \left(\vec{A}^c - \bar{I}_5 \vec{u}_\Omega \right) \cdot \vec{n} M \delta V \\
&= \Psi^\top \begin{pmatrix} -\vec{u}_\Omega \cdot \vec{n} & n_x & n_y & \cdot \\ -v_1(\vec{v} \cdot \vec{n}) + \phi n_x & v_n - (a_o - 1)v_1 n_x & v_1 n_y - a_o v_2 n_x & a_o n_x \\ -v_2(\vec{v} \cdot \vec{n}) + \phi n_y & v_2 n_x - a_o v_1 n_y & v_n - (a_o - 1)v_2 n_y & a_o n_y \\ (\vec{v} \cdot \vec{n})(\phi - H) & -a_o v_1(\vec{v} \cdot \vec{n}) + H n_x & -a_o v_2(\vec{v} \cdot \vec{n}) + H n_y & \gamma \vec{v} \cdot \vec{n} - \vec{u}_\Omega \cdot \vec{n} \end{pmatrix} M \delta V \\
&= \Psi^\top \begin{pmatrix} v_n & \rho n_x & \rho n_y & \cdot \\ v_1 v_n & \rho v_n + \rho v_1 n_x & \rho v_1 n_y & n_x \\ v_2 v_n & \rho v_2 n_x & \rho v_n + \rho v_2 n_y & n_y \\ \frac{\phi}{a_o} v_n & \rho v_1 v_n + \rho H n_x & \rho v_2 v_n + \rho H n_y & \vec{v} \cdot \vec{n} + \frac{1}{a_o} v_n \end{pmatrix} \delta V \\
&= \{\psi_\rho, \psi_{\rho v_1}, \psi_{\rho v_2}, \psi_{\rho E}\} \begin{pmatrix} \cdot & \rho n_x & \rho n_y & \cdot \\ \cdot & \rho v_1 n_x & \rho v_1 n_y & n_x \\ \cdot & \rho v_2 n_x & \rho v_2 n_y & n_y \\ \cdot & \rho H n_x & \rho H n_y & \vec{v} \cdot \vec{n} \end{pmatrix} \begin{Bmatrix} \delta \rho \\ \delta v_1 \\ \delta v_2 \\ \delta p \end{Bmatrix} \\
&= \{ \cdot, (\rho \psi_\rho + \rho \vec{v} \cdot \vec{\varphi} + \rho H \psi_{\rho E}) n_x, (\rho \psi_\rho + \rho \vec{v} \cdot \vec{\varphi} + \rho H \psi_{\rho E}) n_y, \vec{\varphi} \cdot \vec{n} + \psi_{\rho E}(\vec{v} \cdot \vec{n}) \} \begin{Bmatrix} \delta \rho \\ \delta v_1 \\ \delta v_2 \\ \delta p \end{Bmatrix} \\
&= \delta v_1 (\rho \psi_\rho + \rho \vec{v} \cdot \vec{\varphi} + \rho H \psi_{\rho E}) n_x + \delta v_2 (\rho \psi_\rho + \rho \vec{v} \cdot \vec{\varphi} + \rho H \psi_{\rho E}) n_y + \delta p [\vec{\varphi} \cdot \vec{n} + \psi_{\rho E}(\vec{v} \cdot \vec{n})] \\
&= (\rho \psi_\rho + \rho \vec{v} \cdot \vec{\varphi} + \rho H \psi_{\rho E})(\delta \vec{v} \cdot \vec{n}) + [\vec{\varphi} \cdot \vec{n} + \psi_{\rho E}(\vec{v} \cdot \vec{n})] \delta p \\
&= \vartheta \delta \vec{v} \cdot \vec{n} + [\vec{\varphi} \cdot \vec{n} + \psi_{\rho E}(\vec{v} \cdot \vec{n})] \delta p, \tag{3.66}
\end{aligned}$$

where $v_n = (\vec{v} - \vec{u}_\Omega) \cdot \vec{n}_\infty$ as a shorthand, $M = \frac{\partial U}{\partial V}$ is the transformation matrix from conservative to primitive variables (given in the appendix), $\delta V = \{\delta \rho, \delta \vec{v}, \delta p\}^\top$ is the vector of primitive variable perturbations, and $\vartheta = (\rho \psi_\rho + \rho \vec{v} \cdot \vec{\varphi} + \rho H \psi_{\rho E})$ as a shorthand.

Note that, in going from the fourth to the fifth line of Eqn. 3.66, we have used the solid wall boundary condition to remove any term that includes $(\vec{v} - \vec{u}_\Omega) \cdot \vec{n}$, and that the final result remains the same for both viscous and inviscid flow. For flows on stationary domains, the term involving $\vec{v} \cdot \vec{n}$ in Eqn. 3.66 vanishes.

Detailed Evaluation of Boundary Integrals Involving B_2 on S

A simplification for the terms involving B_2 from Eqn. 3.62 can be made by recalling the definition of the variation of the viscous flux vector under the frozen viscosity assumption:

$$\delta(\mu_{tot}^k \vec{F}^{vk}) = \mu_{tot}^k \frac{\partial \vec{F}^{vk}}{\partial U} \delta U + \mu_{tot}^k \frac{\partial \vec{F}^{vk}}{\partial (\nabla U)} \cdot \delta(\nabla U) = \mu_{tot}^k \vec{A}^{vk} \delta U + \mu_{tot}^k \vec{D}^{vk} \cdot \delta(\nabla U). \tag{3.67}$$

Introducing this definition into Eqn. 3.62 and again noting that $\delta(\nabla U) = \nabla(\delta U)$ in a continuum allows for simplification without the need for extensive algebra:

$$\begin{aligned}
B_2 &= \Psi^\top \mu_{tot}^k \vec{A}^{vk} \delta U \cdot \vec{n} + \Psi^\top \mu_{tot}^k \vec{\bar{D}}^{vk} \cdot \nabla(\delta U) \cdot \vec{n} \\
&= \Psi^\top \left[\mu_{tot}^k \vec{A}^{vk} \delta U + \mu_{tot}^k \vec{\bar{D}}^{vk} \cdot \nabla(\delta U) \right] \cdot \vec{n} \\
&= \Psi^\top \delta(\mu_{tot}^k \vec{F}^{vk}) \cdot \vec{n} \\
&= \Psi^\top \delta(\mu_{tot}^1 \vec{F}^{v1}) \cdot \vec{n} + \Psi^\top \delta(\mu_{tot}^2 \vec{F}^{v2}) \cdot \vec{n} \\
&= \Psi^\top \delta \left\{ \begin{array}{c} \cdot \\ \bar{\sigma} \\ \bar{\sigma} \cdot \vec{v} \end{array} \right\} \cdot \vec{n} + \Psi^\top \delta \left\{ \begin{array}{c} \cdot \\ \mu_{tot}^2 c_p \nabla T \end{array} \right\} \cdot \vec{n} \\
&= \{\psi_\rho, \vec{\varphi}, \psi_{\rho E}\} \left\{ \begin{array}{c} \cdot \\ \delta \bar{\sigma} \cdot \vec{n} \\ \delta(\bar{\sigma} \cdot \vec{v}) \cdot \vec{n} \end{array} \right\} + \{\psi_\rho, \vec{\varphi}, \psi_{\rho E}\} \left\{ \begin{array}{c} \cdot \\ \mu_{tot}^2 c_p \delta(\nabla T) \cdot \vec{n} \end{array} \right\} \\
&= \vec{\varphi} \cdot \delta \bar{\sigma} \cdot \vec{n} + \psi_{\rho E} \delta(\bar{\sigma} \cdot \vec{v}) \cdot \vec{n} + \psi_{\rho E} \mu_{tot}^2 c_p \delta(\nabla T) \cdot \vec{n} \\
&= \vec{\varphi} \cdot \delta \bar{\sigma} \cdot \vec{n} + \psi_{\rho E} \vec{v} \cdot \delta \bar{\sigma} \cdot \vec{n} + \psi_{\rho E} \delta \vec{v} \cdot \bar{\sigma} \cdot \vec{n} + \psi_{\rho E} \mu_{tot}^2 c_p \partial_n(\delta T). \tag{3.68}
\end{aligned}$$

Note that, for flows on stationary domains, the second term on the right-hand side of Eqn. 3.68 vanishes due to the no-slip condition ($\vec{v} = 0$).

Detailed Evaluation of Boundary Integrals Involving B_3 on S

The expression for B_3 from Eqn. 3.63 can be divided into components related to the viscous stresses (B_3^1) and heat transfer (B_3^2) as

$$B_3 = B_3^k = B_3^1 + B_3^2, \tag{3.69}$$

and these terms are considered separately in two dimensions before generalizing to three dimensions using vector notation. First, we expand B_3^k in two dimensions:

$$\begin{aligned}
B_3^k &= \nabla \Psi^\top \cdot \mu_{tot}^k \vec{\bar{D}}^{vk} \delta U \cdot \vec{n} \\
&= (\partial_x \Psi^\top, \partial_y \Psi^\top) \cdot \mu_{tot}^k \begin{pmatrix} D_{xx}^{vk} & D_{xy}^{vk} \\ D_{yx}^{vk} & D_{yy}^{vk} \end{pmatrix} \delta U \cdot (n_x, n_y) \\
&= (\partial_x \Psi^\top, \partial_y \Psi^\top) \cdot \mu_{tot}^k \begin{pmatrix} D_{xx}^{vk} n_x + D_{xy}^{vk} n_y \\ D_{yx}^{vk} n_x + D_{yy}^{vk} n_y \end{pmatrix} \delta U \\
&= \mu_{tot}^k [\partial_x \Psi^\top (D_{xx}^{vk} n_x + D_{xy}^{vk} n_y) + \partial_y \Psi^\top (D_{yx}^{vk} n_x + D_{yy}^{vk} n_y)] \delta U \\
&= \mu_{tot}^k (\partial_x \Psi^\top D_{xx}^{vk} n_x \delta U + \partial_x \Psi^\top D_{xy}^{vk} n_y \delta U + \partial_y \Psi^\top D_{yx}^{vk} n_x \delta U + \partial_y \Psi^\top D_{yy}^{vk} n_y \delta U), \tag{3.70}
\end{aligned}$$

where

$$\partial_x \Psi^\top = \{\partial_x \psi_\rho, \partial_x \psi_{\rho v_1}, \partial_x \psi_{\rho v_2}, \partial_x \psi_{\rho E}\}, \quad (3.71)$$

$$\partial_y \Psi^\top = \{\partial_y \psi_\rho, \partial_y \psi_{\rho v_1}, \partial_y \psi_{\rho v_2}, \partial_y \psi_{\rho E}\}, \quad (3.72)$$

and $\delta U = \{\delta \rho, \delta(\rho v_1), \delta(\rho v_2), \delta(\rho E)\}^\top$. Each term in Eqn. 3.70 can be evaluated given our knowledge of the Jacobian of the viscous flux vector with respect to the gradient of the conservative variables $\bar{\bar{D}}^{vk}$ (see the appendix) as well as the behavior of the governing equations at the no-slip wall.

Viscous Stress Component Here, the terms involving the Jacobian of the viscous stresses $\bar{\bar{D}}^{v1}$ that make up B_3^1 in Eqn. 3.69 are considered. More specifically, the first term on the right-hand side of Eqn. 3.70 is evaluated in a detailed manner as an example before systematically extending and simplifying the remaining terms:

$$\begin{aligned} \partial_x \Psi^\top D_{xx}^{v1} n_x \delta U &= \partial_x \Psi^\top \frac{n_x}{\rho} \begin{pmatrix} \cdot & \cdot & \cdot & \cdot \\ -(1 + \frac{1}{3})v_1 & (1 + \frac{1}{3}) & \cdot & \cdot \\ -v_2 & \cdot & 1 & \cdot \\ -\bar{v}^2 - \frac{1}{3}v_1^2 & (1 + \frac{1}{3})v_1 & v_2 & \cdot \end{pmatrix} \begin{Bmatrix} \delta \rho \\ \delta(\rho v_1) \\ \delta(\rho v_2) \\ \delta(\rho E) \end{Bmatrix} \\ &= \partial_x \Psi^\top \frac{n_x}{\rho} \begin{pmatrix} \cdot & \cdot & \cdot & \cdot \\ -(1 + \frac{1}{3})v_1 & (1 + \frac{1}{3}) & \cdot & \cdot \\ -v_2 & \cdot & 1 & \cdot \\ -\bar{v}^2 - \frac{1}{3}v_1^2 & (1 + \frac{1}{3})v_1 & v_2 & \cdot \end{pmatrix} \begin{Bmatrix} \delta \rho \\ v_1 \delta \rho + \rho \delta v_1 \\ v_2 \delta \rho + \rho \delta v_2 \\ E \delta \rho + \rho \delta E \end{Bmatrix} \\ &= \{\partial_x \psi_\rho, \partial_x \psi_{\rho v_1}, \partial_x \psi_{\rho v_2}, \partial_x \psi_{\rho E}\} \frac{n_x}{\rho} \begin{Bmatrix} \cdot \\ (1 + \frac{1}{3})\rho \delta v_1 \\ \rho \delta v_2 \\ (1 + \frac{1}{3})\rho v_1 \delta v_1 + \rho v_2 \delta v_2 \end{Bmatrix} \\ &= \partial_x \psi_{\rho v_1} (1 + \frac{1}{3}) \delta v_1 n_x + \partial_x \psi_{\rho v_2} \delta v_2 n_x + \partial_x \psi_{\rho E} (1 + \frac{1}{3}) v_1 \delta v_1 n_x + \partial_x \psi_{\rho E} v_2 \delta v_2 n_x. \end{aligned} \quad (3.73)$$

It can be shown in a similar fashion that the other three terms on the right-hand side of Eqn. 3.70 for $k = 1$ can be simplified to give

$$\partial_x \Psi^\top D_{xy}^{v1} n_y \delta U = -\partial_x \psi_{\rho v_1} \frac{2}{3} \delta v_2 n_y + \partial_x \psi_{\rho v_2} \delta v_1 n_y - \partial_x \psi_{\rho E} \frac{2}{3} v_1 \delta v_2 n_y + \partial_x \psi_{\rho E} v_2 \delta v_1 n_y, \quad (3.74)$$

$$\partial_y \Psi^\top D_{yx}^{v1} n_x \delta U = \partial_y \psi_{\rho v_1} \delta v_2 n_x - \partial_y \psi_{\rho v_2} \frac{2}{3} \delta v_1 n_x + \partial_y \psi_{\rho E} v_1 \delta v_2 n_x - \partial_y \psi_{\rho E} \frac{2}{3} v_2 \delta v_1 n_x, \quad (3.75)$$

and

$$\partial_y \Psi^\top D_{yy}^{v1} n_y \delta U = \partial_y \psi_{\rho v_1} \delta v_1 n_y + \partial_y \psi_{\rho v_2} (1 + \frac{1}{3}) \delta v_2 n_y + \partial_y \psi_{\rho E} v_1 \delta v_1 n_y + \partial_y \psi_{\rho E} (1 + \frac{1}{3}) v_2 \delta v_2 n_y. \quad (3.76)$$

The results from Eqns. 3.73–3.76 can be introduced into Eqn. 3.70 with $k = 1$, combined, and organized by the normal components and velocity perturbations as

$$\begin{aligned}
B_3^1 &= \partial_x \psi_{\rho v_1} \left(1 + \frac{1}{3}\right) \delta v_1 n_x + \partial_x \psi_{\rho v_2} \delta v_2 n_x + \partial_x \psi_{\rho E} \left(1 + \frac{1}{3}\right) v_1 \delta v_1 n_x + \partial_x \psi_{\rho E} v_2 \delta v_2 n_x \\
&\quad - \partial_x \psi_{\rho v_1} \frac{2}{3} \delta v_2 n_y + \partial_x \psi_{\rho v_2} \delta v_1 n_y - \partial_x \psi_{\rho E} \frac{2}{3} v_1 \delta v_2 n_y + \partial_x \psi_{\rho E} v_2 \delta v_1 n_y \\
&\quad + \partial_y \psi_{\rho v_1} \delta v_2 n_x - \partial_y \psi_{\rho v_2} \frac{2}{3} \delta v_1 n_x + \partial_y \psi_{\rho E} v_1 \delta v_2 n_x - \partial_y \psi_{\rho E} \frac{2}{3} v_2 \delta v_1 n_x \\
&\quad + \partial_y \psi_{\rho v_1} \delta v_1 n_y + \partial_y \psi_{\rho v_2} \left(1 + \frac{1}{3}\right) \delta v_2 n_y + \partial_y \psi_{\rho E} v_1 \delta v_1 n_y + \partial_y \psi_{\rho E} \left(1 + \frac{1}{3}\right) v_2 \delta v_2 n_y \\
&= n_x \left(\frac{4}{3} \partial_x \psi_{\rho v_1} - \frac{2}{3} \partial_y \psi_{\rho v_2} \right) \delta v_1 + n_x (\partial_x \psi_{\rho v_2} + \partial_y \psi_{\rho v_1}) \delta v_2 \\
&\quad + n_y (\partial_x \psi_{\rho v_2} + \partial_y \psi_{\rho v_1}) \delta v_1 + n_y \left(\frac{4}{3} \partial_y \psi_{\rho v_2} - \frac{2}{3} \partial_x \psi_{\rho v_1} \right) \delta v_2 \\
&\quad + n_x \left(\frac{4}{3} \partial_x \psi_{\rho E} v_1 - \frac{2}{3} \partial_y \psi_{\rho E} v_2 \right) \delta v_1 + n_x (\partial_x \psi_{\rho E} v_2 + \partial_y \psi_{\rho E} v_1) \delta v_2 \\
&\quad + n_y (\partial_x \psi_{\rho E} v_2 + \partial_y \psi_{\rho E} v_1) \delta v_1 + n_y \left(\frac{4}{3} \partial_y \psi_{\rho E} v_2 - \frac{2}{3} \partial_x \psi_{\rho E} v_1 \right) \delta v_2 \\
&= (n_x, n_y) \cdot \left(\begin{array}{l} \left(\frac{4}{3} \partial_x \psi_{\rho v_1} - \frac{2}{3} \partial_y \psi_{\rho v_2} \right) \delta v_1 + (\partial_x \psi_{\rho v_2} + \partial_y \psi_{\rho v_1}) \delta v_2 \\ (\partial_x \psi_{\rho v_2} + \partial_y \psi_{\rho v_1}) \delta v_1 + \left(\frac{4}{3} \partial_y \psi_{\rho v_2} - \frac{2}{3} \partial_x \psi_{\rho v_1} \right) \delta v_2 \end{array} \right) \\
&\quad + (n_x, n_y) \cdot \left(\begin{array}{l} \left(\frac{4}{3} \partial_x \psi_{\rho E} v_1 - \frac{2}{3} \partial_y \psi_{\rho E} v_2 \right) \delta v_1 + (\partial_x \psi_{\rho E} v_2 + \partial_y \psi_{\rho E} v_1) \delta v_2 \\ (\partial_x \psi_{\rho E} v_2 + \partial_y \psi_{\rho E} v_1) \delta v_1 + \left(\frac{4}{3} \partial_y \psi_{\rho E} v_2 - \frac{2}{3} \partial_x \psi_{\rho E} v_1 \right) \delta v_2 \end{array} \right) \\
&= (n_x, n_y) \cdot \left(\begin{array}{cc} \frac{4}{3} \partial_x \psi_{\rho v_1} - \frac{2}{3} \partial_y \psi_{\rho v_2} & \partial_x \psi_{\rho v_2} + \partial_y \psi_{\rho v_1} \\ \partial_x \psi_{\rho v_2} + \partial_y \psi_{\rho v_1} & \frac{4}{3} \partial_y \psi_{\rho v_2} - \frac{2}{3} \partial_x \psi_{\rho v_1} \end{array} \right) \cdot (\delta v_1, \delta v_2) \\
&\quad + (n_x, n_y) \cdot \left(\begin{array}{cc} \frac{4}{3} \partial_x \psi_{\rho E} v_1 - \frac{2}{3} \partial_y \psi_{\rho E} v_2 & \partial_x \psi_{\rho E} v_2 + \partial_y \psi_{\rho E} v_1 \\ \partial_x \psi_{\rho E} v_2 + \partial_y \psi_{\rho E} v_1 & \frac{4}{3} \partial_y \psi_{\rho E} v_2 - \frac{2}{3} \partial_x \psi_{\rho E} v_1 \end{array} \right) \cdot (\delta v_1, \delta v_2), \quad (3.77)
\end{aligned}$$

and the final result can be immediately generalized in vector notation as

$$\begin{aligned}
B_3^1 &= \mu_{tot}^1 \left[\bar{\vec{n}} \cdot (\nabla \bar{\vec{\varphi}} + \nabla \bar{\vec{\varphi}}^\top - \frac{2}{3} \bar{I} \nabla \cdot \bar{\vec{\varphi}}) \cdot \delta \bar{\vec{v}} + \bar{\vec{n}} \cdot (\nabla \psi_{\rho E} \bar{\vec{v}} + \nabla \psi_{\rho E} \bar{\vec{v}}^\top - \frac{2}{3} \bar{I} \nabla \psi_{\rho E} \cdot \bar{\vec{v}}) \cdot \delta \bar{\vec{v}} \right] \\
&= \bar{\vec{n}} \cdot \bar{\vec{\Sigma}}^\varphi \cdot \delta \bar{\vec{v}} + \bar{\vec{n}} \cdot \bar{\vec{\Sigma}}^{\psi_{\rho E}} \cdot \delta \bar{\vec{v}} \\
&= \bar{\vec{n}} \cdot \left(\bar{\vec{\Sigma}}^\varphi + \bar{\vec{\Sigma}}^{\psi_{\rho E}} \right) \cdot \delta \bar{\vec{v}}, \quad (3.78)
\end{aligned}$$

where we have introduced $\bar{\vec{\Sigma}}^\varphi = \mu_{tot}^1 (\nabla \bar{\vec{\varphi}} + \nabla \bar{\vec{\varphi}}^\top - \frac{2}{3} \bar{I} \nabla \cdot \bar{\vec{\varphi}})$ and $\bar{\vec{\Sigma}}^{\psi_{\rho E}} = \mu_{tot}^1 (\nabla \psi_{\rho E} \bar{\vec{v}} + \nabla \psi_{\rho E} \bar{\vec{v}}^\top - \frac{2}{3} \bar{I} \nabla \psi_{\rho E} \cdot \bar{\vec{v}})$ as shorthand expressions.

Heat Transfer Component The same process for the terms involving \bar{D}^{v2} that make up B_3^2 in Eqn. 3.69 can also be performed. Again, the first term on the right-hand side of Eqn. 3.70 is evaluated in a detailed manner in two dimensions as an example before systematically extending

and simplifying the remaining terms:

$$\begin{aligned}
\partial_x \Psi^\top D_{xx}^{v^2} n_x \delta U &= \partial_x \Psi^\top \frac{n_x \gamma}{\rho} \begin{pmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \frac{|\bar{v}|^2}{2} - \frac{p}{\rho(\gamma-1)} & -v_1 & -v_2 & 1 \end{pmatrix} \begin{Bmatrix} \delta \rho \\ \delta(\rho v_1) \\ \delta(\rho v_2) \\ \delta(\rho E) \end{Bmatrix} \\
&= \partial_x \Psi^\top \frac{n_x \gamma}{\rho} \begin{pmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \frac{|\bar{v}|^2}{2} - \frac{p}{\rho(\gamma-1)} & -v_1 & -v_2 & 1 \end{pmatrix} \begin{Bmatrix} \delta \rho \\ v_1 \delta \rho + \rho \delta v_1 \\ v_2 \delta \rho + \rho \delta v_2 \\ E \delta \rho + \rho \delta E \end{Bmatrix} \\
&= \partial_x \Psi^\top \frac{n_x \gamma}{\rho} \left\{ \begin{array}{c} \cdot \\ \cdot \\ \cdot \\ \frac{|\bar{v}|^2}{2} \delta \rho - \frac{p}{\rho(\gamma-1)} \delta \rho - |\bar{v}|^2 \delta \rho - \rho v_1 \delta v_1 - \rho v_2 \delta v_2 + E \delta \rho + \rho \delta E \end{array} \right\}. \tag{3.79}
\end{aligned}$$

Recalling the definition of the energy, or $E = \frac{|\bar{v}|^2}{2} + \frac{p}{\rho(\gamma-1)}$, we can write the total variation of the energy as

$$\begin{aligned}
\delta E &= \delta \left(\frac{|\bar{v}|^2}{2} + \frac{p}{\rho(\gamma-1)} \right) \\
&= \delta \left(\frac{1}{2}(v_1^2 + v_2^2) \right) + \delta \left(\frac{p}{\rho(\gamma-1)} \right) \\
&= v_1 \delta v_1 + v_2 \delta v_2 + \frac{1}{\rho(\gamma-1)} \delta p + \frac{p}{\rho^2(\gamma-1)} \delta \rho, \tag{3.80}
\end{aligned}$$

and substituting the expressions for the energy and its variation from Eqn. 3.80 into Eqn. 3.79 yields

$$\partial_x \Psi^\top D_{xx}^{v^2} n_x \delta U = \partial_x \Psi^\top n_x \left\{ \begin{array}{c} \cdot \\ \cdot \\ \cdot \\ \frac{\gamma}{(\gamma-1)} \frac{1}{\rho} \delta p - \frac{\gamma}{(\gamma-1)} \frac{p}{\rho^2} \delta \rho \end{array} \right\}. \tag{3.81}$$

Ideally, we would like to further simplify the remaining variations if possible, and by using the equation of state for a perfect gas in Eqn. 2.7 and the relationship in Eqn. 2.8, we can relate the variations in pressure and density to a variation in the temperature:

$$\begin{aligned}
\delta T &= \delta \left(\frac{p}{\rho R} \right) = \delta \left(\frac{\gamma}{c_p(\gamma-1)} \frac{p}{\rho} \right) \\
&= \frac{\gamma}{c_p(\gamma-1)} \frac{1}{\rho} \delta p - \frac{\gamma}{c_p(\gamma-1)} \frac{p}{\rho^2} \delta \rho. \tag{3.82}
\end{aligned}$$

Upon substitution into Eqn. 3.81, we arrive at a final expression:

$$\partial_x \Psi^\top D_{xx}^{v2} n_x \delta U = \{\partial_x \psi_\rho, \partial_x \psi_{\rho v_1}, \partial_x \psi_{\rho v_2}, \partial_x \psi_{\rho E}\} n_x \left\{ \begin{array}{c} \cdot \\ \cdot \\ \cdot \\ c_p \delta T \end{array} \right\} = c_p \partial_x \psi_{\rho E} n_x \delta T. \quad (3.83)$$

Similarly, the other three terms on the right hand side of Eqn. 3.70 can be simplified to give

$$\partial_x \Psi^\top D_{xy}^{v1} n_y \delta U = 0, \quad (3.84)$$

$$\partial_y \Psi^\top D_{yx}^{v1} n_x \delta U = 0, \quad (3.85)$$

and

$$\partial_y \Psi^\top D_{yy}^{v1} n_y \delta U = c_p \partial_y \psi_{\rho E} n_y \delta T, \quad (3.86)$$

and after combining Eqns. 3.83–3.86, we can immediately generalize the result for B_3^2 to vector notation as

$$\begin{aligned} B_3^2 &= \mu_{tot}^2 c_p (\partial_x \psi_{\rho E} n_x + \partial_y \psi_{\rho E} n_y) \delta T \\ &= \mu_{tot}^2 c_p (\vec{n} \cdot \nabla \psi_{\rho E}) \delta T \\ &= \mu_{tot}^2 c_p \partial_n (\psi_{\rho E}) \delta T. \end{aligned} \quad (3.87)$$

Total Contribution from B_3 The contribution from evaluating the surface integral involving term B_3 is the sum of the stress tensor (B_3^1) and heat transfer (B_3^2) components derived above. Therefore, summing the results in Eqns. 3.78 and 3.87 gives

$$B_3 = B_3^1 + B_3^2 = \vec{n} \cdot \left(\bar{\bar{\Sigma}}^\varphi + \bar{\bar{\Sigma}}^{\psi_{\rho E}} \right) \cdot \delta \vec{v} + \mu_{tot}^2 c_p \partial_n (\psi_{\rho E}) \delta T \quad (3.88)$$

Detailed Evaluation of Boundary Integrals on Γ_∞

Each of the boundary integrals involving B_1 , B_2 , and B_3 is over the entire disconnected boundary of the domain, which includes the far-field. Here, we assume that the far-field is far enough removed from the surface S that the viscous perturbation terms (B_2 and B_3) are negligible on this boundary. This leaves only the convective boundary term B_1 at the far-field, and we would like to remove any contributions from this integral.

The integral over Γ_∞ involving B_1 in Eqn. 3.60 will indeed vanish if the scalar integrand is zero at every point on the boundary with each physical time step, or

$$\Psi^\top \left(\vec{A}^c - \bar{\bar{I}}_5 \vec{u}_\Omega \right) \cdot \vec{n}_\infty \delta U = 0. \quad (3.89)$$

Therefore, we seek the adjoint state Ψ at the boundary that eliminates any contribution from this

Table 3.1: Number of physical (P) and numerical (N) boundary conditions required for the direct and adjoint problems in three dimensions (the state vector U has five components).

	Supersonic Inflow	Subsonic Inflow	Subsonic Outflow	Supersonic Outflow
Direct	5 P, 0 N	4 P, 1 N	1 P, 4 N	0 P, 5 N
Adjoint	0 P, 5 N	1 P, 4 N	4 P, 1 N	5 P, 0 N

integral to the variation of the functional by satisfying the preceding expression. A common strategy for the removal of this integral is the imposition of a homogeneous adjoint boundary condition: $\Psi = \mathbf{0}$. While this does force Eqn. 4.61 to hold, it may be an over-specification and does not take into account the mathematical character of the equations at the boundary. Compatibility between the governing PDEs and the boundary conditions should be maintained in order to avoid issues related to solution accuracy or code convergence.

The concept of *physical* and *numerical* boundary conditions, as discussed by Hirsch [48], is useful when considering the behavior of the governing PDEs and their adjoint equations at a boundary. First, the convective flux Jacobian \vec{A}^c is diagonalized, and the resulting eigenvalues represent the characteristic speeds for the propagation of information within the fluid. Note that with the ALE form of the equations, the characteristic speeds also include the domain velocity \vec{u}_Ω . Incoming characteristics (negative eigenvalues) correspond to the propagation of information originating at the boundary into the interior of the domain. At these locations, physical boundary conditions, such as free-stream conditions, stagnation conditions, or back pressure, are prescribed. Outgoing characteristics (positive eigenvalues) correspond to information propagation originating from the interior out of the domain. Here, numerical boundary conditions are imposed that extrapolate the characteristic variable information from within the domain in order to complete the update to the state vector at the boundary, i.e., the number of physical and numerical conditions imposed must total the length of the state vector.

It is important to highlight that the sign of the characteristic velocities has been flipped in the adjoint problem, causing characteristic information to propagate in the reverse direction. This sign reversal occurs due to the integration by parts procedure during the adjoint derivation, as evidenced by the appearance of a negative sign in front of the final term of Eqn. 3.57. The result is that the required type of boundary conditions for the direct and adjoint problem are also reversed, i.e., physical conditions in the direct problem become numerical ones in the adjoint problem and vice versa, and the reversal implies the existence of a complementary set of conditions for the adjoint problem. This relationship is summarized for inflow and outflow boundaries in Table 3.1.

Based on this idea, Giles and Pierce [40] proposed suitable physical adjoint boundary conditions involving the characteristic speeds based on a manipulation of Eqn. 4.61 after introducing the diagonalization of the flux Jacobian. Recent work by Hayashi et al. [46] advances the notion of characteristic-based boundary conditions for the adjoint equations by connecting the direction of information propagation to the imposition of physical conditions at the boundaries in the direct problem. More specifically, rather than impose conditions using the characteristic variables,

one should consider the realizable perturbations in the flow solution, δU , allowed at the boundaries given the particular choice of physical boundary conditions in the direct problem. The latter derivation approach will now be further detailed and extended in systematic fashion.

Consider the far-field boundary Γ_∞ to be split into a subsonic inflow and subsonic outflow region for the direct problem. At the inflow in three dimensions, four physical conditions can be prescribed, and for clarity in the derivation, we choose to specify mass flow (ρ and \vec{v}) at this boundary. At the outlet boundary, only one physical condition can be imposed, and we choose the typical imposition of back pressure (p). Therefore, the type of physical conditions chosen for the direct problem has constrained the allowable perturbations at the boundaries: we have prescribed $\delta\rho = \delta\vec{v} = 0$ at the inlet and $\delta p = 0$ at the outlet. To derive suitable adjoint boundary conditions for this situation, we introduce these primitive variable perturbations directly into Eqn. 4.61:

$$\Psi^\top \left(\vec{A}^c - \bar{I}_5 \vec{u}_\Omega \right) \cdot \vec{n}_\infty M \delta V = 0, \quad (3.90)$$

where $M = \frac{\partial U}{\partial V}$ is the transformation matrix from conservative to primitive variables (see the appendix) and $\delta V = \{\delta\rho, \delta\vec{v}, \delta p\}^\top$ is the vector of primitive variable perturbations. After taking the transpose of Eqn. 3.90 and evaluating $(\vec{A}^c - \bar{I}_5 \vec{u}_\Omega) \cdot \vec{n}_\infty M$ in three dimensions, we can write the expanded result as

$$\begin{pmatrix} \delta\rho \\ \delta v_1 \\ \delta v_2 \\ \delta v_3 \\ \delta p \end{pmatrix}^\top \begin{pmatrix} v_n & uv_n & vv_n & wv_n & \frac{\vec{v}^2}{2} v_n \\ \rho n_x & \rho u n_x + \rho v_n & \rho v n_x & \rho w n_x & \rho H n_x + \rho u v_n \\ \rho n_y & \rho u n_y & \rho v n_y + \rho v_n & \rho w n_y & \rho H n_y + \rho v v_n \\ \rho n_z & \rho u n_z & \rho v n_z & \rho w n_z + \rho v_n & \rho H n_z + \rho w v_n \\ 0 & n_x & n_y & n_z & \frac{1}{\gamma-1} [\gamma(\vec{v} \cdot \vec{n}) - (\vec{u}_\Omega \cdot \vec{n})] \end{pmatrix} \begin{pmatrix} \psi_\rho \\ \psi_{\rho v_1} \\ \psi_{\rho v_2} \\ \psi_{\rho v_3} \\ \psi_{\rho E} \end{pmatrix} = 0, \quad (3.91)$$

where $v_n = (\vec{v} - \vec{u}_\Omega) \cdot \vec{n}_\infty$ as a shorthand, $\vec{n}_\infty = \{n_x, n_y, n_z\}^\top$, and $\vec{v}^2 = v_1^2 + v_2^2 + v_3^2$. In this format, physical boundary conditions for Ψ are easily recovered by imposing the constraints on the flow perturbations resulting from the choice of direct boundary conditions and then manipulating the remaining expressions.

