A Viscous Continuous Adjoint Approach for the Design of Rotating Engineering Applications

Thomas D. Economon[*], Francisco Palacios[†], and Juan J. Alonso[‡]
Stanford University, Stanford, CA 94305, U.S.A.

A viscous continuous adjoint formulation for optimal shape design is developed and applied. The arbitrary Lagrangian-Eulerian version 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 differential geometry is derived. This surface formulation provides the gradient information necessary for performing gradient-based aerodynamic shape optimization. To analyze the effectiveness of the methodology, two design cases in a rotating reference frame are considered. A two-dimensional test case consisting of a rotating airfoil at a low Reynolds number is studied. The shape of the airfoil is then optimized for drag minimization with a geometric constraint. In three-dimensions, the formulation is demonstrated using the well-known NREL Phase VI wind turbine geometry.

Nomenclature

Variable Definition

c	Airfoil chord length
c_p	Specific heat at constant pressure
$\vec{d}$	Force projection vector
$\vec{f}$	Force vector on the surface
j	Scalar function defined at each point on S
$\vec{n}$	Unit normal vector
p	Static pressure
t	Time variable
t_o	Initial time
t_f	Final time
$\vec{u}_\Omega$	Velocity of a moving domain (mesh velocity)
$\vec{v}$	Flow velocity vector
v_∞	Freestream velocity
$\vec{A}^c$	Jacobian of the convective flux with respect to U
$\vec{A}^{vk}$	Jacobian of the viscous fluxes with respect to U
$\bar{\bar{D}}^{vk}$	Jacobian of the viscous fluxes with respect to ∇U
C_f	Skin friction coefficient
C_p	Coefficient of pressure
E	Total energy per unit mass
$\vec{F}^c$	Convective flux
$\vec{F}^c_{ale}$	Convective flux in arbitrary Lagrangian-Eulerian (ALE) form
$\vec{F}^{vk}$	Viscous fluxes
H	Stagnation enthalpy

[*]Ph.D. Candidate, Department of Aeronautics & Astronautics, AIAA Student Member.
[†]Engineering Research Associate, Department of Aeronautics & Astronautics, AIAA Senior Member.
[‡]Associate Professor, Department of Aeronautics & Astronautics, AIAA Associate Fellow.

$\bar{\bar{I}}$	Identity matrix
J	Cost function defined as an integral over S
$\mathcal{J}$	Lagrangian
M_∞	Freestream Mach number
Pr_d	Dynamic Prandtl number
Pr_t	Turbulent Prandtl number
R	Gas constant
$\mathcal{R}(U)$	System of governing flow equations
S	Solid wall flow domain boundary
T	Temperature
$\mathbb{T}$	Time interval, $t_f - t_o$
U	Vector of conservative variables
W	Vector of characteristic variables
γ	Ratio of specific heats, $\gamma = 1.4$ for air
ρ	Fluid density
$\vec{\varphi}$	Adjoint velocity vector
$\bar{\bar{\sigma}}$	Second order tensor of viscous stresses, $\bar{\bar{\sigma}} = \mu^1_{tot}\bar{\bar{\tau}} = \mu^1_{tot}[\nabla\vec{v} + \nabla\vec{v}^\mathsf{T} - \frac{2}{3}\bar{\bar{I}}(\nabla\cdot\vec{v})]$
μ^1_{tot}	Total viscosity as a sum of dynamic and turbulent components, $\mu^1_{tot} = \mu_{dyn} + \mu_{tur}$
μ^2_{tot}	Effective thermal conductivity, $\mu^2_{tot} = \frac{\mu_{dyn}}{Pr_d} + \frac{\mu_{tur}}{Pr_t}$
$\vec{\omega}$	Specified angular velocity vector of a rotating reference frame
Γ	Flow domain boundary
Ψ	Vector of adjoint variables
Ω	Flow domain

Mathematical Notation

$\vec{b}$	Spatial vector $b \in \mathbb{R}^n$, where n is the dimension of the physical cartesian space (in general, 2 or 3)
B	Column vector or matrix B, unless capitalized symbol clearly defined otherwise
$\vec{B}$	$\vec{B} = (B_x, B_y)$ in two dimensions or $\vec{B} = (B_x, B_y, B_z)$ in three dimensions
$\nabla(\cdot)$	Gradient operator
$\nabla\cdot(\cdot)$	Divergence operator
$\partial_n(\cdot)$	Normal gradient operator at a surface point, $\vec{n}_S \cdot \nabla(\cdot)$
$\nabla_S(\cdot)$	Tangential gradient operator at a surface point, $\nabla(\cdot) - \partial_n(\cdot)\vec{n}_S$
$\cdot$	Vector inner product
$\times$	Vector cross product
$\otimes$	Vector outer product
B^T	Transpose operation on column vector or matrix B
$\delta(\cdot)$	Denotes first variation of a quantity

I. Introduction and Motivation

MANY practical flows of aerodynamic interest are unsteady in nature, and between 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. Several examples of engineering applications that could immediately benefit from a truly time-accurate design approach are open rotors, rotorcraft, turbomachinery, wind turbines, 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. In some instances involving 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 which was unsteady in the inertial frame, and can therefore considerably reduce the computational cost of these calculations.

In the context of optimal shape design, adjoint formulations as a means of sensitivity analysis have been the subject of a rich volume of research literature over the past two decades. Many advances and extensions have been made during this period, and the effectiveness of these formulations for use in aerodynamic design, especially for steady problems, is well established.[1–3] Less common and more challenging are adjoint formu-

lations for unsteady problems due to the potentially prohibitive storage requirements associated with storing the time-accurate solution data that is needed for reverse time integration when solving the corresponding adjoint equations. Moreover, the engineering applications mentioned above also involve aerodynamic surfaces that are in motion which must be taken into account by the governing flow equations (including the boundary conditions) and subsequently, by the adjoint equations.

Despite the challenges, recent work demonstrating the viability of unsteady adjoint approaches across a range of applications[4–10] and the aforementioned improvements in computational power and algorithms suggest a growing interest and capability for design in unsteady flows. Several publications have also addressed adjoint-based shape design using the non-inertial governing flow equations.[11–14]

Adjoint formulations are typically classified 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). A large amount of the previous work on unsteady adjoints has been discrete in nature, and while a discrete adjoint approach can often be more straightforward to implement, especially if automatic differentiation is available, we pursue advances in the continuous approach with this article.

The continuous approach results in a set of continuous partial differential equations (PDEs) for the adjoint system which 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 and can be more computationally efficient than solving a potentially large and memory intensive linear system, as required by the discrete adjoint approach. An additional advantage of the continuous approach is the ability to recover a surface formulation for computing gradients. More specifically, the continuous adjoint treatment pursued here is a systematic methodology centered around finding *surface sensitivities* with the aid of differential geometry formulas. This type of surface formulation has no dependence on volume mesh sensitivities and has been successfully applied on three-dimensional, unstructured meshes to full aircraft configurations and even extended to the Reynolds-averaged Navier-Stokes (RANS) equations.[15] Once derived, the continuous adjoint equations, their admissible boundary conditions, and the expressions for surface sensitivities can be easily implemented within existing solvers while leveraging many of the same numerical methods used in the direct problem.

Despite many advantages, continuous adjoint approaches can suffer from issues related to their derivation and implementation. Depending on the governing equations and choice of objective functions, the mathematical manipulations required to arrive at the continuous adjoint system may be quite complicated or even impossible. In particular, deriving consistent boundary conditions and expressions for the surface sensitivity that accompany the continuous adjoint equations can be difficult, and unfortunately, clear strategies for their derivation are less prevalent in the literature. This matter is made even more complicated when the flow is unsteady, the solid walls are moving, or in the presence of source terms.

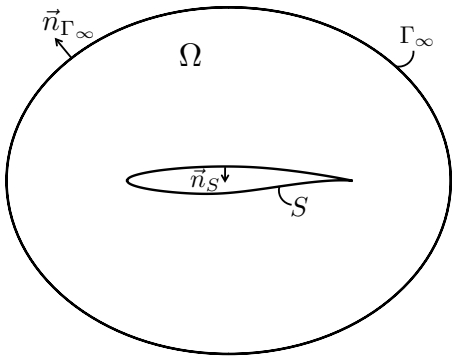


Figure 1. Notional schematic of the flow domain, Ω, the boundaries, Γ_∞ and S, as well as the definition of the surface normals.

However, 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) make the continuous adjoint approach particularly attractive for large-scale optimal shape design problems involving complex geometries. Accordingly, the major contribution of the present article is a detailed derivation of a new continuous adjoint approach for the unsteady RANS equations with a generic source term allowing for surfaces in arbitrary motion and complete with accompanying boundary conditions and surface sensitivity expressions. 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. While the methodology is developed for unsteady flows, the effectiveness of the new methodology is demonstrated by studying two shape design examples in a rotating frame, which can be seen as a straightforward simplification of the general formulation.

The paper is organized as follows. In Section II, a description of the physical problem in which we are in-

terested is given, including the governing flow equations with corresponding boundary conditions. Section III contains a detailed derivation of a viscous continuous adjoint formulation for computing surface sensitivities. Section IV details the numerical implementation of the remaining components needed for automatic shape design: numerical methods, design variables, mesh deformation, and the optimization framework. Lastly, Section V will give results for two- and three-dimensional optimal shape design demonstrations, including the NREL Phase VI wind turbine geometry.

II. Description of the Physical Problem

In this article, we are concerned with time-accurate, viscous flow around aerodynamic bodies in arbitrary motion which is governed by the compressible, unsteady Reynolds-averaged Navier-Stokes (RANS) equations. Consider the equations in a domain, $\Omega \subset \mathbb{R}^3$, with a disconnected boundary that is divided into a far-field component, Γ_∞, and an adiabatic wall boundary, S, as seen in Fig. 1. The surface S represents the outer mold line of an aerodynamic body, and it is considered continuously differentiable (C^1). These conservation equations along with a generic source term, $\mathcal{Q}$, can be expressed in an arbitrary Lagrangian-Eulerian (ALE)[16] differential form as

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

where

$$U = \begin{Bmatrix} \rho \\ \rho\vec{v} \\ \rho E \end{Bmatrix}, \vec{F}^c_{ale} = \begin{Bmatrix} \rho(\vec{v} - \vec{u}_\Omega) \\ \rho\vec{v} \otimes (\vec{v} - \vec{u}_\Omega) + \bar{\bar{I}}p \\ \rho E(\vec{v} - \vec{u}_\Omega) + p\vec{v} \end{Bmatrix}, \vec{F}^{v1} = \begin{Bmatrix} \cdot \\ \bar{\bar{\tau}} \\ \bar{\bar{\tau}} \cdot \vec{v} \end{Bmatrix}, \vec{F}^{v2} = \begin{Bmatrix} \cdot \\ \cdot \\ c_p \nabla T \end{Bmatrix}, \mathcal{Q} = \begin{Bmatrix} q_\rho \\ \vec{q}_{\rho\vec{v}} \\ q_{\rho E} \end{Bmatrix}, \tag{2}$$

ρ is the fluid density, $\vec{v} = \{v_1, v_2, v_3\}^\mathsf{T} \in \mathbb{R}^3$ is the flow speed in a Cartesian system of reference, $\vec{u}_\Omega$ 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, and the viscous stress tensor can be written in vector notation as

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

The second line of Eqn. 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 depending on the sign of the eigenvalues.[17] Note that the boundary conditions take into account any domain motion. The temporal conditions will be problem dependent, and for this article, we will be interested in time-periodic flows where the initial and terminal conditions do not affect the time-averaged behavior over the time interval of interest, $\mathbb{T} = t_f - t_o$. Assuming a perfect gas with a ratio of specific heats, γ, and gas constant, R, the pressure is determined from

$$p = (\gamma - 1)\rho\left[E - \frac{1}{2}(\vec{v} \cdot \vec{v})\right], \tag{4}$$

and the temperature is given by

$$T = \frac{p}{\rho R}. \tag{5}$$

As usual in turbulence modeling that is based upon the Boussinesq hypothesis, which states that the effect of turbulence can be represented as an increased viscosity, the viscosity is divided into a laminar, μ_{dyn}, and a turbulent, μ_{tur}, component. The laminar, or dynamic viscosity, is usually taken to be only dependent on the temperature, $\mu_{dyn} = \mu_{dyn}(T)$, whereas μ_{tur} is obtained from a suitable turbulence model

involving the flow and a set of new variables, $\hat{\nu}$, 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, and in this article, the one-equation Spalart-Allmaras turbulence model[18] is used.

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

$$\mu^1_{tot} = \mu_{dyn} + \mu_{tur}, \qquad \mu^2_{tot} = \frac{\mu_{dyn}}{Pr_d} + \frac{\mu_{tur}}{Pr_t}, \tag{6}$$

where Pr_d and Pr_t are the dynamic and turbulent Prandtl numbers, respectively. Here, μ^2_{tot} represents the effective thermal conductivity that is written in a nonstandard notation to obtain reduced expressions in the calculus below.

When simulating flow about 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 reference frame that rotates with the body of interest. 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 more efficiently, without the need for grid motion. This can be viewed as a simplification of the general unsteady formulation above by choosing

$$\frac{\partial U}{\partial t} = 0, \quad \vec{u}_\Omega = \vec{\omega} \times \vec{r}, \quad \mathcal{Q} = \left\{ \begin{array}{c} \cdot \\ -\rho(\vec{\omega} \times \vec{v}) \\ \cdot \end{array} \right\}, \tag{7}$$

where $\vec{\omega} = \{\omega_x, \omega_y, \omega_z\}^\mathsf{T}$ 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 $\vec{r} = \{(x - x_o), (y - y_o), (z - z_o)\}^\mathsf{T}$. In this case, $\vec{u}_\Omega$ is the velocity due to rotation, which is also sometimes called the whirl velocity.

III. Surface Sensitivities via a Viscous Continuous Adjoint Approach

A typical shape optimization problem seeks the minimization of a certain cost function, $J(S)$, with respect to changes in the shape of the boundary, S. We will concentrate on functionals defined as time-averaged, integrated quantities on the solid surface in the following general form,

$$\begin{array}{ll} \textbf{Minimize} & J(S) = \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S j(\vec{f}, \vec{n})\, ds\, dt \\ \textbf{such that} & \mathcal{R}(U) = 0 \end{array} \tag{8}$$

where $\vec{f} = p\,\vec{n} - \bar{\bar{\sigma}} \cdot \vec{n}$ is the force on the surface, $\bar{\bar{\sigma}} = \mu^1_{tot} \bar{\bar{\tau}}$ is the second order tensor of viscous stresses, and $\vec{n}$ is the outward-pointing unit vector normal to the surface S. Other cost functions are possible (temperature on the surface, for instance), but will not be considered in this article. The minimization of Eqn. 8 can be considered a problem of optimal control whereby the behavior of the governing flow equation system is controlled by the shape of S with deformations of the surface acting as the control input. Therefore, the goal is to compute the variation of Eqn. 8 caused by arbitrary but small (and multiple) deformations of S and to use this information to drive our geometric changes in order to find an optimal shape for the design surface, S. This leads directly to a gradient-based optimization framework. The shape deformations applied to S will be infinitesimal in nature and can be described mathematically by

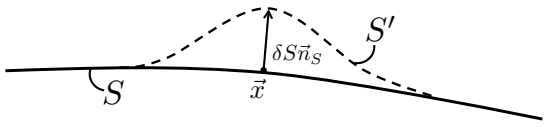


Figure 2. An infinitesimal shape deformation in the local surface normal direction.

$$S' = \{\vec{x} + \delta S \vec{n}_S,\ \vec{x} \in S\}, \tag{9}$$

where S has been deformed to a new surface shape, S', by applying an infinitesimal profile deformation, δS, in the local normal direction, $\vec{n}_S$, at a point, $\vec{x}$, on the surface, as shown in Fig. 2.

Following the adjoint approach to optimal design, Eqn. 8 can be transformed into an unconstrained optimization problem by adding the inner product of an unsteady adjoint variable vector, Ψ, and the governing equations integrated over the domain (space and time) to form the Lagrangian:

$$\mathcal{J} = \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S j(\vec{f}, \vec{n})\, ds\, dt + \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_\Omega \Psi^\mathsf{T} \mathcal{R}(U)\, d\Omega\, dt, \tag{10}$$

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

$$\Psi = \left\{ \begin{array}{c} \psi_\rho \\ \psi_{\rho v_1} \\ \psi_{\rho v_2} \\ \psi_{\rho v_3} \\ \psi_{\rho E} \end{array} \right\} = \left\{ \begin{array}{c} \psi_\rho \\ \vec{\varphi} \\ \psi_{\rho E} \end{array} \right\}. \tag{11}$$

Note that because the flow equations must be satisfied in the domain, or $\mathcal{R}(U) = 0$, Eqn. 8 and Eqn. 10 are equivalent. To find the gradient information needed to minimize the objective function, we take the first variation of Eqn. 10 with respect to small perturbations of the surface shape:

$$\delta\mathcal{J} = \delta J + \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_\Omega \Psi^\mathsf{T} \delta\mathcal{R}(U)\, d\Omega\, dt. \tag{12}$$

A. Variation of the Functional

The first term in Eqn. 12 is the variation of the original objective function, or

$$\delta J = \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_{\delta S} j(\vec{f}, \vec{n})\, ds\, dt + \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \delta j(\vec{f}, \vec{n})\, ds\, dt. \tag{13}$$

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. It can be further simplified using differential geometry formulas:

$$\int_{\delta S} j(\vec{f}, \vec{n})\, ds = \int_S \left(\frac{\partial j}{\partial \vec{f}} \partial_n \vec{f} - 2H_m j \right) \delta S\, ds, \tag{14}$$

$$\begin{aligned} \int_S \delta j(\vec{f}, \vec{n})\, ds &= \int_S \frac{\partial j}{\partial \vec{f}} \delta\vec{f} - \frac{\partial j}{\partial \vec{n}} \cdot \nabla_S(\delta S)\, ds \\ &= \int_S \frac{\partial j}{\partial \vec{f}} \cdot (\delta p \vec{n} - \delta\bar{\bar{\sigma}} \cdot \vec{n}) - \left(\frac{\partial j}{\partial \vec{n}} + \frac{\partial j}{\partial \vec{f}} p - \frac{\partial j}{\partial \vec{f}} \cdot \bar{\bar{\sigma}} \right) \cdot \nabla_S(\delta S)\, ds. \end{aligned} \tag{15}$$

We have used $\int_{\delta S} q\, ds = \int_S [\partial_n(q) - 2H_m q] \delta S\, ds$ where q is an arbitrary scalar, $\delta\vec{n} = -\nabla_S(\delta S)$,[19] which holds for small deformations, and 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. Here, ∇_S represents the tangential gradient operator on S. Assuming that the objective function depends only on $\vec{f}$ in the following way

$$j(\vec{f}) = \vec{f} \cdot \vec{d} \tag{16}$$

where $\vec{d}$ is a constant vector (this is the case in drag or lift optimization problems), after further simplification,[15] the variation of the Lagrangian can be written concisely as

$$\delta\mathcal{J} = \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_S \vec{d} \cdot (\delta p \vec{n} - \delta\bar{\bar{\sigma}} \cdot \vec{n})\, ds\, dt + \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_\Omega \Psi^\mathsf{T} \delta\mathcal{R}(U)\, d\Omega\, dt. \tag{17}$$

B. The Linearized Navier-Stokes Equations

The second term on the right hand side of Eqn. 17 can be expanded by including the version of the governing equations that has been linearized with respect to the small perturbations of the surface (which induce perturbations in U and ∇U). This set of linearized governing equations and boundary conditions is here detailed.