For example, at the inlet, the imposition of mass flow forces $\delta\rho = \delta\vec{v} = 0$, and therefore, the terms associated with the products in the first four rows of Eqn. 3.91 are all zero. As δp is the only allowable perturbation in the flow variables remaining, in order to completely remove contributions from the boundary integral, the following expression must be satisfied:

$$\delta p \left\{ \vec{\varphi} \cdot \vec{n}_\infty + \frac{\psi_{\rho E}}{\gamma-1} [\gamma(\vec{v} \cdot \vec{n}_\infty) - (\vec{u}_\Omega \cdot \vec{n}_\infty)] \right\} = 0. \quad (3.92)$$

The perturbation δp is arbitrary, so setting the bracketed expression in Eqn. 3.92 to zero gives

$$\psi_{\rho E} = - \frac{\gamma-1}{\gamma(\vec{v} \cdot \vec{n}_\infty) - (\vec{u}_\Omega \cdot \vec{n}_\infty)} \vec{\varphi} \cdot \vec{n}_\infty, \quad (3.93)$$

which represents the admissible physical boundary condition at the mass flow inlet.

Using the same derivation approach, we can determine physical boundary conditions for the adjoint variables at the outlet boundary, where only $\delta p = 0$ and all other flow variable perturbations are allowable. This eliminates any contributions from the final row of the expression in Eqn. 3.91, and leaves a system of four equations with five unknowns from the first four rows of Eqn. 3.91. Choosing $\psi_{\rho E}$ as the free variable and solving the other four equations in terms of it gives the following four physical boundary conditions:

$$\psi_{\rho} = \psi_{\rho E} \left[\frac{\gamma e (\vec{v} \cdot \vec{n}_{\infty})}{(\vec{v} - \vec{u}_{\Omega}) \cdot \vec{n}_{\infty}} + \frac{\vec{v}^2}{2} \right], \quad (3.94)$$

$$\psi_{\rho v_1} = -\psi_{\rho E} \left[\frac{\gamma e n_x}{(\vec{v} - \vec{u}_{\Omega}) \cdot \vec{n}_{\infty}} + v_1 \right], \quad (3.95)$$

$$\psi_{\rho v_2} = -\psi_{\rho E} \left[\frac{\gamma e n_y}{(\vec{v} - \vec{u}_{\Omega}) \cdot \vec{n}_{\infty}} + v_2 \right], \quad (3.96)$$

and

$$\psi_{\rho v_3} = -\psi_{\rho E} \left[\frac{\gamma e n_z}{(\vec{v} - \vec{u}_{\Omega}) \cdot \vec{n}_{\infty}} + v_3 \right], \quad (3.97)$$

where $e = E - \frac{\vec{v}^2}{2}$ is the internal energy per unit mass. These conditions are equivalent to those of Hayashi et al. at an outlet if they are reduced to two dimensions in the absence of domain motion ($\vec{u}_{\Omega} = 0$).

The above steps can be repeated as a systematic procedure for finding boundary conditions when other types of physical conditions are imposed in the direct problem (such as prescribing stagnation conditions at an inlet). One must simply introduce the perturbations of the chosen quantities along with the appropriate transformation matrix from the conservative variables into Eqn. 4.61 and solve for the admissible conditions in the same manner. This procedure could be particularly useful for internal flows that are sensitive to inlet/outlet boundary conditions (often solved on truncated domains) or for flow control applications, such as jets, where sensitivities with respect to inlet/outlet quantities might be desired.

Supersonic inlet and outlet boundaries are straightforward in comparison. At a supersonic inlet, all of the flow variables can be prescribed as physical boundary conditions in the direct problem (all characteristics are incoming), which means that none of the adjoint variables will have prescribed values in the adjoint problem. This approach assumes that $\delta U = 0$ due to the direct problem boundary condition, and therefore, Eqn. 4.61 is automatically satisfied. In the case of a supersonic outlet, no flow variables can be specified in the direct problem (all characteristics are outgoing), so all of the adjoint variables can be prescribed with $\Psi = 0$ being the choice that exactly satisfies Eqn. 4.61.

3.5.3 Complete System of Adjoint Equations and Surface Sensitivities

The form of the adjoint system is dependent on the chosen functional, and for viscous flows, the system also changes depending on the type of temperature boundary condition imposed on S . By introducing into Eqn. 3.60 the resulting expressions from the previous sections concerning the evaluation of the boundary integrals (Eqns. 3.66, 3.68, and 3.88) while assuming the proper choice of boundary conditions has removed variations of the flow variables in the far-field, we recover an expanded version of $\delta\mathcal{J}$:

$$\begin{aligned}
\delta\mathcal{J} = & \delta J - \frac{1}{\mathbb{T}} \int_{\Omega} [\Psi^{\top} \delta U]_{t_o}^{t_f} d\Omega - \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \{ \vartheta(\delta \vec{v} \cdot \vec{n}) + [\vec{\varphi} \cdot \vec{n} + \psi_{\rho E}(\vec{v} \cdot \vec{n})] \delta p \} ds dt \\
& + \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S [\vec{\varphi} \cdot \delta \vec{\sigma} \cdot \vec{n} + \psi_{\rho E} \vec{v} \cdot \delta \vec{\sigma} \cdot \vec{n} + \psi_{\rho E} \delta \vec{v} \cdot \vec{\sigma} \cdot \vec{n} + \psi_{\rho E} \mu_{tot}^2 c_p \partial_n(\delta T)] ds dt \\
& - \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S [\vec{n} \cdot (\bar{\Sigma}^{\varphi} + \bar{\Sigma}^{\psi_{\rho E}}) \cdot \delta \vec{v} + \mu_{tot}^2 c_p \partial_n(\psi_{\rho E}) \delta T] ds dt \\
& - \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_{\Omega} \left[-\frac{\partial \Psi^{\top}}{\partial t} - \nabla \Psi^{\top} \cdot (\vec{A}^c - \bar{I}_5 \vec{u}_{\Omega} - \mu_{tot}^k \vec{A}^{vk}) - \nabla \cdot (\nabla \Psi^{\top} \cdot \mu_{tot}^k \bar{D}^{vk}) - \Psi^{\top} \frac{\partial \mathcal{Q}}{\partial U} \right] \delta U d\Omega dt.
\end{aligned} \tag{3.98}$$

Eqn. 3.98 represents the most general form for the variation of the Lagrangian in the context of unsteady, compressible, viscous fluid flow governed by the RANS equations in ALE form under the frozen viscosity assumption.

At this point, a particular objective function must be chosen, so that the appropriate form of δJ can be introduced. There are only two possible options for eliminating all remaining flow perturbations on the surface:

1. A suitable adjoint boundary condition on S is formed to eliminate perturbations.
2. The linearized boundary conditions are used to relate flow perturbations to shape perturbations δS .

Therefore, given a particular problem (objective function and governing equation system with boundary conditions), the adjoint system is defined in order to eliminate any perturbations that can not be related to the shape perturbations. Satisfying this adjoint system with the admissible boundary conditions will then eliminate the remaining flow perturbations. Due to the requirement that all flow perturbations are eliminated, the choice of objective function can not be arbitrary in the continuous adjoint framework. Previous work has addressed this issue [4].

The final expression for the functional variation is an integral over the time interval of interest and S that contains terms involving only the flow and adjoint variables multiplied by δS . These computable formulas are what we call the *surface sensitivity*, and they are the key result of the continuous adjoint derivation. The surface sensitivity provides a measure of the variation of the objective function with respect to infinitesimal variations of the surface shape in the direction of the local surface normal. With each physical time step, this value is computed at every surface node of

the numerical grid with negligible computational cost. In this manner, the functional variation for an arbitrary number of shape perturbations is computable at the fixed cost of solving the flow and adjoint PDE systems.

Within the next several sections, the complete adjoint PDE systems and surface sensitivity formulas are presented for a variety of governing PDE systems and objective functions.

Force-based Functional in Viscous Flow with an Adiabatic Boundary Condition

In this scenario, δJ is given by Eqn. 3.18, and the linearized no-slip (Eqn. 3.36) and heat flux (Eqn. 3.48) boundary conditions are available for use. After introducing these terms into Eqn. 3.98 and rearranging for clarity, we find that the variation of the functional takes the following form:

$$\begin{aligned}
\delta \mathcal{J} = & \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \vec{d} \cdot (\bar{I} \delta p - \delta \bar{\sigma}) \cdot \vec{n} \, ds \, dt - \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S (\bar{\varphi} + \psi_{\rho E} \vec{v}) \cdot (\bar{I} \delta p - \delta \bar{\sigma}) \cdot \vec{n} \, ds \, dt \\
& + \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \left\{ \vec{d} \cdot [\bar{q}_{\rho \vec{v}} - \partial_t(\rho \vec{v})] + \nabla \vec{d} : (\bar{I} p - \bar{\sigma}) - (\bar{I} p - \bar{\sigma}) \cdot \vec{n} \cdot \partial_n \vec{d} \right\} \delta S \, ds \, dt \\
& - \frac{1}{\mathbb{T}} \int_{\Omega} [\Psi^T \delta U]_{t_o}^{t_f} \, d\Omega + \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \vartheta \partial_n (\vec{v} - \vec{u}_{\Omega}) \delta S \cdot \vec{n} \, ds \, dt \\
& + \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \left\{ -\psi_{\rho E} \partial_n (\vec{v} - \vec{u}_{\Omega}) \delta S \cdot \bar{\sigma} \cdot \vec{n} + \psi_{\rho E} \mu_{tot}^2 c_p [\nabla T \cdot \nabla_S (\delta S) - \partial_n^2(T)] \delta S \right\} \, ds \, dt \\
& - \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \left[-\vec{n} \cdot (\bar{\Sigma}^{\varphi} + \bar{\Sigma}^{\psi_{\rho E}}) \cdot \partial_n (\vec{v} - \vec{u}_{\Omega}) \delta S + \mu_{tot}^2 c_p \partial_n (\psi_{\rho E}) \delta T \right] \, ds \, dt \\
& - \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_{\Omega} \left[-\frac{\partial \Psi^T}{\partial t} - \nabla \Psi^T \cdot (\bar{A}^c - \bar{I}_5 \vec{u}_{\Omega} - \mu_{tot}^k \bar{A}^{vk}) - \nabla \cdot (\nabla \Psi^T \cdot \mu_{tot}^k \bar{D}^{vk}) - \Psi^T \frac{\partial \mathcal{Q}}{\partial U} \right] \delta U \, d\Omega \, dt.
\end{aligned} \tag{3.99}$$

The form of the adjoint equations along with the admissible adjoint boundary conditions has become clear, and many of the terms on the right-hand side of Eqn. 3.99 can be eliminated by satisfying the adjoint system:

$$\begin{cases} -\frac{\partial \Psi^T}{\partial t} - \nabla \Psi^T \cdot (\bar{A}^c - \bar{I}_5 \vec{u}_{\Omega} - \mu_{tot}^k \bar{A}^{vk}) - \nabla \cdot (\nabla \Psi^T \cdot \mu_{tot}^k \bar{D}^{vk}) - \Psi^T \frac{\partial \mathcal{Q}}{\partial U} = 0 & \text{in } \Omega \quad t > 0 \\ \bar{\varphi} = \vec{d} - \psi_{\rho E} \vec{v} & \text{on } S \\ \partial_n (\psi_{\rho E}) = 0 & \text{on } S, \end{cases} \tag{3.100}$$

Note that a sign change has occurred for the terms involving the time derivative and the convective flux due to the integration by parts procedure. As a result, the sign of the characteristic velocities is flipped in the adjoint problem, which causes characteristic information to propagate in the reverse direction.

The remaining terms can be gathered as

$$\begin{aligned}
\delta\mathcal{J} &= \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \left\{ \vec{d} \cdot [\vec{q}_{\rho\vec{v}} - \partial_t(\rho\vec{v})] + \nabla\vec{d} : (\bar{I}p - \bar{\sigma}) - (\bar{I}p - \bar{\sigma}) \cdot \vec{n} \cdot \partial_n\vec{d} \right\} \delta S \, ds \, dt \\
&+ \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \left[\vartheta \partial_n(\vec{v} - \vec{u}_\Omega) \cdot \vec{n} - \psi_{\rho E} \partial_n(\vec{v} - \vec{u}_\Omega) \cdot \bar{\sigma} \cdot \vec{n} + \vec{n} \cdot \left(\bar{\Sigma}^\varphi + \bar{\Sigma}^{\psi_{\rho E}} \right) \cdot \partial_n(\vec{v} - \vec{u}_\Omega) \right] \delta S \, ds \, dt \\
&+ \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \psi_{\rho E} \mu_{tot}^2 c_p \left[\nabla T \cdot \nabla_S(\delta S) - \partial_n^2(T) \delta S \right] \, ds \, dt. \tag{3.101}
\end{aligned}$$

One final simplification can be made in order to avoid the need to compute higher order derivatives of the temperature. Starting from the term in the final integral of Eqn. 3.101,

$$\begin{aligned}
&\psi_{\rho E} \mu_{tot}^2 c_p \left[\nabla T \cdot \nabla_S(\delta S) - \partial_n^2(T) \delta S \right] \\
&= \psi_{\rho E} \mu_{tot}^2 c_p \nabla T \cdot \nabla_S(\delta S) - \psi_{\rho E} \left[\nabla \cdot (\mu_{tot}^2 c_p \nabla T) - \nabla_S \cdot (\mu_{tot}^2 c_p \nabla_S T) \right] \delta S \\
&= \psi_{\rho E} \mu_{tot}^2 c_p \nabla T \cdot \nabla_S(\delta S) + \psi_{\rho E} \delta S \nabla_S \cdot (\mu_{tot}^2 c_p \nabla_S T) - \psi_{\rho E} \nabla \cdot (\mu_{tot}^2 c_p \nabla T) \delta S, \tag{3.102}
\end{aligned}$$

and the second term on the right hand side of Eqn. 3.102 can be integrated by parts as

$$\begin{aligned}
&\nabla_S \cdot [\psi_{\rho E} \delta S \mu_{tot}^2 c_p \nabla_S T] = \nabla_S(\psi_{\rho E} \delta S) \cdot (\mu_{tot}^2 c_p \nabla_S T) + \psi_{\rho E} \delta S \nabla_S \cdot (\mu_{tot}^2 c_p \nabla_S T) \\
&\int_{t_o}^{t_f} \int_S \nabla_S \cdot [\psi_{\rho E} \delta S \mu_{tot}^2 c_p \nabla_S T] \, ds \, dt = \int_{t_o}^{t_f} \int_S \nabla_S(\psi_{\rho E} \delta S) \cdot (\mu_{tot}^2 c_p \nabla_S T) \, ds \, dt \\
&\quad + \int_{t_o}^{t_f} \int_S \psi_{\rho E} \delta S \nabla_S \cdot (\mu_{tot}^2 c_p \nabla_S T) \, ds \, dt \\
&\int_{t_o}^{t_f} \int_S \psi_{\rho E} \delta S \nabla_S \cdot (\mu_{tot}^2 c_p \nabla_S T) \, ds \, dt = \int_{t_o}^{t_f} \int_S -\nabla_S(\psi_{\rho E} \delta S) \cdot (\mu_{tot}^2 c_p \nabla_S T) \, ds \, dt, \tag{3.103}
\end{aligned}$$

where we have formed the product rule, integrated in space and time in going from the first to second lines, and changed the order of integration, used the identity $\int_S \nabla_S \cdot (\cdot) \, ds = 0$ on a closed surface [114], and rearranged in going to the third line. Using the result in Eqn. 3.103 in Eqn. 3.102 gives,

$$\begin{aligned}
&\psi_{\rho E} \mu_{tot}^2 c_p \left[\nabla T \cdot \nabla_S(\delta S) - \partial_n^2(T) \delta S \right] \\
&= \psi_{\rho E} \mu_{tot}^2 c_p \nabla T \cdot \nabla_S(\delta S) - \nabla_S(\psi_{\rho E} \delta S) \cdot (\mu_{tot}^2 c_p \nabla_S T) - \psi_{\rho E} \nabla \cdot (\mu_{tot}^2 c_p \nabla T) \delta S \\
&= \mu_{tot}^2 c_p \nabla_S T \cdot [\psi_{\rho E} \cdot \nabla_S(\delta S) - \nabla_S(\psi_{\rho E} \delta S)] - \psi_{\rho E} \nabla \cdot (\mu_{tot}^2 c_p \nabla T) \delta S \\
&= -\mu_{tot}^2 c_p \nabla_S T \cdot \nabla_S(\psi_{\rho E} \delta S) - \psi_{\rho E} \nabla \cdot (\mu_{tot}^2 c_p \nabla T) \delta S, \tag{3.104}
\end{aligned}$$

and by expressing the energy equation on the surface (i.e., imposing the no-slip and adiabatic

boundary conditions) and substituting the result into Eqn. 3.104, we find that

$$\begin{aligned} & \psi_{\rho E} \mu_{tot}^2 c_p [\nabla T \cdot \nabla_S(\delta S) - \partial_n^2(T) \delta S] \\ &= -\mu_{tot}^2 c_p \nabla_S(\psi_{\rho E}) \cdot \nabla_S(T) - \psi_{\rho E} [p(\nabla \cdot \vec{v}) - \bar{\sigma} : \nabla \vec{v} + \partial_t(\rho E) + (\vec{q}_{\rho \vec{v}} - \partial_t(\rho \vec{v})) \cdot \vec{v} - q_{\rho E}]. \end{aligned} \quad (3.105)$$

Therefore, the final expression for the variation of the Lagrangian becomes

$$\begin{aligned} \delta \mathcal{J} &= \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \{ \vec{d} \cdot [\vec{q}_{\rho \vec{v}} - \partial_t(\rho \vec{v})] + \nabla \vec{d} : (\bar{I} \bar{p} - \bar{\sigma}) - (\bar{I} \bar{p} - \bar{\sigma}) \cdot \vec{n} \cdot \partial_n \vec{d} + \vartheta \partial_n(\vec{v} - \vec{u}_\Omega) \cdot \vec{n} \\ &\quad - \psi_{\rho E} \partial_n(\vec{v} - \vec{u}_\Omega) \cdot \bar{\sigma} \cdot \vec{n} + \vec{n} \cdot (\bar{\Sigma}^\varphi + \bar{\Sigma}^{\psi_{\rho E}}) \cdot \partial_n(\vec{v} - \vec{u}_\Omega) - \mu_{tot}^2 c_p \nabla_S(\psi_{\rho E}) \cdot \nabla_S(T) \\ &\quad - \psi_{\rho E} [p(\nabla \cdot \vec{v}) - \bar{\sigma} : \nabla \vec{v} + \partial_t(\rho E) + (\vec{q}_{\rho \vec{v}} - \partial_t(\rho \vec{v})) \cdot \vec{v} - q_{\rho E}] \} \delta S \, ds \, dt \\ &= \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \left\{ \frac{\partial \mathcal{J}}{\partial S} \right\} \delta S \, ds \, dt, \end{aligned} \quad (3.106)$$

where all of the terms composing $\frac{\partial \mathcal{J}}{\partial S}$ form the *surface sensitivity*. Note that the final expression for the variation involves only a surface integral in space and has no dependence on the volume mesh. Furthermore, several new terms appear that directly involve time derivatives, source terms, or the arbitrary motion of the surface. By studying the terms in the expression for surface sensitivity, deeper physical insight and designer intuition can be gained. Visualizing the sensitivities as a map on S can also help define design variables or geometry parameterizations by highlighting the regions of the geometry that are most sensitive for a given objective function.

For a steady problem with a fixed surface ($\vec{v} = 0$ on S), a constant force projection vector \vec{d} , and no source terms, this expression reduces to that found previously using differential geometry [14]:

$$\delta \mathcal{J} = \int_S \left\{ \vec{n} \cdot \bar{\Sigma}^\varphi \cdot \partial_n(\vec{v}) - \mu_{tot}^2 c_p \nabla_S(\psi_{\rho E}) \cdot \nabla_S(T) \right\} \delta S \, ds. \quad (3.107)$$

Furthermore, it is important to check the individual contributions from the terms in Eqn. 3.106 and to simplify the expression if possible. When certain terms are known to evaluate to zero analytically, it is often the case that they are neglected in order to ensure accuracy in the calculated gradients, as some terms can contaminate the sensitivity on the surface if the solution is not properly converged or if the quality of the mesh is poor. The expression in Eqn. 3.106 is a very general result, and for the results in this dissertation with a constant force projection vector, the terms involving derivatives of \vec{d} in the surface sensitivity do not appear.

Force-based Functional in Viscous Flow with an Isothermal Boundary Condition

In this scenario, δJ is given by Eqn. 3.18, and the linearized no-slip (Eqn. 3.36) and isothermal (Eqn. 3.51) boundary conditions are available for use. Introducing these terms into Eqn. 3.99 and

rearranging for clarity yields

$$\begin{aligned}
\delta\mathcal{J} &= \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \vec{d} \cdot (\bar{I}\delta p - \delta\bar{\sigma}) \cdot \vec{n} \, ds \, dt - \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S (\vec{\varphi} + \psi_{\rho E} \vec{v}) \cdot (\bar{I}\delta p - \delta\bar{\sigma}) \cdot \vec{n} \, ds \, dt \\
&+ \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \left\{ \vec{d} \cdot [\vec{q}_{\rho\vec{v}} - \partial_t(\rho\vec{v})] + \nabla\vec{d} : (\bar{I}p - \bar{\sigma}) - (\bar{I}p - \bar{\sigma}) \cdot \vec{n} \cdot \partial_n \vec{d} \right\} \delta S \, ds \, dt \\
&- \frac{1}{\mathbb{T}} \int_{\Omega} [\Psi^T \delta U]_{t_o}^{t_f} \, d\Omega + \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \vartheta \partial_n (\vec{v} - \vec{u}_{\Omega}) \delta S \cdot \vec{n} \, ds \, dt \\
&+ \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \left\{ -\psi_{\rho E} \partial_n (\vec{v} - \vec{u}_{\Omega}) \delta S \cdot \bar{\sigma} \cdot \vec{n} + \psi_{\rho E} \mu_{tot}^2 c_p \partial_n (\delta T) \right\} \, ds \, dt \\
&- \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \left[-\vec{n} \cdot (\bar{\Sigma}^{\varphi} + \bar{\Sigma}^{\psi_{\rho E}}) \cdot \partial_n (\vec{v} - \vec{u}_{\Omega}) \delta S - \mu_{tot}^2 c_p \partial_n (\psi_{\rho E}) \partial_n (T) \delta S \right] \, ds \, dt \\
&- \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_{\Omega} \left[-\frac{\partial \Psi^T}{\partial t} - \nabla \Psi^T \cdot (\vec{A}^c - \bar{I}_5 \vec{u}_{\Omega} - \mu_{tot}^k \vec{A}^{vk}) - \nabla \cdot (\nabla \Psi^T \cdot \mu_{tot}^k \vec{D}^{vk}) - \Psi^T \frac{\partial \mathcal{Q}}{\partial U} \right] \delta U \, d\Omega \, dt.
\end{aligned} \tag{3.108}$$

The form of the adjoint equations along with the admissible adjoint boundary conditions is again clear, and many of the terms on the right hand side of Eqn. 3.108 can be eliminated by satisfying the following adjoint system:

$$\begin{cases} -\frac{\partial \Psi^T}{\partial t} - \nabla \Psi^T \cdot (\vec{A}^c - \bar{I}_5 \vec{u}_{\Omega} - \mu_{tot}^k \vec{A}^{vk}) - \nabla \cdot (\nabla \Psi^T \cdot \mu_{tot}^k \vec{D}^{vk}) - \Psi^T \frac{\partial \mathcal{Q}}{\partial U} = 0 & \text{in } \Omega \quad t > 0 \\ \vec{\varphi} = \vec{d} - \psi_{\rho E} \vec{v} & \text{on } S \\ \psi_{\rho E} = 0 & \text{on } S, \end{cases} \tag{3.109}$$

A small change in the boundary condition for the adjoint energy on the surface has occurred due to the change from a heat flux to an isothermal boundary condition in the flow problem, as the adjoint system is required to eliminate a different term than in the adiabatic case. The linearized boundary condition for temperature is also slightly different, resulting in a change to the surface sensitivity when the remaining terms are gathered:

$$\begin{aligned}
\delta\mathcal{J} &= \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \left\{ \vec{d} \cdot [\vec{q}_{\rho\vec{v}} - \partial_t(\rho\vec{v})] + \nabla\vec{d} : (\bar{I}p - \bar{\sigma}) - (\bar{I}p - \bar{\sigma}) \cdot \vec{n} \cdot \partial_n \vec{d} + \vartheta \partial_n (\vec{v} - \vec{u}_{\Omega}) \cdot \vec{n} \right. \\
&\quad \left. - \psi_{\rho E} \partial_n (\vec{v} - \vec{u}_{\Omega}) \cdot \bar{\sigma} \cdot \vec{n} + \vec{n} \cdot (\bar{\Sigma}^{\varphi} + \bar{\Sigma}^{\psi_{\rho E}}) \cdot \partial_n (\vec{v} - \vec{u}_{\Omega}) - \mu_{tot}^2 c_p \partial_n (\psi_{\rho E}) \partial_n (T) \right\} \delta S \, ds \, dt, \\
&= \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \left\{ \frac{\partial \mathcal{J}}{\partial S} \right\} \delta S \, ds \, dt,
\end{aligned} \tag{3.110}$$

where all of the terms composing $\frac{\partial \mathcal{J}}{\partial S}$ form the *surface sensitivity*.

Force-based Functional in Inviscid Flow

In inviscid flow, the general form for the variation of the Lagrangian can first be greatly simplified before introducing the variation of the functional and linearized boundary conditions:

$$\begin{aligned} \delta\mathcal{J} = \delta J - \frac{1}{\mathbb{T}} \int_{\Omega} [\Psi^{\top} \delta U]_{t_o}^{t_f} d\Omega - \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \{ \vartheta(\delta \vec{v} \cdot \vec{n}) + [\vec{\varphi} \cdot \vec{n} + \psi_{\rho E}(\vec{v} \cdot \vec{n})] \delta p \} ds dt \\ - \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_{\Omega} \left[-\frac{\partial \Psi^{\top}}{\partial t} - \nabla \Psi^{\top} \cdot (\vec{A}^c - \bar{I}_5 \vec{u}_{\Omega}) - \Psi^{\top} \frac{\partial \mathcal{Q}}{\partial U} \right] \delta U d\Omega dt. \end{aligned} \quad (3.111)$$

Now, for a force-based functional in inviscid flow, δJ is given by Eqn. 3.22, and the linearized flow tangency boundary condition from Eqn. 3.42 is available for use. Introducing these relationships into Eqn 3.111 gives

$$\begin{aligned} \delta\mathcal{J} = \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \vec{d} \cdot \bar{I} \delta p \cdot \vec{n} ds dt - \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S (\vec{\varphi} + \psi_{\rho E} \vec{v}) \cdot \bar{I} \delta p \cdot \vec{n} ds dt \\ + \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S (\vec{d} \cdot \nabla p + p \nabla \cdot \vec{d} - \bar{I} p \cdot \vec{n} \cdot \partial_n \vec{d}) \delta S ds dt \\ - \frac{1}{\mathbb{T}} \int_{\Omega} [\Psi^{\top} \delta U]_{t_o}^{t_f} d\Omega - \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \{ \vartheta[(\vec{v} - \vec{u}_{\Omega}) \cdot \nabla_S(\delta S) - \partial_n(\vec{v} - \vec{u}_{\Omega}) \delta S \cdot \vec{n}] \} ds dt \\ - \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_{\Omega} \left[-\frac{\partial \Psi^{\top}}{\partial t} - \nabla \Psi^{\top} \cdot (\vec{A}^c - \bar{I}_5 \vec{u}_{\Omega}) - \Psi^{\top} \frac{\partial \mathcal{Q}}{\partial U} \right] \delta U d\Omega dt. \end{aligned} \quad (3.112)$$

Many of the terms on the right hand side of Eqn. 3.112 can be eliminated, including the variation of pressure δp , by satisfying the following adjoint system:

$$\begin{cases} -\frac{\partial \Psi^{\top}}{\partial t} - \nabla \Psi^{\top} \cdot (\vec{A}^c - \bar{I}_5 \vec{u}_{\Omega}) - \Psi^{\top} \frac{\partial \mathcal{Q}}{\partial U} = 0 & \text{in } \Omega \quad t > 0 \\ \vec{\varphi} \cdot \vec{n} = (\vec{d} - \psi_{\rho E} \vec{v}) \cdot \vec{n} & \text{on } S. \end{cases} \quad (3.113)$$

The remaining terms that involve the perturbations in the surface δS can be gathered as

$$\begin{aligned} \delta\mathcal{J} = \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S (\vec{d} \cdot \nabla p + p \nabla \cdot \vec{d} - \bar{I} p \cdot \vec{n} \cdot \partial_n \vec{d}) \delta S ds dt \\ - \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \{ \vartheta[(\vec{v} - \vec{u}_{\Omega}) \cdot \nabla_S(\delta S) - \partial_n(\vec{v} - \vec{u}_{\Omega}) \delta S \cdot \vec{n}] \} ds dt. \end{aligned} \quad (3.114)$$

A final simplification can be made by integrating the term involving $\nabla_S(\delta S)$ in Eqn. 3.114 by parts

as

$$\begin{aligned}
\nabla_S \cdot [\vartheta(\vec{v} - \vec{u}_\Omega)\delta S] &= \nabla_S \cdot [\vartheta(\vec{v} - \vec{u}_\Omega)] \delta S + \vartheta(\vec{v} - \vec{u}_\Omega) \cdot \nabla_S(\delta S) \\
\int_{t_o}^{t_f} \int_S \nabla_S \cdot [\vartheta(\vec{v} - \vec{u}_\Omega)\delta S] ds dt &= \int_{t_o}^{t_f} \int_S \nabla_S \cdot [\vartheta(\vec{v} - \vec{u}_\Omega)] \delta S ds dt \\
&\quad + \int_{t_o}^{t_f} \int_S \vartheta(\vec{v} - \vec{u}_\Omega) \cdot \nabla_S(\delta S) ds dt \\
\int_{t_o}^{t_f} \int_S \vartheta(\vec{v} - \vec{u}_\Omega) \cdot \nabla_S(\delta S) ds dt &= \int_{t_o}^{t_f} \int_S -\nabla_S \cdot [\vartheta(\vec{v} - \vec{u}_\Omega)] \delta S ds dt, \tag{3.115}
\end{aligned}$$

where we have used the identity $\int_S \nabla_S \cdot (\cdot) ds = 0$ on a closed surface [114]. Substituting the result into Eqn. 3.114, while recalling that the divergence in local coordinates can be expressed as $\nabla \cdot \vec{a} = \nabla_S \cdot \vec{a} + \partial_n(\vec{a} \cdot \vec{n}) - 2H_m(\vec{a} \cdot \vec{n})$ with $\vec{a} = \vartheta(\vec{v} - \vec{u}_\Omega)$ in this case, one has

$$\begin{aligned}
\delta \mathcal{J} &= \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \left(\vec{d} \cdot \nabla p + p \nabla \cdot \vec{d} - \bar{I}p \cdot \vec{n} \cdot \partial_n \vec{d} + \nabla_S \cdot [\vartheta(\vec{v} - \vec{u}_\Omega)] + \vartheta \partial_n(\vec{v} - \vec{u}_\Omega) \cdot \vec{n} \right) \delta S ds dt \\
&= \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \left\{ \vec{d} \cdot \nabla p + p \nabla \cdot \vec{d} - \bar{I}p \cdot \vec{n} \cdot \partial_n \vec{d} + \vartheta \partial_n(\vec{v} - \vec{u}_\Omega) \cdot \vec{n} \right. \\
&\quad \left. + \nabla \cdot [\vartheta(\vec{v} - \vec{u}_\Omega)] - \partial_n[\vartheta(\vec{v} - \vec{u}_\Omega) \cdot \vec{n}] + 2H_m \vartheta(\vec{v} - \vec{u}_\Omega) \cdot \vec{n} \right\} \delta S ds dt \\
&= \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \left\{ \vec{d} \cdot \nabla p + p \nabla \cdot \vec{d} - \bar{I}p \cdot \vec{n} \cdot \partial_n \vec{d} + \vartheta \partial_n(\vec{v} - \vec{u}_\Omega) \cdot \vec{n} \right. \\
&\quad \left. + \nabla \cdot [\vartheta(\vec{v} - \vec{u}_\Omega)] - \partial_n(\vartheta)(\vec{v} - \vec{u}_\Omega) \cdot \vec{n} - \vartheta \partial_n(\vec{v} - \vec{u}_\Omega) \cdot \vec{n} + 2H_m \vartheta(\vec{v} - \vec{u}_\Omega) \cdot \vec{n} \right\} \delta S ds dt \\
&= \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \left\{ \vec{d} \cdot \nabla p + p \nabla \cdot \vec{d} - \bar{I}p \cdot \vec{n} \cdot \partial_n \vec{d} + \nabla \cdot [\vartheta(\vec{v} - \vec{u}_\Omega)] \right\} \delta S ds dt \\
&= \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \left\{ \vec{d} \cdot \nabla p + p \nabla \cdot \vec{d} - \bar{I}p \cdot \vec{n} \cdot \partial_n \vec{d} + \nabla(\vartheta) \cdot (\vec{v} - \vec{u}_\Omega) + \vartheta \nabla \cdot (\vec{v} - \vec{u}_\Omega) \right\} \delta S ds dt \\
&= \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \left\{ \frac{\partial \mathcal{J}}{\partial S} \right\} \delta S ds dt, \tag{3.116}
\end{aligned}$$

where all of the terms composing $\frac{\partial \mathcal{J}}{\partial S}$ form the *surface sensitivity*.