Consider a perturbation to the flow equations while assuming constant, or frozen, viscosity, ($\delta\mu^k_{tot} = 0$):

$$\begin{aligned}
\delta\mathcal{R}(U,\nabla U) &= \delta\left[\frac{\partial U}{\partial t} + \nabla\cdot\vec{F}^c_{ale} - \nabla\cdot\mu^k_{tot}\vec{F}^{vk} - \mathcal{Q}\right] \\
&= \delta\left[\frac{\partial U}{\partial t} + \nabla\cdot\vec{F}^c - \nabla\cdot(U\otimes\vec{u}_\Omega) - \nabla\cdot\mu^k_{tot}\vec{F}^{vk} - \mathcal{Q}\right] \\
&= \frac{\partial}{\partial t}(\delta U) + \nabla\cdot\left(\frac{\partial\vec{F}^c}{\partial U}\delta U\right) - \nabla\cdot\left[\frac{\partial(U\otimes\vec{u}_\Omega)}{\partial U}\delta U\right] - \nabla\cdot\mu^k_{tot}\left[\frac{\partial\vec{F}^{vk}}{\partial U}\delta U + \frac{\partial\vec{F}^{vk}}{\partial(\nabla U)}\delta(\nabla U)\right] - \frac{\partial\mathcal{Q}}{\partial U}\delta U \\
&= \frac{\partial}{\partial t}(\delta U) + \nabla\cdot\left(\vec{A}^c - \bar{\bar{I}}\vec{u}_\Omega - \mu^k_{tot}\vec{A}^{vk}\right)\delta U - \nabla\cdot\mu^k_{tot}\bar{\bar{D}}^{vk}\delta(\nabla U) - \frac{\partial\mathcal{Q}}{\partial U}\delta U,
\end{aligned} \tag{18}$$

where

$$\left.\begin{aligned}
&\vec{A}^c = \left(A^c_x, A^c_y, A^c_z\right), && A^c_i = \left.\tfrac{\partial\vec{F}^c_i}{\partial U}\right|_{U(x,y,z)} \\
&\vec{A}^{vk} = \left(A^{vk}_x, A^{vk}_y, A^{vk}_z\right), && A^{vk}_i = \left.\tfrac{\partial\vec{F}^{vk}_i}{\partial U}\right|_{U(x,y,z)} \\
&\bar{\bar{D}}^{vk} = \begin{pmatrix} D^{vk}_{xx} & D^{vk}_{xy} & D^{vk}_{xz} \\ D^{vk}_{yx} & D^{vk}_{yy} & D^{vk}_{yz} \\ D^{vk}_{zx} & D^{vk}_{zy} & D^{vk}_{zz} \end{pmatrix}, && D^{vk}_{ij} = \left.\tfrac{\partial\vec{F}^{vk}_i}{\partial(\partial_j U)}\right|_{U(x,y,z)}
\end{aligned}\right\} \quad i,j = 1\ldots3, \quad k = 1,2, \tag{19}$$

and these Jacobian matrices can be found in the appendix. Note that in the first line of Eqn. 18, the terms involving the domain velocity have been separated from the traditional inviscid convective fluxes, $\vec{F}^c$. While a frozen viscosity approach has been chosen here, extensions for considering the sensitivity of the viscosity in the presence of a turbulence model (or a simplified approximation) are being pursued. Gathering the final result from Eqn. 18 along with the linearized boundary conditions that can be found in the appendix while imposing that there are no incoming characteristics from the far-field, one obtains the full system of linearized Navier-Stokes equations,

$$\begin{cases}
\delta\mathcal{R}(U) = \frac{\partial}{\partial t}(\delta U) + \nabla\cdot\left(\vec{A}^c - \bar{\bar{I}}\vec{u}_\Omega - \mu^k_{tot}\vec{A}^{vk}\right)\delta U - \nabla\cdot\mu^k_{tot}\bar{\bar{D}}^{vk}\delta(\nabla U) - \frac{\partial\mathcal{Q}}{\partial U}\delta U = 0 & \text{in } \Omega, \qquad t > 0 \\
\delta\vec{v} = -\partial_n(\vec{v} - \vec{u}_\Omega)\delta S & \text{on } S, \\
\partial_n(\delta T) = \nabla T\cdot\nabla_S(\delta S) - \partial^2_n(T)\delta S & \text{on } S, \\
(\delta W)_+ = 0 & \text{on } \Gamma_\infty,
\end{cases} \tag{20}$$

C. The Continuous Adjoint Equations

Eqn. 20 can now be introduced into Eqn. 12 to produce

$$\begin{aligned}
\delta\mathcal{J} = \delta J &+ \frac{1}{\mathbb{T}}\int_{t_o}^{t_f}\int_\Omega \Psi^\mathsf{T}\frac{\partial}{\partial t}(\delta U)\,d\Omega\,dt + \frac{1}{\mathbb{T}}\int_{t_o}^{t_f}\int_\Omega \Psi^\mathsf{T}\nabla\cdot\left(\vec{A}^c - \bar{\bar{I}}\vec{u}_\Omega - \mu^k_{tot}\vec{A}^{vk}\right)\delta U\,d\Omega\,dt \\
&- \frac{1}{\mathbb{T}}\int_{t_o}^{t_f}\int_\Omega \Psi^\mathsf{T}\nabla\cdot\mu^k_{tot}\bar{\bar{D}}^{vk}\delta(\nabla U)\,d\Omega\,dt - \frac{1}{\mathbb{T}}\int_{t_o}^{t_f}\int_\Omega \Psi^\mathsf{T}\frac{\partial\mathcal{Q}}{\partial U}\delta U\,d\Omega\,dt.
\end{aligned} \tag{21}$$

By removing any dependence on variations of the flow variables and their gradients, δU and $\delta(\nabla U)$, the variation of the objective function for multiple surface deformations can be found *without* the need for multiple flow solutions. This results in a computationally efficient method for aerodynamic shape design within a large design space, as the computational cost no longer depends on the number of design variables. We

now perform manipulations to remove this dependence. After changing the order of integration, integrating the second term on the right hand side of Eqn. 21 by parts gives

$$\int_\Omega \int_{t_o}^{t_f} \Psi^\mathsf{T} \frac{\partial}{\partial t}(\delta U)\, dt\, d\Omega = \int_\Omega \left[\Psi^\mathsf{T} \delta U\right]_{t_o}^{t_f} d\Omega - \int_\Omega \int_{t_o}^{t_f} \frac{\partial \Psi^\mathsf{T}}{\partial t} \delta U\, dt\, d\Omega. \tag{22}$$

A zero-valued initial condition for the adjoint variables can be imposed, and assuming an unsteady flow with periodic behavior, the first term on the right hand side of Eqn. 22 can be eliminated with the following temporal conditions (the cost function does not depend on t_f):

$$\Psi(\vec{x}, t_o) = 0, \tag{23}$$

$$\Psi(\vec{x}, t_f) = 0. \tag{24}$$

Now, integrating the third term on the right hand side of Eqn. 21 by parts and using the divergence theorem (assuming a smooth solution) results in

$$\begin{aligned}
&\frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_\Omega \Psi^\mathsf{T} \nabla \cdot \left(\vec{A}^c - \bar{\bar{I}} \vec{u}_\Omega - \mu_{tot}^k \vec{A}^{vk}\right) \delta U\, d\Omega\, dt \\
&\quad = \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_{\partial\Omega} \Psi^\mathsf{T} \left(\vec{A}^c - \bar{\bar{I}} \vec{u}_\Omega - \mu_{tot}^k \vec{A}^{vk}\right) \cdot \vec{n}\, \delta U\, ds\, dt - \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_\Omega \nabla \Psi^\mathsf{T} \cdot \left(\vec{A}^c - \bar{\bar{I}} \vec{u}_\Omega - \mu_{tot}^k \vec{A}^{vk}\right) \delta U\, d\Omega\, dt.
\end{aligned} \tag{25}$$

The third term on the right hand side of Eqn. 21 requires integrating by parts twice, and after the first integration, one recovers the following,

$$\begin{aligned}
&\frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_\Omega \Psi^\mathsf{T} \nabla \cdot \mu_{tot}^k \bar{\bar{D}}^{vk} \delta(\nabla U)\, d\Omega\, dt \\
&\quad = \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_{\partial\Omega} \Psi^\mathsf{T} \mu_{tot}^k \bar{\bar{D}}^{vk} \cdot \delta(\nabla U) \cdot \vec{n}\, ds\, dt - \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_\Omega \nabla \Psi^\mathsf{T} \cdot \left[\mu_{tot}^k \bar{\bar{D}}^{vk} \cdot \delta(\nabla U)\right] d\Omega\, dt.
\end{aligned} \tag{26}$$

Integrating again the final term of Eqn. 26 by parts (while noting that $\delta(\nabla U) = \nabla(\delta U)$ in a continuum) gives

$$\begin{aligned}
&\frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_\Omega \nabla \Psi^\mathsf{T} \cdot \left[\mu_{tot}^k \bar{\bar{D}}^{vk} \cdot \nabla(\delta U)\right] d\Omega\, dt \\
&\quad = \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_{\partial\Omega} \nabla \Psi^\mathsf{T} \cdot \mu_{tot}^k \bar{\bar{D}}^{vk} \delta U \cdot \vec{n}\, ds\, dt - \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_\Omega \nabla \cdot \left(\nabla \Psi^\mathsf{T} \cdot \mu_{tot}^k \bar{\bar{D}}^{vk}\right) \delta U\, d\Omega\, dt.
\end{aligned} \tag{27}$$

Collecting the results from Eqns. 22, 25, 26, and 27 and rearranging terms results in an intermediate expression for the variation of the objective function,

$$\begin{aligned}
\delta \mathcal{J} = \delta J &+ \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_{\partial\Omega} (B_1 - B_2 + B_3)\, ds\, dt \\
&+ \frac{1}{\mathbb{T}} \int_{t_o}^{t_f} \int_\Omega \left[-\frac{\partial \Psi^\mathsf{T}}{\partial t} - \nabla \Psi^\mathsf{T} \cdot \left(\vec{A}^c - \bar{\bar{I}} \vec{u}_\Omega - \mu_{tot}^k \vec{A}^{vk}\right) - \nabla \cdot \left(\nabla \Psi^\mathsf{T} \cdot \mu_{tot}^k \bar{\bar{D}}^{vk}\right) - \Psi^\mathsf{T} \frac{\partial \mathcal{Q}}{\partial U}\right] \delta U\, d\Omega\, dt,
\end{aligned} \tag{28}$$

where, as a shorthand,

$$B_1 = \Psi^\mathsf{T} \left(\vec{A}^c - \bar{\bar{I}} \vec{u}_\Omega\right) \delta U \cdot \vec{n}, \tag{29}$$

$$B_2 = \Psi^\mathsf{T} \mu_{tot}^k \vec{A}^{vk} \delta U \cdot \vec{n} + \Psi^\mathsf{T} \mu_{tot}^k \bar{\bar{D}}^{vk} \cdot \nabla(\delta U) \cdot \vec{n}, \tag{30}$$

$$B_3 = \nabla \Psi^\mathsf{T} \cdot \mu_{tot}^k \bar{\bar{D}}^{vk} \delta U \cdot \vec{n}. \tag{31}$$

By introducing into Eqn. 28 the details of the variation of the functional, δJ, and the evaluation of the boundary integrals involving B_1, B_2, and B_3 (details can be found in the appendix) while assuming the

proper choice of boundary conditions has removed variations of the flow variables at the far-field, a simplified version of $\delta\mathcal{J}$ is recovered:

$$\begin{aligned}\delta\mathcal{J} = &\frac{1}{\mathbb{T}}\int_{t_o}^{t_f}\int_S \vec{d}\cdot\left(\delta p\vec{n} - \delta\bar{\bar{\sigma}}\cdot\vec{n}\right)\,ds\,dt - \frac{1}{\mathbb{T}}\int_{t_o}^{t_f}\int_S \left(\vec{\varphi} + \psi_{\rho E}\,\vec{v}\right)\cdot\left(\delta p\vec{n} - \delta\bar{\bar{\sigma}}\cdot\vec{n}\right)\,ds\,dt \\ &- \frac{1}{\mathbb{T}}\int_{t_o}^{t_f}\int_S \mu_{tot}^2 c_p \partial_n(\psi_{\rho E})\delta T\,ds\,dt - \frac{1}{\mathbb{T}}\int_{t_o}^{t_f}\int_S \frac{\partial\mathcal{J}}{\partial S}\delta S\,ds\,dt \\ &+ \frac{1}{\mathbb{T}}\int_{t_o}^{t_f}\int_\Omega \left[-\frac{\partial\Psi^\mathsf{T}}{\partial t} - \nabla\Psi^\mathsf{T}\cdot\left(\vec{A}^c - \bar{\bar{I}}\vec{u}_\Omega - \mu_{tot}^k\vec{A}^{vk}\right) - \nabla\cdot\left(\nabla\Psi^\mathsf{T}\cdot\mu_{tot}^k\bar{\bar{D}}^{vk}\right) - \Psi^\mathsf{T}\frac{\partial\mathcal{Q}}{\partial U}\right]\delta U\,d\Omega\,dt, \end{aligned} \tag{32}$$

where $\frac{\partial\mathcal{J}}{\partial S}$ is comprised of the remaining boundary terms and will be discussed more below.