Heat Flux Functional

For designing the heat flux on the surface, δJ is given by Eqn. 3.27, and the linearized no-slip (Eqn. 3.36) and isothermal (Eqn. 3.51) boundary conditions are available for use. Introducing these

terms into Eqn. 3.99 and rearranging for clarity gives

$$\begin{aligned}
\delta \mathcal{J} &= \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S c \partial_n(\delta T) ds dt + \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \psi_{\rho E} \mu_{tot}^2 c_p \partial_n(\delta T) ds dt + \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \nabla \cdot (c \nabla T) \delta S ds dt \\
&\quad - \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S (\vec{\varphi} + \psi_{\rho E} \vec{v}) \cdot (\bar{I} \delta p - \delta \bar{\sigma}) \cdot \vec{n} ds dt - \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \psi_{\rho E} \partial_n(\vec{v} - \vec{u}_\Omega) \delta S \cdot \bar{\sigma} \cdot \vec{n} ds dt \\
&\quad - \frac{1}{\mathbb{T}} \int_\Omega [\Psi^\top \delta U]_{t_o}^{t_f} d\Omega + \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \vartheta \partial_n(\vec{v} - \vec{u}_\Omega) \delta S \cdot \vec{n} ds dt \\
&\quad - \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \left[-\vec{n} \cdot (\bar{\Sigma}^\varphi + \bar{\Sigma}^{\psi_{\rho E}}) \cdot \partial_n(\vec{v} - \vec{u}_\Omega) \delta S - \mu_{tot}^2 c_p \partial_n(\psi_{\rho E}) \partial_n(T) \delta S \right] ds dt \\
&\quad - \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_\Omega \left[-\frac{\partial \Psi^\top}{\partial t} - \nabla \Psi^\top \cdot (\vec{A}^c - \bar{I}_5 \vec{u}_\Omega - \mu_{tot}^k \vec{A}^{vk}) - \nabla \cdot (\nabla \Psi^\top \cdot \mu_{tot}^k \bar{D}^{vk}) - \Psi^\top \frac{\partial \mathcal{Q}}{\partial U} \right] \delta U d\Omega dt.
\end{aligned} \tag{3.117}$$

We can eliminate many of the terms on the right-hand side of Eqn. 3.117 by satisfying the corresponding adjoint system:

$$\begin{cases} -\frac{\partial \Psi^\top}{\partial t} - \nabla \Psi^\top \cdot (\vec{A}^c - \bar{I}_5 \vec{u}_\Omega - \mu_{tot}^k \vec{A}^{vk}) - \nabla \cdot (\nabla \Psi^\top \cdot \mu_{tot}^k \bar{D}^{vk}) - \Psi^\top \frac{\partial \mathcal{Q}}{\partial U} = 0 & \text{in } \Omega \quad t > 0 \\ \vec{\varphi} = -\psi_{\rho E} \vec{v} & \text{on } S \\ \psi_{\rho E} = -\frac{c}{\mu_{tot}^2 c_p} & \text{on } S, \end{cases} \tag{3.118}$$

The adjoint boundary conditions on the surface have again changed, as the adjoint system has adapted in order to remove perturbations given the heat flux functional. In the case of heat flux design, it is required that an isothermal boundary condition be applied in the direct problem. Otherwise, a perturbation in temperature δT would remain, as the linearized isothermal boundary condition would not be available to remove it. The final form of the variation can then be written as

$$\begin{aligned}
\delta \mathcal{J} &= \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \left\{ \vartheta \partial_n(\vec{v} - \vec{u}_\Omega) \cdot \vec{n} - \psi_{\rho E} \partial_n(\vec{v} - \vec{u}_\Omega) \cdot \bar{\sigma} \cdot \vec{n} + \vec{n} \cdot (\bar{\Sigma}^\varphi + \bar{\Sigma}^{\psi_{\rho E}}) \cdot \partial_n(\vec{v} - \vec{u}_\Omega) \right. \\
&\quad \left. + \mu_{tot}^2 c_p \partial_n(\psi_{\rho E}) \partial_n(T) + \nabla \cdot (c \nabla T) \right\} \delta S ds dt \\
&= \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \left\{ \frac{\partial \mathcal{J}}{\partial S} \right\} \delta S ds dt,
\end{aligned} \tag{3.119}$$

where all of the terms composing $\frac{\partial \mathcal{J}}{\partial S}$ form the *surface sensitivity*.

Chapter 4

Numerical Implementation

The following sections contain numerical implementation strategies for each of the major components needed for performing optimal shape design in unsteady flows. The complete design loop requires PDE analysis with dynamic meshes for computing both the objective function and sensitivities, the definition of suitable design variables for parameterizing the geometry, a mesh deformation algorithm for perturbing the numerical grid after shape changes, and a gradient-based optimizer to drive the design variables toward an optimum for the chosen cost function and constraints.

All of the above mentioned components were implemented within the open-source SU2 software suite [98, 100]. This collection of C++ codes is built specifically for PDE analysis and PDE-constrained optimization on unstructured meshes, and it is particularly well-suited for aerodynamic shape design. Modules for performing flow and adjoint solutions, acquiring gradient information by projecting surface sensitivities into the design space, and mesh deformation techniques are included in the suite, amongst others.

An overview of practical implementation details for the numerical methods is given in this chapter. Both the flow and adjoint problems are solved numerically on unstructured meshes with an edge-based data structure. The unstructured grid technology plays an important role in the design of complex, realistic geometries, as the nodes and elements in the corresponding volume mesh surrounding the geometry are not constrained to a particular topology. This added flexibility makes it much easier to generate grids for complex geometry (and even makes automatic mesh refinement and adaptation simpler), but it comes at the cost of some computational efficiency, as the relationship between adjacent cells is no longer implicit and the connectivity information must be stored in additional data structures.

Following the method of lines, we discretize the governing equations in space and time separately. This decoupling of space and time allows for the selection of different types of schemes for the spatial and temporal integration. In general, spatial integration is performed using the finite volume method (FVM), while integration in time is achieved through several available explicit and implicit methods. For time-accurate calculations, a dual time-stepping approach is used.

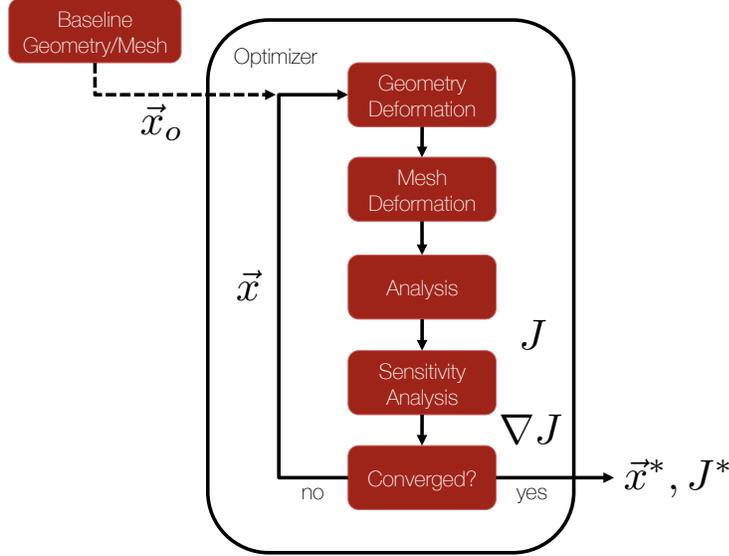


Figure 4.1: Flow chart for a typical shape optimization problem. J is the objective function, and \vec{x} is the vector of design variables. J^* and \vec{x}^* represent an optimum.

4.1 Spatial Integration via the Finite Volume Method

Both the governing flow and adjoint PDEs are spatially discretized on unstructured meshes via the FVM [7, 48, 127, 106, 56, 70, 130, 53, 54, 122] using a median-dual, vertex-based scheme with a standard edge-based structure. Instances of the state vector (U or Ψ) are stored at the nodes of the primal mesh, and the dual mesh is constructed by connecting the primal cell centroids, face centroids, and edge midpoints surrounding a particular node, as shown in Fig. 4.2.

Consider the following general form for a PDE that contains the physics in which we are interested:

$$\frac{\partial U}{\partial t} + \nabla \cdot \vec{F}^c - \nabla \cdot \vec{F}^v - Q = 0, \quad (4.1)$$

where U is the vector of state variables, \vec{F}^c represents convective fluxes, \vec{F}^v represents viscous fluxes, and Q is an arbitrary source term. Applying the FVM, we integrate the differential form of the equations over a control volume Ω_i surrounding node i , or

$$\int_{\Omega_i(t)} \left(\frac{\partial U}{\partial t} + \nabla \cdot \vec{F}^c - \nabla \cdot \vec{F}^v - Q \right) d\Omega = 0, \quad (4.2)$$

which, after application of the Divergence theorem, becomes

$$\int_{\Omega_i(t)} \frac{\partial U}{\partial t} d\Omega + \int_{\partial\Omega_i(t)} (\vec{F}^c - \vec{F}^v) \cdot \vec{n} ds - \int_{\Omega_i} \mathcal{Q} d\Omega = 0, \quad (4.3)$$

where the convective and viscous flux terms have been transformed from domain integrals to integrals over the boundaries of the domain. Note that the geometry of the control volumes may change with time, as denoted by $\Omega_i(t)$, and this will be required for dynamically deforming meshes. After discretizing the convective and viscous fluxes with suitable methods (to be discussed below), moving the time derivative out of the first integral, and assuming a piecewise constant source term results in a semi-discretized form of the PDE, one has

$$\frac{d}{dt} \int_{\Omega_i(t)} U d\Omega + \sum_{j \in \mathcal{N}(i)} (\tilde{F}_{ij}^c - \tilde{F}_{ij}^v) \Delta S_{ij} - |\Omega_i| Q_i = 0, \quad (4.4)$$

where \tilde{F}_{ij}^c and \tilde{F}_{ij}^v are the projected numerical approximations of the convective and viscous fluxes along an edge ij (evaluated at the midpoint), respectively, ΔS_{ij} is the area of the face associated with the edge ij , $|\Omega_i|$ is the volume of the current cell in the dual mesh, and $\mathcal{N}(i)$ is the set of neighboring nodes to node i . Fig. 4.2 contains graphical descriptions of the geometric terms. Each of the terms resulting from the spatial discretization are often summed together into a single numerical residual $R_i(U)$ at each node and written concisely as

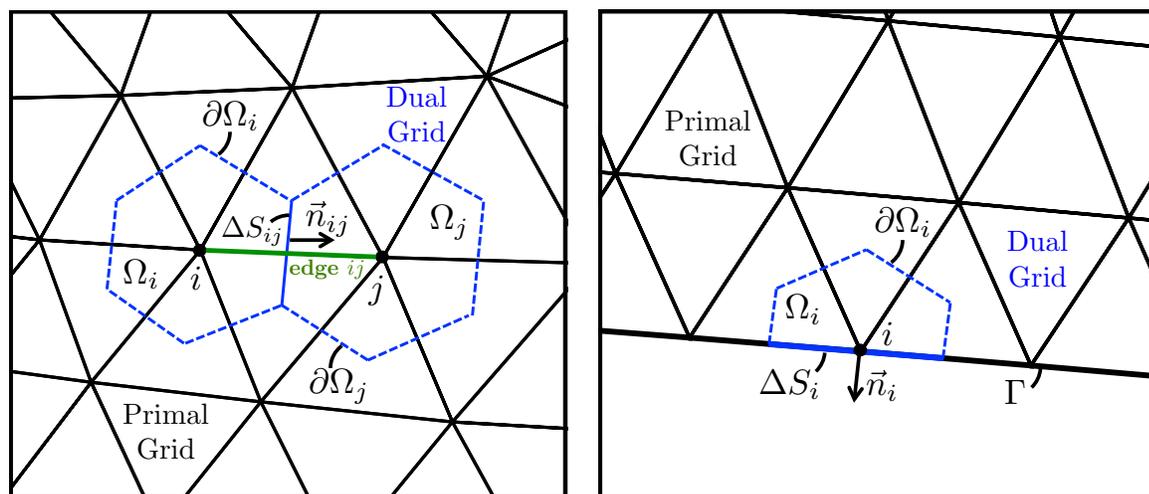
$$\frac{d}{dt} \int_{\Omega_i(t)} U d\Omega + R_i(U) = 0. \quad (4.5)$$

Eqn. 4.5 represents a system of coupled, ordinary differential equations for each node that can be marched forward in time with an appropriate time integration method.

The FVM with a median-dual, cell-vertex scheme, as described above, is amenable to an edge-based data structure, meaning that mesh data (nodal coordinates, face areas, local normals, etc.) are stored on an edge-by-edge basis. The edge-based structure offers efficiency in terms of memory usage and also computation, as a single loop over the edges in the mesh allows for the numerical fluxes for each node to be computed. In practice, the numerical residual is evaluated using a sequence of loops over the edges and nodes:

- a loop over all of the edges in the primal mesh in order to calculate the convective and viscous fluxes,
- a loop over all of the nodes in the primal mesh and compute source terms in each dual control volume given the current state, and
- a loop over all of the boundary nodes in the primal mesh in order to impose boundary conditions.

Additional edge loops may also appear for computing other necessary quantities. This series of steps results in a value of $R_i(U)$ at each node at a single instance in time, which can then be substituted



(a) Dual mesh control volumes surrounding two nodes (i and j) in the domain interior.

(b) Dual mesh control volume surrounding a boundary node i .

Figure 4.2: Schematics showing the geometry of the primal and dual meshes.

into Eqn. 4.5 and integrated forward in time to either arrive at a steady state or a time-accurate solution for the state vector U . In the following sections, the key algorithms for computing $R_i(U)$ are detailed for both the flow and adjoint equations, including convective and viscous flux schemes, source term integration, and boundary conditions.

4.1.1 Discretizing the Governing Flow Equations

The spatial discretization of the compressible Navier-Stokes equations will be discussed first, and it can often be best understood by considering the mathematical character of the governing PDEs. The characteristic behavior describes how solution information travels within the domain and will dictate stability limits and the allowable conditions that can be prescribed at the domain boundaries. Understanding characteristic behavior can aid in developing stable numerical methods for the spatial integration of fluxes (upwind schemes, in particular) and also accurate strategies for the numerical implementation of boundary conditions.

Studying the hyperbolic nature of the Euler equations enables the analysis of the characteristic behavior. As most practical flows of interest in aeronautics are at a relatively high Reynolds number and thus convection dominated, methods that observe the characteristic behavior of the Euler equations often form the basis for many numerical methods in viscous flow with the addition of centrally-differenced viscous fluxes.

By analyzing the quasi-linear form of the equations, one can diagonalize the convective flux Jacobian in order to find the eigenvalues that represent the characteristic speeds. In a multi-dimensional flow, these characteristic speeds are typically projected into the normal direction at control volume faces and domain boundaries, so that the flow can be treated as locally one-dimensional. In

three dimensions on a stationary mesh, the diagonal matrix containing the characteristic speeds at a boundary is given by,

$$\Lambda = \begin{pmatrix} \lambda_1 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \lambda_2 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \lambda_3 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \lambda_4 & \cdot \\ \cdot & \cdot & \cdot & \cdot & \lambda_5 \end{pmatrix} = \begin{pmatrix} \vec{v} \cdot \vec{n} & \cdot & \cdot & \cdot & \cdot \\ \cdot & \vec{v} \cdot \vec{n} & \cdot & \cdot & \cdot \\ \cdot & \cdot & \vec{v} \cdot \vec{n} & \cdot & \cdot \\ \cdot & \cdot & \cdot & \vec{v} \cdot \vec{n} + c & \cdot \\ \cdot & \cdot & \cdot & \cdot & \vec{v} \cdot \vec{n} - c \end{pmatrix}, \quad (4.6)$$

where \vec{v} is the flow velocity, \vec{n} is the local normal vector, and c is the speed of sound. The first three eigenvalues are repeated and the final two entries are related to the acoustic waves. When the equations are treated in ALE form, there is a direct consequence for the characteristic speeds of the system. This can be shown by considering the eigenvalues of the flux Jacobian \vec{A}_{ale}^c , which are the solution of the following eigenvalue problem:

$$\begin{aligned} 0 &= \det \left| \vec{A}_{ale}^c \cdot \vec{n} - \lambda \bar{I}_5 \right| \\ &= \det \left| \frac{\partial \vec{F}_{ale}^c}{\partial U} \cdot \vec{n} - \lambda \bar{I}_5 \right| \\ &= \det \left| \frac{\partial}{\partial U} (\vec{F}^c - \vec{u}_\Omega U) \cdot \vec{n} - \lambda \bar{I}_5 \right| \\ &= \det \left| \vec{A}^c \cdot \vec{n} - (\vec{u}_\Omega \cdot \vec{n} + \lambda) \bar{I}_5 \right|, \end{aligned} \quad (4.7)$$

where \vec{u}_Ω is the velocity of the moving domain. Solving Eqn 4.7, one has for the eigenvalues

$$\lambda_1 = \lambda_2 = \lambda_3 = (\vec{v} - \vec{u}_\Omega) \cdot \vec{n}, \quad (4.8)$$

$$\lambda_4 = (\vec{v} - \vec{u}_\Omega) \cdot \vec{n} + c, \quad (4.9)$$

$$\lambda_5 = (\vec{v} - \vec{u}_\Omega) \cdot \vec{n} - c. \quad (4.10)$$

This adjustment to the characteristic speeds involving \vec{u}_Ω will repeatedly appear when composing numerical schemes for solving the governing equations in ALE form: almost all of the numerical methods for computing the convective fluxes, boundary conditions, and allowable time step limits require knowledge of the eigenvalues of the system.

Integration of Convective Fluxes

The convective fluxes can be discretized using one of many central or upwind schemes that have been developed for median-dual schemes on unstructured meshes. This section will focus on two classic numerical schemes that are commonly used for computing convective fluxes: the flux-difference-splitting scheme of Roe [107] and the central scheme with artificial dissipation by Jameson, Schmidt, and Turkel (JST) [62]. In particular, the adjustments required for handling the ALE form of the convective terms will be highlighted.

The Roe scheme is an approximate Riemann solver that evaluates the convective fluxes from flow quantities reconstructed separately on both sides of the face of the dual control volume from values at the surrounding nodes:

$$\tilde{F}_{ij}^c = \tilde{F}_{ale}^c(U_i, U_j) = \left(\frac{\vec{F}_i^c + \vec{F}_j^c}{2} \right) \cdot \vec{n}_{ij} - \frac{1}{2} P |\Lambda| P^{-1} (U_i - U_j) - (\vec{u}_{\Omega_{ij}} \cdot \vec{n}_{ij}) \left(\frac{U_i + U_j}{2} \right), \quad (4.11)$$

where \vec{n}_{ij} is the outward unit normal associated with the face between nodes i and j , $\vec{u}_{\Omega_{ij}} = (\vec{u}_{\Omega_i} + \vec{u}_{\Omega_j})/2$ is the grid velocity at the dual cell face as an average of the grid velocities at nodes i and j , U_i is the vector of the conserved variables at point i , and \vec{F}_i^c is the convective flux at node i . P is the matrix of eigenvectors of the convective flux Jacobian matrix \vec{A}^c constructed using the Roe-averaged variables and projected in the \vec{n}_{ij} direction, and $|\Lambda|$ is a diagonal matrix with entries corresponding to the absolute value of the eigenvalues of the flux Jacobian matrix. The final term in Eqn. 4.11 is the adjustment required for the convective flux due to the motion of the domain when expressing the equations in ALE form. It can be computed by first averaging the grid velocities \vec{u}_{Ω} and solution vectors U at nodes i and j and then subtracting their product from the traditional Roe flux. It is also important to note that the eigenvalues in $|\Lambda|$ must take into account the grid velocity when computing the dissipation term, as seen in Eqns. 4.8-4.10.

As written, Eqn. 4.11 is first-order accurate in space. However, second-order accuracy can be achieved via reconstruction of variables on the cell interfaces by using a Monotone Upstream-centered Schemes for Conservation Laws (MUSCL) approach [125]. In the MUSCL approach, the conservative variable state on both sides of the control volume face (typically identified as the left state U_L and right state U_R) is reconstructed from the gradients of the variables at each node to project a higher-order approximation of the state at the mid-point of an edge. For second-order accuracy, a linear approximation is applied:

$$U_L \approx U_i + \phi (\nabla U|_i \cdot \vec{r}_{ij}), \quad (4.12)$$

$$U_R \approx U_j + \phi (\nabla U|_j \cdot \vec{r}_{ji}), \quad (4.13)$$

where \vec{r}_{ij} is the vector pointing from node i to the midpoint of edge ij , \vec{r}_{ji} is the vector pointing from node j to the midpoint of edge ij , and we have also introduced a slope limiter that is represented by ϕ . Slope limiting is applied within upwind schemes in order to preserve monotonicity in the solution by limiting the gradients during reconstruction. It has been shown that limiting is essential for obtaining smooth solutions for flows with shocks, in particular. The slope limiters of Barth and Jespersen [8] and Venkatakrisnan [126] are popular options on unstructured meshes. The reconstructed left and right states in Eqns. 4.12 and 4.13 are then used in place of U_i and U_j in Eqn. 4.11, respectively, in order to compute the second-order convective flux.

The JST scheme approximates the convective flux using a central difference with a blend of two types of artificial dissipation to maintain numerical stability by preventing even-odd decoupling of the solution at adjacent nodes. The artificial dissipation terms are computed using the differences in the undivided Laplacians (higher-order dissipation) of connecting nodes and the difference in the

conserved variables (lower-order dissipation) on the connecting nodes. The two levels of dissipation are blended based on a pressure switch for triggering lower-order dissipation in the vicinity of shock waves. The result is a second-order scheme in space. The final expression for the numerical flux using the JST method on unstructured meshes is [76, 82]:

$$\tilde{F}_{ij}^c = \tilde{F}_{ale}^c(U_i, U_j) = \tilde{F}^c \left(\frac{U_i + U_j}{2} \right) \cdot \vec{n}_{ij} - d_{ij} - (\vec{u}_{\Omega_{ij}} \cdot \vec{n}_{ij}) \left(\frac{U_i + U_j}{2} \right). \quad (4.14)$$

Again, the final term in Eqn. 4.14 represents the adjustment to the flux due to grid motion based on the average grid velocity at the cell face. The artificial dissipation d_{ij} along the edge connecting nodes i and j can be expressed as

$$d_{ij} = \left(\varepsilon_{ij}^{(2)}(U_j - U_i) - \varepsilon_{ij}^{(4)}(\nabla^2 U_j - \nabla^2 U_i) \right) \varphi_{ij} \lambda_{ij}, \quad (4.15)$$

where the undivided Laplacians $\nabla^2 U$, local spectral radius, stretching in the grid and pressure switches are computed as

$$\nabla^2 U_i = \sum_{k \in \mathcal{N}(i)} (U_k - U_i), \quad (4.16)$$

$$\lambda_{ij} = (|\vec{u}_{ij} - \vec{u}_{\Omega_{ij}}| + c_{ij}) \Delta S, \quad \lambda_i = \sum_{k \in \mathcal{N}(i)} \lambda_{ik}, \quad (4.17)$$

$$\varphi_{ij} = 4 \frac{\varphi_i \varphi_j}{\varphi_i + \varphi_j}, \quad \varphi_i = \left(\frac{\lambda_i}{4 \lambda_{ij}} \right)^\alpha, \quad (4.18)$$

$$\varepsilon_{ij}^{(2)} = \kappa^{(2)} s_2 \left(\left| \sum_{k \in \mathcal{N}(i)} (p_k - p_i) \right| / \sum_{k \in \mathcal{N}(i)} (p_k + p_i) \right), \quad (4.19)$$

$$\varepsilon_{ij}^{(4)} = s_4 \max \left(0, \kappa^{(4)} - \varepsilon_{ij}^{(2)} \right), \quad (4.20)$$

where $\mathcal{N}(i)$ represents the set of neighboring points to node i , p_i is the pressure at node i , s_2 and s_4 are stretching parameters, α is typically set to 0.3, and $\kappa^{(2)}$ and $\kappa^{(4)}$ are adjustable parameters (typical values on unstructured meshes are $\kappa^{(2)} = 0.5$ and $\kappa^{(4)} = 0.02$).

In this work, the convective term for the scalar variable in the S-A turbulence model is discretized using an upwind scheme. Typically, a first-order scheme is chosen, but the turbulence variable can also take advantage of a MUSCL approach and limiters in order to obtain second-order accuracy. The grid velocity term seen in Eqn. 4.11 must also be subtracted from the upwind flux for the turbulence variable.

Integration of Viscous Fluxes

In order to evaluate the viscous fluxes using the finite volume method, flow quantities and their first derivatives are required at the faces of the control volumes. Unlike the convective flux, the viscous flux is diffusive in nature with no preferential direction of information propagation, and therefore, a central differencing of the viscous flux is a stable and typical practice. The values of the flow

variables (the primitive variables, the dynamic viscosity μ , the heat conduction coefficient k , etc.) and their gradients are averaged at the cell faces for computing the viscous flux in the Navier-Stokes equations:

$$\tilde{F}_{ij}^v = (\mu_{tot}^k \tilde{F}^{vk}) \cdot \vec{n} = \left\{ \begin{array}{c} \cdot \\ \bar{\bar{\sigma}} \\ \bar{\sigma} \cdot \vec{v} + \mu_{tot}^2 c_p \nabla T \end{array} \right\} \cdot \vec{n}. \quad (4.21)$$

Additionally, the following correction [129] is applied in order to reduce the truncation error of the scheme:

$$\nabla \phi \cdot \vec{n} = \frac{\phi_j - \phi_i}{|x_j - x_i|} \alpha_f + \frac{1}{2} (\nabla \phi|_i + \nabla \phi|_j) \cdot (\vec{n} - \alpha_f \vec{s}), \quad (4.22)$$

where \vec{n} is the face normal, \vec{s} is the normalized vector connecting the cell centroid across the face, $|x_j - x_i|$ is the distance between node i and j and α_f is chosen to be the dot product $\alpha_f = \vec{s} \cdot \vec{n}$. The gradients $\nabla \phi|_i$ at node i are computed using either the Green-Gauss or least-squares approach, which will be described below.

Source Term Integration

Source terms are approximated using piecewise constant reconstruction within each of the finite volume cells, as seen in the last term in Eqn. 4.4. More specifically, the source term Q_i is evaluated using the local solution state at node i and then multiplied by the cell volume $|\Omega_i|$ to form the component of the numerical residual due to the source term.

The non-inertial form of the fluid equations contains a source term due to the Coriolis force. As we will see, solving the governing equations in a time-accurate manner using the Dual time-stepping approach also results in a source term involving the discretization of the time derivative term.

Boundary Conditions

In this section, we will describe the boundary conditions that are imposed and their practical implementation for the current work. A variety of strategies exist for imposing boundary conditions with median-dual, cell-vertex based schemes. It should be noted that, unlike cell-centered schemes, the instances of the solution reside on the boundary surface, which requires special consideration.

Typical implementations fall into two categories: strong or weak boundary conditions. Strong boundary conditions directly impose the values of the state variables at the boundary nodes as a Dirichlet condition. In the weak form, rather than impose values of the variables directly, the governing equations are expressed on the boundary (taking into account the imposed physical and numerical boundary conditions) in order to form the corresponding flux representing that condition. At convergence, the weak imposition will satisfy the specified boundary condition. In some instances, a combination of weak and strong boundary conditions is employed.

Flow Tangency Wall Boundary Condition The Euler equations represent fluid flow in the absence of viscosity, and therefore, there is no mechanism for creating friction at a solid wall. Consequently, the solid boundary simply prohibits any flow penetration, and the fluid velocity remains tangent to the surface. In terms of the flow characteristics, there is only one positive eigenvalue, meaning that only one physical boundary condition can be imposed. Therefore, for a solid wall in motion, the flow tangency condition is enforced through the condition

$$(\vec{v} - \vec{u}_\Omega) \cdot \vec{n} = 0. \quad (4.23)$$

Given that the viscous fluxes have vanished ($\vec{F}^v = 0$) for inviscid flow, only the convective fluxes at the wall remain. To form a weak boundary condition, the projected convective flux in ALE form is expressed at the wall with the imposition of the physical condition in Eqn. 4.23, which gives,

$$\vec{F}_{ale}^c \cdot \vec{n} = \left\{ \begin{array}{c} \rho(\vec{v} - \vec{u}_\Omega) \\ \rho\vec{v} \otimes (\vec{v} - \vec{u}_\Omega) + \bar{I}p \\ \rho E(\vec{v} - \vec{u}_\Omega) + p\vec{v} \end{array} \right\} \cdot \vec{n} = \left\{ \begin{array}{c} \cdot \\ p\vec{n} \\ p(\vec{u}_\Omega \cdot \vec{n}) \end{array} \right\}. \quad (4.24)$$

Note that the motion of the surface has resulted in a contribution to the flux at the wall for the energy equation. For fixed walls ($\vec{u}_\Omega = 0$), this term disappears, and only a contribution from the pressure in the momentum equations remains. The flux in Eqn. 4.24 is computed with the local pressure, grid velocity, and normal vector at each boundary node. The flux is then multiplied by the face area on the boundary ΔS before being added to the residual for the current node.

No-Slip Wall Boundary Condition Unlike the Euler equations, the Navier-Stokes equations contain viscous effects, and at solid walls in viscous flow, it is assumed that there can be no relative velocity between the fluid and the wall. While there is still only one incoming characteristic, the addition of the diffusive terms also changes the mathematical character of the governing equations, and consequently, more conditions are needed on the boundary. More specifically, conditions on the fluid velocity and temperature (or their derivatives) are required for the momentum and energy equations.

The no-slip condition implies that, for moving walls, the fluid velocity at the surface must be equal to the velocity of the surface, or

$$\vec{v} = \vec{u}_\Omega. \quad (4.25)$$

Expressing the projected convective and viscous fluxes at the wall while imposing the physical

condition in Eqn. 4.25, one has

$$\begin{aligned}
 (\vec{F}_{ale}^c - \mu_{tot}^k \vec{F}^{vk}) \cdot \vec{n} &= \left\{ \begin{array}{c} \rho(\vec{v} - \vec{u}_\Omega) \\ \rho\vec{v} \otimes (\vec{v} - \vec{u}_\Omega) + \bar{I}p - \bar{\sigma} \\ \rho E(\vec{v} - \vec{u}_\Omega) + p\vec{v} - \bar{\sigma} \cdot \vec{v} - \mu_{tot}^2 c_p \nabla T \end{array} \right\} \cdot \vec{n} \\
 &= \left\{ \begin{array}{c} \cdot \\ p \vec{n} - \bar{\sigma} \cdot \vec{n} \\ p(\vec{u}_\Omega \cdot \vec{n}) - \bar{\sigma} \cdot \vec{u}_\Omega \cdot \vec{n} - \mu_{tot}^2 c_p \partial_n T \end{array} \right\}.
 \end{aligned} \tag{4.26}$$

Eqn. 4.25 can be imposed as a strong condition for the momentum equations by setting the value of the momentum to the current density multiplied by the surface velocity at a given boundary node while removing any contributions to its momentum residual that have been computed from fluxes or source terms involving other nodes. For implicit calculations, this also requires setting the portion of the Jacobian involving the momentum equal to the identity matrix, so that no spurious oscillations in the momentum appear.

After the strong imposition of the condition for the momentum equations, Eqn. 4.26 becomes

$$(\vec{F}_{ale}^c - \mu_{tot}^k \vec{F}^{vk}) \cdot \vec{n} = \left\{ \begin{array}{c} \cdot \\ \cdot \\ p(\vec{u}_\Omega \cdot \vec{n}) - \bar{\sigma} \cdot \vec{u}_\Omega \cdot \vec{n} - \mu_{tot}^2 c_p \partial_n T \end{array} \right\}. \tag{4.27}$$

Again, note that the motion of the surface has resulted in additional contributions to the flux at the wall for the energy equation. The flux terms involving the pressure and shear stress can be computed from the current solution at the node and added to the residual for the energy equation (along with multiplication by the face area). For fixed walls ($\vec{u}_\Omega = 0$), the first two terms left over for the energy equation disappear.

The remaining term in Eqn. 4.27 is evaluated based on the choice of temperature boundary condition at the wall. For adiabatic walls,

$$\partial_n T = 0, \tag{4.28}$$

and there is no contribution from the heat transfer term in Eqn. 4.27. A constant heat flux can also be imposed by setting $\partial_n T$ to a non-zero value, and in this case, the heat transfer term can be directly calculated and added to the flux imposed for the energy equation.

Constant temperature, or isothermal, walls can also be specified by prescribing a value for the temperature on the wall,

$$T = T_S, \tag{4.29}$$

and there are a number of methods for satisfying the boundary condition in this scenario, such as approximating the value of $\partial_n T$ by differencing the wall temperature and the temperature at a suitable (nearest normal) interior neighbor. Other methods involve directly specifying the density

or energy from a prescribed temperature at the wall to give a strong boundary condition [11].

Characteristic-based Flow Boundary Conditions We are interested in the concept of *physical* and *numerical* boundary conditions, as discussed by Hirsch [48]. In general for the flow problem, characteristic-based boundary conditions are imposed at the outer boundaries where the fluid states are updated depending on the sign of the characteristic speeds λ given by the well-known diagonalization of the convective flux Jacobian matrix \bar{A}^c . These eigenvalues are given in Eqn. 4.8–4.10.

Incoming characteristics ($\lambda < 0$) correspond to the propagation of information from the exterior to the interior of the domain. At these locations, physical boundary conditions, such as a far-field state, stagnation conditions, or back pressure, are prescribed.

Outgoing characteristics ($\lambda > 0$) correspond to information propagation from the interior to the exterior of the domain. Numerical boundary conditions are imposed at these locations that extrapolate the characteristic variable information from within the domain in order to complete the update to the state vector at the boundary, i.e., the total number of physical and numerical conditions imposed must total the length of the state vector. The numerical boundary conditions provide the missing information that is not provided by the physical conditions while retaining compatibility with the discretized form of the governing equations. Once the boundary state is constructed in this manner, a weak boundary condition can be imposed by computing the flux (convective and viscous) between the boundary state U_∞ and the current state at a node i on the boundary U_i .

For example, at a far-field boundary, a typical approach involves computing the signs of the eigenvalues and then applying the appropriate combination of the far-field reference state (incoming characteristics) and additional variables that must be extrapolated from the interior of the domain (outgoing characteristics) in order to build the complete boundary state U_∞ . The acoustic Riemann invariants, which correspond to the $\lambda = (\vec{v} - \vec{u}_\Omega) \cdot \vec{n} \pm c$ eigenvalues, can be used to extrapolate the necessary data through the velocity or sound speed:

$$\mathcal{R}^+ = \vec{v} \cdot \vec{n} + \frac{2c}{(\gamma - 1)}, \quad (4.30)$$

$$\mathcal{R}^- = \vec{v} \cdot \vec{n} - \frac{2c}{(\gamma - 1)}. \quad (4.31)$$

Similar approaches using the Riemann invariants are available for subsonic inlets and outlets on truncated domains where different physical conditions are imposed, such as stagnation conditions or back pressure [11]. Supersonic boundaries are considerably more straightforward, as all characteristics propagate in the same direction. The result is that either the entire boundary state can be imposed as a physical condition (supersonic inlet) or the entire boundary state is extrapolated from the interior of the domain as a numerical condition (supersonic outlet).

4.1.2 Discretizing the Continuous Adjoint Equations

Consider now the continuous adjoint equations in the most general form previously derived in Chapter 3, which will require some special treatment given differences in nature from the flow equations.