Finally, by satisfying the system of PDEs commonly referred to as the adjoint equations along with the admissible adjoint boundary conditions that eliminate any dependence on the fluid flow variations (δp, $\delta\bar{\bar{\sigma}}$, and δT), most of the terms on the right hand side of Eqn. 32 can be eliminated:

$$\begin{cases} -\frac{\partial\Psi^\mathsf{T}}{\partial t} - \nabla\Psi^\mathsf{T}\cdot\left(\vec{A}^c - \bar{\bar{I}}\vec{u}_\Omega - \mu_{tot}^k\vec{A}^{vk}\right) - \nabla\cdot\left(\nabla\Psi^\mathsf{T}\cdot\mu_{tot}^k\bar{\bar{D}}^{vk}\right) - \Psi^\mathsf{T}\frac{\partial\mathcal{Q}}{\partial U} = 0, & \text{in } \Omega, \quad t > 0 \\ \vec{\varphi} = \vec{d} - \psi_{\rho E}\,\vec{v}, & \text{on } S, \\ \partial_n(\psi_{\rho E}) = 0, & \text{on } S. \end{cases} \tag{33}$$

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, reverse time integration is required and the sign of the characteristic velocities is flipped in the adjoint problem, causing characteristic information to propagate in the reverse direction.

D. Surface Sensitivities for Shape Design

After satisfying the adjoint equations, only terms involving the surface shape perturbation, δS, remain (see appendix), and the variation of the objective function becomes

$$\begin{aligned}\delta\mathcal{J} &= -\frac{1}{\mathbb{T}}\int_{t_o}^{t_f}\int_S \frac{\partial\mathcal{J}}{\partial S}\,\delta S\,ds\,dt \\ &= -\frac{1}{\mathbb{T}}\int_{t_o}^{t_f}\int_S \{-(\rho\psi_\rho + \rho\vec{v}\cdot\vec{\varphi} + \rho H\psi_{\rho E})[\partial_n(\vec{v} - \vec{u}_\Omega)\cdot\vec{n}] - \vec{n}\cdot\left(\bar{\bar{\Sigma}}^{\varphi} + \bar{\bar{\Sigma}}^{\psi_{\rho E}}\right)\cdot\partial_n(\vec{v} - \vec{u}_\Omega) \\ &\qquad + \psi_{\rho E}\partial_n(\vec{v} - \vec{u}_\Omega)\cdot\bar{\bar{\sigma}}\cdot\vec{n} + \psi_{\rho E}\left[p(\nabla\cdot\vec{v}) - \bar{\bar{\sigma}} : \nabla\vec{v} + \frac{\partial}{\partial t}(\rho E) + (\vec{q}_{\rho\vec{v}} - \frac{\partial}{\partial t}(\rho\vec{v}))\cdot\vec{v} - q_{\rho E}\right] \\ &\qquad + \mu_{tot}^2 c_p\nabla_S(\psi_{\rho E})\cdot\nabla_S(T)\}\delta S\,ds\,dt, \end{aligned} \tag{34}$$

where $\frac{\partial\mathcal{J}}{\partial S}$ is what we call the *surface sensitivity*, and it is a 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. Note that the final expression for the variation involves only a surface integral and has no dependence on the volume mesh. Furthermore, several new terms appear that directly involve the unsteadiness, source terms, and 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, and further simplifications to the above expression are being pursued. For a steady problem with a fixed surface ($\vec{v} = 0$ on S) and no source terms, this expression reduces to that found previously under the frozen viscosity assumption.[15]

IV. Numerical Implementation

The following sections contain numerical implementation strategies for each of the major components needed for unsteady aerodynamic shape optimization. The optimal shape design loop requires PDE analysis with dynamic meshes for computing functional and sensitivity information, 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 optimization problem.

All components were implemented within the SU^2 software suite (**S**tanford **U**niversity **U**nstructured).[20] 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.

A. Numerical Methods for PDE Analysis

Both the flow and adjoint problems are solved numerically using a Finite Volume Method (FVM) formulation on unstructured meshes with an edge-based structure. The median-dual, vertex-based scheme stores instances of the solution at the nodes of the primal grid and constructs the dual mesh around these nodes by connecting the surrounding cell centers and the mid-points of the edges between the primal grid nodes. The code is fully parallel through use of the Message Passing Interface (MPI) standard and takes advantage of an agglomeration multigrid approach for convergence acceleration.

For the flow equations, the convective flux is discretized using the upwind scheme of Roe,[21] while a non-conservative central scheme with Jameson-Schmidt-Turkel (JST)-type scalar artificial dissipation[22] is used for the discretization of the adjoint convective flux. The convection of the turbulence variable, $\hat{\nu}$, is discretized using a fully upwinded scheme. Second order accuracy is easily achieved via reconstruction of variables on the cell interfaces by using a MUSCL approach with limitation of gradients.[23] In all cases, viscous fluxes are computed with the node-gradient-based approach due to Weiss et al.,[24] which, apart of reducing the truncation error of the scheme, avoids the odd-even decoupling of mesh nodes in the computation of residuals, resulting in second-order spatial accuracy. A weighted least-squares method was used to approximate the spatial gradients of the flow and the adjoint variables. Source terms are approximated via piecewise reconstruction in the finite-volume cells. For the calculations in a non-intertial reference frame, relaxation to a steady-state was accomplished using an implicit, backward-Euler scheme.

B. Design Variable Definition

The time-accurate continuous adjoint derivation 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 design surface. While it is possible to use each surface node in the computational mesh as a design variable capable of deformation, this approach is not often pursued in practice. A more practical choice is to compute the surface sensitivities at each mesh node on the surface and then to project this information into a design space made up of a smaller set of design variables (possibly a complete basis). This procedure for computing the surface sensitivities is used repeatedly in a gradient-based optimization framework in order to march the surface shape toward an optimum through gradient projection and mesh deformation.

In the two-dimensional airfoil calculations that follow, Hicks-Henne bump functions were employed[25] 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

$$f_n(x) = \sin^3(\pi x^{e_n}),\ e_n = \frac{\log(0.5)}{\log(x_n)},\ x \in [0,1], \tag{35}$$

so that the total deformation of the surface can be computed as $\Delta y = \sum_{n=1}^{N} \delta_n f_n(x)$, with N being the number of bump functions and δ_n the design variable step. These functions are applied separately to the upper and lower surfaces.

In three dimensions, a Free-Form Deformation (FFD)[26] strategy has been adopted. 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

$$X(u,v,w) = \sum_{i,j,k=0}^{l,m,n} P_{i,j,k} B_j^l(u) B_j^m(v) B_k^n(w), \tag{36}$$

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. 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. 36.

C. Mesh Deformation

Once the boundary displacements have been computed using either of the above strategies, a technique based on the linear elasticity equations[27] is used to deform the remaining vertices in the unstructured volume mesh. Linear elasticity governs small displacements, $u = (u_1, u_2, u_3)^\mathsf{T}$, of an elastic solid subject to body forces and surface tractions,

$$\nabla\sigma = f \quad \text{in } \Omega, \tag{37}$$

with f being a body force and σ the stress tensor given in terms of the strain tensor, ϵ, by the constitutive relation

$$\sigma = \lambda Tr(\epsilon)I + 2\mu\epsilon, \quad \epsilon = \frac{1}{2}(\nabla u + \nabla u^\mathsf{T}), \quad \lambda = \frac{\nu E}{(1+\nu)(1-2\nu)}, \quad \mu = \frac{E}{2(1+\nu)}, \tag{38}$$

where Tr is the trace, λ and μ are the Lamé constants, 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, small mesh cells near viscous walls will transform more rigidly than larger cells, thus helping to preserve mesh quality in these sensitive regions.

The 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 are applied as a Dirichlet boundary condition. The system of equations is solved iteratively by a Newton-Krylov algorithm. 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.

D. Optimization Framework

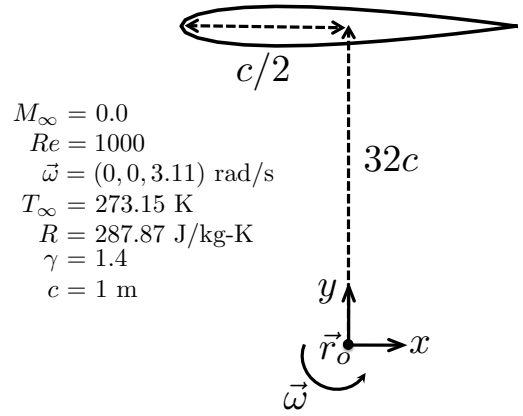
Scripts written in the Python programming language are used to automate execution of the SU^2 suite components, especially for performing shape optimization. The optimization results presented in this work make use of the *SciPy* library (`http://www.scipy.org`), a well-established, open-source software package for mathematics, science, and engineering. The *SciPy* library provides many user-friendly and efficient numerical routines for the solution of non-linear 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.

V. Numerical Results

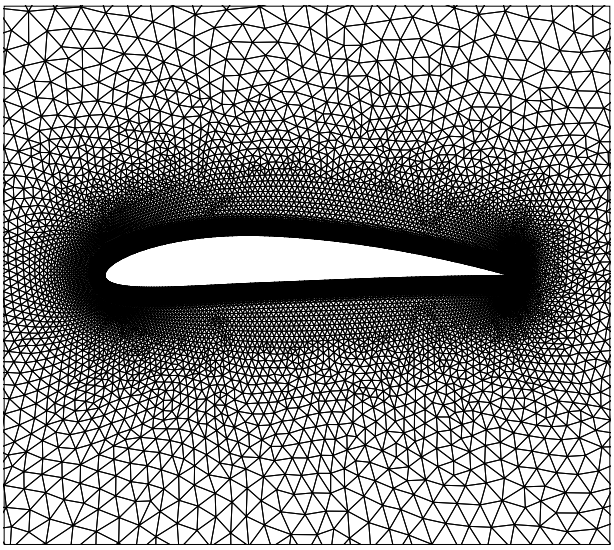
This section contains demonstrations of the new adjoint formulation in both two and three dimensions. In two dimensions, the design of an airfoil that is rotating in still air demonstrates the method, while a redesign of the NREL Phase VI wind turbine illustrates the effectiveness of the new methodology for large-scale, complex engineering systems.

A. Shape Design of a Rotating Airfoil

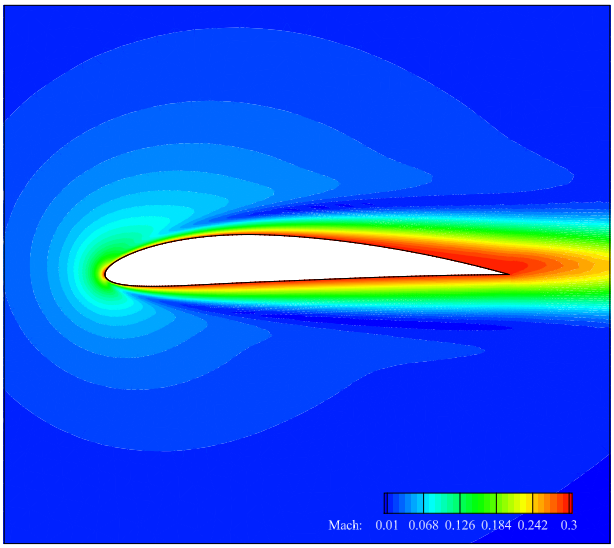
A numerical experiment was devised for an airfoil rotating in still air ($M_\infty = 0$) which can be solved as a steady problem in a rotating reference frame. A low Reynolds number of 1000 was chosen for this case to ensure laminar flow along the airfoil. The goal of this design case is to demonstrate a reduction in skin



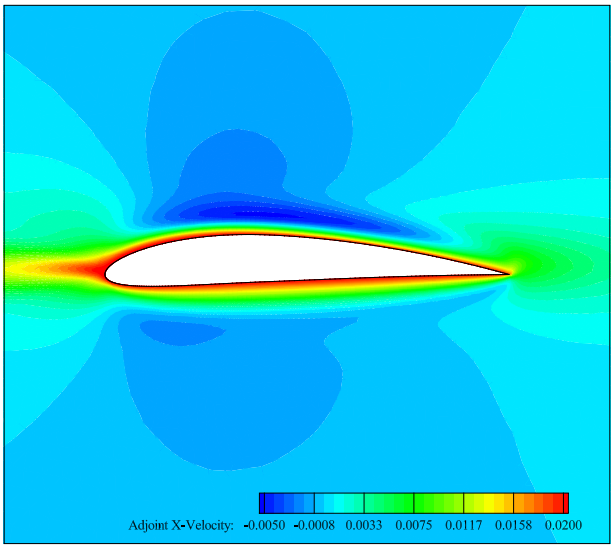
(a) Conditions for the rotating airfoil problem.



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



(c) Absolute Mach number contours.



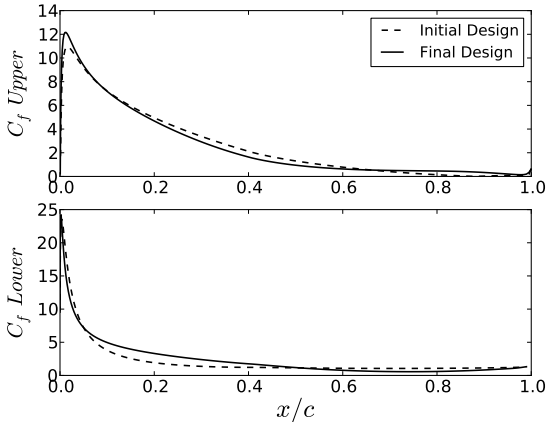
(d) Adjoint x-velocity contours.

Figure 3. Details for the two-dimensional numerical experiment, the computational mesh, and solutions for the initial NACA 4412 geometry.

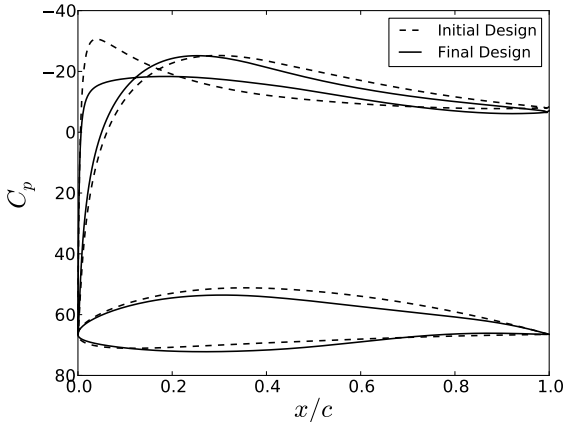
friction drag using the new adjoint formulation while including a realistic geometric constraint. The details for the numerical experiment and the unstructured mesh appear in Fig. 3.

The NACA 4412 profile was chosen as the initial airfoil geometry. A hybrid element mesh was created that consisted of 7,560 quadrilaterals, 24,431 triangles, 19,938 total nodes, 250 edges along the airfoil, and 75 edges along the far-field boundary. The quadrilateral elements were extruded normally from the airfoil surface, and the mesh spacing for this structured region allowed 30 points for adequately resolving the boundary layer.

Fig. 3 shows the absolute Mach number contours around the airfoil. In the inertial frame, the contours show air being pushed out of the path of the rotating airfoil. Fig. 3 also presents contours for $\psi_{\rho v_1}$ near the surface. Note the strong features near the nose in the adjoint solution. Convergence issues can sometimes originate in these regions, but a modified dissipation switch developed in previous work[14] can alleviate the issues by adding extra dissipation only where necessary. However, the switch was not required for this test case.



(a) Skin friction coefficient.



(b) C_p and profile shape comparison for the initial rotating NACA 4412 and the minimum drag airfoil.

Figure 4. Comparison of the initial and final airfoil designs.

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 greater than 12 % of the chord length. A set of 50 Hicks-Henne bump design variables evenly spaced along the upper and lower surfaces were chosen as the design variables. After 10 design cycles, the C_d was successfully reduced from 0.12720 down to 0.12102 (a 4.86 % reduction), and the maximum thickness constraint was met. The value of C_l began at 0.02765 for the initial NACA 4412 design and was 0.03181 for the final design. C_f and C_p distributions as well as the profile shapes of the initial and final designs are compared in Fig. 4. Note that a nonstandard non-dimensionalization due to zero velocity in the free-stream has resulted in large values for the force coefficients.

B. NREL Phase VI Wind Turbine

To demonstrate the effectiveness of the new 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 computational fluid dynamics validation using the data from the NREL Phase VI Unsteady Aerodynamics Experiment.[28,29] The chosen test 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.

The non-inertial governing equations were again used to calculate the flow field around with rotor geometry. However, for this case, the RANS equations with the standard S-A turbulence model were chosen. For validation purposes, Fig. 5 contains the C_p contours on the blade surface, and Fig. 5 gives C_p distributions at two radial stations as computed by SU^2 and compared to experiment. Good agreement is seen overall, apart from near the trailing edge of the blade where some discrepancies are found (large spikes in C_p are also seen at the sharp trailing edge due to the geometry/mesh). More investigation into low-Mach number preconditioning and additional modifications to the S-A model are being pursued to further improve the results. The surface sensitivity was also computed for a torque objective function, and can be seen in Fig. 7. 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.

While a more realistic objective function for wind turbine design might involve total power (and possibly multi-point design), the new methodology is demonstrated with a simple redesign of the rotor blade shape for increasing torque using gradient information obtained via the continuous adjoint approach. In order to redesign the rotor geometry, design variables were defined using a FFD parameterization. First, a box was generated around each of the two blades where shape changes are to be made with the design variables becoming the displacement of the individual control points that define the FFD box. Movements in the vertical direction were allowed for 84 control points on the upper and lower surfaces of each FFD box. In

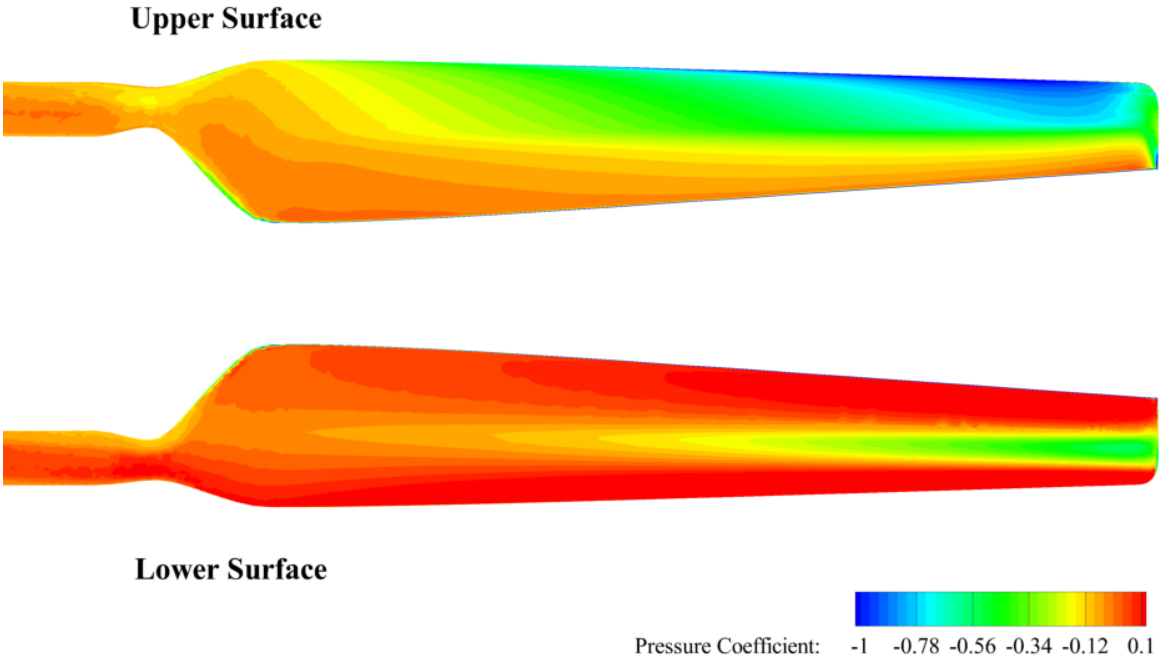
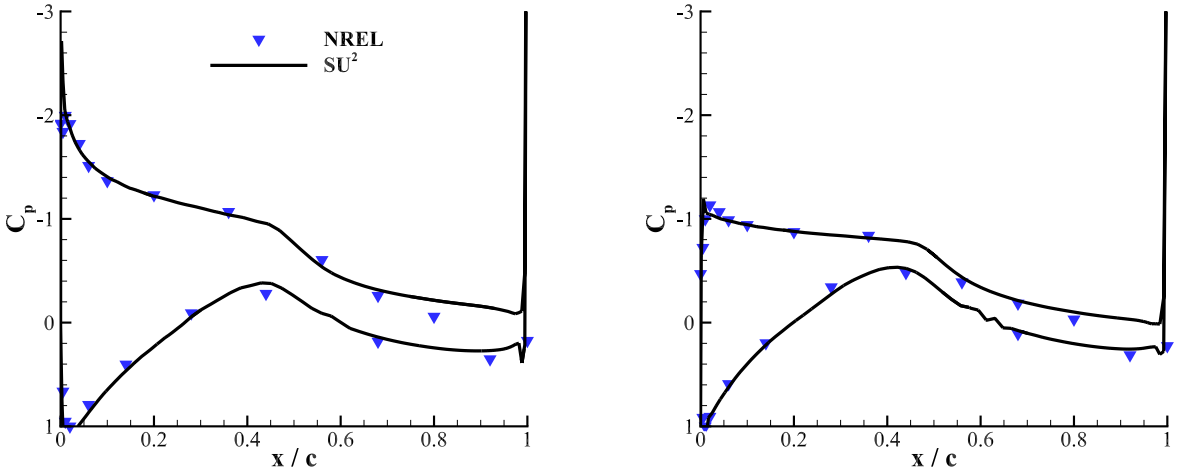