The adjoint equations are typically transposed before discretization to give

$$-\frac{\partial \Psi}{\partial t} - \left(\vec{A}^c - \bar{I}_5 \vec{u}_\Omega - \mu_{tot}^k \vec{A}^{vk} \right)^\top \cdot \nabla \Psi - \nabla \cdot \left(\mu_{tot}^k \bar{D}^{vk^\top} \cdot \nabla \Psi \right) - \frac{\partial Q^\top}{\partial U} \Psi = 0, \quad (4.32)$$

where Ψ is the vector of adjoint variables, \vec{A}^c represents convective flux Jacobian, \vec{u}_Ω is the velocity of the domain, μ_{tot}^k is the viscosity, \vec{A}^{vk} represents the Jacobian of the viscous fluxes with respect to the conservative variables, \bar{D}^{vk} represents the Jacobian of the viscous fluxes with respect to the gradients of the conservative variables, and Q is a source term that appeared in the flow equations. Following again the FVM, we integrate the differential form of the equations over a control volume surrounding node i (Ω_i) to find

$$\int_{\Omega_i} \left(-\frac{\partial \Psi}{\partial t} - \left(\vec{A}^c - \bar{I}_5 \vec{u}_\Omega - \mu_{tot}^k \vec{A}^{vk} \right)^\top \cdot \nabla \Psi - \nabla \cdot \left(\mu_{tot}^k \bar{D}^{vk^\top} \cdot \nabla \Psi \right) - \frac{\partial Q^\top}{\partial U} \Psi \right) d\Omega = 0, \quad (4.33)$$

which can be rearranged into temporal, convective, viscous, and source components as

$$\begin{aligned} - \int_{\Omega_i} \frac{\partial \Psi}{\partial t} d\Omega - \int_{\Omega_i} \left(\vec{A}^{c^\top} - \bar{I}_5 \vec{u}_\Omega \right) \cdot \nabla \Psi d\Omega - \int_{\Omega_i} \nabla \cdot \left(\mu_{tot}^k \bar{D}^{vk^\top} \cdot \nabla \Psi \right) d\Omega \\ + \int_{\Omega_i} \left(\mu_{tot}^k \vec{A}^{vk^\top} \cdot \nabla \Psi - \frac{\partial Q^\top}{\partial U} \Psi \right) d\Omega = 0. \end{aligned} \quad (4.34)$$

The convective term (second term on the left-hand side of Eqn. 4.34) is not expressed in conservative form, and therefore, the Divergence theorem can not be directly applied to transform it into a surface integral over the control volume, as can be done for the viscous term (third term on the left-hand side of Eqn. 4.34). While the Divergence theorem could still be applied if the appearance of an additional term from an integration by parts is treated, a different approach will be pursued here. The convective term can be integrated in a non-conservative fashion by moving the Jacobian matrices outside of the integral and applying the Divergence theorem to the gradient of the adjoint variables. This requires the non-constant coefficients in front of the integral to be evaluated using the data at the current node i and ultimately results in a non-conservative convective scheme where $\tilde{F}_{ij}^c \neq \tilde{F}_{ji}^c$ when computing fluxes through the faces of the dual control volumes associated with each edge in the mesh. Applying this strategy, we transform Eqn. 4.34 into

$$\begin{aligned} - \int_{\Omega_i} \frac{\partial \Psi}{\partial t} d\Omega - \left(\vec{A}_i^{c^\top} - \bar{I}_5 \vec{u}_{\Omega_i} \right) \cdot \int_{\partial \Omega_i} \Psi \vec{n} ds - \int_{\partial \Omega_i} \mu_{tot}^k \bar{D}^{vk^\top} \cdot \nabla \Psi \cdot \vec{n} ds \\ + \int_{\Omega_i} \left(\mu_{tot}^k \vec{A}^{vk^\top} \cdot \nabla \Psi - \frac{\partial Q^\top}{\partial U} \Psi \right) d\Omega = 0. \end{aligned} \quad (4.35)$$

Discretizing the convective and viscous fluxes with suitable methods (discussed below) and assuming a piecewise constant source term, we recover the semi-discretized integral form of the adjoint

equations, or

$$-\int_{\Omega_i} \frac{\partial \Psi}{\partial t} d\Omega - \sum_{j \in \mathcal{N}(i)} (\tilde{F}_{ij}^c + \tilde{F}_{ij}^v) \Delta S_{ij} + |\Omega_i| \mu_{tot_i}^k \vec{A}_i^{vk^T} \cdot \nabla \Psi|_i - |\Omega_i| \left. \frac{\partial \mathcal{Q}}{\partial U} \right|_i \Psi_i = 0, \quad (4.36)$$

where \tilde{F}_{ij}^c and \tilde{F}_{ij}^v are the projected numerical approximations of the convective and viscous fluxes along an edge ij (evaluated at the midpoint), respectively, ΔS_{ij} is the area of the face associated with the edge ij , $|\Omega_i|$ is the volume of the control volume in the dual mesh, and $\mathcal{N}(i)$ are the neighboring nodes to node i . Fig. 4.2 contains graphical descriptions of the geometric terms. Each of the terms resulting from the spatial discretization can be summed together into a single numerical residual $R_i(\Psi)$ at each node:

$$-\int_{\Omega_i} \frac{\partial \Psi}{\partial t} d\Omega + R_i(\Psi) = 0. \quad (4.37)$$

Integration of Convective Fluxes

The convective term of the continuous adjoint equations can be discretized using a modified JST scheme that features a single type of high-order artificial dissipation. The final expression for the modified JST scheme is given as follows:

$$\tilde{F}_{ij}^c = \tilde{F}_{ale}^c(\Psi_i, \Psi_j) = \vec{F}^c \left(\frac{\Psi_i + \Psi_j}{2} \right) \cdot \vec{n}_{ij} - d_{ij} - (\vec{u}_{\Omega_{ij}} \cdot \vec{n}_{ij}) \left(\frac{\Psi_i + \Psi_j}{2} \right). \quad (4.38)$$

Again, the final term in Eqn. 4.38 represents the adjustment to the flux due to grid motion based on the average grid velocity at the cell face. Due to the non-conservative nature ($\tilde{F}_{ij}^c \neq \tilde{F}_{ji}^c$), the non-constant coefficients of \vec{A}^{c^T} for the convective flux are evaluated using the flow solution data U_i at the current node i , or

$$\vec{F}^c \left(\frac{\Psi_i + \Psi_j}{2} \right) \cdot \vec{n}_{ij} = \vec{A}_i^{c^T} \cdot \vec{n}_{ij} \left(\frac{\Psi_i + \Psi_j}{2} \right). \quad (4.39)$$

The artificial dissipation d_{ij} along the edge connecting nodes i and j can be expressed as

$$d_{ij} = \kappa^{(4)} \varepsilon^{(4)} (\nabla^2 \Psi_j - \nabla^2 \Psi_i) \omega_{ij} \lambda_{ij}, \quad (4.40)$$

$$\nabla^2 \Psi_i = \sum_{k \in \mathcal{N}(i)} (\Psi_k - \Psi_i), \quad (4.41)$$

$$\varepsilon^{(4)} = \left(3 \frac{N_i + N_j}{N_i N_j} \right)^2, \quad (4.42)$$

$$\lambda_i = (|\vec{u}_i - \vec{u}_{\Omega_i}| \cdot \vec{n}_i + c_i) \Delta S, \quad (4.43)$$

$$\lambda_{ij} = (|\vec{u}_{ij} - \vec{u}_{\Omega_{ij}}| \cdot \vec{n}_{ij} + c_{ij}) \Delta S, \quad (4.44)$$

$$\omega_i = \left(\frac{\lambda_i}{4\lambda_{ij}} \right)^{\frac{1}{2}}, \quad (4.45)$$

$$\omega_{ij} = \frac{\omega_i \omega_j}{\omega_i + \omega_j}, \quad (4.46)$$

where N_i represents the size of the set of neighboring points to node i , $\vec{u}_{ij} = \frac{1}{2}(\vec{u}_i + \vec{u}_j)$ and $c_{ij} = \frac{1}{2}(c_i + c_j)$ are the fluid and sound speeds at the cell face, and $\kappa^{(4)}$ is an adjustable parameter (typical values on unstructured meshes are $\kappa^{(4)} = 0.02$ or lower in order to minimize artificial dissipation). An artificial dissipation of upwind-type can be used as well [18].

Integration of Viscous Fluxes

As in the flow problem, the viscous fluxes for the adjoint equations are integrated using an average of gradients method with a correction [129]. The viscous flux can be expressed as

$$\tilde{F}_{ij}^v = \mu_{tot}^k \bar{D}^{vk^T} \cdot \nabla \Psi \cdot \vec{n}, \quad (4.47)$$

where the gradients of the adjoint variables are averaged on the faces of the control volumes. However, to be consistent with the convective flux discretization above, the flow solution variables are evaluated at the current node i .

Switching to index notation (repeated indices imply summation), one has the following for the projected viscous adjoint flux along an edge:

$$\tilde{F}^v = \frac{1}{\rho} \left\{ \begin{array}{c} -v_i \Sigma_{ij} n_j + \left(\frac{|\vec{v}|^2}{2} - \frac{p}{(\gamma-1)\rho} \right) \Sigma^{\psi_{\rho E}} \\ \Sigma_{1j} n_j - v_1 \Sigma^{\psi_{\rho E}} \\ \Sigma_{2j} n_j - v_2 \Sigma^{\psi_{\rho E}} \\ \Sigma_{3j} n_j - v_3 \Sigma^{\psi_{\rho E}} \\ \Sigma^{\psi_{\rho E}} \end{array} \right\}, \quad (4.48)$$

where the following abbreviations have been used

$$\Sigma_{ij} = \Sigma_{ij}^{\varphi} + \Sigma_{ij}^{\psi_{\rho E}} \quad (4.49)$$

$$\Sigma_{ij}^{\varphi} = \mu_{tot}^1 \left(\partial_j \varphi_i + \partial_i \varphi_j - \frac{2}{3} \delta_{ij} \nabla \cdot \vec{\varphi} \right) \quad (4.50)$$

$$\Sigma_{ij}^{\psi_{\rho E}} = \mu_{tot}^1 \left(v_j \partial_i \psi_{\rho E} + v_i \partial_j \psi_{\rho E} - \frac{2}{3} \delta_{ij} \vec{v} \cdot \nabla \psi_{\rho E} \right) \quad (4.51)$$

$$\Sigma_i^{\psi_{\rho E}} = \gamma \mu_{tot}^2 \partial_i \psi_{\rho E} \quad (4.52)$$

$$\Sigma^{\psi_{\rho E}} = \gamma \mu_{tot}^2 \nabla \psi_{\rho E} \cdot \vec{n}. \quad (4.53)$$

Source Term Integration

Source terms in the adjoint equations are again approximated using piecewise constant reconstruction within each of the finite volume cells. The remaining terms in Eqn. 4.36 are treated as source terms, and this contribution to the residual at node i can be written as

$$R_i^{source}(\Psi) = |\Omega_i| \mu_{tot_i}^k \vec{A}_i^{vkT} \cdot \nabla \Psi|_i - |\Omega_i| \left. \frac{\partial Q^T}{\partial U} \right|_i \Psi_i, \quad (4.54)$$

where all terms are evaluated using the available flow and adjoint variables (and their gradients) at node i .

Boundary Conditions

In this section, we will describe the adjoint boundary conditions that are imposed and their practical implementation for the current work. These boundary conditions were derived in Chapter 3 as part of the unsteady adjoint derivation, and they are specific to a particular choice of cost function and the type of boundary conditions imposed in the primal problem. Details for solid wall and characteristic-based adjoint boundary conditions for force-based functionals are given below.

Flow Tangency Wall Boundary Condition for the Adjoint Euler Equations

For a force-based objective function in inviscid flow, the admissible boundary conditions for $\vec{\varphi}$ can be found in Eqn. 3.113, or

$$\vec{\varphi} \cdot \vec{n} = (\vec{d} - \psi_{\rho E} \vec{v}) \cdot \vec{n}. \quad (4.55)$$

Eqn. 4.55 can be used to build a weak boundary condition for all equations at the boundary node i . First, Eqn. 4.55 is introduced through an update of $\vec{\varphi}$ at the boundary node by adjusting $\vec{\varphi}$ based on the difference between the left-hand side of Eqn. 4.55 (using the current solution) and the right-hand side (the boundary condition we are imposing). This condition is applied with each iteration of the solver in order to build an updated adjoint state Ψ from the new $\vec{\varphi}$ along with the current values for ψ_{ρ} and $\psi_{\rho E}$.

After building an updated state for Ψ , the boundary condition is applied weakly by computing the adjoint convective flux at the boundary node. Expressing the projected convective flux at the solid wall, one has

$$\begin{aligned} \tilde{F}^c &= (\vec{A}^{c\top} - \bar{I}_5 \vec{u}_\Omega) \cdot \vec{n} \\ &= \left\{ \begin{array}{l} -(\vec{\varphi} \cdot \vec{v})(\vec{v} \cdot \vec{n}) + (\vec{\varphi} \cdot \vec{n})(\gamma - 1) \frac{\bar{v}^2}{2} + \psi_{\rho E}(\vec{v} \cdot \vec{n})(\gamma - 1) \frac{\bar{v}^2}{2} - \psi_{\rho E} H(\vec{v} \cdot \vec{n}) - \psi_\rho (\vec{u}_\Omega \cdot \vec{n}) \\ (\psi_\rho + \vec{\varphi} \cdot \vec{v} + \psi_{\rho E} H) \vec{n} + \vec{\varphi}(\vec{v} \cdot \vec{n}) - \vec{v}(\vec{\varphi} \cdot \vec{n})(\gamma - 1) - \psi_{\rho E} \vec{v}(\vec{v} \cdot \vec{n})(\gamma - 1) - \vec{\varphi}(\vec{u}_\Omega \cdot \vec{n}) \\ (\vec{\varphi} \cdot \vec{n})(\gamma - 1) + \psi_{\rho E}(\vec{v} \cdot \vec{n})\gamma - \psi_{\rho E}(\vec{u}_\Omega \cdot \vec{n}) \end{array} \right\}. \end{aligned} \quad (4.56)$$

The flux contributions in Eqn. 4.58 are then included in the adjoint residual at each node on the surface to complete the boundary update.

No-Slip Wall Boundary Condition for the Adjoint Navier-Stokes Equations

For a force-based objective function and a no-slip wall condition in the flow problem, the admissible boundary conditions for the adjoint momentum equations can be found in Eqn. 3.100, or

$$\begin{aligned} \vec{\varphi} &= \vec{d} - \psi_{\rho E} \vec{v} \\ &= \vec{d} - \psi_{\rho E} \vec{u}_\Omega, \end{aligned} \quad (4.57)$$

where we will assume that the flow boundary condition was satisfied, or $\vec{v} = \vec{u}_\Omega$. Eqn. 4.57 can be imposed as a strong condition for the adjoint momentum equations by setting the value of $\vec{\varphi}$ at a given boundary node while removing any contributions to its momentum residual that have been computed from fluxes or source terms involving other nodes. For implicit calculations, this also requires setting the portion of the Jacobian involving the momentum equal to the identity matrix, so that no spurious oscillations in the adjoint momentum appear.

However, conditions on the adjoint density and adjoint energy must still be defined. Expressing the projected convective and viscous adjoint fluxes at the wall while imposing the condition in Eqn. 4.57, one has

$$\begin{aligned} &(\tilde{F}^c + \tilde{F}^v) \\ &= (\vec{A}^{c\top} - \bar{I}_5 \vec{u}_\Omega + \mu_{tot}^k \bar{D}^{vk\top} \cdot \nabla \Psi) \cdot \vec{n} \\ &= \left\{ \begin{array}{l} -(\psi_\rho + \vec{\varphi} \cdot \vec{u}_\Omega + \psi_{\rho E} H)(\vec{u}_\Omega \cdot \vec{n}) + (\vec{d} \cdot \vec{n})(\gamma - 1) \frac{\bar{u}_\Omega^2}{2} - \frac{\vec{u}_\Omega \cdot \bar{\Sigma} \cdot \vec{n}}{\rho} + \frac{1}{\rho} \left(\frac{\bar{u}_\Omega^2}{2} - \frac{p}{(\gamma - 1)\rho} \right) \Sigma^{\psi_{\rho E}} \\ (\vec{d} \cdot \vec{n})(\gamma - 1) + \frac{1}{\rho} \Sigma^{\psi_{\rho E}} \end{array} \right\}, \end{aligned} \quad (4.58)$$

where $\bar{\Sigma} = \Sigma_{ij}$ using the definitions in Eqn. 4.49–4.53. The flux contributions to the adjoint density and adjoint energy in Eqn. 4.58 are added to the adjoint residual at each node on the surface in

order to impose a weak boundary condition for those equations.

When adiabatic walls are imposed in the flow problem, the additional boundary condition on the adjoint energy (from Eqn. 3.100) becomes

$$\partial_n \psi_{\rho E} = \nabla \psi_{\rho E} \cdot \vec{n} = 0. \quad (4.59)$$

Applying Eqn. 4.59, we have the result that $\Sigma^{\psi_{\rho E}} = 0$, and the flux contributions to the adjoint density and adjoint energy in Eqn. 4.58 simplify. If isothermal walls are specified in the flow problem, the boundary condition on the adjoint energy is different (from Eqn. 3.109):

$$\psi_{\rho E} = 0. \quad (4.60)$$

and the terms involving $\Sigma^{\psi_{\rho E}}$ in Eqn. 4.58 can be calculated and added as part of the weak boundary condition imposed on the adjoint density and adjoint energy.

Characteristic-based Adjoint Boundary Conditions

At far-field, inlet, or outlet boundaries, characteristic-based boundary conditions can be applied to the adjoint problem that are complementary to their characteristic-based counterparts for the flow problem. The reader is referred to the discussion in Chapter 3 concerning the derivation of the appropriate adjoint boundary conditions. A brief summary of the practical application is given here.

We assume that the far-field is far enough removed that the viscous perturbation terms are negligible on this boundary. This leaves only the convective boundary term at the far-field, and we would like to remove any contributions from this integral. The integral over Γ_∞ involving B_1 in Eqn. 3.60 will indeed vanish if the scalar integrand is zero at every point on the boundary with each physical time step, or

$$\Psi^T \left(\vec{A}^c - \bar{\bar{I}}_5 \vec{u}_\Omega \right) \cdot \vec{n} \delta U = 0. \quad (4.61)$$

Therefore, we seek the adjoint state Ψ at the boundary that eliminates any contribution from this integral to the variation of the functional by satisfying the preceding expression. A common strategy for the removal of this integral is the imposition of a homogeneous adjoint boundary condition: $\Psi = \mathbf{0}$. However, in Chapter 3, we derived mathematically consistent types of characteristic-based boundary conditions for subsonic inlets (mass flow) and outlets (back pressure). The strategy detailed in that section can be repeated as a systematic procedure for finding boundary conditions when other types of physical conditions are imposed in the direct problem (such as prescribing stagnation conditions at a subsonic inlet).

Once the appropriate characteristic-based condition has been applied to compute an adjoint state Ψ_∞ at the boundary, a weak boundary condition can be imposed by computing the convective flux between the boundary state (U_∞ and Ψ_∞) and the current state at a node i on the boundary (U_i and Ψ_i).

4.1.3 Evaluating Gradients of the Flow and Adjoint Variables

Gradients of the flow and adjoint variables are needed in order to compute higher-order convective flux reconstructions, such as with the second-order Roe scheme, viscous fluxes, and the expressions for surface sensitivity that result from the adjoint derivations. Two well-established strategies for computing these gradients on unstructured meshes with a median-dual cell-vertex scheme take advantage of the Green-Gauss and least-squares methods [6, 79]. The Green-Gauss method is briefly outlined below for completeness.

Green-Gauss Method

A straightforward method for evaluating gradients can be formulated by taking advantage of the Green-Gauss theorem in order to relate the gradient of a scalar ϕ to an integral over the surface of a control volume, or

$$\int_{\Omega_i} \nabla \phi d\Omega = \int_{\partial\Omega_i} \phi \vec{n} ds. \quad (4.62)$$

By assuming a constant value of the gradient over the dual control volume, a semi-discrete expression for the gradient can be written as

$$|\Omega_i| \nabla \phi|_i \approx \int_{\partial\Omega_i} \phi \vec{n} ds, \quad (4.63)$$

which can be evaluated at each node of the computational mesh by looping over all edges connected to node i and averaging the values of ϕ at the neighboring nodes j to construct a value for each face of the control volume associated with edge ij , or

$$\nabla \phi|_i \approx \frac{1}{|\Omega_i|} \sum_{j \in \mathcal{N}(i)} \frac{1}{2} (\phi_i + \phi_j) \vec{n}_{ij} \Delta S_{ij}. \quad (4.64)$$

Note that this includes a contribution from any boundary faces if node i lies on the boundary of the domain.

4.2 Time Integration

This section discusses techniques for integrating both the flow equations and adjoint equations in time. Several schemes will be described, including explicit and implicit schemes for both steady state and time-accurate integration. For dynamic meshes, methods for moving the grid and computing grid velocities for the ALE form of the governing equations are needed, and several options that fall under the general category of rigid transformation or dynamic deformation for grids will be discussed.

4.2.1 Time-marching Schemes for the Governing Flow Equations

We now consider the techniques for time-marching the coupled system of ordinary differential equations for the flow problem represented by Eqn. 4.5, which is repeated here:

$$\frac{d}{dt} \int_{\Omega_i(t)} U d\Omega + R_i(U) = 0. \quad (4.65)$$

By discretizing the time derivative term, one obtains a fully-discrete finite volume form of the governing equations. The choice of time-marching method depends on whether a steady state or a time-accurate solution is desired. In both cases, explicit and implicit methods are available. For simplicity, Eqn. 4.65 can be rewritten as

$$\frac{d}{dt} (|\Omega_i|U_i) + R_i(U) = 0, \quad (4.66)$$

where $|\Omega_i| = \int_{\Omega_i(t)} d\Omega$.

Integration to Steady State

For problems with a steady state solution, time-accuracy is not required during integration, as the time derivative term will vanish as the solution approaches the steady state. Therefore, accuracy in time is not the primary driver when choosing a time discretization for steady problems. In fact, a simple time discretization, such as a first-order accurate scheme, is often selected.

The governing equations must be satisfied over the entire time interval during integration, so Eqn. 4.66 can be discretized either explicitly or implicitly by evaluating the residual $R_i(U)$ at the time t^n or t^{n+1} , respectively. Time level n corresponds to the known solution in its current state, while time level $n + 1$ represents the new solution state that is being sought after advancing one time step Δt where $\Delta t = t^{n+1} - t^n$.

For example, the simplest explicit scheme is a forward Euler method where the residual is evaluated using the solution in the current state U^n and the time derivative is discretized with a first-order forward difference. Applying this to Eqn. 4.66, one has

$$|\Omega_i| \frac{\Delta U_i}{\Delta t_i} = -R_i(U^n), \quad (4.67)$$

where we have assumed that $|\Omega_i|$ is constant for the steady problem and $\Delta U_i = U_i^{n+1} - U_i^n$. An expression for the new solution state can be obtained by rearranging terms to give

$$U_i^{n+1} = U_i^n - \frac{\Delta t_i}{|\Omega_i|} R_i(U^n). \quad (4.68)$$

The solution is relaxed to a steady state by the repeated application of Eqn. 4.68 with a suitable time step Δt , which must be estimated in a manner that obeys the stability limits of the particular time-marching scheme. For steady problems, a constant time step for all cells is not required, and

a local time-stepping technique can be used to accelerate convergence to a steady state. Local time-stepping allows each cell in the mesh to advance the solution at a local time step that can be calculated from an estimation of the spectral radii at every node i according to

$$\Delta t_i = N_{CFL} \min \left(\frac{|\Omega_i|}{\lambda_i^{conv}}, \frac{|\Omega_i|}{\lambda_i^{visc}} \right), \quad (4.69)$$

where N_{CFL} is the Courant-Friedrichs-Lewy (CFL) number and λ_i^{conv} is the integrated convective spectral radius [32] computed as

$$\lambda_i^{conv} = \sum_{j \in \mathcal{N}(i)} (|(\vec{u}_{ij} - \vec{u}_{\Omega_{ij}}) \cdot \vec{n}_{ij}| + c_{ij}) \Delta S, \quad (4.70)$$

where $\vec{u}_{ij} = (\vec{u}_i + \vec{u}_j)/2$, $\vec{u}_{\Omega_{ij}} = (\vec{u}_{\Omega_i} + \vec{u}_{\Omega_j})/2$, and $c_{ij} = (c_i + c_j)/2$ denote the velocity, grid velocity, and the speed of sound at the cell face as an average of the neighboring nodes, respectively. The viscous spectral radius λ_i^{visc} is computed as

$$\lambda_i^{visc} = \sum_{j \in \mathcal{N}(i)} C \frac{\mu_{ij}}{\rho_{ij}} S_{ij}^2, \quad (4.71)$$

where C is a constant, μ_{ij} is the sum of the laminar and eddy viscosities in a turbulent calculation and ρ_{ij} is the density evaluated at the midpoint of the edge ij .

While the forward Euler method is very straightforward and easy to implement, it suffers from a small time step requirement in order to maintain numerical stability. Other explicit schemes with larger stability bounds are available, such as Runge-Kutta methods, and these are a popular compromise between convergence rates and computational effort.

For particularly stiff problems (often caused by source terms, for instance), the small time step requirement for explicit schemes may become prohibitive, and implicit methods can be used to improve convergence due to their increased numerical stability. Perhaps the most common implicit method for steady flows is the backward Euler scheme, where the residual is evaluated using the solution state at the new time level, U^{n+1} . Applying this to Eqn. 4.66, one has

$$|\Omega_i| \frac{\Delta U_i}{\Delta t_i} = -R_i(U^{n+1}), \quad (4.72)$$

but the residuals at time level $n+1$ are now a function of the unknown solution state U^{n+1} and can not be directly computed. To remedy this, a first-order linearization about time level n can be performed:

$$\begin{aligned} R_i(U^{n+1}) &= R_i(U^n) + \frac{\partial R_i(U^n)}{\partial t} \Delta t_i^n + \mathcal{O}(\Delta t^2), \\ &= R_i(U^n) + \sum_{j \in \mathcal{N}(i)} \frac{\partial R_i(U^n)}{\partial U_j} \Delta U_j^n + \mathcal{O}(\Delta t^2). \end{aligned} \quad (4.73)$$

Finally, the following linear system should be solved to find the solution update (ΔU_i^n):

$$\left(\frac{|\Omega_i|}{\Delta t_i^n} \delta_{ij} + \frac{\partial R_i(U^n)}{\partial U_j} \right) \cdot \Delta U_j^n = -R_i(U^n), \quad (4.74)$$

where if a flux \tilde{F}_{ij} has a stencil of points $\{i, j\}$, then contributions are made to the Jacobian at four points, or

$$\frac{\partial R}{\partial U} := \frac{\partial R}{\partial U} + \begin{bmatrix} \ddots & & & & \\ & \frac{\partial \tilde{F}_{ij}}{\partial U_i} & \dots & \frac{\partial \tilde{F}_{ij}}{\partial U_j} & \\ & \vdots & \ddots & \vdots & \\ & -\frac{\partial \tilde{F}_{ij}}{\partial U_i} & \dots & -\frac{\partial \tilde{F}_{ij}}{\partial U_j} & \\ & & & & \ddots \end{bmatrix}. \quad (4.75)$$

Note that, while implicit schemes offer more stability, the use of approximate Jacobians (first-order) can impose limits on the allowable time step, especially at the beginning of the solution process when we are not near the converged solution. However, implicit methods enable the use of higher CFL conditions than with explicit methods, which translate to the specific values of Δt_i^n that are used to relax the problem.

The system given by Eqn. 4.74 is smoothed with each nonlinear iteration of the solver for some number of linear iterations or until reaching a prescribed convergence tolerance. Common linear solver choices for modern solvers include preconditioned Krylov methods. Preconditioning is the application of a transformation to the original system that makes it more suitable for numerical solution [102]. In particular, Jacobi, Lower-Upper Symmetric-Gauss-Seidel (LU-SGS), and line implicit preconditioners have been implemented to improve the convergence rate of the available linear solvers [116, 78]. Currently, the following two Krylov subspace methods are available in SU2:

- The Generalized Minimal Residual (GMRES) method [109].
- The Biconjugate Gradient Stabilized (BiCGSTAB) method [123].

Time-Accurate Integration

For unsteady flows, accuracy in time is desired, and therefore, the time discretization scheme must be treated more carefully. A dual time-stepping strategy [52, 61] has been implemented to achieve high-order accuracy in time. In this method, the unsteady problem is transformed into a series of steady problems at each physical time step that can then be solved using all of the well-known convergence acceleration techniques for steady problems.

To demonstrate the dual time-stepping strategy, we start again from Eqn. 4.66 and choose a second-order backward difference formula to discretize the time derivative of the coupled set of ODEs while allowing for the control volumes to change with time, or

$$\frac{3}{2\Delta t} |\Omega_i|^{n+1} U_i^{n+1} - \frac{2}{\Delta t} |\Omega_i|^n U_i^n + \frac{1}{2\Delta t} |\Omega_i|^{n-1} U_i^{n-1} + R_i(U^{n+1}) = 0, \quad (4.76)$$

where Δt is a chosen physical time step and we have also assumed that the equations will be solved in a fully implicit manner by expressing the spatial residual in terms of the solution at time level $n + 1$. Eqn. 4.76 is then recast as a new form for the residual $R^*(U)$, and a fictitious, or pseudo, time derivative is introduced in front of the original governing equations to give

$$\frac{\partial U}{\partial \tau} + R_i^*(U) = 0, \quad (4.77)$$

where

$$R_i^*(U) = \frac{3}{2\Delta t}U_i + \frac{1}{|\Omega_i|^{n+1}} \left(R_i(U) - \frac{2}{\Delta t}|\Omega_i|^n U_i^n + \frac{1}{2\Delta t}|\Omega_i|^{n-1} U_i^{n-1} \right). \quad (4.78)$$

Therefore, Eqn. 4.77 has assumed the same form that is used to converge the equations to a steady state, and all of the corresponding techniques for relaxing the equations to a steady solution that are detailed in the previous section apply in the new fictitious time, τ . This can include implicit methods, local time-stepping, or multigrid strategies, for example.

Converging the steady problem in pseudo time at each physical time step results in the modified residual being equal to zero, or $R_i^*(U) = 0$, which is equivalent to finding the value of U^{n+1} that satisfies Eqn. 4.76. This can also be seen by considering that the value of U in Eqn. 4.78 is the only unknown, and once the problem has been relaxed in pseudo time, $U = U^{n+1}$. Note also that many of the terms in Eqn. 4.78 are known quantities and can be included as a fixed source term during integration. However, this requires that those values of the solution and cell volumes at previous time levels are stored in memory. Therefore, after relaxing the steady problem at each physical time step, these quantities are stored before moving to the next physical time. For completeness, a first-order accurate in time form for Eqn. 4.78 can be written as

$$R_i^*(U) = \frac{1}{\Delta t}U_i + \frac{1}{|\Omega_i|^{n+1}} \left(R_i(U) - \frac{1}{\Delta t}|\Omega_i|^n U_i^n \right). \quad (4.79)$$

4.2.2 Time-marching Schemes for the Continuous Adjoint Equations

We now consider techniques for time-marching the coupled system of ordinary differential equations for the adjoint problem represented by Eqn. 4.36, which is repeated here:

$$- \int_{\Omega_i} \frac{\partial \Psi}{\partial t} d\Omega + R_i(\Psi) = 0. \quad (4.80)$$

As $R_i(\Psi)$ represents the spatial discretization of the adjoint problem, a fully-discrete finite volume form of the governing equations is obtained by discretizing the time derivative term. For simplicity, Eqn. 4.80 can be rewritten as

$$- \frac{d}{dt} (|\Omega_i| \Psi_i) + R_i(\Psi) = 0, \quad (4.81)$$

where $|\Omega_i| = \int_{\Omega_i(t)} d\Omega$.

Many of the same techniques for time-marching that were applied to the flow equations can also

be used for the continuous adjoint equations. The same type of forward Euler method as described by Eqn. 4.68 for the flow equations could be directly applied to Eqn. 4.81 in order to relax the adjoint problem to a steady state with an appropriate time step. Moreover, as the adjoint problem features an equivalent set of characteristic speeds as compared to the flow problem (apart from a reversal in sign), the time step criteria for the flow problem given by Eqn. 4.69 can also be directly applied. Similarly, other techniques for solving the steady state problem (implicit methods, local time-stepping, multigrid, etc.) can be immediately applied to the adjoint problem.

Due to the reversal of characteristic information in the adjoint problem, solving the adjoint equations requires integration in reverse time. This is accomplished by writing the solution data to disk at each time step during the direct problem and then retrieving the data in reverse order while time-marching the adjoint equations. This includes the state of the grid at each step, including cell volumes and grid velocities, in order to be consistent with what was computed for the direct problem. Some techniques do exist that can ease the burden of data storage for the unsteady adjoint, such as checkpointing [128, 121].

4.2.3 Convergence Acceleration

For typical iterative numerical solution methods, high-frequency (local) errors in the solution are well-damped, while lower frequency (global) errors are poorly damped by the action of methods that have a stencil with a local area of influence. Therefore, the low-frequency errors are difficult to eliminate, which leads to slower solver convergence, especially on fine meshes. A multigrid strategy can recover effective rates of convergence at all scales can be maintained in a solver by leveraging a sequence of grids at various resolutions. With geometric multigrid, multiple levels of physical grids with varying resolution are used to provide better approximations of the solution with each step of an iterative solution method.

SU2 contains a Full-Approximation Storage (FAS) multigrid implementation that generates effective convergence at all length scales of a problem by employing a sequence of grids of varying resolution. SU2 will automatically generate the coarse grids from the provided fine grid at runtime using a built-in agglomeration procedure. In short, the goal is to accelerate the convergence of the numerical solution of a set of equations by computing corrections to the fine-grid solutions on coarser grids and applying this idea recursively [59, 78, 77, 12, 96].

4.2.4 Dynamic Meshes

Apart from solving the governing equations in ALE form, the handling of dynamic meshes forms the other major component of calculating unsteady flows with moving surfaces or domains. With each new physical time step, the nodal coordinates and velocities must be updated using suitable methods for moving any boundaries and interior nodes of the volume mesh and computing the resulting grid velocities. Typical strategies for dynamic meshes can be grouped into several categories:

- Rigidly transforming grids - the entire mesh is treated as a rigid body that can be translated in space or rotated about a point

- Dynamically deforming grids - the displacement of the mesh boundaries is provided and the interior of the volume mesh is deformed to maintain a conforming grid
- Overlapping grid techniques - sliding mesh interfaces or overset grid methods can allow for large relative motion between different surfaces or regions of the domain
- Remeshing - after imposing any surface motions, the computational grid is regenerated (either partially or in full)

As an example, consider the external flow past a pitching airfoil with a typical far-field boundary. Each of the above strategies could be applied, as only a single, rigid surface is in motion. If a solid boundary deforms with time, as with aeroelastic calculations, then deforming grid approaches are needed to adapt the mesh to the changing shape of the geometry. Grid deformation can also be employed when limited relative motion occurs between multiple surfaces within a single domain, such as small pitching or plunging motions of independent surfaces. However, should the relative motions become too large, the grid deformation routines may fail to converge and/or produce poor-quality or completely unusable grids with negative control volumes.

In the more complicated case where multiple surfaces exhibit large relative motion within a single computational domain, such as with counter-rotating rotors, rotor-fuselage interactions for rotorcraft or fixed-wing vehicles, or rotor-stator interactions within a turbomachine, then a system of sub-grids, or zones, with independent motion will be needed. Furthermore, communication of solution information between these zones must be achieved through a sliding mesh interface, an overset mesh approach, a mixing plane approximation, or another method that appropriately considers the relative grid motion [124].