Figure 5. Surface contours of pressure coefficient on the NREL Phase VI wind turbine blade.



(a) $r/R = 0.63$ (b) $r/R = 0.95$

Figure 6. C_p distributions at multiple radial blade stations compared with experimental data.

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, the torque coefficient was increased by 4.0 % from 0.00147 to 0.00153. These optimization results are presented in Fig. 8, including a comparison of the initial and final surface shapes.

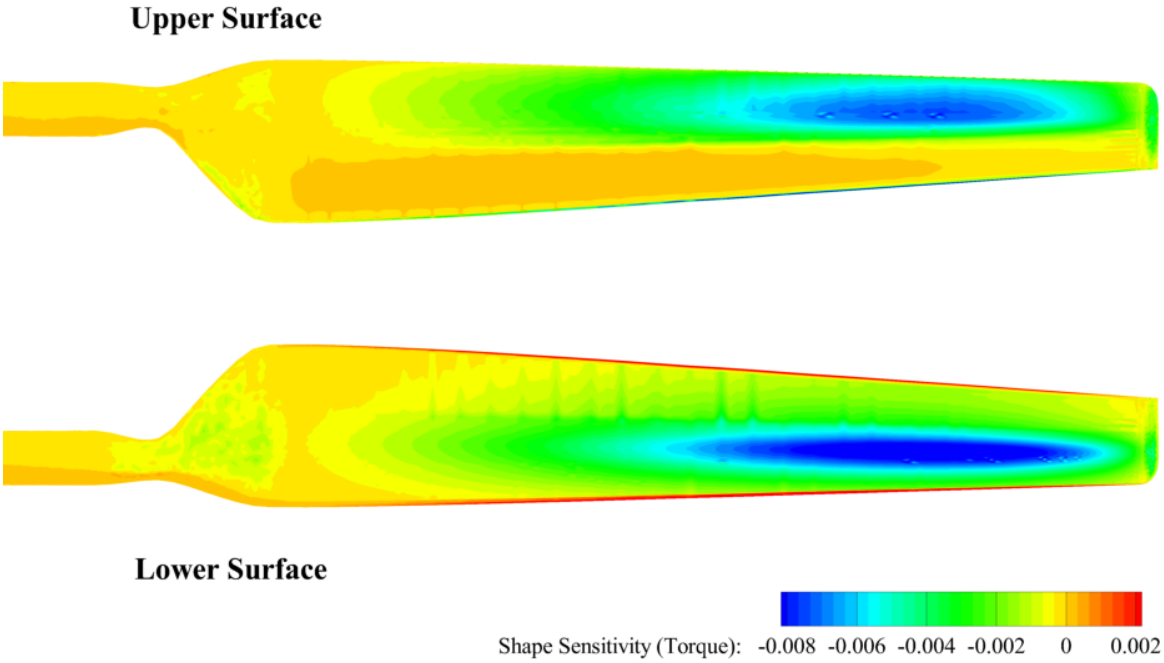


Figure 7. Surface sensitivity contours on the NREL Phase VI wind turbine blade.

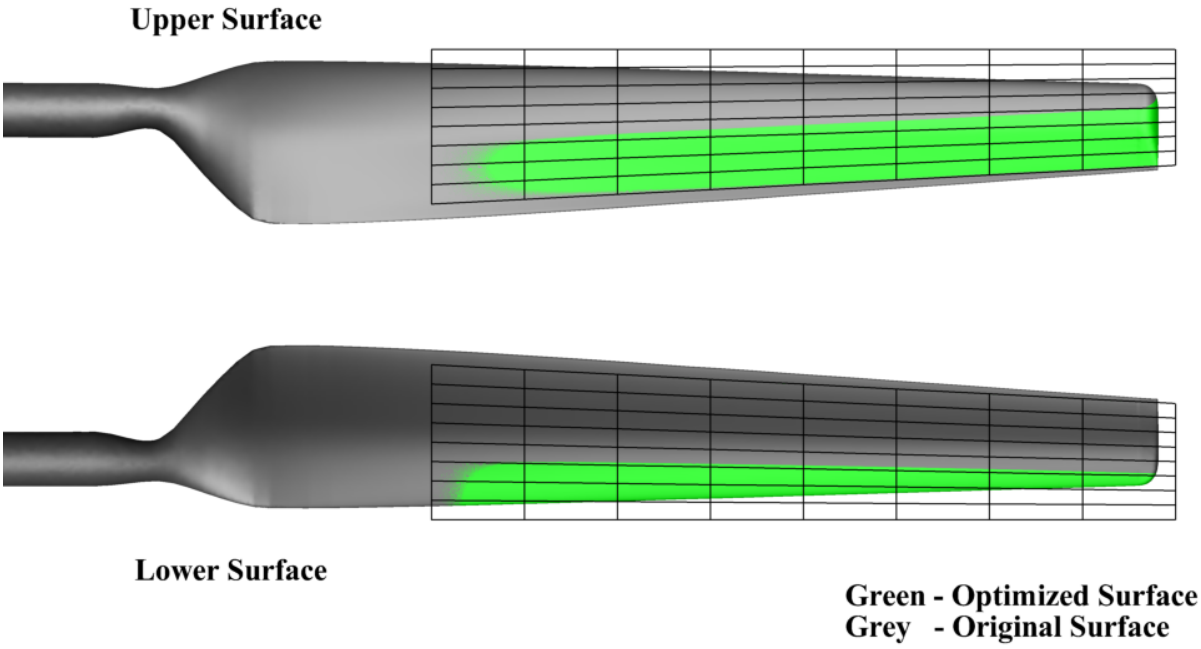


Figure 8. FFD box and initial and final shape for the wind turbine blade.

VI. Conclusions

A viscous continuous adjoint formulation for optimal shape design has been developed and applied. The arbitrary Lagrangian-Eulerian version of the unsteady, compressible RANS equations with a generic source

term is considered, and from these governing flow equations, a new continuous adjoint formulation was developed complete with accompanying boundary conditions and surface sensitivity expressions. The new formulation allows for the design of surfaces in arbitrary motion.

The effectiveness of the new methodology is demonstrated by studying two shape design examples in a rotating frame which can be seen as a straightforward simplification of the general formulation. Skin friction drag was successfully reduced by 4.86 % for a rotating airfoil at a Reynolds number of 1000 while satisfying a maximum thickness constraint. In three-dimensions, the formulation was demonstrated through an application to the well-known NREL Phase VI wind turbine geometry. This large-scale test case shows the applicability of the new formulation to the design of aerospace systems with realistic, complex geometries.

While the methodology is demonstrated with the non-intertial governing equations, the formulation was developed for general unsteady flows with dynamic meshes. Future work includes the straightforward application of the time-accurate adjoint and surface sensitivity for large-scale, fully unsteady problems. The main difference here will be the required computational resources in terms of computational effort and time-accurate data storage. Another area for future work is the removal of the frozen viscosity assumption by considering sensitivities in the viscosity, possibly through the treatment of a turbulence model.

VII. Acknowledgements

T. Economon would like to acknowledge U.S. government support under and awarded by DoD, Air Force Office of Scientific Research, National Defense Science and Engineering Graduate (NDSEG) Fellowship, 32 CFR 168a. The authors would also like to thank Karthik Duraisamy and Mark Potsdam for access to the computational mesh for the NREL turbine geometry.

References

[1]Jameson, A., "Aerodynamic Design Via Control Theory," *AIAA 81-1259*, 1981.

[2]Jameson, A., Alonso, J. J., Reuther, J., Martinelli, L., Vassberg, J. C., "Aerodynamic Shape Optimization Techniques Based On Control Theory," *AIAA-1998-2538*, 29th Fluid Dynamics Conference, Albuquerque, NM, June 15-18, 1998.

[3]Anderson, W. K. and Venkatakrishnan, V., "Aerodynamic Design Optimization on Unstructured Grids with a Continuous Adjoint Formulation," *Journal of Scientific Computing*, Vol. 3, 1988, pp. 233-260.

[4]Nadarajah, S. K., Jameson, A., "Optimum Shape Design for Unsteady Flows with Time-Accurate Continuous and Discrete Adjoint Methods," *AIAA Journal*, Vol. 45, No. 7, pp. 1478-1491, July 2007.

[5]Rumpfkeil, M. P., and Zingg, D. W., A General Framework for the Optimal Control of Unsteady Flows with Applications, *AIAA Paper 2007-1128*, 2007.

[6]Mavriplis, D. J., "Solution of the Unsteady Discrete Adjoint for Three-Dimensional Problems on Dynamically Deforming Unstructured Meshes, *AIAA Paper 2008-727*, 2008.

[7]Mani, K., and Mavriplis, D. J., "Unsteady Discrete Adjoint Formulation for Two-Dimensional Flow Problems with Deforming Meshes, *AIAA Journal*, Vol. 46, No. 6, pp. 13511364, 2008.