A remeshing approach to regenerate the volume grid with each time step after surface motions is the most general, but it is often prohibitively expensive to remesh large, complex geometries with each physical time step, especially if the equations are solved on parallel computers through domain decomposition (the mesh must not only be regenerated, but also re-partitioned and distributed across the parallel machine).

The suitability of the present adjoint-based design methodology has been demonstrated on unstructured meshes with sliding interfaces [29], and it is expected that the extensions to overset meshes or mixing plane approximations will be theoretically straightforward (although significant numerical implementation challenges are also expected to arise, especially on parallel computers). In this section, and in the next chapter containing numerical results, we will focus on rigidly transforming and dynamically deforming grids composed of a single zone.

Rigidly Transforming Grids

If the type of surface motion can be supported by a rigid transformation of the grid (i.e., there is no relative motion between individual grid nodes), then rigid body rotational and translational motion

for a mesh node i with each physical time step can be generally described by

$$\vec{x}_i^{n+1} = \mathfrak{R}(\Delta\vec{\theta}) \vec{r}_i + \Delta\vec{h}, \quad (4.82)$$

where \vec{x}^{n+1} is the updated node location in Cartesian coordinates, $\Delta\vec{h}$ is a vector describing the translation of the nodal coordinates between time steps, $\vec{r}_i = \vec{x}_i^n - \vec{x}_o$ is the position vector pointing from a prescribed motion center for the body, \vec{x}_o , to the point at time level n , and in three dimensions, the rotation matrix, $\mathfrak{R}(\Delta\vec{\theta})$, is given by

$$\mathfrak{R}(\Delta\vec{\theta}) = \begin{bmatrix} \cos\theta_y \cos\theta_z & \sin\theta_x \sin\theta_y \cos\theta_z - \cos\theta_x \sin\theta_z & \cos\theta_x \sin\theta_y \cos\theta_z + \sin\theta_x \sin\theta_z \\ \cos\theta_y \sin\theta_z & \sin\theta_x \sin\theta_y \sin\theta_z + \cos\theta_x \cos\theta_z & \cos\theta_x \sin\theta_y \sin\theta_z - \sin\theta_x \cos\theta_z \\ -\sin\theta_y & \sin\theta_x \cos\theta_y & \cos\theta_x \cos\theta_y \end{bmatrix}, \quad (4.83)$$

with $\Delta\vec{\theta} = \{\theta_x, \theta_y, \theta_z\}^T$ being equal to the change in angular position of the nodal coordinates about a specified rotation center between time t^{n+1} and t^n . Note that this matrix is formed by assuming positive, right-handed rotation first about the x -axis, then the y -axis, and finally the z -axis. The general form of Eqn. 4.82 supports multiple types of motion, including constant rotational or translational rates, pitching, or plunging. With each physical time step, the values of $\Delta\vec{\theta}$ and $\Delta\vec{h}$ are computed, and Eqn. 4.82 is applied at each node of the mesh.

For the results presented below that involve pitching surfaces, the angle of attack as a function of time is given by

$$\alpha(t) = \alpha_o + \alpha_m \sin(\omega t), \quad (4.84)$$

where α_o is the mean angle of attack, α_m is the amplitude of the oscillations, and ω is the angular frequency. The reduced frequency, or $\omega_r = \frac{\omega c}{2v_\infty}$, where c is the chord or characteristic length and v_∞ is the free-stream velocity, is a non-dimensional parameter often specified for consistency between flows. From Eqn. 4.84, $\Delta\vec{\theta}$ between successive time steps can be found.

For prescribed, rigid mesh motion, we can choose the analytic values for the grid velocity (based on the time derivative of the pitching expression above, for instance). For the adjoint problem, the mesh motion must be performed in reverse, and the mesh node locations and velocities that were computed during the direct problem can be written to disk and retrieved with each time step. Note again that the cell volumes will remain fixed for rigid mesh transformations.

Dynamically Deforming Grids

Mesh motion can also be accomplished by first moving the surface boundaries in some specified manner and then deforming the volume mesh to conform to the new surface position with each time step. There are a number of available volumetric grid deformation techniques, including classic spring analogy methods or higher-fidelity methods based on the linear elasticity equations. These techniques for mesh deformation are also used during optimal shape design in order to modify the

volume mesh between major design iterations after an optimizer provides new values for the design variables that control the geometry (surface shape), and they will be described below.

In the case of dynamically deforming grids, the dual control volumes are no longer fixed in size over time and a readily-attainable analytic form for the grid velocities of the interior nodes in the mesh will not be available. This is also the case for non-prescribed motion problems where the integrated forces are used to solve the equations of motion for a body in free-flight or for computing a structural response (aeroelastic response), for instance. For all types of mesh motion, the local grid velocity at node i , \vec{u}_{Ω_i} , can be computed by storing the node coordinates at prior time instances and using a finite differencing approximation that is consistent with the chosen dual time-stepping scheme. For second-order accuracy in time, the mesh velocities are given by

$$\vec{u}_{\Omega_i}{}^{n+1} = \frac{d\vec{x}_i{}^{n+1}}{dt} \approx \frac{3\vec{x}_i{}^{n+1} - 4\vec{x}_i{}^n + \vec{x}_i{}^{n-1}}{2\Delta t}, \quad (4.85)$$

where Δt is the physical time step and $n + 1$ is the current time level. A first-order accurate approximation is similarly expressed as

$$\vec{u}_{\Omega_i}{}^{n+1} \approx \frac{\vec{x}^{n+1} - \vec{x}^n}{\Delta t}. \quad (4.86)$$

Geometric Conservation Law

When computing unsteady flows on dynamic meshes with the ALE form of the equations (especially with dynamically deforming meshes), a Geometric Conservation Law (GCL) should be satisfied. First introduced by Thomas and Lombard [120], it has been shown mathematically and through numerical experiment [9, 69, 34] that satisfying the GCL can improve the accuracy and stability of the chosen scheme.

The basic idea is that the mesh motion and the grid velocity terms should not create spurious oscillations in the solution as the ALE equations are integrated in time. In other words, a uniform flow field should be preserved under the motion or deformation of the grid cells comprising the domain. The GCL can be derived by introducing a spatially and temporally constant flow state into the flow equations. For instance, assuming that U^* represents a constant flow state and starting from the unsteady Euler equations for simplicity, the solution can be introduced during the derivation of

the semi-discretized, finite volume form of the equations [67]:

$$\begin{aligned}
& \frac{\partial U}{\partial t} + \nabla \cdot \vec{F}_{ale}^c = 0 \\
& \frac{\partial U}{\partial t} + \nabla \cdot \vec{F}^c - \nabla \cdot (U \vec{u}_\Omega) = 0 \\
& \int_{\Omega_i(t)} \left(\frac{\partial U}{\partial t} + \nabla \cdot \vec{F}^c - \nabla \cdot (U \vec{u}_\Omega) \right) d\Omega = 0 \\
& \frac{d}{dt} \int_{\Omega_i(t)} U d\Omega + \int_{\partial\Omega_i(t)} \vec{F}^c \cdot \vec{n} ds - \int_{\partial\Omega_i(t)} (U \vec{u}_\Omega) \cdot \vec{n} ds = 0 \\
& U^* \frac{d}{dt} \int_{\Omega_i(t)} d\Omega + \vec{F}^c(U^*) \cdot \int_{\partial\Omega_i(t)} \vec{n} ds - U^* \int_{\partial\Omega_i(t)} \vec{u}_\Omega \cdot \vec{n} ds = 0 \\
& \frac{d|\Omega_i|}{dt} - \int_{\partial\Omega_i(t)} \vec{u}_\Omega \cdot \vec{n} ds = 0 \\
& \frac{d|\Omega_i|}{dt} - \sum_{j \in \mathcal{N}(i)} (\vec{u}_{\Omega_{ij}} \cdot \vec{n}_{ij}) \Delta S_{ij} = 0 \\
& \frac{d|\Omega_i|}{dt} - R_{gcl_i}(\vec{u}_\Omega) = 0, \tag{4.87}
\end{aligned}$$

where in going from the fourth to the fifth line, we have imposed the uniform flow state, we have used the identity $\int_{\partial\Omega_i(t)} \vec{n} ds = \vec{0}$, $\mathcal{N}(i)$ are the neighboring nodes to node i , and $\vec{u}_{\Omega_{ij}} = (\vec{u}_{\Omega_i} + \vec{u}_{\Omega_j})/2$ is the grid velocity at the dual cell face as an average of the grid velocities at nodes i and j . Eqn. 4.87 represents a GCL, and in words, it states that the change in the volume of a cell with time must be balanced by the volume swept out by the moving faces of the cell.

To arrive at a fully discrete version of the GCL, it should be derived in the context of the same time integration scheme that was chosen for the unsteady fluid equations. If the flow equations are solved with a second-order backward difference formula, the Eqn. 4.87 becomes

$$\frac{3}{2\Delta t} |\Omega_i|^{n+1} - \frac{2}{\Delta t} |\Omega_i|^n + \frac{1}{2\Delta t} |\Omega_i|^{n-1} - R_{gcl_i}(\vec{u}_\Omega^{n+1}) = 0. \tag{4.88}$$

A straightforward technique for the numerical implementation of this GCL [85, 10] involves multiplying Eqn 4.88 by the flow solution at time level n and rearranging terms as

$$-\frac{2}{\Delta t} |\Omega_i|^n U^n = R_{gcl_i}(\vec{u}_\Omega^{n+1}) U^n - \frac{3}{2\Delta t} |\Omega_i|^{n+1} U^n - \frac{1}{2\Delta t} |\Omega_i|^{n-1} U^n. \tag{4.89}$$

Eqn. 4.89 can then be substituted directly into Eqn. 4.76 and rearranged to give an augmented version of the dual time-stepping scheme expressed as

$$R_{gcl_i}(\vec{u}_\Omega^{n+1}) U^n + \frac{3}{2\Delta t} |\Omega_i|^{n+1} (U_i^{n+1} - U^n) + \frac{1}{2\Delta t} |\Omega_i|^{n-1} (U_i^{n-1} - U^n) + R_i(U^{n+1}) = 0, \tag{4.90}$$

where the fixed portion of the dual time-stepping source term now includes the GCL. The GCL

residual, $R_{gcl_i}(\vec{u}_\Omega^{n+1})$, is evaluated as given in Eqn. 4.87 while using the grid velocities (either analytically defined or from a finite difference approximation based on the change in the grid coordinates in time), edge normals, and face areas at time level $n + 1$. For completeness, a first-order accurate in time version of Eqn. 4.90 is similarly given by

$$R_{gcl_i}(\vec{u}_\Omega^{n+1})U^n + \frac{1}{\Delta t}|\Omega_i|^{n+1}(U_i^{n+1} - U^n) + R_i(U^{n+1}) = 0, \quad (4.91)$$

4.3 Design Variable Definition

The continuous adjoint derivation above presents a method for computing the variation of an objective function with respect to infinitesimal surface shape deformations in the direction of the local surface normal at points on the surface. However, this information can be used in any number of manners that are consistent with the intent of the designer for changing the shape of the underlying geometry. Adjoint sensitivities for a particular functional are typically projected into a design space composed of suitable variables, as will be described below, and a survey of many available techniques is given by Samareh [110]. In this framework, geometric sensitivities, i.e., the change in the surface shape or coordinates with respect to a change in the design variable, are also required to compute the sensitivity of the functional to the design variable using the chain rule.

For engineering applications, the geometry is often defined by Computer-aided design (CAD) models that may have built-in parametric design variables of interest for the designer, such as the twist or sweep of an aircraft wing. In these situations, a designer may wish to drive a shape optimization problem using the geometric representation and parameterization within the CAD model by importing the changes in the variables prescribed by an optimizer and exporting the new surface shape within a design loop. Unfortunately, CAD-based optimization can often limit the design space to a set of predefined variables that may not include the optimal shape, or it may require rebuilding complicated CAD models if only legacy geometry or meshes are available.

Other strategies exist that are CAD-free and operate directly on the discrete surface geometry. While it is possible to use each surface node in the computational mesh as a design variable capable of deformation in conjunction with gradient smoothing [63], for instance, this approach is not pursued here. Instead, we will compute the surface sensitivities $\frac{\partial \mathcal{J}}{\partial S}$ at each mesh node i on the surface and project this information into a design space made up of a smaller set of design variables (possibly a complete basis). These variables might take the form of analytic functions, or bumps, that define modifications in shape relative to the original geometry, or more advanced variables that rely on computer graphics algorithms for deforming discrete geometry while retaining its topology. One drawback of these approaches is that they require one to transfer the discrete geometry back into a CAD representation if the goal is to manufacture the final design, but some recent tools are alleviating these conversion issues.

4.3.1 Gradient Projection using Surface Sensitivities

To find the gradient of a chosen objective function J with respect to a set of design variables α_j using the continuous adjoint presented in this work, consider first the final result from the continuous adjoint derivation for the variation of the functional (we assume a steady problem here for simplicity):

$$\delta\mathcal{J} = \int_S \left\{ \frac{\partial\mathcal{J}}{\partial S} \right\} \delta S ds. \quad (4.92)$$

After introducing a perturbation for a particular design variable, we can approximate the gradient of the objective function by evaluating the surface integral as

$$\frac{\delta\mathcal{J}}{\delta\alpha_j} = \int_S \left\{ \frac{\partial\mathcal{J}}{\partial S} \right\} \frac{\delta S}{\delta\alpha_j} ds \approx \sum_{i \in \mathcal{N}(S)} \left\{ \frac{\partial\mathcal{J}}{\partial S} \right\}_i \frac{\vec{n}_i \cdot \Delta\vec{x}_i}{\Delta\alpha_j} \Delta S_i, \quad (4.93)$$

where S is the surface being designed, $\mathcal{N}(S)$ represents the set of mesh nodes on the S , $\left\{ \frac{\partial\mathcal{J}}{\partial S} \right\}_i$ is the value of the surface sensitivity from the continuous adjoint at node i , \vec{n}_i is the local unit normal at node i , $\Delta\alpha_j$ is a perturbation in the design variable (a bump function, for instance), $\Delta\vec{x}_i$ is the resulting change in the Cartesian coordinates of node i after applying the design variable perturbation, and ΔS_i is the area of the surface control volume surrounding node i . While $\left\{ \frac{\partial\mathcal{J}}{\partial S} \right\}_i$ is given by the adjoint surface sensitivity, the remaining terms in Eqn. 4.93 are geometric in nature and typically evaluated in a finite difference manner by imposing a small deformation in each design variable in order to find the local change in the nodal coordinates, $\Delta\vec{x}_i$. Two choices of design variables were used in this work, and they are briefly described below.

4.3.2 Bump Function Design Variables

A number of different types of bump functions exist that enable the efficient parameterization of some geometries, such as airfoils. In the 2D airfoil shape optimizations that follow, Hicks-Henne bump functions are employed [47], which can be added to the original airfoil geometry to modify the shape. The Hicks-Henne function with maximum at point x_n is given by

$$f_n(x) = \sin^3(\pi x^{e_n}), \quad e_n = \frac{\log(0.5)}{\log(x_n)}, \quad x \in [0, 1], \quad (4.94)$$

so that the total deformation of the airfoil surface from its original shape at an x location along the chord can be computed as the sum of the smooth shape functions, or $\Delta y = \sum_{n=1}^N \delta_n f_n(x)$, with N being the number of bump functions and δ_n the bump variable step size. These functions are applied separately to the upper and lower surfaces. When $\delta_n = 0$ for all bumps, the original airfoil shape is recovered. Fig. 4.3 plots a set of equally-spaced Hicks-Henne shape functions f_n . While bumps functions, like those of Hicks-Henne, are effective for some specific geometries, they can be difficult to apply in a straightforward manner to complex geometries where a more general, complete parameterization may be required.

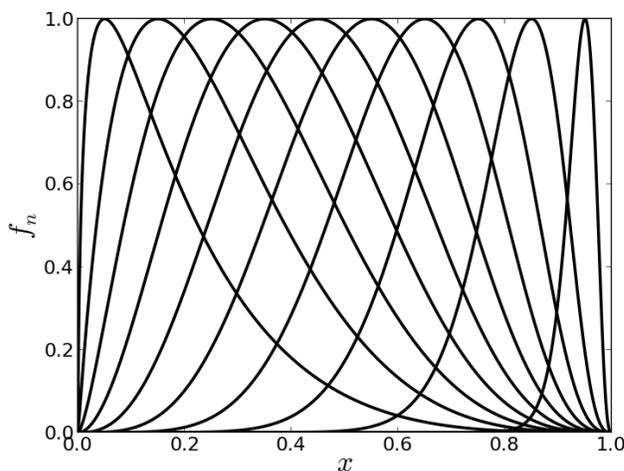


Figure 4.3: Hicks-Henne shape functions for a set of equally spaced bumps.

4.3.3 Free-Form Deformation Variables

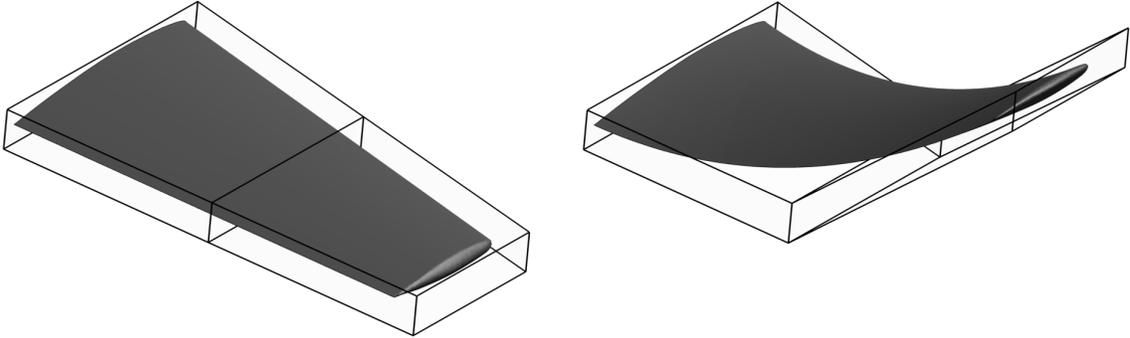
A Free-Form Deformation (FFD) strategy has also been adopted in both two and three dimensions, which has become a popular geometry parameterization technique for aerodynamic shape design [111, 65, 1]. In FFD, an initial box encapsulating the object (rotor blade, wing, fuselage, etc.) to be redesigned is parameterized as a Bézier solid. A set of control points are defined on the surface of the box, the number of which depends on the order of the chosen Bernstein polynomials. The solid box is parameterized by the following expression:

$$X(u, v, w) = \sum_{i,j,k=0}^{l,m,n} P_{i,j,k} B_i^l(u) B_j^m(v) B_k^n(w), \quad (4.95)$$

where $u, v, w \in [0, 1]$, and B^i is the Bernstein polynomial of order i . The Cartesian coordinates of the points on the surface of the object are then transformed into parametric coordinates within the Bézier box.

The control points of the box become design variables, as they control the shape of the solid, and thus the shape of the surface grid inside. The box enclosing the geometry is then deformed by modifying its control points, with all the points inside the box inheriting a smooth deformation. Once the deformation has been applied, the new Cartesian coordinates of the object of interest can be recovered by simply evaluating the mapping inherent in Eqn. 4.95. An example of FFD control point deformation to a wing geometry appears in Fig. 4.4.

In 3D, each control point can move in the x, y , and z directions, and it is important to mention that the movement of the FFD control points can be coordinated in order to create other “engineering” design variables, such as changes to thickness, twist, camber, or sweep, to name a few. Smaller FFD boxes can also be nested within larger ones to help control the granularity of the applied shape



(a) Original wing surface (grey) and FFD bounding box with control points at the intersection of the black lines. (b) Deformed wing surface after displacement of the FFD control points near the wing tip.

Figure 4.4: An example of the FFD technique applied to the ONERA M6 wing.

changes.

4.4 Mesh Deformation

A variety of techniques exist for deforming volumetric grids given the displacements of the boundary nodes of a particular domain, and these techniques are often used both to deform grids during the simulation of unsteady flows on dynamic meshes and also between optimal shape design cycles after perturbing the surface shape. Many of the available grid deformation strategies are motivated by the treatment of the computational mesh as a solid body that can elastically deform as governed by the equations of solid mechanics.

4.4.1 Spring Analogy

A particularly attractive option for meshes that do not contain high aspect ratio cells (for solving the Euler equations, for instance) is based upon a classic spring analogy [9, 22, 13]. Once the boundary displacements have been computed, the rest of vertices of the unstructured mesh can be deformed by defining a stiffness matrix k_{ij} that connects the two ends of a single bar (mesh edge). Equilibrium of forces is then imposed at each mesh node

$$\left(\sum_{j \in \mathcal{N}_i} k_{ij} \vec{e}_{ij} \vec{e}_{ij}^T \right) \vec{u}_i = \sum_{j \in \mathcal{N}_i} k_{ij} \vec{e}_{ij} \vec{e}_{ij}^T \vec{u}_j, \quad (4.96)$$

where the displacement \vec{u}_i is unknown and is computed as a function of the known surface displacements \vec{u}_j , \mathcal{N}_i is the set of neighboring points to node i , and \vec{e}_{ij} the unit vector in the direction connecting both points. The system of linear equations can then be solved with a suitable iterative

method, such as the conjugate gradient algorithm with Jacobi preconditioning, for example.

4.4.2 Linear Elasticity

For meshes with high aspect ratio cells that might be suitable for boundary layers in viscous flow, a grid deformation technique based on the linear elasticity equations [64, 91, 25] may help preserve grid quality near solid surfaces where methods based on the spring analogy often fail, resulting in negative cell volumes. In three dimensions, linear elasticity governs small displacements, $\vec{u} = (u_1, u_2, u_3)^T$, of an elastic solid subject to body forces, \vec{f} , and surface tractions,

$$\begin{cases} \mathcal{M}(\vec{u}) = \nabla \cdot \bar{\sigma} - \vec{f} = 0, & \text{in } \Omega, \\ \vec{u} = \vec{g}, & \text{on } \Gamma, \end{cases} \quad (4.97)$$

where $\bar{\sigma}$ is the stress tensor. The stress, $\bar{\sigma}$, and strain, $\bar{\epsilon}$, tensors can be related using the following constitutive equation

$$\bar{\sigma} = \lambda Tr(\bar{\epsilon})\bar{I} + 2\mu\bar{\epsilon}, \quad (4.98)$$

with the strain-displacement relation written as

$$\bar{\epsilon} = \frac{1}{2}(\nabla\vec{u} + \nabla\vec{u}^T), \quad (4.99)$$

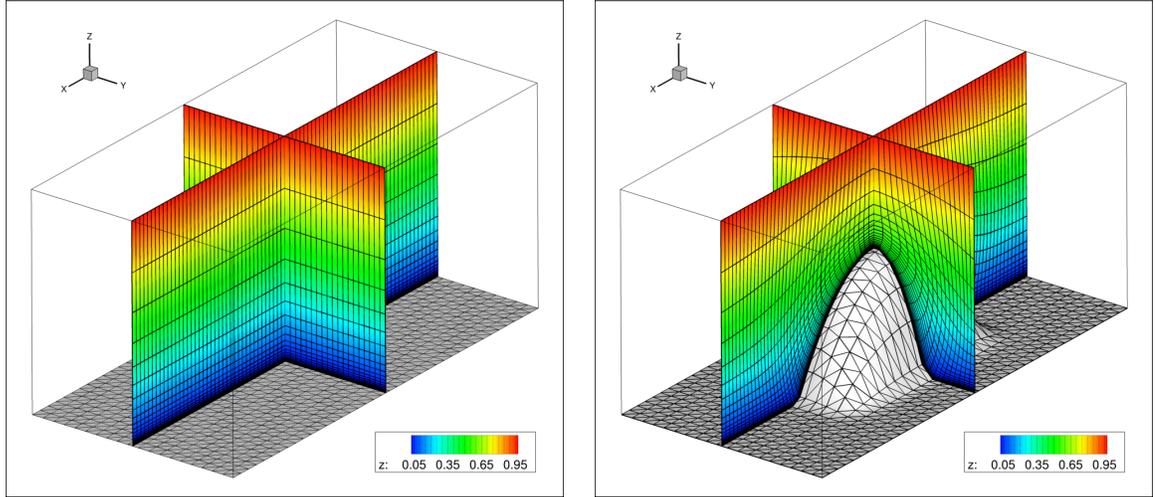
where Tr is the trace, λ and μ are the Lamé constants given by

$$\lambda = \frac{\nu E}{(1 + \nu)(1 - 2\nu)}, \quad \mu = \frac{E}{2(1 + \nu)}, \quad (4.100)$$

and ν is Poisson's ratio, and E is the Young's modulus. Poisson's ratio, ν , describes how a material compresses in the lateral direction as it extends in the axial direction. E is a measure of the stiffness of a material. Each element of the mesh is treated as an elastic solid and, by allowing for variable E throughout the mesh, can have its own rigidity. By choosing a value of E that is inversely proportional to the volume of the element (or, small mesh cells near viscous walls will transform more rigidly than larger cells, thus helping to preserve mesh quality in these regions where the flow solution is particularly sensitive. Fig. 4.5 presents an example deformation for a mesh with high aspect ratio cells.

The linear elasticity equations are discretized using the Finite Element Method (FEM) with a standard Galerkin approximation, and the computed boundary displacements due to changes in the design variables (or dynamic surface motions) are applied as a Dirichlet boundary condition. The system of equations is solved iteratively by a preconditioned GMRES method. For large displacements, it may be required to solve the system in increments, i.e., the linear elasticity equations are solved multiple times as the domain boundaries are marched in increments from their original to final locations.

A Finite Element Method (FEM) is also available to numerically evaluate the linear elasticity



(a) Original domain showing two slices of a prism mesh with high aspect ratio cells near the lower boundary.

(b) Deformed grid after applying an FFD control point deformation to the lower boundary surface.

Figure 4.5: An example of volume mesh deformation using the linear elasticity equations.

equations. Finite element methods are based upon approximations to a variational formulation of the problem. A variational formulation requires the introduction of a space of trial functions, $\mathcal{T} = \{V(t, \vec{x})\}$, and a space of weighting functions, $\mathcal{W} = \{W(t, \vec{x})\}$. The problem consists of finding $V(t, \vec{x})$ in \mathcal{T} satisfying the problem boundary conditions, such that

$$\int_{\Omega} W^T (\nabla^2 V) d\Omega = 0. \quad (4.101)$$

To produce an approximate solution to the variational problem, a grid of finite elements is constructed on the domain, Ω . It will be assumed that the discretization employs p nodes. Finite-dimensional subspaces $\mathcal{T}^{(p)}$ and $\mathcal{W}^{(p)}$ of the trial and weighting function spaces, respectively, are defined by

$$\mathcal{T}^{(p)} = \left\{ V^{(p)}(\vec{x}) \mid V^{(p)} = \sum_{J=1}^p V_J N_J(\vec{x}) \right\}, \quad \mathcal{W}^{(p)} = \left\{ W^{(p)}(\vec{x}) \mid W^{(p)} = \sum_{J=1}^p a_J N_J(\vec{x}) \right\}, \quad (4.102)$$

where V_J is the value of $V^{(p)}$ at node J . On the other hand, a_1, a_2, \dots, a_p are constant and $N_J(\vec{x})$ is the piecewise linear trial function associated with node J . We now apply the finite element approximation by discretizing the domain of the problem into elements and introducing functions that interpolate the solution over nodes that compose the elements. The Galerkin approximation is determined by applying the variational formulation of Eq. 4.101 in the following form: find $V^{(p)}$ in

$\mathcal{T}^{(p)}$, satisfying the problem boundary conditions, such that

$$\int_{\Omega} N_I^T (\nabla^2 V) d\Omega = 0, \quad (4.103)$$

for $I = 1, 2, \dots, p$. The form assumed for $V^{(p)}$ in Eq. 4.102 can now be inserted into the left hand side of Eq. 4.103 and the result can be written as

$$\int_{\Omega} N_I^T \left(\sum_{J=1}^p V_J \nabla^2 N_J \right) d\Omega = \sum_{J=1}^p V_J \left(\int_{\Omega} N_I^T \nabla^2 N_J d\Omega \right) = 0. \quad (4.104)$$

Applying the divergence theorem, one has

$$\sum_{J=1}^p V_J \left(\int_{\Gamma} N_I^T (\nabla N_J \cdot \vec{\nu}) d\Gamma - \int_{\Omega} \nabla N_I^T \cdot \nabla N_J d\Omega \right) = 0, \quad (4.105)$$

where $\vec{\nu}$ is the outward unit normal associated with the control volume surface and the boundary integral disappears unless we are computing a boundary element with non-homogeneous Neumann conditions (I is an exterior node). The result at a typical interior node I is

$$\sum_{E \in I} \sum_{J \in E} V_J \left(\int_{\Omega_E} \nabla N_I^T \cdot \nabla N_J d\Omega \right) = 0, \quad (4.106)$$

where the first summation extends over the elements E in the numerical grid that contain node I and the second summation extends over nodes J of the elements E . Ω_E is the portion of Ω which is represented by element E .

4.5 Optimization Framework

Scripts written in the Python programming language are used to automate execution of the SU2 suite components, especially for performing shape optimization. The optimization results presented in this work make use of the *SciPy* library (<http://www.scipy.org>), a well-established, open-source software package for mathematics, science, and engineering. The *SciPy* library provides many user-friendly and efficient numerical routines for the solution of nonlinear constrained optimization problems, such as conjugate gradient, Quasi-Newton, or sequential least-squares programming algorithms. At each design iteration, the *SciPy* routines require as input only the values and gradients of the objective functions, computed by means of our continuous adjoint approach, as well as the set of any chosen constraints. The gradient for any flow variable constraints (e.g., lift, drag, etc.) can be computed by solving an additional adjoint problem for each constraint.

Chapter 5

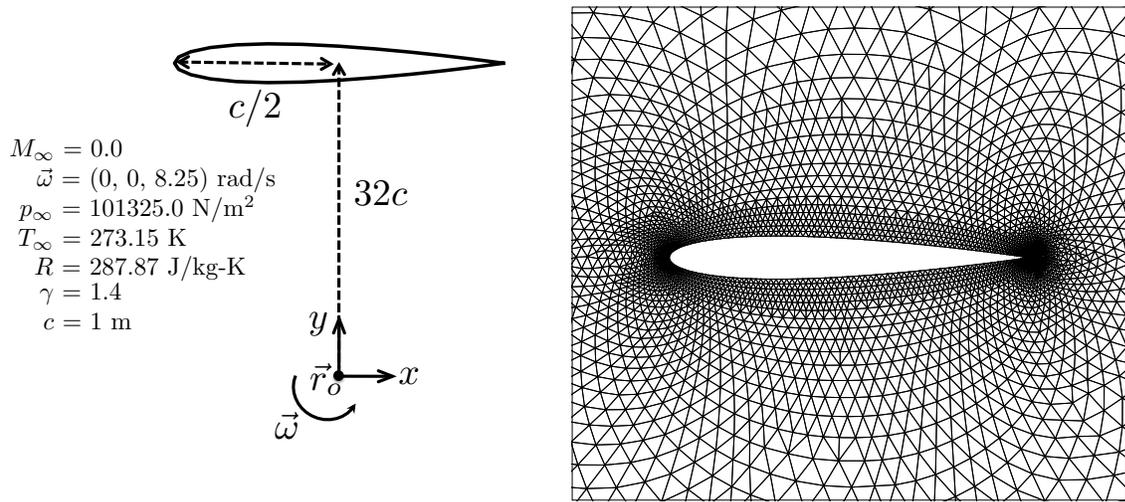
Optimal Shape Design Applications

This chapter presents results for two- and three-dimensional optimal shape design demonstrations for a range of problems governed by the non-inertial or unsteady flow equations [28, 29, 30, 31]. For a number of the examples, studies of the gradient accuracy and verification against finite differencing gradients are provided. The final design case, a pitching wing in turbulent flow, is meant to be representative of a viscous shape optimization problem of industrial interest through the inclusion of the required mathematical, numerical, and geometrical complexity.

5.1 Rotating Airfoil in Inviscid Flow

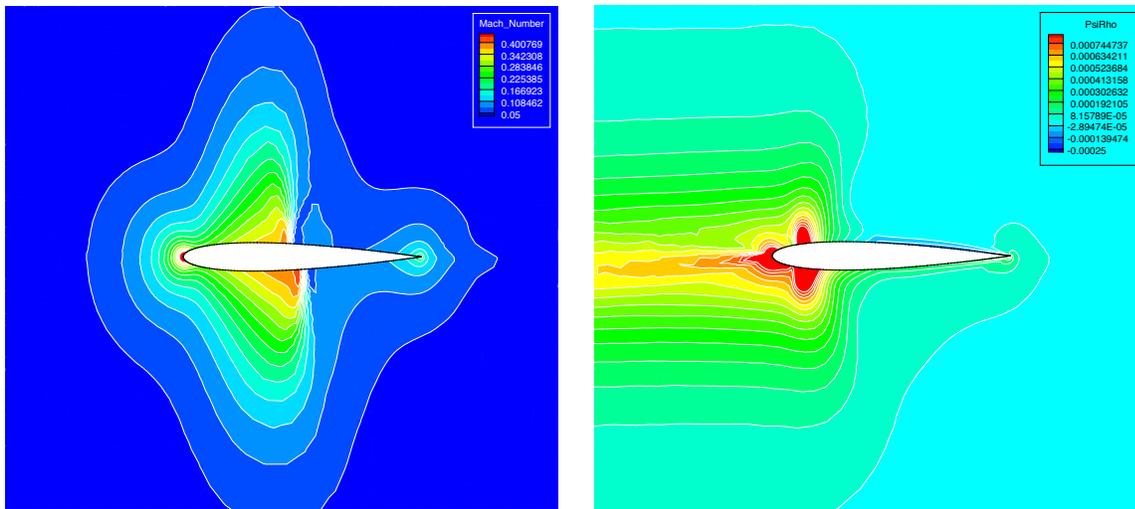
As a verification test for the gradient information obtained by the continuous adjoint formulation, a numerical experiment was devised for a NACA 0012 airfoil rotating in still air ($M_\infty = 0$), which can be solved using the Euler equations expressed in a rotating frame. The flow is two-dimensional in the x - y plane with rotation out of the page in the z -direction. The specific conditions for the problem, and in particular the angular velocity of the airfoil rotation, were chosen such that the flow was transonic and shocks appeared on the upper and lower airfoil surfaces. The goals of the test case are two-fold: to verify the gradient of the coefficient of drag C_d with respect to a set of Hicks-Henne design variables obtained from the continuous adjoint formulation against finite differencing, and to perform an airfoil shape optimization for minimizing C_d . The details of the numerical experiment and the unstructured mesh appear in Fig. 5.1. The mesh consisted of 10,216 triangular elements, 5,233 nodes, 200 edges along the airfoil, and 50 edges along the far-field boundary.

Fig. 5.1 shows the absolute Mach number contours around the airfoil. In the inertial frame, the flow appears entirely subsonic as the air is pushed out of the path of the rotating airfoil. However, there are clear shock structures on both the upper and lower surface, and when the velocity due to rotation is taken into account to form the relative velocity, the local Mach number near the airfoil



(a) Conditions for the rotating airfoil problem.