[8]Nielsen, E. J., Diskin, B., Yamaleev, N. K., "Discrete Adjoint-Based Design Optimization of Unsteady Turbulent Flows on Dynamic Unstructured Grids," *AIAA Journal*, Vol. 48, No. 6, pp. 1195-1206, June 2010.

[9]Nielsen, E. J., Diskin, B., "Discrete Adjoint-Based Design for Unsteady Turbulent Flows On Dynamic Overset Unstructured Grids," *AIAA-2012-0554*, 50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, Nashville, Tennessee, Jan. 9-12, 2012.

[10]Economon, T. D., Palacios, F., Alonso, J. J., "Unsteady Aerodynamic Design on Unstructured Meshes with Sliding Interfaces," *AIAA 2013-0632*, 51st AIAA Aerospace Sciences Meeting and Exhibit, January 7-10, 2013. Grapevine, Texas, USA.

[11]Lee, S. W., Kwon, O. J., "Aerodynamic Shape Optimization of Hovering Rotor Blades in Transonic Flow Using Unstructured Meshes," *AIAA Journal*, Vol. 44, No. 8, pp. 1816-1825, August, 2006.

[12]Nielsen, E. J. Lee-Rausch, E. M. Jones, W. T., "Adjoint-Based Design of Rotors using the Navier-Stokes Equations in a Noninertial Reference Frame," AHS International 65th Forum and Technology Display, Grapevine, TX, May 27-29, 2009.

[13]Dumont, A., Le Pape, A., Peter, J., Huberson, S., "Aerodynamic Shape Optimization of Hovering Rotors Using a Discrete Adjoint of the Reynolds-Averaged Navier-Stokes Equations," *Journal of the American Helicopter Society*, Vol. 56, No. 3, pp. 1-11, July, 2011.

[14]Economon, T. D., Palacios, F., Alonso, J. J., "Optimal Shape Design for Open Rotor Blades," *AIAA-2012-3018*, 30th AIAA Applied Aerodynamics Conference, New Orleans, Louisiana, June 25-28, 2012.

[15]Bueno-Orovio, A., Castro, C., Palacios, F., and Zuazua, E., "Continuous Adjoint Approach for the Spalart-Allmaras Model in Aerodynamic Optimization," *AIAA Journal* , Vol. 50, No. 3, pp. 631-646, March 2012.

[16]Donea, J., Huerta, A., Ponthot, J.-Ph., Rodriguez-Ferran, A., "Arbitrary Lagrangian-Eulerian Methods," Encyclopedia of Computational Mechanics, Vol. 1, 2004.

[17]Hirsch, C., "Numerical Computation of Internal and External Flows," Wiley, New York, 1984.

[18]Spalart, P., and Allmaras, S., "A One-Equation Turbulence Model for Aerodynamic Flows," *AIAA-1992-0439*, 1992.

[19]Sokolowski, J. Zolesio, J.-P., *Introduction to Shape Optimization*, Springer Verlag, New York, 1991.

[20]Palacios, F., Colonno, M. R., Aranake, A. C., Campos, A., Copeland, S. R., Economon, T. D., Lonkar, A. K., Lukaczyk, T. W., Taylor, T. W. R., Alonso, J. J., "Stanford University Unstructured (SU^2): An open-source integrated computational environment for multi-physics simulation and design.," *AIAA-2013-0287*, 51st AIAA Aerospace Sciences Meeting and Exhibit, January 7-10, 2013, Grapevine, Texas, USA.

[21]P. L. Roe, "Approximate riemann solvers, parameter vectors, and difference schemes," *Journal of Computational Physics*, 43:357-372, 1981.

[22]Jameson, A., Schmidt, W., and Turkel, E., "Numerical Solution of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes," *AIAA 81-1259*, 1981.

[23]Venkatakrishnan, V., "On the Accuracy of Limiters and Convergence to Steady State Solutions," *AIAA-1993-0880*, 1993.

[24]J. M. Weiss, J. P. Maruszewski, and A. S. Wayne, "Implicit solution of the Navier-Stokes equation on unstructured meshes," *AIAA-1997-2103*, 1997.

[25]Hicks, R. and Henne, P., "Wing design by numerical optimization, *Journal of Aircraft*, Vol. 15, pp. 407-412, 1978.

[26]Samareh, J. A., "Aerodynamic shape optimization based on Free-Form deformation, *AIAA-2004-4630*, 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Albany, New York, Aug. 2004.

[27]Johnson A. A., Tezduyar, T. E., "Simulation of multiple spheres falling in a liquid-filled tube," *Computer Methods in Applied Mechanics and Engineering*, Vol. 134, Issues 3-4, pp. 351-373, August 1996.

[28]Potsdam, M., A., Mavriplis, D. A., "Unstructured Mesh CFD Aerodynamic Analysis of the NREL Phase VI Rotor," *AIAA 2009-1221*, 47th AIAA Aerospace Sciences Meeting Including The New Horizons Forum and Aerospace Exposition, 5 - 8 January 2009, Orlando, Florida.

[29]Simms, D. A., Schreck, S., Hand, M., and Fingersh, L., J., "NREL Unsteady Aerodynamics Experiment in the NASA-Ames Wind Tunnel: A Comparison of Predictions to Measurements," NREL/TP-500-29494, June 2001.

[30]Castro, C., Lozano, C., Palacios, F., and Zuazua, E., "A Systematic Continuous Adjoint Approach to Viscous Aerodynamic Design on Unstructured Grids," *AIAA Journal*, Vol. 45, No. 9, pp. 21252139, 2007.

A. Jacobian Matrices

Using index notation and defining for convenience $a_0 = (\gamma - 1)$, $\phi = (\gamma - 1)\frac{|\vec{v}|^2}{2}$, the Jacobian matrices are defined as:

$$A_i^c = \begin{pmatrix} \cdot & \delta_{i1} & \delta_{i2} & \delta_{i3} & \cdot \\ -v_i v_1 + \delta_{i1}\phi & v_i - (a_0 - 1)v_i\delta_{i1} & v_1\delta_{i2} - a_0 v_2\delta_{i1} & v_1\delta_{i3} - a_0 v_3\delta_{i1} & a_0\delta_{i1} \\ -v_i v_2 + \delta_{i2}\phi & v_2\delta_{i1} - a_0 v_1\delta_{i2} & v_i - (a_0 - 1)v_i\delta_{i2} & v_2\delta_{i3} - a_0 v_3\delta_{i2} & a_0\delta_{i2} \\ -v_i v_3 + \delta_{i3}\phi & v_3\delta_{i1} - a_0 v_1\delta_{i3} & v_3\delta_{i2} - a_0 v_2\delta_{i3} & v_i - (a_0 - 1)v_i\delta_{i3} & a_0\delta_{i3} \\ v_i\left(\phi - H\right) & -a_0 v_i v_1 + H\delta_{i1} & -a_0 v_i v_2 + H\delta_{i2} & -a_0 v_i v_3 + H\delta_{i3} & \gamma v_i \end{pmatrix}$$

$$A_i^{v1} = \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot \\ -\eta_{i1} & \partial_i\left(\frac{1}{\rho}\right) + \frac{1}{3}\partial_1\left(\frac{1}{\rho}\right)\delta_{i1} & \partial_1\left(\frac{1}{\rho}\right)\delta_{i2} - \frac{2}{3}\partial_2\left(\frac{1}{\rho}\right)\delta_{i1} & \partial_1\left(\frac{1}{\rho}\right)\delta_{i3} - \frac{2}{3}\partial_3\left(\frac{1}{\rho}\right)\delta_{i1} & \cdot \\ -\eta_{i2} & \partial_2\left(\frac{1}{\rho}\right)\delta_{i1} - \frac{2}{3}\partial_1\left(\frac{1}{\rho}\right)\delta_{i2} & \partial_i\left(\frac{1}{\rho}\right) + \frac{1}{3}\partial_2\left(\frac{1}{\rho}\right)\delta_{i2} & \partial_2\left(\frac{1}{\rho}\right)\delta_{i3} - \frac{2}{3}\partial_3\left(\frac{1}{\rho}\right)\delta_{i2} & \cdot \\ -\eta_{i3} & \partial_3\left(\frac{1}{\rho}\right)\delta_{i1} - \frac{2}{3}\partial_1\left(\frac{1}{\rho}\right)\delta_{i3} & \partial_3\left(\frac{1}{\rho}\right)\delta_{i2} - \frac{2}{3}\partial_2\left(\frac{1}{\rho}\right)\delta_{i3} & \partial_i\left(\frac{1}{\rho}\right) + \frac{1}{3}\partial_3\left(\frac{1}{\rho}\right)\delta_{i3} & \cdot \\ v_j\pi_{ij} & v_j\partial_j\left(\frac{1}{\rho}\right)\delta_{i1} + \zeta_{i1} + \frac{1}{\rho}\tau_{i1} & v_j\partial_j\left(\frac{1}{\rho}\right)\delta_{i2} + \zeta_{i2} + \frac{1}{\rho}\tau_{i2} & v_j\partial_j\left(\frac{1}{\rho}\right)\delta_{i3} + \zeta_{i3} + \frac{1}{\rho}\tau_{i3} & \cdot \end{pmatrix}$$

$$A_i^{v2} = \gamma \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \frac{1}{a_0}\partial_i\left(\frac{\phi}{\rho} - \frac{p}{\rho^2}\right) & -\partial_i\left(\frac{v_1}{\rho}\right) & -\partial_i\left(\frac{v_2}{\rho}\right) & -\partial_i\left(\frac{v_3}{\rho}\right) & \partial_i\left(\frac{1}{\rho}\right) \end{pmatrix}$$

$$D_{ii}^{v1} = \frac{1}{\rho}\begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot \\ -\left(1 + \frac{1}{3}\delta_{i1}\right)v_1 & \left(1 + \frac{1}{3}\delta_{i1}\right) & \cdot & \cdot & \cdot \\ -\left(1 + \frac{1}{3}\delta_{i2}\right)v_2 & \cdot & \left(1 + \frac{1}{3}\delta_{i2}\right) & \cdot & \cdot \\ -\left(1 + \frac{1}{3}\delta_{i3}\right)v_3 & \cdot & \cdot & \left(1 + \frac{1}{3}\delta_{i3}\right) & \cdot \\ -|\vec{v}|^2 - \frac{1}{3}v_i^2 & \left(1 + \frac{1}{3}\delta_{i1}\right)v_1 & \left(1 + \frac{1}{3}\delta_{i2}\right)v_2 & \left(1 + \frac{1}{3}\delta_{i3}\right)v_3 & \cdot \end{pmatrix}$$

$$D_{ij}^{v1} = \frac{1}{\rho}\begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot \\ -v_i\delta_{j1} + \frac{2}{3}v_j\delta_{i1} & \delta_{j1}\delta_{i1} - \frac{2}{3}\delta_{i1}\delta_{j1} & \delta_{j1}\delta_{i2} - \frac{2}{3}\delta_{i1}\delta_{j2} & \delta_{j1}\delta_{i3} - \frac{2}{3}\delta_{i1}\delta_{j3} & \cdot \\ -v_i\delta_{j2} + \frac{2}{3}v_j\delta_{i2} & \delta_{j2}\delta_{i1} - \frac{2}{3}\delta_{i2}\delta_{j1} & \delta_{j2}\delta_{i2} - \frac{2}{3}\delta_{i2}\delta_{j2} & \delta_{j2}\delta_{i3} - \frac{2}{3}\delta_{i2}\delta_{j3} & \cdot \\ -v_i\delta_{j3} + \frac{2}{3}v_j\delta_{i3} & \delta_{j3}\delta_{i1} - \frac{2}{3}\delta_{i3}\delta_{j1} & \delta_{j3}\delta_{i2} - \frac{2}{3}\delta_{i3}\delta_{j2} & \delta_{j3}\delta_{i3} - \frac{2}{3}\delta_{i3}\delta_{j3} & \cdot \\ -\frac{1}{3}v_iv_j & v_j\delta_{i1} - \frac{2}{3}v_i\delta_{j1} & v_j\delta_{i2} - \frac{2}{3}v_i\delta_{j2} & v_j\delta_{i3} - \frac{2}{3}v_i\delta_{j3} & \cdot \end{pmatrix} \quad (i \neq j)$$

$$D_{ii}^{v2} = \frac{\gamma}{\rho}\begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \frac{1}{a_0}\left(\phi - \frac{p}{\rho}\right) & -v_1 & -v_2 & -v_3 & 1 \end{pmatrix}$$

$$D_{ij}^{v2} = \mathbf{0}_{5\times 5} \quad (i \neq j)$$

where tensors $\bar{\eta}$, $\bar{\pi}$ and $\bar{\zeta}$ in the definition of A_i^{v1} are given by

$$\begin{aligned} \eta_{ij} &= \partial_i\left(\frac{v_j}{\rho}\right) + \partial_j\left(\frac{v_i}{\rho}\right) - \frac{2}{3}\delta_{ij}\nabla\cdot\left(\frac{\vec{v}}{\rho}\right) \\ \pi_{ij} &= v_j\partial_i\left(\frac{1}{\rho}\right) + v_i\partial_j\left(\frac{1}{\rho}\right) - \frac{2}{3}\delta_{ij}\,\vec{v}\cdot\nabla\left(\frac{1}{\rho}\right) = \eta_{ij} - \frac{1}{\rho}\tau_{ij} \\ \zeta_{ij} &= v_j\partial_i\left(\frac{1}{\rho}\right) - v_i\partial_j\left(\frac{1}{\rho}\right) + \frac{1}{3}v_i\partial_j\left(\frac{1}{\rho}\right). \end{aligned}$$

The source term Jacobian for the flow equations expressed in a rotating frame is

$$\frac{\partial \mathcal{Q}}{\partial U} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\omega_z & \omega_y & 0 \\ 0 & \omega_z & 0 & -\omega_x & 0 \\ 0 & -\omega_y & \omega_x & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{39}$$

B. Linearized Navier-Stokes Boundary Conditions

The following sections contain details on the linearization of the Navier-Stokes boundary conditions.

A. No-Slip Solid Wall

The linearized boundary conditions will also be required in order to remove any dependence on flow variations. The details for linearizing the no-slip wall boundary condition are given here. We start with the no-slip boundary condition for a surface in arbitrary motion:

$$(\vec{v} - \vec{u}_\Omega) = 0 \text{ on } S, \tag{40}$$

where $\vec{v}$ is the absolute flow velocity and $\vec{u}_\Omega$ is the local velocity of the domain in motion. Consider linearization with respect to small perturbations in the surface, δS,

$$(\vec{v} - \vec{u}_\Omega)' = (\vec{v} - \vec{u}_\Omega) + \delta(\vec{v} - \vec{u}_\Omega) + \partial_n(\vec{v} - \vec{u}_\Omega)\delta S, \tag{41}$$

where the second term on the right hand side of Eqn. 41 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. Keeping in mind that the linearized boundary condition still must equal zero, Eqn. 41 can be rearranged to give a useful result for the continuous adjoint derivation:

$$\delta\vec{v} = -\partial_n(\vec{v} - \vec{u}_\Omega)\delta S, \tag{42}$$

where in order to simplify we have used the original boundary condition (Eqn. 40) and $\delta\vec{u}_\Omega = 0$.

B. Adiabatic Wall

In this work, we consider only an adiabatic condition on solid walls, although other conditions, such as isothermal walls, are possible. The details for linearizing the adiabatic wall boundary condition are given here. We start with the adiabatic wall boundary condition which is unaffected by any motion of the surface:

$$\partial_n T = \nabla T \cdot \vec{n}_S = 0 \text{ on } S, \tag{43}$$

where T is the temperature and $\vec{n}_S$ is the local wall unit normal. Consider linearization with respect to small perturbations in the surface, δS, for the both temperature and the normal terms separately,

$$(\nabla T)' = \nabla T + \delta(\nabla T) + \partial_n(\nabla T)\delta S, \tag{44}$$

$$(\vec{n}_S)' = \vec{n}_S + \delta\vec{n}_S, \tag{45}$$

where the second term on the right hand side of Eqn. 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 of Eqn. 45 does not involve the flow equations, so the only change is due to the deformation. The complete linearized boundary condition can be obtained by taking the dot product of the two linearized components,

$$\begin{aligned}(\nabla T)' \cdot (\vec{n}_S)' &= [\nabla T + \delta(\nabla T) + \partial_n(\nabla T)\delta S] \cdot (\vec{n}_S + \delta\vec{n}_S) \\ &= \nabla T \cdot \delta\vec{n}_S + \delta(\nabla T) \cdot \vec{n}_S + \partial_n^2(T)\delta S,\end{aligned} \tag{46}$$

where in order to simplify we have used the original boundary condition (Eqn. 43) and the approximation that any products of variations are negligible. Keeping in mind that the linearized boundary condition still must equal zero, Eqn. 45 can be rearranged as

$$\delta(\nabla T) \cdot \vec{n}_S = -(\nabla T) \cdot \delta\vec{n}_S - \partial_n^2(T)\delta S. \tag{47}$$

Finally, using $\delta\vec{n}_S = -\nabla_S(\delta S)$, which holds for small deformations, and the fact that in a continuum $\delta(\nabla T) = \nabla(\delta T)$, gives a useful result for the continuous adjoint derivation:

$$\partial_n(\delta T) = \nabla T \cdot \nabla_S(\delta S) - \partial_n^2(T)\delta S. \tag{48}$$

C. Evaluation of the Boundary Integral Terms for the Adjoint Derivation

The reduced expressions for the evaluations of the boundary integral terms appearing during the continuous adjoint derivation (B_1, B_2, and B_3) are here presented. Given our knowledge of the Jacobian matrices evaluated on the surface using both the flow boundary conditions and the linearized boundary conditions, the following expressions are obtained:

$$\begin{aligned}B_1 &= \Psi^\mathsf{T}\left(\vec{A}^c - \bar{\bar{I}}\vec{u}_\Omega\right)\delta U \cdot \vec{n} \\ &= -(\rho\psi_\rho + \rho\vec{v}\cdot\vec{\varphi} + \rho H\psi_{\rho E})[\partial_n(\vec{v} - \vec{u}_\Omega)\delta S \cdot \vec{n}] + [\vec{\varphi}\cdot\vec{n} + \psi_{\rho E}(\vec{v}\cdot\vec{n})]\delta p,\end{aligned} \tag{49}$$

$$\begin{aligned}B_2 &= \Psi^\mathsf{T}\mu_{tot}^k\vec{A}^{vk}\delta U \cdot \vec{n} + \Psi^\mathsf{T}\mu_{tot}^k\bar{\bar{D}}^{vk} \cdot \nabla(\delta U) \cdot \vec{n} \\ &= \vec{\varphi}\cdot\delta\bar{\bar{\sigma}}\cdot\vec{n} + \psi_{\rho E}\vec{v}\cdot\delta\bar{\bar{\sigma}}\cdot\vec{n} - \psi_{\rho E}\partial_n(\vec{v} - \vec{u}_\Omega)\delta S \cdot \bar{\bar{\sigma}}\cdot\vec{n} + \psi_{\rho E}\mu_{tot}^2 c_p[\nabla T \cdot \nabla_S(\delta S) - \partial_n^2(T)\delta S],\end{aligned} \tag{50}$$

and

$$B_3 = -\vec{n}\cdot\left(\bar{\bar{\Sigma}}^{\varphi} + \bar{\bar{\Sigma}}^{\psi_{\rho E}}\right)\cdot\partial_n(\vec{v} - \vec{u}_\Omega)\delta S + \mu_{tot}^2 c_p\partial_n(\psi_{\rho E})\delta T, \tag{51}$$

where $\bar{\bar{\Sigma}}^{\varphi} = \mu_{tot}^1(\nabla\vec{\varphi} + \nabla\vec{\varphi}^\mathsf{T} - \frac{2}{3}\bar{\bar{I}}\nabla\cdot\vec{\varphi})$, and $\bar{\bar{\Sigma}}^{\psi_{\rho E}} = \mu_{tot}^1(\nabla\psi_{\rho E}\vec{v} + \nabla\psi_{\rho E}\vec{v}^\mathsf{T} - \frac{2}{3}\bar{\bar{I}}\nabla(\psi_{\rho E})\cdot\vec{v})$. By using the governing equations written on the surface and integration by parts, the terms involving the second order derivative of temperature can be reduced to a more computable form:[30]

$$\begin{aligned}&\psi_{\rho E}\mu_{tot}^2 c_p[\nabla T \cdot \nabla_S(\delta S) - \partial_n^2(T)\delta S] \\ &= \{-\psi_{\rho E}[p(\nabla\cdot\vec{v}) - \bar{\bar{\sigma}} : \nabla\vec{v} + \frac{\partial}{\partial t}(\rho E) + (\vec{q}_{\rho\vec{v}} - \frac{\partial}{\partial t}(\rho\vec{v}))\cdot\vec{v} - q_{\rho E}] - \mu_{tot}^2 c_p\nabla_S(\psi_{\rho E})\cdot\nabla_S(T)\}\delta S,\end{aligned} \tag{52}$$

where $\bar{\bar{\sigma}} : \nabla\vec{v} = \sigma_{ij}\partial_i v_j$.