(b) Zoom view of the unstructured mesh near the airfoil.



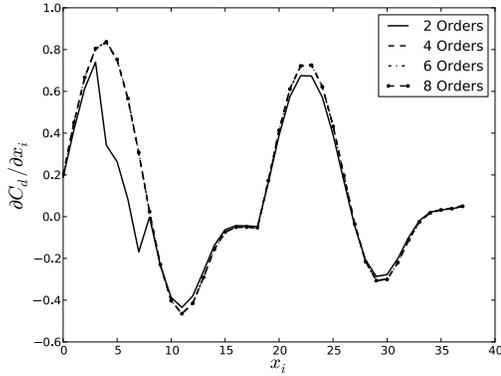
(c) Absolute Mach number contours.

(d) Adjoint density contours.

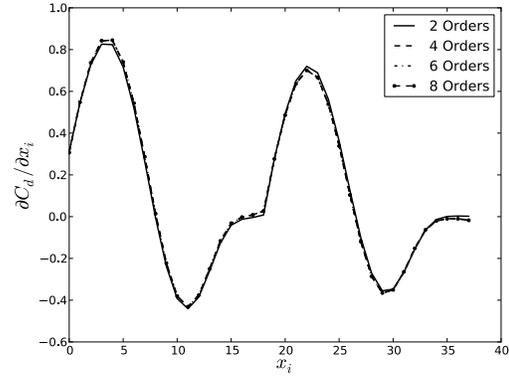
Figure 5.1: Details for the 2D rotating airfoil numerical experiment, the computational mesh, and solutions for the baseline geometry.

surface is supersonic in some locations. Fig. 5.1 also presents contours for the adjoint density ψ_ρ near the surface. Note the strong features near the nose and sonic points in the adjoint solution.

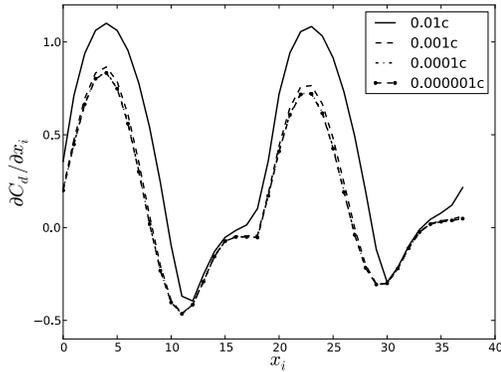
In order to verify the accuracy of the gradient information obtained by the continuous adjoint formulation, Hicks-Henne bump functions were chosen as design variables along the upper and lower surfaces of the NACA 0012. The numbering for the bumps starts from 0 at the trailing edge on the lower surface and increases as one wraps around the leading edge and upper surface of the airfoil. A comparison was then made between the gradient of C_d with respect to the design variables resulting



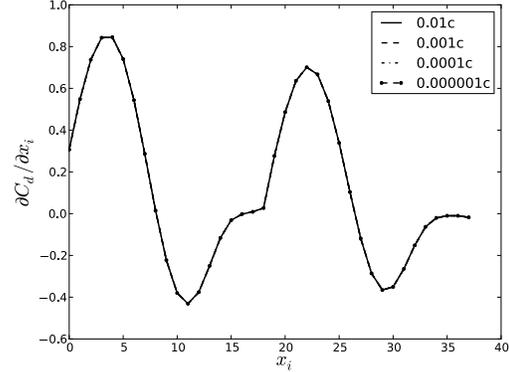
(a) C_d gradients from finite differencing based upon various levels of convergence in the density residual (order of magnitude reduction). Very low levels of convergence affect the gradient accuracy. The step size for each case was 0.0001c.



(b) C_d gradients for the continuous adjoint based upon various levels of convergence in the density and adjoint density residuals (order of magnitude reduction). The results here show that the adjoint gradients are fairly insensitive to convergence level. The gradient projection step size for each case was 0.0001c.



(c) C_d gradients from finite differencing with various step sizes. It is clear that the step size impacts the accuracy of the gradient information, and that a sufficiently small step must be taken. Little difference is apparent between 0.0001c and 0.000001c. All cases were converged 8 orders of magnitude in the density residual.



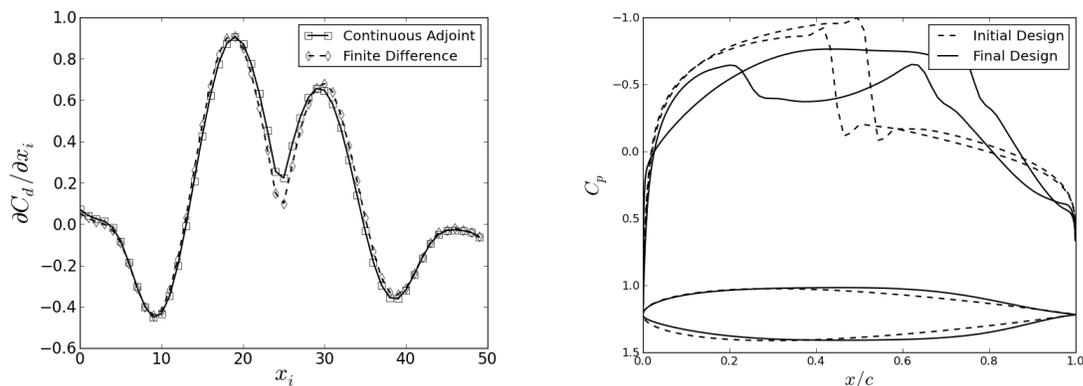
(d) C_d gradients for the continuous adjoint with different gradient projection step sizes for the Hicks-Henne bump deformations. As expected, there is no dependence on the step size for the adjoint, as the surface sensitivities are computed independently of the geometric sensitivities (gradient projection). The solutions were converged 8 orders of magnitude in the density and adjoint density residuals.

Figure 5.2: Comparison studies between the continuous adjoint and finite differencing for the gradient of C_d . A set of 38 Hicks-Henne bump function variables (x_i) are along the x -axis.

from the continuous adjoint approach and a finite differencing approach using small step sizes for the bump deformations. For this problem, the force coefficients (C_l , C_d , and C_p) were computed using ρ_∞ , p_∞ , and the velocity due to rotation at the nose of the airfoil. The gradients compare very favorably, although there are slight differences between the adjoint and finite differencing, as

seen in Fig. 5.3 for a set of 50 bump functions.

Further studies were performed to explore the sensitivity of the gradients to both the step size of the bump deformations and the level of convergence attained by the solver for both the flow and adjoint problems. These comparisons are given in Fig. 5.2 for a set of 38 bump functions. Similar to gradient accuracy results shown by Kim et al. [66], the finite difference gradients are quite sensitive to the chosen step size and level of solver convergence, whereas the adjoint gradients are largely insensitive to these parameters.



(a) Direct comparison of the gradients obtained by the continuous adjoint and finite differencing. Direct and adjoint solutions were converged 8 orders of magnitude in the density residual and adjoint density residual, respectively. The step size for the bump deformations was $10^{-6}c$.

(b) C_p and profile shape comparison for the baseline rotating NACA 0012 and the minimum drag airfoil. The optimizer has effectively removed the shocks by making the airfoil thinner in the forward half and thicker in the aft half.

Figure 5.3: Gradient verification and a comparison of the baseline and minimum drag airfoil designs.

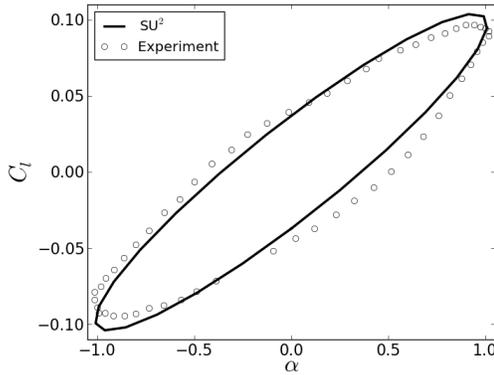
Finally, a redesign of the rotating airfoil was performed using the gradient information obtained from the adjoint formulation. The specific shape optimization problem was for drag minimization with a geometric constraint that the maximum thickness of the airfoil remain larger than $0.12c$. A set of 50 Hicks-Henne bumps were chosen as the design variables. Upon completion, the C_d was successfully reduced from 0.00725 down to 0.00009, which is a 98.7 % reduction, and the maximum thickness of the final airfoil design met the constraint at a value of $0.122c$. C_p distributions as well as the profile shapes of the initial and final designs are compared in Fig. 5.3. The optimization process has eliminated the shocks on the upper and lower surfaces that originally appeared on the rotating NACA 0012 by making the airfoil thinner in the forward half and thicker in the aft half while also satisfying the thickness constraint.

5.2 Pitching Airfoil in Inviscid Flow

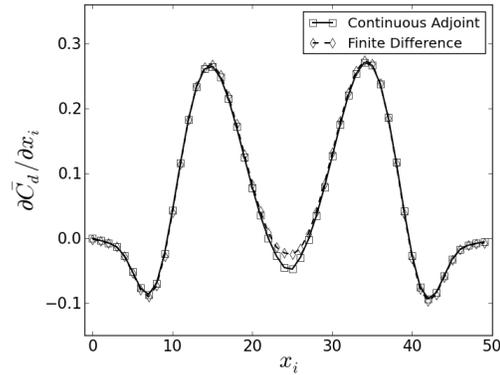
For validation of the unsteady Euler equations in ALE form, a comparison of numerical results was made against the well-known CT6 data set of Davis [21]. The physical experiment measured the

unsteady performance for the NACA 64A010 airfoil while pitching about the quarter-chord point. The particular experimental case of interest studied pitching motion with a reduced frequency w_r of 0.202, $M_\infty = 0.796$, a mean angle of attack of 0 degrees, and a maximum pitch angle of 1.01 degrees.

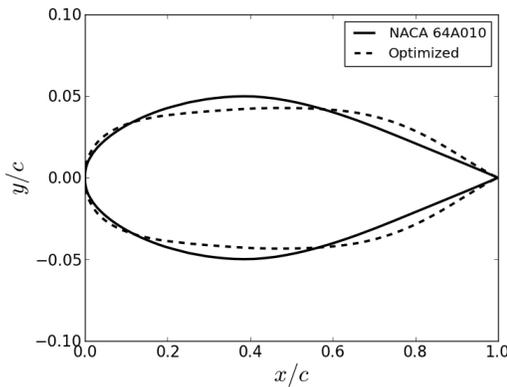
A baseline unstructured mesh consisting of 16,937 triangular elements, 8,606 nodes, 200 edges along the airfoil, and 75 edges along the far-field boundary was constructed, as seen in Fig. 5.5. All simulations of the pitching airfoil were time-accurate (dual time-stepping) with 25 time steps per period for a total of 10 periods. The equations were relaxed in pseudo-time for each physical time step until a reduction of 3 orders of magnitude in the density residual was achieved. This afforded adequate time for transient effects to wash away and for well-resolved time-averaged behavior in the periodic steady state.



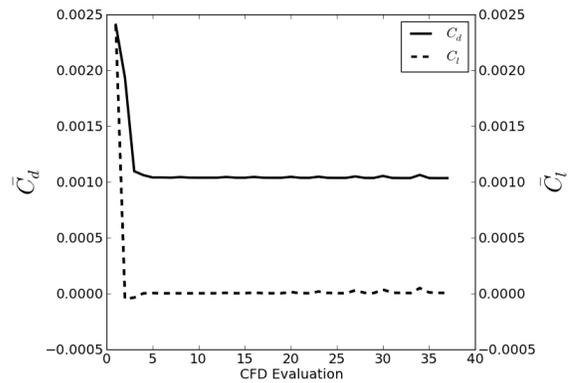
(a) Coefficient of lift versus angle of attack (degrees) compared against experimental data. Note that nonlinear effects cause lift hysteresis.



(b) Direct comparison of the time-averaged drag gradients with respect to 50 Hicks-Henne bumps as obtained by the continuous adjoint and finite differencing.



(c) Profile shape comparison for the initial NACA 64A010 and the minimum time-averaged drag airfoil.



(d) Optimization history of a time-averaged drag minimization for a pitching airfoil with a time-averaged lift constraint.

Figure 5.4: Verification, validation, and optimization results for a pitching NACA 64A010.

Fig. 5.4 shows a comparison of the lift coefficient versus angle of attack between simulation and experiment during the final period of oscillation. In physical time, the curve is traversed in a counterclockwise fashion. The flow around the airfoil features two strong shocks on the upper and lower surfaces. Note that nonlinear behavior corresponding to moving shock waves during pitching results in a hysteresis effect. The numerical results agree well with experimentally measured values and also compare favorably with other inviscid results.

In order to verify the accuracy of the gradient information obtained via the inviscid unsteady adjoint, 50 Hicks-Henne bump functions were chosen as design variables. The bumps were equally spaced along the upper and lower surfaces of the NACA 64A010 (25 bumps each on the upper and lower surfaces). The numbering for the bumps starts from 0 at the trailing edge on the lower surface and increases as one wraps around the leading edge and upper surface of the airfoil. After solving the adjoint equations using the stored solution data from the numerical experiment performed above, we can compare the time-averaged drag gradients with respect to the Hicks-Henne design variables ($\frac{\partial \bar{C}_d}{\partial x_i}$) as calculated using both the continuous adjoint and a finite differencing approach with small step sizes for the bump deformations ($10^{-6}c$). As seen in Fig. 5.4, the gradients exhibit excellent agreement.

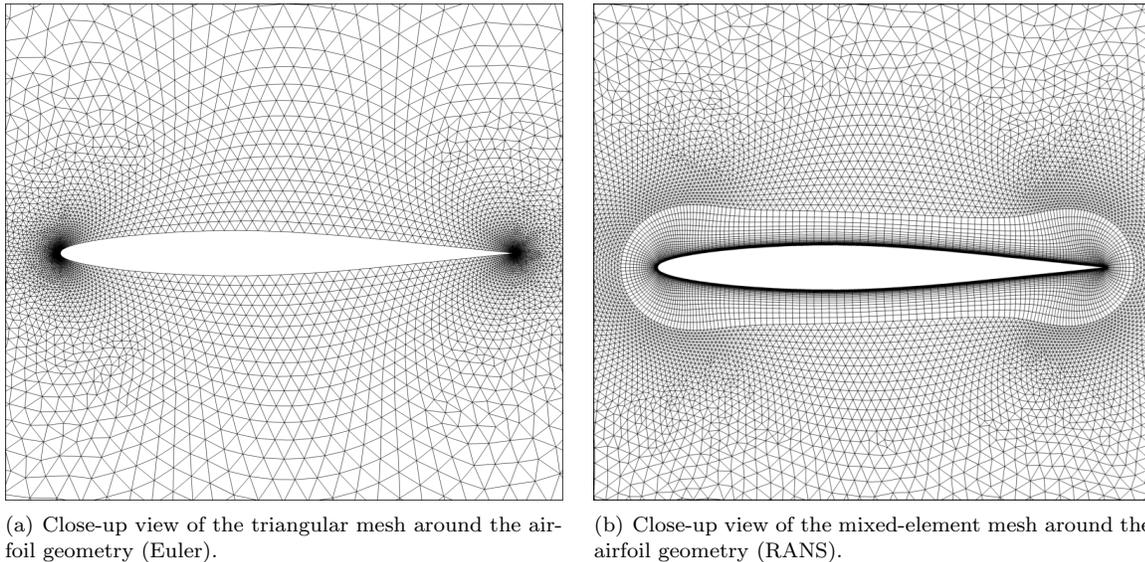


Figure 5.5: Numerical grids for the pitching NACA 64A010 calculations.

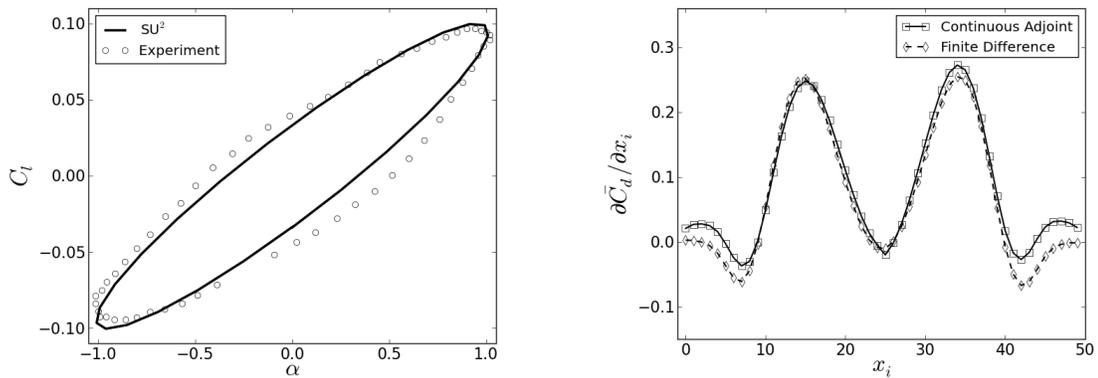
A redesign of the pitching NACA 64A010 airfoil in inviscid flow was performed using the gradient information obtained from the unsteady adjoint formulation. The specific shape optimization problem was a constrained time-averaged drag minimization using the 50 Hicks-Henne variables from the gradient verification. The time-averaged coefficient of lift \bar{C}_l was constrained to be equal to zero, and a separate, geometric constraint was applied to fix the internal area of the airfoil to its original value. After 36 function evaluations, \bar{C}_d was successfully reduced by 57.0 % while satisfying both

of the constraints. Profile shapes of the baseline and final designs are presented in Fig. 5.4, along with the optimization history for the \bar{C}_l and \bar{C}_d values. The shocks have been removed from the design by a thinning of the profile shape near the mid-chord, while maintaining a constant airfoil area by increasing the thickness near the trailing edge. The final design also remains symmetric, which results from the constraint on \bar{C}_l and the imposed mean pitching angle of zero.

5.3 Pitching Airfoil in Turbulent Flow

In order to validate the implementation of the unsteady RANS equations in ALE form and to verify the unsteady viscous adjoint, the NACA 64A010 test case was again studied in turbulent flow. All of the details for the flow simulation remain the same as for the inviscid case above, with the addition of specifying a Reynolds number of 12.5 million. A mixed-element, unstructured mesh was constructed (see Fig. 5.5) that consisted of 22,904 triangular elements, 12,500 quadrilaterals near the airfoil surface to capture the boundary layer, 24,111 nodes in total, 250 edges along the airfoil, and 68 edges along the far-field boundary. The spacing at the wall was chosen to achieve a y^+ value less than 1.

The unsteady RANS equations with the Spalart-Allmaras turbulence model were solved for the flow around the pitching airfoil, and a comparison was again made against the CT6 data set of Davis [21]. The viscous results show slightly better agreement in the lift over the inviscid results, as seen in Fig. 5.6. The flow field again features strong shocks on the upper and lower surfaces.



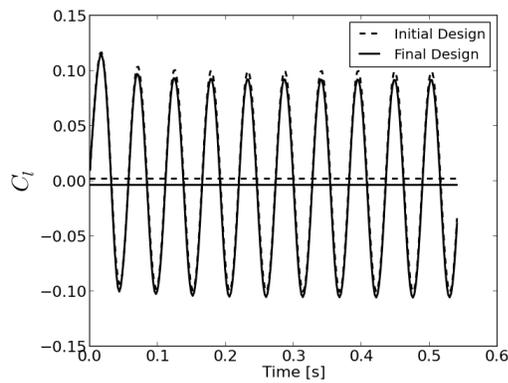
(a) Coefficient of lift versus angle of attack (degrees) compared against experimental data. Note that nonlinear effects cause lift hysteresis.

(b) Comparison of the time-averaged drag gradients with respect to 50 Hicks-Henne bumps as obtained by the continuous adjoint and finite differencing.

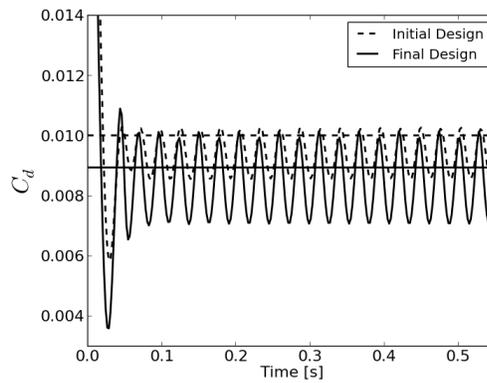
Figure 5.6: Numerical results for a pitching NACA 64A010 in turbulent flow.

50 Hicks-Henne bump variables were chosen as design variables and used to compute gradients, and the bumps were equally spaced along the upper and lower surfaces of the NACA 64A010. The numbering for the bumps starts from 0 at the trailing edge on the lower surface and increases as one

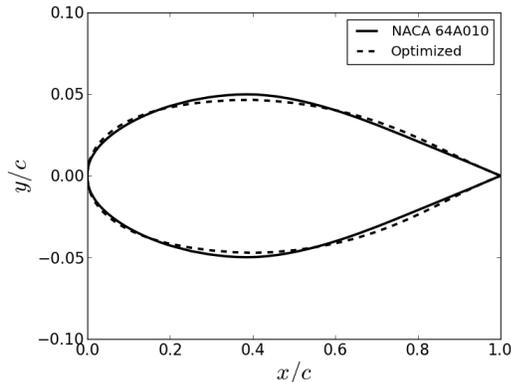
wraps around the leading edge and upper surface of the airfoil. In order to verify the accuracy of the gradient information obtained via the time-accurate adjoint, a comparison was made between the time-averaged drag gradients with respect to the design variables as calculated using both the continuous adjoint (after solving the adjoint equations in reverse time using the stored solution data from the numerical experiment performed above) and a finite differencing approach with small step sizes for the bump deformations ($10^{-6}c$). The resulting gradients are compared in Fig. 5.6 and exhibit very good agreement apart from some discrepancies near the trailing edge where the turbulence model is more active, which might be expected due to the frozen viscosity assumption in the adjoint formulation.



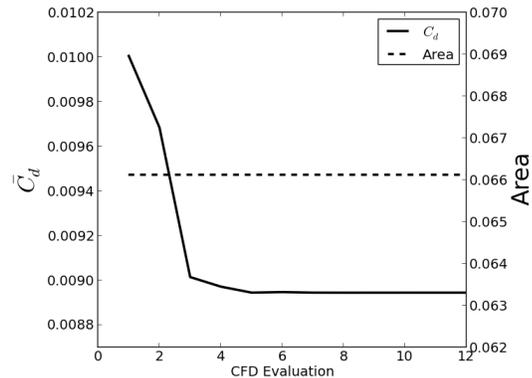
(a) Comparison of lift coefficient versus angle of attack in degrees between simulation and experiment using URANS.



(b) C_D history for the initial and final pitching airfoil designs. The average values are also shown as horizontal lines. The average drag is greatly reduced for the final design.



(c) Comparison of the initial and final airfoil profiles.



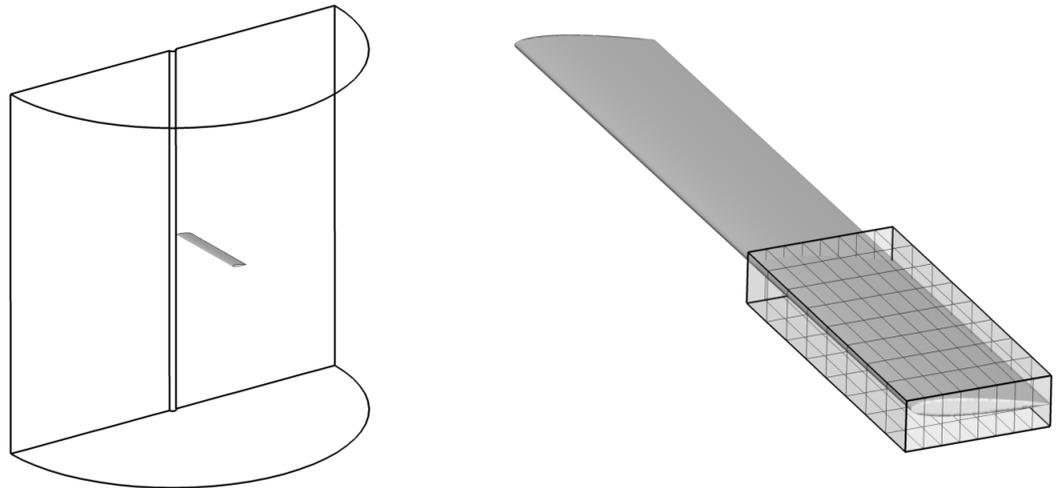
(d) Average drag and area for each CFD evaluation during the optimization process.

Figure 5.7: Force coefficient histories, shape comparison, and optimization history for the pitching airfoil design.

A redesign of the pitching NACA 64A010 airfoil was performed using the gradient information obtained from the time-accurate viscous adjoint formulation. The specific shape optimization problem was for time-averaged drag minimization using the same 50 bump design variables from the gradient verification with a constraint that the internal area of the airfoil remain constant. After 12 CFD evaluations, the time-averaged drag, \bar{C}_d , was successfully reduced by 10.6 %. A time history of the lift and drag is shown in Fig. 5.7, along with the optimization history. Profile shapes of the initial and final designs are also compared in Fig. 5.7. The shocks have been removed from the design by a thinning of the profile shape near the mid-chord, while maintaining a constant airfoil area. It should be noted that the shape changes in the final profile for the RANS-based design are more subtle than those for the same case with the Euler equations, especially near the trailing edge where large deflection could result in separation.

5.4 Rotor in Hover

For the first design case in 3D, a simple rotor geometry was chosen so that comparisons of simulation against the experimental data of Caradonna and Tung [16] could be made. The rotor geometry consists of two untwisted, untapered blades with an aspect ratio of 6 and a constant NACA 0012 airfoil section along the entire span. A lifting case was chosen with a collective pitch angle of 8 degrees and a pre-cone angle of 0.5 degrees. The flow conditions are that of hover at 2500 RPM which results in a tip Mach number of 0.877. The Euler equations expressed in a rotating frame (and their corresponding adjoint) were used to simulate this case.



(a) Mesh topology showing the rotor blade surface, outer boundaries, and periodic faces making up the half-cylinder.

(b) Rotor geometry with an FFD box surrounding the blade tip.

Figure 5.8: Mesh and FFD box details for the Caradonna and Tung numerical experiment.

The computational mesh, the topology of which can be seen in Fig. 5.8, takes advantage of

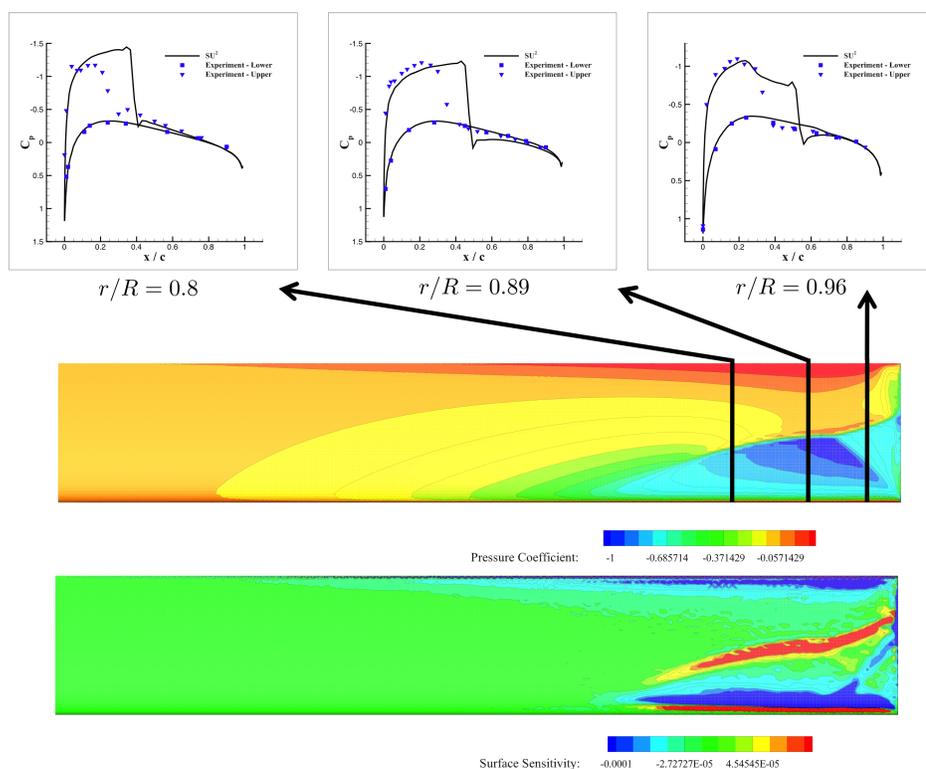


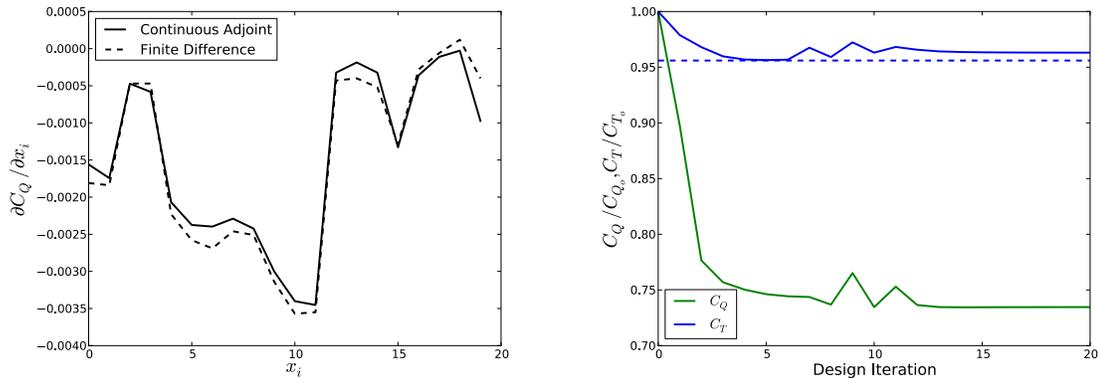
Figure 5.9: C_p contours on the upper surface of the baseline rotor geometry along with a comparison to experiment at multiple span locations. The blade tip is on the right, which is rotating toward the bottom of the page. The surface sensitivity contours for a torque objective function are also shown. Note the high sensitivity to shape deformations in the vicinity of the shock.

rotational periodicity by simulating flow over a single blade in a half-cylinder. The mixed-element mesh uses 3.36 million tetrahedra and 19,422 pyramids with a total of 588,572 nodes. The outer faces of the domain are near enough to the rotor that care must be taken to allow for subsonic induced velocities, including the rotor wake, to pass through them. Therefore, characteristic-based inlet and outlet conditions are applied on these boundaries. Flow tangency is satisfied at the blade surface and the small central hub. The small hub does not exist in the physical model, but adding it eases the construction of the periodic boundaries in the computational grid with little change in the solution.

Fig. 5.9 contains the C_p contours on the upper blade surface along with C_p distributions at several radial stations compared to experiment. Note the shock near the blade tip where the flow is transonic, and as seen in the C_p comparisons with experiment, the shock is crisply captured. There is a discrepancy in the location of the shock compared to experiment due to the inviscid approximation, although the present results agree with other available inviscid simulations. Also shown in Fig. 5.9 are surface sensitivity contours from the rotating adjoint solution based on a coefficient of torque

C_Q objective function. It should be noted that the most sensitive locations on the blade surface are in the vicinity of the shock and the expansion region upstream of the shock. Visualizing the surface sensitivities in this manner can offer designer intuition and can also aid the designer in defining appropriate design variables for automatic shape design.

In order to verify the gradient information and optimize the rotor geometry, design variables were defined using a FFD parameterization. First, a box was generated around the tip region of the blade where shape changes are to be made, as seen in Fig. 5.8. The design variables are the displacement of the individual control points that define this FFD box. Note that the surface sensitivities have guided the design variable definition by locating the FFD box around the more sensitive tip region. Displacements in the vertical direction were allowed for 84 total control points on the upper and lower surfaces of the FFD box. Control points near the trailing edge and inboard side of the FFD box were held fixed to maintain a smooth surface during deformation. Before attempting shape design, we compared the gradients of C_Q with respect to a subset of the FFD control point variables on the upper surface given by both the continuous adjoint and finite differencing. The gradient verification appears in Fig. 5.10. While the adjoint and finite differencing gradients are not identical, they exhibit excellent agreement for a complex 3D cases on unstructured meshes.



(a) Continuous adjoint and finite difference gradient comparison for 19 FFD control point variables.

(b) Optimization history for a thrust-constrained (dotted line) inviscid torque minimization of the rotor geometry.

Figure 5.10: Gradient verification using the FFD control point variables and optimization results.

Lastly, a redesign of the rotor blade shape for minimizing torque with a minimum thrust constraint of $C_T = 0.0055$ was performed using gradient information obtained via the continuous adjoint approach. After 20 design cycles, C_Q was reduced by 26.9 % from 0.0006098 to 0.0004458, and a C_T value of 0.00553 was maintained (down from a starting value of 0.00575). The optimization histories for C_T and C_Q are presented in Fig. 5.10. The baseline and final surface shapes near the blade tip along with C_p contours are compared in Fig. 5.11. The strong shock on the upper surface has been removed due to a pronounced change in the shape near the tip. The optimized design features a blade tip with thinner, asymmetric sections. Camber has been introduced near the leading edge to

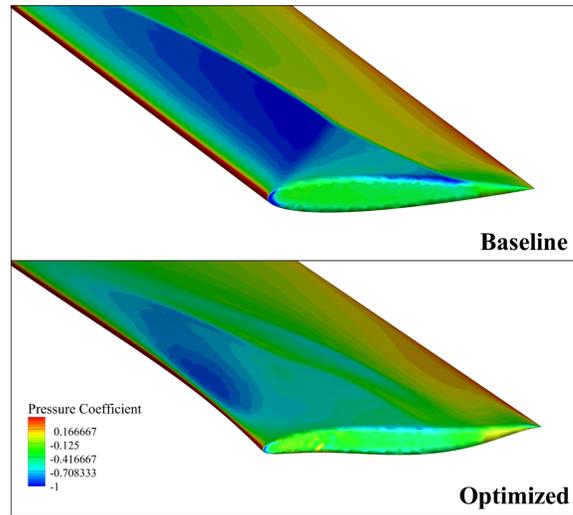


Figure 5.11: Comparison of the baseline and optimized rotor geometries along with C_p contours. The strong shock has been removed due to a distinct change in the tip shape.

help avoid the development of a shock while maintaining thrust.

5.5 Wind Turbine Design

To demonstrate the effectiveness of the non-inertial viscous adjoint methodology for large-scale, complex geometries, the NREL Phase VI wind turbine was chosen. The turbine geometry consists of two blades with a radius of 5.029 m and a constant S809 airfoil section along the entire span. This geometry has been used widely for CFD validation studies along with the data from the NREL Phase VI Unsteady Aerodynamics Experiment [113, 105]. The selected case for the present study is Sequence S with a 7 m/s wind speed and an RPM of 72. The computational mesh consists of 3.2 million nodes and 7.9 million elements, with triangles on the surface of the blade and prismatic elements in the boundary layer before transitioning to tetrahedra in the far-field.

The flow field around the turbine blades was calculated with the non-inertial RANS equations with the S-A turbulence model. For validation purposes, Fig. 5.12 gives C_p distributions at two radial stations as computed and compared to experiment, and Fig. 5.13 contains the C_p contours on the blade surface. Good agreement is seen overall, apart from near the trailing edge of the blade where some discrepancies are found (spikes in C_p are also seen at the sharp trailing edge, which are common with this type of unstructured mesh). More investigation into low-Mach number preconditioning and additional modifications to the S-A model could also be pursued to further improve the results. The surface sensitivity was computed for a torque objective function, and sensitivity contours can be seen on the blade in Fig. 5.13. It should be noted that the most sensitive locations on the blade surface are outboard locations along the span highlighted by the surface sensitivity contours.

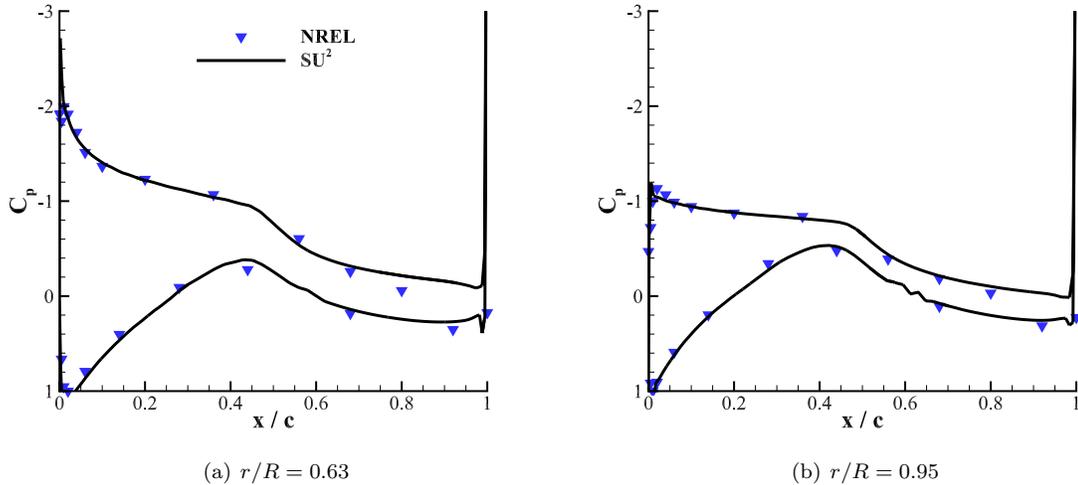


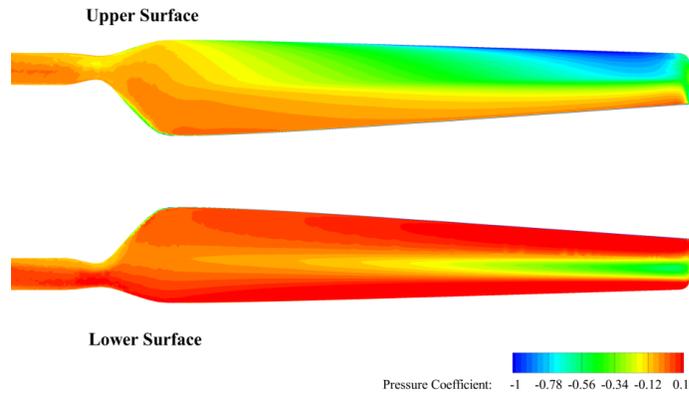
Figure 5.12: C_p distributions at multiple radial blade stations compared with experimental data.

While a more realistic objective function for wind turbine design might involve total power (and possibly multi-point design), we demonstrate the viscous adjoint here with a simple redesign of the rotor blade shape for increasing torque using gradient information obtained via the continuous adjoint approach. The blade geometry is parameterized by a free-form deformation approach. First, boxes were generated around the two blades and located where shape changes are to be made. The displacement of the individual control points that define the FFD boxes are then manipulated as design variables. Displacements in the vertical direction were allowed for 84 control points on the upper and lower surfaces of each FFD box. In order to maintain a smooth surface during deformation, control points near the trailing edge and inboard side of the FFD box were held fixed. After 3 design cycles (terminated after this point to save excessive cost with this large problem), the torque coefficient was increased by 4.0 % from 0.00147 to 0.00153. These optimization results are presented in Fig. 5.13, including a comparison of the initial and final surface shapes.

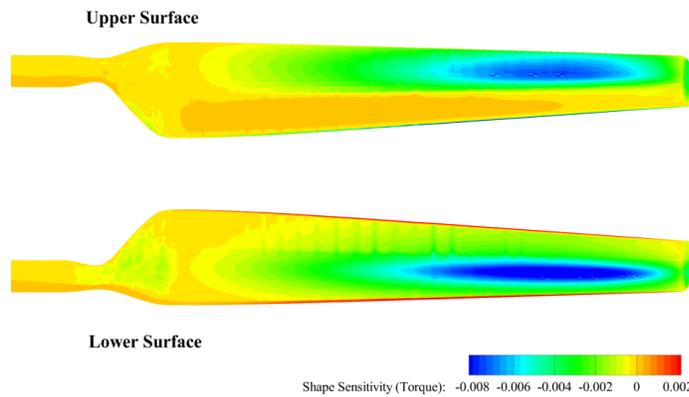
5.6 Pitching Wing in Inviscid Flow

In order to test the unsteady adjoint capabilities on a realistic geometry in three dimensions, the ONERA M6 wing was selected as a baseline geometry. The ONERA M6 wing was designed in 1972 by the ONERA Aerodynamics Department as an experimental geometry for studying three-dimensional, high Reynolds number flows with some complex flow phenomena (transonic shocks, shock-boundary layer interaction, separated flow). It has become a classic validation case for CFD codes due to the available geometric description, complicated flow physics, and the availability of experimental data.

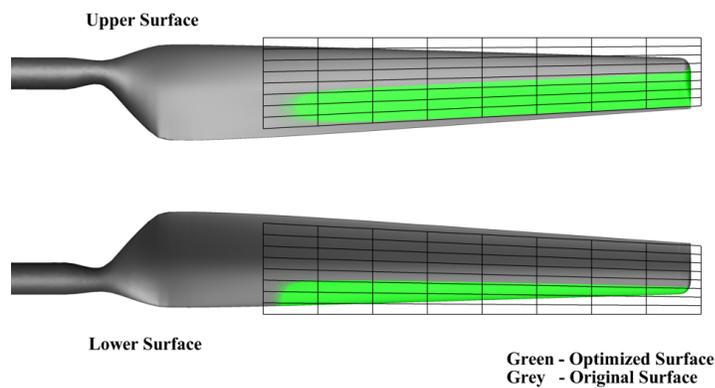
This unsteady test case was performed in inviscid flow at a transonic Mach number with the



(a) Surface contours of pressure coefficient.



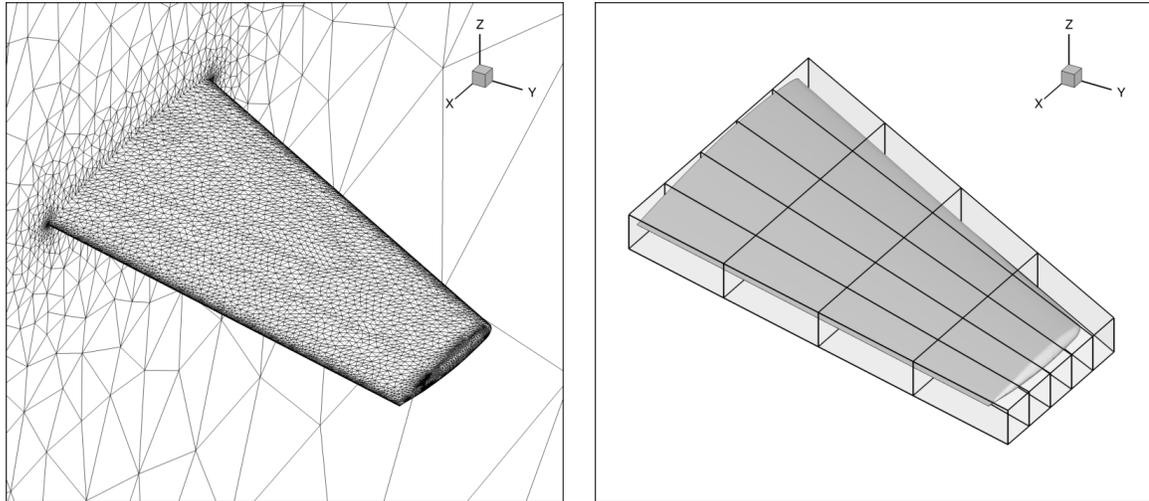
(b) Surface sensitivity contours for a torque objective function.



(c) FFD box along with the baseline and final shape for the turbine design.

Figure 5.13: Pressure coefficient, surface sensitivity, and shape modifications for the NREL Phase VI wind turbine.

wing pitching about an axis that is perpendicular to the root airfoil section of the wing and passes through its quarter-chord location. The specific flow conditions were as follows: a reduced frequency w_r of 0.1682, $M_\infty = 0.8395$, a mean angle of attack of 3.06 degrees, and a pitching amplitude of 2.5 degrees.



(a) Surface mesh topology showing the wing surface and symmetry plane.

(b) Wing surface with the surrounding FFD box.

Figure 5.14: Surface mesh for the inviscid case and FFD box for the pitching ONERA M6 numerical experiment.

The baseline unstructured mesh consists of 582,752 tetrahedral elements and a total of 108,396 nodes. Flow tangency is satisfied on the wing surface, a symmetry plane is used to reflect the flow about the plane of the root airfoil section to mimic the effect of the full wing planform, and characteristic-based boundary conditions are applied at the far-field. The surface meshes for the wing geometry and symmetry plane are shown in Fig. 5.14. All simulations of the pitching wing were time-accurate (dual time-stepping) with 25 times steps per period for a total of 10 periods. This afforded adequate time for transient effects to wash away and for well-resolved time-averaged behavior in the periodic steady state. For the unsteady adjoint, the solution at each physical time step from the direct problem (including grid node coordinates and velocities) was written to disk and then each was loaded in reverse fashion while integrating the adjoint equations backward in physical time.

Fig. 5.15 contains the C_p on the upper blade surface for three different time instances that correspond to the incidence of minimum drag (left), mean drag (middle), and maximum drag (right). From left to right, one can see the wing develop a strong shock in the outboard region. Fig. 5.15 also contains surface sensitivity contours from the unsteady adjoint solution based on a drag objective function. The corresponding surface sensitivity maps highlight the locations on the wing that, if deflected inward or outward in the local normal direction, would most influence the drag. Notice

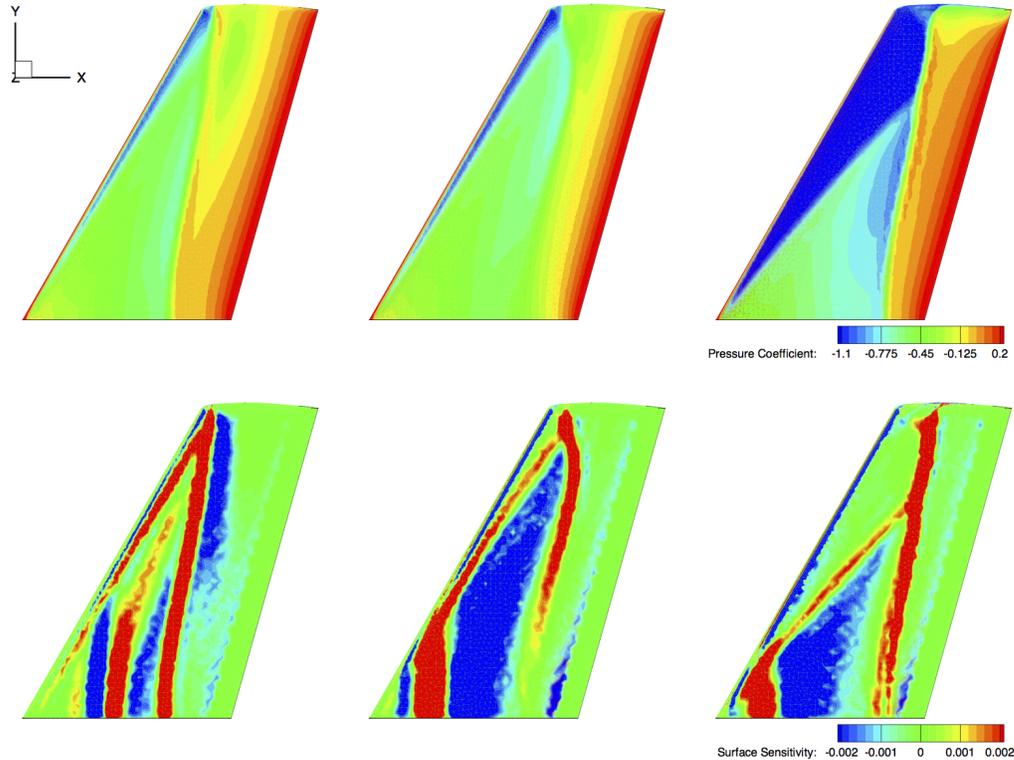
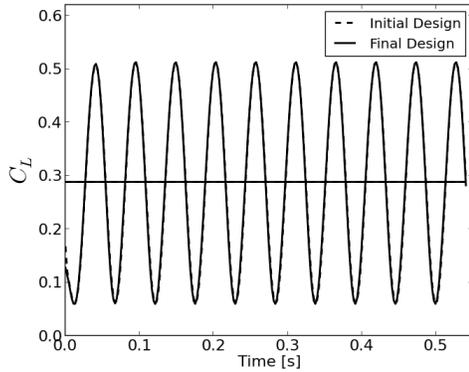


Figure 5.15: Pressure coefficient (top row) and surface sensitivity contours (drag objective function, bottom row) for three time instances during one period of oscillation. The figures in the left column correspond to the minimum drag incidence, those in the middle column are near the mean angle of incidence, and those on the right correspond to the incidence of maximum drag.

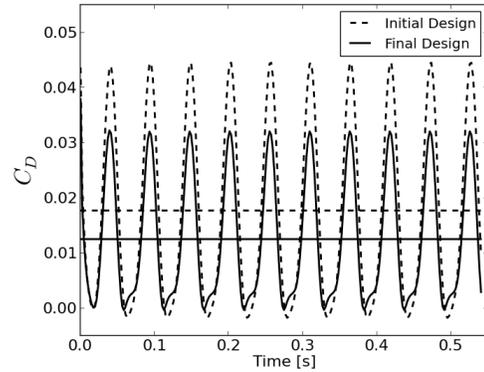
that the highly sensitive areas correspond to the shock locations and the expansion regions upstream of the shocks.

For gradient verification and for optimizing the pitching wing geometry, three-dimensional design variables were defined using an FFD parameterization. Displacements in the vertical direction (z -direction) was allowed for 50 total control points on the upper and lower surfaces of the FFD box. The entire wing geometry was included within the FFD box, but the sensitivity at the nodes along the sharp trailing edge was removed before computing gradient information. Before performing shape design, we verified the continuous adjoint gradient of the time-averaged drag \bar{C}_D with respect to a subset of the FFD control point variables on the upper surface against a finite difference gradient. Fig. 5.14 contains a view of the FFD box around the wing geometry, and the gradient comparison appears in Fig. 5.16. The adjoint and finite difference gradients are in good agreement for this 3D problem.

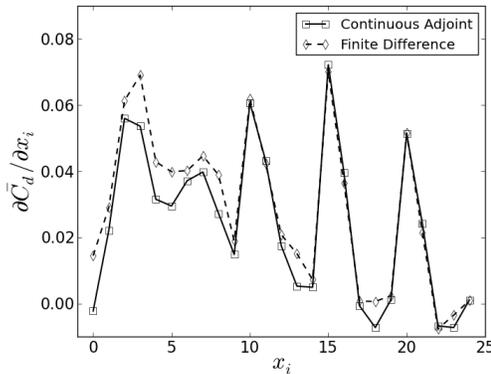
A redesign of the pitching ONERA M6 wing was then performed using gradients obtained via



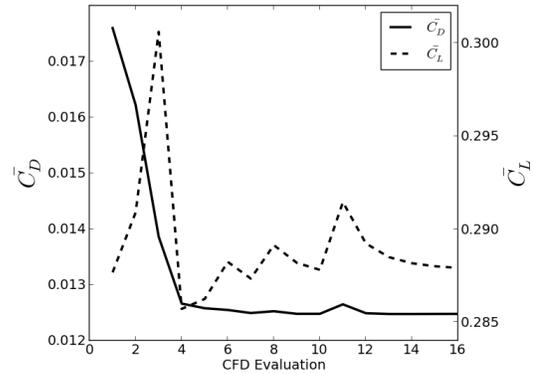
(a) C_L history for the initial and final pitching wing designs. The average values are also shown as horizontal lines. The average lift only changed slightly with the new design.



(b) C_D history for the initial and final pitching wing designs. The average values are also shown as horizontal lines. The average drag is greatly reduced for the final design, especially near the incidence of maximum drag.



(c) Comparison of the time-averaged drag gradient obtained by the continuous adjoint and finite differencing for a subset of the FFD control points above the upper wing surface.



(d) Lift and drag values for each CFD evaluation during the optimization process.

Figure 5.16: Force coefficient histories, gradient comparison, and optimization history for the inviscid pitching wing design.

the unsteady adjoint. The specific shape optimization problem was for the minimization of the time-averaged drag \bar{C}_D with time-averaged lift \bar{C}_L and geometric constraints. The \bar{C}_L was required to be greater than the original value of 0.2864 for the baseline pitching ONERA M6. The maximum thickness at five spanwise sections of the wing was also constrained to be larger than a specified value. After 16 evaluations, \bar{C}_D was successfully reduced by 29.1 %, and all of the constraints were successfully met. Table 5.1 contains a summary of the results.

Fig. 5.16 contains the lift and drag histories over the 10 periods of oscillation for the initial and final geometries. While the C_L history is largely unchanged, the large peaks in the C_D history have

Table 5.1: Objective and constraint values for the baseline and final pitching wing design. The maximum thickness is denoted by $t_{y/b}$ where the subscript gives the spanwise location of the section as a percentage of span, b .

	\bar{C}_D	\bar{C}_L	$t_{0.07}/b$	$t_{0.29}/b$	$t_{0.51}/b$	$t_{0.72}/b$	$t_{0.94}/b$
Constraints	.	0.2864	0.04765	0.04288	0.03820	0.03335	0.02867
ONERA M6	0.01760	0.28768	0.06359	0.05724	0.05095	0.04458	0.03831
Optimized	0.01248	0.28792	0.05887	0.05223	0.04707	0.04237	0.03746

been greatly reduced with the optimized design. This can be attributed to a reduction in the shock strengths during pitching when the wing is at a high angle of attack. The pressure contours on the upper surface at the incidence of maximum drag demonstrate this result, as shown in Fig. 5.17. The strong shocks on the ONERA M6 wing are much weaker for the optimized design, and in particular, the strong shock in the outboard region is almost entirely removed. The section profile shapes and the surface geometries of the initial and final designs are compared in Fig. 5.18. The optimized geometry features increased camber (to maintain lift), especially near the tip, and a slight thickening of the wing near the trailing edge. All of the optimized sections exhibit a slightly reduced thickness, which helps avoid drag penalties due to the development of shocks.

5.7 Pitching Wing in Turbulent Flow

As a final test of the unsteady adjoint capabilities, the pitching ONERA M6 wing design case from above was repeated using the URANS equations. This problem is meant to demonstrate a large-scale, viscous shape optimization with all of the mathematical and numerical complexity required for tackling larger problems of industrial interest (with the only additional requirements being a larger mesh and more computational resources).

The test case was performed in turbulent flow at a transonic Mach number with the wing pitching about an axis that is perpendicular to the root airfoil section of the wing and passes through its quarter-chord location. The specific flow conditions were as follows: a reduced frequency w_r of 0.1682, $M_\infty = 0.8395$, a mean angle of attack of 3.06 degrees, a pitching amplitude of 2.5 degrees, and a Reynolds number of 11.72 million.

The baseline unstructured mesh around the ONERA M6 consisted of 545,438 tetrahedral elements and a total of 96,252 nodes. The mesh spacing near the wall was set to achieve a $y^+ < 1$ over the entire wing surface. A no-slip condition is satisfied on the wing surface, a symmetry plane is used to reflect the flow about the plane of the root airfoil section to mimic the effect of the full wing planform, and characteristic-based boundary conditions are applied at the far-field boundary. The surface meshes for the wing geometry and symmetry plane are shown in Fig. 5.19. All simulations of the pitching wing were time-accurate (dual time-stepping) with 25 times steps per period for a total of 7 periods until reaching a periodic steady state. The total number of periods was reduced from the Euler design problem simply to reduce computational cost.

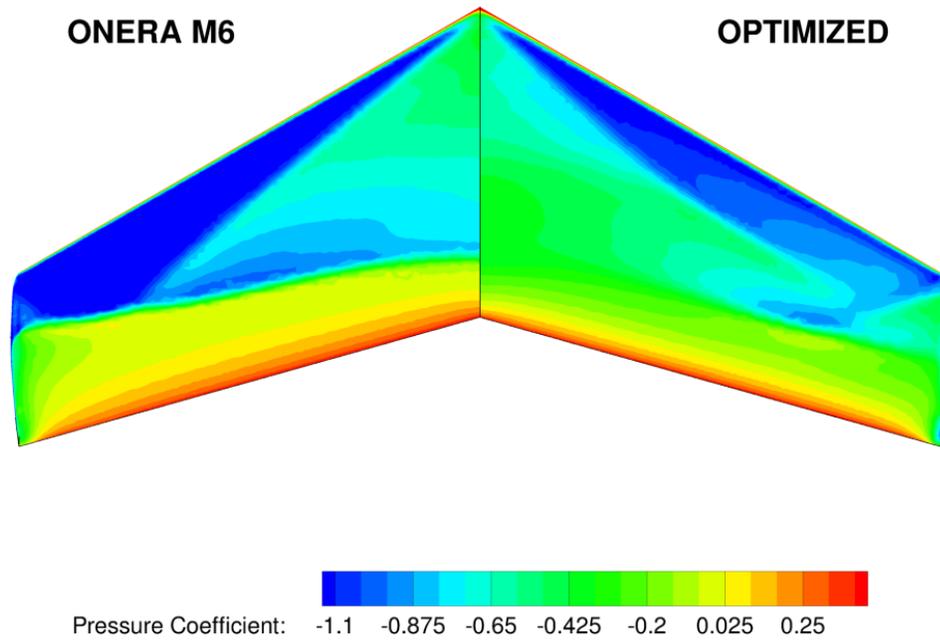


Figure 5.17: Pressure coefficient contour comparison between the ONERA M6 and the final design at the incidence of maximum drag for the inviscid case.

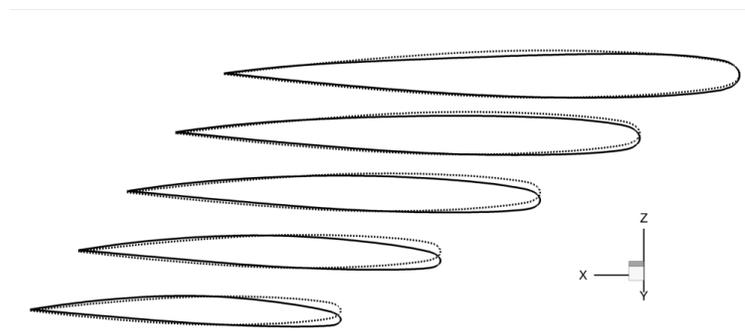


Figure 5.18: Shape comparison of the baseline (dotted) and final (solid) inviscid wing sections.

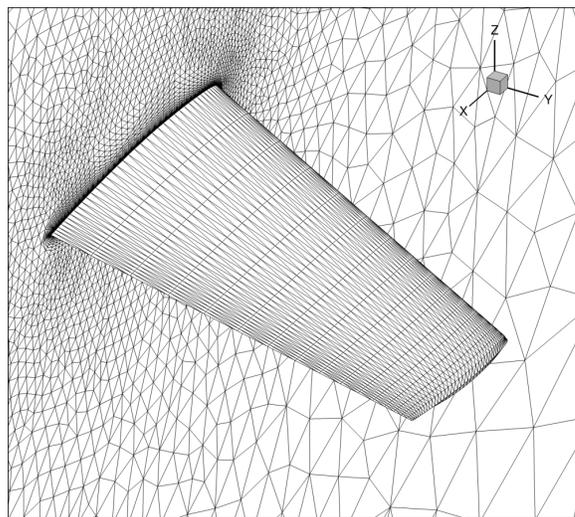
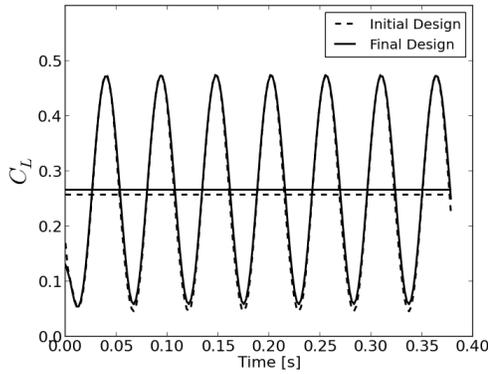


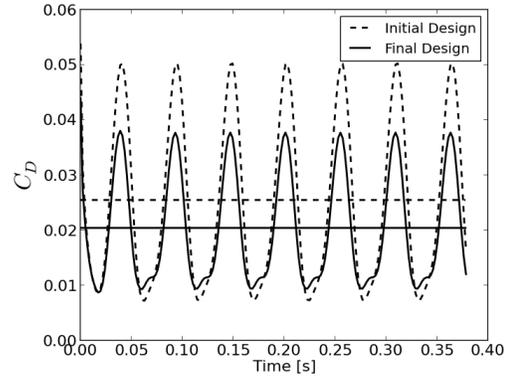
Figure 5.19: Surface mesh topology showing the wing surface and symmetry plane for the URANS case.

The pitching ONERA M6 wing was redesigned using gradients obtained via the viscous, time-accurate adjoint. The shape optimization problem was the same as in the Euler case: minimization of the time-averaged drag \bar{C}_D with time-average lift \bar{C}_L and geometric constraints on the spanwise thicknesses. An aggressive constraint was imposed on \bar{C}_L , as it was required to be greater than a value of 0.268 (a 4 % increase over the \bar{C}_L for the baseline geometry). The maximum thickness at five spanwise sections of the wing was also constrained to be larger than specified values. The same set of FFD control point variables were recycled from the Euler problem for the design variable parameterization. After 11 evaluations, \bar{C}_D was successfully reduced by 20.3 %, and the geometric constraints were met. The optimizer was also successfully able to increase the \bar{C}_L to a value of 0.266, although this was slightly less than the prescribed constraint value. The optimization was terminated at this point to save excessive computation.

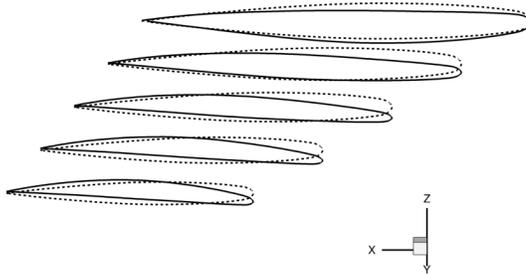
Fig. 5.20 presents the lift and drag histories over the 7 periods of oscillation for the initial and final geometries. The C_L history reflects slightly increased time-averaged lift, while the large peaks in the C_D history have again been reduced in the optimized design due to a reduction in the shock strengths when the wing is at large incidence angles. Fig. 5.21 shows a comparison of the pressure contours on the upper wing surface at the incidence of maximum drag. The strong shock in the outboard region is almost entirely removed. The section profile shapes for the baseline and final designs are compared in Fig. 5.20. As in the Euler case, the optimized geometry features increased camber and a slight thinning of the sections. The modifications to the camber across the entire wing span are more apparent in the RANS-based design.



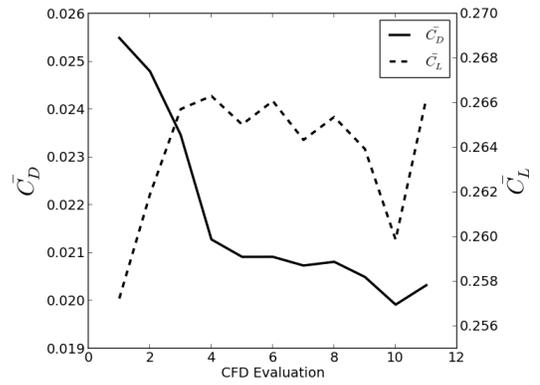
(a) C_L history for the initial and final pitching wing designs. The average values are also shown as horizontal lines. The average lift increased slightly with the new design.



(b) C_D history for the initial and final pitching wing designs. The average values are also shown as horizontal lines. The average drag is greatly reduced for the final design.



(c) Shape comparison between sections of the baseline (dotted line) and final (solid line) wing designs.



(d) Lift and drag values for each CFD evaluation during the optimization process.

Figure 5.20: Force coefficient histories, section shape comparison, and optimization history for the pitching wing design in turbulent flow.

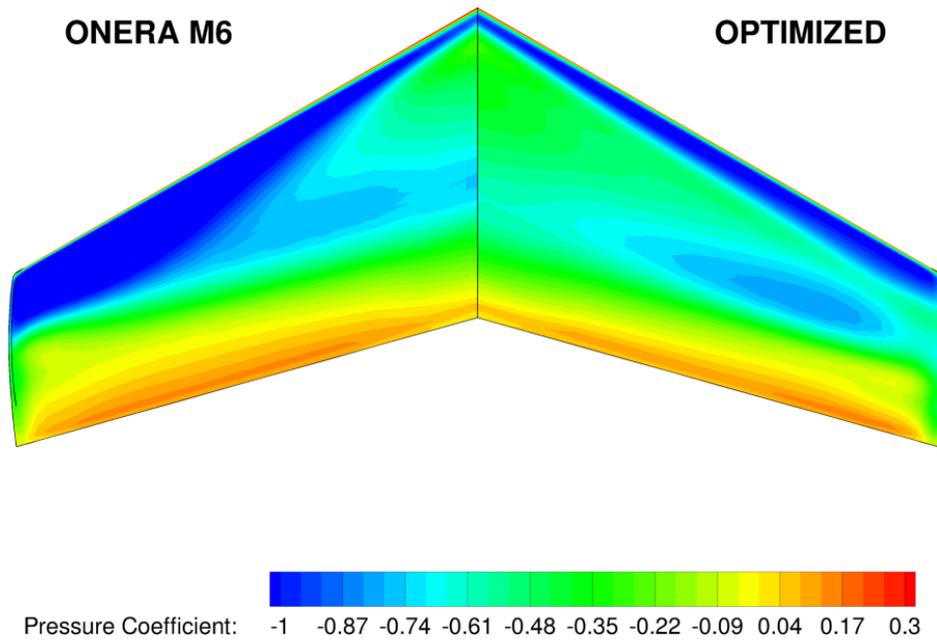


Figure 5.21: Pressure coefficient contour comparison between the ONERA M6 and the final design at the incidence of maximum drag for the URANS case.

Chapter 6

Conclusions and Future Directions

The major contributions of this dissertation are the detailed derivation, implementation, and application of a new unsteady continuous adjoint formulation based on shape calculus for aerodynamic design on dynamic meshes. A general framework has been presented that enables the immediate recovery of adjoint formulations for different scenarios, such as for design in unsteady turbulent flow, flow in rotating reference frames, or even steady inviscid flow. To ensure that the appropriate physics are included, the present continuous adjoint surface formulation has been derived for the unsteady, compressible RANS equations in ALE form with a generic source term.

A shape design framework was constructed within an open-source software suite for the numerical solution of PDEs and PDE-constrained optimization problems on general, unstructured meshes. The core of the suite is a three-dimensional, finite volume solver within which the ALE form of the governing equations (for solving unsteady flows on dynamic meshes) and the new unsteady adjoint formulation have been implemented. Additional elements needed for shape design, such as mesh deformation based on the linear elasticity equations, have also been incorporated into the suite as needed during the course of this dissertation. The shape design framework was successfully demonstrated through a collection of optimal shape design examples of rotating and pitching problems in both two and three dimensions.

It is important to note again that, from the general scenario of viscous, unsteady flow under which the present formulation was derived, the corresponding adjoint formulations for a variety of problems can be immediately recovered from the general framework. Moreover, as the unsteady continuous adjoint equations are a system of PDEs, they can be discretized in space and time using any valid approach, which offers valuable flexibility. For example, the equations can be immediately discretized with a time-spectral operator to give a time-spectral adjoint approach. This and other discretization strategies are currently under investigation and are included as future work.

In summary, the continuous adjoint formulation presented offers a flexible and efficient technique (in terms of compute and memory costs) for obtaining shape design gradients. Obtaining a surface formulation for shape design gradients (without a dependence on volume mesh sensitivities) and

the ability to tailor numerical solution methods for the adjoint equations (to help mitigate numerical stiffness and other convergence issues while avoiding memory overhead) make the continuous adjoint approach particularly attractive. While not explicitly discussed in this dissertation, the unsteady continuous adjoint formulation can also directly enable multidisciplinary design, analysis, and optimization involving other time-dependent physics associated with these systems, such as their structural or acoustic responses. Some preliminary work in this area has already begun [27].

The time-accurate formulation in this dissertation represents an advancement of continuous adjoint methods for performing design in unsteady flows. However, there are many potential future directions, including tackling additional functionals, developing new numerical methods for the adjoint equations, or treating interesting multi-physics problems, for instance. The following is a list of some possible future directions for this research:

- Remove the frozen viscosity assumption by either treating a turbulence model directly or by injecting some information from an approximation or surrogate.
- Continuous, unsteady coupled-adjoints for multi-physics problems, such as aeroacoustics and aeroelastics.
- Objective functions defined as integrals away from the design surface (inlet/outlets). In particular, there are interesting applications in active flow control (suction/blowing) that can be treated with further development of the characteristic-based adjoint boundary conditions.
- Design variable parameterization is always critical and could use more attention. Gradient smoothing approaches and advanced, “engineering-like” FFD variables could be pursued.
- Additional time discretizations can be directly applied to the present unsteady adjoint formulation (time spectral, for instance).
- Investigate other numerical methods for spatially integrating the convective, viscous, and source terms of the adjoint equations, or develop new numerical methods for the adjoint equations with customized artificial dissipation to help maintain accuracy and robustness.
- Efficient techniques for handling large quantities of unsteady solution data for use with the unsteady adjoint equations, including checkpointing techniques.
- Free trajectory problems (including optimization) where forces from CFD are coupled to the equations of motion for rigid bodies (6DOF).
- Demonstrate the use of temperature-based functionals for unsteady problems.
- Demonstrate design in transient conditions (non-periodic problems) using the present formulation.

Appendix A

Jacobians

This appendix contains the Jacobians of the governing flow equations that arise during the linearization process for the continuous adjoint derivation [14]. Here, we use index notation (repeated indices imply summation), and δ_{ij} is the Kronecker delta. Defining for convenience $a_0 = (\gamma - 1)$ and $\phi = (\gamma - 1) \frac{|\vec{v}|^2}{2}$, we have

$$\begin{aligned}
 A_i^c &= \begin{pmatrix} \cdot & \delta_{i1} & \delta_{i2} & \delta_{i3} & \cdot \\ -v_i v_1 + \delta_{i1} \phi & v_i - (a_0 - 1)v_i \delta_{i1} & v_1 \delta_{i2} - a_0 v_2 \delta_{i1} & v_1 \delta_{i3} - a_0 v_3 \delta_{i1} & a_0 \delta_{i1} \\ -v_i v_2 + \delta_{i2} \phi & v_2 \delta_{i1} - a_0 v_1 \delta_{i2} & v_i - (a_0 - 1)v_i \delta_{i2} & v_2 \delta_{i3} - a_0 v_3 \delta_{i2} & a_0 \delta_{i2} \\ -v_i v_3 + \delta_{i3} \phi & v_3 \delta_{i1} - a_0 v_1 \delta_{i3} & v_3 \delta_{i2} - a_0 v_2 \delta_{i3} & v_i - (a_0 - 1)v_i \delta_{i3} & a_0 \delta_{i3} \\ v_i (\phi - H) & -a_0 v_i v_1 + H \delta_{i1} & -a_0 v_i v_2 + H \delta_{i2} & -a_0 v_i v_3 + H \delta_{i3} & \gamma v_i \end{pmatrix}, \\
 A_i^{v1} &= \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot \\ -\eta_{i1} & \partial_i \left(\frac{1}{\rho} \right) + \frac{1}{3} \partial_1 \left(\frac{1}{\rho} \right) \delta_{i1} & \partial_1 \left(\frac{1}{\rho} \right) \delta_{i2} - \frac{2}{3} \partial_2 \left(\frac{1}{\rho} \right) \delta_{i1} & \partial_1 \left(\frac{1}{\rho} \right) \delta_{i3} - \frac{2}{3} \partial_3 \left(\frac{1}{\rho} \right) \delta_{i1} & \cdot \\ -\eta_{i2} & \partial_2 \left(\frac{1}{\rho} \right) \delta_{i1} - \frac{2}{3} \partial_1 \left(\frac{1}{\rho} \right) \delta_{i2} & \partial_i \left(\frac{1}{\rho} \right) + \frac{1}{3} \partial_2 \left(\frac{1}{\rho} \right) \delta_{i2} & \partial_2 \left(\frac{1}{\rho} \right) \delta_{i3} - \frac{2}{3} \partial_3 \left(\frac{1}{\rho} \right) \delta_{i2} & \cdot \\ -\eta_{i3} & \partial_3 \left(\frac{1}{\rho} \right) \delta_{i1} - \frac{2}{3} \partial_1 \left(\frac{1}{\rho} \right) \delta_{i3} & \partial_3 \left(\frac{1}{\rho} \right) \delta_{i2} - \frac{2}{3} \partial_2 \left(\frac{1}{\rho} \right) \delta_{i3} & \partial_i \left(\frac{1}{\rho} \right) + \frac{1}{3} \partial_3 \left(\frac{1}{\rho} \right) \delta_{i3} & \cdot \\ v_j \pi_{ij} & v_j \partial_j \left(\frac{1}{\rho} \right) \delta_{i1} + \zeta_{i1} + \frac{1}{\rho} \tau_{i1} & v_j \partial_j \left(\frac{1}{\rho} \right) \delta_{i2} + \zeta_{i2} + \frac{1}{\rho} \tau_{i2} & v_j \partial_j \left(\frac{1}{\rho} \right) \delta_{i3} + \zeta_{i3} + \frac{1}{\rho} \tau_{i3} & \cdot \end{pmatrix}, \\
 A_i^{v2} &= \gamma \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \frac{1}{a_0} \partial_i \left(\frac{\phi}{\rho} - \frac{p}{\rho^2} \right) & -\partial_i \left(\frac{v_1}{\rho} \right) & -\partial_i \left(\frac{v_2}{\rho} \right) & -\partial_i \left(\frac{v_3}{\rho} \right) & \partial_i \left(\frac{1}{\rho} \right) \end{pmatrix}, \\
 D_{ii}^{v1} &= \frac{1}{\rho} \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot \\ -\left(1 + \frac{1}{3} \delta_{i1}\right) v_1 & \left(1 + \frac{1}{3} \delta_{i1}\right) & \cdot & \cdot & \cdot \\ -\left(1 + \frac{1}{3} \delta_{i2}\right) v_2 & \cdot & \left(1 + \frac{1}{3} \delta_{i2}\right) & \cdot & \cdot \\ -\left(1 + \frac{1}{3} \delta_{i3}\right) v_3 & \cdot & \cdot & \left(1 + \frac{1}{3} \delta_{i3}\right) & \cdot \\ -|\vec{v}|^2 - \frac{1}{3} v_i^2 & \left(1 + \frac{1}{3} \delta_{i1}\right) v_1 & \left(1 + \frac{1}{3} \delta_{i2}\right) v_2 & \left(1 + \frac{1}{3} \delta_{i3}\right) v_3 & \cdot \end{pmatrix},
 \end{aligned}$$

$$D_{ij}^{v_1} = \frac{1}{\rho} \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot \\ -v_i \delta_{j1} + \frac{2}{3} v_j \delta_{i1} & \delta_{j1} \delta_{i1} - \frac{2}{3} \delta_{i1} \delta_{j1} & \delta_{j1} \delta_{i2} - \frac{2}{3} \delta_{i1} \delta_{j2} & \delta_{j1} \delta_{i3} - \frac{2}{3} \delta_{i1} \delta_{j3} & \cdot \\ -v_i \delta_{j2} + \frac{2}{3} v_j \delta_{i2} & \delta_{j2} \delta_{i1} - \frac{2}{3} \delta_{i2} \delta_{j1} & \delta_{j2} \delta_{i2} - \frac{2}{3} \delta_{i2} \delta_{j2} & \delta_{j2} \delta_{i3} - \frac{2}{3} \delta_{i2} \delta_{j3} & \cdot \\ -v_i \delta_{j3} + \frac{2}{3} v_j \delta_{i3} & \delta_{j3} \delta_{i1} - \frac{2}{3} \delta_{i3} \delta_{j1} & \delta_{j3} \delta_{i2} - \frac{2}{3} \delta_{i3} \delta_{j2} & \delta_{j3} \delta_{i3} - \frac{2}{3} \delta_{i3} \delta_{j3} & \cdot \\ -\frac{1}{3} v_i v_j & v_j \delta_{i1} - \frac{2}{3} v_i \delta_{j1} & v_j \delta_{i2} - \frac{2}{3} v_i \delta_{j2} & v_j \delta_{i3} - \frac{2}{3} v_i \delta_{j3} & \cdot \end{pmatrix} \quad (i \neq j),$$

$$D_{ii}^{v_2} = \frac{\gamma}{\rho} \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \frac{1}{a_0} \left(\phi - \frac{p}{\rho} \right) & -v_1 & -v_2 & -v_3 & 1 \end{pmatrix},$$

$$D_{ij}^{v_2} = \mathbf{0}_{5 \times 5} \quad (i \neq j),$$

where tensors $\bar{\eta}$, $\bar{\pi}$ and $\bar{\zeta}$ in the definition of $A_i^{v_1}$ are given by

$$\begin{aligned} \eta_{ij} &= \partial_i \left(\frac{v_j}{\rho} \right) + \partial_j \left(\frac{v_i}{\rho} \right) - \frac{2}{3} \delta_{ij} \nabla \cdot \left(\frac{\vec{v}}{\rho} \right), \\ \pi_{ij} &= v_j \partial_i \left(\frac{1}{\rho} \right) + v_i \partial_j \left(\frac{1}{\rho} \right) - \frac{2}{3} \delta_{ij} \vec{v} \cdot \nabla \left(\frac{1}{\rho} \right) = \eta_{ij} - \frac{1}{\rho} \tau_{ij}, \\ \zeta_{ij} &= v_j \partial_i \left(\frac{1}{\rho} \right) - v_i \partial_j \left(\frac{1}{\rho} \right) + \frac{1}{3} v_i \partial_j \left(\frac{1}{\rho} \right). \end{aligned}$$

In the above Jacobians, ρ is the fluid density, $\vec{v} = \{v_1, v_2, v_3\}^T \in \mathbb{R}^3$ is the flow speed in a Cartesian system of reference, p is the static pressure, γ is the ratio of specific heats, and H is the total enthalpy. A_i^c is the Jacobian of the convective flux with respect to the conservative variables, A_i^v is the Jacobian of the viscous flux with respect to the conservative variables, and D_{ij}^v is the Jacobian of the viscous flux with respect to the gradient of the conservative variables.

The Jacobian of the source term appearing during the linearization of the non-inertial form of the governing equations is given here in both two and three dimensions. It is assumed that the reference frame is under a steady rotation ($\vec{\omega} = \{\omega_x, \omega_y, \omega_z\}^T$), where the three components are in the Cartesian x -, y -, and z -directions. In 2D, it is assumed that the fluid exists in the x - y plane with an angular velocity in the z -direction. The Jacobian in 2D is

$$\frac{\partial \mathcal{Q}}{\partial U} = \begin{pmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & -\omega_z & \cdot \\ \cdot & \omega_z & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{pmatrix}.$$

In 3D, the steady rotation of the frame along an arbitrary axis results in the following Jacobian:

$$\frac{\partial Q}{\partial U} = \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & -\omega_z & \omega_y & \cdot \\ \cdot & \omega_z & \cdot & -\omega_x & \cdot \\ \cdot & -\omega_y & \omega_x & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix}.$$

Finally, it is often useful to apply the following transformation matrices in order to switch between conservative and primitive variables [48]:

$$M = \frac{\partial U}{\partial V} = \begin{pmatrix} 1 & \cdot & \cdot & \cdot & \cdot \\ v_1 & \rho & \cdot & \cdot & \cdot \\ v_2 & \cdot & \rho & \cdot & \cdot \\ v_3 & \cdot & \cdot & \rho & \cdot \\ \frac{|\vec{v}|^2}{2} & \rho v_1 & \rho v_2 & \rho v_3 & \frac{1}{(\gamma-1)} \end{pmatrix},$$

and

$$M^{-1} = \begin{pmatrix} 1 & \cdot & \cdot & \cdot & \cdot \\ -\frac{v_1}{\rho} & \frac{1}{\rho} & \cdot & \cdot & \cdot \\ -\frac{v_2}{\rho} & \cdot & \frac{1}{\rho} & \cdot & \cdot \\ -\frac{v_3}{\rho} & \cdot & \cdot & \frac{1}{\rho} & \cdot \\ \frac{(\gamma-1)|\vec{v}|^2}{2} & (\gamma-1)v_1 & (\gamma-1)v_2 & (\gamma-1)v_3 & (\gamma-1) \end{pmatrix}.$$

Bibliography

- [1] G. Anderson, M. Aftosmis, and M. Nemec. Parametric deformation of discrete geometry for aerodynamic shape design. *AIAA Paper 2012-0965*, 2012.
- [2] W. K. Anderson and D. L. Bonhaus. Airfoil design on unstructured grids for turbulent flows. *AIAA Journal*, 37(2):185–191, 1999.
- [3] W. K. Anderson and V. Venkatakrishnan. Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation. *AIAA Paper 1997-0643*, 1997.
- [4] E. Arian and M. D. Salas. Admitting the inadmissible: Adjoint formulation for incomplete cost functionals in aerodynamic optimization. *AIAA Journal*, 37(1):37–44, 1999.
- [5] A. Baeza, C. Castro, F. Palacios, and E. Zuazua. 2-D Euler shape design on nonregular flows using adjoint Rankine-Hugoniot relations. *AIAA Journal*, 47(3), 2009.
- [6] T. J. Barth. A 3-D upwind euler solver for unstructured meshes. *AIAA Paper 1991-1548*, 1991.
- [7] T. J. Barth. Aspects of unstructured grids and finite-volume solvers for the Euler and Navier-Stokes equations. In *AGARD, Special Course on Unstructured Grid Methods for Advection Dominated Flows*, volume 1, 1992.
- [8] T. J. Barth and D. Jespersen. The design and application of upwind schemes on unstructured grids. *AIAA Paper 1989-0366*, 1989.
- [9] J. T. Batina. Unsteady Euler airfoil solutions using unstructured dynamic meshes. *AIAA Journal*, 28(8):1381–1388, August 1990.
- [10] R. T. Biedron and J. L. Thomas. Recent enhancements to the FUN3D flow solver for moving-mesh applications. *AIAA Paper 2009-1360*, 2009.
- [11] J. Blazek. *Computational Fluid Dynamics: Principles and Applications*. Elsevier, Oxford, 2005.
- [12] A. Borzi. Introduction to multigrid methods. Technical report, Institut für Mathematik und Wissenschaftliches Rechnen (Karl-Franzens-Universität Graz), 2003.

- [13] C. L. Bottasso, D. Detomi, and R. Serra. The ball-vertex method: a new simple spring analogy method for unstructured dynamic meshes. *Computer Methods in Applied Mechanics and Engineering*, 194:4244–4264, 2005.
- [14] A. Bueno-Orovio, C. Castro, F. Palacios, and E. Zuazua. Continuous adjoint approach for the Spalart–Allmaras model in aerodynamic optimization. *AIAA Journal*, 50(3), 2012.
- [15] G. W. Burgreen and O. Baysal. Three-dimensional aerodynamic shape optimization using discrete sensitivity analysis. *AIAA Journal*, 34(9):1761–1770, 1996.
- [16] F. X. Caradonna and C. Tung. Experimental and analytical studies of a model helicopter rotor in hover. Technical Report NASA Technical Memorandum 81232, NASA, NASA Ames Research Center, Moffett Field, CA, 1981.
- [17] A. Carnarius, F. Thiele, E. Ozkaya, and N. R. Gauger. Adjoint approaches for optimal flow control. *AIAA Paper 2010-5088*, 2010.
- [18] C. Castro, C. Lozano, F. Palacios, and E. Zuazua. A systematic continuous adjoint approach to viscous aerodynamic design on unstructured grids. *AIAA Journal*, 45(9):2125–2139, 2007.
- [19] S. Choi, M. Postdam, K. Lee, G. Iaccarino, and J. J. Alonso. Helicopter rotor design using a time-spectral and adjoint-based method. *AIAA Paper 2008-5810*, 2008.
- [20] G. F. Corliss, C. Faure, A. Griewank, and L. Hascoet. *Automatic Differentiation of Algorithms: From Simulation to Optimization*. Springer Verlag, New York, 2002.
- [21] S. S. Davis. NACA 64A010 (NASA Ames model) oscillatory pitching, compendium of unsteady aerodynamic measurements. Technical Report AGARD R-702, AGARD, 1982.
- [22] C. Degand and C. Farhat. A three-dimensional torsional spring analogy method for unstructured dynamic meshes. *Computers & Structures*, 80:305–316, 2002.
- [23] J. Donea, A. Huerta, J.-Ph. Ponthot, and A. Rodriguez-Ferran. *Arbitrary Lagrangian-Eulerian Methods in Encyclopedia of Computational Mechanics*. John Wiley and Sons, 2004.
- [24] A. Dumont, A. le Pape, J. Peter, and S. Huberson. Aerodynamic shape optimization of hovering rotors using a discrete adjoint of the Reynolds-averaged Navier-Stokes equations. *Journal of the American Helicopter Society*, 56(3):1–11, July 2011.
- [25] R. P. Dwight. Robust mesh deformation using the linear elasticity equations. In *Computational Fluid Dynamics 2006*, Proceedings of the Fourth International Conference on Computational Fluid Dynamics, pages 401–406, Ghent, Belgium, July 2006. ICCFD.
- [26] R. P. Dwight and J. Brezillon. Effect of approximations of the discrete adjoint on gradient-based optimization. *AIAA Journal*, 44 (12):3022–3031, 2006.

- [27] T. D. Economon, F. Palacios, and J. J. Alonso. A coupled-adjoint method for aerodynamic and aeroacoustic optimization. *AIAA Paper 2012-5598*, 2012.
- [28] T. D. Economon, F. Palacios, and J. J. Alonso. Optimal shape design for open rotor blades. *AIAA Paper 2012-3018*, 2012.
- [29] T. D. Economon, F. Palacios, and J. J. Alonso. Unsteady aerodynamic design on unstructured meshes with sliding interfaces. *AIAA Paper 2013-0632*, 2013.
- [30] T. D. Economon, F. Palacios, and J. J. Alonso. A viscous continuous adjoint approach for the design of rotating engineering applications. *AIAA Paper 2013-2580*, 2013.
- [31] T. D. Economon, F. Palacios, and J. J. Alonso. An unsteady continuous adjoint approach for aerodynamic design on dynamic meshes. *AIAA Paper 2014-2300*, 2014.
- [32] P. Eliasson. Edge, a Navier-Stokes solver for unstructured grids. Technical Report FOI-R-0298-SE, FOI Scientific Report, 2002.
- [33] J. Elliott and J. Peraire. Practical three-dimensional aerodynamic design and optimization using unstructured meshes. *AIAA Journal*, 35(9):1479–1485, 1997.
- [34] C. Farhat, P. Geuzaine, and C. Grandmont. The discrete geometric conservation law and the nonlinear stability of ALE schemes for the solution of flow problems. *Journal of Computational Physics*, 174(2):669–694, December 2001.
- [35] K. J. Fidkowski and D. L. Darmofal. Review of output-based error estimation and mesh adaptation in computational fluid dynamics. *AIAA Journal*, 49(4), 2011.
- [36] J. A. Fike. *Multi-Objective Optimization using Hyper-Dual Numbers*. PhD thesis, Department of Aeronautics and Astronautics, Stanford University, 2013.
- [37] P. D. Frank and G. R. Shubin. A comparison of optimization-based approaches for a model computational aerodynamics design problem. *Journal of Computational Physics*, 98(1):74 – 89, 1992.
- [38] M. B. Giles, M. C. Duta, and J.-D. Muller. Adjoint code developments using the exact discrete approach. *AIAA Paper 2001-2596*, 2001.
- [39] M. B. Giles, M. C. Duta, J.-D. Muller, and N. A. Pierce. Algorithm developments for discrete adjoint methods. *AIAA Journal*, 41(2):198–205, February 2003.
- [40] M. B. Giles and N. A. Pierce. Adjoint equations in CFD: duality, boundary conditions and solution behaviour. *AIAA Paper 1997-1850*, 1997.
- [41] M. B. Giles and N. A. Pierce. An introduction to the adjoint approach to design. *Flow, Turbulence and Combustion*, 65:393–415, 2000.

- [42] M. B. Giles and E. Süli. Adjoint methods for PDEs: a posteriori error analysis and postprocessing by duality. *Acta Numerica*, 11:145–236, 2002.
- [43] A. K. Gopinath and A. Jameson. Time spectral method for periodic unsteady computations over two- and three-dimensional bodies. *AIAA Paper 2005-1220*, 2005.
- [44] A. Griewank and A. Walther. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation (2nd ed.)*. SIAM, Philadelphia, 2008.
- [45] K. C. Hall, J. P. Thomas, and W. S. Clark. Computation of unsteady nonlinear flows in cascades using harmonic balance technique. *AIAA Journal*, 40(5):879–886, 2002.
- [46] M. Hayashi, M. Ceze, and E. Volpe. Characteristics-based boundary conditions for the Euler adjoint problem. *International Journal for Numerical Methods in Fluids*, 71(10):1297–1321, 2012.
- [47] R. M. Hicks and P. A. Henne. Wing design by numerical optimization. *Journal of Aircraft*, 15:407–412, 1978.
- [48] C. Hirsch. *Numerical Computation of Internal and External Flows*. Wiley, New York, 1984.
- [49] D. G. Holmes and S. S. Tong. A three-dimensional Euler solver for turbomachinery blade rows. *Journal of Engineering for Gas Turbines and Power*, 107(2):258–264, April 1985.
- [50] A. Jameson. Aerodynamic design via control theory. *Journal of Scientific Computing*, 3:233–260, 1988.
- [51] A. Jameson. Computational aerodynamics for aircraft design. *Science*, 245(4916):361–371, 1989.
- [52] A. Jameson. Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings. *AIAA Paper 1991-1596*, 1991.
- [53] A. Jameson. Analysis and design of numerical schemes for gas dynamics 1 artificial diffusion, upwind biasing, limiters and their effect on accuracy and multigrid convergence. *RIACS Technical Report 94.15, International Journal of Computational Fluid Dynamics*, 4:171–218, 1995.
- [54] A. Jameson. Analysis and design of numerical schemes for gas dynamics 2 artificial diffusion and discrete shock structure. *RIACS Report No. 94.16, International Journal of Computational Fluid Dynamics*, 5:1–38, 1995.
- [55] A. Jameson. Optimum aerodynamic design using CFD and control theory. *AIAA Paper 1995-1729*, 1995.
- [56] A. Jameson. A perspective on computational algorithms for aerodynamic analysis and design. *Progress in Aerospace Sciences*, 37:197–243, 2001.

- [57] A. Jameson, J. J. Alonso, J. Reuther, L. Martinelli, and J. C. Vassberg. Aerodynamic shape optimization techniques based on control theory. *AIAA Paper 1998-2538*, 1998.
- [58] A. Jameson and S. Kim. Reduction of the adjoint gradient formula for aerodynamic shape optimization problems. *AIAA Journal*, 41(11):2114–2129, 2003.
- [59] A. Jameson, L. Martinelli, and F. Grasso. A multigrid method for the Navier-Stokes equations. *AIAA Paper 1986-0208*, 1986.
- [60] A. Jameson, N. A. Pierce, and L. Martinelli. Optimum aerodynamic design using the Navier-Stokes equations. *Theoretical and Computational Fluid Dynamics*, 10:213–237, 1998.
- [61] A. Jameson and S. Schenectady. An assessment of dual-time stepping, time spectral and artificial compressibility based numerical algorithms for unsteady flow with applications to flapping wings. *AIAA Paper 2009-4273*, 2009.
- [62] A. Jameson, W. Schmidt, and E. Turkel. Numerical solution of the Euler equations by finite volume methods using Runge-Kutta time stepping schemes. *AIAA Paper 1981-1259*, 1981.
- [63] A. Jameson, S. Sriram, and L. Martinelli. A continuous adjoint method for unstructured grids. *AIAA Paper*, 2003-3955, 2003.
- [64] A. A. Johnson and T. E. Tezduyar. Simulation of multiple spheres falling in a liquid-filled tube. *Computer Methods in Applied Mechanics and Engineering*, 134:351–373, 1996.
- [65] G. Kenway, G. Kennedy, and J. R. R. A. Martins. A CAD-free approach to high-fidelity aerostructural optimization. *AIAA Paper 2010-9231*, 2010.
- [66] S. Kim, J. J. Alonso, and A. Jameson. A gradient accuracy study for the adjoint-based Navier-Stokes design method. *AIAA Paper 1999-0299*, 1999.
- [67] B. Koobus and C. Farhat. Second-order time-accurate and geometrically conservative implicit schemes for flow computations on unstructured dynamic meshes. *Computer Methods in Applied Mechanics and Engineering*, 170:103–129, 1999.
- [68] S. W. Lee and O. J. Kwon. Aerodynamic shape optimization of hovering rotor blades in transonic flow using unstructured meshes. *AIAA Journal*, 44(8):1816–1825, August 2006.
- [69] M. Lesoinne and C. Farhat. Geometric conservation laws for flow problems with moving boundaries and deformable meshes, and their impact on aeroelastic computations. *Computer Methods in Applied Mechanics and Engineering*, 134(1-2):71 – 90, 1996.
- [70] R. LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press, 2002.
- [71] Y. Li, Y. Allaneau, and A. Jameson. Continuous adjoint approach for adaptive mesh refinement. *AIAA Paper 2011-3982*, 2011.

- [72] J.-L. Lions. *Optimal Control of Systems Governed by Partial Differential Equations*. Springer-Verlag, New York, 1971.
- [73] C. A. Mader, J. R. R. A. Martins, J. J. Alonso, and E. van der Weide. ADjoint: An approach for the rapid development of discrete adjoint solvers. *AIAA Journal*, 46(4):863–873, April 2008.
- [74] K. Mani and D. J. Mavriplis. Unsteady discrete adjoint formulation for two-dimensional flow problems with deforming meshes. *AIAA Journal*, 46(6):1351–1364, 2008.
- [75] J. R. R. A. Martins, P. Sturdza, and J. J. Alonso. The complex-step derivative approximation. *ACM Transactions on Mathematical Software*, 29(3):245–262, 2003.
- [76] D. J. Mavriplis. Accurate multigrid solution of the Euler equations on unstructured and adaptive meshes. *AIAA Journal*, 28(2):213–221, February 1990.
- [77] D. J. Mavriplis. Multigrid techniques for unstructured meshes. Technical Report 95-27, ICASE, 1995.
- [78] D. J. Mavriplis. On convergence acceleration techniques for unstructured meshes. Technical report, Institute for Computer Applications on Science and Engineering (ICASE), 1998.
- [79] D. J. Mavriplis. Revisiting the least-squares procedure for gradient reconstruction on unstructured meshes. *AIAA Paper 2003-3986*, 2003.
- [80] D. J. Mavriplis. Discrete adjoint-based approach for optimization problems on three-dimensional unstructured meshes. *AIAA Journal*, 45(4):740–750, 2007.
- [81] D. J. Mavriplis. Solution of the unsteady discrete adjoint for three-dimensional problems on dynamically deforming unstructured meshes. *AIAA Paper 2008-727*, 2008.
- [82] D. J. Mavriplis and A. Jameson. Multigrid solution of the Navier-Stokes equations on triangular meshes. *AIAA Journal*, 28(8):1415–1425, August 1990.
- [83] M. McMullen, A. Jameson, and J. J. Alonso. Application of a non-linear frequency domain solver to the Euler and Navier-Stokes equations. *AIAA Paper 2002-0120*, 2002.
- [84] B. Mohammadi and O. Pironneau. Shape optimization in fluid mechanics. *Annual Review of Fluid Mechanics*, 36:255–279, 2004.
- [85] S. A. Morton, R. B. Melville, and M. R. Visbal. Accuracy and coupling issues of aeroelastic Navier-Stokes solutions on deforming meshes. *Journal of Aircraft*, 35(5):798–805, 1998.
- [86] S. K. Nadarajah. *The Discrete Adjoint Approach to Aerodynamic Shape Optimization*. PhD thesis, Department of Aeronautics and Astronautics, Stanford University, 2003.
- [87] S. K. Nadarajah and A. Jameson. A comparison of the continuous and discrete adjoint approach to automatic aerodynamic optimization. *AIAA Paper 2000-0667*, 2000.

- [88] S. K. Nadarajah and A. Jameson. Optimum shape design for unsteady flows with time-accurate continuous and discrete adjoint methods. *AIAA Journal*, 45(7):1478–1491, July 2007.
- [89] S. K. Nadarajah and A. Jameson. Optimum shape design for unsteady three-dimensional viscous flows using a nonlinear frequency-domain method. *Journal of Aircraft*, 44(5), 2007.
- [90] S. K. Nadarajah, M. McMullen, and A. Jameson. Non-linear frequency domain based optimum shape design for unsteady three-dimensional flow. *AIAA Paper 2006-1052*, 2006.
- [91] E. J. Nielsen and W. K. Anderson. Recent improvements in aerodynamic design optimization on unstructured meshes. *AIAA Journal*, 40(6), June 2002.
- [92] E. J. Nielsen and B. Diskin. Discrete adjoint-based design for unsteady turbulent flows on dynamic overset unstructured grids. *AIAA Paper 2012-0554*, 2012.
- [93] E. J. Nielsen, B. Diskin, and N. K. Yamaleev. Discrete adjoint-based design optimization of unsteady turbulent flows on dynamic unstructured grids. *AIAA Journal*, 48(6):1195–1206, 2010.
- [94] E. J. Nielsen, E. M. Lee-Rausch, and W. T. Jones. Adjoint-based design of rotors using the Navier-Stokes equations in a noninertial reference frame. *AHS International 65th Forum and Technology Display*, Grapevine, Texas, May 2009.
- [95] E. J. Nielsen, J. Lu, M. A. Park, and D. L. Darmofal. An implicit exact dual adjoint solution method for turbulent flows on unstructured grids. *Computers and Fluids*, 33:1131–1155, 2004.
- [96] F. Palacios and J. J. Alonso. New convergence acceleration techniques in the Joe code. Technical report, Center for Turbulence Research, Annual Research Brief 2011, 2011.
- [97] F. Palacios, J. J. Alonso, M. Colonno, J. Hicken, and T. Lukaczyk. Adjoint-based method for supersonic aircraft design using equivalent area distributions. *AIAA Paper 2012-0269*, 2012.
- [98] F. Palacios, M. R. Colonno, A. C. Aranake, A. Campos, S. R. Copeland, T. D. Economon, A. K. Lonkar, T. W. Lukaczyk, T. W. R. Taylor, and J. J. Alonso. Stanford University Unstructured (SU²): An open-source integrated computational environment for multi-physics simulation and design. *AIAA Paper 2013-0287*, 2013.
- [99] F. Palacios, K. Duraisamy, J. J. Alonso, and E. Zuazua. Robust grid adaptation for efficient uncertainty quantification. *AIAA Journal*, 50(7):1538–1546, July 2012.
- [100] F. Palacios, T. D. Economon, A. C. Aranake, S. R. Copeland, A. K. Lonkar, T. W. Lukaczyk, D. E. Manosalvas, K. R. Naik, A. S. Padron, B. Tracey, A. Variyar, and J. J. Alonso. Stanford University Unstructured (SU²): Open-source analysis and design technology for turbulent flows. *AIAA Paper 2014-0243*, 2014.
- [101] J. E. V. Peter and R. P. Dwight. Numerical sensitivity analysis for aerodynamic optimization: A survey of approaches. *Computers and Fluids*, 39:373–391, 2010.

- [102] N.A. Pierce and M.B. Giles. Preconditioned multigrid methods for compressible flow calculations on stretched meshes. *J. Comput. Phys.*, 136:425–445, 1997.
- [103] O. Pironneau. On optimum design in fluid mechanics. *Journal of Fluid Mechanics*, 64:97–110, 1974.
- [104] O. Pironneau. *Optimal Shape Design for Elliptic Systems*. Springer-Verlag, New York, 1984.
- [105] M. A. Potsdam and D. J. Mavriplis. Unstructured mesh CFD aerodynamic analysis of the NREL Phase VI Rotor. *AIAA Paper 2009-1221*, 2009.
- [106] A. Quarteroni and A. Valli. *Numerical approximation of partial differential equations*, volume 23 of *Springer series in computational mathematics*. Springer-Verlag Berlin Heidelberg New York, 1997.
- [107] P. L. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, 43:357–372, 1981.
- [108] M. P. Rumpfkeil and D. W. Zingg. A general framework for the optimal control of unsteady flows with applications. *AIAA Paper 2007-1128*, 2007.
- [109] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7:856–869, 1986.
- [110] J. A. Samareh. Survey of shape parameterization techniques for high-fidelity multidisciplinary shape optimization. *AIAA Journal*, 39(5):877–884, 2001.
- [111] J. A. Samareh. Aerodynamic shape optimization based on free-form deformation. *AIAA Paper 2004-4630*, 2004.
- [112] S. Schmidt, V. Schulz, C. Ilic, and N. R. Gauger. Three-dimensional large-scale aerodynamic design based on shape calculus. *AIAA Journal*, 51(11), November 2013.
- [113] D. A. Simms, S. Schreck, M. Hand, and L. J. Fingersh. NREL unsteady aerodynamics experiment in the NASA-Ames wind tunnel: A comparison of predictions to measurements. Technical Report NREL/TP-500-29494, NREL, June 2001.
- [114] J. Simon. Differentiation with respect to the domain in boundary value problems. *Numerical Functional Analysis and Optimization*, 2(7-8):649–687, 1980.
- [115] J. Sokolowski and J.-P. Zolesio. *Introduction to Shape Optimization*. Springer Verlag, New York, 1991.
- [116] O. Soto, R. Lohner, and F. Camelli. A linelet preconditioner for incompressible flow solvers. *Int. J. Numer. Meth. Heat Fluid Flow*, 13(1):133–147, 2003.
- [117] P. Spalart and S. Allmaras. A one-equation turbulence model for aerodynamic flows. *AIAA Paper 1992-0439*, 1992.

- [118] S. Ta'asan. Introduction to shape design and control. In *VKI Lecture Series*, Rhode Saint Genese Begium, 1997. von Karman Institute for Fluid Dynamics.
- [119] S. Ta'asan, G. Kuruwila, and M. D. Salas. Aerodynamic design and optimization in one shot. *AIAA Paper 1992-0025*, 1992.
- [120] P. D. Thomas and C. K. Lombard. Geometric conservation law and its application to flow computations on moving grids. *AIAA Journal*, 17(10):1030–1037, October 1979.
- [121] S. Torbert and R. Lohner. Checkpointing schemes for adjoint methods and strongly unsteady flows. *AIAA Paper 2012-0572*, 2012.
- [122] E. F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics: a Practical Introduction*. Springer-Verlag, 1999.
- [123] H. A. Van der Vorst. Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM J. Sci. and Stat. Comput.*, 13(2), 1992.
- [124] E. van der Weide, G. Kalitzin, J. Schluter, and J. J. Alonso. Unsteady turbomachinery computations using massively parallel platforms. *AIAA Paper 2006-0421*, 2006.
- [125] B. van Leer. Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method. *Journal of Computational Physics*, 32(1):101–136, July 1979.
- [126] V. Venkatakrishnan. On the accuracy of limiters and convergence to steady state solutions. *AIAA Paper 1993-0880*, 1993.
- [127] M. Vinokur. An analysis of finite-difference and finite-volume formulations of conservation laws. *Journal of Computational Physics*, 81(1):1–52, March 1989.
- [128] Q. Wang, P. Moin, and G. Iaccarino. Minimal repetition dynamic checkpointing algorithm for unsteady adjoint calculation. *SIAM Journal on Scientific Computing*, 31(4):2549–2567, 2009.
- [129] J. M. Weiss, J. P. Maruszewski, and A. S. Wayne. Implicit solution of the Navier-Stokes equation on unstructured meshes. *AIAA Paper 1997-2103*, 1997.
- [130] P. Wesseling. *Principles of Computational Fluid Dynamics*, volume 29 of *Springer Series in Computational Mathematics*. Springer-Verlag Berlin Heidelberg New York, 2000.
- [131] F. M. White. *Viscous Fluid Flow*. McGraw Hill Inc., New York, 1974.
- [132] D.C. Wilcox. *Turbulence Modeling for CFD*. 2nd Ed., DCW Industries, Inc., 1998.